Technische Universität München
TUM School of Computation, Information and Technology

Robust Multipath-based Localization in Dynamic Indoor Environments

Alexandra Vadymivna Zayets, M. Sc.

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades einer

Doktorin der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. Wolfgang Utschick
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Eckehard Steinbach
	2. Prof. Dr. Christoph Günther

Die Dissertation wurde am 22.02.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 10.11.2022 angenommen.

Abstract

With the fast evolution of smart buildings, automated factories and warehouses, indoor localization systems are becoming more popular. Indoor localization is needed to navigate a smartphone-user or a robot through a large building. It is also needed to locate resources and equipment in a warehouse or factory. While Global Positioning System (GPS) signals are effective at localizing an electromagnetic receiver outdoors, they are attenuated indoors. A number of high-precision indoor localization systems (ILS) exist, however their performance suffers when the environment is complex, contains non-line-of-sight (NLoS) conditions and moving objects. ILSs that measure the distances or angles between its access points (APs)/routers and the user require a line-of-sight (LoS) between the user and the APs. In order for such systems to maintain a high localization accuracy in a complex environment, a very large number of APs needs to be installed, increasing the system's cost. In contrast, fingerprinting ILS collect calibration measurements, construct a fingerprint map and afterwards compare the user's measurements to that map to localize him. While such systems require less APs, perform well under NLoS conditions, the necessary calibration measurements take a long time to collect, again increasing system cost. Additionally, if any changes or movement occur in the environment the user's measurements will no longer be similar to the map and the localization accuracy will decrease.

The main contribution of this thesis is the development of a fingerprinting-based localization approach that functions even when there are moving objects in the environment. This is achieved by the use of multipath delay profile fingerprints. A fast database look-up scheme is also designed for the proposed localization algorithm. In addition, a novel communication protocol between the user and the ILS that protects the user's privacy is proposed in this thesis. With the help of the presented approach, the user can obtain his location from the ILS without the ILS knowing what that location is or being able to track the user. Another contribution of this work is the development of a virtual transmitter-based fingerprint interpolation scheme for multipath fingerprints. The interpolation scheme allows fewer calibration measurements to be made. This significantly reduces the cost of the ILS. Lastly, an algorithm that reconstructs a 3D model of the environment from a multipath fingerprint map is developed. The concept of virtual-transmitters is also used by the proposed approach. To summarize, this thesis shows the potential and advantages of the use of the multipath delay profile for indoor localization.

Kurzfassung

Das Aufgabenfeld der Indoor-Lokalisierung wird benötigt, um Menschen mithilfe von Smartphones oder Robotern durch größere Gebäudekomplexe zu navigieren. Gleichermaßen ist es relevant, um Ressourcen und Equipment in Lagern und Fabriken zu lokalisieren. Generell werden für diese Aufgabe Signale des globalen Positionsbestimmungssystems (GPS) außerhalb von Gebäuden genutzt, allerdings sind diese in Innenbereichen von Gebäuden stark gedämpft. Eine Vielzahl präziser Indoor-Lokalisierungssystem (ILS) existieren, jedoch sinkt deren Leistungsfähigkeit in komplexen Umgebungen mit Nicht-Sichtverbindung (NLoS) Bedingungen und sich bewegenden Objekten. ILSs, welche auf Messungen der Distanzen oder den Winkeln zwischen Access Points (APs) oder Routern und der Nutzer basieren, benötigen direkten Sichtkontakt. Um eine hohe Lokalisierungsgenauigkeit derartiger Systeme sicherzustellen, wird eine große Anzahl installierter APs benötigt. Diese erhöhen somit die Systemkosten. Im Gegensatz dazu nutzen Fingerprinting-basierte ILSs Kalibrierungsmessungen, konstruieren eine Fingerprint-Karte und vergleichen Nutzermessungen zu dieser für eine Lokalisierung. Diese Systeme benötigen vergleichsweise weniger APs und zeigen hinreichende Genauigkeit unter NLoS Bedingungen, jedoch benötigen Kalibrierungsmessungen eine vergleichsweise längere Zeit und erhöhen somit ebenfalls die Systemkosten. Außerdem führen Bewegungen oder Änderungen in der Umgebung dazu, dass die Nutzermessungen nicht mehr ähnlich zu der gemessenen Karte sind, womit die Genauigkeit reduziert wird.

Diese Dissertation präsentiert einen Fingerprinting-basierten Lösungsansatz, der in komplexen Umgebungen auch sich-bewegende Objekte berücksichtigt. Hierfür werden Mehrwegverzögerungsprofil Fingerprints ausgewertet. Ein schnelleres Datenbank Look-up Schema wurde mitentwickelt und ausgewertet. Außerdem wurde das Kommunikationsprotokoll so überarbeitet, dass es vertrauliche Informationen des Nutzers schützt. Der ILS hat demnach keine Möglichkeit, die Position des Nutzers zu verfolgen. Als weitere Neuerung wird ein Interpolationsschema für Mehrweg Fingerprints vorgestellt, welches weniger Kalibrierungsmessungen benötigt und somit die Kosten des ILS senkt. Abschließend wurde ein Algorithmus entwickelt, der ein 3D-Modell aus dem Datenbank von Mehrweg Fingerprints, basierend auf virtuellen Sendern, erstellt. In Hinblick auf diese erreichten Punkte zeigt die vorliegende Dissertation das Potential und die Vorteile von Mehrwegprofilen für die Indoor-Lokalisierung.

Acknowledgements

First and foremost, all the praises belong to Allah, the greatest and the exalted, on whom ultimately, we depend for sustenance and guidance. I thank Him for bestowing upon me the opportunity, strength and faculties to undertake this research work.

Second of all, I would like to express my deep gratitude to my supervisor Prof. Dr. Eckehard Steinbach for providing me with the opportunity to conduct research in his group at the Chair of Media Technology (LMT) at the Technical University of Munich. I would like to thank him for his invaluable guidance, support, extensive feedback and helpful discussions. I am also deeply grateful to my all fellow teammates at LMT for their help, cooperation and numerous stimulating discussions which benefited my research work greatly. Amongst them, I always felt at home. My sincere thanks especially goes to Dr. Matti Strese, Dr. Dima Bobkov, Jingyi Xu, Dr. Elena Nölke, Martin Piccolrovazzi, Mojtaba Karimi and Michael Adam.

I would also like to thank Dr. Christian Gentner from the Institute of Communications and Navigation at the German Aerospace Center (DLR) for his time, help and support during the creation of the MCA prototype and the validation measurements. I would also like to thank him for his assistance with the journal publication.

Furthermore, I would like to thank Prof. Dr. Christoph Günther for agreeing to become the second examiner and Prof. Dr. Wolfgang Utschick for chairing the thesis committee.

Last but not the least, I would like to thank my family to whom I owe everything. They stood with me at every step of my life and longed to see this achievement come true. I dedicate this work to them, in particular to my affectionate father and loving mother, supportive husband Ovais, and caring sister Sonia.

Contents

Notation

Abbreviations

Abbreviation	Description	Definition
1D	One-dimensional	page 16
2D	Two-dimensional	page 16
3D	Three-dimensional	page 16
BW	Bandwidth	page 51
GNSS	Global navigation satellite system	page 7
GPS	Global positioning system	page 7
RF	Radio frequency	page 7
APs	Access points	page 1
ILS	Indoor localization system	page 1
AoA	Angle of arrival	page 12
SLAM	Simultaneous localization and mapping	page 10
EKF	Extended Kalman Filter	page 11
CFR	Channel frequency response	page 13
CIR	Channel impulse response	page 13
CSI	Channel state information	page 13
DFT	Discreet Fourier transform	page 35
FFT	Fast Fourier transform	page 56
ILS	Indoor localization system	page 1
LoS	Line-of-sight	page 16
NLoS	Non-line-of-sight	page 1
RSSI	Received signal strength indicator	page 12
ToA	Time of arrival	page 12
TDoA	Time difference of arrival	page 12
ToF	Time-of-flight	page 12
KNN	K-nearest neighbors	page 28
AWGN	Additive white Gaussian noise	page 12
LS	Least squares algorithm	page 16
USRP	Universal software radio peripheral	page 56
IWL5300	Intel N Ultimate WiFi Wireless Link 5300	page 54
NIC	Network interface card	page 55
IPIN	International conference on indoor positioning and navigation	page 58
AP	Access point	page 8
DM	Diffuse multipath	page 56
MPC	Multipath component	page 79
MDP	Multipath delay profile	page 80

Abbreviation	Description	Definition
MCA	Multipath component analysis	page 81
IMU	Inertial measurement unit	page 10
PDR	Pedestrian dead reckoning	page 10
OFDM	Orthogonal frequency-division multiplexing	page 13
MDP	Multipath delay profile	page 80
MCA	Multipath component analysis	page 81
VT	Virtual transmitter	page 127
VA	Virtual anchor	page 127
ES	Expected scatterer	page 128
BLE	Bluetooth low energy	page 51
WSN	Wireless sensor network	page 51
RFID	Radio frequency identification device	page 52
UWB	Ultra wide-band	page 51
LE	Low energy	page 50
PCA	Principle component analysis	page 30
EM	Expectation minimization	page 90
DCT	Discrete cosine transform	page 30
DWT	Discrete wavelet transform	page 38
ML	Maximum likelihood	page 30
PFD	Probability density function	page 66
CDF	Cumulative density function	page 66
MSE	Mean squared error	page 63
MAE	Mean absolute error, mean average error	page 63
RMSE	Root mean squared error	page 63
f/m^2	Fingerprints per m^2	page 116
Lidar	Light detection and ranging	page 142
SAGE	space-alternating generalized expectation-maximization (algorithm)	page 89
IR	Infra-red	page 142
CE	Coverage error	page 146
GC	Geometry coverage	page 146

Terminology

Terminology	Definition
Access points/transmitters	Transmitters placed throughout the environment with which the receiver communicates to obtain signal measurements and calculate its location. The APs may or may not be used by the receiver to communicate with the ILS server.
Receiver	An electromagnetic receiver carried by the user that communicates with the APs in order to determine its location. Also referred to as a tag.
Tag	Synonymous with receiver.
ToF; ToA; TDoA	Time the signal takes to travel from an AP receiver to the receiver; time at which the signal arrives at the receiver, according to the receiver's clock; the difference between the times at which the signals from two APs arrive at the receiver.
AoA	The incident angle at which the signal from an AP arrives at a receiver.
Multipath component	The length of one propagation path, line-of-sight or containing reflections, that the signal travels from an AP to the receiver. Also used to describe the delay corresponding to the propagation path.
Propagation path	A path traveled by the signal from the transmitter to the receiver. Normally there are multiple propagation paths between a transmitter and receiver.
Multipath delay profile RSSI	Signal power in dB at the receiver.
Channel impulse response	Function that characterizes the electromagnetic channel between a transmitter and receiver. The Received signal is equal to the channel impulse response convoluted with the original signal.
Channel frequency response	Frequency domain representation of the channel impulse response.
Channel state information	Term that can either described the received power at each OFDM carrier or a characterization of the channel such as the channel impulse or frequency response or multipath profile.
ILS server	Stores the fingerprint map, compares the query fingerprint measured by the user to the fingerprint map, runs the localization algorithm.

Terminology	Definition
Fingerprint	Signal characteristics which are calculated from the signals received by the user and compared to the database. The the database contains fingerprints measured when the localization system was installed.
Reference fingerprint	Fingerprint stored in the database
Query fingerprint	Fingerprint measured by the user to determine his or her location.
Reference point	Location where a reference fingerprint was recorded
Query point	Location of the user
Labeled fingerprint	A reference or query fingerprint for which the location at which the fingerprint was measured is known.
Fingerprint map	A database of all reference fingerprints together with the locations where they were measured.
Fingerprint density	Number of reference points per square meter.
Target location	A location at which a fingerprint is inter- or extrapolated.
Calibration phase/off-line/training phase	Process of recording reference fingerprints before the ILS can be deployed.
On-line phase	In the on-line phase the fingerprint map has already been created and the user can communicate with the ILS server and run the localization algorithm.
Similarity metric	Scalar value calculated for a query and a reference fingerprint that represents how similar the two fingerprints are.
Partial similarity metric	A similarity metric value the computation of which is not yet completed. The optimized MCA algorithm updates the partial similarity metrics of the candidate fingerprints at each algorithm iteration.
PDR	The process of calculating the user's location using his motion data.
Odometry	Motion data of a robot with wheels, usually obtained from the wheels motion.
Principle component analysis	Machine learning technique to map data onto a lower dimensional subspace or find useful features in the data
Discrete cosine transform	Transform applied to a signal which is used for decorrelating the data.
Independent component analysis	transform that makes the components of the non-Gaussian data as independent as possible

Terminology	Definition
Ground-truth	The correct (x,y,z) coordinates at which measurements were performed. In general ground-truth is the set of correct values to which the measurement of estimation results are compared to evaluate a system.
k-anonymity	Privacy protection scheme, where the users obscure their IDs from the server or randomly generate pseudonym IDs.
Paillier cryptosystem	Encoding scheme that allows arithmetic operations to be performed on encoded data
Encoding based privacy protection	Privacy protection schemes where the user encodes his or her measured fingerprint and the server encodes the reference fingerprints in the database and performs calculations with encoded data.
Camouflage privacy protection	Privacy protection scheme, where the user generates camouflage fingerprints and sends them to the ILS together with the measured fingerprint.
Full Fingerprint	Complete set of multipath components or the complete information extracted from the signals received from the APs by the user.
Partial fingerprint	Subset of multipath components or of the information extracted from the signals received by the user from the APs.
Camouflage fingerprint	Fake fingerprints artificially constructed by the user and set to the ILS.
Candidate locations	The set of the possible location estimates that the ILS server calculates and sends to the user.
Candidate fingerprint	In the context of privacy protection these are the reference fingerprints corresponding to the candidate locations. The user selects his or her location by running the MCA algorithm locally on the candidate fingerprints. In the context of the optimized MCA algorithm the candidate fingerprints are the fingerprints marked by the algorithm as potential locations of the user. The similarity metrics corresponding to the candidate fingerprints are updated and new fingerprints may be added to the candidate set at each iteration of the algorithm.

Terminology	Definition
Virtual Transmitter (VT)	The reflection of a transmitter (AP) that is located behind the wall or obstacle as shown in Fig. 6.1. The length of the reflected path from the AP to the user is the same as the length of the direct path from the user to the VT. The degree of a VT is defined as the number of time the signal from the AP would be reflected before it reaches the user. It is also the number of reflection steps needed to calculate the VT's position. Also known as a virtual anchor (VA).
Virtual Anchor (VA)	Same as virtual transmitter.
Expected scatterer (ES)	VT equivalent used to model scattering. The length of a propagation path is equal to the distance between a point and the ES. However, unlike the case of reflection, the position of the ES is different for each point at which the scattered signal is received.
Target point	The point at which an interpolated or extrapolated fingerprint is calculated.
Interpolated fingerprint	A fingerprint that is calculated from a set set of measured fingerprints that surround the target point.
Extrapolated fingerprint	A fingerprint that is calculated from a set of measured fingerprints that correspond to reference points which do not surround the target point.
Geometry polygons	It is assumed that the indoor environments is composed of reflective *polygons*. Each polygon is located in one plane and can have an arbitrary number of corner points. Multiple polygons can be located in one plane and can overlap.
Geometry deviation	Metric to evaluate the reconstructed geometry, equal to the average distance from the vertices of the reconstructed polygons to the projections projections of those vertices onto the planes in the ground-truth geometry.
Geometry coverage	The percentage of the area of the ground truth geometry that is covered by the projections of the reconstructed polygons. This is the metric used to determined the percentage of correctly reconstructed geometry.
Coverage error	Percentage of the area of the reconstructed polygons that does not coincide with the ground truth geometry.

Scalars and vectors

x	scalar

$\lvert x \rvert$	absolute value of scalar x
$\hat{x}$	estimated/predicted value of x
$f(\cdot)$	scalar function
$\boldsymbol{x}$	array
$\overline{\overline{\boldsymbol{x}}} = N$	cardinality, amount of elements in array $\boldsymbol{x}$
x_i	i-th component of vector $\boldsymbol{x}$
$\lVert \boldsymbol{x} \rVert$	Euclidean norm of vector $\boldsymbol{x}$
$\bar{\boldsymbol{x}}$	mean of $\boldsymbol{x}$
$\sigma(\boldsymbol{x}) = \sigma_{\boldsymbol{x}}$	standard deviation of $\boldsymbol{x}$
$\boldsymbol{x}(\cdot)$	vector function
$\boldsymbol{x} \cdot \boldsymbol{y}$	dot product of $\boldsymbol{x}$ and $\boldsymbol{y}$
$\boldsymbol{X}$	[M x N] matrix
$\boldsymbol{X}(\cdot)$	matrix function
$\mathbf{X}$	set
$\mathbf{X}(i)$	ith element of a set
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	scalar product between two vectors $\boldsymbol{x}$, $\boldsymbol{y}$
$\hat{\boldsymbol{X}}$	estimated value of a variable X

Symbols

Symbol	Definition
$\mathbb{R}^2$	2D space of real numbers.
σ	standard deviation
$s(t)$	Signal at the transmitter.
$x(t)$	Signal at the receiver.
$h(t)$	Channel impulse response between the receiver and the transmitter.
a_k	Attenuation undergone by the signal along the k-th propagation path from the transmitter to the receiver
τ_k	The delay of a multipath component
d_k	The length of a propagation path
c	The speed of light
$n(t)$	Noise added at the receiver
P_T	Transmit power of a signal.
N_{σ^2}	AWGN noise with variance σ^2.
$\boldsymbol{Y}(t)$	Total received signal in the SAGE algorithm.
$Y_j(t)$	The signal received by the j antenna

Symbol	Definition
$\boldsymbol{X}_k(t)$ and $\boldsymbol{x}_k(t)$	The signals received from the k-th propagation path. Notation used in the SAGE algorithm.
d_0	Distance a signal travels from a transmitter to the receiver. Either the
	LoS or the length of the smallest reflected path.
K_L	Number of multipath components estimated by the SAGE algorithm from a received signal.
$\mathbf{MAP} = \{(\mathbf{D}_i, \boldsymbol{X}_i), i = 1...N\}$	fingerprint map.
N	Number of reference fingerprints stored in the fingerprint map.
M	Number of APs.
i	Index used to describe a reference fingerprint or location.
j	Index used to describe an AP.
k	Index used to describe a multipath component.
$\boldsymbol{T}_j$	j-th transmitter/AP.
$\boldsymbol{T}'_{j,k}$	k-th first degree (1 reflection) virtual transmitter corresponding to AP $\boldsymbol{T}_j$.
$\boldsymbol{T}''_{j,k}$	k-th second degree (2 reflections) virtual transmitter corresponding to AP $\boldsymbol{T}_j$.
$\boldsymbol{T}'^{(l)}_{j,k}$	k-th l-th degree (l reflections) virtual transmitter corresponding to AP $\boldsymbol{T}_j$.
$\boldsymbol{X}_i$	Coordinates of the i-th reference point.
$\boldsymbol{X}$	True location of the user.
$\hat{\boldsymbol{X}}$	Estimate of the user′s location calculated by the ILS.
$\mathbf{D}_i = \{\boldsymbol{d}_1, ..., \boldsymbol{d}_M\}$	i-th reference fingerprint.
$\boldsymbol{d}_{ij} = [d_{ij,1}, ..., d_{ij,K}] \in \mathbf{D}_i$	Vector containing the multipath components corresponding to the j-th AP in the i-th reference fingerprint.
$d_{ij,k} \in \boldsymbol{d}_{ij}$	k-th multipath component that correspond to the j-th AP in the i-th reference fingerprint.
K_{ij}	Number of propagation paths between a receiver $\boldsymbol{T}_j$ and point $\boldsymbol{X}_i$.
$\mathbf{D}^q$	Query fingerprint measured by the user.
$\boldsymbol{d}^q_j \in \mathbf{D}^q$	Vector containing the signal propagation distances from the query point to the j-th AP.
$d^q_{j,k} \in \boldsymbol{d}^q_j$	k-th multipath component that corresponds to the j-th AP in the query fingerprint.

Symbol	Definition
K_j^q	Number of multipath components in the query fingerprint that correspond to the j-th AP.
$\mathbf{D}^i$	Fingerprint interpolated at the target location.
$\boldsymbol{d}_j^i \in \mathbf{D}^i$.	Vector containing multipath components interpolated at the target point that correspond to the k-th AP.
$d_{j,k}^i \in \boldsymbol{d}_j^i$	k-th multipath component that corresponds to the j-th AP in the fingerprint interpolated at the target point.
d^*	Element of the vector $\boldsymbol{d}_{ij}$ that is closest to the value $d_{j,k}^q$
$\gamma(\mathbf{D}^q, \mathbf{D}_i)$	Similarity metric calculated between the query and the i-th reference fingerprint.
$\gamma(\mathbf{D}^q, \mathbf{D}_i \| \boldsymbol{T}_j)$	Similarity metric between two fingerprints $\mathbf{D}^q$ and $\mathbf{D}_i$ calculated for transmitter $\boldsymbol{T}_j$
$\mathbf{Q}_{ji}$	A set of all indexes k for which $\exists d^* \in \boldsymbol{d}_{ij}$,$\|d_{jk} - d^*\| \rightarrow min$ and $\|d_{jk} - d^*\| < \varepsilon$
ε	Similarity threshold
N_p	Number of partial fingerprints generated by the user
np	Length of a partial fingerprint
N_c	Number of *fake* camouflage fingerprints generated by the user to one *real* partial fingerprint
$N_T = N_p + N_p \times N_c$	Total number of partial fingerprints sent by the user to the server
$\{\mathbf{D}_{p1}^q, ..., \mathbf{D}_{pN_p}^q\}$	*Real* partial fingerprints
$\{\mathbf{D}_{c1}^q, ..., \mathbf{D}_{c(N_T-N_p)}^q\}$	Camouflage partial fingerprints
κ	Number of candidates returned by the server for each partial fingerprint
$\{\mathbf{D}_{ci}\}$	Candidate fingerprints returned by the server
$\{\boldsymbol{X}_{ci}\}$	Candidate locations that correspond to the candidate fingerprints
$\|d_{ci}\|$	The distances from each candidate point to the nearest other candidate point
$f_{d_c}(d)$	Probability distribution of $\|d_{ci}\|$
$\mathbf{L}(\mathbf{D}_i)$	Set of partial fingerprints for which the server selected $\mathbf{D}_i$ as a candidate
N_i	Number of times reference fingerprint $\mathbf{D}_i$ is picked by the server as a candidate to a partial fingerprint sent by the user within one localization step
d_L	Average distance the user moves between two consecutive localization requests.
$\boldsymbol{B}$	Sorted fingerprint database

Symbol	Definition
$\boldsymbol{bin}(j,x)$	Sorted data structure that contains all of the reference multipath components, the values of which are in the interval $((x-1)\varepsilon, x\varepsilon]$ and which correspond to the AP $\boldsymbol{T}_j$
d_{max}	The largest value of a multipath component that can be stored in the database
(j,k)	Iteration of the optimized MCA algorithm at which the query multipath component $d^q_{j,k}$ corresponding to transmitter $\boldsymbol{T}_j$ is used to update the similarity metric values.
$\gamma_{i,jk}$	Partial similarity metric calculated for fingerprint $\mathbf{D_i}$ at the (j,k)-th algorithm iteration. This means that the similarity metrics does not yet include the data from all query multipath components.
$\boldsymbol{f}$	Set of all candidate fingerprints for the optimized MCA algorithm
$\boldsymbol{\gamma}_{(j,k)}$	Set of partial similarity metric values corresponding to the candidate fingerprints in the optimized MCA algorithm
$\gamma^{'}_{i,jk}$	Update term added to the partial similarity metric $\boldsymbol{\gamma}_{(j,k)}$ at the (j,k)-th algorithm iteration.
$\boldsymbol{l}$	Sorted array that contains all the multipath components in the search area of the query multipath component at a iteration of the optimized MCA.
γ_{thr}	Threshold value such that the partial similarity metric values $\gamma^{'}ij,k$ of the neighbors of the best match to $d^q_{j,k}$ in $\boldsymbol{l}$ don't need to be calculated if they are larger than γ_{thr} (Algorithm 5).
$\gamma^{'}_{min,jk}$	Partial similarity metric calculated for reference multipath component most similar to the query multipath component at the (j,k)-th iteration of the algorithm. As the multipath component is most similar to the query, $\gamma^{'}_{min,jk}$ will actually be the largest partial similarity metric value calculated in one iteration. However, the notation min is kept for consistency with [2].
$\boldsymbol{T^*_{j,k}}$	k-th candidate VT corresponding to AP $\boldsymbol{T}_j$ calculated by the VT reconstruction algorithm. The candidate VT may not be in the set of real VTs. A filtering step is needed to determine the set of estimated VTs from the set of candidate VTs.
γ_{VT}	Threshold parameter used in VT filtering (Algorithm 9).

Symbol	Definition
$\boldsymbol{Y}_i$	Reference locations, the fingerprints of which are used to calculate the interpolated fingerprint.
ε_c	Parameter used for multipath component clustering (Algorithm 7)).
$\varepsilon(\boldsymbol{c}_1, \boldsymbol{c}_2)$	similarity metric between two clusters of multipath components (Algorithm 8).
α	Parameter used by the cluster matching algorithm (Algorithm 8).
ε_{group}	Threshold parameter used by the cluster matching algorithm (Algorithm 8).
$\mathbf{G}_i$	$i-th$ ground truth polygon in the indoor geometry.
$\hat{\boldsymbol{G}}_{i,l}$	l-th reconstructed polygon that corresponds to the real polygon $\mathbf{G}_i$.
$\hat{\boldsymbol{G}}_i(l)$	l-th vertex of the i-th reconstructed polygon.
$P(\hat{\mathbf{G}}_i(l), \mathbf{G}_i)$	the projection of point $\hat{\mathbf{G}}_i(l)$ onto the corresponding ground-truth plane $\mathbf{G}_i$.
δG	Value of the geometry deviation metric.
$\triangle\,\mathbf{G}_i$	The area of the polygon $\mathbf{G}_i$.
GC	Value of geometry coverage metric.
CE	Value of the coverage error metric.
$perimeter$	Value of the perimeter coverage metric.
$Perimeter(\mathbf{G}_i)$	The length of the line created by projecting the vertical polygon $\mathbf{G}_i$ onto the floor-plane.

Chapter 1

Introduction

Autonomous robots need to know their precise location in order to navigate through a building and to interact with the objects that building contains. Boxes and equipment need to be located on request in a warehouse or factory. Smartphone apps need to know the location of the user in order to guide him or her through a shopping mall or to provide location specific content, such as sales deals and augmented reality. The location of the robot, equipment and user is provided by an indoor localization system (ILS) .

The goal of this thesis is to develop an ILS that is able to to maintain a high accuracy in complex non-line-of-sight (NLoS) environments that contain moving objects and that can change over time. The ILS should consist of a set of radio frequency (RF) transmitters or access points (APs) and a target device, or receiver, that is carried by the user and communicates with the APs. It should be possible to obtain the location of a moving user and of a static object, for example a box to which a receiver is attached.

1.1 Motivation

Indoor localization is a service that is in high demand both for user-oriented smartphone applications and in industrial systems. Objects that need to be localized include people carrying smartphones, robots and mobile equipment and boxes. A small receiver device can easily be attached to all of them, communicate with a set of access points (APs) or electromagnetic transmitters and calculate its location. The same principle is applied for outdoor localization when a GPS receiver is attached to the object that needs to be localized.

Localization is much more challenging inside buildings than in outdoor environments. GPS signals are blocked and absorbed by building walls. Therefore, dedicated electromagnetic transmitters or access points need to be installed in the indoor environment in order to setup the ILS. At the same time, indoor environments are complex and the line-of-sight (LoS) between the receiver and the access points is often blocked. This means that triangulation-based algorithms used by GPS chips are not directly applicable in indoor scenarios and new algorithms need to be developed for indoor localization.

Although many indoor localization systems have been proposed in publications or even made available on the market, most do not completely fulfill the desired requirements of an ILS. Many industrial applications such as robot navigation would require a localization accuracy in the centimeter range. A number of approaches have demonstrated cm level indoor localization accuracy in academic publications. Multiple commercial ILS on the market promise a 10-30cm accuracy. However, cm-accuracy systems in publications commonly use fingerprinting and are often tested over a small area in ideal conditions. Extending these algorithms to a full public building may require a system calibration time of several days or more. This will make such systems extremely expensive. While fingerprinting systems can function well under non-line-of-sight (NLoS) conditions, their accuracy will significantly decrease in real environments, which contain moving people and where furniture is rearranged and moved from time to time. On the other hand, commercial ILS generally use ranging and triangulation. Such systems can tolerate some amount of movement in the environment but require a LoS between the base stations/access points (AP) and the user. In order to ensure constant LoS conditions, a very large number APs is necessary, significantly increasing the system costs. Therefore, the goal of this thesis is to develop a high-accuracy RF-based indoor localization system that requires a relatively small number of base stations/APs and is able to maintain its performance in a NLoS environment with moving objects.

Several scarcely addressed aspects of indoor localization are also considered in this thesis. The user of an ILS commonly receives his or her location from the ILS server. This means that the ILS server is able to track the user's movements. If the ILS is provided by a third party, the data calculated by the ILS server could be used to analyze the user's behavior and to obtain sensitive and private information about the user's health or protected industrial data. For example, a third-party ILS installed in a hospital could track the offices visited by its users and make assumptions about the users' health. The movement of robots and machines in a factory can give an insight into proprietary technologies. Therefore, a privacy protection scheme needs to be integrated into ILSs.

Algorithm complexity and run-time is the second ILS aspect discussed in this thesis. Many state-of-the-art high accuracy ILSs, including the one proposed in this thesis, use a database to which the user's measurements are compared to estimate his or her location. The larger or denser the database, the better the location estimation of the ILS can be. However, in conventional ILSs the database lookup is slow as the query needs to be compared to every single database entry. This would make the practical implementation of such an ILS challenging in a building with a very large area, such as a shopping center or factory. The fingerprinting database in this case would be extremely large. Therefore, an efficient data-base lookup scheme is needed.

At the same time, in order to construct the necessary calibration database, or fingerprint map, multiple measurements need to be collected. The measurement data and the coordinates at which this data was collected are added to the database. As mentioned above high-accuracy localization systems generally require measurements to be collected at a very dense set of locations. This is very impractical if the ILS needs to be set up in a large building such as a shopping mall or factory. The time necessary for calibrating and setting up the ILS can be significantly reduced if some of the database entries can be interpolated from measurements collected at the surrounding locations. Such an interpolation scheme also needs to be developed.

Additionally, blueprints of buildings are often not available. Time consuming measurements are required to reconstruct them. Since a calibration database is collected by many ILSs by default, it would be much more convenient if an algorithm was developed to automatically reconstruct a map of the indoor environment from that database.

In summary, an ideal indoor localization system (ILS) should be able to localize moving and stationary receivers, maintain a high localization accuracy in a NLoS and dynamic environment, provide privacy protection for the users location, allow a fast data-base look up for real-time localization, be extendable to large buildings, allow database interpolation for faster system calibration and reconstruct the indoor geometry from calibration data. The author of this thesis is unaware of an existing ILS that fulfills all of the requirements listed above. The aim of this thesis is to develop such an ILS.

1.2 Main Contributions

The following contributions are made in this thesis:

- The development of a low-cost fingerprinting-based indoor localization scheme that is able to localize a static or moving receiver device and is robust to the presence of moving objects and changes in the environment. As a fingerprinting-based scheme, the proposed approach does not require a LoS between the user and the base-stations and thus requires less infrastructure than the high accuracy systems currently on the market. At the same time, the use of multipath delay fingerprints in this thesis allows the proposed ILS to function in the presence of moving objects, in contrast to existing state-of-the-art fingerprinting algorithms. Both these facts reduce the systems installation and maintenance cost and increase its robustness.

- The development of a privacy protection scheme that allows the user to obtain his or her location from the ILS server, without the ILS server being able to track the user. The proposed privacy protection scheme is made possible by the multipath-based data format used for localization.

- The development of a database structure that enables a more efficient query lookup. The database is designed for the multipath based fingerprint structure and multipath component analysis algorithm proposed in this thesis. As a trade-off exists between the algorithm's run-time and accuracy, the optimized data based lookup scheme allows the user to set the localization accuracy according to the available computational power. The optimized algorithm also produces and updates a preliminary location estimate during run-time.

- A scheme for interpolating the ILS database is proposed. The novel interpolation/ extrapolation scheme relies on the concept of virtual transmitters (VTs). Given a set of reference fingerprints, composed of multipath propagation delays, the algorithm identifies and groups together the the multipath propagation delays across different fingerprints which correspond to propagation paths reflected from the same surface. The multipath components which were grouped together are then used to reconstruct th VT positions with trilateration and calculate the interpolated/extrapolated fingerprints. Interpolating and extrapolating some of the database entries increases the database size and the ILS's accuracy without increasing the calibration time.

- A scheme for reconstructing the indoor geometry from the ILS calibration database is proposed. The algorithm is designed for the multipath fingerprint structure and is based on VT reconstruction.

- Measurement and simulation-based evaluation of the proposed approaches is conducted in this thesis. The proof of concept testing is performed with a prototype using the Decawave board [7] and the ultra-wide-band (UWB) communication standard. A localization error of 6cm is obtained for a slow moving user and around 30 cm for a fast moving user with an average distance of respectively 3cm and 10cm between the ground-truth user location and the closest location at which an entry in the database was measured.

There are two main differences between the localization approach proposed in this thesis and the cm-level accuracy state-of-the-art fingerprinting approaches. Calibration signal measurements are collected and stored in a database in both cases. The user's measurements are later compared against that database for localization. However, before being stored in the database, the signal measured by the receiver device needs to be converted to a certain representation. A set of properties needs to be extracted from the signal or a transform needs to be applied to it. Such characterizations include the signal received power, the channel impulse response (CIR), the channel frequency response (CFR) and the channel state information (CSI) estimated from the received signals. The approach proposed in this theses uses the multipath delay values for localization. This is done by only a small subset of state-of-the art approaches. The second difference between the state-of-the-art and the proposed approach is in the way in which the user's measurement is compared to the database entries. In the state-of-the-art approaches the received signal information is processed and used as one entity. The approach in this thesis identifies parts and the segments of the signal information which correspond to the objects moving in the environment. Only the multipath components that correspond to the background indoor geometry are used for localization.

The focus of this thesis is the development of a one-sensor localization scheme. Therefore, the proposed scheme does not rely on additional sensors like an inertial measurement unit (IMU) and on motion filters such as the Kalman filter. The developed scheme needs to be able to localize a static receiver, such as a chip mounted on equipment in a warehouse. Additional sensors and algorithms can be used afterwards to augment the base localization scheme.

1.3 Thesis Organization

This thesis is structured as follows. Chapter 1 motivates the need for a new robust indoor localization system and lists the requirements which the ILS proposed in this thesis aims to fulfill. Chapter 2 summarizes the existing indoor localization algorithms and hardware implementations which have been proposed in publications or are commercially available. Chapter 3 describes in detail the multipath-based localization system proposed in this thesis. The simulation environment and practical implementation of the proposed system are also described in this chapter together with the performance evaluation of the core algorithm. Chapter 4 describes the novel user privacy protection scheme. Chapter 5 presents the proposed database structure and fast fingerprint lookup algorithm. Chapter 6 presents the proposed virtual transmitter (VT) reconstruction, fingerprint inter- and extrapolation and environment reconstruction algorithms. Eventually, Chapter 7 concludes this thesis and indicates possible directions for future research.

Parts of the work presented in this thesis have been published in [2], [1], [3]–[5]. The digital object identifiers (DOIs) of these publications are listed in the bibliography.

Chapter 2

Overview of Indoor Localization Techniques

This chapter gives an overview of existing indoor localization schemes and systems. Indoor localization is required when a smart system containing movable devices is installed indoors. Robots need to know their position to navigate through and find objects in a factory, residential and commercial spaces. Large warehouses require a method for localizing resources and equipment. Objects need to be tracked to construct a thorough electronic inventory. Smart phone-based navigation apps guiding people though large public buildings and indoor spaces, such as train stations, also need to know the user's positions.

Outdoors global navigation satellite system (GNSS) signals can be used to calculate the location of a electromagnetic receiver with a precision of several meters. Unfortunately, as GNSS signals are attenuated by walls and ceilings, the global positioning system (GPS) and other state-of-the-art GNSS outdoor localization techniques cannot always produce a reliable and accurate position estimation indoors.

In this thesis only indoor localization systems that rely on the reception and transmission of electromagnetic signals are discussed. The focus is placed on radio-frequency (RF) localization techniques due to the wide availability and fast development of RF technology and hardware. Additionally, most large buildings already have a communication network of transmitters and routers installed. The communication signals can be directly used for localization. Alternatively, additional transceivers or boards can be integrated into the existing communication routers. In both cases, pre-installed infrastructure is reused and the installation and maintenance costs of the ILS are kept low.

This chapter is organized as follows. The requirements of an ILS and the challenges that arise during ILS development are listed in Section 2.1. Electromagnetic signal properties that are measured or calculated for indoor localization are summarized in Section 2.3. Localization approaches based on ranging and triangulation and multilateration are discussed in Section 2.4. Fingerprinting-based indoor localization approaches are summarized in Section 2.5. Existing communication standards and hardware used for indoor localization are discussed in Section 2.7. A number of commercially available ILSs are listed in Section 2.9. Due to the large number of publications in indoor localization, most publications in the above mentioned tables were selected at random. However, a number of publications were also included due to their novelty, distinctive methodology or superior results and performance. This was done in order for the tables to include both frequently used and state-of-the art localization approaches.

2.1 System Model

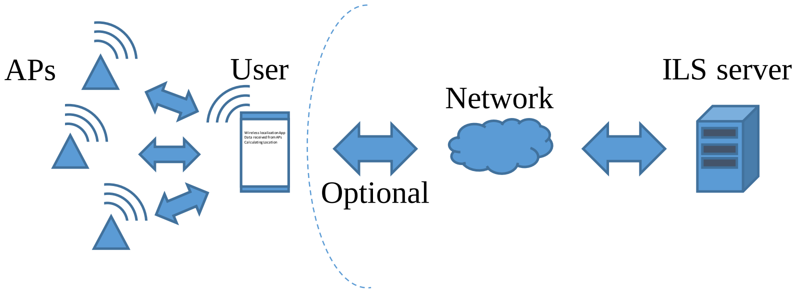


Figure 2.1: System overview for RF-based indoor localization. A receiver communicates with the transmitters to measure electromagnetic signal properties. The measured signal properties are then either used to calculate the receiver's position locally at the receiver, or are sent to an ILS server. In the second case the receiver then obtains its position (or the information from which it can determine its position) from the ILS server.

This thesis focuses on indoor localization systems (ILSs), which are based on the transmission, reception and processing of RF electromagnetic signals. The system model for such a localization system is shown in Fig. 2.1. An electromagnetic receiver, also referred to as a tag, processes signals from several transmitters or access points (APs) installed in the indoor environment. The receiver may calculate its location locally or send the received data to an ILS server. The ILS server calculates the receiver's location and communicates it to the user.

There are several differences between a localization system that is installed indoors and GNSS-based systems used outdoors.

- Outdoors, there is a high likelihood that there is a direct line-of-sight (LoS) between the user and the satellites. Indoors, the user may be located in different room than an access point or behind a corner. An object may also be blocking the LoS between the user and an AP.
- Additional transmitters can easily be installed in an ILS.
- Since the distances between transmitters and receivers is significantly smaller for ILS than for GPS, an ILS may rely on a wider range of frequencies.
- An ILS often covers a much smaller area, therefore, it is possible to spend some time calibrating the system before it is used.

Figure 2.2 shows the most common channel parameters [8] that the receiver extracts from the AP's signals. Those characteristics of the wireless channel between the receiver an transmitter are then used as an input to one of the localization algorithms described in Sections

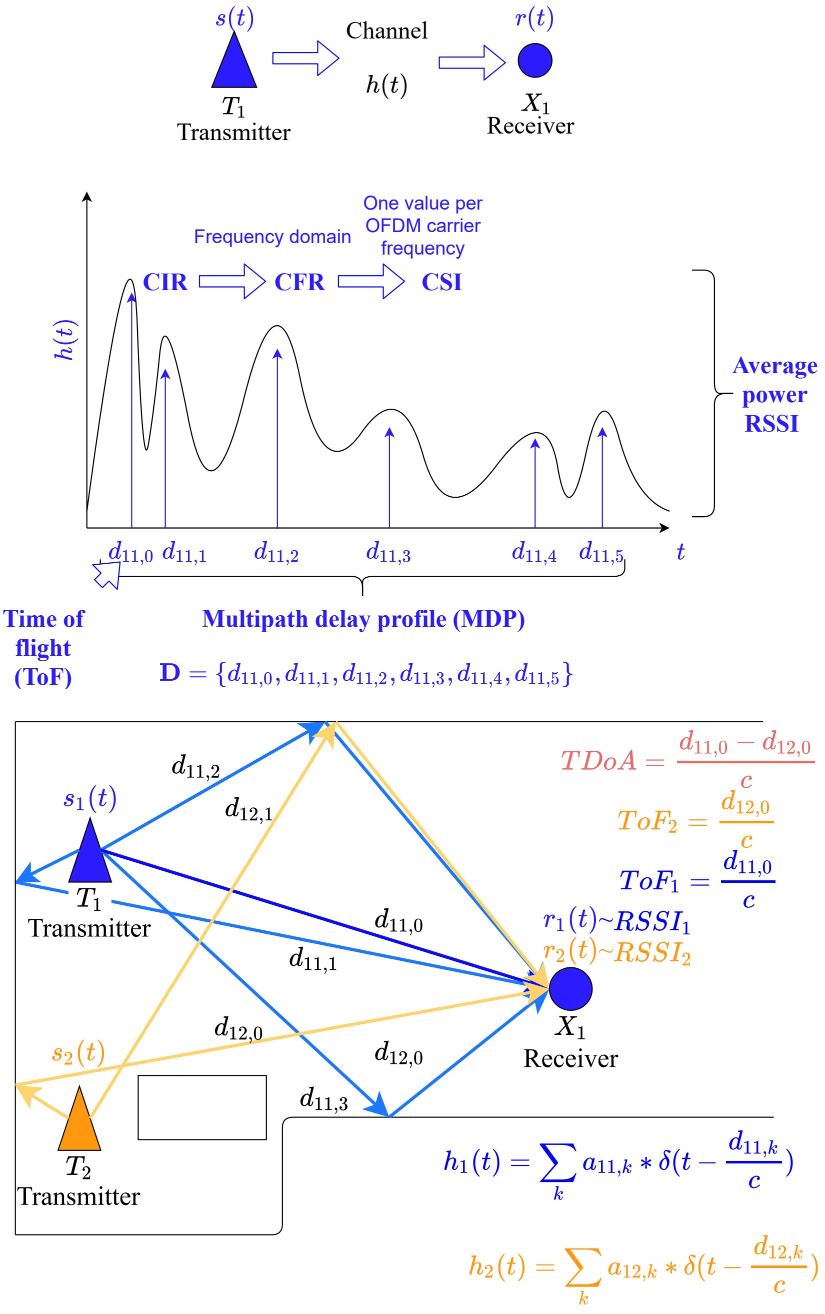

$s(t)$
Channel
$h(t)$
$r(t)$
T_1
Transmitter
X_1
Receiver
One value per OFDM carrier frequency
Frequency domain
CIR
CFR
CSI
$h(t)$
Average power RSSI
$d_{11,0}$ $d_{11,1}$ $d_{11,2}$ $d_{11,3}$ $d_{11,4}$ $d_{11,5}$
t
Time of flight (ToF)
Multipath delay profile (MDP)
$\mathbf{D} = \{d_{11,0}, d_{11,1}, d_{11,2}, d_{11,3}, d_{11,4}, d_{11,5}\}$
$TDoA = \frac{d_{11,0} - d_{12,0}}{c}$
$ToF_2 = \frac{d_{12,0}}{c}$
$ToF_1 = \frac{d_{11,0}}{c}$
$r_1(t)$~$RSSI_1$
$r_2(t)$~$RSSI_2$
$d_{11,2}$
$s_1(t)$
$d_{12,1}$
T_1
Transmitter
$d_{11,0}$
$d_{11,1}$
$d_{12,0}$
$s_2(t)$
$d_{12,0}$
X_1
Receiver
$d_{11,3}$
T_2
Transmitter
$h_1(t) = \sum_k a_{11,k} * \delta(t - \frac{d_{11,k}}{c})$
$h_2(t) = \sum_k a_{12,k} * \delta(t - \frac{d_{12,k}}{c})$

Figure 2.2: Received signal characteristics that are used to localize a device. $h(t)$ denotes the channel impulse response. It should be noted that in the bottom figure there is no LoS between X_2 with T_1, therefore using the ToF_2 and the $TDoA$ for multilateration will produce the incorrect result.

2.4 - 2.6.2. The basics of indoor localization techniques described in this section are also summarized in [9]–[11].

2.1.1 Collaborative ILS

It is possible for multiple ILS to collaborate and share data as discussed in [12]. However, the more accurate the individual localization systems are, the better the results of the collaboration will be. Such an ILS collaboration requires higher level design features and protocols which are not considered in this thesis. Therefore, the system model assumed in this thesis contains one ILSs.

2.1.2 Note on Passive Localization Systems

A subset of localization systems use electromagnetic signals to localize a user, however do not require a user to carry a receiver or transmitter. These systems analyze the signals and electromagnetic channels between the APs and find perturbations in these channels to track the user [13]–[15]. One simple approach is to track whether the user blocks the line-of-sight (LoS) the APs. The work in [15] analyzes the channel impulse response (CIR) between ultra-wideband transmitters and receivers. The approach in [13] tracks if any of the multipaths between the APs are blocked. Passive localization systems are outside the scope of this thesis. Therefore, the user is assumed to carry an electromagnetic receiver and/or transmitter.

2.2 Simultaneous Localization and Mapping (SLAM) and Static Localization

There are two types of localization algorithms. Some localization systems, including GPS, use a snapshot of the signal data obtained by the electromagnetic receiver to calculate its position. Such schemes are as a rule able to localize both a stationary object, such as a box in a warehouse to which a tag has been attached, and a moving user. When a user is moving too fast, the localization accuracy of the systems may decrease due to Doppler effects or insufficient signal transmission and reception rate.

Alternatively, pedestrian dead reckoning (PDR) and simultaneous localization and mapping (SLAM) algorithms use motion data for localization. A PDR algorithm calculates the user's location from a previously known location and the motion information, for example by integrating twice the acceleration data measured by an accelerator. In a SLAM algorithm, the user measures some signal or environment characteristics. As he or she moves, the SLAM algorithm also acquires the motion information. The algorithm estimates by how much and in which direction the user moved. It then uses this information in combination with the signal measurements to calculate the user's location relative to an initial starting position. A SLAM-based algorithm also constructs a map of the indoor environment as the user moves. SLAM algorithms commonly use the following data:

- Data from an inertial measurement unit (IMU) or gyroscope, that shows in which direction and how much the user has moved.
- New data about the environment. This can include radar measurements, signal time-of-arrival or time-of-flight and the signal strength. Motion information can be inferred from this data as well when it is analyzed over time.

While an IMU is frequently used for localization and is commonly built into smartphones, there are two main challenges in IMU-based SLAM.

- The measurements of the IMU are imprecise.
- The measurement error accumulates over time. If the location of the moving user is calculated by adding up the IMU-based estimates of incremental distance and direction, by which the user moved relative to a previous estimate, the error of the user's location will increase after every measurement. This is referred to as *drift*.

Very often the location estimates obtained from electromagnetic signals, a SLAM algorithm, and, potentially, additional sensors are fused together to obtain a more accurate location estimate. Initial SLAM approaches were designed for data from a laser scanner, however they are not discussed in this thesis. Sections 2.6.2 and 2.6 describe a number of systems that perform SLAM-based localization with the data obtained by an electromagnetic receiver. The following mathematical techniques are commonly used for SLAM and sensor fusion.

- A *Kalman filter* is a recursive filter that
  - Provides the optimal state estimate of the user.
  - Is composed of the location estimate and Kalman gain matrix.
  - In the prediction step an a priory estimate of the user's location is computed from a previous location estimate and the current control input.
  - In the update step the a priory location estimate is used to predict the sensor measurements. The difference between predicted and measured data is weighted by the Kalman gain matrix and added to the a priory estimate to compute the posteriory estimate of the user's location. The Kalman gain matrix is also recomputed in this step.
  - The Kalman gain is computed from the uncertainty of the prediction and of the measured data.
- An *Extended Kalman filter (EKF)*
  - A variation of the Kalman filter designed for non-linear systems.
  - Linearizes the system locally around the users state estimate by using Jacobian matrices in the prediction and update states of the Kalman filter.
- A *particle filter*
  - Uses a set of particles that represent the likely positions of the user.
  - As the user moves, the positions of the particles are updated with the IMU data. The particle distribution is re-sampled. Particles that move into regions where the user cannot be or which don't match sensor measurements are deleted. New particles are created at locations that the user is likely to be at.

This thesis is focused on developing a localization algorithm that is able to localize a static receiver. This is done for two reasons. The localization of resources and equipment in factories and warehouses is an important ILS use case. Additionally, an algorithm that can localize a static object can always be augmented with the motion data from an IMU and made more precise. The better the base localization algorithm is, the more accurate the result of the data fusion will be.

2.3 Signal Properties used for Localization

Once a user receives a signal from one of the APs, or vice versa, that signal needs to be processed before it can be used for localization. The characteristics of the received signal or the electromagnetic channel between the user and APs illustrated in Fig. 2.2 are commonly used for localization.

2.3.1 ToA,TDoA, AoA

The Time-of-Arrival (ToA) [9]–[11] is the time at which a signal reaches the receiver. This time is measured according to the receiver's internal clock. If the clocks of the receiver and the transmitter are synchronized, the time stamp of the received signal is used to calculate the Time-of-Flight (ToF) of the signal. The ToF is the time it takes the signal to travel from the transmitter to the receiver. The ToF is also the first peak of the channel impulse response between the transmitter and receiver shown in Fig. 2.2. If there is no obstacle blocking the LoS, the distance between the receiver and transmitter can be calculated as

$$d_0 = \frac{ToF}{c}, \tag{2.1}$$

where c is the speed of light. The Time-Difference-of-Arrival (TDoA) is the difference between the times at which the signals from different transmitters reach the receiver. An advantage of TDoA measurements is that the receiver and transmitters do not need to be synchronized. The Angle-of-Arrival (AoA) is the angle at which the signal arrives at the receiver and is easily obtained if the receiver contains an antenna array [16], [17].

ToA, TDoA, AoA can be used with any localization algorithm described in Sections 2.4 - 2.5.3.3, but are more commonly used in ranging localization approaches in Section 2.4.

2.3.2 RSSI

A very popular signal parameter used for localization is the *received signal strength indicator (RSSI)*. The RSSI represents average power in dB of a signal received by the user [18]–[23]. A commonly used model for the RSSI is [24]:

$$RSSI(d_0) = P_T - P_{d_*} - 10 \cdot \eta \cdot log_{10}\frac{d_0}{d_*} + X_\sigma, \tag{2.2}$$

where the RSSI is in dB, d_0 is the distance the signal travels from the transmitter to the receiver, P_T is the transmit power, P_{d_*} is the signal power measured at a reference distance d_* from the transmitter, η is the pass loss exponent and X_σ represents additive white Gaussian noise (AWGN) with variance σ^2.

The RSSI is most commonly used by fingerprinting-based algorithms detailed in Sections 2.5-2.5.3.3. The RSSI can also be used for ranging and distance estimation, according to Eq. 2.2, however the results generally lack in accuracy.

The RSSI is measured by default by most mobile devices, such as smartphones, which has made the RSSI popular for ILS algorithms. It can also be calculated from the channel impulse response, channel frequency response and channel state information [18].

2.3.3 CSI, CFR, CIR and Multipath Information

As a signal propagates through an environment, it is reflected from walls and objects as shown in Fig. 2.2. Because of this, the signal at the receiver is attenuated, delayed and shifted in phase. The signal from one AP generally travels though several paths to reach the receiver. All of these effects are included in the channel impulse response (CIR) [18]. The received signal is equal to the convolution of the original signal with the CIR. If the transmitted signal is in the form of a Dirac delta impulse, then the received signal is equal to the CIR. A common wide-band model for the CIR $h(t)$ shown in Fig. 2.2 is:

$$x(t) = h(t) * s(t) = \sum_k a_k s(t - \tau_k) + n(t) = \sum_k a_k s(t - d_k/c) + n(t), \tag{2.3}$$

where $x(t)$ is the received signal, $s(t)$ is the original signal, a_k is the attenuation undergone by the signal along the k-th propagation path from the transmitter to the receiver and τ_k is the delay and d_k is the length of that path. c is the speed of light. The term $n(t)$ represents random noise added to the signal at the receiver [8], [18]. The channel frequency response (CFR) is the frequency domain representation of the CIR.

The CIR is directly used for localization by some publications. Alternatively, several parameters or features can be derived from the CIR for indoor localization algorithms. The *channel state information (CSI)* most commonly describes the received power and, optionally phase, at each orthogonal frequency-division multiplexing (OFDM) carrier [25]–[27]. In that case, the CSI is a downsampled version of the CFR [28], [29]. The *multipath components* are the delays with which the reflected copies of the sounding signal reached the receiver equal to the locations of the peaks in the channel impulse response [3], [30]–[32]. The multipath components can also be expressed as the distances traveled along the individual propagation paths shown in Fig. 2.2. In that case, the delays are multiplied by the speed of light.

The CIR and CFR can be measured by a channel sounder. CSI data is provided by certain WiFi network interface cards (NICs), such as the Intel IWL 5300, which has become a common platform for WiFi CSI-based localization [26], [29]. There exist multiple methods for estimating multipath information from a measured signal as shown in Table 2.1.The MUSIC algorithm was proposed for multipath azimuth estimation in [16], making it possible to obtain the angles of arrival of the individual multipath components can be obtained. The principle component analysis has been used for delay estimation in [33]. The ESPRIT [17] and Unitary ESPRIT [34] algorithm, while also designed for azimuth estimation, have later been correspondingly extended to jointly estimate the delays and the azimuth [35], [36] and elevation angles [37] of a received multipath signal. The independent component analysis is used in [38] to estimate the multipath delays and azimuth angles of the global navigation satellite system (GNSS) signals. The space-alternating generalized expectation-maximization (SAGE) algorithm is proposed in [39]. Alternatively [40] uses the volume cross correlation for estimating multipath TDoAs.

The CSI, CIR, CFR and multipath information are predominantly used for fingerprinting-based localization schemes detailed in Section 2.5.

Existing multipath estimation methods				
Authors	Year	Estimates	Algorithm	Platform, communication standard
Schmidt at al. [16]	1986	azimuth	MUSIC	-
Roy at. al. [17]	1989	azimuth	ESPRIT	**simulation**
Haardth at al. [34]	1995	azimuth	Unitary ESPRIT	**simulation**
Zoltowski at al. [37]	1996	azimuth, elevation angles	ESPRIT	**simulation**
Van Der Veen at al. [35]	1997	delays and the azimuth	ESPRIT	**simulation**
Lo at al. [33]	1994	multipath delays	principle component analysis (PCA)	**simulation**
Fleury at al. [39]	1999	relative delay, incidence azimuth, Doppler frequency, and complex amplitude	space-alternating generalized expectation-maximization (SAGE), derived from the maximum likelihood (ML) principle	**simulation**, **wide-band channel sounder** with a carrier at the frequency of 1.98 GHz modulated with a PN sequence of period of K = 255

Luo et al. [41]	2010	multipath ToAs, TDoAs	modified complex to real amplitude least-squares algorithm, aimed at better resolution between close MPCs	**simulation**, measurements **UWB-IR system**, active tag and receiver
Gifford at. al. [42]	2011	delays and gains of multipath components	Search-Subtract-Readjust Algorithm based on ML estimation	**Agilent E8362B vector network analyzer**, center frequency 4-16GHz BW 1-16GHz
Gentner at al. [43]	2016	angles of arrival, multipath delays	Kalman enhanced super resolution tracking (KEST)	**broadband channel sounder**, a 1 mW multitone signal, with N = 1281 subcarriers with equal gains at a center frequency of 1.51 GHz with a bandwidth of B = 100 MHz
Zhao at al. [38]	2017	delays, steering vector	independent component analysis	**GNSS receiver**, **GNSS**
Wielandt at al. [44]	2018	multipath AoAs	steering the beam electronically over all directions and measuring the output power of the beam former, MVDR algorithm	**horizontally omnidirectional mobile transmitter and a uniform linear antenna array**, **vector network analyzer**
Sousa at al. [40]	2019	multipath TDoAs	volume cross correlation	CIR extracted from **ray-tracing simulation**

Jaffe at al. [45] Chen at. al. [32]	2014 2019	multipath AoAs and differential delays	Singular value decomposition (SVD) of the signal covariance matrix	**a circular antenna array of a WLAN base station**
Ayvasik at al. [46]	2019	multipath delays	pilot aided channel estimation (known pilot sequence), LS estimate	**USRP**
Cimins at al. [15]	2020	multipath delays	as peaks of CIR	**UWB, Decawave DW1000 chip**
Gentner at al. [47]	2021	multipath delays	Channel SLAM	**UWB, Decawave DW1000 chip**

Table 2.1: Existing multipath estimation methods. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

2.4 Ranging and Triangulation

Ranging-based algorithms measure the distances and angles between the user and 3 or more APs. If there is no obstacle blocking the LoS (line-of-sight) to the AP, the received signal is used to estimate the incidence angle and distance between the user the AP. The incidence angle is the angle at which the signal arrives at the receiver from an AP. Distances and/or angles to at least 4 APs are required to localize the receiver in 3D . Trilateration/multilateration is used to calculate the receiver's location from ToF values and triangulation from AoA measurements [9], [48], [49]. A 2D version of multilateration is illustrated in Fig. 2.3. Multilateration finds the location of the receiver as the intersection of spheres centered at the APs, the radii of which equal to the speed of light multiplied by the ToF values. The Least-Squares (LS) algorithm is generally used. Some algorithms, such as the one proposed in [48] combine AoA and ToF measurements. A good overview of basic ToA and AoA localization techniques is provided in [48]. Table 2.2 lists several publications on ranging-based localization.

While ranging algorithms are most commonly applied to ToF, TDoA and AoA data, the distances from the receivers to the APs can also be extracted from RSSI data using Eq. 2.2 [50]. The accuracy of such measurements is, however, generally low.

Example Multilateration and Triangulation Ranging based ILSs.						
Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Qiao at al. [51]	2014	expectation maximization (EM) method for TDoA	smaller computational complexity	**simulation**	-	$\simeq$ non linear LS
Xie at al. [52]	2017	TDoA based on RSS-assisted Cross-correlation method	RSS used to mitigate the effect of multipath interference	**Software defined radio**, **WiFi**	CDF, Precision	0.3m
Djaja-Josko at al. [53]	2018	TDoA	method for simplifying synchronization	**UWB**, **DW1000chip**	CDF, max lower 90% of errors	1ns
Bonnin-Pascual at al. [54]	2019	UWB ToA, filtering of range data,localization with a modified Iterative Closest Point (ICP) algorithm	designed for environments with metalic objects and surfaces close to the antennas	**Pozyx UWB kit**	trajectory plots	error < 0.5m
Zhang at al. [55]	2020	TDoA with Chan algorithm	a non-recursive algorithm with analytic expression solution	**simulation**	CDF, RMSE	

Table 2.2: Example approaches that rely on ranging, triangulation and multilateration. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

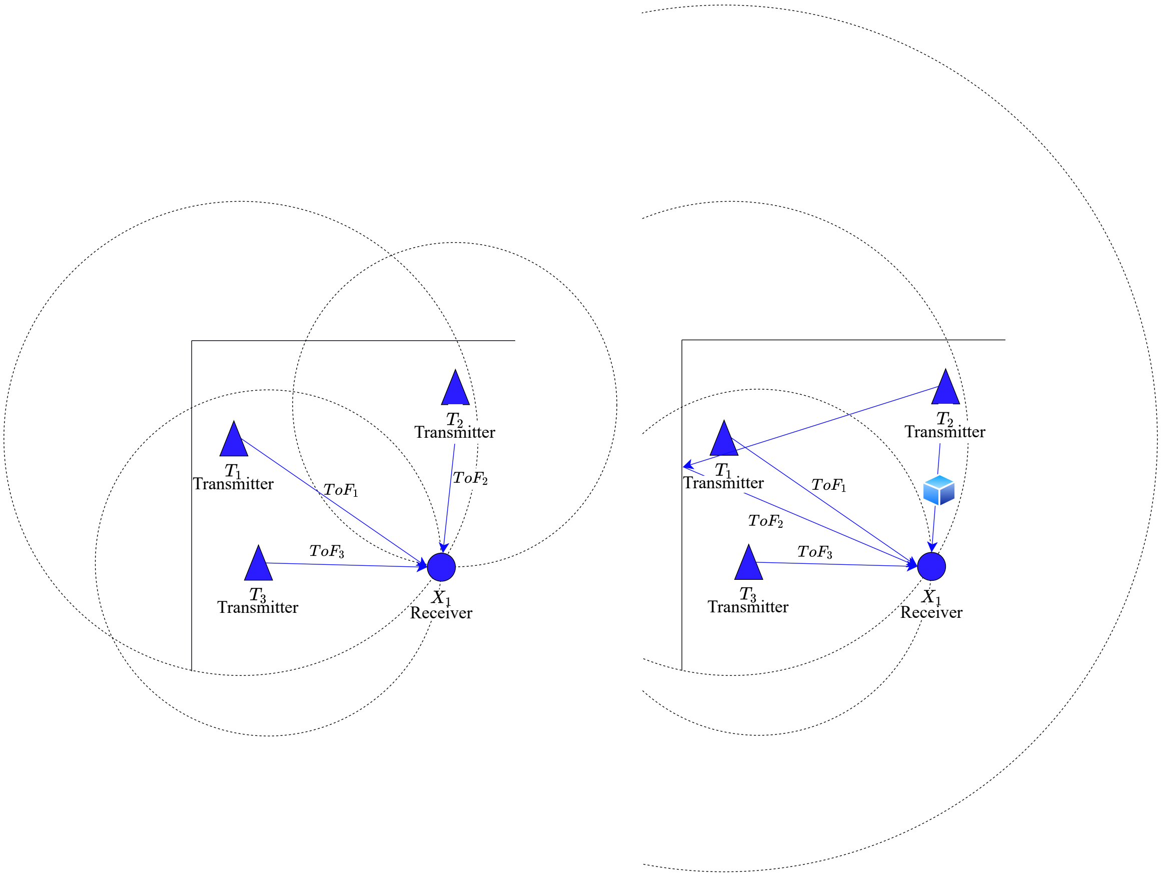


Figure 2.3: Trilateration. Left: it is possible to calculate the location of the receiver for line-of-sight (LoS) conditions; Right: due to an obstacle blocking the line-of-sight with transmitter T_2 (non-line-of-sight (NLoS) conditions) it is not possible to calculate the location of the receiver.

In order to obtain ToF values, the receiver and the APs need to be synchronized. Since electromagnetic waves propagate with the speed of light, a measurement error of 1 μs alters the ToF values by 300 m. [48] proposes a method that can increase the accuracy of ToF measurements through sending a predefined message multiple times. The authors of [49] apply the time-reversal technique to more accurately calculate the distance between the transmitter and the receiver. With a slight modification of the LS algorithm, the location of the receiver can be calculated from TDoA values. In that case synchronization is not required between the receiver and APs.

Unlike the fingerprinting algorithms detailed in Section 2.5, ranging and AoA-based algorithms do not require time consuming calibration measurements. However, the main challenge of triangulation and multilateration localization algorithms is the requirement of line-of-sight (LoS) conditions. Figure 2.3 illustrates that an incorrect location is calculated when an obstacle blocks the LoS path from the receiver to one of the APs. Since there is no LoS path, the algorithm assumes that the length of the shortest reflected path or multipath component is equal to the distance between the receiver and the AP. If that value is used by the LS algorithm, the localization error increases significantly. In the example in Fig. 2.3 the circles do not intersect and the calculated localization error is very high. Some ILS attempt to reduce the localization errors by detecting NLoS measurements and excluding them from the trilateration, i.e. distance, and triangulation, i.e. angle, measurements. At the same time, ranging measurements with accuracies down to $10cm$ can be obtained from ultra-wide band (UWB) devices such as the Decawave board [7]. Because of this, as shown in Section 2.9, most high precision commercial ILS use ToF/TDoA/AoA estimation and multilateration. The LoS requirement means that in order for those ILS to reliably localize a user in a complex environment a large number of APs needs to be installed. Therefore, existing commercial ILS may be low cost in an open environment but become very expensive if the environment is complex and contains moving objects.

Additional ranging algorithms can be found in fusion systems in Table 2.12, commercial systems in Table 2.18 and machine learning and deep learning approaches in Tables 2.8-2.9.

2.5 Fingerprinting-based Localization Approaches

Fingerprinting algorithms localize the user by comparing his or her measurements to a prerecorded map. The main advantage of fingerprinting-based indoor localization algorithms is that their performance is not affected by non-line-of-sight (NLoS) conditions [18]. As is illustrated in Fig. 2.4, the algorithms consist of two main steps. First, during the *off-line*, *calibration* or *training phase*, certain measurements, or *reference fingerprints*, are collected at a set of *reference points* in the indoor environment. The reference fingerprints and the corresponding reference points are stored in a map called a *fingerprint map*, *radio map* or *calibration database*. The work in [9], [56], [57] provides a detailed overview of fingerprinting-based localization.

In the *on-line* phase, the user communicates with the APs. He or she extracts the same type of parameters or features from the received signals, as were calculated during the calibration phase. The ILS server compares the resulting *query* fingerprint to the map. A scalar similarity value is computed for each reference fingerprint stored in the map. The similarity metric represents how similar a reference fingerprint is to the query fingerprint. Example similarity metrics include [56]:

- The Euclidean distance between two fingerprint vectors [10],
- Correlation of two fingerprint vectors,
- Time-reversal correlation of the fingerprint vectors [28].

It should be noted that localization systems such as [32], [58], require the communication protocols between user and the APs to be reversed. Instead of the user receiving a signal from the APs, the APs receive a signal from the user, compare it to a fingerprint database

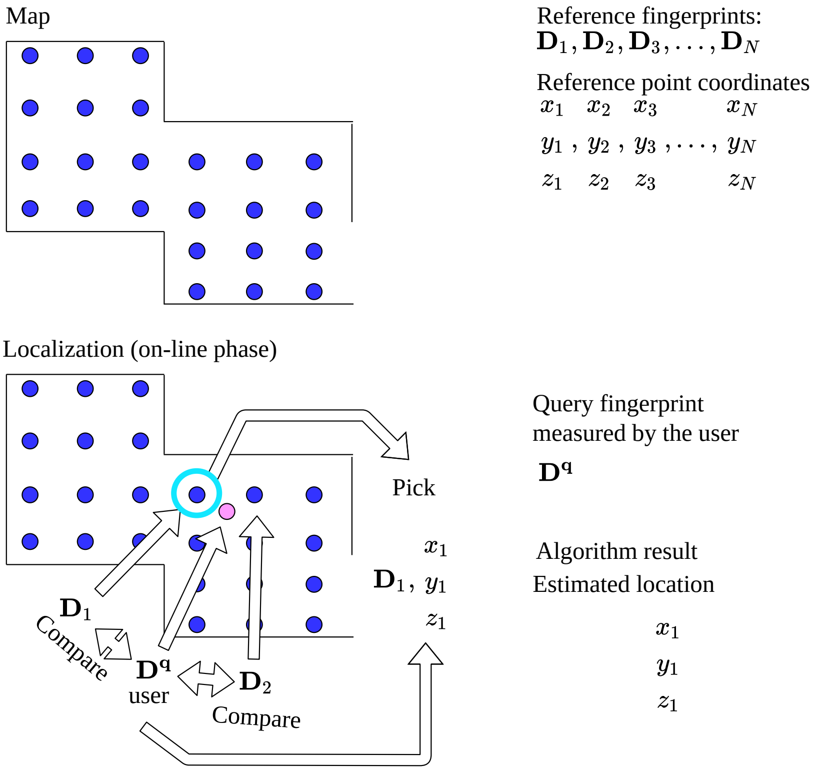

Map
Reference fingerprints:
$\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \dots, \mathbf{D}_N$
Reference point coordinates
$x_1 \; x_2 \; x_3 \; x_N$
$y_1, y_2, y_3, \dots, y_N$
$z_1 \; z_2 \; z_3 \; z_N$
Localization (on-line phase)
Query fingerprint
measured by the user
$\mathbf{D}^{\mathbf{q}}$
Pick
x_1
$\mathbf{D}_1, y_1$
z_1
Algorithm result
Estimated location
x_1
y_1
z_1
$\mathbf{D}_1$
Compare
$\mathbf{D}^{\mathbf{q}}$
user
$\mathbf{D}_2$
Compare

Figure 2.4: Fingerprinting-based localization.

and localize the user. The same algorithms apply. An advantage of such hardware is that a simpler off-the shelf device can be carried by the user, while more complicated antenna arrays can be installed in the APs to analyze the user's data. This is done in [32], [58]. However, it is challenging for a such a system to simultaneously track a large number of users. Specialized communication protocols are needed to handle the situation when several users contact the APs simultaneously.

The five signal or channel characteristics listed below are most often used as fingerprints:

- Received Signal Strength Indicator (RSSI) (see Sec. 2.3.2),

Channel impulse response (CIR) (see Sec. 2.3.3),

Channel frequency response (CFR) (see Sec. 2.3.3),

Channel State Information (CSI). The most common definition of a CSI fingerprint is the received power at each of the OFDM carriers (see Sec. 2.3.3),

Multipath components (see Sec. 2.3.3).

TOF and AoA values can be used for fingerprinting but if they are available they are more commonly used for multilateration and triangulation.

Fingerprint	Parameter or feature vector created from the electromagnetic signal received by the user from the APs that is used for localization.
APs	Fixed electromagnetic transmitters or transceivers that the user communicates with to calculate his or her location. Also referred to as transmitters.
ILS server	Core of the system that stores the fingerprint map and that the user communicates with to obtain his or her location
Receiver	Device carried by the user, term used interchangeably with user
Fingerprint map	Data collected during system calibration and possibly updated while the system is on-line. The fingerprint map contains the fingerprints measured at reference locations.
Query fingerprint	Fingerprint measured by the user to estimate his or her location.
Reference fingerprints	Fingerprints stored in the map.
Reference locations	Coordinates at which the reference fingerprints were measured
MDP	Multipath delay profile, all of the multipath components extracted from the signals the user received from the APs at a given location.
Multipath component	The length of one propagation path that the signal travels from an AP to the receiver.
Similarity metric	A scalar value calculated between the query and a reference fingerprint that indicates how similar the query is to the reference.

Table 2.3: Terms and definitions related to fingerprinting in this thesis.

2.5.1 Fingerprint Types

2.5.1.1 RSSI Fingerprinting

The received signal strength indicator (RSSI) was one of the first signal properties to be used as a fingerprint. The RSSI is automatically measured by many hand held devices, such as smartphones, however, it fluctuates significantly over time due to the fading of the channel and changes in the environment [59]. This happens because the RSSI aggregates the complete channel information into one scalar value. Therefore, the complete RSSI measured by the receiver fluctuates significantly if changes in the environment alter even one aspect of the electromagnetic channel between the receiver and the AP. This results in low accuracies of practical ILSs that rely on only RSSI fingerprinting. Therefore, most ILS that use RSSI fingerprinting fuse its result with the data from other localization techniques or sensors mounted on a mobile device, such as an IMU. These ILS are introduced in Section 2.2 and discussed further in Section 2.6.2. Table 2.4 lists a number of publications that present pure RSSI fingerprinting schemes.

Example RSSI Fingerprinting ILS.						
Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Bahl et al. [60] RADAR	2000	**WiFi** RSSI fingerprinting	Measured data and linear regression used to calculate signal propagation parameters. The propagation parameters are used to create the fingerprint map.	office area	resolution, 25th percentile	4.3m, 1.86m
Van Haute et al. [61]	2016	**WiFi** RSSI fingerprinting	Active hospital environment	error evaluation over a grid of known points	average error, room level accuracy, latency	1.21m, 96%, 5.43s
Shu at al. [62]	2016	WiFi fingerprinting using RSSI gradient + IMU + extended particle filter	A binary RSSI gradients calculated from the RSSI fingerprint map a gradient fingerprint database (Gmap) ia established. More robust against fluctuations, measured RSSI gradients compared to the map.	**smartphone**, walking traces	cdf, 80th percentile	3.6m-5.6m

Zhang at al. [50]	2017	**WiFi** RSSI path-loss-based fingerprint localization and dual-scanned fingerprint localization	RSSI and distances from RSSI path loss model used for fingerprinting	**simulation using the KIOS dataset**	mean error, CFD	3m
Järvinen et al. [63]	2019	Faster fingerprinting with quantized **WiFi** RSSI	Quantized RSSI is used, multiple similarity metrics compared	**smartphone**-based evaluation	mean error	5.68 m 8 bit - 9m 1 bit

Table 2.4: Examples of exclusive RSSI fingerprinting ILSs. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

2.5.1.2 CSI, CIR, CFR Fingerprinting

Due to the low accuracy of RSSI fingerprinting, the CSI, CIR and CFR, detailed in Section 2.3.3 are presented in many publications as a better alternative. The CSI, CIR and CFR contain more information about the electromagnetic channel and therefore, have been shown to produce up to a cm-level localization accuracy when used as fingerprints. As is explained in Section 2.3.3, CSI fingerprints can have several different formats.

- In [64], [65] a set of parameters calculated from the channel impulse response (CIR) are used as the fingerprint.
- The channel frequency response (CFR) is used as a fingerprint in [28], [29].
- A vector containing the received power and phase at each OFDM carrier for each AP is used in [27]. This is the most common definition of a CSI fingerprint. It can also be viewed as the downsampled CFR.
- The multipath components extracted from the CIR between the receiver and the AP are used as the fingerprint in [3], [30]–[32] and can also be viewed as CSI information. The multipath components are the length of the propagation paths of the signal, both direct and reflected. The multipath components divided by the speed of light are equal to the locations of the peaks of the CIR (as shown by Eq. 2.3). Multipath fingerprinting is discussed in the next section.

The similarity between CSI/CIR/CFR query and reference fingerprints can be compared in several different ways:

- Correlation,
- Time reversal technique [25], [28], [66],
- Multipath similarity metrics detailed in Section 2.5.1.3 and Chapter 3.

Example CSI Fingerprinting Schemes						
Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Magsino at al. [67]	2017	CFR fingerprinting + time reversal technique	Fingerprint database updated while system is online	**USRP** 11 carriers, 60-110 MHz, 5cm spacing between reference points, 1 AP, dynamic environment	correct identification of the point	100%, 5cm
Chen at al. [28]	2017	**WiFi** CFR fingerprinting	Time-reversal resonating strength similarity metric, leveraging frequency diversity, a large effective bandwidth achieved with frequency hopping	**USRP**	accuracy	1-2 cm
Chen at al. [66]	2017	**WiFi** CFR fingerprinting	Similarity metric: time-reversal resonating strength with residual synchronization errors compensated, CFR obtained from MIMO-OFDM system	**off-the-shelf WiFi device is equipped with 3 omnidirectional antennas, 3 × 3 MIMO configuration**	accuracy	1-2 cm

Tseng at al. [68]	2017	CIR fingerprinting, kNN, correlation similarity	Two fingerprint maps, measured and ray-tracing simulation	**Agilent E4438C vector signal analyzer**, center frequency 900 MHz and BW 100 MHz	RMSE, CDF	1.8-3.2m
Chapre at al. [27]	2014	CSI-MIMO fingerprinting with kNN	Similarity calculated as Euclidean distance between two fingerprints	**laptop, IWL5300 NIC**	mean error	0.89m
Sen at al.[69]	2012	**WiFi** CSI fingerprinting	CSI data adjusted before use, log likelihood similarity metric	**IWL5300 NIC**, measurements in rooms and offices, a cafe and a museum	Accuracy, false positives	≈ 90%, up to 7.3%

Table 2.5: Examples of CSI fingerprinting ILSs. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

Table 2.5 lists several publications that present different CSI, CIR, CFR fingerprinting algorithms. More CSI, CIR, CFR probabilistic and machine learning based fingerprinting approaches are listed in Tables 2.7, 2.8 and 2.9. In [28], the CFR is acquired through frequency hopping over a multitude of frequency bands. The experimental results in [28] demonstrate that, given a fingerprint database with 5cm spacing, the localization algorithm can achieve an accuracy of up to 1-2 cm in an NLOS environment.

However, conventional similarity metrics for CSI/CIR/CFR fingerprinting are still calculated using the complete channel information. They do not allow to identify the parts of the reference and query fingerprints that correspond to changes in the environment. Therefore, if any changes occur in the environment after the reference fingerprints were measured, a query fingerprint may be very different from the reference fingerprint measured at the same location. The similarity metric between a reference and query fingerprint measured at the same location will be lower than between the query and a reference fingerprint measured at a different location, and the localization estimate will not be correct. Therefore, a multipath fingerprinting approach is proposed in this thesis.

2.5.1.3 Multipath-based Fingerprinting

Table 2.6 lists several example multipath fingerprinting algorithms. As was mentioned in Section 2.3.3 and Fig. 2.2, the *multipath components* (MPCs) $\frac{d_1}{c}, ..\frac{d_K}{c}$ are the lengths of the different propagation paths along which the signal travels from the APs to the receiver. In some publications the term multipath components is used to describe the delays that correspond to the propagation paths $d_1, ..d_K$. Both definitions essentially describe the same thing as the delays of the propagation paths multiplied by the speed of light equal their lengths.

Example Multipath Fingerprinting-based ILSs.						
Authors	Year	Localization scheme	Algorithm remarks	Device, experimental setup	Evaluation metric	Results
Chen at al. [32]	2019	Spatial-temporal covariance matrix of the multipath signals used as a fingerprint, Euclidean distance between location descriptors, represented as points on the reproducing kernel hilbert space, used as the similarity metric	Robust to outliers, the query is first compared to larger clusters and then to the reference fingerprints within the best matching cluster, only one AP needed	**antenna array of a WLAN base station**, measurements in a warehouse, **3D ray tracing simulation**	CDF, 92.19 percentile, max error	1m, 2.5m
Kuperstein at al. [58]	2013	A lower dimensional signal subspace of the spatial-temporal covariance matrix as fingerprint, ML	Only one AP needed	**antenna array of a WLAN base station**, ray tracing wave propagation simulator	CDF, 90% errors, 78% errors	< 3m, < 1m.

Wielandt et al. [44]	2018	WiFi AoA fingerprinting, angular arrival spectrum used as a fingerprint	Fingerprint map generated with ray-tracing, correlation used to compare fingerprints	**vector network analyzer**, tested in several rooms, only one anchor node	50th and 95th percentile, CDF, normalized errors	-
Sousa at al. [40], [70]	2018 2019	TDOA ranging combined with with TDOA multipath fingerprinting under NLoS	Joint probability function of the possible points and TDoA results [70] machine learning approach [40]	**RF sensors with omnidirectional antennas**,[70], **3 dBi whip antenna, OFDM signal** [40]	mean, variance σ^2, CDF, 65% of errors <[40]	79-94%, 210m
Phelan at al. [71]	2012	Fingerprinting with mean and standard deviation of the delays, AoAs, and signal strength of the MPCs, the delay spread	Ray tracing is used to create the multipath database, measured feature vectors compared to statistical features of clusters calculated in the database	**ray tracing simulation**	signal strength and AoA plots for the simulated environment	-

Table 2.6: Examples of multipath fingerprinting ILSs. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

However, existing multipath-based fingerprinting algorithms again use the complete multipath information to calculate the similarity metric between a query and reference fingerprint. Same as most fingerprinting algorithms, their accuracy decreases significantly if changes occur in the environment after the fingerprint map is created. This is also the main difference between the state-of-the art and the localization algorithm proposed in this thesis and detailed in Chapter 3.

2.5.2 Fingerprinting Optimization

Once a similarity metric value has been calculated between the query and each reference fingerprint, the most straightforward way to obtain the users location is to pick the reference point corresponding to the reference fingerprint with the highest or lowest similarity metric value. However, many publications increase the localization accuracy of the algorithm by using filtering and averaging algorithms such as *k-nearest neighbors (kNN)* [57], [72]. The KNN algorithm calculates the user's location as an average of the K reference points for which the highest or lowest similarity metric value was calculated. *Weighted KNN* (WkNN) calculates the user's location as a weighted average of the K best reference points. The calculated similarity values can be used as the weights in the average.

2.5.3 Alternative Similarity Metrics and Fingerprinting Algorithms

There are a number of ways to calculate the user's location given a prerecorded database or fingerprint map and a measured query. The approaches in Section 2.5.1 calculate a similarity value between the query and the database entries. The location at which the reference fingerprint which is most similar to the query was recorded is picked as the user's location. Alternative approaches are presented in this section.

2.5.3.1 Probabilistic Fingerprinting Approaches and Maximum Likelihood

Example Schemes based on Probabilistic Fingerprinting						
Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Youssef at al. [73] HORUS	2005	**WiFi RSSI**, ML, discrete and continuous space estimator, Correlation Modeling and Handling	Location-clustering techniques to reduce the computational complexity	**Lucent Orinoco silver network interface card (NIC)**	Median, average error, σ, best 90%, Max	39-51, 42-64, 28-53, 86-132, 121-289 (cm)

Kuperstein at al. [58]	2013	A lower dimensional signal subspace of the spatial-temporal covariance matrix as multipath fingerprint, ML	Only one AP needed	**antenna array of a WLAN base station**, measurements in an office floor, 3D ray tracing radio wave propagation simulator	CDF	90% errors < 3m. 78% errors < 1m.
Wu at al. [29]	2013	**WiFi** CSI fingerprinting	ML approach, Pearson correlation used to calculate similarity	**laptop** with **IWL5300 NIC**	median accuracy, CDF	0.65m
Fazelinia at al. [74]	2019	WiFi RSSI, ML classification based on Bayesian decision theory (BDT)	For faster computation classifier's discriminant function is simplified into second order by taking a logarithm. Only features with high discriminatory information are used.	RSSI dataset	classification accuracy	97.9%
Yu at al. [75]	2021	**WiFi RSSI**, improved probabilistic linear discriminant analysis (PLDA)	Supervised learning	**WiFi APs and smartphones**	mean error	1.38m

Table 2.7: Examples of ILSs based on probabilistic fingerprinting. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

A maximum likelihood (ML) approach can be used instead of a similarity metric calculation to find the best matching fingerprint. The ML approach is implemented as follows. Given a measured fingerprint $\boldsymbol{s}^q$, the goal of the algorithm is to find a reference location $\boldsymbol{X}_i$ where the user is most likely to be located given that $\boldsymbol{s}^q$ was measured, that is the goal is to find $\boldsymbol{X}_i$ that maximizes the probability $P(\boldsymbol{X}_i|\boldsymbol{s}^q)$. The probability $P(\boldsymbol{X}|\boldsymbol{s})$ is unknown. However, according to the Bayes' rule:

$$P(\boldsymbol{X}_i|\boldsymbol{s}^q) = \frac{P(\boldsymbol{s}^q|\boldsymbol{X}_i) * P(\boldsymbol{X}_i)}{P(\boldsymbol{s}^q)}. \tag{2.4}$$

Therefore,

$$argmax_i P(\boldsymbol{X}_i|\boldsymbol{s}^q) = argmax_i P(\boldsymbol{s}^q|\boldsymbol{X}_i). \tag{2.5}$$

A distinguishing feature of probabilistic RSSI approaches is that their fingerprint map contains the probability distribution of RSSI or other fingerprint values measured for each AP at each reference location. A number of fingerprint samples measured at the reference point $\boldsymbol{X_i}$ during the calibration phase combined with a Gaussian model is typically used to approximate $P(\boldsymbol{s}^q|\boldsymbol{X}_i)$. Probabilistic CSI approaches generally approximate $P(\boldsymbol{s}^q|\boldsymbol{X}_i)$ in other ways, for example by the Pearson correlation between two fingerprints [29].

2.5.3.2 Machine Learning-based Fingerprinting

Machine learning is used for the following tasks by ILSs:

- Some ILSs improve the localization accuracy of a fingerprinting scheme by extracting relevant features from the fingerprints before they are stored in the map or used as a query. This way some ILSs also reduce the size of the calibration database. The following mathematical techniques are used:
  - Principle component analysis (PCA) .
  - The discrete cosine transform (DCT) is also used to decorrelate and compress RSSI vectors [76].
  - The independent component analysis is also used to decorrelate the entries in the RSSI vectors [76].
- The work in [77] clusters unlabeled RSSI measurements in a way that each cluster corresponds to a subarea in the floor plan.
  - RL-clustering [78] is used in [77] to cluster RSSI fingerprints.
  - Multi-dimensional scaling is used by [77] to obtain a 2D representation of the RSSI vectors in such as way that the distance between similar RSSI fingerprints is as small as possible. The multi-dimensional scaling coordinates are used to map RSSI clusters to the subareas.
- Multiple classifiers can be trained to predict the location of the user, as is done in [79]. The fusion can be used to calculate the users location from the output locations of the classifiers. An example classifier is the linear discriminant analysis.

- Identification of NLoS conditions from a CIR is done in [80] using linear discriminant analysis and support vector machines.

Example Machine Learning-based Schemes						
Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Liu at al. [81]	2021	RSSI fingerprinting with k-means weighted K nearest neighbors (WKNN)	K-means algorithm used to divide reference measurements into non-overlapping clusters. One representative sample per cluster calculated. New fingerprint set that is smaller and more robust to noise.	**simulation**, **UjiIndoorLoc dataset**	mean error, CDF	1.45m-8.77m
Wang at. al. [82]	2021	**WiFi** CSI fingerprinting with PCA and DWT	DWT used for de noising, PCA for dimentionality reduction, eliminate redundancy	**laptop**, **Intel IWL5300**	MSE, σ	1.37, 6.52m
Guo at al. [79]	2019	**WiFi** RSSI fingerprinting with pre-trained classifiers	Fusion of the output of multiple classifiers. unsupervised fusion determines the users location from the location estimates which are above a probability threshold from each classifier.	**WiFi APs, smartphone**	RMSE	2.6m

Han at al. [83]	2019	LS Trilateration with **WiFi** ToF, time measurements used for LoS identification.	Support vector machines are used with Fine Time Measurements and RSSI.	**phone** measurements	accuracy of NLOS identification, RMSE	0.766, 2.296m
Salamah at al. [84]	2016	**WiFi** RSSI fingerprinting with principle component analysis and random forest classifier	PCA is used to create an uncorrelated space and make the fingerprints smaller, a random forest classifier is used to calculate the corresponding location.	**smartphone**, hall of an apartment with randomly moving people.	mean error, variance, RMSE	1m
Fang at al. [76]	2008	**WiFi** RSSI maximum-likelihood based fingerprinting on a reduced dimension space	PCA, independent component analysis and discrete cosine transform (DCT) applied to the RSSI data to compress it before fingerprinting	office space, **laptop**	average error, variance of the error, CDF	0.86m for 3APS 0.5m for 15 APs

Table 2.8: Examples of machine learning based ILSs. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

2.5.3.3 Deep Learning and Neural Network-based Fingerprinting

Neural networks have been used for localization in the following ways:

- Training a neural network with labeled reference fingerprints and using the network to calculate the locations of query fingerprints [26], [85]. As illustrated in Fig. 2.5, instead of calculating a similarity metric between the query and each reference fingerprint stored in the database, the trained neural network becomes the fingerprint map and also performs the similarity metric calculation and filtering.

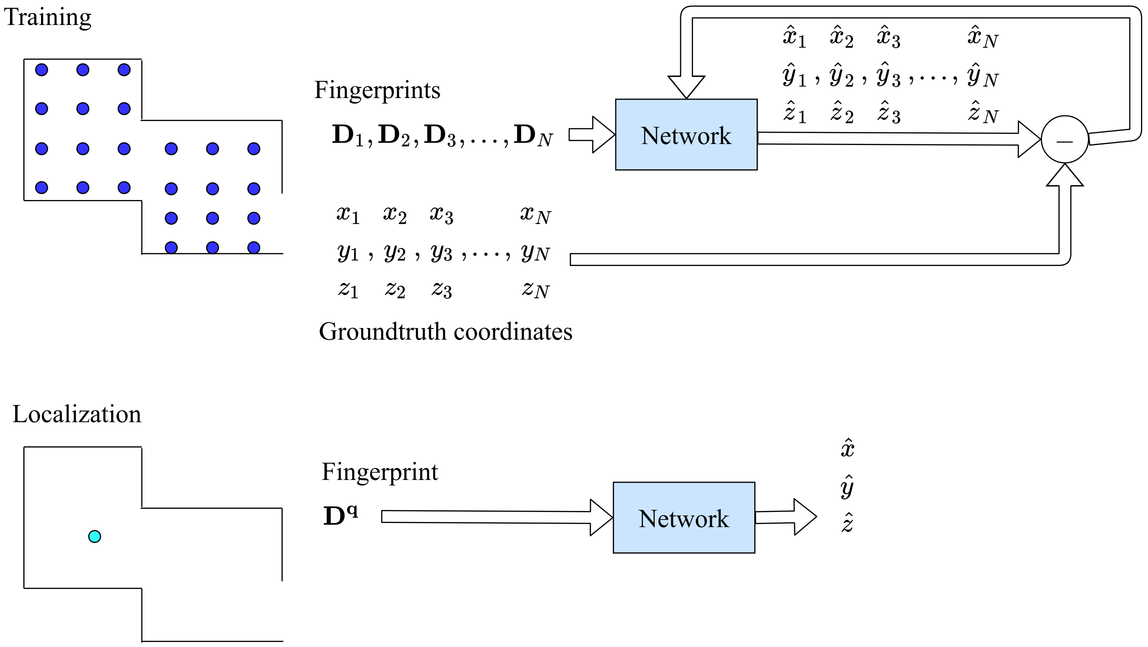

Training
Fingerprints
$\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3, \ldots, \mathbf{D}_N$
Network
$\hat{x}_1 \ \hat{x}_2 \ \hat{x}_3 \ \hat{x}_N$
$\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_N$
$\hat{z}_1 \ \hat{z}_2 \ \hat{z}_3 \ \hat{z}_N$
$-$
$x_1 \ x_2 \ x_3 \ x_N$
$y_1, y_2, y_3, \ldots, y_N$
$z_1 \ z_2 \ z_3 \ z_N$
Groundtruth coordinates
Localization
Fingerprint
$\mathbf{D}^\mathbf{q}$
Network
$\hat{x}$
$\hat{y}$
$\hat{z}$

Figure 2.5: Fingerprinting-based localization using a deep neural network.

- In the *calibration phase* the network is trained using a set of pre-measured fingerprints and the locations at which they were recorded.
- In the *on-line phase* the user's fingerprint is the input to the network and his or her location is the output.

This can be done either by

- Training the network with labeled reference fingerprints, that is reference fingerprints the locations of which are known. This is referred to as *supervised learning*. Such a network is equivalent to classical fingerprinting. However, a network has the ability to filter and interpolate the reference fingerprints, filter noisy data. Therefore, a network-based approach has the potential to achieve higher accuracy with calibration measurements than a classical fingerprinting algorithm. This is shown in [85].
- Training the network with a small amount of labeled fingerprints and a larger number of unlabeled fingerprints. This is referred to as *semi-supervised learning*. The network uses the unlabeled data to find patterns it can use. The network training algorithms require a large amount of unlabeled fingerprints. Fingerprint data with location labels is expensive to collect [86]. The location of the receiver needs to be obtained through manual measurement or external ILS. Unlabeled data, in contrast, is cheap to collect [86]. The receiver can be randomly moved around the environment continuously recording measurements. Crowdsourcing as detailed in Section 2.5.5, where the ILS adds measurements collected by users

  to the fingerprint map, is another approach used to cheaply obtain a large number of unlabeled fingerprints.

  - Training the network only with unlabeled data. This is referred to as *unsupervised learning*. The work in [87] uses the floor plan, an unlabeled RSSI data that was collected in a known sequence and a hidden Markov model to estimate the position of each reference fingerprint. The work in [88] uses deep reinforcement learning with unlabeled RSSI data. Features are identified in the data and the calculated location of those features are used as reference. The initial features are the locations close to the APs which are identified using the near field model and the RSSI. While the calibration phase of these algorithms is the least time consuming, the author of this thesis is not aware of a high-accuracy unsupervised indoor localization network.

- Using a neural network to transform the fingerprints before they are used in a localization algorithm. This is similar to some of the machine learning techniques presented in Section 2.5.3.2. The network will either reduce the dimension of the fingerprint vectors or extract the most important features [89]. The following networks are often used:

  - An autoencoder trained by the unlabeled data. An autoencoder network has a butterfly shape with much fewer neurons in the middle then in the input and output layers. Therefore, the input of the autoencoder is essentially encoded by the left side of the network, the output of the smallest middle layer of neurons is equal to the encoded data, the right side of the network decodes the data.
  - A network trained by a small portion of labeled data and a large amount of unlabeled data.

- Using an unsupervised network such as an autoencoder to extract important features from the fingerprint vectors and another network can be trained to use those features to calculate the users location.

- The work [90] uses a neural network to create two databases, one for a noise-free assumption for the data and the second database considers the data to be noisy. Once the user obtains his or her measurements the ILS uses the variance of the data to determine whether the measurements are noisy and which database needs to be applied.

Example deep learning-based localization algorithms are listed in Table 2.9.

Example Deep Learning-based Schemes						
Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Gassner at al. [91]	2021	CSI CNN fingerprinting	Convolutional neural network (CNN) trained on labeled CSI (Fourier transform of CIR) fingerprints	**USRP B200mini** attached to mobile robot, **LTE signals**, fingerprint collected over a $3.5m \times 5m$ area	average MSE of x and y coordinates, median error	<200cm, 175cm
Gönültaş at al. [92]	2021	**WiFi** CSI ML localization	Neural networks (probability maps) used to combine probability estimates from each AP, inverse discreet Fourier transform (DFT) and auto-correlation used for feature extraction	**two antenna IEEE 802.11ac transceiver**	mean distance error, median, 95th percentile	0.05 - 0.7m
Hao at al. [93]	2021	Deep neural network RSSI fingerprinting	Multi-objective regression DNN combines RSSI measurements from different WiFi channels sets	**smartphone, laptop, tablet**	RMSE	1.508m-1.815m

Chen at al. [94]	2020	Fingerprinting with environment identification, labeled samples	A convolutional neural network (CNN) performs indoor environment identification by extracting the inherent features, another CNN performs localization taking environment type into account	**vector network analyzer, omnidirectional antennas**	RMSE	0.28-0.37m
Sandhu MSc. [85]	2019	Fully connected network with 3 layers	Multipath data	**Ray tracing simulation**	mean error	down to 4cm , 20 cm with a 1m reference grid
Li et al. [88]	2019	RSSI fingerprinting, fingerprint map trained with a unsupervised data	The continuous localization task is modeled as a Markov decision processes, making it possible to calculate the location with deep reinforcement learning. Features are extracted from the RSSI data. Pass-loss model used initially for setting the rewards to the reinforcement learning.	tests in an open field, **Bluetooth**	RMSE, 95% quantile	12.2m, 24.7m

Chidlovskii at al. [86]	2019	Variational auto-encoder trained with RSSI data	Only some of the data is labeled, more unlabeled data is used.	**WiFi RSSI UJI-IndoorLoc dataset [95]**	RMSE	4.65m
Abbas at al. [96]	2019	WiFi RSSI Fingerprinting, deep learning used to extract the correlation between signals from different APs.	Stacked auto-encoders used for de-noising, regularization used to avoid over fitting, a deep network is trained for each reference point, artificial noise added to the data before it is used to train the autoencoders. The output of the autoencoders is used for probabilistic localization.	629 m^2 area with 29 reference points and 65m^2 area with 81 reference points. **Android phones**.	mean error, cdf, 50th, 75th and 100th percentiles	629m^2 - 2.64m, 2.38m, 3.38m, 7.12m, 65m^2 - 1.21m, 1.07m, 1.62m, 3.74m
Berruet at al. [97]	2018	CFR fingerprinting using a convolutional neural network (CNN)	Trained on a limited number of labeled samples, one one data packet needed for localization	5 GHz **WiFi** data, **channel sounder**, 5 room apartment	90th and 99th percentiles, median, mean error, CDF	4.36m, 7.21m, 1.55m, 0.14m
Hsu at al. [90]	2018	RSSI Fingerprinting	Deep belief network used to build an adjust the fingerprint database	**WiFi RSSI measurement** in a parking garage	mean error	>1.2m

Mohammadi et al. [98]	2018	BLE RSSI variational auto encoder	Part of the dataset labeled, most unlabeled	BLE RSSI measurements in a library	mean error, error percentages	4.3m, 47% $\geq$3m
Khatab et al. [99]	2018	WiFi RSSI Autoencoder-based extreme deep learning machine (ELM)	Features learned by the autoencoder are used by the ELM	**simulation on the path loss model**	correct classification	92.82%
Le at al. [89]	2018	WiFi RSSI support vector machine trained on labeled features for fingerprinting	Feature extraction on unsupervised data with PCA + deep belief network (DBN), features calculated from labeled data and used to train support vector machine	**UJIIndoorLoc dataset** with 10% to 100% of data used as labeled data. 1%	RMSE, CDF	>6m
Szabo et al. [100]	2011	Bluetooth CIR fingerprinting	Fingerprint map initialized with data generated by a model, labeled measurements used improve the map.	**Bluetooth tag**, lab environment	mean error, σ, CDF, percentiles at 50% and 90%	0.68m, 0.54m, 0.54m, 1.47m
Nerguizian at al. [65]	2007	CIR fingerprinting with generalized regression neural network	Discrete wavelet transform (DWT) applied to the CIR, the result is the input of the network	**simulation**	precision for 91% trained and 70% untrained data, CDF	2m

Wang at al. [101]	2017	AoA and CSI fingerprinting	Autoencoder network, three hidden layers and one input layer	**Intel 5300 NIC, Dell laptop**	mean error, σ	1.5m-2.2m, 0,8m-1.5m

Table 2.9: Example deep learning based ILSs. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

The main disadvantage of network-based ILS is the requirement that the structure of the data cannot change after the network was trained. In a master thesis, written at the Chair of Media Technology at the Technical University of Munich [85], the multipath delay profile was used as the fingerprint structure. The simulation results showed that the selected convolutional neural network succeeded at interpolating between the reference fingerprints and produced a high accuracy even with a smaller number of reference fingerprints. The ILS also showed a high accuracy despite a noisy input. However, a neural network requires a very structured input. When changes in the environment modified the order or removed or added multipath components to a fingerprint, the network produced high localization errors.

2.5.4 Main Challenges of Fingerprinting-based Systems

2.5.4.1 Calibration and Maintenance Effort

For most fingerprinting algorithms, the more entries the fingerprint map contains the higher the localization accuracy will be. The accuracy of deep learning and machine learning approaches also increases with the amount of training data. In order to gather labeled fingerprints, a person or robot needs to go to each reference location to record the fingerprint and the ground truth coordinates. The ground truth can be collected though manual measurements, SLAM using a laser scanner, Lidar or certain infrared or ultrasonic localization systems. The construction of the fingerprint map is time consuming and the map needs to be recomputed every time structural changes occur in the environment [102]. This increases the overall cost of fingerprinting ILSs. One possible solution to this problem is an approach where the fingerprint map is constantly updated by the data received by the ILS from the users of the localization system [103] and [20]. Either the ILS calculates the locations of the new crowdsourced fingerprints using a localization algorithm or IMU data, or a deep network or machine learning is used to update the map without calculating the locations of the unlabeled crowdsourced fingerprints. In some cases the system uses IMU data to track the users displacement and direction of movement and fuses that information together with the current and previous reported locations to update the fingerprint map. If machine or deep learning is used, labeled crowdsourced data can be use to find useful features in the fingerprint vectors.

2.5.4.2 Temporal Variations of the Channel

Due to the movement of people and objects or the opening and closing of doors, the channel between a receiver and a transmitter will not remain constant over time [8]. In [8] the author provides measurements and experimental data on temporal variations of indoor channels. In [69], [104], [105] the influence of the presence of people on an RSSI fingerprinting-based localization system is experimentally measured. The results presented in [104] show that the presence of people can increase the localization error by 11% on average and by as much as 50% in corridors. The work done within a master thesis at the Chair of Media Technology at the Technical University of Munich [106] showed the variations in the CSI fingerprint map when people moved around the indoor environment. In [67] the CFR fingerprints are shown to become uncorrelated within a 15min window in a dynamic environment. The author of [67] suggests slowly updating the fingerprint database as the ILS is on-line.

In addition, certain APs (equipment from Netgear, Cisco and Aruba Networks) can monitor the signal propagation environment and automatically adjust the transmit power to optimize overall network performance [62]. To overcome the variations in AP transmit power [62] proposes a system that relies on the RSSI gradient and not absolute value fingerprints. This approach, however, cannot account for global changes in the environment.

2.5.4.3 Device Variation

The work in [105] details the differences in the measured RSSI that can be observed by different smartphones. There may be differences in fingerprints of other formats if different devices are used for creating the map and by the user.

2.5.5 Crowdsourcing Fingerprints

Crowdsourcing approaches both reduce the calibration effort of an ILS and make it more robust to changes in the environment. An initial fingerprint map is created when the ILS is setup. During the online-phase measurements collected by users are added to the fingerprint map. Crowdsourcing is active when the user lets the ILS know at which location the fingerprint was collected, for example though PDR or data fusion. It is passive when the user only send the measured fingerprint and not his location to the ILS [77].

Crowdsourcing algorithms can be implemented the following way:

- Odometry or an IMU is used to calculate the location of the new fingerprints. The calculated location and measured fingerprint are added to the map. Filters can be used to improve the fingerprint map quality [107].

- The location of a new fingerprint is estimated using the existing map. The new fingerprint and the calculated location are added to the map.

- The work in [87] uses the floor plan and a hidden Markov model to construct and afterwards update the fingerprint map with unlabeled reference fingerprints.

A number of crowdsourcing approaches are listed in Table 2.10.

Example Crowdsourcing Fingerprinting ILS.						
Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Chai at al. [108]	2007	**WiFi RSSI** Probabilistic fingerprinting with crowdsourcing	Unlabeled sequential data provided by the user used to improve the fingerprint map	office area, **laptop**	accuracy	close to 90%
Chen et al. [77]	2015	**WiFi** RSSI fingerprinting, unsupervised crowdsourcing	Crowdsourced fingerprints are clustered together, a floor plan is used to map clusters to subareas.	office rooms, **smartphone**	sub-area correctness rate	90% with raw RSSI to 95% with averaged RSSI
Jung at al. [87]	2018	WiFi RSSI fingerprinting	Unsupervised learning and a floor plan is used to build the fingerprint map from unlabeled fingerprints. Fingerprint map updated.	**WiFi RSSI smartphone measurement**	average error	>2m

Yu at al. [107]	2019	RSSI fingerprinting with IMU-based crowdsourcing	Fused and weighted data used to create a higher quality database, detrministic and Baysean similarity metrics used		CDF, accuracy of 80% of test points, mean error deterministic and probabilistic fingerprints	> 2.08m, 2.438m, 1.556m
An at al. [109]	2021	Manifold regularization used to map WiFi RSSI fingerprints onto a lower dimensional subspace, semi supervised learning, labeled and unlabeled fingerprints	Reduced computational requirements, privacy preserving localization	UJIIndoorLoc	mean error, min error, max error	2.67m, 0.5m, 34.14m
Xiang at al. [110]	2021	Multi-kernel transfer learning for crowdsourcing	Multi-kernel maximum mean discrepancy with WiFi CSI distributions, IMU data, deep transfer learning network for localization	smartphone	CDF, 50th percentile	1.08m

Zhao at al. [111]	2021	**WiFi** RSSI, AoA and IMU data integrated into a graph formulation	Nodes represent the positions of devices or user steps in a time series. The edges are estimated based on the IMU, WiFi data. multidimensional scaling (MDS) algorithm used to solve graph formulation	**smart-phone**	CDF, accuracy	1.71m
Jiang at al. [112]	2021	**WiFi** RSSI measurements, gyroscope, reference fingerprints and locations stored in a graph	Multi-dimensional scaling used to compute the distance between two reference locations from WiFi fingerprints, fingerprints grouped into clusters	**smart-phone**	error and accuracy CDFs	80% < 4m

Table 2.10: Examples of crowdsourcing-based fingerprinting ILSs. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

2.6 Summary of Existing Multipath-based Localization Schemes

2.6.1 Multipath Simultaneous Localization and Mapping (SLAM)

As the indoor localization system proposed in this thesis uses the multipath components of the received signal as a fingerprint, this subsection summarizes existing multipath-based localization schemes.

Two categories of multipath-based localization approaches can be distinguished. In the first category of algorithms the multipath components are used to calculate more compact features. Those features are then used to calculate the location. Such features include points on the reproducing kernel Hilbert space calculated from the spatial-temporal covariance matrix of the multipath signals received by the base-station antenna array [32]. In the second algorithm class, the multipath components are used directly for localization and for obtaining information about the geometry. Example multipath-based fingerprinting algorithms are summarized in Table 2.6 and Section 2.5.1.3.

Multipath components are also used in multipath SLAM algorithms summarized in Table 2.11. In this case, multipath components measured by a moving user are associated to so-called virtual transmitters (VTs), detailed in Chapter 6. The user's motion data, and the differences between the multipath components measured along the user's trajectory are used to determine which multipath components corresponds to which VT. The measured multipath delays and estimated VT locations and a Kalman or particle filter are then used to localize the user.

Example multipath SLAM-based ILSs.

Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Gentner at al. [43]	2016	Recursive Bayesian filtering called Channel SLAM	VTs estimated from multipath components of a moving receiver, Kalman enhanced super resolution tracking (KEST) used to estimate multipaths	**MEDAV RUSK-DLR broadband channel sounder** 1 mW multitone signal, N = 1281 subcarriers , center frequency of 1.51 GHz, bandwidth of B = 100 MHz	RMSE of the estimated VT positions, localization RMSE	5cm - 1m
Leitinger at al. [113], [114]	2019	Multipath SLAM with Baysean probabilistic feature estimation	VTs used as features, feature and location belief iteratively updated	**UWB**, **channel sounder**, from 3 to 10GHz, 2GHz BW a center frequency of 7GHz [113], [114], **simulation** [114]	CDF [113], [114], RMSE over time moving user, max error, max error of min90% [114]	max error 6cm [113], simulation <0.12m, measurement 0.2m, 0.083m [114]

Ulmschneider et al. [115]	2020	Channel SLAM extended with transmitter visibility information	Use of transmitter visibility maps of the environment	**simulation**	Mean absolute error, 95th and 5th percentiles	< 5m
Gentner at al. [47]	2021	Channel SLAM	Environment reconstruction performed	**UWB, Decawave DW1000**	accuracy in 95% of cases, trajectory plots	$\approx 1m$

Table 2.11: Examples of multipath SLAM ILSs. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

2.6.2 ILSs Based on Sensor Fusion

In order to increase the accuracy of an ILS, locations calculated by multiple algorithms and the data from multiple sources and sensors are often fused together. Fusion is most commonly used with RSSI fingerprinting described in Section 2.5.1.1, as RSSI fingerprinting on its own has low accuracy and RSSI data is cheap and easy to acquire. In general, the accuracy of any localization scheme can be improved with filters, algorithm or sensor fusion in the following ways:

- If the user is moving:
    - As was discussed in Section 2.2, if the user is moving the data from the IMU and from other sensors is often used to improve the localization accuracy of an algorithm. An IMU would give a noisy estimate of the direction and distance in which the user moved since the last sensor measurement. IMU data can not be integrated over time to obtain the users location as the noise accumulated over each measurement makes the calculated location imprecise. RSSI fingerprinting can be used to correct the IMU drift.
    - The fact that estimates of the users location made one after the other should be next to each other is space and the users motion should be smooth, can be used to filter the output of a fingerprinting algorithm.
- If the user is stationary or moving:

- Data from other localization schemes, such as magnetic field fingerprinting, can be fused together with electromagnetic fingerprinting or multilateration.

The following filters are most commonly used for calculating the location of the user based on data from multiple sensors and algorithms:

- Kalman filter detailed in Section 2.2.
- Particle filter detailed in Section 2.2.
- Averaging filters such as the *k-nearest neighbors* (KNN), applied to multiple estimates.

Table 2.12 lists a number of publications that present localization algorithms based on fingerprinting fused with additional sensor data or motion filters. It should be noted that any fingerprinting or trilateration approach described in this chapter can be used for data fusion. However, it is most common for RSSI fingerprinting since the accuracy of the fingerprinting scheme on its own is very low but RSSI values are measured by default by hand-held devices. In addition, an IMU is usually inbuilt in a smartphone. A filter or fusion algorithm can be added to any localization scheme, the more accurate the positioning performed by the original scheme, the more accurate the output of the filter or fusion will be.

Example Fusion-based Schemes						
Authors	Year	Localization scheme	Algorithm remarks	Evaluation setup	Evaluation metric	Results
Yu at al. [116]	2021	**BLE** RSSI fingerprinting+ **inertial sensor** PDR + **magnetometer** + adaptive unscented Kalman filter	Dynamic-time-warping algorithm used to detect the local BLE landmark from RSSI	**BLE nodes use TI's CC2640 chip, smartphone**	trajectory, CDF, average error, RMSE, 50% accuracy, 90% accuracy	0.75m, 1.02m, 0.57m, 1.12m
Zhong at. al. [117]	2018	**UWB** TDoA + IMU + EKF	Designed to used **IMU** data under NLoS conditions, processing done by server	6 **tags** and 10 **UWB antennas**	trajectory plots, error over trajectory	majority < 0.2m

Zhang et al. [118]	2018	**WiFi** + **BLE** + RSSI fingerprinting + propagation model localization, landmarks + PDR	Fusion done with a particle filter, the fingerprint database is built in the online phase, fingerprinting used for initial localization	3000m^2 office building, smartphone	average error, RMSE, CDF, max positioning error, average initial positioning error	1.11m, 1.26m, 2.77m, 2.15m
Ettlinger et al. [119]	2016	Fingerprinting(**magnetic** + **WiFi** + **RFID**) + Kalman filter	WiFi fingerprinting for an approximate solution, refined though magnetic fingerprinting, Euclidean and Mahalanobis distances used to compare fingerprints	office environment, static positioning - 10 test points, kinematic positioning - walk through a corridor	average positioning error static, average lateral deviation from the reference trajectory	2.04m, $\approx$1m
Driusso at al. [120]	2016	EKF + triangulation + ToA from CFR estimation + ESPRIT and Kalman filter for ToA tracking	**LTE signals**	**USRP**, 3LTE service providers and 5 **base stations**, indoor environment	RMSE 50th and 95th percentiles, CDF	4.8m, 3.19m, 9.21m
Van Haute et al. [61]	2016	**ZigBee** TOA + particle filter*	Active hospital environment	error evaluation over a grid of known points	average error, room level accuracy, latency	3.89m, 47%, 0.5s

Nurminen at al. [121]	2015	ToA, PDR with **three-axis accelerometers** and gyroscopes, EKF	Skew-t measurement noise model used top detect NLoS outliers	**UWB beacons**	RMSE, mean, median	1.23m-0.75m
Jiang et al. [122]	2014	Crowd-sourcing + **WiFi** CSI + WiFi RSSI fingerprinting	Moving speed and distance of the user estimated from the 802.11n CSI, mapping between RSSI and location built using unsupervised learning	localization and tracking - a user walks on a predefined testing trace for 5 rounds.	performance of individual algorithm components, CDF, upper bound of smallest 80% of errors	≈2m
Jiang et al. [123]	2012	**WiFi** RSSI-based fingerprinting + **accelerometer** data	Crowd-sourcing, no manual room annotation or map creation	193 rooms in a wide range of buildings, 21 **smartphone** users	correct room identification	95%
Wang et al. [124]	2012	Crowd-sourcing + SLAM - fingerprinting (**WiFi** RSSI + urban sensing landmarks + floor plan) + PDR	Recognize landmarks, e.g. turns, from **inertial sensors** and WiFi data, use landmarks to improve PDR and PDR to refine landmark positions, no calibration needed	3 **smartphone** with 2 phones (hand and pocket) walk around, independent ground-truth	landmark detection performance, mean localization error, CDF	1.69m

Woodman [125] [126]	2008 2010	Particle filter + floor plan + PDR	Particle filter initialized with a uniform distribution, number of particles in the filter adapted dynamically, **foot mounted IMU**	5 walks through a three floor building each 10-15 minutes	upper bound of 95% of errors	0.62m [126]
Beauregard et al. [127]	2008	Backtracking particle filter + floor plan + PDR	State estimate refined based on particle trajectory histories, **foot mounted IMU**	10 min walk in and out of an office building + stairs	mean localization error	0.74m

Table 2.12: Examples fusion-based ILSs. The hardware platforms and communication standards used to implement the algorithms are highlighted in bold blue text.

2.7 Practical Aspects of ILS Implementation

2.7.1 Communication Standard

The receiver and APs communicate with each other over certain frequencies and protocols. The following common communication standards are summarized in Table 2.13.

Communication standards used for indoor localization				
Protocol	Frequency	Devices	Common ILS types	Comments
WiFi IEEE802.11a-n	2.4 GHz and 5GHz, bandwidth 40MHz, range 50m to 100m	**Smartphone, laptop WiFi cards, IWL5300**	RSSI and CSI fingerprinting, fingerprinting+fusion, RSSI ranging, multipath fingerprinting [32], [45]	Generally cheap due to the availability of routers and measurement of RSSI by smartphone WiFi cards. Produces low accuracy localization with a few meters, generally fused with smartphone sensors such as an IMU.

ZigBee IEEE 802.15.4	most commonly uses the 2.4 GHz band, range 10m to 100m	wireless sensor network sensors and nodes	RSSI, ranging	Available in **wireless sensor network devices**, not available on user devices such as smartphones.
RFID	ultra high frequency, microwave frequency, range more than 100m(active RFID), 1m (passive RFID)	**RFID tags and reader**	RSSI, ranging proximity	Reader receives data emitted by tags, RFID tags are cheap and small, not available in user devices such as smartphones.
Bluetooth IEEE 802.15.1	2.4GHz, bandwidth 1MHz to 2MHz, range 10m to several 100m	**BLE beacons**	RSSI fingerprinting, ranging	Low energy (LE) devices can be used.
UWB IEEE 802.15.4-2015, IEEE 802.15.6-2012	3.1GHz to 10.6GHz, bandwidth larger than 500MHz, range 80m	**Channel sounder, Decawave chip**	Ranging, CSI and CFR fingerprinting	high accuracy 1cm-10cm localization is possible, high sampling rates.
Custom frequency range	-	**Channel sounder, spectrum analyzer**	used in prototype systems	High accuracy 1cm-10cm localization is possible

Table 2.13: Communication standards used for indoor localization

2.7.1.1 WiFi

The 801.11 WiFi communications standard is the prime candidate for implementing an indoor localization systems. WiFi access points and routers are already present in most public and private buildings. Most areas in larger buildings receive a signal from at least one WiFi router. If a ILS is developed based on WiFi no additional hardware needs to be setup and ILS installation only requires a software update on the routers and possibly a calibration of the fingerprint map. This would significantly reduce the cost of the ILS.

Most commonly WiFi is used with RSSI fingerprinting and fusion systems. With several exceptions the accuracy of such systems is in the range of several meters.

2.7.1.2 UWB

Due to it's high bandwidth (BW) the ultra-wideband communication standard is becoming more and more popular for high-precision localization. As it's name suggests, UWB signals have an absolute bandwidth, that is the difference between the upper and lower frequencies, larger than 500 MHz and a very good time-domain resolution [128]. This makes them well suited for localization, this is also the feature of UWB which makes it possible to calculate individual multipath delays. Because of their ultra-wide absolute bandwidth, UWB signals also do not significantly interfere with conventional communication systems [128]. Very short pulses are used in impulse radio transmission with careful signal and architecture design. This allows UWB transmitters to have a very low power consumption with an average across devices of 30 mW [128]. A detailed description of the UWB communications standard can be found in [128]–[130]. Hardware created for UWB localization includes

- Decawave DW1000 chip that provides ranging with an accuracy of around $10cm$ and CIR estimation.
- UWB solution integrated into a smartphone by the company BeSpoon [131]. The UWB system offers distance estimation with an accuracy down to $10cm$ within a $880m$ range. The UWB antenna can also be used for WiFi communication.

In this thesis the ultra wide-band (UWB) technology is selected for the communication between the receiver and the APs. This was done for two reasons:

- The UWB standard allows for the use of a wider bandwidth, has a very good time domain resolution and thus has the potential for high precision multipath estimation.
- As explained in the Section 2.7.2.3, the Decawave dw-1000 chip [7] is an off-the-shelf UWB transceiver that can be used for precise CIR and multipath estimation.

2.7.1.3 Bluetooth low energy (BLE)

Bluetooth low energy (BLE) is a standard often used for communication between devices and in wireless sensor networks (WSNs). As their name suggests, an advantage of BLE receivers or tags is their high energy efficiency. BLE can be used for RSSI, ToA and AoA-based localization [132]. RSSI BLE systems same as WiFi-based RSSI ILS tend to have a rather low accuracy. As can be seen from Table 2.18, many commercial systems do use BLE. Some combine BLE with UWB. In that case BLE is often used for AoA estimation such as in [133].

2.7.1.4 ZigBee

ZigBee is a communication protocol that is used for personal area networks and wireless sensor networks (WSN), is low cost, low data rate and energy efficient. ZigBee is often supported by WSN sensors but is not often supported by user devices [132].

2.7.1.5 RFID

The Radio Frequency Identification Device (RFID) communication system is commonly used for data transfer. A reader communicates with RFID tags. The RFID tags emit data using certain frequencies and a certain communication protocol that are known to the reader a priori. Active RFIDs use the Ultra High Frequency (UHF) and microwave frequency range. They are generally set up to transmit their ID at regular intervals. Passive RFIDs do not require an LoS to the reader, however they can communicate with the reader only in a small rage of 1-2m. Active RFID tags can be located more than a hundred meters away from the RFID reader and can be used for localization. They are also cheap and small [132]. RFID is well suited to track objects in a warehouse, however, RFID technology is not available in most user devices such as smartphones.

2.7.1.6 LTE signals

Signals from mobile networks are also used for localization, for example in [120]. However, publications or commercial ILS that make use of LTE signals are not common.

2.7.1.7 Custom signals

When a channel sounder or spectrum analyzer is used to implement a prototype localization system, the developers can choose the spectrum and bandwidth of the signals. Since a spectrum analyzer or a channel sounder are very expensive and regulations do not allow private use of certain frequency bands, the ILS needs to be adjusted to one of the above mentioned communication standards before it can be made commercially available.

2.7.2 Devices Used as the Receiver

A receiver device communicates with the APs and calculates the signal properties used by the localization algorithms. To provide an overview of how often and for which applications different ILS receivers are used, they are highlighted in bold blue text in Tables 2.1-2.15.

Devices used as a Receiver by ILSs.					
Device	Communication Standard	Localization Algorithms	Accuracy	Comments	Examples
Smartphone/ laptop	**WiFi**	RSSI fingerprinting, fusion with IMU and other sensor data	1m-15m	Use of in-built communications chip	[63], [75]

Universal software radio peripheral (USRP)	any	any, CFR fingerprinting [67]	5cm [67]	Any ILS can be implemented on the device	[67]
Intel IWL5300 chip[134]	**WIFI**	CSI fingerprinting	30cm	In laptops, calculates power and phase for groups of OFDM carriers, currently not in production	[26], [29]
WLAN base station	**WiFi**	multipath fingerprinting	1m	Multi-channel receiver equipped with a circular antenna array, standard communications hardware	[32]
Decawave board [7]	**UWB**	CSI, CIR, CFR and multipath fingerprinting, multilateration	2cm-30cm	Performs channel estimation, ranging measurements with 10cm accuracy.	[15], [47]
Channel sounding equipment (vector analyzer, channel sounder)	any	Fingerprinting and multipath SLAM	high accuracy depending on the algorithm	Most precise measurements, can cost in the range of 200k USD.	[43], [135], [136]
BLE tag	BLE	RSSI fingerprinting	1m	Low power, standard devices	[88], [116]
RFID tag, reader	RFID	RSSI fingerprinting, proximity sensing	several m	low power, WSN sensors	[137], [138]

Custom chip	any	any	down to 10cm	In commercial solutions	

Table 2.14: Devices used as a receiver in ILSs.

2.7.2.1 RSSI Measurement - Smartphones and Laptops

RSSI is automatically measured by most WiFi cards, therefore RSSI-based ILS are implemented on phones and laptops. Phone-based ILS are cheap and have automatic access to IMU data. RSSI is also measured by BLE beacons. The RSSI can be calculated from the CSI, CFR and CIR measured by devices such as the IWL5300, channel sounder or Decawave chip, however this is generally not done.

2.7.2.2 IWL5300 - WiFi CSI Standard

The Intel N Ultimate WiFi Wireless Link 5300 [134] (IWL5300)[26], [29] is a IEEE 802.11a/b/g/ Draft-N1 wireless network adapter that operates in both the 2.4 GHz and 5.0 GHz spectra and delivers up to 450 Mbps of bandwidth The features of the IWL5300 include

- Up to 450 Mbps of bandwidth,
- IEEE 802.11a/b/g and Draft-N1 compliance,
- Low power consumption,
- Advanced security via 802.11i,
- Easy-to-use Intel® PROSet v12.0 WLAN software,
- Support for Cisco compatible extensions* v4,
- Performance-optimized with connect with Intel® Centrino® processor technology certified access points.

The Intel IWL5300 became a popular choice for ILS implementation due to the existence of the Linux CSI tool [139] [140]. The CSI tool obtains the CSI from channel measurements performed by Intel Wi-Fi Wireless Link 5300 802.11n MIMO radios, using a custom modified firmware and open source Linux wireless drivers [139]. The CSI obtained by the IWL5300 is structured as channel matrices for 30 subcarrier groups, which is about one group for every 2 subcarriers at 20 MHz or one in 4 at 40 MHz. Each channel matrix entry is a complex number, the real and imaginary parts of which have a signed 8-bit resolution. Each channel matrix entry specifies the gain and phase of the signal path between a single transmit-receive antenna pair [139]. Because the CSI Tool only works with the IWL5300, the IWL5300 continues to be used, although it was taken out of production by Intel. The latest available driver for the IWL5300 is from 2017 [141].

Laptops that are compatible with or are sold with the IWL5300 network interface card (NIC) include [142]:

- Lenovo ThinkPad X200, Elite ThinkPad R500, Elite ThinkPad R400, ThinkPad T400, ThinkPad T500, Elite ThinkPad X200, ThinkPad X200s, Elite ThinkPad X301, ThinkPad W500, ThinkPad W700 [143],
- Dell Latitude E4300, E4200, E5400, E5500, E6400, E6500, E6400 ATG, Precision M6400, M4400, M6400 Covet, M2400,
- HP Elite Book 2530p, 8530p, 8530w, 8730w, 6930p, Business Notebook 6530b, 6730b,
- Panasonic Toughbook F8, 19, 30, 52, 74, T8, W8.

The list above comes from a web-based source [142].

The IWL5300 was used to implement a fingerprinting ILS in a master thesis performed at the Chair of Media Technology at the Technical University of Munich [106]. The evaluation of the ILS showed that the CSI values changed when people moved around the indoor environment.

2.7.2.3 Decawave Chip - UWB CIR Measurement

The Decawave DW1000 is a fully integrated low power CMOS chip that communicates over the UWB standard [7], [131]. The Decawave board uses a two-way-ranging method and is able to perform two functions:

- Estimation of the distance between a receiver and transmitter with an accuracy of up to $10cm$. Decawave estimates a localization accuracy of around $30cm$ for a moving tag when using ToF or TDoA multilateration.
- CIR estimation.

2.7.2.4 BeSpoon - UWB on a Smartphone

The french company BeSpoon has created a localization system in which the UWB antenna is integrated into a smartphone [131]. The system provides a ranging accuracy down to $10cm$ within $880m$. In addition to ranging measurements, the UWB antenna is also used for WiFi communication.

2.7.2.5 Ubisense

The Ubisense system consists of UWB sensors and tags. The sensors contain antenna arrays and estimate the azimuth and elevation AoA of the UWB signal from each tag. The TDoA information is collected between pairs of sensors connected by a timing cable. The AoA and TDoA measurements are combined to localize the tag.

2.7.2.6 Universal Software Radio Peripheral

Universal Software Radio Peripherals (USPR) are a type of software defined radios. It is a receiver and transmitter where the signal processing is manly done by software running on general purpose processors and not implemented in circuitry. The main advantage of USPRs is that they are programmable and useful for prototyping. The USPR N210s was used in [67] as an AP and as the receiver to measure CFR fingerprints. 11 center frequencies were used with an effective bandwidth of 10MHz.

2.7.2.7 Channel Sounding Equipment

A channel sounder or network analyzer can be used with any communication standard to perform channel estimation and obtain fingerprints and time-of-flight information. A channel sounder is generally not used in a commercial ILS as it can easily cost several hundred thousand dollars.

2.7.3 Effect of Bandwidth

It would be intuitive to assume that a higher bandwidth would enable a higher localization accuracy. The work in [144] investigates the dependency of AoA estimation accuracy on bandwidth for two AoA estimation methods. The AoA estimation is applied to the LoS component in the time domain CIR. The CIR is obtained by applying the inverse which is acquired by applying the fast Fourier transform (FFT) to the simulated channel transfer function in each test bandwidth value between 20kHz and 10MHz. The single input, multiple outputs (SIMO) CIR is calculated as an impulse response containing 10 paths, including the LoS. The amplitudes are set by the K-factor of the Rician distribution with a typical urban channel maximum delay spread of $10\mu s$ [144]. AoA estimation is performed using phase interferometry and Multiple Signal Classification (MUSIC). The simulation results showed that a larger bandwidth increase the AoA estimation accuracy for both algorithms. Experimental results in [145] showed that the ranging errors also decreased for larger bandwidths. They also showed that using a directional antenna also increased the ranging accuracy.

The dependence of the LoS and multipath estimation on the signal bandwidth is analyzed using a geometry-based stochastic channel model in [146]. In a narrow band signal the LoS component becomes fused with the other multipaths denoted as diffuse multipath (DM). The DM interfering with the LoS path leads to certain multipath effects, for example amplitude fading and pulse distortion. At a very low bandwidth there is almost no pulse distortion as the complete DM interferes with the LoS delay. When the DM is approximately equal to the inverse of the root mean square (RMS) delay spread of the power delay profile, the largest amount of pulse distortion will occur. The measurements performed in [146] in the UWB frequency show an increase of signal to interference and noise ratio for the LoS path and a decrease of the ranging error at higher bandwidth. The use of diversity and multiple antennas in single-input-multiple-output (SIMO) and multiple-input-multiple-output (MIMO) systems decrease the ranging error and allow for the use of lower bandwidths for ranging. An example ML estimator performs correctly at a bandwidth of at least 500MHz.

The related work in [147] shows that using the pulse distortion and performing a whitening operation before range estimation allows the ML estimator to operate at 50MHz and using a MIMO system improves the performance of a ML estimator at 50MHz and allows the estimator to work at 10MHz.

In [67] CFR fingerprints were compared with the time reversal technique, described in Section 2.5.1.2. The environment where the fingerprints were collected was dynamic. Two fingerprints were collected with the bandwidth of both 60MHz and 11MHz. The results obtained with the two types of fingerprints showed that the localization accuracy did not necessarily increase when the bandwidth of the fingerprints was increased from 60MHz to 110MHz. The off-diagonals of the dataset did decrease with more bandwidth. The authors of [67] theorize that the additional bandwidth introduces more of the environments dynamics and thus lowers the similarities between the CFR fingerprints collected at different times at the same indoor location.

The work in [42] studied the number of multipath components extracted from wireless signals with different bandwidths and center frequencies. The Search-Subtract-Readjust Algorithm based on an ML estimator was used to obtain the multipath components. Center frequencies between 4 and 16 GHz and bandwidths between 1 and 1 GHz were used. The results showed that the choice of center frequency did not significantly effect the number of estimated multipath components. The number of multipath components increased with the bandwidth and eventually saturated. The effect of bandwidth was more prominent in NLoS environments compared to LoS environments. In general, systems with wider bandwidth offer a better time resolution [148]. This is the reason that more multipath components can be observed.

2.8 Objectively Comparing Indoor Localization Systems

When two systems are built using different technologies it is not always easy to determine which one is better or which one should be bought or implemented. The best way to objectively compare several systems is to perform measurements with both of them at the same coordinates in the same environment. If multiple ILSs are evaluated on different datasets or in different environments, factors such as a dynamic or stationary environments, the proportion of LOS and NLOS conditions at the measurement points, different fingerprint densities, hardware imperfections, interference from other electromagnetic devices, could all effect the estimated accuracies. For example, in publication A the localization error was obtained with a fingerprint map density of 5cm and in publication B with a fingerprint map density of 30cm. If the system from publication A was shown to perform better, it would be unclear whether localization system from publication A would still have a higher accuracy if the two systems were evaluated with the same fingerprint density.

As is explained in Sections 2.3, 2.5.1.1-2.5.3.3, 2.7.2, different signal parameters and receiver devices are used by different ILSs for localization. The only way to create a single measurement dataset that can be used for comparing the performance of several arbitrary localization schemes is by collecting channel measurements with a channel sounder or USRP.

However, this is not a objective comparison since imperfections or characteristics of the target devices on which the ILSs would eventually be deployed, may not appear in the channel sounder measurements and may alter the localization errors of the two systems.

Even if different devices are used to collect the signal data, ILS comparison can still be objective if the data was collected in the same environment at the same reference and query locations. Unfortunately, this is commonly not the case [149], each publication implements the proposed ILS in a unique measurement setup.

The ISO/IEC 18305:2016 International Standard [150] describes a standard methodology and metrics that can be used for evaluating indoor localization systems. The standard is analyzed in [150]. The standard uses a black-box testing approach, where the details of the ILS and available sensors are not known, and describes 14 scenarios and 5 types of buildings in which tests need to be carried out alongside 30 evaluation metrics. The testing scenarios include a walking, running user, as well as the presence of people. The specifications on the distances between and the randomness of test points are also included.

The international conference on information processing and the international conference on indoor positioning and navigation (IPIN) hold competitions known as the Microsoft and IPIN competitions where different indoor localization systems are tested in the same environment [150]. Both competitions work on to identifying the criteria and metrics needed for ILS comparison.

The following system properties generally influence the localization accuracy of an ILS:

- The more APs are used, the higher the localization accuracy will be.
- The more reference locations are stored in the fingerprint map, the higher the localization accuracy will be.
- Localization accuracies are higher when there is less noise at the receiver and less movement in the environment.
- If the system is using ranging or multipath information:
  - The smaller the number of test locations with NLoS to the APs, the higher the calculated accuracy of a ranging-based system will be.
  - Certain materials promote more multipath and reflections in the environment. An environment with more reflections will decrease the estimated accuracy of a system with RSSI ranging and increase the accuracy of a multipath-based system.

2.8.1 Ground-truth Data Collection

When an ILS is evaluated, the coordinates of the user estimated by the ILS at the query points $(\hat{x}, \hat{y}, \hat{z})$ are compared to the actual, or *ground-truth*, locations query points. One challenge in evaluating an ILS is obtaining or measuring the correct ground-truth coordinates. Manually measuring the coordinates of each query point is time consuming and would lead to fewer query points to be used for testing. Ground-truth values are collected in the following ways:

- Installation of an additional static ILS, for example an optical localization system such as the Vicon motion capturer [151].
- SLAM can be performed by the measurement platform. The measurement platform can use a laser scanner or odometry data to calculate its location.

2.8.2 Existing Datasets

There are several datasets available that can be used for testing ILSs. Example datasets are listed in Table 2.15. The table shows that most datasets contain RSSI data, some also additionally contain IMU data. Only one dataset collected in 2021 contains CSI data.

Example localization datasets.

Dataset	Year	Data	Comments
Open CSI [91]	2021	CSI (Fourier transform of the CIR), LTE signals	Software-defined radio, a **USRP B200mini**, measurements collected over a $3.5m \times 5m$ area
IoT-TD [152]	2021	**BLE** RSSI, **IMU** data	**BLE sensors**, **smart-phone**, ground truth with millimeter accuracies, optical cameras used, data collected in a $3.5m \times 3.5m$ area
Petros. at al. [153]	2020	RSSI (**ZigBee, Bluetooth, WiFi**)	Collected in meeting room and computer lab, Raspberry Pi 3 Model Bs, **Gimbal Series 10 Beacons**, and **Series 2 Xbees** with Arduino Uno
BLE beacon indoor localization dataset [154]	2018-2019	**Bluetooth** RSSI	**BLE beacon**, **smartphone**
UCI JI-IndoorLoc [155]	2018	**WiFi** RSSI	Multi-building, multi-floor, almost $110.000m^2$, multiple **Android devices**
UCI Dataset [156]	2017	**WiFi** RSSI	Collected on **smartphone**

MagPIE [157]	2017	**magnetometer, IMU, and ground truth position and orientation measurement**	Three different buildings, with **smart-phone** handheld and mounted on a wheeled robot, in environments with and without changes
Barsocchi at al. [158]	2016	**WiFi** RSSI, and **geomagnetic fingerprints**, **accelerometer**, **gyroscope, orientation data**	Collected over $185.12m^2$ using a **smartphone and a smartwatch**
KIOS [159]	2013	**WiFi** RSSI	**PDR**, a **laptop**, an **Android tablet** and two **Android smartphones** $560m^2$ typical office environment that consists of several open cubicle-style and private offices, labs, a conference room and corridors

Table 2.15: Examples of available ILS datasets.

2.8.3 Benchmarking platforms

A benchmarking platform can be used to evaluate an ILS. Such a platform usually:

- Is a device that can be moved around to gather groundtruth data.
- Either already contains a receiver or a receiver can be added to it. The platform combines the receiver measurements with the ground-truth data into a dataset.

The main reasons why there platforms are not more widely used are:

- They are often designed to evaluate a specific type of ILSs and cannot be used to test a custom receiver.
- They are mainly described in publications and are not commercially available.
- Not all benchmarking platforms include the environment where the ILS is tested. In order to objectively compare ILSs they have to be tested in the same or at least similar environments.

2.8.4 Surveys and and Evaluation Metrics in Indoor Localization

Even if the evaluation measurements for several ILSs are performed at the same reference and query points in the same environment, in order to determine quantitatively which ILS performs best, these measurements need to be converted into scalar performance metrics. The use of different performance metrics can result in incorrect ILSs being considered best. Examples performance metrics are the average error at the query points, mean squared error (MSE), root mean squared error (RMSE). The purpose of survey publications is to provide an overview and compare ILSs and ILS technologies. Most surveys do not re-implement the ILSs, but rather use the results provided by the original publications. This also means that the ILSs could be tested in very different environments. This also means that within one survey different ILSs are commonly evaluated by different performance metrics. This makes it difficult to compare the ILSs.

The performance metrics listed in indoor localization surveys are used in this thesis to show which metrics are most commonly used for evaluating ILSs. A number of randomly-selected surveys are summarized in Table 2.17 together with the performance metrics they have used. Some publications do not calculate any scalar performance metrics but, for example, present an error CDF. The right-most column of Table 2.17 lists the performance metrics that appear in each of the surveys together with the percentage of publications in that survey that have used it to evaluate their results. As publications often use more than one metric to evaluate the proposed approach, these percentages in the Table 2.17 add up to a number above 100%. The second to right column of the tables shows if the ILSs compared by the survey have been re-implemented and the environments in which some surveys re-implemented the ILSs.

When an ILS is tested, a localization error is calculated at a number of test locations. In order to quantitatively compare ILSs, a scalar performance metric, such as mean localization error or the MSE, needs to be computed from this set of error values. Table 2.16 summarizes the performance metrics used by surveys in Table 2.17 to compare ILSs. The left-most column of the table contains the names given to the performance metrics. It can be seen from Table 2.16 that the same performance metric is sometimes referred to by different names in different publications. The second column lists the definitions given to the performance metrics. It should be noted that few publications actually define the metrics they use. This can be misleading, since, as can be seen from the first two columns of Table 2.16 the same names have been used to describe different metrics. For example, Adler et al. [149] defines accuracy as the combination of trueness and precision and indicates that the mean squared error (MSE) is an example accuracy metric, [160] defines accuracy as the average Euclidean distance between the estimated and the true location. Adler et al. refers to the average localization error as Trueness. Accuracy has been used by Ferreira et al. [161] to describe the percentage of time that the ILS correctly identifies which room the user is located in. It should be noted that additional factors, such as sensor calibration and battery power requirements, processing power requirements, wearability need to also be considered when comparing ILSs [162], however they are outside the scope of this chapter. The ISO/IEC 18305:2016 international standard lists a number of ILS performance metrics and includes their definitions

[150]. In [150] the argument is made that the error mean and variance are not helpful metrics for evaluating ILSs, the RMSE and error quantiles are suggested as more helpful. When a localization error is very high, the user was not localized and the exact error value is not important. However, it will influence the mean error and MSE. A high MSE may indicate high accuracy ILS with a few outlier measurements with very high error values, or it may indicate that the error in all the measurements was at the same medium value. It is often more important to the user to know that in x percent of localization requests his localization error would be under a certain limit. The 95% and 50% quantiles are mentioned. It is also mentioned that vertical errors in the z direction may have a different importance than horizontal errors. Therefore, a standard calculation for 3D Euclidean distance errors may not be as useful. The 2D distance with floor penalty and the real distance, or the distance to be walked from the location query to the ground truth are mentioned as better metrics. The standard also discusses latency requirements. Additional metrics such as setup time, coverage, relative accuracy and tolerance of interference are also described. More details can be found in [150].

The following conclusions can be made from the overview of the localization surveys:

- There is currently no standard or default performance metric or set of metrics that can be used to characterize and compare ILS localization performance.

- The mean localization error or the mean squared localization error (MSE) are the most commonly used metrics.

- The CDF gives the full information about the error distribution, however it is not a scalar. If two ILS are described in two publications and do not publish their data it may not be possible to show both CDFs on one axis. In that case it is difficult to compare the ILSs.

- Different terms can be used to describe the same metric in different publications.

- Differently calculated performance metrics can be referred to by the same name in different publications.

- Most survey papers do not re-implement the localization systems because they do not have access either to the specialized or custom hardware or to the code that is not made publicly available. Additionally, the performance of a re-implemented ILS might differ from the original publication. Most surveys compare the localization approaches based on the evaluation results reported in the original publications. Therefore, the ILS compared by the surveys can be tested in very different environments and evaluated by different metrics.

Metric	Definition	Used in
Moments of the error distribution		
Accuracy (m)	average localization error obtained at the test points, the closeness of agreement between a measured quantity value and a true quantity value measured [138], $E[\lVert\boldsymbol{x}-\hat{\boldsymbol{x}}\rVert_2] = \frac{\sum \lvert \text{error at test point} \rvert}{\text{num test points}} = E[\text{error}]$	[163]*,[164]*, [137]*, [165]*, [138], [161], [132]*, [166]*
Average error (m),		[167], [124], [168]
Mean absolute error (MAE)(m) ,		[115]
Point accuracy (m)		[61]
Bias		[169]
Trueness	closeness between the average value of a number of measurements and an excepted reference value, i.e., mean average error (MAE)	[149]
MSE	$E[\lVert\boldsymbol{x}-\hat{\boldsymbol{x}}\rVert_2^2] = E(\text{error}^2)$	[169]
RMSE	$\sqrt{\frac{\sum(\text{error at test point})^2}{\text{num test points}}} = \sqrt{E[\text{error}^2]}$	[168]
Accuracy	combination of trueness and precision, i.e., mean squared error (MSE), root mean squared error (RMSE), $\text{MSE} = \text{MAE}^2 + \text{Varience}$ [149]	[149]
Standard deviation σ	standard deviation of the errors computed at the test points	[170],[168]
Precision	the closeness f agreement between independent measurements, i.e., varience, standard deviation	[149]
Error bounds		
Max error (m)	maximum localization error	[73]
Point accuracy min/max/mean (m)	localization error obtained at the test points min/max/mean	[61], [168], [171], [165]

Error range (m)	min and max point errors	[165], [161]
Accuracy range (m)		[165]*,[161]
Mean accuracy	upper bound of the mean error*	[166]*
Error evaluated at the end or over the length of a track		
Final error (m)	localization error computed after the completion of a track (PDR)	[168]
Position error (m)		[168], [172]
Final drift (m)		[168]
Average lateral deviation	lateral deviation of the calculated position from the trajectory*	[119],[168]
Fréchet distance	difference metric between two curves	[168], [171]
Heading error (rad)	difference between the true and measured direction of motion	[168], [172]
Absolute error (mm)	robot location error measured at a particular time instance*	[173]*,[168]
Overall estimate		
Distance accuracy (High, Medium, Low)	Qualitative assessment of the distance measurement between users	[166]
Room identification		
Relative accuracy (%)	percentage of test points where the room was identified correctly	[167]
Room localization accuracy (%)		[123]
Room accuracy (%)		[61]
Room level accuracy		[161]
Mean accuracy		[166]

Room level accuracy (true/false)	system accuracy allows to identify the room the user is located in	[165], [161]
Percent of test points with a defined localization error		
Precision (%)	the probability of successful (or unsuccessful) location estimates with given accuracy	[137], [160]
Relative accuracy (%)	number of test points where localization was performed correctly	[167]
Percentile under 5m (%)	percentage of test points at which the computed localization error was under 5m	[170], [168]
Percentiles		
99% within x(m)	upper bound of the smallest 99% of errors	[160]
95th percentile(m) 95% within x(m)	upper bound of the smallest 95% of errors	[162], [125], [168] [160], [161]
error range within 95%(m)	min and max of smallest 95% of errors	[165]
90th percentile(m) 90% within x(m) accuracy up to with 90%(m)	upper bound of the smallest 90% of errors	[168] [160], [161] [168]
90% accuracy (m)	accuracy achieved 90% of the time (unclear)	[73]
80th percentile(m) 80% within x(m)	upper bound of the smallest 80% of errors	[122] [160]
accuracy with a 50% probability(m)	average of the smallest 50% of errors*	[165]*
50% within x(m)	upper bound of the smallest 50% of errors	[160]
Third quadrille of the error(m)	the ILS provides an Euclidean error below the declared error in three cases out of four [168]	[168]

Error distribution		
PDF, CDF	full distribution of localization errors at the test points	[124], [122]
Histogram	histogram of the error distribution	[168]

Table 2.16: Metrics of ILS localization performance. * indicates the information was not specifically mentioned in the publication and had to be inferred from other system aspects.

Do to the lack of a consistent evaluation metric, the localization approaches presented in this thesis are evaluated using the mean absolute localization error. The histogram of the probability distribution of the error values and localization error scatter plots are also presented since they contain the complete information about the algorithms localization performance. The mean absolute localization error is calculated as follows:

$$e_{avr} = E[\left|\left|\boldsymbol{X} - \hat{\boldsymbol{X}}\right|\right|] = E[|e|]. \tag{2.6}$$

Although it has it drawbacks, the average absolute error represents the localization performance as a scalar value. It is also one of the most commonly used evaluation metric in the literature, as can be seen from Tables 2.17 and 2.16.

Surveys on Indoor Localization Systems						
Authors	Published	Year	Algorithms	Technologies	Re-implemen	Evaluation metrics
Liu et al. [160]	*IEEE Transactions on Systems, Man, and Cybernetics*	2007	20	WiFi, UWB, cellular, RFID, Bluetooth	No	accuracy (all), precision(upper bound of a set % of min errors) (all), complexity, scalability, robustness, cost (all)

Gu et al. [165]	*IEEE Communications Surveys & Tutorials*	2009	17	infrared, ultra-sound, RFID, WiFi (triangulation, fingerprinting), Bluetooth, sensor networks, UWB, magnetic, vision-based	No	cost, robustness, complexity, limitations, room level accuracy (12%), error rage in the lower 95% (12%), max error(12%), average accuracy*(29%), error range (17.6%), average of the lower 50% of errors* (5.8%), upper bound of the lower 90% of errors (5.8%), accuracy cannot be guaranteed (5.8%)
Mautz et al. [163]	*International Conference on Indoor Positioning and Indoor Navigation (IPIN)*	2011	26	optical positioning systems	No	accuracy (order of magnitude,cm,mm,dm, %), coverage, camera properties (all)
Harle [162]	*IEEE Communications Surveys & Tutorials*	2013	23 + 6	step detection + particle filter with PDR/WiFi assisted PDR	No	mean localization error (33%), 95th percentile (33%)
Subbu et al. [167]	*IEEE Wireless Communications*	2014	13	cellular, WiFi, microphone, camera magnetometer+PDR	No	relative accuracy (65%), average error(65%), both (23%)

El-Sayed et al. [169]	*International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*	2014	3	millimeter-wave wireless systems + RSS, TOA, AOA	Monte-Carlo simulation	MSE (all), Bias (all)
Mainetti et al. [174]	*International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*	2014	6	vision, infrared, ultrasound, WiFi, RFID, Bluetooth	No	Accuracy, coverage, cost, complexity (value ranges compared for each technology)
Adler et al. [149]	*International Conference on Indoor Positioning and Indoor Navigation (IPIN)*	2015	138	fingerprinting + TOF + sound + PDR + others	No	characterization of evaluation techniques and metrics: trueness (54%), accuracy (38%), error histogram (16%), trueness and precision (30%)
Van Haute et al. [61]	*International Journal of Health Geographics*	2016	7	RSSI fingerprinting + ranging(WiFi, ZigBee, BLE) + ZigBee ToA with particle filter	real-life functioning hospital environment	room level accuracy, latency, installation time/cots, energy consumption (all)

Basri et al. [137]	*International Conference on Multimedia Computing and Systems (ICMCS)*	2016	5	Infrared, WiFi, ultrasound, RFID, Bluetooth, multiple localization schemes	No	rough accuracy, coverage, power consumption, cost (all)
Khudhair at al. [138]	*Indonesian Journal of Electrical Engineering and Computer Science*	2016	*	TOA, TDOA, AOA, RSS, computer vision, infrared, RFID, Cellular, UWB, WLAN, Bluetooth, ZigBee, FM, Optical	No	accuracy, scalability, complexity, cost (all)
He at al. [166]	*IEEE Communications Surveys and Tutorials*	2016	50	WiFi, Bluetooth Low Energy, RSSI fingerprinting + improvements + PDR	No	error upper bound (24%), overall assessment (10%) , mean accuracy (6%), room identification accuracy (2%)
Potortì et al. [168]	*Sensors*	2017	24	RSSI fingerprinting + hybrid IMU + UWB + magnetic + ultrasound	No	average error (71%), final error (13%), RMSE (8%), max/min/median error(17%), error percentile (13%) PDF/CDF/ histogram (17%), trajectory (33%) and more

Yassin et al. [164]	*IEEE Surveys and Tutorials*	2017	14	WIFI + RSSI, TOA, TDOA, triangulation, fingerprinting	No	range (all), accuracy (all)
Ferreira et al. [161]	*IEEE Surveys and Tutorials*	2017	34	WiFi, UWB, Ultrasound, RF, Bluetooth, IMU, RFID, magnetic, infrared + Fingerprinting, TDOA, ranging kalman filter, PDR, SLAM	No	room level (17.6%), average error (20%), upper bound of all errors (23.5%), upper bound of the smallest % errors (8.8%), error range (3%)
Zafari et al. [132]	*Computing Research Repository (CoRR)*	2017	41	WiFi, UWB, Acoustic, RFID, Bluetooth, Ultrasound, Optical, SigFox, LoRA, TOA, TOF, AoA, TDoA, RSSI, MP	No	Accuracy, Cost, Availability, Energy Efficiency, Reception Range, Latency, Scalability (most)

Table 2.17: Surveys on indoor localization systems

2.9 Commercially available ILS

Most localization system concepts detailed in the section above have been published in academic publications and are not commercially available at this moment.

While a commercially available ILS can be directly installed in a building that corresponds to the system specifications, the following challenges need to be overcome when implementing an ILS described in an academic paper:

- The ILS in a publication may have been tested in only one environment. The performance specifications presented in the publication may not apply to larger environments or environments with moving objects.

- The ILS in a publication may have been tested on a very small area, the effort to collect the reference fingerprints over a larger area with the same density as was used in the publication may be very large.
- Some implementation details or parameter values of the ILS may not be mentioned in the publication.

Table 2.18 [175], [176] summarizes the ILSs that can be purchased and directly used. The information from the company websites listed in Table 2.18 was retrieved in January 2020. Most ILSs listed Table 2.18 rely on the reception of electromagnetic signals, however several of the ILSs also use additional technologies, such as geomagnetic fingerprinting, not discussed in this thesis. It should be noted that some of the data in the table was inferred from the company websites. * indicates the information was not specifically mentioned but inferred from other system aspects.

Company	System	Localization scheme	Specs	System accuracy
Steerpath [177] (Finland)	Smartphone based	WiFi or Bluetooth based*. Ranging-based* - no calibration, may use building models. During localization the user's smartphone can be located in a pocket or a bag. This means the system is likely ToF and not power -based*. First position estimate within 2 seconds - if IMU is used it's not the main data source.	Uses custom beacons that need to be installed.	2-5m in an office, hospital or residential building, 5-10 m in large open spaces such as airports and conference centers, 95% within 10m
LocusLabs [178] (Poland)	Smartphone-based localization.	Fingerprinting* as an RF map is measured. Data from other sensors may be used.	An RF map of the building created during calibration.	Designed for a human user and asset tricking. several m accuracy*.

Terabee [179] (Switzerland)	Anchors and a tag	Two-way ranging	AP size 5x5x6.5cm, receiver device size 14.5x9x3.5cm, update rate 30Hz	-
Cloudleaf [180] (California)	IoT system with a multitude of access points and sensors. BLE*	Proximity-based*, Fingerprinting*, DSP Filtering, Chennelization, Motion Sensing	gateway 9.9x11,8x2.5cm, zone sensor 4.2x7x0.9cm, sensor range 1-13m to 50m	up to 2m
SmartPoint [181](-)	Mapping system	Mapping system that provides locations on a more efficient way. Provides localization algorithms	-	-
AiristaFlow [182] (Finland)	Asset tracking, custom tags designed	RFID over Wifi	uses existing WiFi infrastructure, RFID, BLE and GPS	room-level
Bluvision [183] (Texas)	IoT system with localization using Bluetooth and WiFi.	Data from multiple sensors analyzed.	Active tags and BLE beacons used	2m
Connexient [184] (New York)	smartphone-based indoor navigation.	Sensor fusion	-	good enough for navigation, several m*

Inpixon [185] (California)	Smart-phone based navigation	sensor fusion, WiFi signals used, Bluetooth, Cellular and active RFID can be used to increase accuracy.	existing or WiFi routers can be used. A smart phone can be used or a company developed sensor.	1 - 15m depending on the sensors used
Quuppa [186] (Finland)	Bluetooth Low Energy system with tags and APs	Angle-of-arrival	access points 20.2 x 20.2 x 4cm or 40cm x 30.8cm x 5.9cm, tag 3,9cm x 3.9cm x 9cm	0.5m
Kontakt.io [187] (Poland)	Bluetooth Low Energy system with beacons and tags	localization scheme requires little to no calibration. Some tags include an IMU	tag 4.3x4.3x0.8cm, card tags 8.6x5.5x0.45cm beacons under 11.5x6.5x4cm, signal range under 70m	-
Kinexon [188] (Germany)	UWB, RFID	ToF	-	10cm - 50 cm
Infsoft [133] (Germany)	system with UWB and BLE beacons and tags	AoA, ToF	tag 11x3.6x3.6cm	10-30cm
Prozyx [189] (Belgium)	UWB system with APs and tags	TDoA	-	10-30cm
Proxicon.io [190] (UK)	UWB and Bluetooth tags and anchors	-	-	10cm and higher

nanotron [191] (Germany)	UWB tags and anchors	ToA, TDoA, ToF	tag4x2.4x0.35cm, relatively small anchors	10-30cm, 1m for LE solutions
iViu Technologies [192] (California)	smartphone-based localization, extra WiFi sensors installed	-	installed sensors size of a deck of playing cards	3 degrees - 150 degrees
Ubisense [193] (UK)	UWB sensors and tags	TDoA, AoA,determining the direct path	-	-
BlueBotics [194] (Switzerland)	use of indoor features, laser scanner	-	for vehicles up to 5.0 m/s, weight 0.6 - 1.1kg excluding laser scanner, min size 134x26x160mm	accuracy ± 1 cm and ± 1°, localization rate up to 20 Hz
CloundNav [195] (California)	Sensor fusion. Software solution for mobile devices	Fuse GPS, WiFi, Bluetooth with accelerometers, gyroscopes magnetometers and pressure sensors in mobile devices	Uses the sensors of a mobile device	-
IndoorAtlas [196] (Finnland)	Smartphone-based geomagnetic indoor positioning with other sensors.	Data fusion,PDR, geomagnetic fingerprinting, Bluetooth beacons, WiFI triangulation, Barometric pressure.	Flexible to the signal sources present in the environment	1-2m

GiPStech [197] (Italy)	system that can be run on any smart device and uses that devices sensors	combination of geomagnetic fingerprinting radio frequency algorithms and inertial sensors	would run on systems with different sensor configurations	1m
Accerion [198] (Netherlands)	scanning warehouse and factory floor	-	no additional infrastructure required, suitable for dynamic environments	mm-level
Locatable [199] (Ireland)	system providing own location devices	-	-	5cm

Table 2.18: Commercially available indoor localization systems. Information marked with * was inferred (the information is not mentioned on the websites).

Other companies providing smartphone-based localization include Navenio [200], Insiteo [201], Mapxus [202], Combain Mobile[203], Sensewhere [204], Pointr [205], SIRL [206], situm [207], Nextome [208], Lighthouse Signal Systems [209], Arara [210], Navigine [211], Indoors.rs [212]. Additional companies include Witrac [213], Intranav [214], Sewio [215], Localino [216], Ssafectory [217].

The majority of companies listed in Table 2.18 can be classified into three categories:

- High-cost high-accuracy systems employing a Lidar or laser scanner that are used in the industry and produce cm-level accuracy.
- Industrial ILS that employ custom UWB tags (receivers) and anchors (transmitters) possibly with sensor fusion. These systems produce an accuracy of 10-30cm but require a LoS between the tag and the anchors.
- Smart-phone based system that either use pre-installed WiFi routers or custom WiFi and Bluetooth anchors together with the IMU and other sensors inside the smartphone and have an accuracy of 1-5m.

An environment with many moving objects and people and with no LoS between the receivers and the transmitters remains a challenge.

2.10 Chapter Summary

This chapter provided an overview of existing indoor localization approaches that are based on electromagnetic signal transmission. There are two types of RF-based localization approaches. The first type of localization schemes uses the signals from the access points (APs) to calculate the distances or angles between the APs and the receiver or differences between the distances from the receiver to each of the APs. Triangulation and multilateration algorithms are used to localize the user. While precise, these approaches require a line-of-sight (LoS) between the receiver and the APs. This is not always the case in indoor environment.

Fingerprinting-based localization approaches perform a series of signal measurement during calibration, store the results in a fingerprint map and compare the user's measurement to that map to localize him or her. Fingerprinting-based algorithms perform well in non-line-of-sight (NLoS) conditions. Deep learning and machine learning approaches are used to improve the localization accuracy, reduce the effect of noise and the number of calibration measurements required by a fingerprinting system. The accuracy of fingerprinting systems that use received signal strength (RSSI) is several meters. Fingerprinting systems using channel state information, the channel impulse and the channel frequency response and UWB signals have localization errors as low as several centimeters. However, if any changes occur in the environment after the fingerprint map was created or if the environment contains moving objects, the accuracy of the localization schemes decreases significantly.

The motion data from the inertial measurement unit is often used in the literature to improve the accuracy of localization schemes. IMU data is mostly combined with RSSI measurements as an IMU is integrated into a smart phone and RSSI measurements are performed by a smartphone by default. Such systems have accuracies from one to several meters.

The goal of this thesis is to develop a fingerprinting-based localization scheme that can maintain a high localization accuracy in a dynamic environment. The target ILS should not require motion data and should be able to localize a stationary receiver. This is done because localization of static objects is often required in industrial scenarios and because IMU data can always be used to further improve the accuracy of a high-accuracy localization scheme. The proposed indoor localization system (ILS) is detailed in Chapter 3.

There are many different metrics by which the performance of an ILS is evaluated in the literature. Although there is not standard metric, the average absolute error is the most common and will be used in the remainder of this thesis.

Chapter 3

Robust Multipath-based Fingerprinting

As was shown in Chapter 2, most multilateration-based indoor localization systems are very sensitive to the presence of NLoS conditions. At the same time, fingerprinting-based localization systems do not produce the desirable localization precision if there is movement of objects and changes in the indoor environment. However, ILSs are most needed in dynamic environments with NLOS conditions. Shopping malls contain complex structures and moving people, boxes and transport vehicles are constantly moved in warehouses and factories, robotic arms perform tasks in factories. What is most commonly done to mitigate this problem is to use RF fingerprint or ranging measurements together with a Kalman filter and an IMU. However, since an IMU and a Kalman filter can be used to improve the accuracy of any ILS, the focus of this thesis is to develop a base fingerprinting system that is as precise and robust as possible. In addition, the ILS developed in this thesis should to able to localize a stationary object, which is not possible if it relies on a Kalman filer or IMU data.

The fingerprint structure and multipath matching scheme described in this thesis were originally introduced in [1], [3], [6]. Fingerprinting-based systems perform better in NLoS conditions than ranging schemes. RSSI and CSI fingerprint vectors are power and frequency representations of the channel. They aggregate the information about all of the surrounding objects and materials into each vector entry. If an object in the environment moves, it can effect all of the RSSI, CSI and CFR vector entries. The changes introduced into a fingerprint by a dynamic environment can not be isolated from the features that correspond to the unaltered portion of the environment. In contrast, as illustrated in Fig. 3.3, signal propagation paths can be blocked and created individually as objects move in the indoor environment. As illustrated by Fig. 3.3, even if some of the multipath components are affected by an obstacle or a change in the environment, the rest can still be used to localize the receiver. The propagation path lengths shown in Fig. 3.3 cannot fluctuate, as they are tied to physical distances. They can only appear in and disappear from a fingerprint vector. This makes the multipath delay profile (MDP), described in Section 3.3, a more robust fingerprint than the RSSI, CIR, CFR and CSI.

The system requirements are listed in Section 3.1. The assumed system model is detailed in Section 3.2. The proposed multipath delay profile fingerprint structure is described in

Section 3.3. The multipath component analysis (MCA) localization algorithm is detailed in Section 3.4. The ray-tracing simulation setup and the obtained evaluation results are detailed in Section 3.5. The multipath estimation algorithm and hardware prototype are described in Section 3.6. Section 3.7 contains the conclusion of the chapter.

3.1 Goals and System Requirements

The main goal of this thesis is to develop a high precision ILS that can localize a user in an NLoS environment where furniture may change over time and which contains moving objects and people.

ILS performance requirements:

- The location of the user should be estimated as accurately as possible.
- The localization accuracy should not be affected by NLoS conditions, changes in the environment and the movement of objects.
- The number of APs should be as low as possible.
- The volume of data exchanged between the user and the ILS server should be as low as possible. The ILS server should only send the necessary data to the user and not the complete fingerprint map.
- The delay between the time when the user performs a measurement and the time when he obtains his location from the ILS server should be as small as possible. The system should be able to run in real-time. This means that the data transfer, computations and the data-base queries should be as efficient as possible.
- When possible, the majority of calculations should be performed by the server.
- The standard communication protocol between the user and the server should not need to be altered. No additional encryption should be implemented.
- The calibration time and effort should be kept as low as possible. The number of labeled manually collected reference fingerprints should be as small as possible.

3.2 System Model

The indoor localization system is setup in the following way:

- The user communicates with multiple APs set up in the indoor environment.
- The user estimates the electromagnetic channel and extracts the multipath propagation delays from the received channel. Alternatively, the user sends the received signal as is to the ILS server and the ILS server performs the multipath estimation.

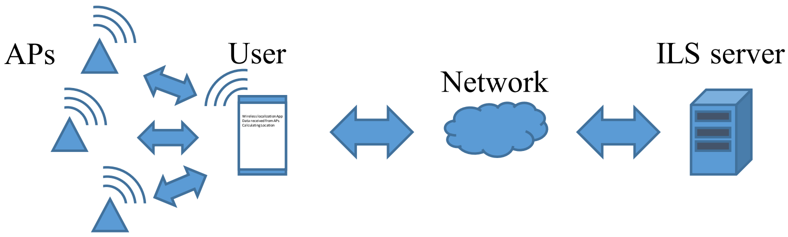


Figure 3.1: Assumed ILS system model. The user measures the signal from the APs and communicates with the ILS server. The communication with the ILS server can occur though the APs or a separate device. Image adapted from [1] ©2021 IEEE.

- The user communicates with and sends the measured signal or multipath delays to the ILS server. The ILS server calculates the user location and sends it to the user.

The communication between the user, APs and ILS server is illustrated in Fig. 3.1.

Signal propagation between the user and the APs is modeled in the following way [1], [3]. The electromagnetic signal propagates with the speed of light, therefore, the distance a signal travels is linearly proportional to the delay at the receiver. When the user and the APs are located indoors, the signal transmitted by the AP is reflected multiple times from the walls, floors, ceilings and objects before reaching the receiver. This is illustrated in Fig.3.3. As a result, the received signal $x(t)$ is a sum of multiple delayed and attenuated copies of the original transmitted signal $s(t)$.

$$x(t) = \sum_{k}^{K} a_k s(t - \tau_k) + n(t), \tag{3.1}$$

where a_k is the attenuation undergone by the signal along the k-th propagation path from the AP to the receiver and τ_k is the delay. The term $n(t)$ represents noise added to the signal at the receiver [18]. K is the number of signal propagation paths between the receiver and the transmitter. With $\tau_k = d_k/c$, we obtain

$$x(t) = \sum_{k}^{K} a_k s(t - d_k/c) + z(t). \tag{3.2}$$

The channel impulse response is then modeled as:

$$x(t) = h(t) * s(t), \tag{3.3}$$

$$h(t) = \sum_{k=1}^{K} a_k \delta(d_k/c), \tag{3.4}$$

where $\delta(t)$ is the unit impulse response. d_k is the length of the k-th propagation path traveled by the signal from the AP to the user and c is the speed of light [8]. In this thesis, the term *multipath component* (MPC) is used interchangeably to describe the delay d_k/c or length d_k of a *propagation path* between an AP and the receiver. Multipath components have been previously used for localization in [218] and for simultaneous localization and mapping (SLAM) in [43], [219].

3.3 Multipath Delay Profile (MDP)

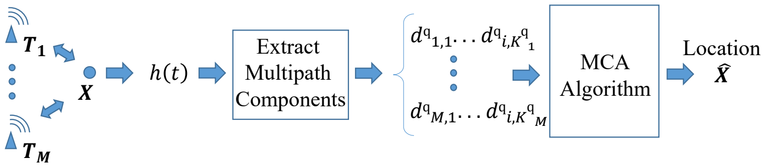

T_1
X
T_M
$h(t)$
Extract Multipath Components
$d^q{}_{1,1}\dots d^q{}_{i,K^q{}_1}$
$d^q{}_{M,1}\dots d^q{}_{i,K^q{}_M}$
MCA Algorithm
Location
$\hat{X}$

Figure 3.2: Proposed localization scheme. The multipath delay profile (MDP) is extracted from the signals received by the user from the APs. Next, the ILS server uses the multipath matching (MCA) algorithm to calculate the user's location. Image adapted from [1] ©2021 IEEE.

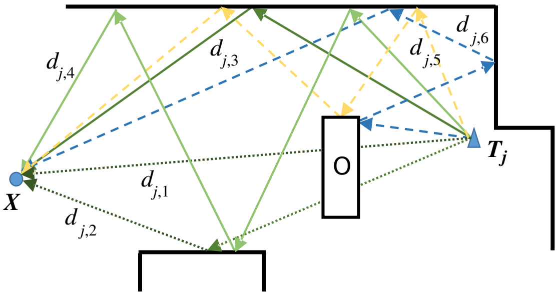


Figure 3.3: Example MDP. If the obstacle, marked with O, is not present in the indoor environment, the MDP calculated at the receiver location $\boldsymbol{X}$ is $\{[d_{j,1}, ..., d_{j,4}]\}$. If the obstacle is present, it alters the MDP. Propagation paths $d_{j,1}$ (yellow) and $d_{j,2}$ (blue) are blocked, and new propagation paths $d_{j,5}$ and $d_{j,6}$ are created. The MDP in the case when the obstacle is present becomes $\{[d_3, ..., d_6]\}$. The image is adapted from [1] ©2021 IEEE.

The multipath delay profile (MDP) fingerprint structure, originally proposed in [3] is used in this thesis to enable robust fingerprinting in an environment that contains moving objects.

The MDP is a set of vectors that contain the multipath propagation distances or multipath delays between a query point and the transmitters [3]. Given a system with M APs, the *reference* MDP fingerprints

$$\mathbf{D}_i = \{\boldsymbol{d}_{i,1}, .., \boldsymbol{d}_{i,M}\} \tag{3.5}$$

are measured in the off-line phase at N reference locations $\boldsymbol{X}_i$, with $i = 1...N$. The multipath propagation distances $d_{ij,k}$ from transmitter $\boldsymbol{T}_j$ to the reference point $\boldsymbol{X}_i$ are stored in the vector

$$\boldsymbol{d}_{ij} = [d_{ij,1}, ..., d_{ij,K_{ij}}] \in \mathbf{D}_i, \tag{3.6}$$

with j taking the values $1...M$, respectively. The reference MDPs $\mathbf{D}_i$ are in turn stored in the fingerprint map

$$\mathbf{MAP} = \{(\boldsymbol{X}_1, \mathbf{D}_1), ..., (\boldsymbol{X}_N, \mathbf{D}_N)\}. \tag{3.7}$$

The *query* MDP

$$\mathbf{D}^q = \{\boldsymbol{d}_1^q, ..., \boldsymbol{d}_M^q\} \tag{3.8}$$

is measured in the on-line phase of the localization algorithm at the user's location $\boldsymbol{X}$, with

$$\boldsymbol{d}_j^{\boldsymbol{q}} = [d_1^q, ..., d_{K^q}^q]. \tag{3.9}$$

When a fingerprint contains all of the extracted multipath components it is referred to it as a *full* fingerprint. A fingerprint containing only a subset of the multipath components measured at a location is referred to as a *partial* fingerprint. The original localization presented in this chapter uses full fingerprints, while partial fingerprints are used by the privacy protection scheme in Chapter 4. Two example MDPs $\{[d_{j,1}^q, .., d_{j,4}^q]\}$ and $\{[d_{j,3}^q, ..., d_{j,6}^q]\}$ are shown in Fig. 3.3. The first MDP is produced by the geometry without the obstacle, and the second MDP corresponds to the geometry with the obstacle O.

3.4 Proposed Approach - Multipath Component Analysis

Algorithm 1: MCA Localization Algorithm [3]

```
Data: Query multipath delay profile $\mathbf{D}^q = \{\boldsymbol{d}_j^q\}, j = 1..M$
      transmitters $\{\boldsymbol{T}_j\}, j = 1..M$
      fingerprint map $\mathbf{MAP} = \{(\boldsymbol{X}_i, \mathbf{D}_i)\}, i = 1..N$
Parameters: Similarity threshold $\varepsilon$
Result: $\hat{\boldsymbol{X}}$ - Estimated location of the receiver
for $\forall \boldsymbol{X}_i$ in the fingerprint map do
    for $\forall$ transmitter $\boldsymbol{T}_j$ do
        $\boldsymbol{d}_{ij} \leftarrow \mathbf{D}_i(j)$
        for $\forall d_{j,k}^q \in \boldsymbol{d}_j^q$ do
            $d^* \leftarrow \text{argmin}_{d^q \in \boldsymbol{d}_{ij}} |d_{j,k}^q - d^*|$
            if $|d_{j,k}^q - d^*| < \varepsilon$ then
                $\gamma(\mathbf{D}^q, \mathbf{D}_i|\boldsymbol{T}_j) \leftarrow +(\varepsilon - |d_{j,k}^q - d^*|)^2$
$i \leftarrow \text{argmax}_i \sum_j \gamma(\mathbf{D}^q, \mathbf{D}_i|\boldsymbol{T}_j)$
$\hat{\boldsymbol{X}} \leftarrow \boldsymbol{X}_i$
```

In order to determine which reference fingerprint is most similar to the query, a similarity metric $\gamma(\mathbf{D}^q, \mathbf{D}_i)$ is calculated for each reference fingerprint $\mathbf{D}_i$. The location of the user is assumed to be the same as the reference point that corresponds to the reference fingerprint for which the largest similarity metric is calculated. The multipath component analysis (MCA) algorithm detailed in [3] calculates the similarity metric between two fingerprints in a way that excludes the changes that occurred in the environment after the creation of the fingerprint map. Only the multipath components that correspond to the unchanged geometry are used to calculate the similarity metric between the reference and query fingerprints. Algorithm 1 shows the pseudo-code of the MCA algorithm. The MCA algorithm compares individual query multipath delays to the reference fingerprints. When two multipath components are matched, their difference is included into the similarity metric. A multipath component $\boldsymbol{d}_j^q$ from the query vector is matched to the multipath component in the reference fingerprint $\boldsymbol{d}_{ij}$ that is closest to it in value and the difference between the two multipath components is smaller than the threshold ε. That is $d_{j,k}^q$ and d^* are considered matched if the difference $|d_{j,k}^q - d^*|$ is minimized by $d^* \in \boldsymbol{d}_{ij}$ and $|d_{j,k}^q - d^*| < \varepsilon$. Because of this, propagation paths that disappeared from or were added to the query MDP due to an obstacle of change in the environment that occurred after the fingerprint map was created, are automatically excluded from the similarity metric. The similarity metric is, therefore, a measure of how many matches were found and how close they were. For each match, the index k is added to the set $\mathbf{Q}_{ij}$ and a term $(\varepsilon - |d_{j,k} - d^*|)^2$ is added to the similarity metric as follows

$$\gamma(\mathbf{D}^q, \mathbf{D}_i | \boldsymbol{T}_j) = \sum_{k \in \mathbf{Q}_{ij}} (\varepsilon - |d_{j,k}^q - d^*|)^2, \tag{3.10}$$

$$\mathbf{Q}_{ij} = \{k | \exists d^* \in \boldsymbol{d}_{ij} : |d_{j,k}^q - d^*| \to min \cap |d_{j,k}^q - d^*| < \varepsilon\}. \tag{3.11}$$

The similarity metric $\gamma(\mathbf{D}^q, \mathbf{D}_i)$ is calculated individually for each transmitter and then summed up to one value as

$$\gamma(\mathbf{D}^q, \mathbf{D}_i) = \sum_{j=1}^{M} \gamma(\mathbf{D}^q, \mathbf{D}_i | \boldsymbol{T}_j). \tag{3.12}$$

The simulation and measurements used to evaluate the proposed approach are described in Sections 3.5 and 3.6. The validation of the MCA algorithm is provided in this chapter. The additional algorithms presented in this thesis are validated in the following chapters.

3.5 Ray-tracing Simulation Setup

A simulation tool was developed to extract the multipath distances from a 3D geometry. The inputs to the simulation are a set of planes representing the walls, ceilings and objects in the room, the locations of the transmitters and the sample points. Ray-tracing is used to precisely extract the MDPs for a given 3D indoor geometry. Ray-tracing [3] is a simulation approach often used in computer graphics. Although originally designed for modeling the propagation of light rays as they are reflected of obstacles and illuminate objects, ray tracing can be also used to model the propagation of electromagnetic signals [220].

The following approach is common in computer graphics. Multiple rays are emitted or 'cast' from a source along all possible directions into the environment. Ray-casting with omnidirectional antenna or source is illustrated in Fig. 3.4. The path of a ray is then followed as it is reflected from multiple surfaces and reaches the target point. The scheme collects the rays that arrive at a target point, observes which paths they followed and which objects they are reflected from. While this approach is effective at calculating lighting conditions in computer graphics, it is less practical for estimating MDPs. The density of the rays decreases as they propagate away from the transmitter. In order for all multipath components to be detected at a target location, a very large number of rays needs to be cast from the source

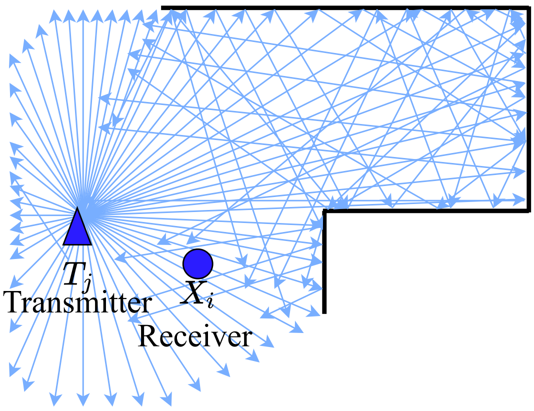


Figure 3.4: Ray casting example. Rays are 'cast' from the transmitter location and are reflected off the walls. Interpolation is needed in order to obtain the multipath profile at point $\boldsymbol{X_i}$.

and a very large radius is needed to catch the rays. If rays are caught with a large radius their lengths will not be equal to the exact multipath components.

The principle illustrated in Fig. 3.5 and the concept of virtual transmitters (VTs) described in Section 6.1 is used instead in the simulation to calculate the exact multipath component values. The distance between a transmitter $\boldsymbol{T}_j$ and a receiver located at point $\boldsymbol{X}_i$ in Fig.3.5 along the reflected path is equal to the distance between $\boldsymbol{X}_i$ and the reflection of $\boldsymbol{T}_j$ $\boldsymbol{T}_j^{''}$. $\boldsymbol{T}_j^{''}$ is the virtual transmitter (VT) corresponding to $\boldsymbol{T}_j$ and the reflection order. The VTs corresponding to a transmitter and a geometry can be calculated recursively. The a set of first order VTs can be calculated as the reflections of the transmitter in each of the planes in the geometry. In the next step for each plane G_i in the geometry a set of new planes (polygons) can be obtained by reflecting all of the other planes (polygons) from G_i. The first order VT corresponding to the plane (polygon) G_i is then reflected in each one of the new second degree planes to obtain second degree VTs. The process is further repeated for each of the second degree VTs and corresponding second degree planes. The computation is stopped when the VTs have been obtained up to and including the desired reflection order. This algorithm is illustrated in Fig. 3.6.

A more computationally efficient ray tracing model was developed in a master thesis at the Chair of Media Technology at the Technical University of Munich [220]. Ray propagation is calculated for each room in the indoor environment using VTs and the results are then combined to create a full map of the building. This model, however, is not used in this thesis. Integrating it into the simulation can be part of the future work.

It should be noted that we are unable to directly compare the proposed approach with RSI and CSI fingerprint algorithms described in Section 2.5, as unlike most existing approaches, the proposed approach is designed for MDP and not RSSI fingerprints. A ray tracing setup is needed to exactly extract multipath components. However, it cannot be used to generate a realistic RSSI distribution due to the complexity of fading in a real scenario.

The geometry used in the simulation is shown in Fig. 3.7. Rectangles are used to represent the walls, ceilings and objects in the room. The environment contains four APs shown

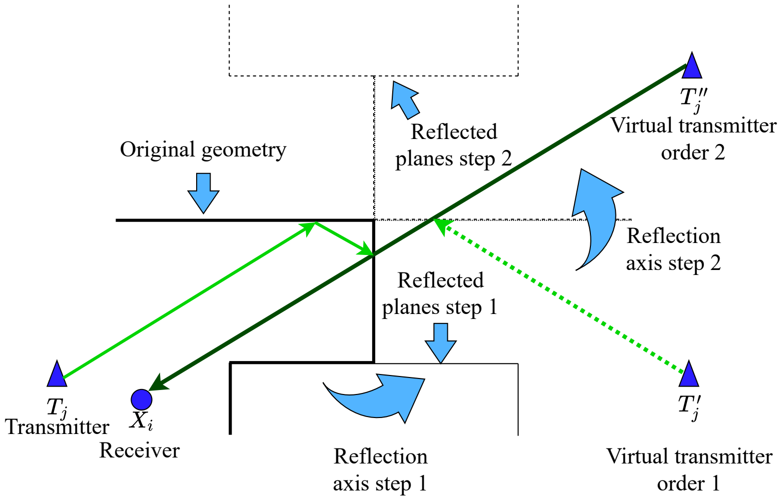


Figure 3.5: Calculation of a multipath component with two reflections.

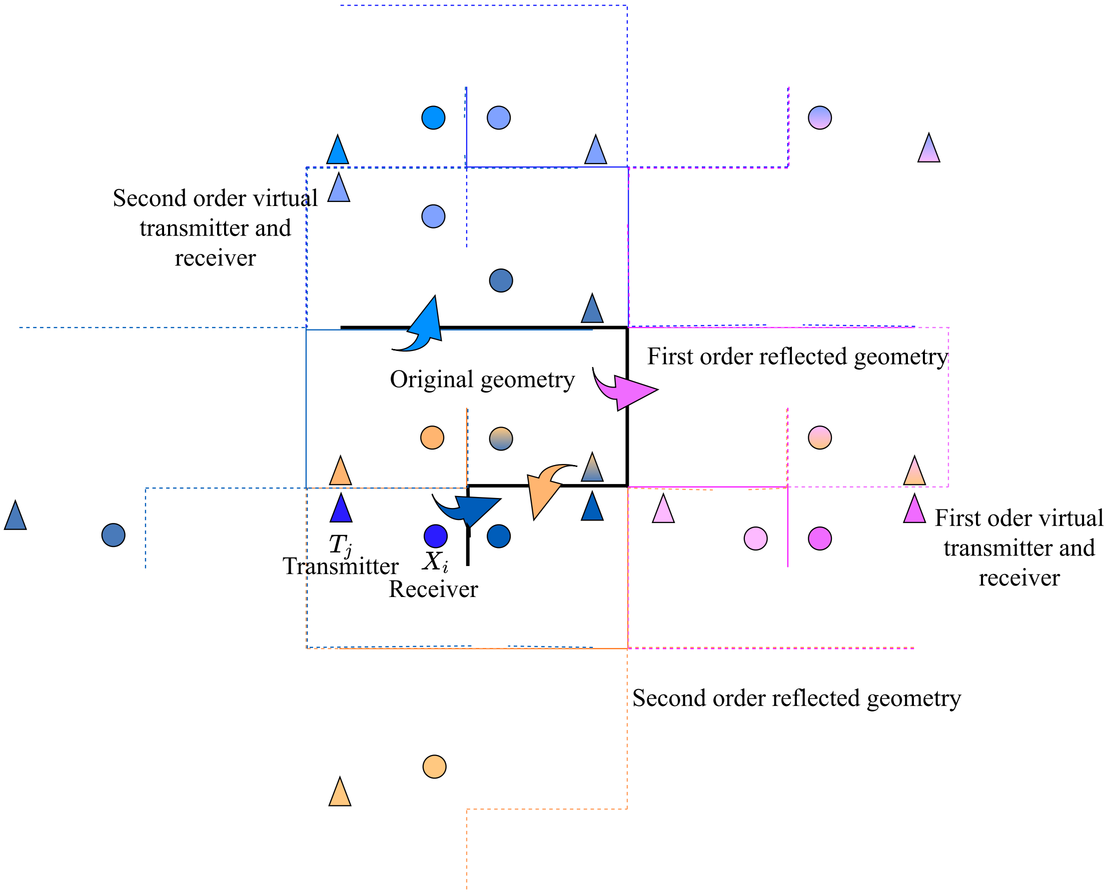

Second order virtual transmitter and receiver
Original geometry
First order reflected geometry
T_j
Transmitter
X_i
Receiver
First oder virtual transmitter and receiver
Second order reflected geometry

Figure 3.6: Recursive calculation of virtual transmitters and receivers. Solid black lines represent the original geometry, solid colored lines represent the geometry that was reflected once, dotted lines represent the geometry that was reflected twice.

```
Algorithm 2: Ray Tracing Simulation Algorithm
  Data: Geometry, Transmitters $\boldsymbol{T}_j$, Query and Refernce points $\{\boldsymbol{X}_i\}$
  Parameters: $max\_num\_reflections$
  Result: Set of VTs $\{\mathbf{VTs}\}$, MDPs at reference and query points
  $\mathbf{VTs} \leftarrow \{\}$,  reflections $\leftarrow 0$
  function $\mathbf{VTs}$ = Calculate_VTs($reflections$,geometry, $\{\boldsymbol{T}_j\}$):
  if $reflections == max\_num\_reflections$ then
      return $\{\}$
  for $\forall$ planes $\in$ geometry do
      new_vts $\leftarrow$ reflect transmitters in the plane
      new_geometry $\leftarrow$ reflect all other planes in the plane
      $\mathbf{VTs} \leftarrow$ Calcualte_VTs($reflections$+1,new_geometry,new_vts)
      return {new_vts, $\mathbf{VTs}$}
  Reference and Query MDPs $\leftarrow$ euclidean distances($\{\boldsymbol{X}_i\}$, $\{\mathbf{VTs}\}$)
```

in Fig. 3.7. The MDPs were calculated at the reference and query points. In that figure, both the query and reference points have been downsampled before being plotted. The geometry colored in green and the floor and ceiling were used to calculate the reference fingerprints. Before the query fingerprints were calculated, the blue color columns were added to the geometry. The MCA similarity threshold ε was set to $1m$. The same simulation environment has been used in [2], [1], [4], [5].

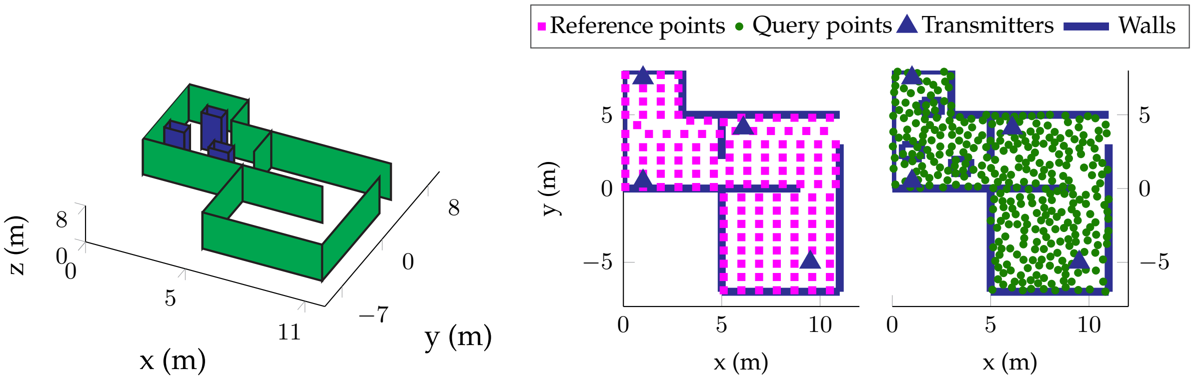


Figure 3.7: Simulation setup. Left: planes representing the indoor geometry (floor and ceiling not displayed). The geometry used for generating the fingerprint map is colored in green. The three columns, colored in blue, were added to the geometry before the query fingerprints were generated. Right: Transmitter locations and the query/reference points. Figure adapted from [4] ©2018 IEEE.

3.5.1 MCA Simulation Results

Figure 3.8 compares the performance of trilateration and the MCA algorithm in a dynamic environment. NLoS conditions are observed at many of the query points. It can be seen that the presence of walls and the obstacles inside the building creates so much NLoS that the trilateration algorithm is unable to localize the user at most locations. The reference data for

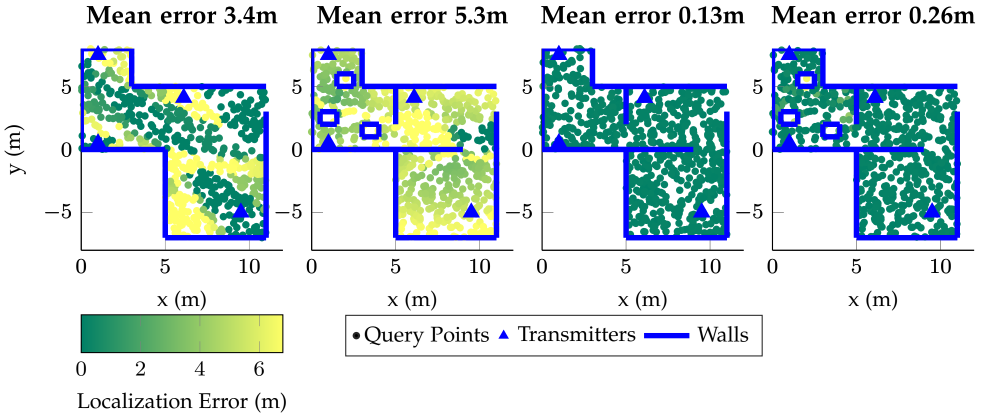


Figure 3.8: Left: Localization error of trilateration with walls between the rooms removed. Center Left: Trilateration with obstacles. Center Right: MCA algorithm, reference and query data generated without obstacles. Right: MCA algorithm, obstacles added when the query data was generated. No noise is included in the simulation. Figure adapted from [1] ©2021 IEEE.

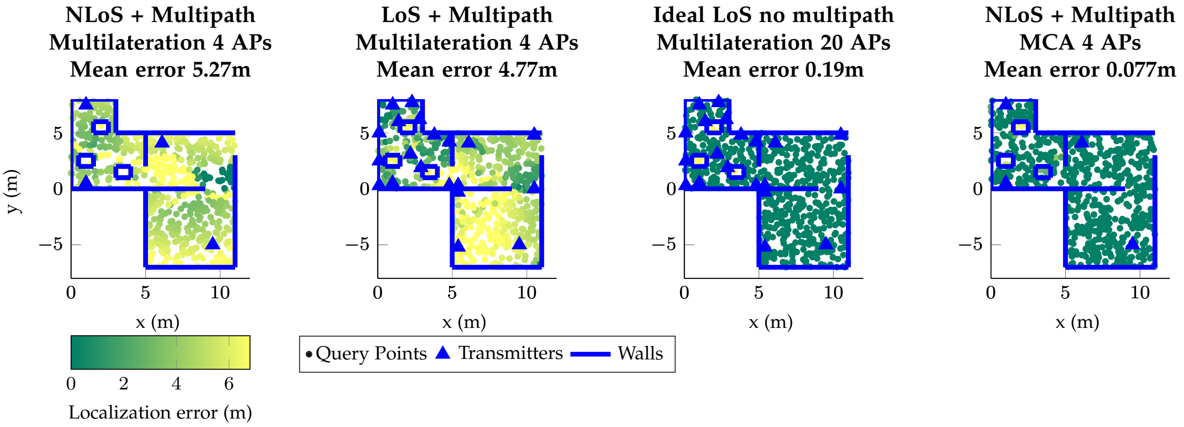


Figure 3.9: From left to right: 1) 4 anchors, multilateration multipath simulated (NLoS + multipath). 2) 20 anchors, multilateration multipath simulated (LoS + multipath). 3) 20 anchors, multilateration no multipath simulated (ideal LoS). 4) 4 anchors, MCA multipath simulated (NLoS + multipath).

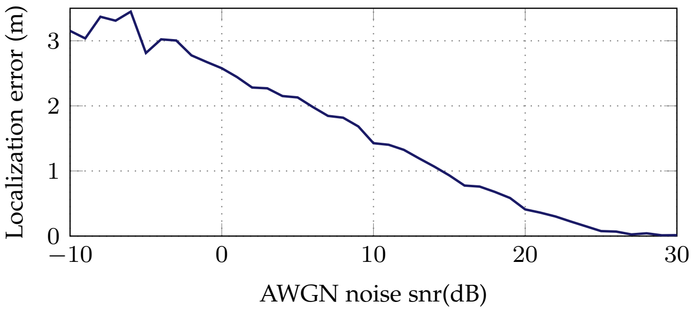


Figure 3.10: Localization error of the MCA algorithm with full fingerprints in the presence of AWGN noise. Figure adapted from [1] ©2021 IEEE.

the MCA algorithm was generated from a geometry without obstacles and the obstacles are added when the query data is generated. The MCA algorithm performs well in a static and

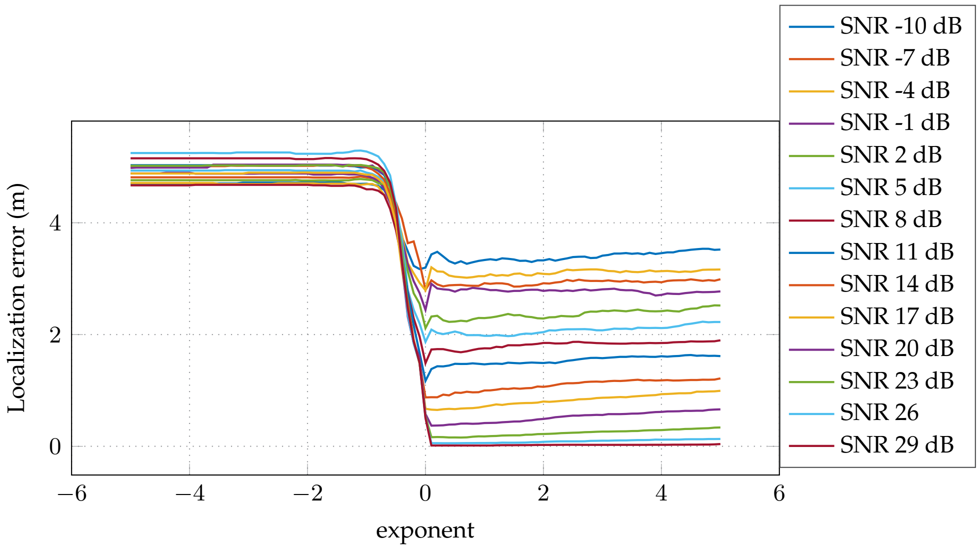


Figure 3.11: Mean localization error for different exponent values in the similarity metric.

a dynamic environment. In Fig.3.9 the performance of the MCA and trilateration algorithms are compared with different numbers of APs. It can be seen from the plots that to reliably localize the user in the given environment, the trilateration algorithm would require 20 APs and NLoS detection. At a given point LoS measurements are received from some APs and NLoS measurements from the others. If no NLoS detection is performed, the user would assume that the smallest reflected path from the NLoS transmitters is a LoS path, include it into the multilateration and obtain a wrong result. In contrast, the MCA algorithm performs well with only 4 APs and no NLoS detection. No noise is added to the MDP fingerprints. The small amount of error is present because most query points do not coincide with the reference points and the MCA algorithm does not perform k-means or any other averaging and instead returns the location of the reference fingerprint which is most similar to the query. Figure 3.10 shows the performance of the MCA algorithm when AWGN noise is added to both reference and query multipath components. Figure 3.11 shows the dependency of the localization accuracy on the exponent α in the similarity metric.

$$\gamma(\mathbf{D}^q, \mathbf{D}_i|\boldsymbol{T}_j) = \sum_{k\in\mathbf{Q}_{ij}} (\varepsilon - |d^q_{j,k} - d^*|)^\alpha, \tag{3.13}$$

$$\mathbf{Q}_{ij} = \{k|\exists d^* \in \boldsymbol{d}_{ij} : |d^q_{j,k} - d^*| \to min \cap |d^q_{j,k} - d^*| < \varepsilon\}. \tag{3.14}$$

In all other experiments and in [2], [1], [4] , the exponent $\alpha = 2$. The results show the optimal exponent value should be above 0 below 2. Note if the exponent is 0, the similarity metric is equal to the number of matches found between the two fingerprints.

3.6 Evaluation with Measurement Data

3.6.1 Measurement Setup

Measurements were conducted in collaboration with the Institute of Communications and Navigation at the German Aerospace Center (DLR). The UWB communication standard and the Decawave chip were chosen in this thesis to construct a prototype of the proposed multipath-based localization system. The UWB communication standard is described in Section 2.7.1.2 and the Decawave chip is described in Section 2.7.2.3. The measurements were conducted in one room in an area the size of which was approximately $10\ \mathrm{m}$ by $4\ \mathrm{m}$.

3.6.1.1 Receiver and APs

The Decawave DWM1000 chips[1] with UWB two-way-ranging were used both as the receiver and the APs. The DWM1000 module was attached by a breakout board to the GPIO outputs of a Raspberry Pi 3 B+. The DW1000 chip was operated and time stamps needed to calculate distance estimates were transferred by a communication software and SPI interface on the Raspberry Pi. The setup has 9 APs which are static and mounted on the walls at a height of roughly $1m$ as shown in Fig. 3.14. The receiver, also called a tag is carried by a person or attached to a robot in various experiments.The UWB system is configured to a bandwidth of 500 MHz and a carrier frequency of 3.5 GHz.

The receiver initiates the two-way-ranging method by a ranging request (poll) to one of the APs. The receiver and AP exchange four messages, two of which are sent by the receiver, and two by the AP. The estimated distance is afterwards calculated on the receiver side, see [219]. The Decawave DWM1000 chip also provides an estimated CIR. The DWM1000 chip includes a large bank of memory that holds the accumulated CIR data which contains complex values representing a $1ns$ sample interval. The receiver software running on the Raspberry Pi accesses this memory after a message from the AP is received. The space-alternating generalized expectation-maximization (SAGE) detailed in Section 3.6.2 is used to estimate the multipath delays [39]. It should be noted that the ILS prototype presented in this thesis uses active localization. The receiver or tag actively communicates with the APs to perform CIR estimation. In order for the system to support a very large number of users, it would be advantageous to modify the communication protocols into a scheme were the receivers only listen to the data transmitted by the APs. As the author of this thesis does not have access to the software or hardware of state-of-the-art ILSs, it was not possible to compare the performance of the proposed algorithm against the state-of-the-art algorithms in the same environment.

3.6.1.2 Ground-Truth Measurement

Existing bench-marking platforms described in Section 2.8 could not be used to evaluate the proposed localization approach due to lack of access and compatibility to the systems in question. Several reflectors were attached to the board and its location was tracked by a Vicon motion capturer [151] to obtain the ground truth data. 16 infrared sensitive cameras and infrared strobes were used. The accuracy of the Vicon system was better than $1\ \mathrm{cm}$ with a sampling rate higher than $100\ \mathrm{Hz}$. The correct ground-truth and multipath measurements were grouped together using the timestamps of the two measurement systems.

[1] https://www.decawave.com/product/dwm1000-module/

3.6.2 Multipath Extraction

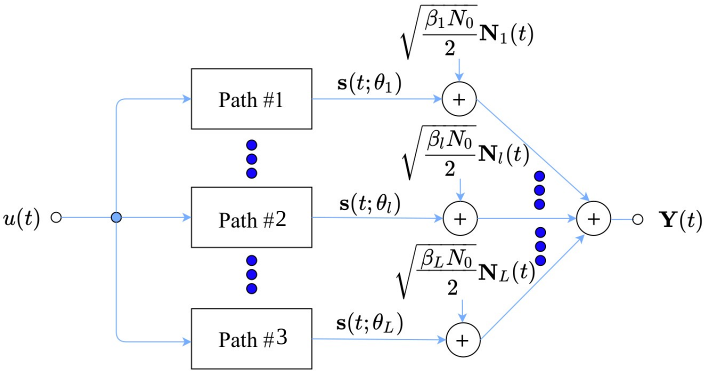

$u(t)$
Path #1
Path #2
Path #3
$\mathbf{s}(t;\theta_1)$
$\mathbf{s}(t;\theta_l)$
$\mathbf{s}(t;\theta_L)$
$\sqrt{\frac{\beta_1 N_0}{2}}\mathbf{N}_1(t)$
$\sqrt{\frac{\beta_l N_0}{2}}\mathbf{N}_l(t)$
$\sqrt{\frac{\beta_L N_0}{2}}\mathbf{N}_L(t)$
$\mathbf{Y}(t)$

Figure 3.12: System model used by the SAGE algorithm [39].

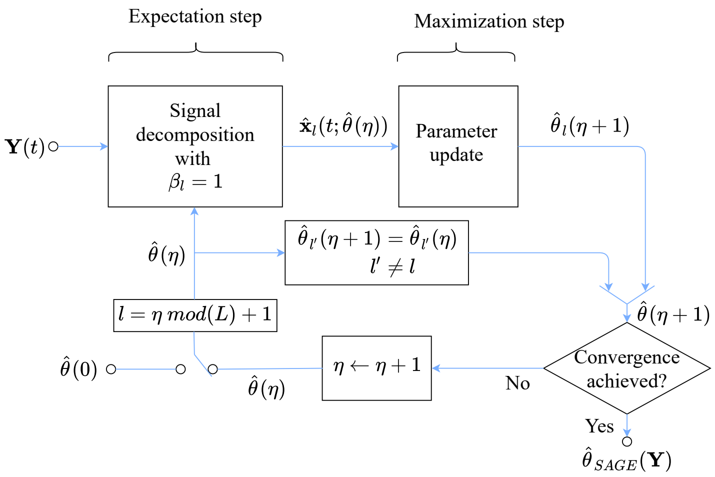


Figure 3.13: Signal flow graph of the SAGE algorithm [39].

In order to implement the MCA algorithm described in Section 3.4, multipath components need to be extracted from a received electromagnetic signal. The space-alternating generalized expectation-maximization (SAGE) algorithm is used in this thesis. It is designed to jointly estimate the relative delay, incidence azimuth, Doppler frequency, and complex amplitude of impinging electromagnetic waves. According to [39], the main advantage of the SAGE algorithm is its high resolution. The SAGE algorithm is derived from the maximum likelihood (ML) principle [221]. It is an extension of the expectation-maximization (EM) algorithm [222]–[224]. In the SAGE algorithm, the high-dimensional optimization process that is needed to compute the ML estimates of the multipath signal parameters is replaced by several sequential low-dimensional optimization steps [221].

To implement the SAGE algorithm, a transmitter sends out the sounding sequence $u(t)$ that consists of periodically repeated bursts. The sounding sequence is assumed by the SAGE algorithm to be of the following form [39]:

$$u(t) = \sum_{i=-\infty}^{\infty} a(t - iT), \tag{3.15}$$

$$a(t) = \sum_{k=0}^{K-1} a_k p(t - kT_p), \tag{3.16}$$

where the coefficients $[a_0, ..., a_{K-1}]$ denote the possibly complex sounding sequence, $p(t)$ denotes the shaping pulse of duration T_p and $T = KT_p$ denotes the time interval between consecutive signal bursts. The SAGE algorithm assumes that a set number of multipath components K_L is to be extracted from the received signal. If an array of M antennas is used, the received signal vector has the following form:

$$[Y_1(t), ..., Y_M(T)]^T = \boldsymbol{Y}(t) = \sum_{k=1}^{K_L} \boldsymbol{s}(t; \boldsymbol{\theta}_k) + \boldsymbol{N}(t), \tag{3.17}$$

$$[s_1(t; \boldsymbol{\theta}_k), ..., s_M(t; \boldsymbol{\theta}_k)] = \boldsymbol{s}(t; \boldsymbol{\theta}_k) = \boldsymbol{c}(\phi_k)\alpha_k exp\{j2\pi\nu_k t\}u(t - \tau_k), \tag{3.18}$$

where $\boldsymbol{c}(\phi) = [c_1(\phi), ..., c_M(\phi)]$ is the steering vector of the antenna array with $c_m(\phi) = f_m(\phi)exp\{j2\pi\lambda^{-1}\langle \boldsymbol{e}(\phi), \boldsymbol{r}_m\rangle\}$. The vector $\boldsymbol{\theta}_k = [\tau_k, \phi_k, \nu_k, \alpha_k]$ contains the parameters of the k-th multipath component, where τ_k is the relative delay of the multipath component, ϕ_k is the incidence azimuth, μ_k is the Doppler frequency, and α_k is the complex amplitude. λ denotes the wavelength, $\boldsymbol{e}(\phi)$ is the unit vector in $\mathbb{R}^2$ in the direction of ϕ, $f_m(\phi)$ is the complex electric field pattern of the m-th antenna. The expectation maximization (EM) algorithm then restructures the ML estimation into K_L separate maximization procedures, one for estimating the parameters for each of the K_L expected signal propagation paths. The SAGE algorithm further partitions the optimization problems according to the parameters being optimized. In this way, the algorithm performs a series of one-dimensional optimization steps [225]. The optimization is performed iteratively, from an initial set of estimated parameter values $\hat{\boldsymbol{\theta}}(0)$ a sequence $\hat{\boldsymbol{\theta}}(\eta)$ is generated, such that at each step $\eta + 1$ the parameter estimation is improved.

The ML parameters estimation of the k-th multipath component θ_k for the observation

$$\boldsymbol{X}_k(t) = \boldsymbol{x}_k(t) = s(t, \boldsymbol{\theta}_k) + \sqrt{\frac{\beta_k N_0}{2}}\boldsymbol{N}_k(t) \tag{3.19}$$

can be written as

$$(\hat{\boldsymbol{\theta}}_k)_{ML}(x_k) \in argmax_{\boldsymbol{\theta}_k}\{\Lambda(\boldsymbol{\theta}_k; \boldsymbol{x}_k)\}, \tag{3.20}$$

where $\Lambda(\boldsymbol{\theta}_k; \boldsymbol{x}_k)$ is the log-likelihood function of $\boldsymbol{\theta}_k$ for the observation $\boldsymbol{X}_k(t) = \boldsymbol{x}_k(t)$ and $\sqrt{\frac{\beta_k N_0}{2}}\boldsymbol{N}_k(t)$ is the component of the noise $N(t)$ that corresponds to the k-th multipath component. This is illustrated in Fig. 3.12. The signal $\boldsymbol{x}_k(t)$ received through the k-th multipath component is part of the total received signal $\boldsymbol{Y}(t)$ and cannot be directly observed. It can be estimated using the conditional expectation of $\boldsymbol{X}_k(t)$ on the received signal $\boldsymbol{Y}(t)$ and a previous estimate of the parameters $\hat{\boldsymbol{\theta}}_k(\eta)$:

$$\hat{\boldsymbol{x}}_k(t; \hat{\boldsymbol{\theta}}_k(\eta)) = E_{\hat{\boldsymbol{\theta}}_k(\eta)}[\boldsymbol{X}_k(t)|\boldsymbol{Y}(t)] \quad (k = 1, ..., K_L). \tag{3.21}$$

This is referred to as the *Expectation step* of the SAGE algorithm. In [39] the estimate $\hat{\boldsymbol{x}}_k(t;\hat{\boldsymbol{\theta}}(\eta))$ is calculated as:

$$\hat{\boldsymbol{x}}_k(t;\hat{\boldsymbol{\theta}}(\eta)) = \boldsymbol{s}(t;\hat{\boldsymbol{\theta}}_k(\eta)) + \beta_k[\boldsymbol{Y}(t) - \sum_{k'=1}^{K_L} \boldsymbol{s}(t;\hat{\boldsymbol{\theta}}_{k'}(\eta))], \tag{3.22}$$

where the first term $\boldsymbol{s}(t;\hat{\boldsymbol{\theta}}(\eta))$ is the contribution of the k-th multipath to the received signal, under the assumption that the multipath parameters have been calculated correctly. The term in the brackets is the noise estimate, based again on the hypothesis that the multipath parameters have been correctly estimated. Given the constraints that the coefficients β_k in the EM algorithm are non-negative and $\sum_{k=1}^{K_L} \beta_k = 1$, they are selected to maximize the convergence rate of the algorithm. For the SAGE algorithm in [39] $\beta_k = 1$. After the expectation step of the algorithm is completed, the multipath parameters in $\hat{\boldsymbol{\theta}}$ can be re-estimated by computing the maximum likelihood using the newly computed received signal component $\hat{\boldsymbol{x}}_k(t,\hat{\boldsymbol{\theta}}(\eta))$:

$$\hat{\boldsymbol{\theta}}_k(\eta+1) = (\hat{\boldsymbol{\theta}}_k)_{ML}(\hat{\boldsymbol{x}}_k(t;\hat{\boldsymbol{\theta}}(\eta))) \quad (k = 1, ..., K_L). \tag{3.23}$$

This is referred to as the *Maximization step* of the algorithm. Starting from an initial set of parameter values $\hat{\boldsymbol{\theta}}(0)$, the expectation and maximization steps are iterated until the distance between parameter estimates reaches a predefined threshold or the estimated parameter sequence $\hat{\boldsymbol{\theta}}(\eta)$ stabilizes. The SAGE algorithm differs from the EM algorithm because at each iteration of the algorithm not all but only a subset of the components of $\hat{\boldsymbol{\theta}}$ are updated, while the other components are fixed and are updated in another iteration. In the implementation of the SAGE algorithm in [39], each multipath parameter is estimated separately, therefore, the maximization step consists of three 1-dimensional optimization procedures. In addition, the SAGE algorithm is less complex and converges faster than the EM algorithm [39]. The signal flow of the SAGE algorithm is illustrated in Fig. 3.13.

Two ways of estimating the initial parameter vales $\hat{\boldsymbol{\theta}}(0)$ are proposed in [39]: either a initialization iteration of the algorithm is performed and $\hat{\boldsymbol{\theta}}(0)$ is calculated over several iterations from $\hat{\boldsymbol{\theta}}(-K_L) = [0, ..., 0]$, or the MUSIC algorithm is used to compute the initial estimate of the delays values and an initialization cycle of the SAGE algorithm is used to compute the initial values of the rest of the parameters.

A more detailed description and the performance evaluation of the SAGE algorithm can be found in [39].

In this thesis, the SAGE [39] algorithm was used for extracting the multipath components from the CIR provided by the Decawave board [2] described in Section 2.7.2.3. The number of multipath components extracted from the signal was set to K_L =15.

It should be noted that the multipath extraction and the SAGE algorithm were run offline after the measurements were completed.

[2] The measurement campaign was performed in collaboration with the Institute of Communications and Navigation at German Aerospace Center (DLR), Oberpfaffenhofen, Münchner Straße 20, 82234 Wessling, Germany. The hardware setup and the implementation of the SAGE algorithm was performed by the group at the Institute of Communications and Navigation.

If the receiver is moving, multipath components can be tracked over time and multipath estimation is made more accurate using an extended Kalman filter (EKF) or the Kalman Enchansed Super Resolution Tracking (KESRT) algorithm in [226], [227]. Filtering algorithms for multipath estimation is not used in this thesis.

3.6.2.1 MCA Localization Performance

Figures 3.16 - 3.21 show the localization error calculated at the reference points in the indoor environment. In all of the plots both the query and reference points are additionally downsampled only when plotting for better visibility. Much denser data-sets were used during calculation. The evaluation results are published in [1]. In the case of Figure 3.16, both the reference and query data were collected by the robot. The robot was stopped at every 30 cm on the trajectory shown in Figure 3.16 and moved at a slow speed between the these points. The robot was controlled by hand though a remote controller device. In the following this data-set is referred to as stationary. The stationary data is used as the fingerprint map or reference in all of the following experiments. There are a total of 897 reference fingerprints, with $22.4\ fingerprints/m^2$. The reference points are averaged such that the distance between two consecutive points is not less than 10 cm. The mean distance between two reference points on sections where the robot was moving and was not turned or artificially re-positioned was 12 cm. When two reference points are combined, their MDPs are also averaged. Each MDP contains 15 sorted multipath components.

In Fig. 3.16 the sets of query fingerprints and reference fingerprints are non-overlapping subsets of the above described static dataset. An average localization error of 0.057 m was obtained for a mean distance of 0.029 m between the query and closest reference fingerprints. Figure 3.15 shows the average localization error as a function of the average distance between the query fingerprints and the closest reference fingerprint. In this case, the reference data is further downsampled at different rates.

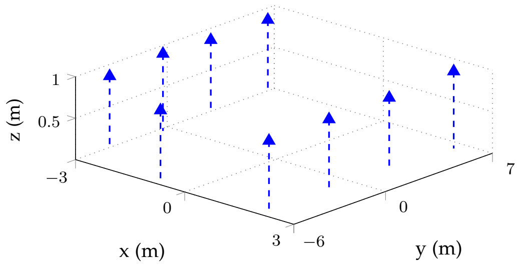


Figure 3.14: Measurement setup, AP locations. APs are mounted on the walls of the room. Figure adapted from [1] ©2021 IEEE.

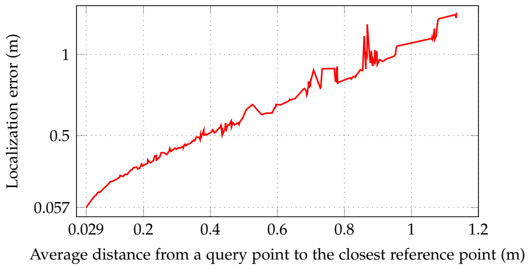


Figure 3.15: Localization error at the query points for different reference fingerprint densities. Static measurements. Figure adapted from [1] ©2021 IEEE.

In Fig. 3.17 the query data was collected by a robot moving at a fast walking pace. The distance between consecutive query fingerprints is 15 to 20 cm and the distance from a query to the closest reference point is $11cm$. A mean localization error of $0.308m$ was obtained. The ratio of the mean error to the mean distance increased compared to the static dataset. In the future work it should be checked whether the system update rate and timestamp synchro-

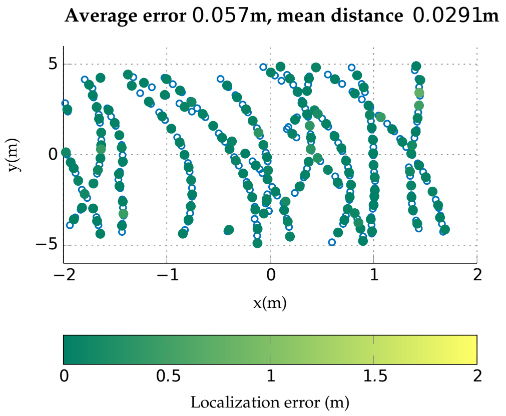


Figure 3.16: Localization error at query points. Near static measurements. Figure adapted from [1] ©2021 IEEE.

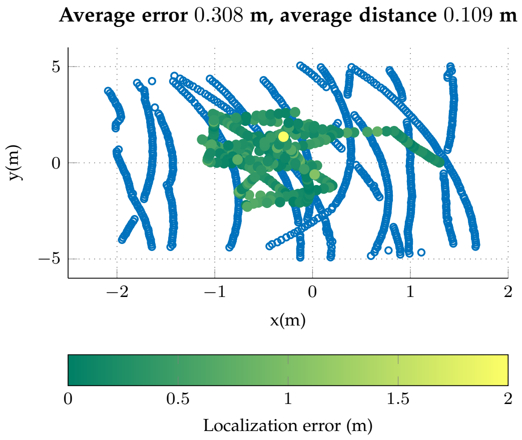


Figure 3.17: Localization error at query points. Query data collected by a moving robot. Figure adapted from [1] ©2021 IEEE.

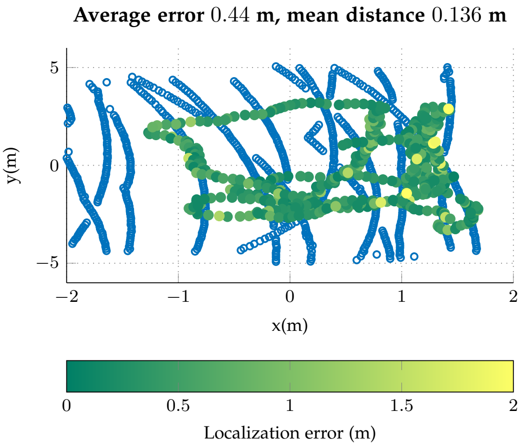


Figure 3.18: Localization error at query points. Query data collected by a moving person. [1] ©2021 IEEE.

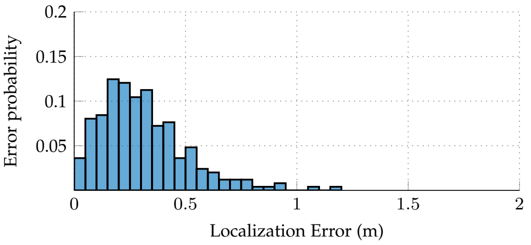


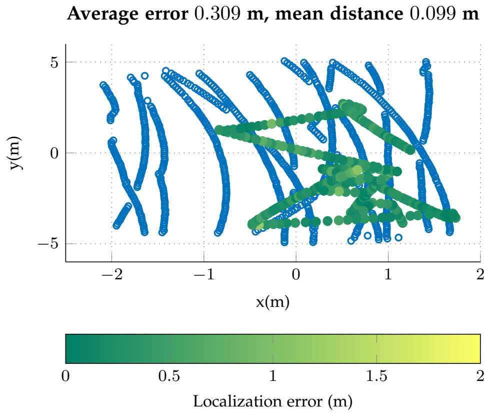
Figure 3.19: Probability distribution of the localization errors. Moving robot Fig. 3.17. Figure adapted from [1] ©2021 IEEE.

nization cause the increase in localization for the fast user measurements. Figure 3.19 shows the histogram of the probability distribution of the localization error values.

In Fig. 3.18 the query data was collected by a moving person holding the RasberryPi board in her hand. The mean distance between successive query points is 12.3 cm. An average error of $0.44m$ was obtained. The increase in error could be explained by the movement of the user's hand holding the Decawave board and the difference between the z-coordinates in the reference and query datasets.

In Fig. 3.20 the query data was again collected by a robot moving faster then walking pace. A person was moving around the room at the same time. The trajectories used in Figs. 3.17 - 3.20 to collect the query data are different due to the manual control of the robot. Figure 3.21 shows the average error of the experiments when only a subset of the APs were used to estimate the user's location.

The results presented in [1] show that the prototype ILS is able to precisely localize the



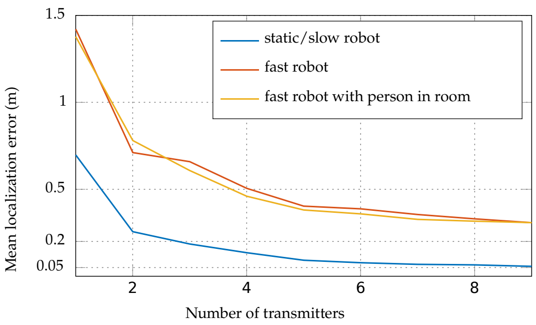


Figure 3.20: Localization error at query points. Query data collected by a moving robot, a person moving around the room during query measurements. Figure adapted from [1] ©2021 IEEE.

Figure 3.21: Average localization error for different numbers of transmitters used. Figure adapted from [1] ©2021 IEEE.

user when he or she is stationary or when his or her movement is slow. Figure 3.15 shows that when the receiver is moving very slowly, the localization error is around double the average distance from the query to nearest reference fingerprint. The localization accuracy of the ILS prototype decreases when the user is moving fast, however it is still comparable to commercial ILS. The accuracy the system most likely decreases because the quality of the CIR and multipath estimation decreases. This can happen because of Doppler effects as the user moves a large distance between two CIR measurements, incorrect timestamps are received and the wrong coordinates from the Vicon system are used to label the fingerprint. If several measurements are used to estimate a CIR it can happen that the user moves and those measurements are made at different locations, changing the resulting CIR and MDP. Figure 3.18 shows that even when the receiver is moving at a fast walking pace, the user is localized with an average error of approximately $30cm$. Figs. 3.20 and 3.21 show that the presence and movement of a person in the indoor environment does not affect the localization performance.

3.7 Summary

This chapter proposed the multipath component analysis (MCA) fingerprinting localization algorithm. The MCA algorithm uses the multipath delay profile (MDP) as a fingerprint to precisely localize the user even if changes or motion occurred in the indoor environment after the fingerprint map was created. The MCA algorithm is validated through simulation and measurements. A prototype was created in collaboration with the Institute for Communication and Navigation at the German Aerospace Center (DLR) using the UWB Decawave board. A slowly moving user was localized with an average accuracy of $6cm$, the average distance from a query point to the closest reference point in the fingerprint map was $3cm$. When the user was moving at a fast walking pace, an average accuracy of $30cm$ was obtained with an average distance from a query to the closest reference point of $11cm$.

Chapter 4

Privacy-preserving Indoor Localization

Tracking implies calculating, storing, and analyzing the user's location to determine patterns in his or her behavior. The malicious party may learn about the user's visits to a clinic, hospital and obtain private health-related information. It may also learn how robots and resources in the factory are moved around. Technology and other trade secrets may be obtained by a competitor. If the malicious party is intercepting communication messages, they can simply be encrypted to protect the user. However, the ILS itself can be malicious when it is installed by a company the user does not belong to. This is, for example, often the case in public spaces such as shopping malls. In this case encryption is not enough, the communication protocols between the ILS and the user have to be restructured such that the user can obtain his or her location from the ILS without the ILS knowing that location.

This chapter shows that the MDP fingerprint structure described in Chapter 3, is innately suited for adding user privacy protection to an ILS. It allows the user to send the ILS only partial data which enables the ILS to calculate several guesses of where the user may be. The user can use the full query fingerprint to choose between these guesses and himself or herself locally calculate a position. The privacy protection scheme detailed in this chapter was originally proposed in [1].

This chapter is organized as follows. Section 4.1 summarizes existing privacy protection schemes. Section 4.2 details the proposed privacy protection scheme. Section 4.3 summarizes the metrics proposed to evaluate privacy. The simulation and measurement results are presented in Sections 4.4 and 4.5. Section 4.6 contains the conclusion.

4.1 Existing Privacy Protection Schemes

This section summarizes existing ILS privacy protection schemes. As discussed in Section 2.5 in Chapter 3, a typical fingerprinting-based ILS follows the design illustrated in Figs. 4.2 and 4.1. The fingerprint map is stored on the ILS server. The user communicates with the APs, calculates a fingerprint and sends that fingerprint to the ILS. The ILS, in turn, sends the calculated location to the user. Note, the communication between the user and the ILS server can occur over any medium, not necessarily over the access points (APs) used for fin-

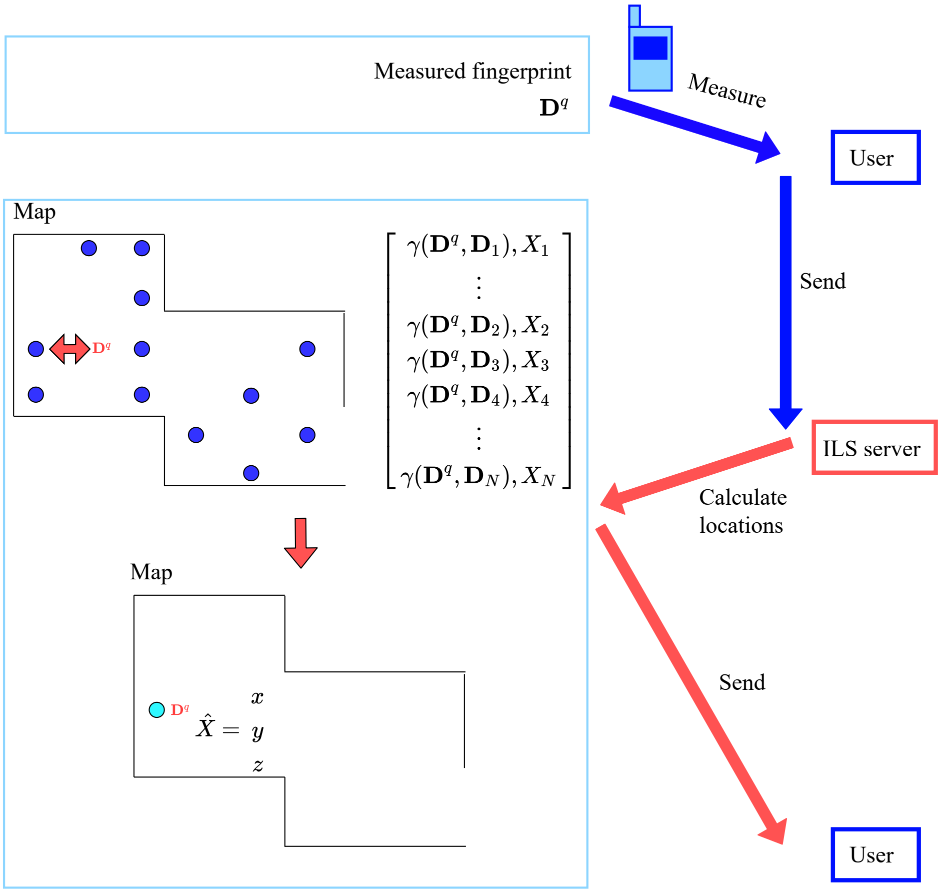

Measured fingerprint
$\mathbf{D}^q$
Measure
User
Map
$\gamma(\mathbf{D}^q, \mathbf{D}_1), X_1$
$\gamma(\mathbf{D}^q, \mathbf{D}_2), X_2$
$\gamma(\mathbf{D}^q, \mathbf{D}_3), X_3$
$\gamma(\mathbf{D}^q, \mathbf{D}_4), X_4$
$\gamma(\mathbf{D}^q, \mathbf{D}_N), X_N$
Send
ILS server
Calculate locations
Map
$\mathbf{D}^q$
$\hat{X} = x\ y\ z$
Send
User

Figure 4.1: Communication between the user and the ILS server without privacy protection. The ILS server calculates the location of the user and knows the user's identity.

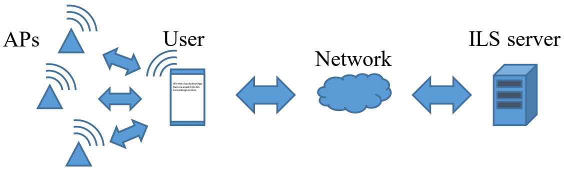


Figure 4.2: Assumed ILS system model. The user measures a fingerprint from the APs and obtains his or her location by communicating with the ILS server [1], [228]. Figure adapted from [1] ©2021 IEEE.

gerprinting. The privacy protection scheme would be built into the communication protocol between the user and the server. The main objective of doing this is to allow the user to obtain his or her location from the ILS server, without the server knowing that location.

Several solutions have been developed to protect the location privacy of the user. If the ILS is simultaneously communicating with several users and cannot tell which fingerprint was sent by which user, it cannot track them. Therefore, users can replace their IDs with frequently changing pseudonyms [228]–[236]. This is referred to as *k-anonymity* and is illustrated in Fig. 4.3. The problem with this approach is that it can create ID collisions and

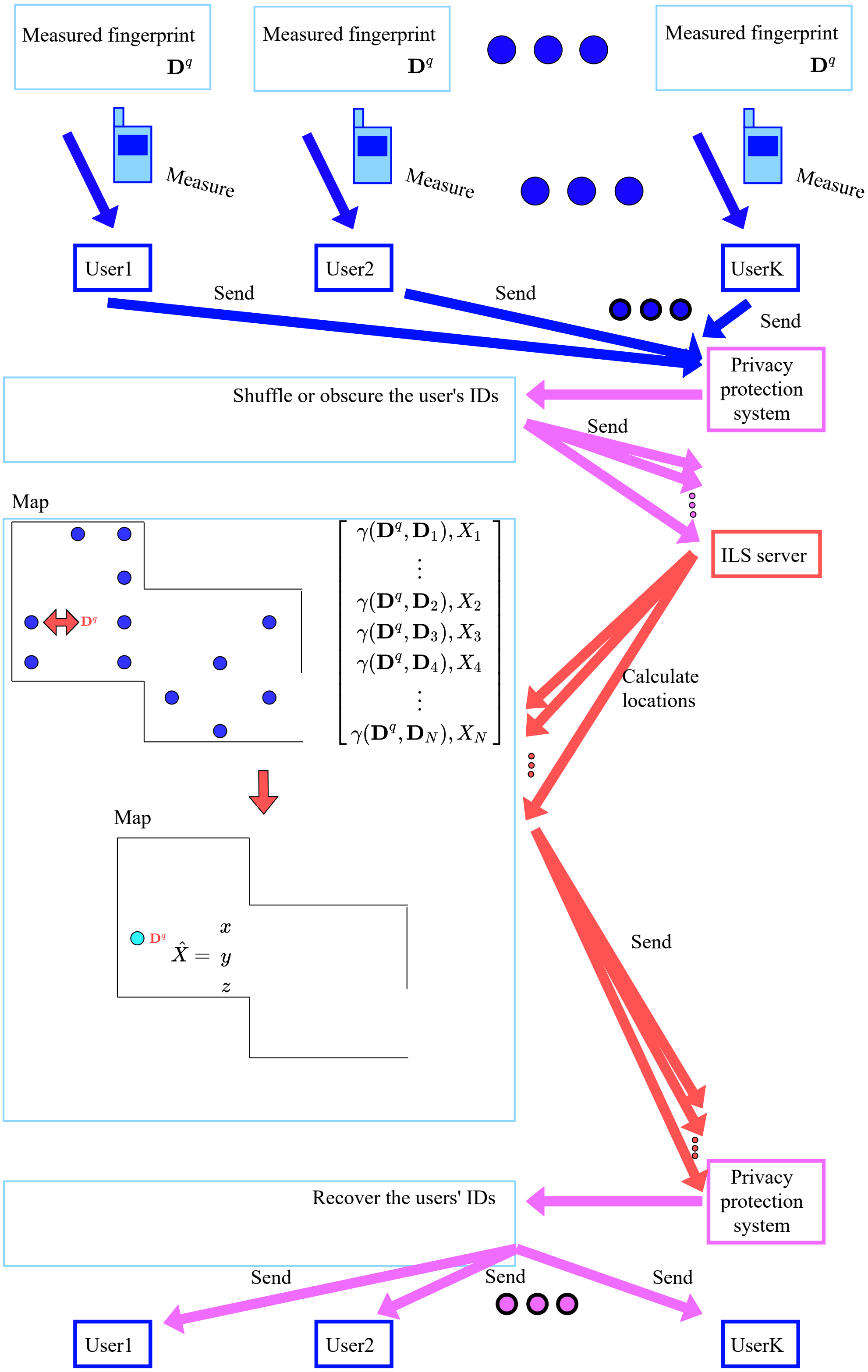

Measured fingerprint $\mathbf{D}^q$
Measured fingerprint $\mathbf{D}^q$
Measured fingerprint $\mathbf{D}^q$
Measure
Measure
Measure
User1
User2
UserK
Send
Send
Send
Privacy protection system
Shuffle or obscure the user's IDs
Send
ILS server
Map
$\mathbf{D}^q$
$\gamma(\mathbf{D}^q, \mathbf{D}_1), X_1$
$\gamma(\mathbf{D}^q, \mathbf{D}_2), X_2$
$\gamma(\mathbf{D}^q, \mathbf{D}_3), X_3$
$\gamma(\mathbf{D}^q, \mathbf{D}_4), X_4$
$\gamma(\mathbf{D}^q, \mathbf{D}_N), X_N$
Calculate locations
Map
$\mathbf{D}^q$
$\hat{X} = x\ y\ z$
Send
Privacy protection system
Recover the users' IDs
Send
Send
Send
User1
User2
UserK

Figure 4.3: Anonymity-based privacy protection. The ILS calculates the location of each user and hence is unable to distinguish between the users.

network disruptions. In addition, private wireless networks only allow users to have certain MAC-addresses. In this case the user cannot randomly alter his ID and retain access to the network [231]. At the same time, the ILS can with time differentiate between and track the users though the proximity of successive measurements [228]. The work in [234] extends the k-anonymity algorithm by segmenting each user's data and injecting noise into each piece. The users then exchange the data pieces among themselves, create mixed data-packets and send them to the ILS server. The ILS server calculates data tuples, which the users use to identify their true locations. While this approach protects the location of the users it requires them to cooperate in an organized manner and exchange data. Alternatively, it requires a privacy system that would know the location of each user. Further extensions of k-anonymity include the work in [235], where k-anonymity is combined with location encryption. The work in [236] assembles the data collected by the users into a motion graph and uses gene sequencing. However this approach was developed only for RSSI fingerprinting and not CSI and multipath fingerprints. In order not to force the user to obscure his or her ID, the Paillier cryptosystem is used to encode the data exchanged between the user and the ILS in [237]–[243]. The Paillier cryptosystem illustrated in Fig. 4.4 is an encryption scheme that allows the server to perform arithmetic operations on the encoded data. The user encodes the measured fingerprint. The ILS server can encode the reference fingerprints using a public key. The ILS server is then able to calculate the similarity metrics between the query and reference fingerprints without knowing the values in the query fingerprint. The resulting similarity values are also encoded and thus unknown to the ILS. The ILS sends the encoded similarity values and the corresponding reference coordinates to the user. In this way the reference fingerprints are also protected from the user. The user locally decodes the similarity values and determines his location. The work in [240] designs a privacy Paillier-based protection scheme specifically for CSI-fingerprints. While encoding-based privacy protection schemes are very thorough and guarantee the user's privacy, their computational complexity is very high. A large part of the computations have to be performed by the user locally. The transferred data volume is also very high as all of the calculated similarity metric values and all of the reference locations need to be sent by the server to the user. The transmitted data volume grows linearly with the size of the fingerprint map [237]. At the same time, the higher the accuracy of the ILS, the denser the reference locations and the larger the fingerprint database tends to be. An example of this is the scheme in [28]. Alternatively, the location privacy of the user is protected through AP fuzzification in [244]. Differential privacy is used in [245] to obscure RSSI data. In [63], [109] several semi-trusted third parties receive parts of encoded RSSI fingerprints from the ILS and calculate parts of encoded or secret-shared location estimates, the user combines the two secret-shared estimates to obtain his or her location. None of these approaches fit the ILS proposed in Chapter 3.

A different privacy protection scheme was proposed in [228], [246]–[248]. The user creates $k-1$ artificial fingerprints and sends them to the ILS together with the real measurements. The ILS does not know which fingerprint corresponds to the true location of the user and calculates a location estimate for all. The user then selects the response of the ILS to the real measurements as his location. Such schemes are referred to as *camouflage-based* pri-

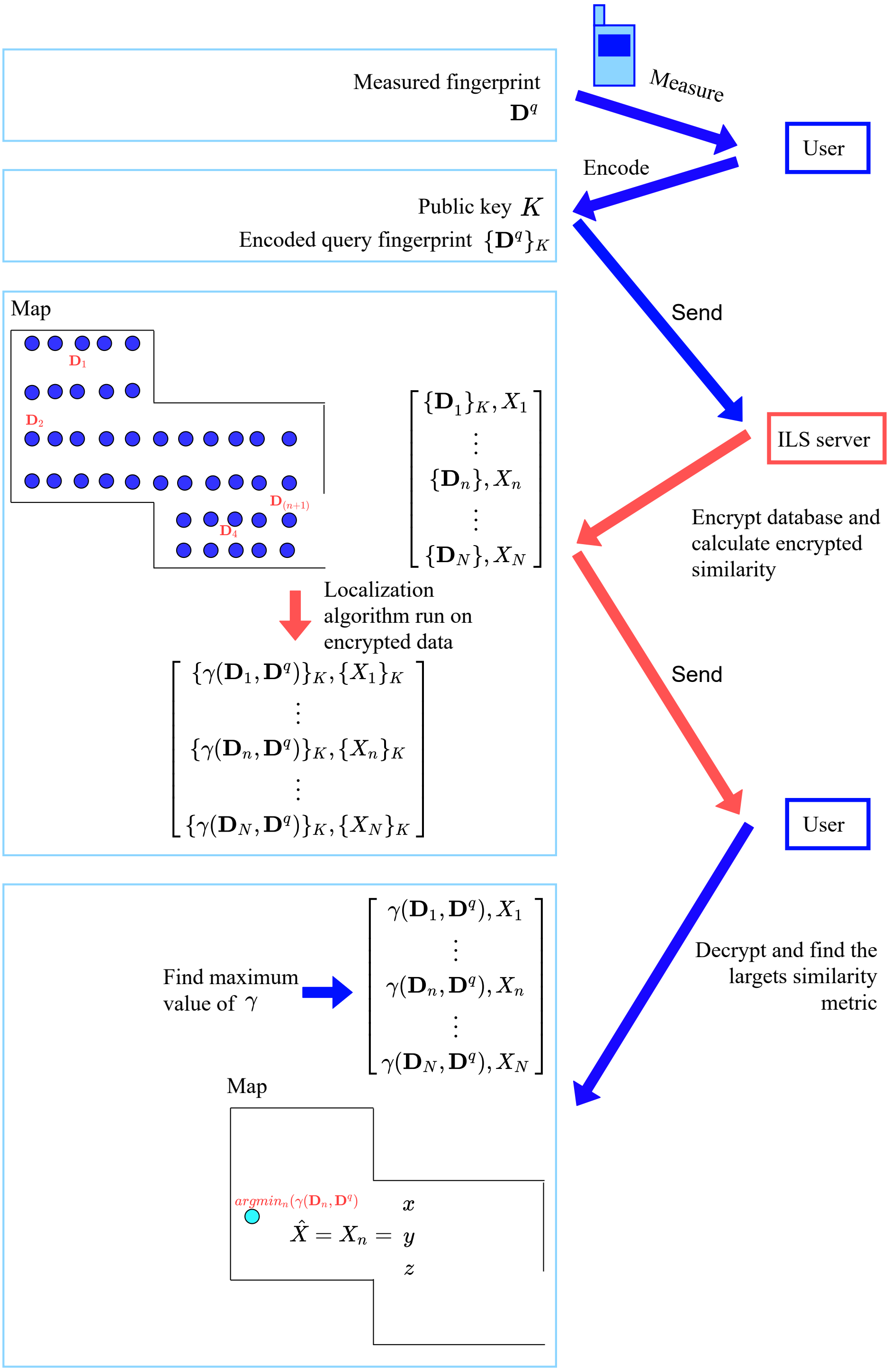

Measured fingerprint
$\mathbf{D}^q$
Measure
User
Encode
Public key K
Encoded query fingerprint $\{\mathbf{D}^q\}_K$
Send
Map
$\mathbf{D}_1$
$\mathbf{D}_2$
$\mathbf{D}_{(n+1)}$
$\mathbf{D}_4$
$\begin{bmatrix} \{\mathbf{D}_1\}_K, X_1 \\ \vdots \\ \{\mathbf{D}_n\}, X_n \\ \vdots \\ \{\mathbf{D}_N\}, X_N \end{bmatrix}$
ILS server
Encrypt database and calculate encrypted similarity
Localization algorithm run on encrypted data
$\begin{bmatrix} \{\gamma(\mathbf{D}_1, \mathbf{D}^q)\}_K, \{X_1\}_K \\ \vdots \\ \{\gamma(\mathbf{D}_n, \mathbf{D}^q)\}_K, \{X_n\}_K \\ \vdots \\ \{\gamma(\mathbf{D}_N, \mathbf{D}^q)\}_K, \{X_N\}_K \end{bmatrix}$
Send
User
Decrypt and find the largets similarity metric
Find maximum value of γ
$\begin{bmatrix} \gamma(\mathbf{D}_1, \mathbf{D}^q), X_1 \\ \vdots \\ \gamma(\mathbf{D}_n, \mathbf{D}^q), X_n \\ \vdots \\ \gamma(\mathbf{D}_N, \mathbf{D}^q), X_N \end{bmatrix}$
Map
$argmin_n(\gamma(\mathbf{D}_n, \mathbf{D}^q))$
$\hat{X} = X_n = \begin{matrix} x \\ y \\ z \end{matrix}$

Figure 4.4: Privacy protection using the Prailler cryptosystem. The user encrypts the query fingerprints, the ILS server encodes the fingerprint database with the user's public key and performs all of the computations on the encrypted data.

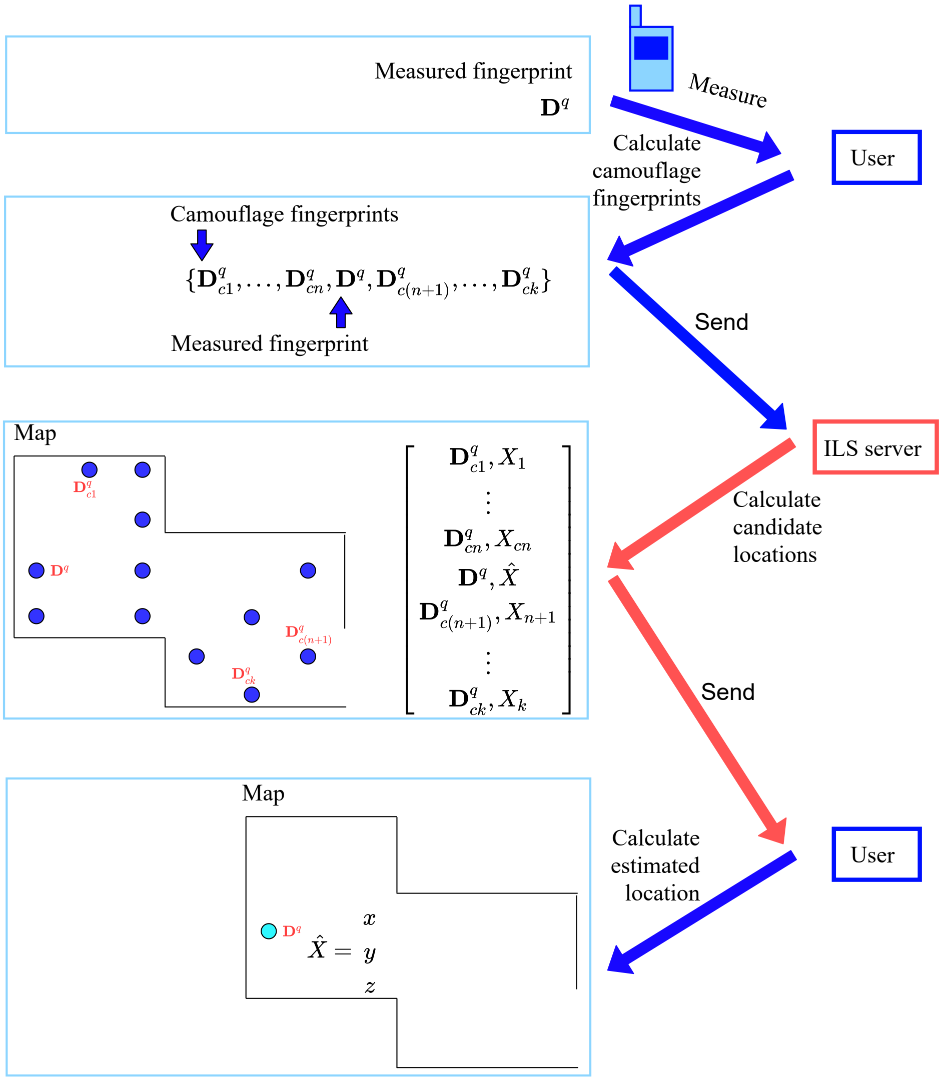


Figure 4.5: Camouflage-based privacy protection. The user generates fake camouflage fingerprints and sends them to the ILS server alongside the real measurements.

vacy protection schemes, the concept behind them is illustrated in Fig. 4.5. The artificially created fingerprints are referred to as *camouflage fingerprints* and the corresponding locations as *candidate locations*. This scheme has two main advantages. Firstly, no changes have to be made to the communication infrastructure and protocol between the user and the ILS server. Secondly, the user's degree of privacy and the scheme's complexity can be set by the number of camouflage fingerprints the user sends to the ILS server. Therefore, the user can set the degree of privacy according to his or her requirements, available computation power and communication bandwidth.

The main challenge of camouflage approaches is that the user needs to be able to generate credible fake fingerprints which the ILS server cannot distinguish from the real ones. RSSI fingerprints are vectors that fluctuate strongly throughout the environment. Believable RSSI fingerprints can be thus generated as vectors of random numbers. The CSI, CFR, CIR and multipath profile at a given location are more complex and are a function of the geometry. Therefore, not every vector can be a valid fingerprint. Unless proper care is taken, the server can identify the camouflage fingerprints as very dissimilar to all of the entries in the map. In addition, if the ILS is tracking a moving user and if the localization requests are very frequent, the ILS can identify the user's location as points which are close to each other and form a trajectory. This is illustrated in Fig. 4.10. The algorithms proposed in [246], [247] compute the camouflage RSSI fingerprints so that they appear to have a natural movement pattern. The scheme in [247] combines camouflage fingerprint generation with k-anonymity and uses the RSSI data from multiple users to generate credible RSSI fingerprints. To the best knowledge of the author of this thesis, no camouflage-based privacy scheme currently exists for channel based-fingerprints other than the one proposed in [1]. At the same time as shown in Chapters 2 and 3, CSI, CFR or multipath fingerprints are needed for high-precision localization.

4.2 Proposed Privacy Protection Scheme

Full Fingerprint	Complete set of multipath components or the complete information extracted from the signals received from the APs by the user.
Partial fingerprint	Subset of multipath components or of the information extracted from the signals received by the user from the APs.
Camouflage fingerprint	fake fingerprints artificially constructed by the user and sent to the ILS.
Candidate locations	the set of the possible location estimates that the ILS server calculates and sends to the user.

Table 4.1: Terms and definitions related to indoor localization privacy in this thesis.

In [1], the following approach is proposed. The user creates a number of *partial* fingerprints by selecting a subset of the measured multipath components.

$$\mathbf{D}_p^q = \{\boldsymbol{d}_{p1}^q \subseteq \boldsymbol{d}_1^q, ..., \boldsymbol{d}_{pM}^q \subseteq \boldsymbol{d}_M^q\} \tag{4.1}$$

The user also generates a number of fake partial fingerprints. Those fingerprints are then sent to the ILS server. The server matches each partial fingerprints to several locations and sends those guesses $\boldsymbol{X}_{ci}$ and the corresponding reference fingerprints $\mathbf{D}_{ci}$ to the user. The reference fingerprints selected by the server are referred to as the *candidate fingerprints*. The corresponding locations are named *candidate locations*. The user locally runs the MCA algorithm on the candidate fingerprints and locations he received and determines his location.

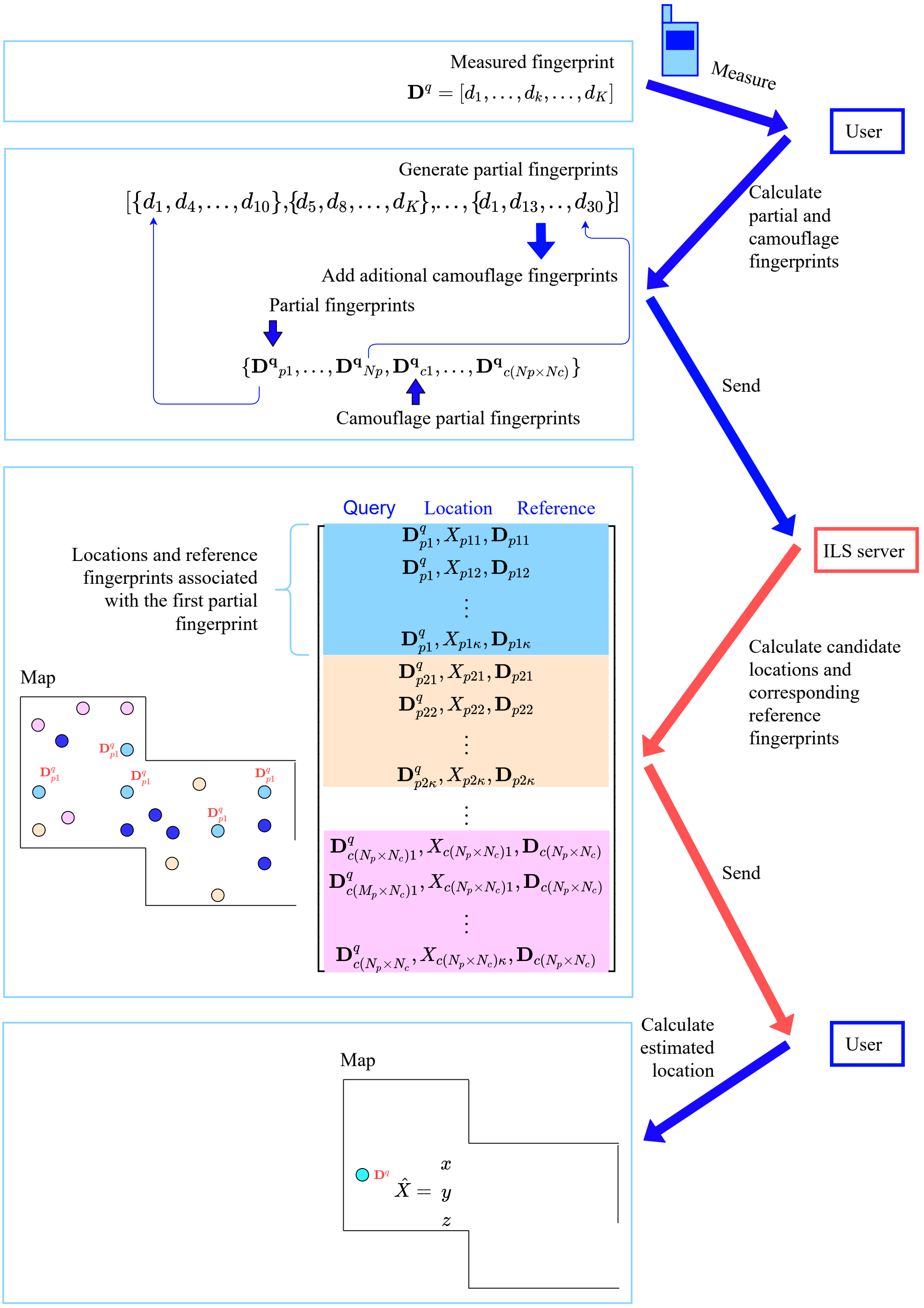

Measured fingerprint
$\mathbf{D}^q = [d_1, \ldots, d_k, \ldots, d_K]$
Measure
User
Generate partial fingerprints
$[\{d_1, d_4, \ldots, d_{10}\}, \{d_5, d_8, \ldots, d_K\}, \ldots, \{d_1, d_{13}, \ldots, d_{30}\}]$
Calculate partial and camouflage fingerprints
Add aditional camouflage fingerprints
Partial fingerprints
$\{\mathbf{D^q}_{p1}, \ldots, \mathbf{D^q}_{Np}, \mathbf{D^q}_{c1}, \ldots, \mathbf{D^q}_{c(Np \times Nc)}\}$
Camouflage partial fingerprints
Send
Query
Location
Reference
Locations and reference fingerprints associated with the first partial fingerprint
$\mathbf{D}^q_{p1}, X_{p11}, \mathbf{D}_{p11}$
$\mathbf{D}^q_{p1}, X_{p12}, \mathbf{D}_{p12}$
$\mathbf{D}^q_{p1}, X_{p1\kappa}, \mathbf{D}_{p1\kappa}$
$\mathbf{D}^q_{p21}, X_{p21}, \mathbf{D}_{p21}$
$\mathbf{D}^q_{p22}, X_{p22}, \mathbf{D}_{p22}$
$\mathbf{D}^q_{p2\kappa}, X_{p2\kappa}, \mathbf{D}_{p2\kappa}$
$\mathbf{D}^q_{c(N_p \times N_c)1}, X_{c(N_p \times N_c)1}, \mathbf{D}_{c(N_p \times N_c)}$
$\mathbf{D}^q_{c(M_p \times N_c)1}, X_{c(N_p \times N_c)1}, \mathbf{D}_{c(N_p \times N_c)}$
$\mathbf{D}^q_{c(N_p \times N_c}, X_{c(N_p \times N_c)\kappa}, \mathbf{D}_{c(N_p \times N_c)}$
Map
$\mathbf{D}^q_{p1}$
$\mathbf{D}^q_{p1}$
$\mathbf{D}^q_{p1}$
$\mathbf{D}^q_{p1}$
$\mathbf{D}^q_{p1}$
ILS server
Calculate candidate locations and corresponding reference fingerprints
Send
User
Calculate estimated location
Map
$\mathbf{D}^q$
$\hat{X} = \begin{matrix} x \\ y \\ z \end{matrix}$

Figure 4.6: Proposed privacy protection scheme. The user generates partial fingerprints from the query fingerprint. The user further generates partial camouflage fingerprints. The ILS server returns several candidate locations and reference fingerprints for each partial fingerprint.

The main advantage of this approach is that a partial fingerprint is likely to match multiple reference fingerprints in the map. Thus partial fingerprints are harder to identify as fake than full ones. In Fig. 3.3, $\{[d^q_{j_3}, d^q_{j_4}]\}$ is a partial fingerprint of the query fingerprint $\{[d^q_{j1}, .., d^q_{j4}]\}$. $\{d^q_{j1}, .., d^q_{j4}\}$ is a *full* query fingerprint.

This privacy protection scheme is only possible if MDP fingerprints are used. If partial fingerprints are selected as subsets of CFR or CSI taps, it is highly likely that those taps will only match the user's location. The assumption behind the approach [1] is that a small subset of multipath components can be associated by the ILS server to several different locations.

The MCA algorithm detailed in Section 3.4, can be used for calculating the similarity metric values for both full and partial query fingerprints. In the scenario when $\mathbf{D}^q$ is a full fingerprint, the MCA algorithm returns the user's estimated location. The full measured fingerprint is compared locally by the user to the candidate fingerprints. When the input of the MCA algorithm is a partial fingerprint, it will return a predetermined number κ of candidate locations and candidate fingerprints. The parameters ε and κ of the MCA algorithm are to be chosen empirically.

The proposed approach is summarized in Fig. 4.6 and Algorithms 3 and 4. The user randomly generates N_p partial fingerprints of length np from the measured query fingerprint $\mathbf{D}^q$. The partial fingerprints are denoted as $\{\mathbf{D}^q_{p1}, ..., \mathbf{D}^q_{pN_p}\}$. Since the generation of the partial fingerprints may not be enough to guarantee user privacy, N_c camouflage fingerprints of the same length np are generated for each partial fingerprint. These fingerprints are denoted as $\{\mathbf{D}^q_{c1}, ..., \mathbf{D}^q_{c(N_T - N_p)}\}$. Therefore, a total of N_T real and camouflage partial fingerprints of length np are sent by the user to the ILS. Using the MCA algorithm, the ILS determines κ best matches from the map for each partial fingerprint it receives, and returns them to the user. The user locally runs the MCA algorithm to determine his or her true location with the full measured fingerprint as the query and the candidate fingerprints and locations it receives from the ILS as the new map.

The user can generate the camouflage partial fingerprints in two different ways as illustrated by Algorithms 3 and 4. In Algorithm 3 a number of random multipath propagation distances is generated for each transmitter. The total number of multipath components generated is equal to the parameter np. In Algorithm 4 the multipath distances in the camouflage partial fingerprints $\{\mathbf{D}^q_{p1}, ..., \mathbf{D}^q_{pN_p}\}$ are the same as in the measured query, however, their transmitter IDs are randomly altered.

Using MDP partial camouflage fingerprints for privacy protection has the following advantages:

- The computational load on the user is low, as no complex operations need to be performed in order to generate credible partial MDP camouflage fingerprints. This statement may not apply if a different fingerprint structure is used.

- The camouflage partial fingerprints are automatically considered credible by the ILS server. If the number of multipath components in the partial fingerprints is small they are more likely to be present in multiple reference fingerprints. Since the transmitter

```
Algorithm 3: Camouflage Fingerprint Generation I [1] ©2021 IEEE.
  Data: Measured fingerprint $\mathbf{D}^q = \{\boldsymbol{d}^q_j\}, j = 1...M$
  Result: Partial fingerprints to be sent to the ILS $\{\mathbf{D}^q_{p1}, ..., \mathbf{D}^q_{pN_p}, \mathbf{D}^q_{c1}, ..., \mathbf{D}^q_{c(N_T-N_p)}\}$
  for $i = 1...N_p$ do
      Select random subsets: $\mathbf{D}^q_{pi} \leftarrow \{\boldsymbol{d}^q_{p1} \subseteq \boldsymbol{d}^q_1, ..., \boldsymbol{d}^q_{pM} \subseteq \boldsymbol{d}^q_M\}$
      Enforce: $\sum_j |\boldsymbol{d}_{pj}| = np$
  for $l = 1...(N_T - N_p)$ do
      Set random propagation distances: $\mathbf{D}^q_{cl} \leftarrow \{[rand_1, ..., rand_{K_j}]\}, j = 1...M$
      Enforce: $\sum_j K_j = np$
```

IDs of the camouflage fingerprints are generated randomly, the candidate locations will be distributed across the environment.

- The user can control the computational complexity, the data volume, i.e., the number of candidate locations, and the degree of privacy by setting the parametersN_T and N_p.

In order to apply the proposed privacy protection protocol to CIR or CFR-based fingerprints, the multipath components need to be estimated from the CIR or CFR and the fingerprints need to be converted to an MDP structure.

```
Algorithm 4: Camouflage Fingerprint Generation II [1] ©2021 IEEE.
  Data: Measured fingerprint $\mathbf{D}^q = \{\boldsymbol{d}^q_j\}, j = 1...M$
  Result: Partial fingerprints to be sent to the ILS $\{\mathbf{D}^q_{p1}, ..., \mathbf{D}^q_{pN_p}, \mathbf{D}^q_{c1}, ..., \mathbf{D}^q_{c(N_T-N_p)}\}$
  for $i = 1...N_p$ do
      Select random subsets: $\mathbf{D}^q_{pi} \leftarrow \{\boldsymbol{d}^q_{p1} \subseteq \boldsymbol{d}^q_1, ..., \boldsymbol{d}^q_{pM} \subseteq \boldsymbol{d}^q_M\}$
      Enforce: $\sum_j \left|\boldsymbol{d}^q_{pj}\right| = np$
      for $l = 1...N_c$ do
          Shuffle the transmitter IDs: $\mathbf{D}^q_{c(N_c\times(i-1)+l)} \leftarrow \{[d^q_{j1}, ..., d^q_{jK_j}]\}, j = 1...M$
          Enforce: $d^q_{jk} \in \boldsymbol{d}^q_{pr} \in \mathbf{D}^q_{pi} \cap \sum_j K_j = np$
```

4.3 Evaluating Privacy

A privacy metric is needed to quantify the degree of location privacy of a user in a system. Figure 4.7 shows the distribution of the candidate locations produced by four example camouflage privacy protection schemes. It can be qualitatively seen that the scheme in the upper left corner guarantees the most privacy to the user. However, a metric is needed to quantify the degree of privacy. Entropy has been used before to evaluate privacy, however the quantity for which the entropy was calculated varies for different privacy protection schemes. For example, the entropy-based privacy metric used in [234] is specific to the k-anonymity and

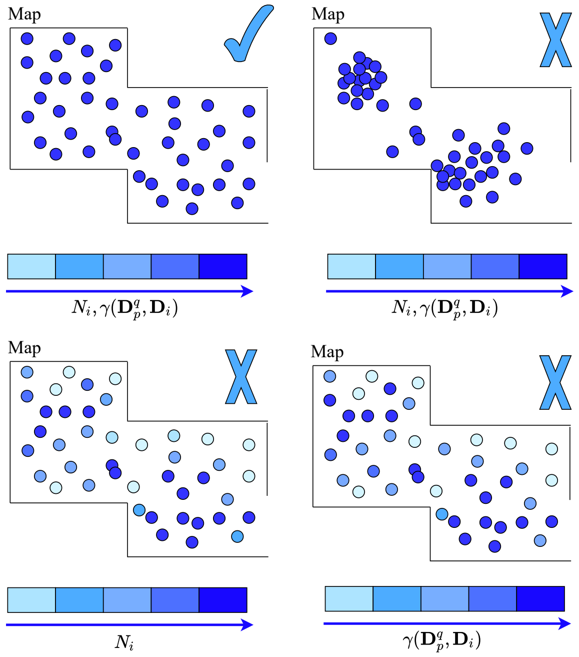

Map
Map
$N_i, \gamma(\mathbf{D}_p^q, \mathbf{D}_i)$
$N_i, \gamma(\mathbf{D}_p^q, \mathbf{D}_i)$
Map
Map
N_i
$\gamma(\mathbf{D}_p^q, \mathbf{D}_i)$

Figure 4.7: Example distributions of the candidate locations calculated by the ILS. Top left: the ILS cannot deduce the location of the user. Top right: the ILS can guess the location of the user due to the distribution of the candidate points. Bottom left: the ILS can estimate the location of the user because some reference points were selected as candidates more often than other. Bottom right: the ILS can estimate the position of the user as the calculated similarity metric to the query is much higher for some reference fingerprints than for others. Figure adapted from [1] ©2021 IEEE.

differential privacy-based scheme. Since no commonly used metric was available, three privacy evaluation schemes were proposed by the author of this thesis in [1]. The first two are quantitative measures of the degree of privacy of a stationary user. The privacy of a moving user is also evaluated. The privacy metrics are designed to evaluate camouflage-based privacy protection schemes for fingerprinting ILSs. It should be noted that the metrics in this section are heuristics that estimate privacy, they do not define its exact numerical value.

In the following, several aspects of a camouflage algorithm that impact user privacy are considered. In order to derive a metric for the privacy protection of a stationary user:

- First, an approximation is derived for the likelihood, from the point of view of the ILS, that the user is located in each of the reference points.
- The entropy of the likelihood distribution is used to approximate the degree of privacy.

4.3.1 Fingerprint Occurrence Frequency

In a camouflage-based privacy protection scheme, the ILS determines the κ best matching fingerprints to each partial fingerprint sent by the user. The user sends more than one real partial fingerprint. Therefore, if the same reference fingerprint is selected as a match to multiple partial fingerprints by the ILS, the ILS may suspect it to be the location of the user. This is illustrated by the bottom left plot of Fig. 4.7. Let N_i be the number of times the reference fingerprint $\mathbf{D}_i$ is considered "matched" within one localization step.

Statement 1. *Let N_i be the number of times reference fingerprint $\mathbf{D}_i$ is picked by the server as a candidate to a partial fingerprint sent by the user within one localization step. The degree of privacy protection in a system is highest when N_i is equal for all reference fingerprints.*

Proof. A higher N_i means a stronger belief of the ILS that the user is located at $\boldsymbol{X}_i$. If N_i is equal for all reference locations, the ILS cannot prefer any of them as a possible user location. □

4.3.2 Camouflage Fingerprint Credibility

The ILS calculates the similarity metric between each partial fingerprint sent by the user and the reference fingerprints in the map. A higher similarity value means that a point is more likely to be the location of the user. If the real partial fingerprints produce high similarity values to some reference fingerprints in the map and the camouflage partial fingerprints only produce small similarity values, it will be easy to differentiate real fingerprints from fake ones. This is illustrated in the bottom right plot of Fig. 4.7.

Proposition 1. *If the sum of the similarity metric values $\rho(\boldsymbol{X}_i)$, calculated each time that the reference point $\boldsymbol{X}_i$ was picked as a candidate location, is higher for $\boldsymbol{X}_i$ than for other reference points that can indicate that the user is located at $\boldsymbol{X}_i$.*

$$\rho(\boldsymbol{X}_i) = \frac{1}{\sum\limits_{k=1}^{N} \rho(\boldsymbol{X}_k)} \sum\nolimits_{\mathbf{D}_s^q \in \mathbf{L}(\mathbf{D}_i)} \gamma(\mathbf{D}_s^q, \mathbf{D}_i), \tag{4.2}$$

$$\mathbf{L}(\mathbf{D}_i) = \{\forall \mathbf{D}_s^q \mid \gamma(\mathbf{D}_s^q, \mathbf{D}_i) \in \textit{ top } \kappa \textit{ max values}\}, \tag{4.3}$$

where $\mathbf{D}_s^q$ are the partial fingerprints sent by the user.

Proof. $\rho(\boldsymbol{X}_i)$ combines the values of $\gamma(\mathbf{D}_p^q, \mathbf{D}_{ci})$ for a candidate location $\boldsymbol{X}_i$ with the value N_i in Statement 1. $\rho(\boldsymbol{X}_i)$ is also normalized. □

Proposition 2. *The entropy $H(\rho(\boldsymbol{X}_i))$ can be used as heuristic estimate of the location privacy of the user.*

Proof. According to the definition of entropy, the higher $H(\rho(\boldsymbol{X}_i))$ is, the higher the uncertainty of the ILS about the user's location will be. Entropy is also used to measure the degree of privacy in [229], [232] and [228]. □

Claim 1. *The upper bound $H(\rho(\boldsymbol{X}_i)) = log_2 N$, where N is the total number of reference fingerprints in the map holds for the proposed metric.*

Proof. The entropy H is maximal for a uniform probability distribution. □

4.3.3 Spatial Distribution

The entropy in the expression above was calculated from the occurrence frequency and similarity values of the candidate fingerprints. It does not take into account the locations of the candidate points. However, in case the user is in a shopping mall or hospital, the ILS server should not be able to identify which shop or doctor's office the user is closest to. If all of the most likely candidate fingerprints are located in close proximity, the ILS will know the user's location within a radius. This is illustrated in the top right plot of Fig. 4.7. Given the same value of $H(\rho(\boldsymbol{X}_i))$, if the most likely points are distributed throughout the environment, the ILS cannot determine where the user is. This is illustrated in the top left plot of Fig. 4.7. Therefore, in a camouflage scheme that well protects the user's privacy, the candidate points should be distributed over as large an area as possible, and the distances between neighboring candidate points should be as large as possible.

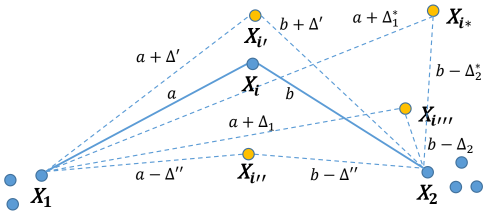


Figure 4.8: Illustration of the proof of Proposition 3. The point $\boldsymbol{X}_i$ is shifted to $\boldsymbol{X}_{i'}$, $\boldsymbol{X}_{i''}$, $\boldsymbol{X}_{i'''}$ and $\boldsymbol{X}_{i*}$, $a \geq b$. Figure adapted from [1] ©2021 IEEE.

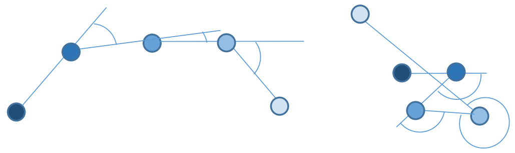


Figure 4.9: Two candidate trajectories. The left trajectory is more likely to be a natural trajectory of the user. Figure adapted from [1] ©2021 IEEE.

Proposition 3. *The parameter ω characterize the spatial distribution of the candidate points when*

$$\omega = \frac{1}{\omega_u} \sum_{k=1}^{N} \sum_{l=1}^{N} \rho(\boldsymbol{X}_k) \times \rho(\boldsymbol{X}_l) \times \sqrt{d(\boldsymbol{X}_k, \boldsymbol{X}_l)}, \tag{4.4}$$

$$\omega_u = \frac{1}{N^2} \sum_{k=1}^{N} \sum_{l=1}^{N} \sqrt{d(\boldsymbol{X}_k, \boldsymbol{X}_l)}, \tag{4.5}$$

where $d(\boldsymbol{X}_k, \boldsymbol{X}_l)$ is the euclidean distance between $\boldsymbol{X}_k$ and $\boldsymbol{X}_l$. N is the number of reference fingerprints in the map.

Proof. Consider the example point distribution in Fig. 4.8. Ideally ω should decrease if the point $\boldsymbol{X}_i$ is moved away from the other points and should decrease if it moves towards them. This is the case if it is shifted to location $\boldsymbol{X}_{i'}$. Similarly, if $\boldsymbol{X}_i$ is shifted to $\boldsymbol{X}_{i''}$, it moves closer to the point clusters around $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. The privacy provided by this candidate point distribution and ω decrease. Let $\boldsymbol{X}_i$ be shifted to location $\boldsymbol{X}_{i*}$. As $\Delta_1^* > \Delta_2^*$, $|\boldsymbol{X}_{i*}\boldsymbol{X}_1| - |\boldsymbol{X}_i\boldsymbol{X}_1|$ is larger than the distance $|\boldsymbol{X}_i\boldsymbol{X}_2| - |\boldsymbol{X}_{i*}\boldsymbol{X}_2|$. This means that $\boldsymbol{X}_i$ moved further from $\boldsymbol{X}_1$ by more than it moved closer to $\boldsymbol{X}_2$. In this case it is not immediately clear whether the area covered by the candidate points and the privacy increased or decreased. The following approximation is made.The change in privacy is assumed to correspond to the change in the value of ω. In the case that $\boldsymbol{X}_i$ be shifted to $\boldsymbol{X}_{i'''}$, and $\Delta_2 > \Delta_1$, the area covered by the reference points decreases. It is shown in [1] that ω also decreases if $\sqrt{a}+\sqrt{b} > \sqrt{a+\Delta_1}+\sqrt{b-\Delta_2}$. □

An alternative approach can be used to evaluate how evenly the candidate locations are distributed throughout the environment. For each candidate point $\boldsymbol{X}_{ci}$ the distance $|d_{ci}|$ to the nearest other candidate point can be calculated. If the value of d_{ci} is the same for all of the candidate points, this would indicate that they are evenly spread throughout the environment. If there is clustering of candidate points this would create spikes in the distribution of $|d_{ci}|$, $f_{d_c}(d)$. The following parameters can be then used to evaluate a candidate point distribution:

- The probability distribution $f_{d_c}(d)$ of the values $|d_{ci}|$, the distances from each candidate point to the nearest other candidate point. If the candidate points are uniformly distributed around the environment, all the $|d_{ci}|$ values will be the same.
- The average distance from a candidate point to the closest other candidate point $E[|d_{ci}|]$. This values should be compared to the size of the environment and the number of candidate points.
- The variance of $|d_{ci}|$ values calculated as $E[(|d_{ci}| - E[|d_{ci}|])^2]$. If the candidate points are uniformly distributed around the environment the variance of the $|d_{ci}|$ values will be small.

Testing the effectiveness of this privacy metric and the comparison of it to the metric detailed above is part of the future work.

4.3.4 Privacy of a Moving User

In the following, privacy estimation is performed for a user who is moving around the indoor environment and sending the ILS localization requests at regular time intervals τ_L. On average the user moves by a distance d_L between two consecutive localization requests. The main difference between the camouflage privacy protection of a stationary and moving user is that the ILS can identify natural movement patterns within the candidate points and thus obtain a better guess about the user's location.

If the user sends in localization requests very often, the real locations of the user will form a path that is easy to identify among the candidate points. This is illustrated in the upper right corner of Fig. 4.10. The trajectory of the user is marked in red. In the lower left plot of Fig. 4.10 the locations on the user's trajectory are spaced further apart. However, after a few localization requests the server can still identify the user's trajectory. All of the other possible trajectories the server can find, marked with black arrows, follow unnatural movement patterns and can be disregarded. Figure 4.9 shows two trajectories, the left trajectory follows a natural movement pattern and the right trajectory is unlikely to follow a person and will be referred to as unnatural. As is shown in Fig. 4.9, the ILS can determine if a trajectory follows a natural pattern by evaluating the angles between the line segments. In the lower right plot of Fig. 4.10, the real locations of the user are spaced far apart and the server cannot easily locate the user. However, if the server follows all possible trajectories over a long period of time it can disregard all trajectories and identify the real one marked in red. However, in this case locating the user would take the server much more time and computational power than in the lower left and upper right plots. In the upper left plot, the candidate points are distributed throughout the indoor environment. There are multiple natural trajectories marked with black arrows and the server cannot easily identify the user.

It can be assumed that if a sever follows all possible trajectories formed by the successive candidate points over time, has unlimited computational power and employs machine learning and pattern recognition algorithms, it can eventually figure out the user's location. Therefore, the goal of the derived camouflage privacy metric is not to show, whether or not the server can identify a moving user, but to estimate the number of localization steps and the computational effort it would take for the server to locate the user.

Proposition 4. *In order for the ILS to be unable to identify a moving user, the condition* $d_L \geq E[d(\boldsymbol{X}_{c,t}, \boldsymbol{X}_{c,t+\tau_L})]$ *must hold.* $E[d(\boldsymbol{X}_{c,t}, \boldsymbol{X}_{c,t+\tau_L})]$ *is the average distance between two candidate locations at time* t *and at time* $t + \tau_L$*.*

Proof. If d_L is small compared to the average distance between two neighboring candidate points across two successive localization steps, the ILS can deduce the location of the user by finding the smallest $d(\boldsymbol{X}_{c,t}, \boldsymbol{X}_{c,t+\tau_L})$. □

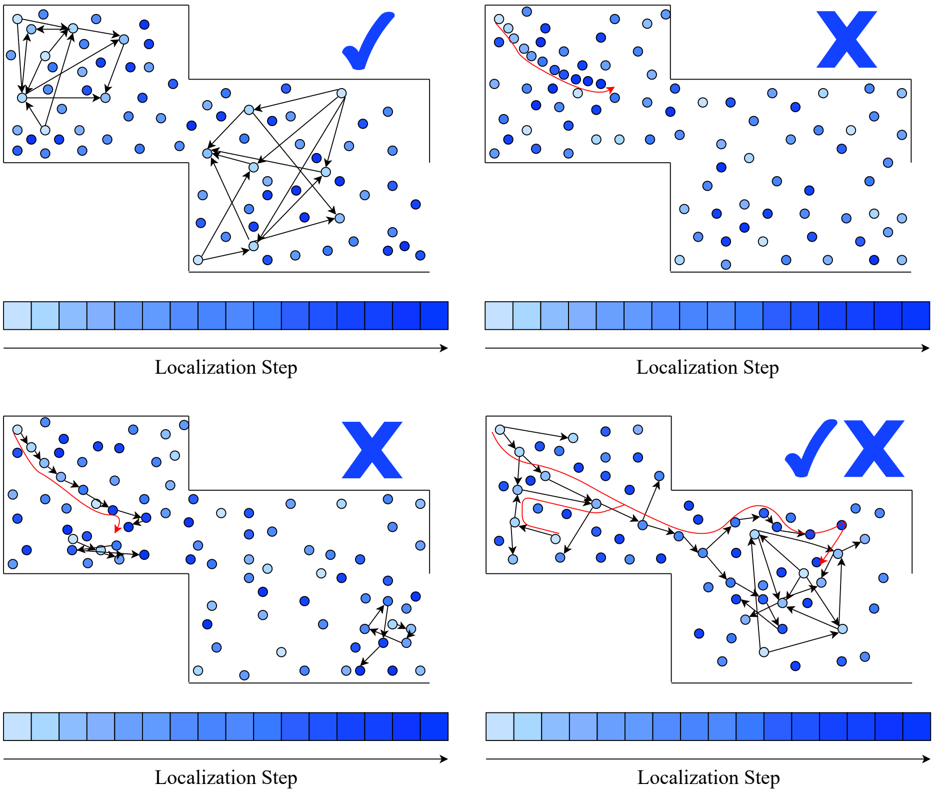


Figure 4.10: Possible distributions of candidate points that the ILS calculates for a moving user. Top left: example of an ideal distribution, top right: the trajectory of the user is easy to identify, bottom left: the ILS has several guesses as to which trajectory belongs to the user, needs several localization steps to identify the user, bottom right: the ILS needs many localization steps in order to find the movement pattern of the user. Figure adapted from [1] ©2021 IEEE.

Statement 2. *Let $\eta(d)$ be defined as the average number of candidate locations $\boldsymbol{X}_{c,t+\tau_L}$, sent by the ILS to the user at time $t+\tau_L$, the distance from which to to a candidate location $\boldsymbol{X}_{c,t}$, calculated at time t, is smaller than d. $\eta(d)$ can be used to estimate the degree of privacy of a moving user.*

$$\eta(d) = E\left[\,|\{\forall \boldsymbol{X}_{c,t+\tau_L} \mid d(\boldsymbol{X}_{c,t+\tau_L}, \boldsymbol{X}_{c,t}) \leq d\}|\,\right]. \tag{4.6}$$

Proof. If $\eta(d_L)$ is large this means that the candidate point set contains a number of candidate trajectories with a step size smaller than d_L. Therefore, the ILS needs to analyze a large number of candidate trajectories to identify the user. At the next iteration of the localization algorithm at time $t+\tau_L$, $\eta(d_L)$, several new candidate trajectories may be created from each candidate trajectory, existing at time t. If more trajectories are created than can be disregarded by the ILS server as unnatural, the number of trajectories the ILS server needs to track and analyze will increase exponentially with $\eta(d_L)$. □

4.4 Ray-tracing Simulation

4.4.1 Partial fingerprint Localization

The ray tracing simulation setup detailed in Section 3.5 is used to evaluate the privacy protection of the multipath-based camouflage privacy protection scheme [1]. It should be noted that it is not possible to directly compare the performance of the multipath camouflage privacy protection algorithms with other privacy protection schemes in a ray tracing simulation. K-anonymity and encryption privacy protection schemes are implemented in the communication protocol with the sever and are not related to the propagation of electromagnetic waves in the environment. In addition, a metric for objectively evaluating privacy for all schemes would need to be designed. This is very difficult because as different mechanisms and physical quantities guarantee the user privacy for each scheme. In k-anonymity approaches the privacy of the user depends on the number of other users in the system. In encoding-based schemes full user privacy is guaranteed, the server cannot know the user's location, however, the required computational power and data is very high. Comparing the computational requirements of an encoding and a camouflage-based scheme is hard because in a camouflage-based scheme the user sets the computational requirements and degree of privacy. The camouflage based privacy scheme was designed not to provide better privacy protection under the same constraints as encryption and k-anonymity, but to avoid the constraints of those schemes while still protecting the user's privacy. Camouflage-based privacy protection works if there are only a few users in the system, the communication protocol does not need to be altered, and the computational requirements can be kept low. At the same time, using multipath-based fingerprints makes it possible to generate credible partial camouflage fingerprints. Camouflage-based RSSI and CSI algorithms are also not simulated or implemented. This is because a ray-tracing approach is not suitable to precisely calculate the CSI and RSSI fingerprints.

The first simulated experiment checks whether the partial multipath fingerprints are sufficient for localization with the MCA algorithm. Figure 4.11 shows the average localization

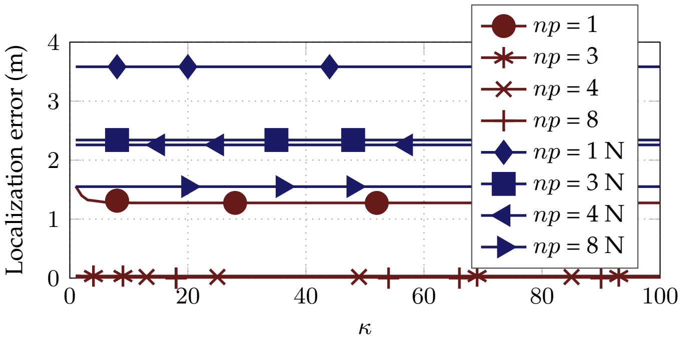


Figure 4.11: Localization accuracy with partial fingerprints. Curves marked with N were generated in the presence of noise. Figure adapted from [1] ©2021 IEEE.

error for different values of the partial fingerprint size np and the number of candidates returned by the ILS for a single partial fingerprint κ. Additive white Gaussian noise (AWGN) with the standard deviation $\sigma^2 = 0.25m^2$ was added to the reference and query multipath components for the curves marked with N. While the localization accuracy expectedly decreases in the presence of noise, for $\kappa > 3$ the set of candidate fingerprints returned by the ILS server contains the fingerprint that would be picked by the localization algorithm. Therefore, $k = 3$ is used in the following simulations.

4.4.2 Privacy Evaluation for Stationary Users

The metrics described in Section 4.3 are used to evaluate the privacy of a stationary user in the proposed approach. Figure 4.12 shows the results of a simulation without noise. Figure 4.13 shows the privacy metrics calculated in the same simulation when additive white Gaussian noise (AWGN) with $\sigma^2 = 0.25m^2$ was added to the query and reference multipath components. All of the figures have the number of partial fingerprints sent by the user as the x-axis. The top sub-figures show the number of candidate fingerprints returned by the ILS. This represents the computational load of the user. The black dashed line plots $\kappa \times N_T$. A reference point can be selected as a candidate for more than one partial fingerprint, therefore the total number of candidates should be less or equal to $\kappa \times N_T$. The two bottom sub-figures show the privacy metrics $H(\rho(\boldsymbol{X_i}))$ and ω, derived in Section 4.3.1. Algorithm 3, that generates random multipath components, is used to generate partial fingerprints for the curves marked with RF. Algorithm 4, that randomly changes the AP IDs was used for the curves marked with RT.

The results show that the degree of privacy increases with the number of partial fingerprints generated by the user N_T. However, for large values of N_T the gain in privacy starts to saturate. This is because more partial fingerprints begin to correspond to the same candidates. The results also show that the number of multipath components in the partial fingerprint np has less effect on privacy. The MCA algorithm described in Chapter 3 is used for comparing the partial fingerprints to the fingerprint map. It only searches for the matching propagation distances. Multipath components in an MDP that were not matched to the reference fingerprint are ignored. However, if an MDP has many multipath components

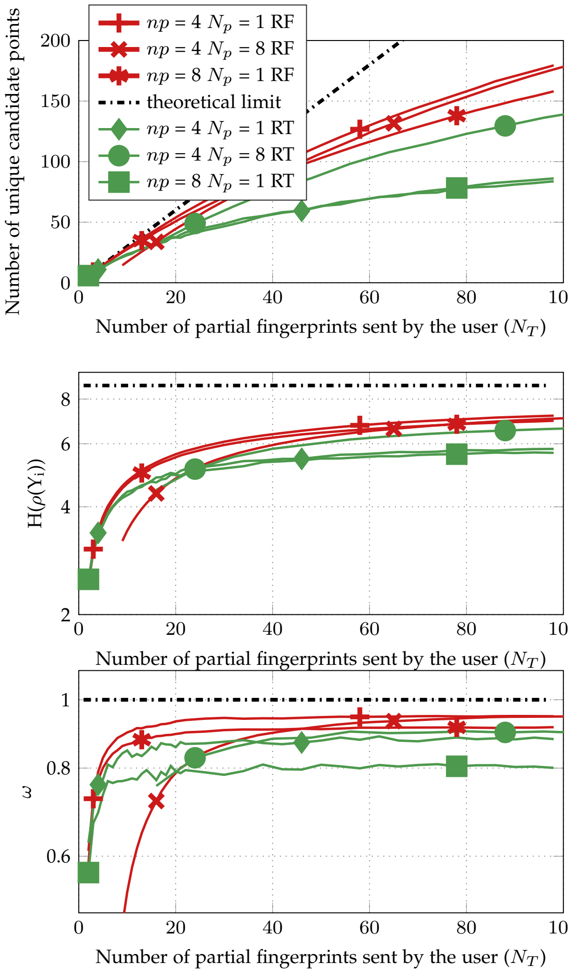


Figure 4.12: Privacy evaluation. Top: number of candidate points received by the user. Middle and bottom: degree of privacy. The camouflage fingerprints for the curves marked with RF were generated using Algorithm 3. Algorithm 4 was used for the curves marked with RT. Figure adapted from [1] ©2021 IEEE.

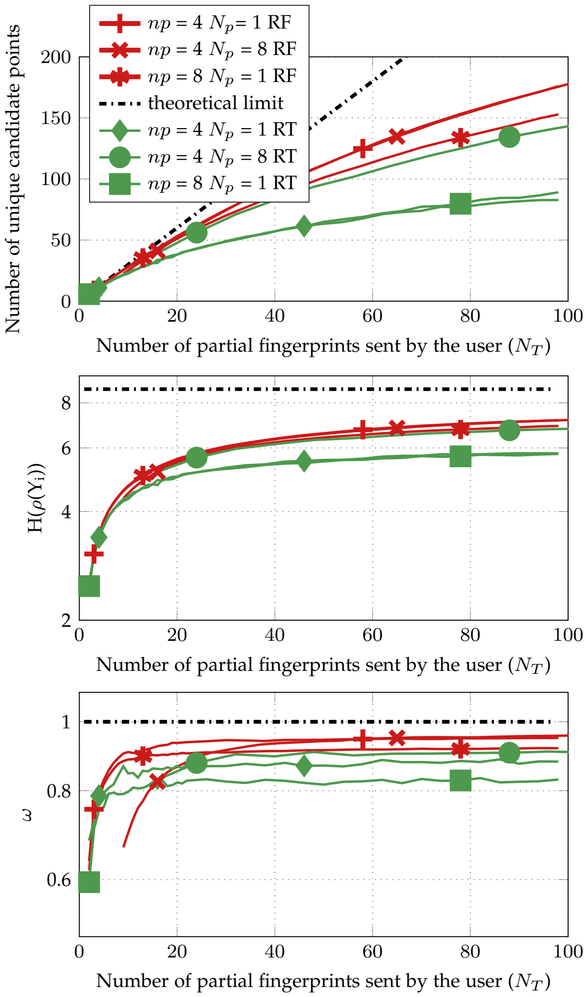


Figure 4.13: Privacy evaluation in the presence of noise. Figure adapted from [1] ©2021 IEEE. Notation identical to Fig. 4.11.

that were not matched it should be considered dissimilar to the map. $\kappa = 3$ was used in the simulation. Results from additional experiments showed that increasing the number of candidates returned for one query κ increased the degree of privacy, however, the number of candidate points increased too much. A better localization performance is achieved for a higher value of the number of real partial fingerprints N_p. It can be seen from Figs. 4.12 and 4.13 that N_p has a large effect on privacy when Algorithm 4 is used. Algorithm 4 randomly changes the AP IDs of the multipath components in the *real* partial fingerprints to create the camouflage partial fingerprints. A higher N_p means a higher diversity of multipath components in the camouflage partial fingerprints. Additionally, when a user sends multiple real partial fingerprints to the ILS, the probability that the partial fingerprints contain multipath components corresponding to the background geometry and the robustness

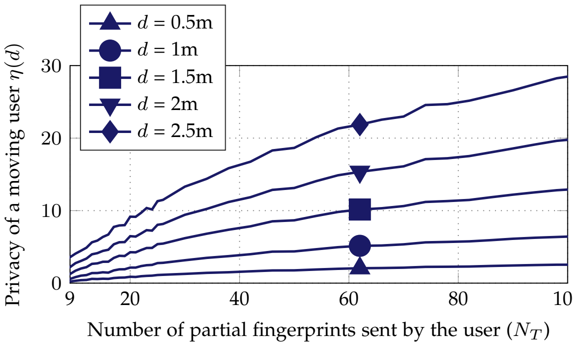


Figure 4.14: Degree of privacy of a moving user as a function of the number of partial fingerprints sent by the user. The camouflage fingerprints were generated using Algorithm 3. Figure adapted from [1] ©2021 IEEE.

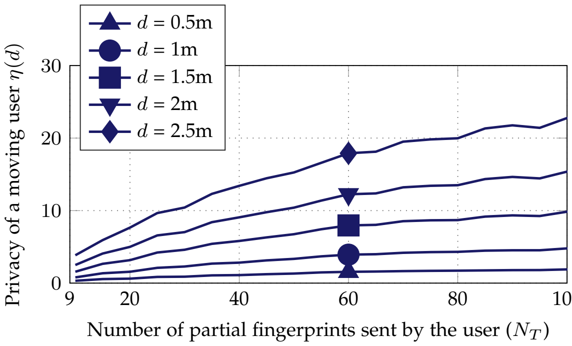


Figure 4.15: Degree of privacy of a moving user as a function of the number of partial fingerprints sent by the user. The camouflage fingerprints were generated using Algorithm 4. Figure adapted from [1] ©2021 IEEE.

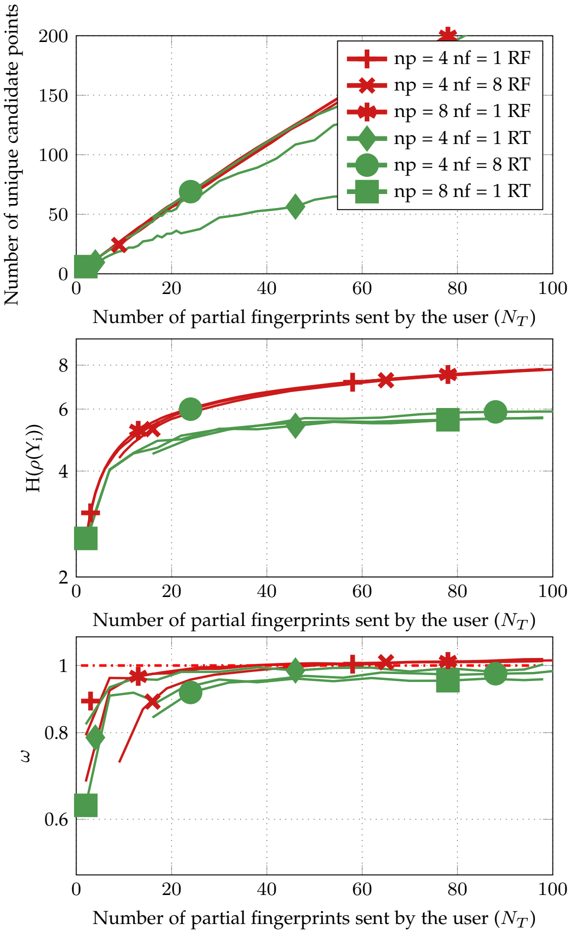


Figure 4.16: Privacy evaluation with measurement data. Top: number of candidate points received by the user [1]. Notation identical to Figs. 4.12 and 4.13. Figure adapted from [1] ©2021 IEEE.

of the localization against changes in the environment increase. The simulation results also show that even though the addition of noise decreases the localization accuracy, the degree of privacy protection is mostly unchanged.

4.4.3 Privacy of a Moving User

The metric $\eta(d)$, proposed in Section 4.3.4 is used to estimate the privacy of a moving user. Noise is not added to the simulation. np is set to 4 and N_p is set to 5. Algorithm 3 is used to generate camouflage partial fingerprints from random multipath values in Figure 4.14. Algorithm 4 is used to generate camouflage fingerprints by randomly changing the AP IDs of *real* partial fingerprints in Figure 4.15. The privacy metric $\eta(d)$ was calculated for several values of d. The results show that better privacy protection was achieved by Algorithm 3.

If the user knows parts of the fingerprint database, or learns it over time, he can generate camouflage fingerprints that mimic natural movement patterns and increase the degree of privacy protection. A moving user can also increase his privacy by submitting localization requests more rarely.

4.5 Measurement Results

The multipath camouflage privacy protection scheme is also validated on the measurement data-set described in Section 3.6. Figure 4.16 shows the privacy metrics evaluated on the static data. The results validate the effectiveness of the proposed privacy protection scheme. Again the generation of partial fingerprints using random multipath components according to Algorithm 3 results in a higher degree of user privacy.

4.6 Conclusion

This chapter demonstrates that the multipath-based fingerprint format is ideally suited for implementing a camouflage-based privacy protection scheme. Partial fingerprints can be created out subsets of the measured multipath components. Believable fake partial fingerprints are easier to create as, unlike a full fingerprint, a partial fingerprint is similar to multiple entries in the map. Camouflage CIR, CFR or CSI and even full multipath fingerprints might not correspond to the geometry and be identified by the server as fake. The server then calculates a number of candidate locations or guesses where the user might be located and sends the candidate locations and candidate fingerprints to the user. The server does not know where exactly the user is. The user locally determines his or her location from the candidate locations and candidate fingerprints received from the server. The degree of privacy protection and the computational complexity can be set by the user when he or she decides on the number of camouflage fingerprints that will be sent to the server.

Several metrics that can be used to estimate the degree of privacy of a stationary and moving users are also proposed in this chapter. The multipath-based privacy protection scheme is evaluated using a ray-tracing simulation and measured data.

Chapter 5

Fast Fingerprinting

A significant delay between when a query is sent to the ILS and when a location is received by the user, can lead to the human user becoming frustrated with the ILS. It can also lead to instability of a system the components of which communicate over a network [249]. One example of this would be vehicles that are traveling at high speeds and are using the ILS to navigate themselves. The position received by a vehicle will not correspond to its actual location but to the location where the vehicle was when the query was sent. Therefore, the larger the computation and communication delays in the ILS are, the larger the difference between a vehicle's actual and received location will be.

A fingerprinting-based localization algorithm determines which entry in the fingerprint map is most similar to the user's query. If the ILS compares the query measured by the user to every reference fingerprint in the map, its computational complexity will increase linearly with the size of the map. As mentioned in Section 2.5.4.1, a trade-off exists between the accuracy of an ILS and the number of reference fingerprints stored in the map. For example, the work in [28] achieves centimeter level localization accuracy. However, it uses a fingerprint database with a $5cm$ grid or 400 fingerprints per m^2. The ILS proposed and validated in Section 3.6.1 requires a $3cm$ fingerprint density to achieve a $6cm$ average localization error. This means that having a dense fingerprint database is advantageous. Interpolation [250],[251] and crowd-sourcing [103] algorithms can be used to create such databases at low cost. However, for a large fingerprint database to be useful, the ILS needs to be able to search it while maintaining a small enough computation delay.

This chapter describes a run-time optimization scheme presented in the authors original publication [2]. The optimization scheme is designed for the MCA algorithm and the MDP fingerprint structure detailed in Section 3.3. It guarantees that the best match to the query data is found, while the similarity metrics between the query and only a subset of the reference fingerprints are calculated. The computational complexity of the MCA localization algorithm is reduced by sorting the fingerprint database and searching by multipath components. While many of the existing optimized fingerprinting algorithms down-sample the fingerprint database before comparing its entries to the user's query, in the proposed scheme the entire database is searched for the best match to the query. However, because the database is sorted, the similarity metric is not calculated for reference fingerprints that do not contain multipath components that are similar to those in the query. Several approximations to the proposed scheme are also described in this chapter. These optimizations allow to further decrease the computation time by reducing the accuracy of the ILS. The simulation results presented in this chapter and in [2] show that using the proposed approximations to search a dense fingerprint database is more effective than applying traditional fingerprinting on a down-sampled database. Another advantage of the proposed optimization approach is that a rough location estimate can be given to the user and refined during the database search until the real location is computed.

5.1 Complexity of State-of-the-Art Fingerprinting Schemes

As mentioned in Section 2.5, common similarity metrics calculated between query and reference fingerprints include the Euclidean distance [10], correlation, time reversed correlation [28], and maximum likelihood estimation [29]. In order to find out which reference fingerprint is most similar to the query, such a metric needs to be calculated between each reference fingerprint and the query. This means that the computational complexity of such an algorithm is at least $O(N)$, where N is the number of reference fingerprints in the map.

The computational complexities of several fingerprinting-based localization approaches are estimated in Tables 5.1 - 5.3 [2]. The localization algorithms were selected randomly, while insuring that different algorithm types are included in the comparison. The algorithms in the tables do not implement run-time optimization. The standard O-notation is used [255]. f/m^2 is used to denote fingerprints per square meter. The computational complexity of the fingerprinting algorithms in Tables 5.1 - 5.3 increases linearly with the number of reference fingerprints N, as they calculate the similarity metric for each reference fingerprint. The complexity of an algorithm using maximum likelihood estimation is up to $O(N^2)$ as all of the reference fingerprints are used to calculate the similarity metric for one reference fingerprint. Deep learning and machine learning-based approaches are not included in the Tables. However, for a neural network, a localization step can be performed in constant time, depending only on the size and depth of the network. The network training time will depend on the number of reference fingerprints.

A number of optimization approaches search the fingerprint map by dividing it into regions or clusters, finding the cluster the user is likely located in and further subdividing and searching the chosen cluster until the location is computed [256]. In [257] the query is only compared to the database entries recorded from the same APs as the query. In [257] an search algorithm with an $O(log(N))$ computational complexity is presented. However, this algorithm requires that the number of reference points corresponding to each AP and the number of APs visible at each reference point are both less than 20. Otherwise, the computational complexity of the approach also becomes $O(N)$. In [258], the lengths of the fingerprints are reduced. While this makes an individual fingerprint comparison faster, the complexity remains $O(N)$.

Tables 5.1 - 5.3 also show that the computational complexity of the algorithms increases at least linearly and sometimes even cubically with the number of APs and the size of the fingerprints. In most ILS, both the number of APs is fixed within a given area and the size of the fingerprint vectors is also constant. However, as mentioned before, the number of reference

Authors	Published	Year	Measurement	Reference fingerprint density	Evaluation, fingerprint similarity	K	Complexity of one fingerprint comparison	Number of fingerprint comparisons	Reported average accuracy
RSSI-based fingerprinting schemes									
Seco et al. [10]	*IEEE Symposium on Intelligent Signal Processing*	2009	RSSI single measurement	-	Euclidean distance between RSSI vectors	1	$O(M)$	N	-
Li et al. [252]	*Globecom*	2014	average RSSI from each AP	0.29 f/m^2	$\sum_{AP}$ (difference between the RSSI values scaled by pre-trained weights $w(RSSI))^2$	1	$O(M\times$ number of RSSI intervals used to calculate weights)	N	$1.93m$
Shu et al. [62]	*IEEE Transactions on Industrial Electronics*	2016	IMU data + nr RSSI measurements for each AP	-	particle filter + gradient between two consecutive WiFi scans subtracted from the map	number of measurements collected at one location	$O(M\times$ number of particles in the particle filter)	$\sim N$	80% $<$ 3.6m, dynamic environment <5.6m
Jung et al. [253]	*IEEE Transactions on Intelligent Transportation Systems*	2017	RSSI multiple measurements, fluctuations between pairs of RSSI values	70 locations	comparison of RSSI fluctuation scores between RSSI pairs	number of fingerprint samples	$O(M \times$ $max(K^2$, (number of discrete values used to represent the RSSI$)^2))$	N	$\approx$ $3.6m$

Table 5.1: Complexity estimation of existing RSSI-based localization schemes. Table adapted from [2] ©2018 IEEE.

Authors	Published	Year	Measurement	Reference fingerprint density	Evaluation, fingerprint similarity	K	Complexity of one fingerprint comparison	Number of fingerprint comparisons	Reported average accuracy
CSI and CFR - based schemes									
Wu et al. [29]	*IEEE transactions on parallel and distributed systems*	2013	CSI	-	maximum likelihood + Pearson correlation	number of OFDM subcarriers	$O(N \times M \times K)$	N	$1.7m$ (corridor)
Chapre et al. [27]	*Conference on Local Computer Networks*	2014	MIMO CSI measurement + averaging	0.42 f/m^2	maximum likelihood	number of OFDM subcarriers -1	$O(N \times M \times K)$	N	$1.05m$
					Euclidean distance		$O(M \times K)$	N	$0.85m$
Chen et al. [28]	*IEEE Internet of Things Journal*	2017	CFR estimated for each OFDM subcarrier and frequency channel	≈ 400 f/m^2	time-reversal resonating strength calculated in the frequency domain	number of frequency channels	$O(M \times K)$	N	$1-2cm$
Song et al. [254]	*IEEE Internet of Things Journal*	2017	CSI for each AP	0.23 f/m^2	triangulation + Euclidean distance between fingerprint vectors, the dimensionality of which was reduced using multidimensional scaling analysis	one average CSI value	$O(M^3)$	N	$1.2m$

Table 5.2: Complexity estimation of existing CIR, CFR and CSI -based localization schemes. Table adapted from [2] ©2018 IEEE.

fingerprints in the database can easily be increased though measurement, crowd-sourcing and interpolation [250],[251]. Therefore, this chapter focuses on reducing the complexity of

Authors	Published	Year	Measurement	Reference fingerprint density	Evaluation, fingerprint similarity	K	Complexity of one fingerprint comparison	Number of fingerprint comparisons	Reported average accuracy
Multipath-based schemes									
Kupershtein et al. [58]	*IEEE Transactions on Signal Processing*	2013	sampled received signal	$\approx 1 f/m^2$	projections of the covariance matrix of a signal onto the signal subspaces of all of the reference points compared	(number of receive antennas × number of signal samples)^2	$O(M \times K^3)$	N	$\approx 1m$
Zayets et al. [3]	*IPIN*	2017	CSI measurement + multipath estimation	19.26 f/m^2	comparison of individual multipaths	number of multipaths	$O(M \times K \times \log(K))$	N	-

Table 5.3: Complexity estimation of existing multipath-based localization schemes. Table adapted from [2] ©2018 IEEE.

the proposed MCA algorithm with respect to the size of the fingerprint map.

5.2 Proposed Run-Time Optimization Approach

The aim of the proposed optimized algorithm [2] is to estimate the location of the user within a certain accuracy while comparing the query to only a subset of reference fingerprints.

5.2.1 Database Structure

MDP fingerprints can be arranged in a sorted database as illustrated in Fig. 5.1:

- An array of bins $\boldsymbol{B} = \{\boldsymbol{bin}(j, x)\}$ with $j = 1, ..., M$, $x = 1, ..., d_{max}/\varepsilon$ is stored. Each $\boldsymbol{bin}(j, x)$ is a sorted data structure that contains all of the reference multipath components, the values of which are in the interval $((x-1)\varepsilon, x\varepsilon]$ and which correspond to the AP $\boldsymbol{T}_j$. ε is the similarity threshold of the MCA algorithm according to Eq. 3.11. d_{max} is the largest value of a multipath component that can be stored in the database.

- Each multipath component in the sorted database contains a link to the fingerprint it belongs to.

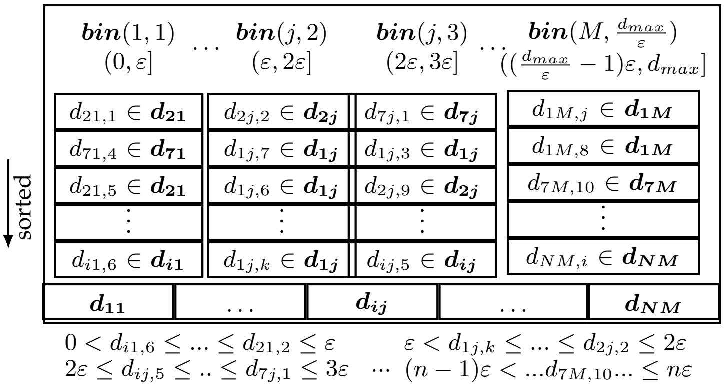

$\boldsymbol{bin}(1,1)$ $(0,\varepsilon]$
...
$\boldsymbol{bin}(j,2)$ $(\varepsilon,2\varepsilon]$
$\boldsymbol{bin}(j,3)$ $(2\varepsilon,3\varepsilon]$
...
$\boldsymbol{bin}(M,\frac{d_{max}}{\varepsilon})$ $((\frac{d_{max}}{\varepsilon}-1)\varepsilon,d_{max}]$
sorted
$d_{21,1}\in \boldsymbol{d_{21}}$
$d_{71,4}\in \boldsymbol{d_{71}}$
$d_{21,5}\in \boldsymbol{d_{21}}$
$d_{i1,6}\in \boldsymbol{d_{i1}}$
$d_{2j,2}\in \boldsymbol{d_{2j}}$
$d_{1j,7}\in \boldsymbol{d_{1j}}$
$d_{1j,6}\in \boldsymbol{d_{1j}}$
$d_{1j,k}\in \boldsymbol{d_{1j}}$
$d_{7j,1}\in \boldsymbol{d_{7j}}$
$d_{1j,3}\in \boldsymbol{d_{1j}}$
$d_{2j,9}\in \boldsymbol{d_{2j}}$
$d_{ij,5}\in \boldsymbol{d_{ij}}$
$d_{1M,j}\in \boldsymbol{d_{1M}}$
$d_{1M,8}\in \boldsymbol{d_{1M}}$
$d_{7M,10}\in \boldsymbol{d_{7M}}$
$d_{NM,i}\in \boldsymbol{d_{NM}}$
$\boldsymbol{d_{11}}$
...
$\boldsymbol{d_{ij}}$
...
$\boldsymbol{d_{NM}}$
$0<d_{i1,6}\le ... \le d_{21,2}\le\varepsilon$
$\varepsilon<d_{1j,k}\le ... \le d_{2j,2}\le 2\varepsilon$
$2\varepsilon\le d_{ij,5}\le .. \le d_{7j,1}\le 3\varepsilon$
$\cdots$ $(n-1)\varepsilon< ...d_{7M,10}... \le n\varepsilon$

Figure 5.1: Proposed structure of the fingerprint database. Figure adapted from [2] ©2018 IEEE.

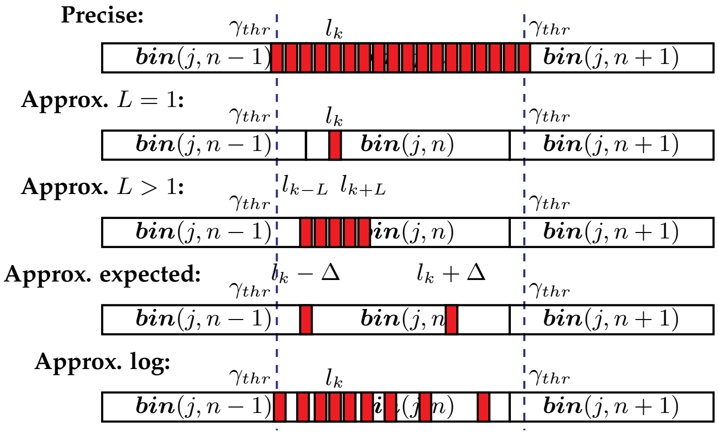


Figure 5.2: Multipath component search. Figure adapted from [2] ©2018 IEEE.

5.2.2 Proposed Fingerprint Matching Procedure

For each multipath component $d^q_{j,k}\in\mathbf{D}^q$ in the query, the similarity metric values for a candidate set of fingerprints are iteratively updated. The location estimate $\hat{\boldsymbol{X}}$ and the candidate set are updated as well. In the following, the iteration of the algorithm, which corresponds to the multipath component $d^q_{j,k}$ from transmitter $\boldsymbol{T}_j$, is referred to as the (j,k)th iteration of the optimized MCA algorithm. A fingerprinting algorithm searches the database to determine which reference fingerprint has the highest similarity metric value with the query. Therefore, the algorithm does not need to calculate the similarity metric $\gamma(\mathbf{D}^q,\mathbf{D}_i)$ for the reference point $\boldsymbol{X}_i$ if it knows beforehand that it will be smaller than the largest similarity metric calculated so far, that is $\gamma(\mathbf{D}^q,\mathbf{D}_i)<\gamma_{max}$. According to Eq. 3.11, a query multipath component $d^q_{j,k}$ can only be matched with reference multipath components in the range $(d^q_{j,k}-\varepsilon, d^q_{j,k}+\varepsilon)$. Let us assume that $n\varepsilon<d^q_{j,k}<(n+1)\varepsilon$ at algorithm iteration (j,k). All of the multipath components, the difference between which and $d^q_{j,k}$ is less than ϵ, and which correspondingly can be included in the similarity metric, are in the bins $\boldsymbol{bin}(j,n-1)$,

$\boldsymbol{bin}(j,n)$, $\boldsymbol{bin}(n+1)$. Therefore, at this step of the algorithm there is no need to search any other bins in the database. $\boldsymbol{f}$ denotes the set of candidate fingerprints and $\boldsymbol{\gamma}_{(j,k)}$ contains the corresponding similarity metrics calculated at algorithm iteration (j,k). At each subsequent iteration of the MCA algorithm $(j,k+1)$ or $(j+1,1)$, the metrics stored in $\boldsymbol{\gamma}_{(j,k+1)}/\boldsymbol{\gamma}_{(j+1,1)}$ are first updated as:

$$\begin{aligned} \mathbf{D}^q_{(j,k)} &= \{\boldsymbol{d}^q_1, ..., \boldsymbol{d}^q_{j-1}, [d^q_{j,1}, ..., d^q_{jk}]\}, \\ \gamma_{i,jk} \in \boldsymbol{\gamma}_{(j,k)} &= \gamma(\mathbf{D}^q_{(j,k)}, \mathbf{D}_i) = \gamma(\mathbf{D}^q_{(j,k-1)}) + \gamma^{'}_{i,jk}, \\ \gamma^{'}_{i,jk} &= \gamma(\{[d^q_{j,k}]\}, \mathbf{D}_i) = (\varepsilon - \mid d^q_{j,k} - l_k \mid)^2. \end{aligned} \tag{5.1}$$

If there are any new fingerprints within the search area of the query multipath component $d^q_{j,k+1}$ $d^q_{j+1,1}$, those fingerprints are added to the candidate set. The location estimate $\hat{\boldsymbol{X}}_{\boldsymbol{jk}}$ is also updated at each algorithm iteration (j,k) according to the maximum partial similarity value $\gamma_{i,jk} \in \boldsymbol{\gamma}_{(j,k)}$ calculated so far.

As the values in the database bins are sorted, it is easy to obtain a sorted search array $\boldsymbol{l}$ for the query multipath components $d^q_{j,k}$ at the algorithm step (j,k). The array $\boldsymbol{l}$ contains multipath components with values in the range $((n-1)\varepsilon, (n+1)\varepsilon]$. Since the array is sorted, it is easier to search. $l_k \in \boldsymbol{l}$ denotes the best match to the query such that $\mid d^q_{j,k} - l_k \mid \rightarrow min$ and $\mid d^q_{j,k} - l_k \mid < \varepsilon$. After the best match l_k to the query is found, the term $\gamma^{'}_{i,jk}$ is calculated for its neighboring multipath components in $\boldsymbol{l}$. The similarity metric value $\gamma_{i,jk}$ corresponding to the reference fingerprint $\mathbf{D}_i$ does not need to be calculated if it can be shown beforehand that it will be less than the largest similarity metric $\gamma_{max,jk}$ calculated so far. Therefore, the threshold value γ_{thr} is derived such that the partial similarity metric values $\gamma' ij, k$ of the neighbors of the best match to $d^q_{j,k}$ in $\boldsymbol{l}$ don't need to be calculated if they are larger than γ_{thr}.

$$\begin{aligned} (\varepsilon - \mid d^q_{j,k} - l_i \mid)^2 &= \gamma^{'}_{i,jk} \leq \gamma_{thr} \Rightarrow \gamma_{i,jk} \leq \gamma_{max}, \\ \gamma_{i,jk} &= \gamma^{'}_{i,jk} + \gamma^{'}_{i,j(k-1)} + ... + \gamma^{'}_{i,(j-1)1} + ... + \gamma^{'}_{i,11}, \\ \gamma^{'}_{i,jk} &\leq \gamma^{'}_{min,jk}, \; ..., \; \gamma^{'}_{i,11} \leq \gamma^{'}_{min,11}, \\ \gamma_{i,jk} &\leq \gamma^{'}_{i,jk} + \gamma^{'}_{min,j(k-1)} + ... + \gamma^{'}_{min,11}, \\ \gamma_{max} &\geq \gamma^{'}_{i,jk} + \gamma^{'}_{min,j(k-1)} + ... + \gamma^{'}_{min,11} \Rightarrow \gamma_{max} \geq \gamma_{i,jk}, \\ \gamma_{thr} &= \gamma_{max} - \sum_{l=1}^{k-1} \gamma^{'}_{min,jl} - \sum_{q=1}^{j-1} \sum_{\forall l} \gamma^{'}_{min,ql}, \end{aligned} \tag{5.2}$$

where $\gamma^{'}_{min,jk}$ is the partial similarity metric calculated for the reference multipath component most similar to the query multipath component at the (j,k)-th iteration of the algorithm. As the multipath component is most similar to the query, $\gamma^{'}_{min,jk}$ will actually be the largest partial similarity metric value calculated in one iteration. However, the notation min is kept for consistency with [2]. Each algorithm iteration starts by updating the similarity metric values in $\boldsymbol{\gamma}_{(j,k)}$ that correspond to the candidate fingerprints in $\boldsymbol{f}$, the maximum similarity metric value γ_{max} and γ_{thr}. It should be noted that line 3 of Eq. 5.2 holds because the multipath components in the fingerprint database are sorted. Therefore, the calculated $\gamma^{'}_{i,jk}$ monotonically decreases as the algorithm moves away from l_k to the start and end of the array $\boldsymbol{l}$. If the calculated similarity metric becomes smaller than the threshold $\gamma'_{i,jk} < \gamma_{thr}$, a new iteration of the algorithm can be started. If $\gamma'_{i,jk} \geq \gamma_{thr} \cap \mathbf{D_i} \notin \boldsymbol{f}$ then the full similarity metric $\gamma(\mathbf{D}^q_{(j,k)}, \mathbf{D}_i)$ is calculated and the reference fingerprint $\mathbf{D}_i$ is added to the candidate set $\boldsymbol{f}$.

Algorithm 5: Fast Fingerprint Matching [2] ©2018 IEEE.

```
Data: Reference multipath components $\mathbf{D_i}, i = 1...N$, reference database $\mathbf{B}$, query
      fingerprint $\mathbf{D^q}$
Result: Location estimate $\hat{\boldsymbol{X}} = \hat{\boldsymbol{X}}_{(M,K^q_M)}$
for $\forall$ transmitter $\boldsymbol{T_j}$ do
    for $k = 1...K^q_j$ do
        find $x : x\varepsilon < d^q_{j,k} \leq (x+1)\varepsilon$
        $\boldsymbol{l} \leftarrow [\boldsymbol{bin}(j,(x-1)), \boldsymbol{bin}(j,x), \boldsymbol{bin}(j,(x+1))]$
        $l_k \in \boldsymbol{l} : \mid d^q_{j,k} - l_k \mid < \varepsilon \cap \mid d^q_{j,k} - l_k \mid \rightarrow min$
        $l_k \in \boldsymbol{d_{ij}} \Rightarrow \boldsymbol{f}_{(j,k)} \leftarrow [\boldsymbol{f}_{(j,k)}, \mathbf{D_i}]$
        Proposed Approach: search neighbors of $l_k$
        while $\gamma'_{l_{k+\Delta}} \geq \gamma_{thr}$ do
            $\boldsymbol{f} \leftarrow [\boldsymbol{f}, \mathbf{D_{l_{i+\Delta}}}]$
            $\Delta \leftarrow \pm 1$, update $\gamma_{max}$, $\gamma_{thr}$, $\gamma'_{min,jk}$
        Approximations: search $\boldsymbol{l}$ according to Fig. 5.2
        $\hat{\boldsymbol{X}}_{(j,k)} \leftarrow \boldsymbol{Y}_{argmin_i \gamma(\mathbf{D^q}_{(j,k)}, \mathbf{D_i} \in \boldsymbol{f})}$
```

The optimized MCA algorithm is guaranteed to find the reference fingerprint with largest similarity metric value. The complexity of the proposed approach is:

$$O(M^q \times K^q \times log(\frac{d_{max}}{\varepsilon}) \times (log(\frac{K \times N \times \varepsilon}{d_{max}}) + E[N_{th}])), \tag{5.3}$$

where M^q is the number of APs from which query multipath components were received, d_{max} is the largest multipath component in the map such that $\frac{d_{max}}{\epsilon}$ is the number of bins in the fingerprint database, $N_{th} = \frac{|\boldsymbol{f}|}{M \times K^q}$ is the number of partial similarity metrics that are calculated at one algorithm iteration. $O(log(\frac{d_{max}}{\varepsilon}))$ is the complexity of finding the correct bins and constructing search array $\boldsymbol{l}$. $O(log(\frac{K \times N \times \varepsilon}{d_{max}}))$ is the complexity of finding the multipath component most similar to the query l_k within $\boldsymbol{l}$ given that $\boldsymbol{l}$ is sorted.

The proposed approach can also be used with techniques such as k-nearest neighbors that calculate an average of k best reference locations. k largest similarity metric values would need to be selected at each algorithm iteration. If the similarity metric needs to be averaged over neighboring reference locations, a number of largest similarity metric values can first be obtained by the optimized MCA algorithm. The similarity metrics can then be calculated for their neighboring points. Since the proposed optimization scheme stored the data from different APs separately, it is very advantageous for large buildings, where only a small subset of the APs is visible at each location. In that case, the fingerprints with data from other APs will be ignored by the algorithm and will not effect the computational complexity. The proposed optimization approaches can also be combines with clustering and down-sampling [257] and [256].

5.2.3 Approximate Fingerprint Matching

The run-time of the algorithm can be further reduced if it is allowed to find a reference fingerprint, which is similar, but not necessarily the most similar to the query. The work in [2] describes three approximation schemes for fingerprint matching. All three approximation algorithms create the search vector $\boldsymbol{l}$ in the same way as the optimized MCA algorithm. However, instead of calculating partial similarity metrics for all the multipath components in the search area, only a few multipath components are considered. This is illustrated by Fig. 5.2. Same as for the original and optimized MCA algorithm, the location estimate is the reference point which corresponds to the highest computed similarity value.

- **Best Multipath Component Match 'Approx. $L = \_$'**

  In the first approximation algorithm, instead of the full search area, a preset number L of multipath components around the best march l_k are considered. As $\boldsymbol{l}$ is sorted, those multipath components correspond to the L highest partial similarity metric values at that algorithm iteration. The complexity of the scheme becomes:

$$O(M^q \times K \times log(\frac{d_{max}}{\varepsilon}) \times (log(\frac{K \times N \times \varepsilon}{d_{max}}) + L)). \tag{5.4}$$

- **Expected Multipath Component Value 'Approx. expected'**

  It is unlikely that the user's location is the same as a reference point. Therefore, the multipath components of a reference and query fingerprints are unlikely to be identical. This algorithm approximation adds the reference points closest to $d^q_{j,k} \pm \Delta_{map}$ to the candidate set $\boldsymbol{f}$, where Δ_{map} is the average distance from a query location to the closest reference point. The complexity becomes:

$$O(M^q \times K \times log(\frac{d_{max}}{\varepsilon}) \times log(\frac{K \times N \times \varepsilon}{d_{max}})). \tag{5.5}$$

- **Logarithmic Search 'Approx log search'**

  Instead of searching $\boldsymbol{l}$ the optimized MCA algorithm starts at the best matching multipath component l_k. However, instead of going through the neighbors of l_k one by one, this algorithm approximation goes through the search area with exponentially increasing steps until $\gamma^{'}_{i,k} < \gamma_{thr}$ is found. Thus some of the multipath components in the search area are skipped. The complexity of the fingerprint matching scheme becomes:

$$O(M^q \times K \times log(\frac{d_{max}}{\varepsilon}) \times 2 \times log(\frac{K \times N \times \varepsilon}{d_{max}}). \tag{5.6}$$

5.3 Evaluation in a Ray-Tracing Simulation

The simulation setup used to evaluate the optimized fingerprint-matching scheme is detailed in Section 3.5 in Chapter 3. AWGN noise with variance $0.25m^2$ was added to the query multipath components. In this way, the query multipath components differ from the reference ones. The reference fingerprints were down-sampled to and the approach was evaluated for different reference fingerprint densities. The MCA algorithm described in Section 3.4 and the fast fingerprint matching algorithms described in this chapter were run for each fingerprint density. An average localization error was calculated over the query points. The average number of the similarity metric calculations, representing the complexity of the algorithm was also calculated. The simulation data was originally presented in [2].

5.3.1 Performance Evaluation

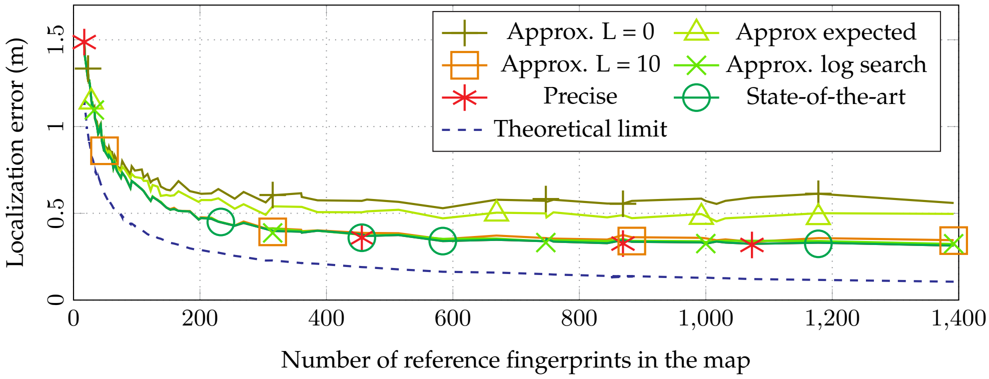


Figure 5.3: Average localization error of the algorithms. Figure adapted from [2] ©2018 IEEE.

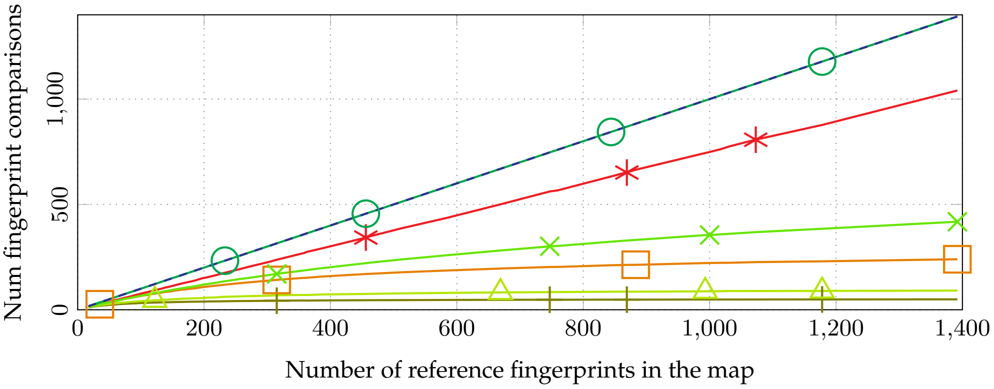


Figure 5.4: Number of fingerprint comparisons of the algorithms. Figure adapted from [2] ©2018 IEEE.

Figure 5.3 shows the average localization error obtained at different fingerprint map densities by each optimized MCA algorithm illustrated in Fig. 5.2. 'Precise' refers to the optimized MCA algorithm from Section 5.2.2 that guarantees that the reference fingerprint with the highest similarity metric is found. 'Approx. L = 0' and 'Approx. L = 10' refer to the approximation approach from Section 5.2.3 with the number of additional partial similarity metric calculations per algorithm iteration L equal correspondingly to 1 and 10. 'Approx. expected' is the second approximation algorithm which adds a shift to the query multipath components and only performs one partial similarity metric calculation per algorithm iteration. The algorithm 'Approx. log' iterates through the multipath components in the search

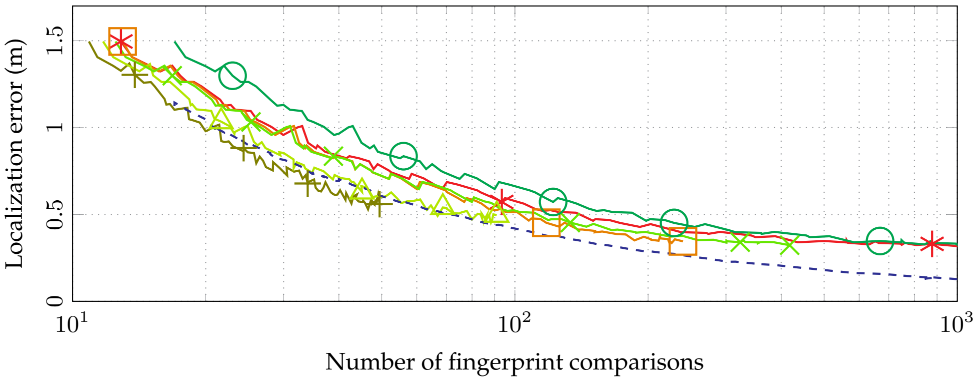


Figure 5.5: Efficiency of the algorithms. Figure adapted from [2] ©2018 IEEE.

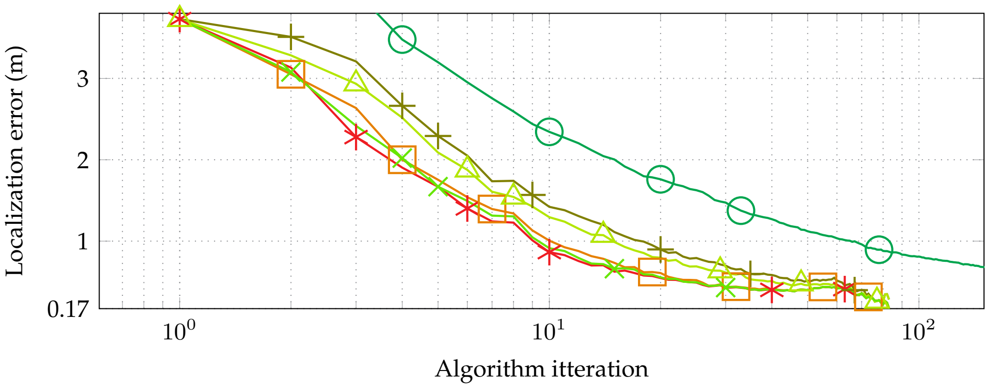


Figure 5.6: Localization error at each algorithm iteration. Figure adapted from [2] ©2018 IEEE.

vector with an exponentially increasing step until the calculated partial similarity metric becomes smaller than γ_{thr}. The original MCA algorithm is marked as 'State-of-the-art'. The 'Theoretical limit' curves correspond to the case when the similarity metric is calculated for each reference fingerprint and the average localization error is equal to the average distance from a query location to the closest reference point. This notation applies to all figures in this chapter. The results of the original MCA algorithm differ from the theoretical limit due to the noise added to the query fingerprints. It can be seen from Figure 5.3 that the localization errors of 'Precise' and 'Approx. L = 10' are almost indistinguishable from the state-of-the-art, which is the original MCA algorithm. The figure also shows that the accuracy of all of the approximation approaches, especially 'Approx. L = 0' and 'Approx. L = 10' starts to saturate at high map densities. It should be noted, that when noise is present in the multipath components increasing the map density may not result in as large an accuracy increase compared to the noiseless case.

The number of fingerprint comparisons and correspondingly, the complexity of the original and optimized MCA algorithms, are shown in Figure 5.4. 'Precise' performs around 25% less comparisons than the original MCA algorithm. Its complexity, however, still grows linearly with the map size. On the other hand, the number of similarity metric calculations performed by the approximation algorithms saturates for large map sizes. It should be noted that the complexity of all algorithms is at least $O(log(N))$ since l_k needs to be found in the sorted vector $\boldsymbol{l}$. Figure 5.5 plots the localization error as a function of the number of similarity metric calculations performed by each algorithm. The optimized MCA algorithm and all of the approximation algorithms are able to beat the state-of-the art by up to 34%. The highest efficiency is shown by 'Approx. L = 0' and 'Approx. expected'. Those are the algorithms that only calculate one similarity metric value per algorithm iteration. However, the algorithms are not able to achieve the lowest localization error.

Figure 5.6 shows the average errors of the location estimates calculated at each iteration of the algorithms. The simulation was run with 456 reference fingerprints. One iteration of the original MCA algorithm is considered to be one comparison of the query with a randomly selected reference fingerprint. The plot shows that an accurate location estimate is produced before the optimized MCA algorithms finish, and that that location estimate is improved with subsequent iterations.

5.4 Chapter Summary

This chapter presented an approach to reduce the computational complexity of the MCA algorithm. The complexity is reduced by using a sorted multipath component database. Several approximations of the optimized approach, which trade-off localization accuracy and complexity, are also presented. Given a fixed number of similarity metric calculations, the localization error of the proposed approach is up to 34% lower than that of the original MCA algorithm. It should be noted that if the number of fingerprint comparisons is restricted, the original MCA algorithm has to be used with a down-sampled database. The optimized algorithm can keep using the original database. The proposed optimized MCA algorithm also produces a location estimate at each iteration. This location estimate can also be used to reduce perceived delays in the system. The proposed optimization scheme is possible due to the MCA algorithm and MDP fingerprint structure. If traditional RSSI and CSI fingerprinting is used, the similarity metric would need to be calculated for each reference fingerprint in order to locate the user.

Chapter 6

Virtual Transmitters, Fingerprint Interpolation and Environment Reconstruction

This chapter focuses on the virtual transmitter (VT) model and on its applications. A virtual transmitter is the reflection of a transmitter or AP that is located behind a reflecting wall or obstacle. Virtual transmitters were briefly introduced in Section 3.5, where they were used to calculate the multipath delay profile of the channel at a point in a simulated indoor environment. VTs are used in this chapter to interpolate and extrapolate the fingerprint map and to reconstruct the indoor geometry.

As was mentioned in Section 2.5.4.1, building the fingerprint map can take a lot of time. Since the demand for ILS is the highest in large public buildings, such as airports, shopping malls and hospitals, the area over which the fingerprints need to be collected can be several thousands of m^2. At the same time, there is a trade-off between the density of reference fingerprints and the accuracy of the ILS. For example, in [28] a 5 cm grid is required to achieve a centimeter level localization accuracy. Figure 3.15 in the experimental validation of the proposed MCA scheme in Section 3.6.1 shows the increase of localization error with the decrease of fingerprint density. In order to achieve a localization density of around 6cm, the MCA algorithm proposed in this thesis requires a fingerprint density of around 3cm. If some of the fingerprints are interpolated and extrapolated, the amount of calibration measurements, the required calibration time and the system cost will be significantly reduced. If a map or model of the building is constructed from available calibration measurements, such a model can be used for fingerprint interpolation. It can also be used for the navigation of a robot or human user. As will be shown in this chapter, VTs can be used to extrapolate and interpolate a fingerprint map and to create a model of the environment, and therefore increase the precision and decrease the cost of the localization system.

The following terms are defined in this chapter:

- A *virtual transmitter* (VT), also known in literature as a *virtual anchor* (VA), $\boldsymbol{T}'^{(l)}_{j,k}$ is the reflection of a transmitter (AP) $\boldsymbol{T}_j$ that is located behind the wall or obstacle that reflects the APs signal. As shown in Fig. 6.1, the length of the reflected path over which a signal travels from the AP to the user is the same as the length of the direct line from the user to the VT. The degree of a VT is defined as the number of times the signal from the AP would be reflected before it reaches the user. It is also the number of reflection steps needed to calculate the VT's position in the ray tracing setup detailed in Section 3.5. A first degree VT is denoted as $\boldsymbol{T}'_{j,k}$, second degree VT as $\boldsymbol{T}''_{j,k}$ and an l-th degree VT is denoted as $\boldsymbol{T}'^{(l)}_{j,k}$.

- An interpolated or extrapolated fingerprint is calculated at a *target point*.
- An *interpolated fingerprint* is a fingerprint that is calculated from a set set of measured fingerprints that surround the target point.
- An *extrapolated fingerprint* is a fingerprint that is calculated from a set of reference fingerprints measured at points which do not surround the target point.
- It is assumed that the indoor environments is composed of reflective *polygons*. Each polygon is located in one plane and can have an arbitrary number of corner points. Multiple polygons can be located in one plane and can overlap.

The chapter is organized as follows. The concept of the relevant literature on VTs is presented in Section 6.1. The proposed VT reconstruction scheme is detailed in Section 6.2. The proposed fingerprint interpolation and extrapolation algorithms are detailed and evaluated in Section 6.3. The proposed environment reconstruction scheme is detailed in Section 6.4.

6.1 Virtual Transmitters (VTs)

An example VT is shown in Fig. 6.1. The lengths of the reflected paths from $\boldsymbol{T_j}$ to the point $\boldsymbol{X}$ are equal to the direct distances from $\boldsymbol{X}$ to the virtual transmitters $\boldsymbol{T'_{j,1}}, ..., \boldsymbol{T''_{j,45}}$. Figure 6.1 shows an example of iterative VT calculation over multiple reflections from a known geometry, similar to the work in [218]. If the indoor geometry is not known, VT positions can be estimated from measured data. Angle-of-arrival and multipath measurements are used to estimate the VT positions in [43]. The differences between the multipath profiles measured at several locations are used in [135].

Once obtained, the positions of the VTs and the measured multipath components, equal to the distance from the VTs to the receiver, are used to localize the receiver in [43], [218]. Our previous work [4] interpolates and extrapolates the fingerprint map using VT positions. The indoor geometry is reconstructed from the VT positions in [136], [259] and in our previous work in [5].

It should be noted that VTs can be used to model multipath delays created by signal reflection. However, some multipath components appear due to signal scattering or a combination of scattering and reflection [227]. An expected scatterer (ES) model can be used in this case. The distances and angles from the ESs to the receiver are the same as the lengths of the corresponding propagation paths. However, the position of the ES is determined by both the position of the real scatterer and the position of the receiver at which the multipath

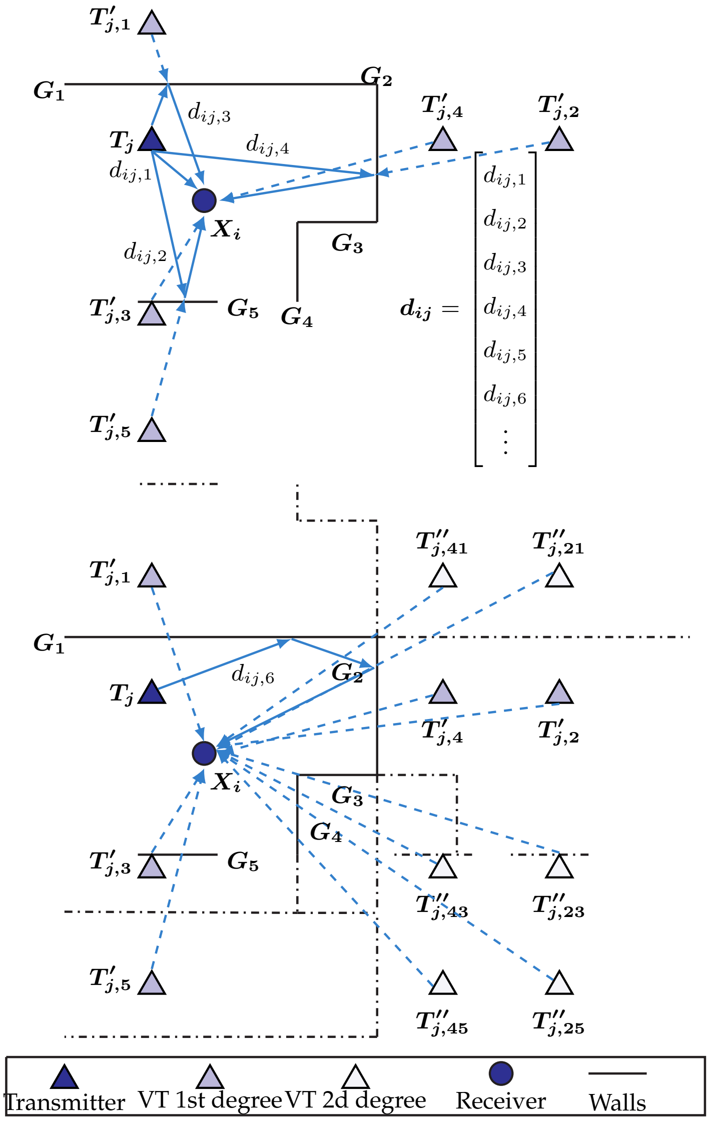


Figure 6.1: Example of first degree (top) and second degree (bottom) virtual transmitters (VTs). To save space not all second degree VTs are shown. Figure adapted from [5] ©2019 IEEE.

profile is observed [227]. Essentially, the ES moves as the receiver moves. Because of this, the position of a real and expected scatterer can not be calculated analogously to VT positions though triangulation of multipath components measured at different locations. Multipath components that are created by scattering are not considered in this thesis for VT reconstruction. The MCA algorithm in Chapter 3 is based on fingerprinting. The multipath profile needs to change somewhat smoothly over space, however it does not matter to the algorithm whether a multipath component was created by scattering or reflection. For geometry reconstruction, only VTs and reflected multipath components are used. If an ES is used for interpolation, the scattered multipath components will be automatically interpolated in the same way as reflected ones, resulting in slight inaccuracies and noise which is in any case present in the interpolated fingerprints. The ray tracing simulation tool used in this thesis only models reflected multipath components.

Existing VT estimation schemes					
Authors	Year	Input	Standard	VT Calculation	Application
Guo et al. [259]	2006	multipath delays	-	identify 1st and 2d degree VTs and use wall ceiling assumptions, geometry based method	environment reconstruction
Meissner et al. [218]	2010	geometry, multipath delays	UWB /channel sounder	from geometry	Position estimation
Kuang et al. [135]	2013	multipath delays	UWB / vector analyzer	from differences between the multipath profiles measured at several locations, solution of a linear system	Position estimation
Gentner et al. [43]	2016	multipath delays, phase and angle-of-arrival, movement information of the user	UWB / channel sounder	particle filter, Channel SLAM	Position estimation (SLAM)
Naseri et al. [136]	2017	multipath delays, angles-of-arrival	5.3 GHz, 120 MHz bandwidth / channel sounder	combining AoA and multipath information	environment reconstruction
Ulmschneider et al. [115], [260]	2020	multipath delays, phase and angle-of-arrival, movement information of the user	simulation	Channel SLAM [43], visibility regions, matching maps from different users	Position estimation (SLAM)
Gentner et al. [261]	2021	multipath delays	UWB Decawave DW1000 chip	Channel SLAM [43]	Position estimation (SLAM), environment reconstruction

Table 6.1: Existing work relying on the use of VTs.

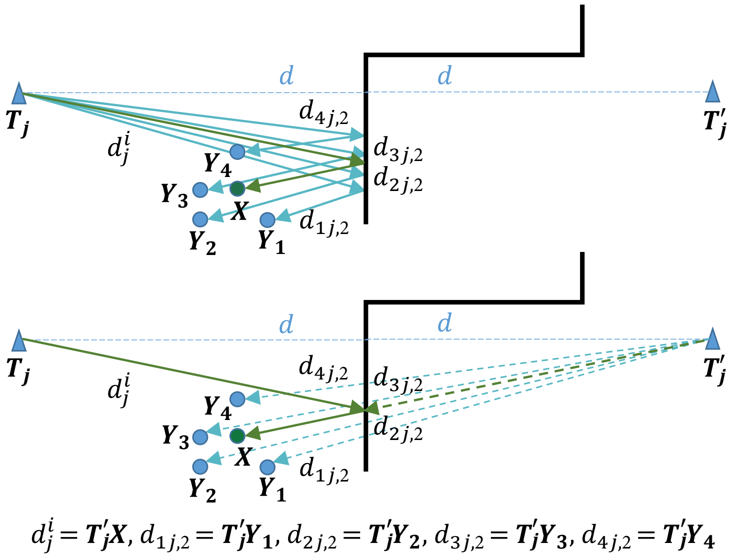


Figure 6.2: Multipath component interpolation. The multipath components are equal to the distances from the points to the VT $\boldsymbol{T}'_j$. The points $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_4$ and distances $d_{1j,2}, ..., d_{4j,2}$ can be used to calculate the location of $\boldsymbol{T}'_j$, and therefore, the propagation distance from the target point $\boldsymbol{X}$ to $\boldsymbol{T}_j$. Figure adapted from [4] ©2018 IEEE.

6.2 Proposed VT Reconstruction Scheme

The VT reconstruction scheme detailed in this thesis was initially proposed in the conference publications of the author [4] and [5]. Let $\boldsymbol{X}_i$,...,$\boldsymbol{X}_{i+\delta}$ be 4 or more distinct points, the received signals at which contain multipath components corresponding to the VT $\boldsymbol{T}'_{j,k}$. Those multipath components are equal to $\left|\left|\boldsymbol{T}'_{j,k} - \boldsymbol{X_i}\right|\right|, ..., \left|\left|\boldsymbol{T}'_{j,k} - \boldsymbol{X}_{i+\delta}\right|\right|$. As illustrated in Fig. 6.2, trilateration can be used to calculate the position of the VT $\boldsymbol{T}'_{j,k}$ from the coordinates of the reference points and multipath components.

However, it is not clear which multipath components in the fingerprints $\boldsymbol{d}_{i,j} \in \mathbf{D}_i, ...,$ $\boldsymbol{d}_{i+\delta,j} \in \mathbf{D}_{i+\delta}$ are created by signal reflections from the exact series of surfaces corresponding to the VT $\boldsymbol{T}'_{j,k}$. An MDP fingerprint contains multiple multipath components and the angle-of-arrival information, which could be used to easily identify the multipath component-VT correspondences, is not commonly known. In addition, as shown in Figs. 6.5 and 6.6, the number of multipath components contained in MDP vectors can vary.

The following concept is used. We know that if multipath components $d_{1j,x}...d_{4j,x}$ measured at $\boldsymbol{X}_1...\boldsymbol{X}_4$ correspond to $\boldsymbol{T}'_{j,k}$

$$\left|\left|\boldsymbol{T}'_{j,k} - \boldsymbol{X_1}\right|\right| = d_{1j,x}, ..., \left|\left|\boldsymbol{T}'_{j,k} - \boldsymbol{X}_4\right|\right| = d_{4j,x}. \tag{6.1}$$

Finding the correspondences directly by solving an optimization problem is difficult and computationally expensive. However, if a hypothesis grouping of multipath components and a VT exists, Eq. 6.1 can be used to verify that hypothesis. The following heuristic is used *if two locations are close together their multipath components, which were created by the same VTs, will be close in value*. This means that multipath delay profiles (MDPs) measured at close by locations will contain similar patterns in the values of their multipath components. Therefore,

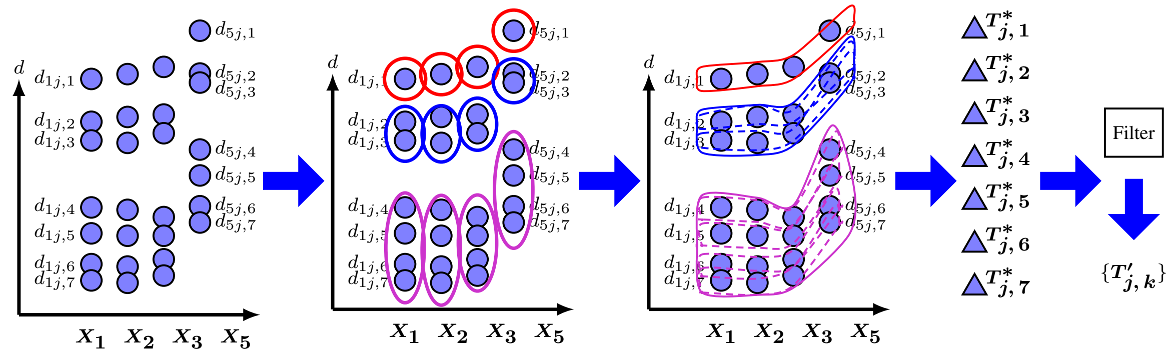


Figure 6.3: Algorithm used to estimate VT positions [5]. Candidate VT positions are generated and then filtered to obtain the VTs. Figure adapted from [5] ©2019 IEEE.

the proposed algorithm considers a number of reference multipath components measured within a small area. Pattern recognition is used to identify multipath components that correspond to the same VTs. However, since the reference points are located close together, any noise present in the multipath components will be magnified in the calculated VT positions. In addition, multipath correspondences may be misidentified. Therefore, VT estimation is repeated for multiple sets of points. A number of candidate VTs is generated. Equation 6.1 is then used to identify the correct VT positions.

Even if two reference points are located close to each other, pattern recognition is necessary to identify multipath correspondences for the following reason. If $d_{ij,1} \in \mathbf{D}_i$ corresponds to $\boldsymbol{T}''_{1,1}$, this does not mean that the multipath component in $d_{kj,12} \in \mathbf{D}_k$, which is closest in value to $d_{ij,1}$, will also correspond to $\boldsymbol{T}''_{1,1}$. In Fig. 6.6, for example, the value $d_{2j,2}$ is close to $d_{1j,3}$ and may be grouped together when in actuality they correspond to two different VTs and reflection surfaces. The correct MDP grouping, however, is $\{d_{1j,2}, d_{2j,3}\}$, $\{d_{1j,3}, d_{2j,4}\}$ and $\{d_{1j,1}, d_{2j,1}\}$. In a similar example in Fig. 6.3, $d_{3j,3}$ can be incorrectly grouped with $d_{5j,4}$. Therefore, instead of matching closest multipath components, the scheme proposed in this paper finds patterns within the multipath profiles and matches similar patterns across fingerprints. In order for similar patterns to be present in the MDPs, the reference fingerprints need to be located close to each other. As will be demonstrated later, an ideal fingerprint map for VT reconstruction consists of clusters of reference points distributed across the environment.

6.2.1 VT Reconstruction based on Clustering and Cluster Matching

Figure 6.3 illustrates the scheme proposed in [4], [5] for calculating the VT positions from a fingerprint map. In the case of environment reconstruction, several reference fingerprints located close to each other are selected from the fingerprint map as the input to the algorithm. Clusters of multipath components are first identified within each MDPs. Similar clusters are then matched across different fingerprints. The clusters which were determined to be similar are grouped together for interpolation. Multipath components are again matched across the clusters in the group and then interpolated or used for VT estimation. The proposed Algorithm 6 is illustrated in Fig. 6.3. It should be noted that this algorithm will produce a number of additional VTs along-side the correct ones. To obtain an exact set of VTs describing the

Algorithm 6: MDP Interpolation [4] ©2018 IEEE.

```
Data: Target point $\boldsymbol{X}$, reference points and fingerprints selected for interpolation
      $\{(\boldsymbol{Y}_k, \mathbf{D_k})\}$, $k = 1...K$, transmitters $\{\boldsymbol{T}_j\}, j = 1...M$
Result: Interpolated fingerprint $\mathbf{D}^i$
for $\forall$ transmitter $\boldsymbol{T}_j$ do
    for $l \leftarrow 1...K$ do
        $\mathbf{C}_l \leftarrow CalculateClusters(\boldsymbol{d}_{lj} \in \mathbf{D}_l)$
    $\mathbf{G} \leftarrow GroupClusters(\mathbf{C}_1, ..., \mathbf{C}_K)$
    for $\forall \boldsymbol{g} \in$ Group $\in \mathbf{G}$ do
        $d \leftarrow MCI(\boldsymbol{g}, \boldsymbol{Y}_g, \boldsymbol{X})$
        $\boldsymbol{d}^i_j \leftarrow [\boldsymbol{d}^i_j, d]$
$\mathbf{D^i} \leftarrow \{\boldsymbol{d}^i_1, ..., \boldsymbol{d}^i_M\}$
```

indoor geometry, the incorrect VTs need to be filtered out.

In the work in [4], clustering and cluster matching is performed by a classical algorithm with hand-tuned parameters. The algorithm is improved in [5] where clustering and cluster matching are performed instead by parameter-less machine-learning techniques. An alternative approach, later proposed in [262], finds the correspondences between multipath components directly, without clustering and cluster matching.

Algorithm 7: Multipath Component Clustering [4] ©2018 IEEE.

```
Data: Multipath components $[d_{ij,1}, ..., d_{ij,K}] \in \boldsymbol{d_{ij}}$
Parameters: Clustering threshold $\varepsilon_c$
Result: Set of multipath component clusters $\mathbf{C}$
Function CalculateClusters($\boldsymbol{d}_{ij}$)
    Sort($\boldsymbol{d}_{ij}$)
    $\boldsymbol{CurrentCluster} \leftarrow [\boldsymbol{d}_{ij}(1)]$
    for $l \leftarrow 2...K$ do
        if $|d_{ij,l} - d_{ij,(l-1)}| < \varepsilon_c$ then
            Add $d_{ij,l}$ to $\boldsymbol{CurrentCluster}$
        else
            Add $\boldsymbol{CurrentCluster}$ to $\mathbf{C}$
            $\boldsymbol{CurrentCluster} \leftarrow [d_{ij,l}]$
    Add $\boldsymbol{CurrentCluster}$ to $\mathbf{C}$
    return $\mathbf{C}$
```

6.2.2 Direct Clustering and Cluster Matching

In [4] multipath component clusters are first determined within each reference fingerprint. It is assumed that if two reference fingerprints are located next to each other, the clusters within them will look similar. Therefore, a cluster comparison and matching step is performed. Multipath components across the matched clusters are matched again and used to

```
Algorithm 8: Cluster Matching
  Data: Clustered multipath components $\{\mathbf{C}_i\}$ corresponding to sample points $\{\boldsymbol{Y}_i\}$
  Parameters: Cluster similarity threshold $\varepsilon_{group}$, $\alpha$
  Result: Delay clusters grouped for interpolation $\mathbf{G}$
  Function $\varepsilon(\boldsymbol{c}_1, \boldsymbol{c}_2)$ - Similarity metric
      $d_{offset} \leftarrow min(\boldsymbol{c}_1) - min(\boldsymbol{c}_2)$
      $n_d \leftarrow length(\boldsymbol{c}_1) - length(\boldsymbol{c}_2)$
      $d_s \leftarrow (max(\boldsymbol{c}_1) - min(\boldsymbol{c}_1)) - (max(\boldsymbol{c}_2) - min(\boldsymbol{c}_2))$
      $\epsilon \leftarrow \alpha|d_{offset}| + (1 - \alpha) \times (|n_d| + |d_s|)$
      return $\varepsilon$
  $\mathbf{G} \leftarrow \{[\boldsymbol{c}_i \in \mathbf{C}_i, ..., \boldsymbol{c}_j \in \mathbf{C}_j]\}, i \neq j$, such that
  $\varepsilon(\boldsymbol{c}_i, \boldsymbol{c}_j) \to min \bigwedge \varepsilon(\boldsymbol{c}_i, \boldsymbol{c}_j) \leq \varepsilon_{group}$
```

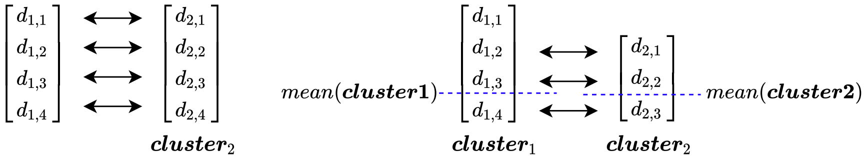


Figure 6.4: Matching multipath components between clusters. Left: Both clusters have the same number of elements. Right: The mean of the smaller cluster is aligned with the mean of the larger cluster.

triangulate or trilaterate the VT position. The clustering procedure proposed in [4] is summarized in Algorithm 7. The multipath component values in the MDP vectors are first sorted. A heuristically determined threshold ε_c is used. If the difference between two multipath components is smaller than ε_c they are added to a cluster.

Similar clusters would be grouped together such that a group $\{\mathbf{C_i}\}$ would contain at most one cluster from each fingerprint. Algorithm 8 is used to calculate the similarity metric ε between two clusters. The smallest elements of similar clusters should be close together (see Fig. 6.3), they each should contain a similar number of multipath components and have a similar distance between the smallest and largest multipath component. The Lagrangian parameter α is used to combine the three conditions into one value. A similarity threshold ε_{group} is defined. Clusters, the similarity metric between which is the smallest, and is below a set threshold ε_{group}, that is $\varepsilon(\boldsymbol{c_1}, \boldsymbol{c_2}) \leq \varepsilon_{group}$, are grouped together.

If the clusters in a group have the same number of elements, then the multipath components at the same positions in the sorted cluster vectors are assumed to correspond to the same VT. The mean of the clusters in the group which have a smaller number of multipath components are aligned with the mean of the sorted larger cluster vectors as shown in Fig. 6.4. It then becomes possible to again group multipath components in the same positions. If more than 3 multipath components are grouped together, the position of the VT is calculated using trilateration. It should be noted that this step relies on heuristics more than the VT reconstruction scheme in [5], as its goal is fingerprint interpolation rather than perfect VT reconstruction.

6.2.3 ML-based clustering and cluster matching

The algorithm used to calculate the VT positions for environment reconstruction in [5] performs clustering and cluster-matching, same as [4]. However, the performance of the original clustering and cluster matching schemes proposed in [4], was shown to be strongly dependent on the choice of the internal parameters. Therefore, in order to improve the matching accuracy and robustness, machine learning-based clustering and cluster matching algorithms were proposed in a Bachelor thesis written at the Chair of Media Technology of the Technical University of Munich [263]. In this work, the dynamic k-means clustering algorithm with cosine similarity [264] is used to cluster the multipath components and the Wards method [265] is used for cluster matching.

6.2.4 Direct VT Reconstruction

An alternative approach for multipath component clustering is proposed in another Bachelor thesis written at the Chair of Media Technology of the Technical University of Munich [262]. It achieves superior VT reconstruction accuracy under noise conditions compared to [263]. Here, the correspondences between the multipath components across fingerprints are found directly through pattern matching. The clustering and cluster matching steps are not used. Applying this VT reconstruction method to the proposed interpolation and environment reconstruction schemes is part of the future work.

6.2.5 VT Filtering

Algorithm 9: VT Filtering [5] ©2019 IEEE.

```
Data: Candidate VTs $\boldsymbol{T^*_{j,k}}$, reference data $\mathbf{MAP} = \{(\mathbf{D_i}, \boldsymbol{X_i}), i = 1...N\}$
Result: VT positions $\boldsymbol{T'_{j,k}}$
for ∀ candidate VTs $\boldsymbol{T^*_{j,k}}$ do
    $n_{j,k} \leftarrow 0$
    for $i = 1...N$ do
        Calculate $d_{ik} \leftarrow \left|\left|\boldsymbol{T^*_{j,k}} - \boldsymbol{X_i}\right|\right|$
        if $d_{ik} \in \boldsymbol{d_{ij}}$ then
            $n_{j,k}++$
$\{\boldsymbol{T^i_{j,k}}\} \leftarrow \{\boldsymbol{T^*_{j,k}} \mid n_{j,k} \geq \gamma_{VT} \times \frac{1}{|\{\boldsymbol{T^*_{j,k}}\}|} \sum_{j,k} n_{j,k}\}$
$\{\boldsymbol{T'_{j,k}}\} \leftarrow Cluster(\{\boldsymbol{T^i_{j,k}}\})$
```

All of the VT reconstruction algorithms mentioned above require that the reference fingerprints are located in a small area. Therefore, all of the algorithms select multiple small clusters of close-located reference points throughout the environment and individually perform VT estimation for each cluster. This will produce a large number of VTs $\boldsymbol{T^*_j}$. The VTs calculated in this step are referred to as *candidate VTs* as not all of them will align with the real VTs in the environment. This is because false matches can be created by the multipath matching algorithm. Therefore, the candidate VTs are next filtered using Algorithm 9. A number $n_{j,k}$ is calculated for each candidate VT $\boldsymbol{T^*_{j,k}}$. $n_{j,k}$ is the number of reference fingerprints that contain a multipath component equal to the distance between $\boldsymbol{T^*_{j,k}}$ and the point at which that reference fingerprint was measured. It is the number of reference points that receive a signal from the VT $\boldsymbol{T^*_{j,k}}$. If the distance $\left|\left|\boldsymbol{X_i} - \boldsymbol{T^*_{j,k}}\right|\right|$ with a certain tolerance is not present in the fingerprint vector $\boldsymbol{d_{i,j}} \in \mathbf{D_i}$, then neither is the propagation path containing $\boldsymbol{T^*_{j,k}}$. The threshold parameter γ_{VT} is used to filter the VTs. The VTs for which $n_{j,k}$ is higher than $\gamma_{VT}\times$ are kept. The rest are filtered out. If the parameter γ_{VT} is high; the algorithm can filter out some correct VT positions, however, it is less likely to keep wrong ones. If γ_{VT} is low, the algorithm will keep more of the correct VT positions, however it may also keep some false VTs. Afterwards the remaining VT positions are clustered together. The final VT positions are calculated as the averages of each cluster. This is done because there can be multiple candidate VTs corresponding to the same location but not overlapping because of noise. The first order VTs will be separated from the second and higher order VTs, shown in Fig.6.1, later on in the algorithm.

6.3 Fingerprint Interpolation

The aim of fingerprint interpolation/extrapolation is to computationally generate a fingerprint at a location, referred to in this thesis as a *target point*, given fingerprints measured at a sparse set of locations in the environment. In this way, the density of the fingerprint database can be increased without time-consuming measurements. For interpolation, the target point needs to be surrounded by the reference points at which fingerprints were measured. Otherwise, the procedure of generating the fingerprint at the target point is referred to as extrapolation.

6.3.1 Existing Fingerprint Interpolation Strategies

Multiple fingerprint interpolation strategies have been proposed for received signal strength indicator (RSSI) fingerprinting [21], [108], [250], [251], [266]–[270]. In some publications, including [21], [250], [251], [267], [268], an interpolated fingerprint is calculated as a linear combination of several neighboring measured fingerprint vectors. The work in [268] does this though a semi-supervised learning approach. The multivariate polynomial function is used in [270]. The modified Shepard's interpolation method is used in [267]. A propagation path loss model is used to generate RSSI fingerprints in [251], [269], [271]. The standard path loss model used in [269] is:

$$RSSI_d = RSSI_{d_0} - 10nlog_{10}(\frac{d}{d_0}) + X_\sigma, \tag{6.2}$$

where $RSSI_{d_0}$ is the RSSI value at the distance d_0 from the AP. X_σ models the noise with zero mean and standard deviation σ. n is the attenuation factor. In order to keep the model accurate without the need to represent unpredictable channel attenuation, reflections, diffraction and the presence of moving objects and people, the receiver is assumed to be in close proximity to the AP. LS optimization is used to calculate the model parameters in Eq. 6.2. In [108] hidden Markov models and the expectation-maximization algorithm are used to extend the fingerprint map with sequential unlabeled user fingerprints. In [266] the Fourier transform is applied to the fingerprint map. Fingerprints are interpolated by exploiting the sparsity of the fingerprint matrices in the Fourier domain and by solving a minimization problem using the least absolute shrinkage and selection operator. In [272], RSSI fingerprints are interpolated the following way. Regression analysis is performed over distance and frequency domains for path loss modeling. Shadowing at a target frequency is obtained by interpolation in the frequency domain. Afterwards shadowing factors at the target band are interpolated over space using Kriging interpolation.

RSSI fingerprints are cheaper to collect as they are automatically measured by most handheld consumer devices, the RSSI measure aggregates the entire channel information into one scalar value. Because of this, there are strong and unpredictable fluctuations in measured RSSI values both over space and time [58], [62], resulting in lower localization accuracy. Therefore, the CSI has recently gained favor for fingerprinting-based localization. Unfortunately, the RSSI interpolation strategies mentioned above cannot be directly applied to CSI fingerprint interpolation. The corresponding taps of CIR vectors can not be simply averaged for the following reason. A CIR is a superposition of several propagation delays.

$$x(t) = h(t) * s(t), \tag{6.3}$$

$$h(t) = \sum_{k=1}^{K} a_k \delta(d_k/c). \tag{6.4}$$

It has a peak at each multipath delay in the MDP, as illustrated in Fig. 6.5. When the CIR is measured at two close-by locations there will be a slight shift between the positions of corresponding peaks. Therefore, if the CIR at a third near-by location is directly interpolated by averaging each tap of the two measured CIRs, the result will be a CIR with wider peaks of low magnitude, for example, wider peaks between $d_{2j,1}$ and $d_{1j,1}$, $d_{2j,2}$ and $d_{2j,3}$, $d_{1j,4}$ and $d_{2j,3}$ in Fig. 6.5. The correctly interpolated CIR should have peaks of the same sharpness as the measured CIRs. The positions of its peaks $d^i_{j,1}$, $d^i_{j,2}$ and $d^i_{j,3}$ should be interpolated from the positions of the peaks in the measured CIRs. It should be noted that the number of peaks in the CIRs of neighboring points may vary as certain signal propagation paths may exist at one point and be blocked at others.

There is very limited literature on the interpolation of CIR and CSI fingerprints over space. In [273] and [274], the multipath components in a measured CIR are tracked as they appear and disappear as a user moves through an indoor environment. The obtained information is used to interpolate two CIRs. In the work in [275], CIR-based fingerprints are averaged as power values before interpolation and localization. Because of the challenges of CIR and CFR interpolation, this work focuses on the interpolation of multipath components. A CIR or CFR at a target location can afterwards be calculated from the interpolated multipath components.

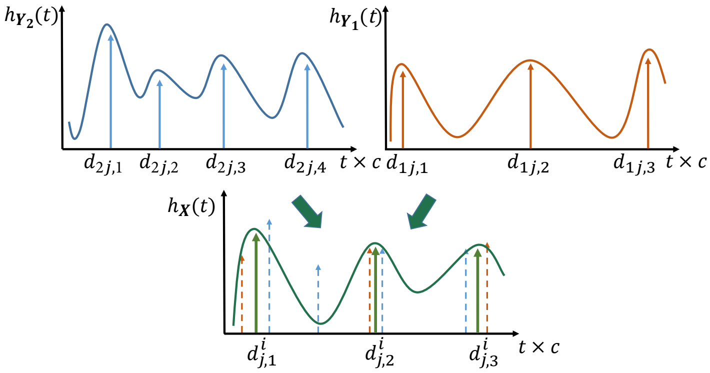


Figure 6.5: Channel impulse response (CIR) interpolation. The positions of the CIR peaks need to be interpolated as well as their magnitudes. Corresponding MDPs are shown in Fig. 6.6. Figure adapted from [4] ©2018 IEEE.

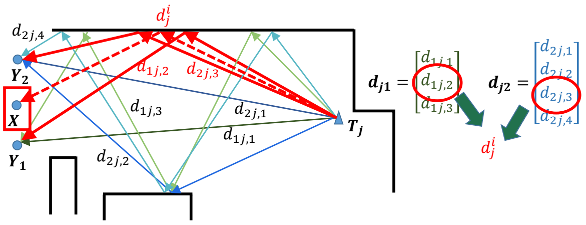

d^i_j
$d_{2j,4}$
Y_2
X
Y_1
$d_{1j,2}$ $d_{2j,3}$
$d_{1j,3}$
$d_{2j,1}$
$d_{1j,1}$
$d_{2j,2}$
T_j
$\mathbf{d}_{j1} = \begin{bmatrix} d_{1j,1} \\ d_{1j,2} \\ d_{1j,3} \end{bmatrix}$
$\mathbf{d}_{j2} = \begin{bmatrix} d_{2j,1} \\ d_{2j,2} \\ d_{2j,3} \\ d_{2j,4} \end{bmatrix}$
d^i_j

Figure 6.6: Interpolation of multipath fingerprints. Before the multipath components can be interpolated, the correspondences between the propagation paths need to be determined. In the example above, $\boldsymbol{d}^i_j \in \mathbf{D^i}$ needs to be interpolated from the pairs $\{d_{1j,3}, d_{2j2}\}$, $\{d_{1j,2}, d_{2j,3}\}$ and $\{d_{1j,1}, d_{2j,1}\}$. Figure adapted from [4] ©2018 IEEE.

6.3.2 Proposed Approach

This section details the novel multipath fingerprint interpolation scheme based on multipath component matching and the calculation of VT positions and initially proposed in [4].

6.3.3 Interpolation of Multipath Components

An advantage of the multipath-based fingerprint structure detailed in Chapter 3 is that the propagation distances or multipath components can be interpolated individually. The known locations of the reference points are denoted as $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_K$ and the corresponding multipath components as $d_{1j} \in \boldsymbol{d}_{ij}, ..., d_{Kj} \in \boldsymbol{d}_{Kj}$. $\boldsymbol{X}$ denotes the target point. The propagation distance from $\boldsymbol{T}_j$ to $\boldsymbol{X}$, that is being interpolated, is denoted d^i_j (see Fig. 6.2).

Theorem 1. *"The propagation distance d^i_j from the transmitter $\boldsymbol{T}_j$ to the point $\boldsymbol{X}$ is uniquely determined by the locations of 4 or more reference points $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_K$ and the corresponding propagation distances $d_{1j}, ..., d_{Kj}$, with reflections from the same surfaces in the same order" [4].*

Proof. "The lengths of the propagation paths $d_{1j}, ..., d_{Kj}$ are equal to the distances from the reference points to the VT (see Fig. 6.2). If $K \geq 4$, the location of $\boldsymbol{T}'^{(l)}_j$ is fully determined by, and thus can be calculated from $\boldsymbol{Y}_1\boldsymbol{T}^{(}l)'_j, ..., \boldsymbol{Y}_K\boldsymbol{T}'^{(l)}_j$. The length of the propagation path from the target point $\boldsymbol{X}$ to $\boldsymbol{T}_j$ is in turn equal to the distance $\boldsymbol{X}\boldsymbol{T}'^{(l)}_j = d^i_j$." □

In order to interpolate several MDPs, the correspondences between their multipath components need to first be identified. A correspondence means that two multipath components are created by the same VT with reflections from the same surfaces in the same order. For example, in Fig. 6.6 the multipath components $d_{1j,2} \in \boldsymbol{d_{1j}}$ and $d_{2j,3} \in \boldsymbol{d}_{2j}$ correspond to the same reflection order and VT. The clustering and cluster matching algorithms detailed in Section 6.2.1 are used to group together the multipath components. If more than 3 multipath components are grouped together for interpolation, the approach proposed in [4] and detailed in Algorithm 10 first calculates the coordinates of the VT $\boldsymbol{T_j'}$ using multilateration, non-linear least squares and the known multipath components [276]. Least squares (LS) is used since the known multipath components may contain noise. Once the coordinates of the VT are known, the interpolated multipath component is calculated as the distance from the target point to the VT. If the number of known multipath components is less than 3, the interpolated multipath component is set equal to their average. It should be noted that, if the coordinates of the VT $\boldsymbol{T}_j'^{(l)}$ are obtained, the multipath component at the target point d_j^i can be calculated regardless of whether the target point $\boldsymbol{X}$ is enclosed by the reference points or is far away from them. Therefore, it is possible to use the proposed scheme to extrapolate fingerprints in regions where no measured fingerprints are present.

Algorithm 10: Multipath Component Inter-/Extrapolation [4] ©2018 IEEE.

```
**Data:** Reference multipath components $\{d_{kj}\}$ and points $\{\boldsymbol{Y}_k\}, k = 1...K$, target
        location $\boldsymbol{X}$
**Result:** Interpolated multipath delay value $d_j^i$
**Function** $MCI(d_{1j}, ..., d_{Kj}, \boldsymbol{Y}_1, ..., \boldsymbol{Y}_K, \boldsymbol{X})$
    $\boldsymbol{T}'^{(l)}$ - VT corresponding to $d_{1j}, ..., d_{Kj}$
    $\boldsymbol{T}'^{(l)} \leftarrow Trilateration(d_{1j}, ..., d_{Kj}, \boldsymbol{Y}_1, ..., \boldsymbol{Y}_K)$
    $d_j^i \leftarrow \| \boldsymbol{T}'^{l} - \boldsymbol{X} \|$
    **return** $d_j^i$
```

6.3.4 Simulation Results

The geometry used for the simulation setup is identical to the one in Section 3.5.

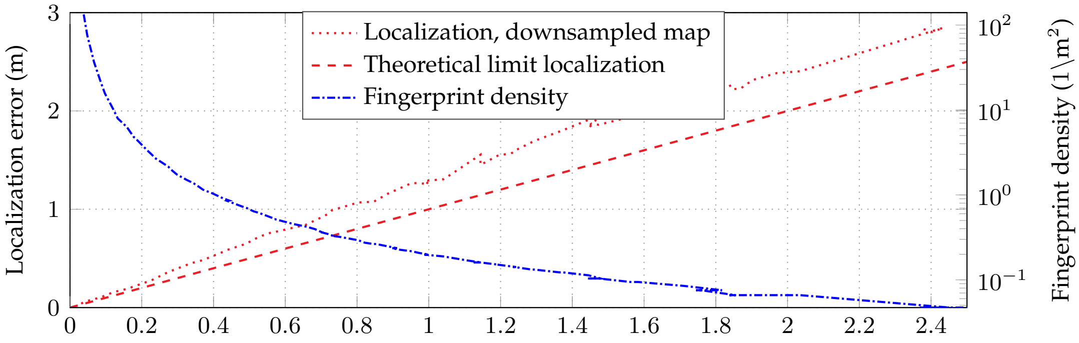


Figure 6.7: Localization error obtained without interpolation or extrapolation. The x-axis shows the average distance from the query points to the closest reference point. Figure adapted from [4] ©2018 IEEE.

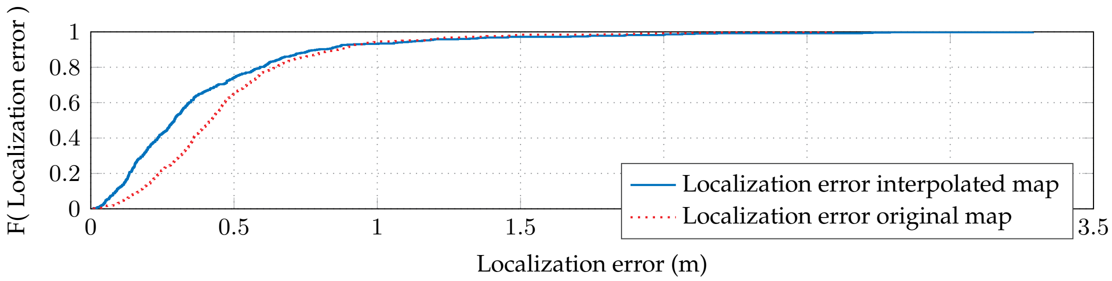


Figure 6.8: Localization error CDF obtained by interpolating the reference points at 2440 (2000%) target points. Figure adapted from [4] ©2018 IEEE.

The localization error is plotted in Fig. 6.7 as a function of the density of the fingerprint map without interpolation. The parameter used to characterize the map density is the average distance from the query points to the closest reference point. The downsampling procedure described in [250] was used. Since the MCA fingerprinting algorithm is used here without averaging or filtering such as k-means, the localization error at a query point cannot be smaller than its distance to the closest reference point, the identity function is plotted as the theoretical limit. The figure shows that the more the map is downsampled the more the localization error deviates from the theoretical limit. The reason for this is that the similarity calculated for two fingerprints, which are far from each other, tends to be unpredictable. If the query fingerprint is far from the closest reference point, it is likely that another fingerprint that is further way from the query has a larger calculated similarity metric value. This also explains the unevenness of the plot.

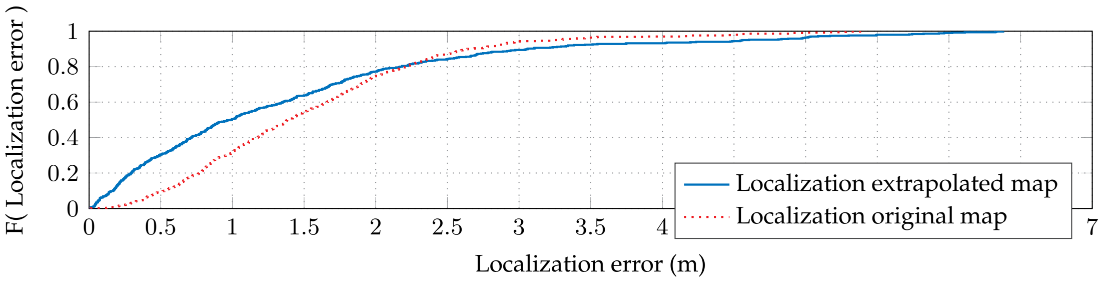


Figure 6.9: Localization error CDF, the reference points marked by a square in Fig. 3.7 were extrapolated at 4000 target points. Figure adapted from [4] ©2018 IEEE.

Table 6.2 shows the change in the average localization error when an interpolated/ extrapolated fingerprint map is used for localization. The table also includes the parameters used by Algorithms 6, 10, 7 and 8. The cumulative distribution functions (CDFs) of the localization errors obtained from a interpolated and extrapolated map are shown in Figures 6.8 and 6.9 respectively. In the extrapolation experiment in Figure 6.9 only 4-5 reference fingerprints are used per room, the rest of the fingerprints in the map are obtained though extrapolation. In another experiment, first the localization errors were calculated at a number of query locations using only 14 reference points. Afterwards 4000 interpolated and extrapolated fingerprints were added to the map. The resulting localization errors are shown on the left and right of Figure 6.10 respectively. The original reference fingerprints are marked

d_{map}	Original error	Interpolated	d_{int}	Reference fingerprints	ε_c	ε_{group}	α	ε_{MCA}	Error gain
Interpolation									
0.38m	0.48m	200 %	0.21m	within 1m	0.1	0.5	0.5	1	13.8 %
0.38m	0.48m	800 %	0.13m	within 1m	0.1	0.5	0.5	1	18%
0.38m	0.48m	2000 %	0.09m	within 1m	0.1	0.5	0.5	1	20%
Extrapolation									
1.34m	1.55m	500 points	0.19m	same room	2	3.5	0.5	1	6 %
1.34m	1.55m	2000 points	0.09m	same room	2	3.5	0.5	1	10 %
1.34m	1.55m	4000 points	0.07m	same room	2	3.5	0.5	1	11.4%

Table 6.2: Localization error gain achieved through interpolation and extrapolation and the corresponding parameter values. Original error is the localization error calculated only using the reference fingerprints, d_{map} - average distance from the query points to the closest reference point, d_{int} - average distance from the query points to the closest reference or target point. Table adapted from [4] ©2018 IEEE.

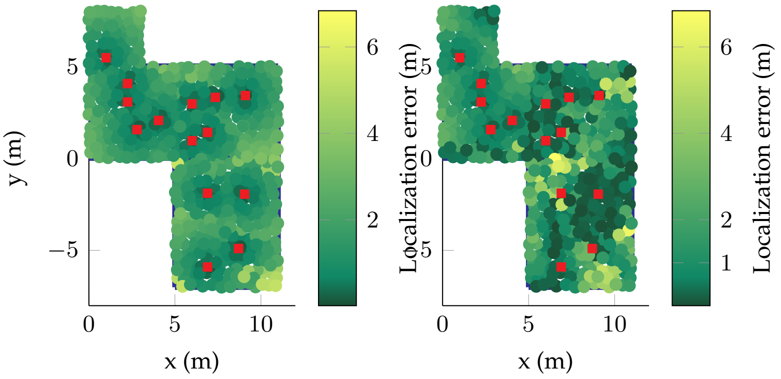


Figure 6.10: Fingerprint extrapolation example. Left: only the reference fingerprints marked with red squares were used to localize the query points. Right: The fingerprint map was extended by 4000 extrapolated points. Figure adapted from [4] ©2018 IEEE.

in red. The results shows an increased number of regions where the user is localized correctly. However, the results also show several locations where a localization error over 2.5m was obtained. In Fig. 6.10 such locations mainly correspond to geometrical features and corners. It can be theorized that the interpolation algorithm did not perform as well there because of certain VTs being blocked and being visible in points in a close proximity. Overall, the probability that the user is localized correctly was much higher when the interpolated/extrapolated map is used. As shown by Figure 6.10, the user can also be localized in regions not immediately next to a recorded reference point.

6.3.5 Discussion Fingerprint Interpolation

The main disadvantage of the scheme proposed in [4] is its sensitivity to the parameter values, the correct configuration of which needs to be selected. The optimal parameter values can be iteratively determined using the fingerprint map. A solution for this problem is proposed in [263] and [262]. In [263] ML-based clustering and cluster matching algorithms without a parameter dependency are used. In [262], the multipath components are matched directly without clustering and cluster matching.

The fingerprint map is interpolated online, therefore, the fingerprint interpolation scheme can be complex and its run-time does not significantly impact system performance. The run-time of the MCA increases linearly with the number of fingerprints in the map. Therefore, the overall run-time of the ILS will increase linearly with the number of interpolated fingerprints. The run-time complexity can be reduced by first performing coarse localization using only measured fingerprints and then refining the localization estimate using interpolated fingerprints calculated within a defined radius of the initially calculated coordinates.

Table 6.2 and Fig. 6.7 show that interpolating the fingerprint map is not as effective as using more measured fingerprints. Table 6.2 also shows a saturation in the localization error at large numbers of interpolated fingerprints. This means that the interpolated fingerprint is not identical to the fingerprint that would have been measured at that location. This can be viewed as noise in the interpolated fingerprints. Because of it, the interpolated fingerprint most similar to the query is not the one calculated at a location that is nearest to the query point. However, using interpolation is much cheaper than measuring a dense fingerprint map and Table 6.2 and Figs. 6.8 and 6.9 show the feasibility of the proposed scheme. The future work on this topic needs to focus on improving the effectiveness of the algorithm and further integration of the scheme proposed in [263] and [262] and similar approaches to fingerprint interpolation.

6.4 Environment Reconstruction

If a 3D model of an environment is known, it can be used to improve path planning, localization and navigation for human users, robots and UAVs. A VT-based environment model can also be combined with an RF multipath-based simultaneous localization and mapping (SLAM) algorithm [43]. In [277] the estimated indoor geometry is used to calculate channel impulse responses and improve localization accuracy. A 3D model of the walls and obstacles can also aid in obtaining a semantic understanding of an environment.

6.4.1 Existing Environment Reconstruction Approaches

Laser scanners and Light Detection and Ranging (Lidar) devices are often used to generate 3D point clouds of an environment. Stereo infra-red (IR) and camera systems are also used to get a model and coordinates of walls and objects. Lidars and camera systems are well studied and effective at creating a digital twin of the environment. However, such systems require expensive hardware. If, on the other hand, a model or digital twin of the environment can be created from an RF fingerprint map or calibration data of the ILS, obtaining such a model would not create extra costs beyond the required computational power. A model obtained from RF measurements can also be fused with Lidar point clouds and camera images.

A number of publications use RF data to gain only specialized information about the environment. Instead of creating a full 3D environment model, the work in [278] detects and tracks individual objects near the APs and receiver using the channel impulse response (CSI). Ultra wide band (UWB) radars are used to track objects behind a wall or a non-conducting obstacle in [279]–[281]. UWB radars are used to obtain information about object shape and properties in [282], [283] and to evaluate the structure and quality of a brick wall in [284]. In [285] a WiFi emitter is placed behind an object and a wavefront measured with a scanning antenna on a meter-sized two dimensional area in front of the object is used to obtain the object's image.

The full 3D geometry can be reconstructed from RF data in several ways. A UWB radar [277], [279]–[281], [286] can create a point cloud by measuring distances to the closest obstacle in a number of directions. A mm-wave radar is used in [287]. The measurements are precise but time consuming as a separate radar position/orientation is needed for each point in the point cloud. In the work in [286], a UWB radar is used to detect indoor features and landmarks, such as walls, edges and corners. Those features are then used to construct a map. Alternatively, dense signal measurements and a large number of fixed transmitters and receivers were used in a simulation in [285] to calculate the coordinates of the emitters and a 3D tomography of absorptive objects in a building.

Multipath components and VTs are used in [259] to reconstruct a rectangular room. The algorithm, however, assumes that the room consists of a parallel floor and ceiling which are perpendicular to the walls. In [136] multipath time-of-arrival (ToA) and direction-of-arrival (DoA) values estimated from a UWB signal are used to calculate transmitter and VT coordinates. The reflector point at which the signal reflected of a surface is calculated as the intersection between the lines connecting the real transmitter and virtual receiver and the line connecting the real receiver and virtual transmitter. Multiple reflection points are calculated as the receiver is moved through the environment and a Hough detector is used to fit planes to the reflector points. Multipath delays and the IMU-based user motion data is used in [47] in combination with Channel SLAM to estimate the coordinates of reflective surfaces and scatterers.

6.4.2 Proposed Geometry Reconstruction Scheme

An indoor geometry is represented as a set of planar polygons. Ground truth polygons are denoted as $\mathbf{G}_i$ with i taking the values $1, ..., K$. A reconstructed polygon that corresponds to $\mathbf{G}_i$ is denoted as $\hat{\boldsymbol{G}}_{i,l}$, as multiple reconstructed polygons can correspond to one real polygon $\mathbf{G}_i$. Algorithm 11, initially presented in [5], consists of three steps.

1. The VT coordinates are calculated using multipath delay profiles (MDPs).
2. Planes in the environment are calculated as in the middle of and perpendicular to the lines from the transmitters to the VTs.
3. The MDPs and estimated reflection planes are used to calculate polygons in the indoor environment.

Since the ID of an AP is generally included in the data it transmits, the algorithm can know which VT $\boldsymbol{T}'_{j,k}$ corresponds to which AP $\boldsymbol{T}_j$. As illustrated in Fig. 6.11, the polygon,

Algorithm 11: Environment Reconstruction [5] ©2019 IEEE.

```
Data: Reference multipath components and reference points
      $\mathbf{MAP} = \{(\mathbf{D}_i, \boldsymbol{X}_i), i = 1...N\}$
Result: Set of polygons representing the reflecting geometry $\{\mathbf{G}_{jk,i}\}$
for $\forall$ transmitter $\boldsymbol{T}_j$ do
    Estimate initial VT candidate set $\{\boldsymbol{T}^*_{j,k}\}$
    Filter the VT positions get the VTs $\{\boldsymbol{T}'_{j,k}\}$
    for $\forall$ VTs $\boldsymbol{T}'_{j,k}$ do
        Calculate the reflector plane $\hat{\mathbf{G}}_k$
        for $\forall(\mathbf{D}_i, \boldsymbol{X}_i) \in \mathbf{MAP}$ do
            Check visibility of $\boldsymbol{T}'_{j,k}$
            Update edges of polygons $\{\hat{\mathbf{G}}_{jk,i}\}$
        Extend $\{\hat{\mathbf{G}}_{jk,i}\}$ by $\frac{1}{2}d$(nearest neighbor)
        Keep polygons that correspond to order 1 VTs
Merge Information from all transmitters
```

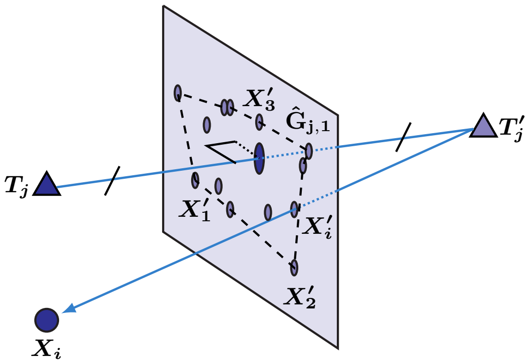


Figure 6.11: Geometry Reconstruction. Figure adapted from [5] ©2019 IEEE.

from which the signal was reflected on the given propagation path, contains the middle point of the line segment between the AP and VT $\boldsymbol{T}_j\boldsymbol{T}'_{j,k}$ and is perpendicular to that line segment.

The polygons corresponding to higher order VTs can be filtered out the following way. Let $\hat{\mathbf{G}}_{\mathbf{jk},\mathbf{l}}$ be a polygon reconstructed from the VT $\boldsymbol{T}'_{\boldsymbol{j},\boldsymbol{k}}$ We check if the path from the reference point $\boldsymbol{X_i}$ to $\boldsymbol{T}'_{\boldsymbol{j},\boldsymbol{k}}$ intersects any other reconstructed polygon $\hat{\mathbf{G}}_{\mathbf{je},\mathbf{n}}$ before $\hat{\mathbf{G}}_{\mathbf{jk},\mathbf{l}}$. Let the corresponding VTs $\boldsymbol{T}'_{\boldsymbol{j},\boldsymbol{k}}$ and $\boldsymbol{T}'_{\boldsymbol{j},\boldsymbol{l}}$ be the reflections to the same transmitter $\boldsymbol{T_j}$. If that is the case, $\boldsymbol{T}'_{\boldsymbol{j},\boldsymbol{k}}$ can be considered a higher order VT and the $\hat{\mathbf{G}}_{\mathbf{jk},\mathbf{l}}$ can be removed from the reconstructed geometry.

6.4.2.1 Reconstructing Polygons

The proposed scheme first determines which reference fingerprints contain multipath components that were created by the 1st degree VT $\boldsymbol{T}'_{j,k}$. This means that the algorithm finds the reference locations $\boldsymbol{X}_i$ which a signal from the AP $\boldsymbol{T}_j$ can reach after being reflected from the polygon corresponding to $\boldsymbol{T}'_{j,k}$. The fingerprint $\boldsymbol{d}_{i,j} \in \mathbf{D}_i$ of such a reference point $\boldsymbol{X}_i$ should contain multipath components approximately equal to the distance $\boldsymbol{X}_i\boldsymbol{T}'_{j,k}$. The set of reference points that fulfill this condition is denoted $\mathbf{S}_{j,k}$. The intersections of the reconstructed plane that corresponds to the VT $\boldsymbol{T}'_{j,k}$ and the line segments $\boldsymbol{T}'_{j,k}\boldsymbol{X}_i$, with $\boldsymbol{X_i} \in \mathbf{S_{j,k}}$ are denoted as $\boldsymbol{X}'_i \in \mathbf{S}'_{j,k}$. As illustrated in Fig. 6.1, $\boldsymbol{X_i} \in \mathbf{S_{j,k}}$ are the points at which the signal from $\boldsymbol{T_j}$ to $\boldsymbol{X}_i$ was reflected from a polygon $\boldsymbol{G}_{jk}$.

It can happen that $\boldsymbol{d}_{ij} \in \mathbf{D}_i$ corresponding to a different VT $\boldsymbol{T}'_h$ can coincidentally contain a multipath component equal to the distance $\boldsymbol{T'_{j,k}X_i}$. Therefore, outlier detection is needed after all points are included in $\mathbf{S}'_{j,k}$. Multiple polygons in an indoor geometry can lie on one plane. An example of this are the walls around a door way or furniture. The algorithm, therefore, searches for clusters of points in $\mathbf{S'_{j,k}}$. For every point in the set its distance to the closest neighbor is calculated as:

$$d_i = min(||\boldsymbol{X'_i} - \boldsymbol{X'_l}|| \mid \boldsymbol{X'_i}, \boldsymbol{X'_l} \in \mathbf{S'_{j,k}}, i \neq l). \tag{6.5}$$

A polygon is then reconstructed for clusters with 3 points or more. Point clustering is performed such that every point $\boldsymbol{X'_i}$ in the cluster $\mathbf{C_r}$ has a distance less than its threshold d_i to at least one other point in the same cluster. A threshold distance d_i is calculated for every point $\boldsymbol{X'_i}$, to account for the differing point densities within and across clusters.

$$\mathbf{C_r} = \{\boldsymbol{X'_i} \in \mathbf{S'_{j,k}} \mid \exists \boldsymbol{X'_l} \in \boldsymbol{C_r} : ||\boldsymbol{X'_i} - \boldsymbol{X'_l}|| \leq d_i\}. \tag{6.6}$$

The algorithm starts at an arbitrary reflection point, and adds each subsequent point to an existing or a new cluster according to Eq.6.6. When a point is contained in more than one cluster, the clusters are merged. A polygon is obtained from the points on the outward boundary of a cluster and extended outward by $\frac{1}{2}$ of the distance of each boundary point to its closest neighbor. This is done for the following reason. If the signal from the VT $\boldsymbol{T'_{j,k}}$ is obtained by a reference point but not by its neighbor, the boundary of the corresponding reflective polygon should be between the two respective reflection points on the plane. Therefore, the polygons are further extended outward by $\frac{1}{2}$ of the distance of each point on the contour of the cluster to its closest cluster point.

As mentioned above, the reflecting polygons in the geometry correspond to 1st degree VTs. A polygon $\hat{\mathbf{G}}_{jk,l}$ reconstructed based on the multipath components from a higher level VT $\boldsymbol{T}'^{(l)}_{j,k}$ can be identified the following way. If the line from a reference point $\boldsymbol{X}_i$ to $\boldsymbol{T}'^{(l)}_{j,k}$ intersects any other reconstructed polygon $\hat{\mathbf{G}}_{je,n}$, corresponding to the same AP $\boldsymbol{T}_j$ as $\hat{\mathbf{G}}_{jk,l}$, before $\hat{\mathbf{G}}_{jk,l}$, then $\boldsymbol{T}'^{(l)}_{j,k}$ is a VT corresponding to paths with more than one reflection.

Once a set of polygons is obtained, contextual information can further be used to fill in the indoor environments. For example, polygons identified as walls can be extended from the floor to the ceiling. There is, however, a trade-off when gap-filling is performed in this way. The algorithm can fill empty spaces such as doors and windows. Filtering can also be useful, but it can also remove important details from the environment.

6.4.3 Evaluation Metrics

Several numerical metrics are derived to evaluate the performance of the proposed algorithm. Geometry deviation represents how much the reconstructed geometry is aligned with the ground truth planes. Geometry coverage represents the percentage of the ground truth that was reconstructed correctly. The coverage error is the percentage of the reconstructed area that is not present in the ground truth. **Geometry deviation** (δG) is defined as the average distance from the vertices of the reconstructed polygons to the projections of those vertices onto the planes in the ground-truth geometry. The obtained number is then weighted by the areas of the polygons.

$$\delta G = \frac{1}{\sum_{i=1}^{K} \triangle \hat{\mathbf{G}}_i} \sum_{i=1}^{K} \frac{\triangle \hat{\mathbf{G}}_i}{K_i} \sum_{l} \| \hat{\mathbf{G}}_i(l) - P(\hat{\mathbf{G}}_\mathbf{i}(l), \mathbf{G}_i) \|, \tag{6.7}$$

where $\hat{\boldsymbol{G}}_{\boldsymbol{i}}(l)$ is the l-th vertex of the i-th reconstructed polygon. $P(\hat{\mathbf{G}}_i(l), \mathbf{G}_i)$ is the projection of the point $\hat{\mathbf{G}}_i(l)$ onto the corresponding ground-truth plane $\mathbf{G}_i$. The polygon is assumed to correspond to the plane that its vertices are on average closest to and that minimizes δG. $\triangle\, \mathbf{G}_i$ is the area of the polygon $\mathbf{G}_i$.

Geometry coverage (GC) is defined as equal to the percentage of the area of the ground truth geometry that is covered by the projections of the reconstructed polygons.

$$GC = \frac{(\bigcup_{i=1}^{K} \triangle P(\hat{\mathbf{G}}_\mathbf{i}, \mathbf{G}_\mathbf{i})) \bigcap (\bigcup_{i=1}^{K} \triangle \mathbf{G}_\mathbf{i})}{\bigcup_{i=1}^{K} \triangle \mathbf{G}_\mathbf{1}} \times 100\%. \tag{6.8}$$

Coverage error (CE) is defined as the percentage of the area of the reconstructed polygons that does not coincide with the ground truth geometry. In order to calculate the CE, the reconstructed polygons are first projected onto the ground truth planes.

$$CE = \frac{(\bigcup_{i=1}^{\hat{K}} \triangle P(\hat{\mathbf{G}}_\mathbf{i}, \mathbf{G}_\mathbf{i})) \setminus (\bigcup_{i=1}^{K} \triangle \mathbf{G}_\mathbf{i})}{\bigcup_{i=1}^{\hat{K}} \triangle \hat{\mathbf{G}}_\mathbf{i}} \times 100\%. \tag{6.9}$$

A parameter is also derived for evaluating the reconstructed percentage of the 2D perimeter of the environment. This metric can be used to evaluate 2D maps. Here the assumption, that all of the polygons in the geometry are horizontal, that is parallel to the ceiling and floor, or vertical, that is perpendicular to the ceiling and floor, is made. First the reconstructed polygons are projected onto the ground truth planes. Afterwards, the ground truth and reconstructed geometry is projected onto the floor plane. The perimeter lengths and the coverage percentage are calculated.

$$perimeter = \frac{Perimeter(\bigcup_{i=1}^{\hat{K}} \hat{\mathbf{G}}_\mathbf{i} \bigcap \bigcup_{i=1}^{K} \mathbf{G}_\mathbf{i})}{Perimeter(\bigcup_{i=1}^{K} \boldsymbol{G}_{\boldsymbol{i}})} \times 100\%, \tag{6.10}$$

where $Perimeter(\mathbf{G}_i)$ is the length of the line created by projecting the vertical polygon $\mathbf{G}_i$ onto the floor-plane. Polygons that are parallel to the floor or ceiling are not considered in the perimeter calculation.

6.4.4 Simulation Results

The simulation setup used to evaluate the proposed scheme is detailed in Section 3.5. The locations of the reference fingerprints are shown in Figure 6.12. Only the multipath components with at most one reflection are included in the simulation.

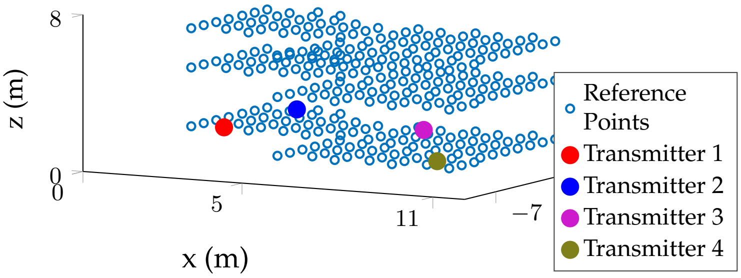


Figure 6.12: Reference points, $z = \{1m, 4m, 6m\}$

A performance comparison to the state-of-the-art is not possible as the necessary data is not provided in the relevant publications [136] and [259].

6.4.4.1 Performance Evaluation

The polygon reconstruction algorithm is first tested with ground-truth VT positions. Figure 6.13 shows the polygons reconstructed from ground-truth VT positions and points with $z = 4m$ in Fig. 6.12. The polygons are shown in the same color as their corresponding VTs, the reference locations are colored in blue. The floor and ceiling polygons are not shown. Figure 6.14 shows the polygons that were reconstructed when all of the simulated reference points shown in Fig. 6.12, with different z coordinate values, were used. The reconstructed area is much higher in this case. However, it can be seen that one of the polygons colored in red is reconstructed incorrectly. The quality of the 2D perimeter reconstruction, i.e., the projections of the reconstructed polygons onto the xy-plane is similar in both experiments.

Figure 6.15 shows the result of combining the proposed VT and polygon estimation algorithms. All of the reference points are used. The parameter γ in Algorithm 9 was set to 1.5. The figure shows a reconstruction quality comparable to when the ground-truth VTs are used. A numerical evaluation of the proposed algorithm is shown Table 6.3. The geometry error δG, geometry coverage GC, coverage error CE and $perimeter$ are calculated for different configurations of the algorithm input data and parameters. In some experiments the complete set of reference points and in others subsets of points ($z = \{1m, 4m, 6m\}$) were used. The values in the top two rows were obtained using ground-truth VT positions. For all other rows the VT estimation algorithm was used. The original reconstruction quality is compared to the case when the vertical polygons are extended between the floor and the ceiling. The reconstruction metric values are plotted for different values of γ in Figure 6.16. In that case the complete set of reference fingerprints and ceiling to floor polygon extension are used. The results show a general trade-off between the total percentage of geometry that is reconstructed and the percentage of geometry that is reconstructed incorrectly in areas which do not have ground-truth polygons. The higher the geometry coverage, the higher the geometry error and percentage of covered doorways and windows and the higher the deviation between the true and reconstructed planes, δG, will be. This trade-off is controlled by the VT filtering threshold γ. When γ is high, more candidate VTs, including more correct VTs, are

Data Set	γ	Perimeter↑	$\delta G \downarrow$	$GC \uparrow$	$CE \downarrow$	$\delta G \downarrow$	$GC \uparrow$	$CE \downarrow$
		Original reconstruction				Polygons extended		
Ground-truth VTs, $z = 1$	-	59.2%	0	10.4%	0%	1.3×10^{-17}	43.9%	$3.4 \times 10^{-13}\%$
Ground-truth VTs, $z = 4$	-	73.4%	1.5×10^{-17}	14.1%	0.56%	2.7×10^{-17}	44.0%	5.4%
Ground-truth VTs, complete reference data	-	94.9%	1.7×10^{-17}	29.2%	4.1%	1.7×10^{-17}	54.0%	8.9%
$z = 1$	1.5	58.1%	58.1%	$12.7\%\%$	1.1%	0.074	42.7%	46.2%
Complete reference data	0.5	81.2%	0.26	28.7%	0.72%	0.28	66.0%	42.5%
Complete reference data	1.5	78.9%	0.00078	25.8%	0.46%	0.0013	49.8%	1.3%
Complete reference data	2	89.5%	0.00068	28.2%	0.46%	0.00088	52.2%	1.41%
Complete reference data	3	71.2%	0.00039	17.5%	0.15%	0.00066	28.9%	0.14%

Table 6.3: Reconstruction performance. The performance of the algorithm is better when metrics marked with ↑ are maximized and metrics marked with ↓ are minimized. Table adapted from [5] ©2019IEEE.

filtered out. When γ is low, more correct and incorrect VTs are kept. The total reconstructed area increases when polygons are extended between the floor and ceiling. However, in this case, doors and windows may be covered by the reconstructed geometry. The reconstructed area also significantly increases when reference points located at different height levels are used. In this case the geometry error does not increase.

It should be noted that the placement of APs in the indoor geometry has a large influence on reconstruction quality. When the reference points and the APs are on the same height level only the perimeter can be reconstructed and not the polygons. When the APs are close

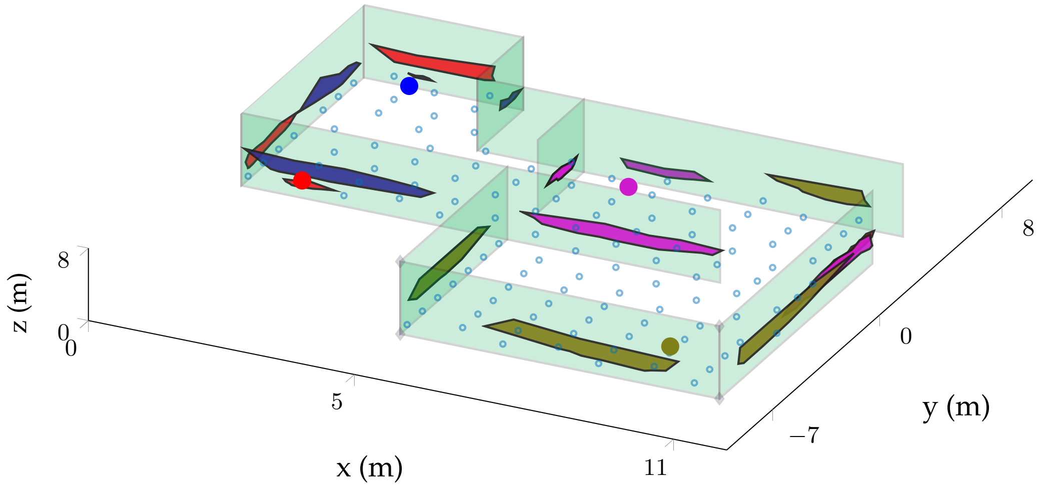


Figure 6.13: Reference data with $z = 4m$, perfect VTs. Figure adapted from [5] ©2019 IEEE.

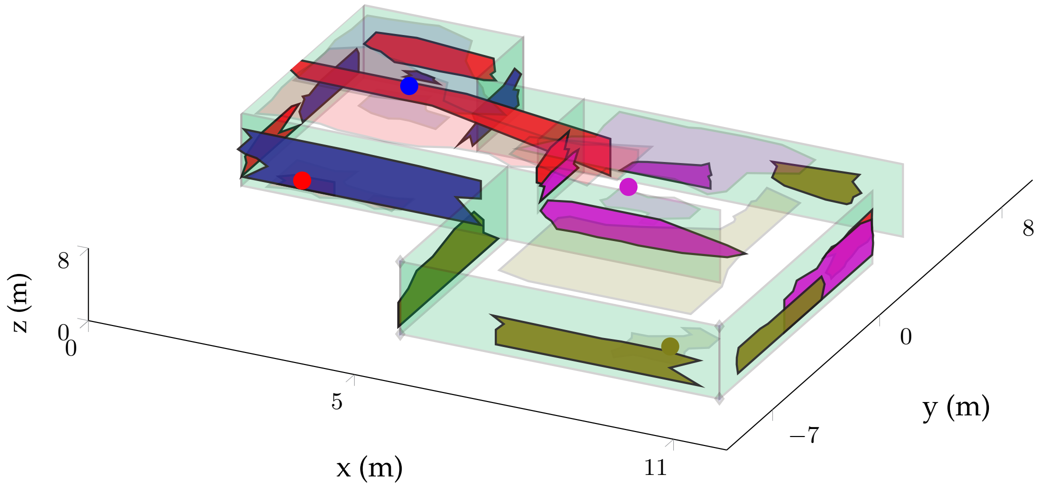


Figure 6.14: Full reference data, perfect VTs. Figure adapted from [5] ©2019 IEEE.

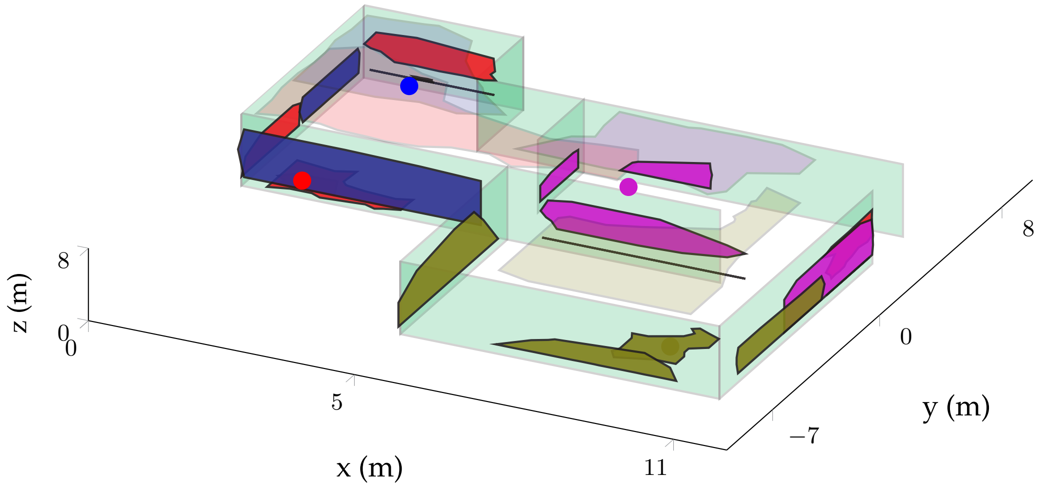


Figure 6.15: Full reference data with VT reconstruction. Figure adapted from [5] 2019 IEEE.

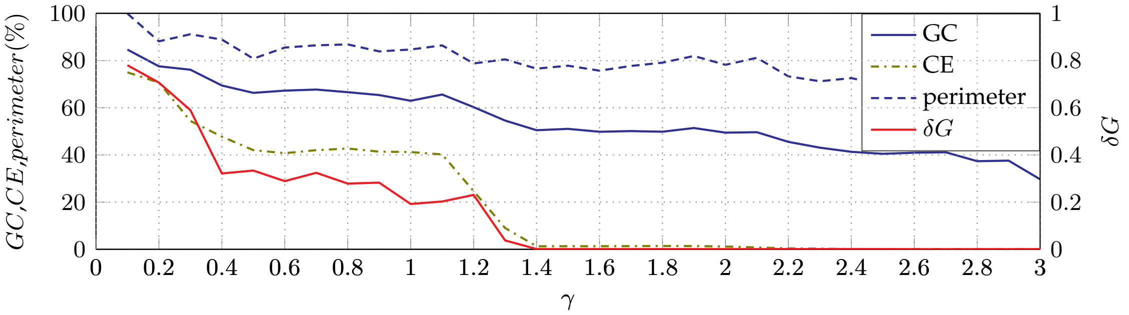


Figure 6.16: Performance metrics over the different values of the parameter γ. Complete multipath data, polygon extension. The higher the value of γ the more candidate VTs are filtered out. Figure adapted from [5] ©2019 IEEE.

to the floor, the area of the ceiling polygons is large and of the floor polygons is very small. And vice versa when the reference points are very close to the ceiling. In order for corners of the room to be reconstructed, reference points and APs need to be located near those corners. In the current example, the corners next the transmitters are well reconstructed, in contrast there are less area is reconstructed next to corners which do not have APs nearby.

6.5 Chapter Summary

This chapter details a novel approach for interpolating the fingerprint map of and reconstructing the 3D indoor geometry from the fingerprint map of a RF multipath-based indoor localization system. The positions of virtual transmitters (VTs) are estimated for both those tasks. The interpolated fingerprints are calculated directly from the VT positions. The environment estimation algorithm first uses real and virtual transmitters to obtain planes in the indoor geometry. The multipath components stored in the reference fingerprints are then used to calculate the boundaries of the polygons that construct indoor geometry. Several metrics are proposed to evaluate the geometry reconstruction scheme. Both the interpolation and geometry reconstruction schemes are evaluated in a ray tracing simulation detailed in Section 3.5. A trade-off is found between the area of the geometry that is reconstructed and the area of the incorrectly reconstructed geometry. The results show that up to 66% of the 3D geometry and up to 81% of the 2D perimeter is reconstructed by the proposed algorithm with the given AP locations and reference points.

Chapter 7

Conclusions and Future Work

This thesis provided an overview of the current state-of-the-art in the field of indoor localization in Chapter 2 and presented the following novel algorithms. Chapter 3 presented the multipath component analysis (MCA) localization algorithm. The advantages of the MCA algorithm are that, unlike conventional fingerprinting algorithms, it is able to maintain a decimeter-level localization precision in a dynamic NLoS environment with moving objects, and that, as a fingerprinting algorithm, it requires much fewer transmitters and access points (AP) and infrastructure than the high-precision ranging localization systems currently on the market. Chapter 4 presented a user privacy protection scheme. The novel scheme allows the user to obtain his or her location from the ILS server without the server knowing what that location is. An advantage of the algorithm is that the user is able to choose the degree of privacy and the corresponding computational load of the scheme. Chapter 5 presented an efficient fingerprint database lookup scheme. The scheme allows the user to obtain a rough, iteratively improving location estimate during the algorithm's run-time. Chapter 6 presented a scheme that obtains the locations of virtual transmitters and uses them for fingerprint map interpolation and extrapolation and environment reconstruction.

All of the novel algorithms above are made possible by the use of multipath delay profile (MDP) fingerprints. An MDP fingerprint allows individual multipath components to be processed and analyzed. Because of this, it is possible to identify multipath components in a fingerprint that correspond to moving objects or any other changes that occurred in the environment after the collection of the ILS calibration data, and to exclude those multipath components from the location calculation. The proposed approach also makes it possible to send the ILS server a subset of multipath components in the query fingerprint and artificially create fake partial fingerprints which contain only a small number of multipath components, match multiple locations in the indoor environment, and therefore, are seen as 'believable' by the ILS. The use of MDP fingerprints also allows for an iterative fingerprint map query algorithm that processes one multipath component of the measured fingerprint at a time and refines the localization estimate at each iteration. In contrast to other fingerprint structures, the use of MDP fingerprints also makes it possible to group the multipath components measured at different locations that are reflected from the same surfaces in the same order together for interpolation and virtual transmitter estimation.

All of the proposed algorithms were validated in a ray-tracing simulation. A prototype was also created to test the performance of the MCA algorithm. The UWB communication standard and the Decawave DWM1000 board were used. An average localization error of 6cm was obtained for a slow moving receiver when 9 APs were used in the 1 room test environment and a 70cm error was obtained when only 1 AP was used. For a fast moving receiver an error of 30cm was obtained for 9APs and 1.5m for 1 AP.

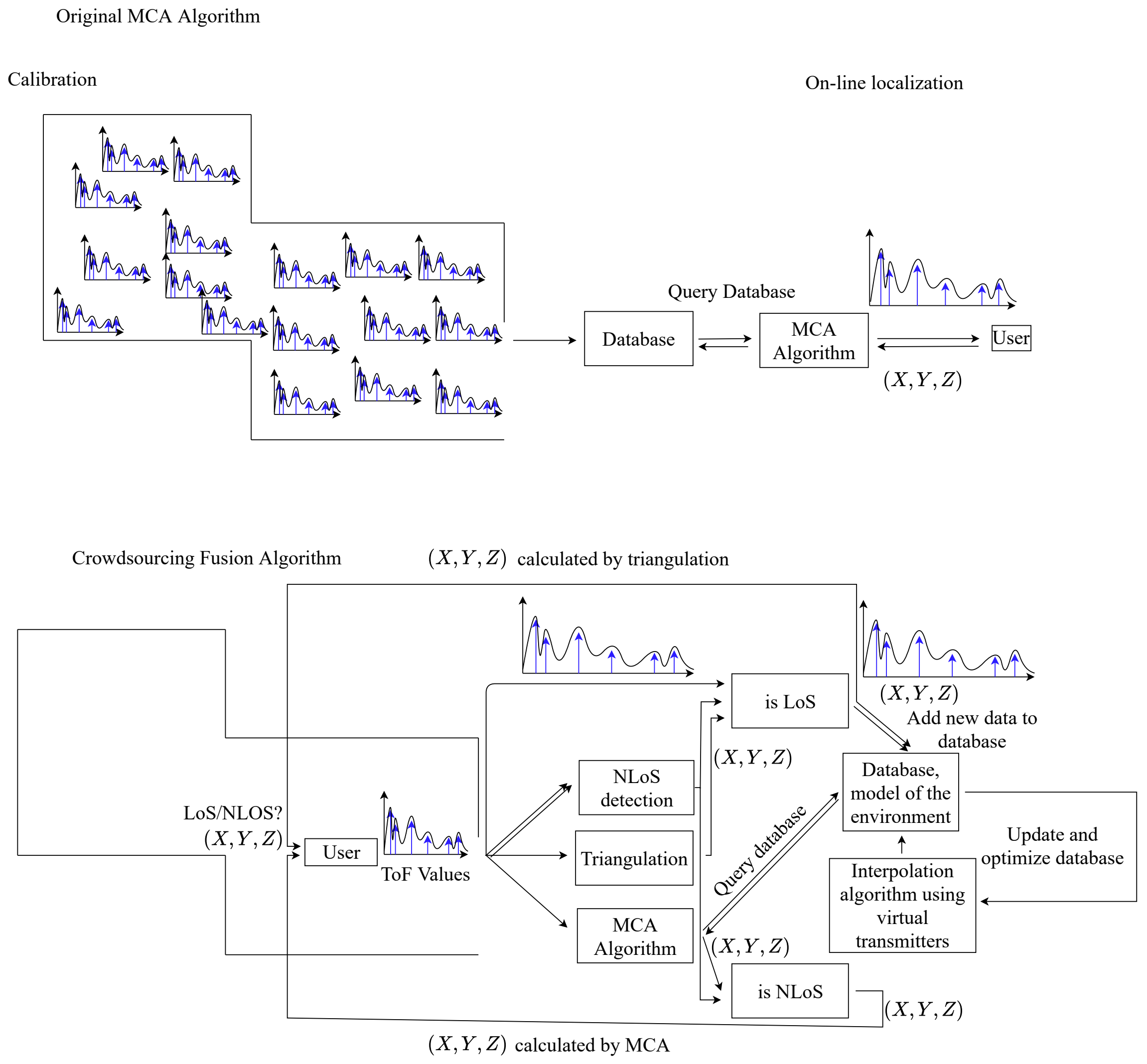


Figure 7.1: Top: original MCA system. Bottom: example crowdsourcing-based MCA system with efficient database management. If a LoS is detected, the measured MDP is added to the database. If NLoS is detected the MCA algorithm is used. Interpolation is used to extend and manage the database.

The potential continuation of the work in this thesis is mainly comprised of creating a more efficient hardware prototype for the system. Each of the algorithms was implemented and validated separately. A commercial prototype would need to integrate of the above algorithms into one system. Such as system would have the following requirements:

- Perform channel and multipath estimation as accurately as possible. This means that the amount of noise present in the calculated multipath components needs to be as low as possible and the multipath resolution needs to be as high as possible. It should be noted that the multipath resolution is connected to the bandwidth used by the communication protocols [42], [146]–[148].

- Use a communication standard and frequency range that produces the maximum number of reflections. The more reflections and multipath delays are present in the received signals, the more accurate and robust the MCA algorithm will be. Different materials absorb different amounts of the power of a signal at different frequencies and thus reflect the signals with different power attenuation. Therefore, the number of detected multipath components will depend on the chosen communications standard, the bandwidth and carrier frequencies, as well as the multipath estimation scheme.

- The APs and ILS server need to be able communicate with a large number of users simultaneously. The communication protocols need to be adjusted accordingly.

- The multipath estimation and MCA algorithm need be to run in real time. This means that the communication between the user and the ILS server needs to be low-latency and fast. The computational load on the receiver should also be kept low.

- The proposed privacy protection, efficient database lookup algorithm, VT reconstruction, interpolation and de-noising need to be integrated into a single system.

- Filters such as the k-means, the Kalman filter, algorithms that use IMU data need to be added to the ILS.

- The following fusion scheme can be implemented to increase the localization accuracy. When a line-of-sight (LoS) is detected to the transmitters, ranging can be used to calculate the user's location. If a LoS is detected, a crowdsourcing-like scheme can be also used to add the calculated location and measured MDP into the fingerprinting database. An schematic of such a fusion-based system is shown on Fig. 7.1. Instead of an explicit calibration step, triangulation-based coordinates and measured multipath profiles are added to the fingerprint map whenever a LoS is detected to enough transmitters. Interpolation is used to make the map more precise, and geometrical VT models are used for interpolation, to keep the size of the map small enough and to reduce noise.

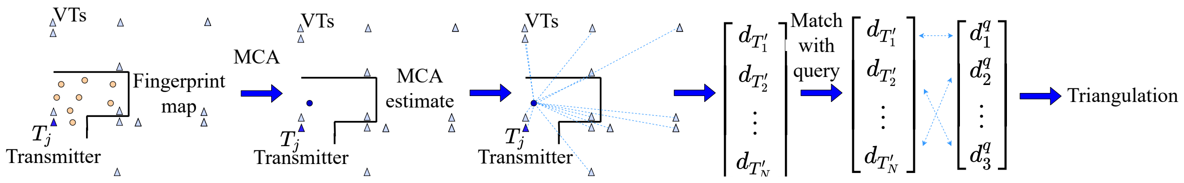


Figure 7.2: Possible combination of the MCA algorithm and VT triangulation.

The fusion of the localization scheme with an environment model also holds a lot of potential. In this thesis, the VTs are estimated from the fingerprint map and then used for interpolation and environment reconstruction. At the same time, a number of companies such as NavVis [288] scan an environment and create its digital model or digital twin. It is possible to use such a model to improve the performance of a localization system. If MDP fingerprints are used, as in this thesis, then the VTs can be extracted from the geometrical model. The VTs can then be used to interpolate or even fully create the fingerprint map. The challenge here would be to correctly model and compensate for the noise present in the measured query MDP and align it with the approximations present in the environment model and noise in the VT positions. An iterative algorithm that uses the VT positions to refine the results of the MCA algorithm can also be designed the following way. The MCA algorithm can be used to obtain an initial estimate of the user's location. The distances between that initial location estimate and the VTs can be stored in an MDP. This MDP can be compared with the query to determine which query multipath components correspond to which VTs. Triangulation can then be used to obtain the user's location from the VT coordinates and measured query multipath components. Such an algorithm can also be made iterative. The challenge here is to correctly design the matching procedure between the set of distances from the intermediate location estimate to the VTs and the multipath components in the measured multipath delay profile. Designing such an algorithm also remains part of the future work.

In summary, this thesis demonstrated the effectiveness of using individual multipath propagation delays for localization in dynamic and complex indoor environments. Moreover, it also demonstrated the potential of using multipath delays in various aspects of indoor localization such as computational efficiency, security and privacy which also motivates further research in of multipath-based indoor localization.

Bibliography

Publications by the author

Journal publications

[1] A. Zayets, C. Gentner, and E. Steinbach, "High-Precision Multipath-Based Indoor Localization Scheme With User Privacy Protection for Dynamic NLoS Environments", *IEEE Access*, vol. 9, pp. 116 033–116 049, 2021. DOI: 10.1109/ACCESS.2021.3104331.

Conference publications

[2] A. Zayets, E. Steinbach, "Low-Complexity Fingerprint Matching for Real-Time Indoor Localization Systems", in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7. DOI: 10.1109/GLOCOM.2018.8647529.

[3] ——, "Robust WiFi-based Indoor Localization using Multipath Component Analysis", *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, 2017. DOI: 10.1109/IPIN.2017.8115943.

[4] ——, "Interpolation and extrapolation of multipath fingerprints using virtual transmitter placement", in *IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7. DOI: 10.1109/ICC.2018.8422206.

[5] A. Zayets, M. Bourguiba, E. Steinbach, "3D reconstruction of indoor geometry using electromagnetic multipath fingerprints", *International Conference in Communications*, pp. 1–7, 2019. DOI: 10.1109/ICC.2019.8761636.

Patents

[6] E. Steinbach, A. Zayets, "Localization based on electromagnetic multipath delay profile", pat. EP3441782, Feb. 2021. [Online]. Available: https://patentscope.wipo.int/search/en/detail.jsf?docId=EP237380397.

General publications

[7] *Decawave website, link to dw1000 chip.* https://www.decawave.com/product/dwm1000-module/, Accessed: 2021-01-17.

[8] H. Hashemi, "The indoor radio propagation channel", *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, 1993, ISSN: 0018-9219. DOI: 10.1109/5.231342.

[9] D. R. Mautz, "Indoor Positioning Technologies", Habilitation thesis, ETH Zurich, 2012.

[10] F. Seco, A. R. Jimenez, C. Prieto, J. Roa, and K. Koutsou, "A survey of mathematical methods for indoor localization", in *2009 IEEE International Symposium on Intelligent Signal Processing*, 2009, pp. 9–14. DOI: 10.1109/WISP.2009.5286582.

[11] Z. Farid, R. Nordin, and M. Ismail, "Recent Advances in Wireless Indoor Localization Techniques and System", *Journal of Computer Networks and Communications*, vol. 2013, Sep. 2013. DOI: 10.1155/2013/185138.

[12] F. Furfari, A. Crivello, P. Barsocchi, F. Palumbo, and F. Potortì, "What is next for Indoor Localisation? Taxonomy, protocols, and patterns for advanced location based services", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pp. 1–8.

[13] M. Schmidhammer, B. Siebler, C. Gentner, S. Sand, and U.-C. Fiebig, "Bayesian Approaches to Multipath-Enhanced Device-Free Localization", in *2021 15th European Conference on Antennas and Propagation (EuCAP)*, 2021, pp. 1–5. DOI: 10.23919/EuCAP51087.2021.9411290.

[14] M. Schmidhammer, C. Gentner, and B. Siebler, "Localization of Discrete Mobile Scatterers in Vehicular Environments Using Delay Estimates", in *2019 International Conference on Localization and GNSS (ICL-GNSS)*, 2019, pp. 1–6. DOI: 10.1109/ICL-GNSS.2019.8752794.

[15] M. Cimdins, S. O. Schmidt, and H. Hellbrück, "MAMPI – Multipath-assisted Device-free Localization with Magnitude and Phase Information", in *2020 International Conference on Localization and GNSS (ICL-GNSS)*, 2020, pp. 1–6. DOI: 10.1109/ICL-GNSS49876.2020.9115529.

[16] R. Schmidt, "Multiple emitter location and signal parameter estimation", *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986. DOI: 10.1109/TAP.1986.1143830.

[17] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989. DOI: 10.1109/29.32276.

[18] D. R. Mautz, "Indoor Positioning Technologies", Habilitation thesis, ETH Zurich, 2012.

[19] A. H. Ismail, H. Kitagawa, R. Tasaki, and K. Terashima, "WiFi RSS fingerprint database construction for mobile robot indoor positioning system", in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 001 561–001 566. DOI: 10.1109/SMC.2016.7844461.

[20] J. Niu, B. Wang, L. Cheng, and J. J. P. C. Rodrigues, “WicLoc: An indoor localization system based on WiFi fingerprints and crowdsourcing”, in *IEEE International Conference on Communications (ICC)*, 2015, pp. 3008–3013. DOI: 10.1109/ICC.2015.7248785.

[21] C. Liu, A. Kiring, N. Salman, L. Mihaylova, and I. Esnaola, “A Kriging algorithm for location fingerprinting based on received signal strength”, *2015 Workshop on Sensor Data Fusion: Trends, Solutions, Applications, SDF 2015*, 2015. DOI: 10.1109/SDF.2015.7347695.

[22] I. Bisio, F. Lavagetto, A. Sciarrone, and S. Yiu, “A Smart2 Gaussian process approach for indoor localization with RSSI fingerprints”, in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7997089.

[23] X. Du, K. Yang, I. Bisio, F. Lavagetto, and A. Sciarrone, “An AP-centred smart probabilistic fingerprint system for indoor positioning”, in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6. DOI: 10.1109/ICC.2018.8422242.

[24] D. Bartlet, *Essentials of Positioning and Location Technology*. Cambridge University Press, 2013.

[25] Q. Song, S. Guo, X. Liu, and Y. Yang, “CSI Amplitude fingerprinting-based NB-IoT indoor localization”, *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1494–1504, 2018, ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2782479.

[26] X. Wang, L. Gao, S. Mao, and S. Pandey, “CSI-based fingerprinting for indoor localization: a deep learning approach”, *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2017, ISSN: 0018-9545. DOI: 10.1109/TVT.2016.2545523.

[27] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, “CSI-MIMO: Indoor Wi-Fi fingerprinting system”, in *39th Annual IEEE Conference on Local Computer Networks*, 2014, pp. 202–209. DOI: 10.1109/LCN.2014.6925773.

[28] C. Chen, Y. Chen, Y. Han, H. Q. Lai, and K. J. R. Liu, “Achieving Centimeter-Accuracy Indoor Localization on WiFi Platforms: A Frequency Hopping Approach”, *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 111–121, 2017, ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2628701.

[29] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, “CSI-based indoor localization”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2013, ISSN: 10459219. DOI: 10.1109/TPDS.2012.214.

[30] E. Kupershtein, M. Wax, and I. Cohen, “Single-Site Emitter Localization via Multipath Fingerprinting”, *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 10–21, 2013, ISSN: 1053-587X. DOI: 10.1109/TSP.2012.2222395.

[31] S. Chen, J. Fan, X. Luo, and Y. Zhang, “Multipath-based CSI fingerprinting localization with a machine learning approach”, in *Wireless Advanced (WiAd)*, 2018, pp. 1–5. DOI: 10.1109/WIAD.2018.8588448.

[32] L. Chen, X. Yang, P. X. Liu, and C. Li, “A novel outlier immune multipath fingerprinting model for indoor single-site localization”, *IEEE Access*, vol. 7, pp. 21 971–21 980, 2019. DOI: 10.1109/ACCESS.2019.2899169.

[33] T. Lo, J. Litva, and H. Leung, "A new approach for estimating indoor radio propagation characteristics", *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 10, pp. 1369–1376, 1994. DOI: 10.1109/8.320742.

[34] M. Haardt and J. A. Nossek, "Unitary ESPRIT: how to obtain increased estimation accuracy with a reduced computational burden", *IEEE Transactions on Signal Processing*, vol. 43, no. 5, pp. 1232–1242, 1995. DOI: 10.1109/78.382406.

[35] A. van der Veen, M. C. Vanderveen, and A. J. Paulraj, "Joint angle and delay estimation using shift-invariance properties", *IEEE Signal Processing Letters*, vol. 4, no. 5, pp. 142–145, 1997. DOI: 10.1109/97.575559.

[36] T. Lindner, S. Hafenecker, N. Hadaschik, and J. Thielecke, "A practical evaluation of joint angle and delay estimation", in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–4. DOI: 10.1109/IPIN.2015.7346780.

[37] M. D. Zoltowski, M. Haardt, and C. P. Mathews, "Closed-form 2-D angle estimation with rectangular arrays in element space or beamspace via unitary ESPRIT", *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 316–328, 1996. DOI: 10.1109/78.485927.

[38] L. Zhao, J. Xu, J. Ding, A. Liu, and L. Li, "Direction-of-arrival estimation of multipath signals using independent component analysis and compressive sensing", *PLOS ONE*, vol. 12, no. 7, pp. 1–17, Jul. 2017. DOI: 10.1371/journal.pone.0181838. [Online]. Available: https://doi.org/10.1371/journal.pone.0181838.

[39] B. H. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. I. Pedersen, "Channel parameter estimation in mobile radio environments using the SAGE algorithm", in *IEEE J. Sel. Areas Commun.,17(3):434–450*, 1999.

[40] M. N. de Sousa and R. S. Thomä Electronic, "Enhanced localization systems with multipath fingerprints and machine learning", in *IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–6.

[41] Y. Luo and C. L. Law, "A Least-Squares Algorithm for Multipath Estimation Using an UWB-IR Link", in *2010 IEEE Wireless Communication and Networking Conference*, 2010, pp. 1–6. DOI: 10.1109/WCNC.2010.5506321.

[42] W. Gifford, W. W.-L. Li, Y. Zhang, and M. Win, "Effect of Bandwidth on the Number of Multipath Components in Realistic Wireless Indoor Channels", Jul. 2011, pp. 1 –6. DOI: 10.1109/icc.2011.5962896.

[43] C. Gentner, T. Jost, W. Wang, S. Zhang, A. Dammann, and U. Fiebig, "Multipath assisted positioning with simultaneous localization and mapping", *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6104–6117, 2016, ISSN: 1536-1276. DOI: 10.1109/TWC.2016.2578336.

[44] S. Wielandt, B. Thoen, and L. De Strycker, "Experimental evaluation of a single anchor multipath assisted indoor angle of arrival localization system in the 2.4 GHz and 5 GHz band", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–7.

[45] A. Jaffe and M. Wax, “Single-Site Localization via Maximum Discrimination Multipath Fingerprinting”, *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1718–1728, 2014. DOI: 10.1109/TSP.2014.2304923.

[46] S. Ayvasik, H. M. Gürsu, and W. Kellerer, “Veni Vidi Dixi: Reliable Wireless Communication with Depth Images”, *CoRR*, vol. abs/1912.01879, 2019. arXiv: 1912.01879. [Online]. Available: http://arxiv.org/abs/1912.01879.

[47] C. Gentner, M. Ulmschneider, R. Karásek, and A. Dammann, “Simultaneous Localization of a Receiver and Mapping of Multipath Generating Geometry in Indoor Environments”, in *2021 IEEE Radar Conference (RadarConf21)*, 2021, pp. 1–6. DOI: 10.1109/RadarConf2147009.2021.9455157.

[48] C. Yang and H. r. Shao, “WiFi-based indoor positioning”, *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015, ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7060497.

[49] Y. Jin, N. O’Donoughue, and J. M. F. Moura, “Position location by time reversal in communication networks”, in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 3001–3004. DOI: 10.1109/ICASSP.2008.4518281.

[50] W. Zhang, R. Sengupta, J. Fodero, and X. Li, “DeepPositioning: intelligent fusion of pervasive magnetic field and WiFi fingerprinting for smartphone indoor localization via deep learning”, in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 7–13. DOI: 10.1109/ICMLA.2017.0-185.

[51] T. Qiao, Y. Zhang, and H. Liu, “Nonlinear Expectation Maximization Estimator for TDOA Localization”, *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 637–640, 2014. DOI: 10.1109/LWC.2014.2364023.

[52] T. Xie, C. Zhang, and Z. Wang, “Wi-Fi TDoA indoor localization system based on SDR platform”, in *2017 IEEE International Symposium on Consumer Electronics (ISCE)*, 2017, pp. 82–83. DOI: 10.1109/ISCE.2017.8355557.

[53] V. Djaja-Josko, J. Kolakowski, and J. Modelski, “TDOA estimation using a pair of synchronized DW1000 based anchor nodes”, in *2018 22nd International Microwave and Radar Conference (MIKON)*, 2018, pp. 57–60. DOI: 10.23919/MIKON.2018.8405288.

[54] F. Bonnin-Pascual and A. Ortiz, “An UWB-based System for Localization inside Merchant Vessels”, in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1559–1562. DOI: 10.1109/ETFA.2019.8869511.

[55] L. Zang, C. Shen, K. Zhang, L. Xu, and Y. Chen, “Research on Hybrid Algorithm Based on TDOA”, in *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, 2020, pp. 539–542. DOI: 10.1109/ICCT50939.2020.9295853.

[56] F. Lemic, A. Behboodi, V. Handziski, and A. Wolisz, “Experimental decomposition of the performance of fingerprinting-based localization algorithms”, in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 355–364. DOI: 10.1109/IPIN.2014.7275503.

[57] S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. Wang, "Indoor fingerprint positioning based on Wi-Fi: an overview", *ISPRS Int. J. Geo-Information*, vol. 6, p. 135, 2017.

[58] E. Kupershtein, M. Wax, and I. Cohen, "Single-Site Emitter Localization via Multipath Fingerprinting", *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 10–21, 2013, ISSN: 1053-587X. DOI: 10.1109/TSP.2012.2222395.

[59] S. Garcia-Villalonga and A. Perez-Navarro, "Influence of human absorption of Wi-Fi signal in indoor positioning with Wi-Fi fingerprinting", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–10. DOI: 10.1109/IPIN.2015.7346778.

[60] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system", in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2, 2000, 775–784 vol.2. DOI: 10.1109/INFCOM.2000.832252.

[61] T. Van Haute, E. De Poorter, P. Crombez, F. Lemic, V. Handziski, N. Wirström, A. Wolisz, T. Voigt, and I. Moerman, "Performance analysis of multiple Indoor Positioning Systems in a healthcare environment", *International Journal of Health Geographics*, vol. 15, no. 1, p. 7, 2016, ISSN: 1476-072X. DOI: 10.1186/s12942-016-0034-z. [Online]. Available: https://doi.org/10.1186/s12942-016-0034-z.

[62] Y. Shu, Y. Huang, J. Zhang, P. Coué, P. Cheng, J. Chen, and K. G. Shin, "Gradient-Based Fingerprinting for Indoor Localization and Tracking", *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2424–2433, 2016, ISSN: 0278-0046. DOI: 10.1109/TIE.2015.2509917.

[63] K. Järvinen, H. Leppäkoski, E. Lohan, P. Richter, T. Schneider, O. Tkachenko, and Z. Yang, "PILOT: practical privacy-Preserving indoor localization using ouTsourcing", in *IEEE European Symposium on Security and Privacy (EuroS P)*, 2019, pp. 448–463.

[64] C. Nerguizian, C. Despins, and S. Affes, "Geolocation in mines with an impulse response fingerprinting technique and neural networks", *IEEE Transactions on Wireless Communications*, vol. 5, no. 3, pp. 603–611, 2006, ISSN: 1536-1276. DOI: 10.1109/TWC.2006.1611090.

[65] C. Nerguizian and V. Nerguizian, "Indoor Fingerprinting Geolocation using Wavelet-Based Features Extracted from the Channel Impulse Response in Conjunction with an Artificial Neural Network", in *2007 IEEE International Symposium on Industrial Electronics*, 2007, pp. 2028–2032. DOI: 10.1109/ISIE.2007.4374919.

[66] C. Chen, Y. Chen, Y. Han, H. Q. Lai, F. Zhang, and K. J. R. Liu, "Achieving Centimeter-Accuracy Indoor Localization on WiFi Platforms: A Multi-Antenna Approach", *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 122–134, 2017, ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2628713.

[67] E. R. Magsino, I. W. Ho, and Z. Situ, “The effects of dynamic environment on channel frequency response-based indoor positioning”, in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6. DOI: 10.1109/PIMRC.2017.8292442.

[68] P.-H. Tseng, Y.-C. Chan, Y.-J. Lin, D.-B. Lin, N. Wu, and T.-M. Wang, “Ray-Tracing-Assisted Fingerprinting Based on Channel Impulse Response Measurement for Indoor Positioning”, *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 5, pp. 1032–1045, 2017. DOI: 10.1109/TIM.2016.2622799.

[69] S. Sen, R. R. Choudhury, B. Radunović, and T. Minka, “You are facing the mona lisa: spot localization using phy layer information”, in *In Proceedings of the ACM MobiSys 2012, MobiSys ’12*, ACM, 2012, pp. 183–196.

[70] M. N. de Sousa and R. S. Thomä, “Localization of UAV in urban scenario using multipath exploiting TDoA fingerprints”, in *IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1394–1399.

[71] B. R. Phelan, E. H. Lenzing, and R. M. Narayanan, “Source localization using unique characterizations of multipath propagation in an urban environment”, in *IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2012, pp. 189–192.

[72] R. Mautz, “The challenges of indoor environments and specification on some alternative positioning systems”, in *6th Workshop on Positioning, Navigation and Communication*, 2009, pp. 29–36. DOI: 10.1109/WPNC.2009.4907800.

[73] M. Youssef and A. Agrawala, “The Horus WLAN location determination system”, Jan. 2005, pp. 205–218. DOI: 10.1145/1067170.1067193.

[74] M. Fazelinia, M. R. Daliri, and S. Ebadollahi, “Wi-Fi RSS-based Indoor Localization Using Reduced Features Second Order Discriminant Function”, in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, 2019, pp. 921–924. DOI: 10.1109/IranianCEE.2019.8786450.

[75] D. Yu and C. Li, “An Accurate WiFi Indoor Positioning Algorithm For Complex Pedestrian Environments”, *IEEE Sensors Journal*, pp. 1–1, 2021. DOI: 10.1109/JSEN.2021.3113376.

[76] S. Fang, T. Lin, and P. Lin, “Location fingerprinting In a decorrelated space”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 685–691, 2008, ISSN: 2326-3865. DOI: 10.1109/TKDE.2007.190731.

[77] Q. Chen and B. Wang, “FinCCM: Fingerprint crowdsourcing, clustering and matching for indoor subarea localization”, *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 677–680, 2015, ISSN: 2162-2345. DOI: 10.1109/LWC.2015.2482971.

[78] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks”, *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014, ISSN: 0036-8075. DOI: 10.1126/science.1242072. eprint: https://science.sciencemag.org/content/344/6191/1492.full.pdf. [Online]. Available: https://science.sciencemag.org/content/344/6191/1492.

[79] X. Guo, S. Zhu, L. Li, F. Hu, and N. Ansari, "Accurate WiFi localization by unsupervised fusion of extended candidate location set", *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2476–2485, 2019, ISSN: 2372-2541. DOI: 10.1109/JIOT.2018.2870659.

[80] J. B. Kristensen, M. Massanet Ginard, O. K. Jensen, and M. Shen, "Non-line-of-sight identification for UWB indoor positioning systems using support vector machines", in *2019 IEEE MTT-S International Wireless Symposium (IWS)*, 2019, pp. 1–3. DOI: 10.1109/IEEE-IWS.2019.8804072.

[81] S. Liu, R. De Lacerda, and J. Fiorina, "WKNN indoor Wi-Fi localization method using k-means clustering based radio mapping", in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–5. DOI: 10.1109/VTC2021-Spring51267.2021.9448961.

[82] J. Wang, X. Wang, J. Peng, J. G. Hwang, and J. G. Park, "Indoor Fingerprinting Localization Based on Fine-grained CSI using Principal Component Analysis", in *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2021, pp. 322–327. DOI: 10.1109/ICUFN49451.2021.9528612.

[83] K. Han, S. M. Yu, and S. Kim, "Smartphone-based indoor localization using Wi-Fi fine timing measurement", in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pp. 1–5. DOI: 10.1109/IPIN.2019.8911751.

[84] A. H. Salamah, M. Tamazin, M. A. Sharkas, and M. Khedr, "An enhanced WiFi indoor localization system based on machine learning", in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8. DOI: 10.1109/IPIN.2016.7743586.

[85] H. Sandhu, "Deep learning-based indoor localization using multipath delay profile data", B.S. Thesis, Technical University of Munich, Jan. 2019.

[86] B. Chidlovskii and L. Antsfeld, "Semi-supervised variational autoencoder for WiFi indoor localization", in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pp. 1–8. DOI: 10.1109/IPIN.2019.8911825.

[87] S. Jung and D. Han, "Automated construction and maintenance of Wi-Fi radio maps for crowdsourcing-based indoor positioning systems", *IEEE Access*, vol. 6, pp. 1764–1777, 2018, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2780243.

[88] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang, and N. El-Sheimy, "Deep reinforcement learning (DRL): Another perspective for unsupervised wireless localization", *IEEE Internet of Things Journal*, pp. 1–1, 2019, ISSN: 2372-2541. DOI: 10.1109/JIOT.2019.2957778.

[89] D. V. Le, N. Meratnia, and P. J. M. Havinga, "Unsupervised Deep Feature Learning to Reduce the Collection of Fingerprints for Indoor Localization Using Deep Belief Networks", in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–7. DOI: 10.1109/IPIN.2018.8533790.

[90] C. Hsu, Y. Chen, T. Juang, and Y. Wu, "An adaptive Wi-Fi indoor localization scheme using deep learning", in *2018 IEEE Asia-Pacific Conference on Antennas and Propagation (APCAP)*, 2018, pp. 132–133. DOI: 10.1109/APCAP.2018.8538191.

[91] A. Gassner, C. Musat, A. Rusu, and A. Burg, *OpenCSI: An Open-Source Dataset for Indoor Localization Using CSI-Based Fingerprinting*, 2021. arXiv: 2104.07963 [eess.SP].

[92] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-Based Multi-Antenna and Multi-Point Indoor Positioning Using Probability Fusion", *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021. DOI: 10.1109/TWC.2021.3109789.

[93] L. Hao, B. Huang, H. Hong, B. Jia, and W. Li, "A Channel Adaptive WiFi Indoor Localization Method based on Deep Learning", in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6. DOI: 10.1109/WCNC49053.2021.9417310.

[94] Z. Chen, M. I. AlHajri, M. Wu, N. T. Ali, and R. M. Shubair, "A Novel Real-Time Deep Learning Approach for Indoor Localization Based on RF Environment Identification", *IEEE Sensors Letters*, vol. 4, no. 6, pp. 1–4, 2020. DOI: 10.1109/LSENS.2020.2991145.

[95] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "UJIIndoorLoc: a new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems", in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 261–270. DOI: 10.1109/IPIN.2014.7275492.

[96] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and M. Youssef, "WiDeep: WiFi-based accurate and robust indoor localization system using deep learning", in *IEEE International Conference on Pervasive Computing and Communications (PerCom*, 2019, pp. 1–10. DOI: 10.1109/PERCOM.2019.8767421.

[97] B. Berruet, O. Baala, A. Caminada, and V. Guillet, "DelFin: A Deep Learning Based CSI Fingerprinting Indoor Localization in IoT Context", in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–8. DOI: 10.1109/IPIN.2018.8533777.

[98] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services", *IEEE Internet of Things Journal*, vol. 5, no. 2, 624–635, 2018, ISSN: 2327-4662. DOI: 10.1109/jiot.2017.2712560. [Online]. Available: http://dx.doi.org/10.1109/JIOT.2017.2712560.

[99] Z. E. Khatab, A. Hajihoseini, and S. A. Ghorashi, "A fingerprint method for indoor localization using autoencoder based deep extreme learning machine", *IEEE Sensors Letters*, vol. 2, no. 1, pp. 1–4, 2018, ISSN: 2475-1472. DOI: 10.1109/LSENS.2017.2787651.

[100] A. Szabo, T. Weiherer, and J. Bamberger, "Unsupervised learning of propagation time for indoor localization", in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, 2011, pp. 1–5. DOI: 10.1109/VETECS.2011.5956591.

[101] X. Wang, L. Gao, and S. Mao, "BiLoc: Bi-Modal Deep Learning for Indoor Localization With Commodity 5GHz WiFi", *IEEE Access*, vol. 5, pp. 4209–4220, 2017. DOI: 10.1109/ACCESS.2017.2688362.

[102] Y. Shu, Y. Huang, J. Zhang, P. Coué, P. Cheng, J. Chen, and K. G. Shin, "Gradient-based fingerprinting for indoor localization and tracking", *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2424–2433, 2016, ISSN: 0278-0046. DOI: 10.1109/TIE.2015.2509917.

[103] P. Wilk, J. Karciarz, and J. Swiatek, "Indoor radio map maintenance by automatic annotation of crowdsourced Wi-Fi fingerprints", in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–8. DOI: 10.1109/IPIN.2015.7346933.

[104] S. Garcia-Villalonga and A. Perez-Navarro, "Influence of human absorption of Wi-Fi signal in indoor positioning with Wi-Fi fingerprinting", in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–10. DOI: 10.1109/IPIN.2015.7346778.

[105] J. Luo and X. Zhan, "Characterization of Smart Phone Received Signal Strength Indication for WLAN Indoor Positioning Accuracy Improvement", *JNW*, vol. 9, pp. 739–746, 2014.

[106] S. Bastola, "Study of spacial and temporal behavior of Wi-Fi channel state information, channel impulse response and multipath delay profile for use in indoor localization.", M.S. Thesis, Technical university of Munich, Oct. 2018.

[107] N. Yu, S. Zhao, X. Ma, Y. Wu, and R. Feng, "Effective Fingerprint Extraction and Positioning Method Based on Crowdsourcing", *IEEE Access*, vol. 7, pp. 162 639–162 651, 2019.

[108] X. Chai and Q. Yang, "Reducing the calibration effort for probabilistic indoor location estimation", *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 649–662, 2007, ISSN: 2161-9875. DOI: 10.1109/TMC.2007.1025.

[109] J. An, Z. Wang, X. He, X. Gui, J. Cheng, and R. Gui, "Know Where You Are: A Practical Privacy-Preserving Semi-Supervised Indoor Positioning via Edge-Crowdsensing", *IEEE Transactions on Network and Service Management*, pp. 1–1, 2021. DOI: 10.1109/TNSM.2021.3107718.

[110] C. Xiang, S. Zhang, S. Xu, and G. C. Alexandropoulos, "Self-Calibrating Indoor Localization with Crowdsourcing Fingerprints and Transfer Learning", in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500623.

[111] Y. Zhao, Z. Zhang, T. Feng, W.-C. Wong, and H. K. Garg, "GraphIPS: Calibration-Free and Map-Free Indoor Positioning Using Smartphone Crowdsourced Data", *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 393–406, 2021. DOI: 10.1109/JIOT.2020.3004703.

[112] H. Jiang, W. Liu, G. Jiang, Y. Jia, X. Liu, Z. Lui, X. Liao, J. Xing, and D. Liu, "Fly-Navi: A Novel Indoor Navigation System With On-the-Fly Map Generation", *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2820–2834, 2021. DOI: 10.1109/TMC.2020.2990446.

[113] E. Leitinger, F. Meyer, F. Tufvesson, and K. Witrisal, “Factor graph based simultaneous localization and mapping using multipath channel information”, in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017, pp. 652–658.

[114] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Z. Win, “A belief propagation algorithm for multipath-based SLAM”, *IEEE Transactions on Wireless Communications*, vol. 18, no. 12, pp. 5613–5629, 2019.

[115] M. Ulmschneider, S. Zhang, C. Gentner, and A. Dammann, “Multipath Assisted Positioning With Transmitter Visibility Information”, *IEEE Access*, vol. 8, pp. 155 210–155 223, 2020. DOI: 10.1109/ACCESS.2020.3017785.

[116] Y. Yu, R. Chen, L. Chen, X. Zheng, D. Wu, W. Li, and Y. Wu, “A Novel 3-D Indoor Localization Algorithm Based on BLE and Multiple Sensors”, *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9359–9372, 2021. DOI: 10.1109/JIOT.2021.3055794.

[117] Y. Zhong, T. Liu, B. Li, L. Yang, and L. Lou, “Integration of UWB and IMU for precise and continuous indoor positioning”, in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 2018, pp. 1–5. DOI: 10.1109/UPINLBS.2018.8559718.

[118] Y. Zhang, J. Chen, and W. Xue, “Unsupervised indoor localization based on Smartphone Sensors, iBeacon and Wi-Fi”, in *Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 2018, pp. 1–8. DOI: 10.1109/UPINLBS.2018.8559713.

[119] A. Ettlinger and G. Retscher, “Positioning using ambient magnetic fields in combination with Wi-Fi and RFID”, in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8. DOI: 10.1109/IPIN.2016.7743657.

[120] M. Driusso, C. Marshall, M. Sabathy, F. Knutti, H. Mathis, and F. Babich, “Indoor positioning using LTE signals”, in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8.

[121] H. Nurminen, T. Ardeshiri, R. Piché, and F. Gustafsson, “A NLOS-robust TOA positioning filter based on a skew-t measurement noise model”, in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–7. DOI: 10.1109/IPIN.2015.7346786.

[122] Z. Jiang, W. Xi, X. Li, S. Tang, J. Zhao, J. Han, K. Zhao, Z. Wang, and B. Xiao, “Communicating Is Crowdsourcing: Wi-Fi Indoor Localization with CSI-Based Speed Estimation”, *Journal of Computer Science and Technology*, vol. 29, pp. 589–604, 2014.

[123] Y. Jiang, X. Pan, K. Li, Q. Lv, R. Dick, M. Hannigan, and L. Shang, “ARIEL: Automatic Wi-Fi based room fingerprinting for indoor localization”, in *UbiComp’12 - Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 441–450, ISBN: 9781450312240. DOI: 10.1145/2370216.2370282.

[124] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No Need to War-drive: Unsupervised Indoor Localization", in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '12, Low Wood Bay, Lake District, UK: ACM, 2012, pp. 197–210, ISBN: 978-1-4503-1301-8. DOI: 10.1145/2307636.2307655. [Online]. Available: http://doi.acm.org/10.1145/2307636.2307655.

[125] O. Woodman and R. Harle, "Pedestrian Localisation for Indoor Environments", in *Proceedings of the 10th International Conference on Ubiquitous Computing*, ser. UbiComp '08, Seoul, Korea: ACM, 2008, pp. 114–123, ISBN: 978-1-60558-136-1. DOI: 10.1145/1409635.1409651. [Online]. Available: http://doi.acm.org/10.1145/1409635.1409651.

[126] O. Woodman, "Pedestrian localisation for indoor environments", Ph.D. dissertation, University of Cambridge, 2010.

[127] S. Beauregard, Widyawan, and M. Klepal, "Indoor PDR performance enhancement using minimal map information and particle filters", in *2008 IEEE/ION Position, Location and Navigation Symposium*, 2008, pp. 141–147. DOI: 10.1109/PLANS.2008.4570050.

[128] J. Zhang, P. V. Orlik, Z. Sahinoglu, A. F. Molisch, and P. Kinney, "UWB Systems for Wireless Sensor Networks", *Proceedings of the IEEE*, vol. 97, no. 2, pp. 313–331, 2009. DOI: 10.1109/JPROC.2008.2008786.

[129] V. Niemelä, J. Haapola, M. Hämäläinen, and J. Iinatti, "An Ultra Wideband Survey: Global Regulations and Impulse Radio Research Based on Standards", *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 874–890, 2017. DOI: 10.1109/COMST.2016.2634593.

[130] S. Soni, M. Shrivastava, and A. Agwekar, "A Detail Concept and Survey on Ultra Wide Band (UWB) Antennas", in *2nd International Conference on Data, Engineering and Applications (IDEA)*, 2020, pp. 1–7. DOI: 10.1109/IDEA49133.2020.9170657.

[131] A. R. Jiménez Ruiz and F. Seco Granja, "Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis", *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 2106–2117, 2017. DOI: 10.1109/TIM.2017.2681398.

[132] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies", *CoRR*, vol. abs/1709.01015, 2017. arXiv: 1709.01015. [Online]. Available: http://arxiv.org/abs/1709.01015.

[133] *infsoft website*, https://www.infsoft.com/, Accessed: 2020-01-16.

[134] *intel ultimate N Wifi link 5300 product brief*, https://www.intel.com/content/www/us/en/products/docs/wireless-products/ultimate-n-wifi-link-5300-brief.html, Accessed: 07-10-2019.

[135] Y. Kuang, K. Åström, and F. Tufvesson, "Single antenna anchor-free UWB positioning based on multipath propagation", in *2013 IEEE International Conference on Communications (ICC)*, 2013, pp. 5814–5818. DOI: 10.1109/ICC.2013.6655524.

[136] H. Naseri, J. Salmi, and V. Koivunen, "Indoor mapping using MIMO radio channel measurements", in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 3321–3325. DOI: 10.1109/ICASSP.2017.7952771.

[137] C. BASRI and A. E. Khadimi, "Survey on indoor localization system and recent advances of WIFI fingerprinting technique", in *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*, 2016, pp. 253–259. DOI: 10.1109/ICMCS.2016.7905633.

[138] A. A. Khudhar, S. Q. Jabbar, M. Q. Sulttan, and D. Wang, "A Survey of Indoor Localization Systems and Technologies", *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 3, no. 2, 2016.

[139] *Linux 802.11n CSI Tool*, https://dhalperi.github.io/linux-80211n-csitool/, Accessed: 07-10-2019.

[140] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: gathering 802.11N traces with channel state information", *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 53–53, 2011, ISSN: 0146-4833. DOI: 10.1145/1925861.1925870. [Online]. Available: http://doi.acm.org/10.1145/1925861.1925870.

[141] *Support information for intel IWL5300*, https://www.intel.com/content/www/us/en/support/products/70971/network-and-io/wireless-networking/legacy-intel-wireless-products/intel-wireless-series/intel-wifi-link-5300.html, Accessed: 07-10-2019.

[142] *Laptops that provide support for IWL5300 and IWL5100*, https://www.airmagnet.com/assets/List_of_Laptops_for_AirMedic.pdf, Accessed: 07-10-2019.

[143] *Intel WiFi Link 5100/5300 WLAN controller*, http://www.thinkwiki.org/wiki/Intel_WiFi_Link_5100/5300_WLAN_controller, Accessed: 07-10-2019.

[144] T. S. Ghazaany, S. Zhu, S. M. R. Jones, R. A. Alhameed, J. M. Noras, T. Van Buren, T. Sugget, and S. Marker, "Effect of the bandwidth on the accuracy of AOA estimation algorithms in a multipath environment", in *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, 2014, pp. 2217–2220. DOI: 10.1109/EuCAP.2014.6902251.

[145] Z. Tarique, W. Malik, and D. Edwards, "Effect of Bandwidth and Antenna Directivity on the Range Estimation Accuracy in a Multipath Environment", vol. 6, Jun. 2006, pp. 2887 –2890. DOI: 10.1109/VETECS.2006.1683396.

[146] S. Hinteregger, E. Leitinger, P. Meissner, J. Kulmer, and K. Witrisal, "Bandwidth dependence of the ranging error variance in dense multipath", in *24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 733–737.

[147] K. Witrisal, E. Leitinger, S. Hinteregger, and P. Meissner, "Bandwidth Scaling and Diversity Gain for Ranging and Positioning in Dense Multipath Channels", *IEEE Wireless Communications Letters*, vol. 5, no. 4, pp. 396–399, 2016.

[148] W.-J. Chang and J.-H. Tarng, "Effects of Bandwidth on Observable Multipath Clustering in Outdoor/Indoor Environments for Broadband and Ultrawideband Wireless Systems", *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 1913–1923, 2007. DOI: 10.1109/TVT.2007.897658.

[149] S. Adler, S. Schmitt, K. Wolter, and M. Kyas, "A survey of experimental evaluation in indoor localization research", in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–10. DOI: 10.1109/IPIN.2015.7346749.

[150] F. Potortì, A. Crivello, P. Barsocchi, and F. Palumbo, "Evaluation of indoor localisation systems: comments on the ISO/IEC 18305 standard", in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–7.

[151] *Vicon website*, http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm, Accessed: 04-10-2019.

[152] M. Salimibeni, Z. Hajiakhondi-Meybodi, P. Malekzadeh, M. Atashi, K. N. Plataniotis, and A. Mohammadi, "IoT-TD: IoT Dataset for Multiple Model BLE-based Indoor Localization/Tracking", in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1697–1701. DOI: 10.23919/Eusipco47968.2020.9287547.

[153] P. Spachos, *RSSI Dataset for Indoor Localization Fingerprinting*, 2020. DOI: 10.21227/jrk5-qw26. [Online]. Available: https://dx.doi.org/10.21227/jrk5-qw26.

[154] M. Kennedy, P. Spachos, and G. W. Taylor, *BLE beacon indoor localization dataset*, version V1, 2019. DOI: 10.5683/SP2/UTZTFT. [Online]. Available: https://doi.org/10.5683/SP2/UTZTFT.

[155] *UCI RSSI dataset*, https://archive.ics.uci.edu/ml/machine-learning-databases/00422/, Accessed: 23-10-2021.

[156] *UCI RSSI dataset*, https://archive.ics.uci.edu/ml/machine-learning-databases/00422/, Accessed: 23-10-2021.

[157] D. Hanley, A. B. Faustino, S. D. Zelman, D. A. Degenhardt, and T. Bretl, "MagPIE: A dataset for indoor positioning with magnetic anomalies", in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017, pp. 1–8. DOI: 10.1109/IPIN.2017.8115961.

[158] P. Barsocchi, A. Crivello, D. La Rosa, and F. Palumbo, "A multisource and multivariate dataset for indoor localization methods based on WLAN and geo-magnetic field fingerprinting", in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8. DOI: 10.1109/IPIN.2016.7743678.

[159] C. Laoudias, R. Piché, and C. Panayiotou, *KIOS WiFi RSS dataset*, Sep. 2013.

[160] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007, ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.905750.

[161] A. F. G. Ferreira, D. M. A. Fernandes, A. P. Catarino, and J. L. Monteiro, “Localization and Positioning Systems for Emergency Responders: A Survey”, *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2836–2870, 2017. DOI: 10.1109/COMST.2017.2703620.

[162] R. Harle, “A Survey of Indoor Inertial Positioning Systems for Pedestrians”, *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013, ISSN: 1553-877X. DOI: 10.1109/SURV.2012.121912.00075.

[163] R. Mautz and S. Tilch, “Survey of optical indoor positioning systems”, in *2011 International Conference on Indoor Positioning and Indoor Navigation*, 2011, pp. 1–7. DOI: 10.1109/IPIN.2011.6071925.

[164] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, “Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications”, *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2017. DOI: 10.1109/COMST.2016.2632427.

[165] Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks”, *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 13–32, 2009, ISSN: 1553-877X. DOI: 10.1109/SURV.2009.090103.

[166] S. He and S. H. G. Chan, “Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons”, *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 466–490, 2016, ISSN: 1553-877X. DOI: 10.1109/COMST.2015.2464084.

[167] K. P. Subbu, C. Zhang, J. Luo, and A. V. Vasilakos, “Analysis and status quo of smartphone-based indoor localization systems”, *IEEE Wireless Communications*, vol. 21, no. 4, pp. 106–112, 2014, ISSN: 1536-1284. DOI: 10.1109/MWC.2014.6882302.

[168] F. Potortì, S. Park, A. Jiménez, P. Barsocchi, M. Girolami, A. Crivello, S. Yeon Lee, J. Hyun Lim, J. Torres-Sospedra, F. Seco, R. Montoliu, G. Mendoza Silva, M. Del Carmen Pérez Rubio, C. Losada, F. Espinosa, and J. Macias-Guarasa, “Comparing the Performance of Indoor Localization Systems through the EvAAL Framework”, vol. 17, p. 2327, Oct. 2017.

[169] H. El-Sayed, G. Athanasiou, and C. Fischione, “Evaluation of localization methods in millimeter-wave wireless systems”, in *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2014, pp. 345–349. DOI: 10.1109/CAMAD.2014.7033263.

[170] M. Dombois and S. Döweling, “A pipeline architecture for indoor location tracking”, in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8. DOI: 10.1109/IPIN.2016.7743652.

[171] A. Mathisen, S. K. Sørensen, A. Stisen, H. Blunck, and K. Grønbæk, “A comparative analysis of Indoor WiFi Positioning at a large building complex”, in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8. DOI: 10.1109/IPIN.2016.7743666.

[172] Q. Song, Y. Gu, M. Ma, Y. Li, and Z. Zhou, "An anchor based framework for trajectory calibration in inertial pedestrian positioning systems", in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8. DOI: 10.1109/IPIN.2016.7743689.

[173] A. Stancovici, M. V. Micea, and V. Cretu, "Cooperative positioning system for indoor surveillance applications", in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–7. DOI: 10.1109/IPIN.2016.7743584.

[174] L. Mainetti, L. Patrono, and I. Sergi, "A survey on indoor positioning systems", in *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2014, pp. 111–120. DOI: 10.1109/SOFTCOM.2014.7039067.

[175] *VentureRadar website*, https://www.ventureradar.com/keyword/Indoor$%$20Positioning?, Accessed: 2020-01-15.

[176] *Cio applications website*, https://indoor-positioning.cioapplications.com/vendors/top-indoor-positioning-solution-companies.html, Accessed: 2020-01-15.

[177] *Steerpath website*, https://steerpath.com/, Accessed: 2020-01-14.

[178] *LocusLabs website*, https://locuslabs.com/, Accessed: 2020-01-14.

[179] *Terabee website*, https://www.terabee.com/, Accessed: 2020-01-15.

[180] *Cloundleaf website*, https://www.cloudleaf.com/, Accessed: 2020-01-15.

[181] *SmartPoint website*, https://www.smartpoint.co/, Accessed: 2020-01-15.

[182] *AiristaFlow website*, https://www.airistaflow.com/, Accessed: 2020-01-15.

[183] *Bluevision website*, https://www.hidglobal.com/, Accessed: 2020-01-15.

[184] *Connexient website*, https://www.connexient.com/, Accessed: 2020-01-15.

[185] *inpixon website*, https://inpixon.com/, Accessed: 2020-01-15.

[186] *Quuppa website*, https://quuppa.com/, Accessed: 2020-01-15.

[187] *Kontakt io website*, https://inpixon.com/, Accessed: 2020-01-15.

[188] *Kinexon website*, https://kinexon.com/, Accessed: 2022-01-29.

[189] *prozyx website*, https://www.pozyx.io/, Accessed: 2020-01-17.

[190] *proxicon.io website*, https://proxicon.io/, Accessed: 2020-01-15.

[191] *nanotron website*, https://nanotron.com/EN/, Accessed: 2020-01-21.

[192] *iviu technologies website*, http://www.iviutech.com/, Accessed: 2020-01-21.

[193] *Ubisense website*, https://ubisense.com/, Accessed: 2020-01-21.

[194] *BlueBotics*, https://www.bluebotics.com/, Accessed: 2020-01-14.

[195] *CloudNav website*, https://cloudnav.net/, Accessed: 2020-01-15.

[196] *IndoorAtlas website*, https://www.indooratlas.com/product-features/, Accessed: 2020-01-15.

[197] *GiPStech website*, http://www.gipstech.com/, Accessed: 2020-01-16.

[198] *Accerion website*, https://accerion.tech/, Accesssed: 2020-01-17.

[199] *locatible website*, https://locatible.com/, Acessed: 2020-01-21.

[200] *navenio website*, https://www.navenio.com/, Accessed: 2020-01-17.

[201] *insiteo website*, https://www.insiteo.com/plateforme, Accessed: 2020-01-17.

[202] *mapxus website*, http://www.mapxus.com/solutions/, Accessed: 2020-01-17.

[203] *combainmobile website*, https://combain.com/, Accessed: 2020-01-21.

[204] *sensewhere website*, http://www.sensewhere.com/, Accessed: 2020-01-21.

[205] *Pointr website*, https://www.pointr.tech/, Accessed: 2020-01-21.

[206] *Sirl website*, https://www.sirl.io/, Accessed: 2020-01-21.

[207] *situm website*, https://situm.es/en/technology, Accessed: 2020-01-21.

[208] *Nextome website*, https://www.nextome.net/, Accessed: 2020-01-21.

[209] *Lighhouse signal systems*, http://www.lighthousesignal.com/, Accessed: 2020-01-21.

[210] *arara website*, https://www.arara.ai/?lang=en, Acessed: 2020-01-21.

[211] *Navigine website*, https://navigine.com/de, Accessed: 2020-01-21.

[212] *indoors website*, https://indoo.rs/, Accessed: 2020-01-21.

[213] *witrac website*, https://witrac.es/, Accessed: 2020-01-17.

[214] *intranav website*, https://intranav.com/, Accessed: 2020-01-21.

[215] *sewio website*, https://www.sewio.net/, Accessed: 2020-01-21.

[216] *localino website*, https://www.localino.net/en/, Accessed: 2020-01-21.

[217] *safectory website*, https://safectory.com/, Accessed: 2020-01-21.

[218] P. Meissner, C. Steiner, and K. Witrisal, "UWB positioning with virtual anchors and floor plan information", in *2010 7th Workshop on Positioning, Navigation and Communication*, 2010, pp. 150–156. DOI: 10.1109/WPNC.2010.5650374.

[219] C. Gentner, P. Robert, M. Ulmschneider, T. Jost, and S. Zhang, "Positioning using terrestrial multipath signals and inertial sensors", *Mobile Information Systems*, 2017.

[220] R. S. Kathan, "Signal propagation modeling through room stitching for indoor localization applications", B.S. Thesis, Technical University of Munich, Jul. 2019.

[221] B. H. Fleury, D. Dahlhaus, R. Heddergott, and M. Tschudin, "Wideband angle of arrival estimation using the SAGE algorithm", in *Proceedings of ISSSTA'95 International Symposium on Spread Spectrum Techniques and Applications*, vol. 1, 1996, 79–85 vol.1. DOI: 10.1109/ISSSTA.1996.563747.

[222] A. P. Dempster and et al., "Maximum likelihood from incomplete data via the EM algorithm", *JSTOR, www.jstor.org/stable/2984875.*, vol. Vol. 39, no. 1, pp. 1–38. 1977.

[223] M. Feder and E. Weinstein, “Parameter estimation of superimposed signals using the EM algorithm”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 477–489, 1988. DOI: 10.1109/29.1552.

[224] D. Dahlhaus, B. H. Fleury, and A. Radović, “A sequential algorithm for joint parameter estimation and multiuser detection in DS/CDMA systems with multipath propagation”, *Wireless Personal Communications*, vol. 6, no. 1, pp. 161–178, 1998, ISSN: 1572-834X. DOI: 10.1023/A:1008848721645. [Online]. Available: https://doi.org/10.1023/A:1008848721645.

[225] K. I. Pedersen, B. H. Fleury, and P. E. Mogensen, “High resolution of electromagnetic waves in time-varying radio channels”, in *Proceedings of 8th International Symposium on Personal, Indoor and Mobile Radio Communications - PIMRC '97*, vol. 2, 1997, 650–654 vol.2. DOI: 10.1109/PIMRC.1997.631112.

[226] T. Jost, W. Wang, U. Fiebig, and F. Perez-Fontan, “Detection and tracking of mobile propagation channel paths”, *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 10, pp. 4875–4883, 2012.

[227] W. Wang, T. Jost, U. Fiebig, and C. Gentner, “Modeling three different types of multipath components for mobile radio channel”, in *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, 2014, pp. 3065–3069.

[228] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, “A Game-Theoretic Approach for Achieving k -Anonymity in Location Based Services”, *Proceedings - IEEE INFOCOM*, pp. 2985–2993, 2013.

[229] Y. Wang, F. Li, and B. Xu, “L2P2 : Location-aware Location Privacy Protection for Location-based Services”, *Proceedings - IEEE INFOCOM*, pp. 1996–2004, 2012.

[230] D. Yang, “Truthful Incentive Mechanisms for K-Anonymity Location Privacy”, *Proceedings - IEEE INFOCOM*, pp. 2994–3002, 2013.

[231] M Gruteser, M Gruteser, D Grunwald, and D Grunwald, “Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis”, *Mobile Networks and Applications*, vol. 10, no. 3, pp. 315–325, 2005. DOI: 10.1007/s11036.

[232] A. R. Beresford and F. Stajano, “Location Privacy in Pervasive Computing”, *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, Jan. 2003, ISSN: 1536-1268. DOI: 10.1109/MPRV.2003.1186725. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2003.1186725.

[233] V. Sadhu, D. Pompili, S. Zonouz, and V. Sritapan, “CollabLoc: Privacy-preserving multi-modal localization via collaborative information fusion”, in *26th International Conference on Computer Communication and Networks (ICCCN)*, 2017, pp. 1–9. DOI: 10.1109/ICCCN.2017.8038390.

[234] P. Zhao, H. Jiang, J. C. S. Lui, C. Wang, F. Zeng, F. Xiao, and Z. Li, “P3-LOC: A Privacy-preserving paradigm-driven framework for indoor localization”, *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2856–2869, 2018, ISSN: 1063-6692. DOI: 10.1109/TNET.2018.2879967.

[235] N. Alikhani, V. Moghtadaiee, A. M. Sazdar, and S. A. Ghorashi, “A Privacy Preserving Method for Crowdsourcing in Indoor Fingerprinting Localization”, in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2018, pp. 58–62. DOI: 10.1109/ICCKE.2018.8566402.

[236] M. Zhou, Y. Liu, W. Nie, L. Xie, and Z. Tian, “Secure mobile crowdsourcing for WLAN indoor localization”, in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 480–485. DOI: 10.1109/INFCOMW.2018.8406947.

[237] H. Li, L. Sun, H. Zhu, X. Lu, and X. Cheng, “Achieving privacy preservation in WiFi fingerprint-based localization”, *Proceedings - IEEE INFOCOM*, pp. 2337–2345, 2014, ISSN: 0743166X. DOI: 10.1109/INFOCOM.2014.6848178.

[238] R. Zhang, J. Zhang, Y. Zhang, and S. Member, “Privacy-Preserving Profile Matching for Proximity-Based Mobile Social Networking”, vol. 31, no. 9, pp. 656–668, 2013.

[239] S. Li, H. Li, and L. Sun, “Privacy-preserving crowdsourced site survey in WiFi fingerprint-based localization”, *EURASIP Journal on Wireless Communications and Networking*, 2016, ISSN: 1687-1499. DOI: 10.1186/s13638-016-0624-2. [Online]. Available: http://dx.doi.org/10.1186/s13638-016-0624-2.

[240] X Wang, Y Liu, Z Shi, X Lu, and L Sun, “A Privacy-Preserving Fuzzy Localization Scheme with CSI Fingerprint”, *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2015. DOI: 10.1109/GLOCOM.2015.7417168.

[241] Z. Yang and K. Järvinen, “The death and rebirth of privacy-preserving WiFi fingerprint localization with Paillier encryption”, in *IEEE INFOCOM - IEEE Conference on Computer Communications*, 2018, pp. 1223–1231. DOI: 10.1109/INFOCOM.2018.8486221.

[242] L. Xiang, B. Li, and B. Li, “Privacy-preserving inference in crowdsourcing systems”, in *IEEE Conference on Communications and Network Security (CNS)*, 2017, pp. 1–9. DOI: 10.1109/CNS.2017.8228623.

[243] T. Shu, Y. Chen, and J. Yang, “Protecting Multi-Lateral Localization Privacy in Pervasive Environments”, *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1688–1701, 2015. DOI: 10.1109/TNET.2015.2478881.

[244] Y. Wang, M. Huang, Q. Jin, and J. Ma, “DP3: A differentialprivacy-based privacy-preserving indoor localization mechanism”, *IEEE Communications Letters*, vol. 22, no. 12, pp. 2547–2550, 2018, ISSN: 1089-7798. DOI: 10.1109/LCOMM.2018.2876449.

[245] Y. Zhu, Y. Wang, Q. Liu, Y. Liu, and P. Zhang, “WiFi fingerprint releasing for indoor localization based on differential privacy”, in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6. DOI: 10.1109/PIMRC.2017.8292470.

[246] A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, and Y. Theodoridis, "Privacy-preserving indoor localization on smartphones", *2016 IEEE 32nd International Conference on Data Engineering, ICDE 2016*, vol. 27, no. 11, pp. 1470–1471, 2016, ISSN: 10414347. DOI: 10.1109/ICDE.2016.7498379.

[247] H. Singh, S. Sarkar, A. Dimri, A. Bhaskara, N. Patwari, S. Kasera, S. Ramirez, and K. Derr, "Privacy enabled crowdsourced transmitter localization using adjusted measurements", in *IEEE Symposium on Privacy-Aware Computing (PAC)*, 2018, pp. 95–106. DOI: 10.1109/PAC.2018.00016.

[248] P. Zhao, W. Liu, G. Zhang, Z. Li, and L. Wang, "Preserving Privacy in WiFi Localization With Plausible Dummy Locations", *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 909–11 925, 2020. DOI: 10.1109/TVT.2020.3006363.

[249] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems", *IEEE Control Systems*, vol. 21, no. 1, pp. 84–99, 2001, ISSN: 1066-033X. DOI: 10.1109/37.898794.

[250] J. Talvitie, M. Renfors, and E. S. Lohan, "Distance-based interpolation and extrapolation methods for RSS-based localization with indoor wireless signals", *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1340–1353, 2015, ISSN: 00189545. DOI: 10.1109/TVT.2015.2397598.

[251] R. Kubota, S. Tagashira, Y. Arakawa, T. Kitasuka, and A. Fukuda, "Efficient survey database construction using location fingerprinting interpolation", *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, pp. 469–476, 2013, ISSN: 1550445X. DOI: 10.1109/AINA.2013.53.

[252] D. Li, B. Zhang, Z. Yao, and C. Li, "A feature scaling based k-nearest neighbor algorithm for indoor positioning system", in *2014 IEEE Global Communications Conference*, 2014, pp. 436–441. DOI: 10.1109/GLOCOM.2014.7036847.

[253] S. H. Jung, B. C. Moon, and D. Han, "Performance Evaluation of Radio Map Construction Methods for Wi-Fi Positioning Systems", *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 880–889, 2017, ISSN: 1524-9050. DOI: 10.1109/TITS.2016.2594479.

[254] Q. Song, S. Guo, X. Liu, and Y. Yang, "CSI Amplitude Fingerprinting Based NB-IoT Indoor Localization", *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017. DOI: 10.1109/JIOT.2017.2782479.

[255] G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*, 2d corrected printing 2003. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1999, ISBN: 3540654313.

[256] A. Razavi, M. Valkama, and E. S. Lohan, "K-Means Fingerprint Clustering for Low-Complexity Floor Estimation in Indoor Mobile Localization", in *2015 IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1–7. DOI: 10.1109/GLOCOMW.2015.7414026.

[257] S. M. J. Sadegh, S. Shahidi, and S. Valaee, “An efficient database management for cloud-based indoor positioning using Wi-Fi fingerprinting”, in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6. DOI: 10.1109/PIMRC.2017.8292265.

[258] G. Pecoraro, S. D. Domenico, E. Cianca, and M. D. Sanctis, “LTE signal fingerprinting localization based on CSI”, in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2017, pp. 1–8. DOI: 10.1109/WiMOB.2017.8115803.

[259] W. Guo and N. P. Filer, “2.5D Indoor mapping and location-sensing using an impulse radio network”, in *IET Seminar on Ultra Wideband Systems, Technologies and Applications*, 2006, pp. 211–215.

[260] M. Ulmschneider, C. Gentner, and A. Dammann, “Matching Maps of Physical and Virtual Radio Transmitters Using Visibility Regions”, in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 375–382. DOI: 10.1109/PLANS46316.2020.9110139.

[261] C. Gentner, P Robert, M. Ulmschneider, T. Jost, and S. Zhang, “Positioning Using Terrestrial Multipath Signals and Inertial Sensors”, *Mobile Information Systems*, 2017.

[262] M. Abdmoulah, “Virtual transmitters reconstruction using multipath delay profile fingerprints based on a novel pattern matching algorithm.”, B.S. Thesis, Technical University of Munich, May 2019.

[263] M. Bourguiba, “Interpolation of reference fingerprints using machine learning and pattern matching among the multipath components.”, B.S. Thesis, Technical University of Munich, Jul. 2018.

[264] D. Pelleg and A. W. Moore, “X-means: Extending K-means with efficient estimation of the number of clusters”, in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML ’00, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 727–734, ISBN: 1-55860-707-2. [Online]. Available: http://dl.acm.org/citation.cfm?id=645529.657808.

[265] C. Shalizi. “Distannces between clustering, hierarchical clustering, 36-350 data mining, Carnegie Mellon University”. (2018), [Online]. Available: https://www.stat.cmu.edu/~cshalizi/350/.

[266] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, “Structured Group Sparsity: A Novel Indoor WLAN Localization, Outlier Detection, and Radio Map Interpolation Scheme”, *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6498–6510, 2017, ISSN: 0018-9545. DOI: 10.1109/TVT.2016.2631980.

[267] A. H. Ismail, H. Kitagawa, R. Tasaki, and K. Terashima, “WiFi RSS fingerprint database construction for mobile robot indoor positioning system”, in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 001 561–001 566. DOI: 10.1109/SMC.2016.7844461.

[268] M. Zhou, Y. Tang, Z. Tian, and X. Geng, “Semi-Supervised Learning for Indoor Hybrid Fingerprint Database Calibration With Low Effort”, *IEEE Access*, vol. 5, pp. 4388–4400, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2678603.

[269] J. Bi, Y. Wang, Z. Li, S. Xu, J. Zhou, M. Sun, and M. Si, “Fast radio map construction by using adaptive path loss model interpolation in large-scale building.”, *Sensors*, vol. vol. 19, no. 3, p. 712, Feb. 2019.

[270] M. Zhang and W. Cai, “Multivariate Polynomial Interpolation Based Indoor Fingerprinting Localization Using Bluetooth”, *IEEE Sensors Letters*, vol. 2, no. 4, pp. 1–4, 2018. DOI: 10.1109/LSENS.2018.2878558.

[271] Y. H. Wu, Y. L. Chen, and S. T. Sheu, “Indoor location estimation using virtual fingerprint construction and zone-based remedy algorithm”, in *2016 International Conference On Communication Problem-Solving (ICCP)*, 2016, pp. 1–3. DOI: 10.1109/ICCPS.2016.7751132.

[272] K. Sato, K. Suto, K. Inage, K. Adachi, and T. Fujii, “Space-Frequency-Interpolated Radio Map”, *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 714–725, 2021. DOI: 10.1109/TVT.2021.3049894.

[273] B. Meiners, J. Barowski, A. Nalobin, and I. Rolfes, “Investigation on the geometric properties of multipath components in indoor radio channels”, pp. 1–5, 2015, ISSN: 2164-3342.

[274] ——, “Comparison of the Channel Impulse Response Interpolation Algorithm and Channel Measurements”, pp. 1–4, 2014.

[275] A. Haniz, G. K. Tran, R. Iwata, K. Sakaguchi, J. Takada, D. Hayashi, T. Yamaguchi, and S. Arata, “Propagation Channel Interpolation for Fingerprint-Based Localization of Illegal Radios”, *IEICE Transactions on Communications*, vol. E98.B, no. 12, pp. 2508–2519, 2015. DOI: 10.1587/transcom.E98.B.2508.

[276] W. S. Murphy, Jr., and W. Hereman, “Determination Of A Position In Three Dimensions Using trilateration and Approximate Distances”, Tech. Rep., 1999.

[277] H. Luecken and A. Wittneben, “UWB radar imaging based multipath delay prediction for NLOS position estimation”, in *2011 IEEE International Conference on Ultra-Wideband (ICUWB)*, 2011, pp. 101–105. DOI: 10.1109/ICUWB.2011.6058804.

[278] K. N. Poudel, D. Schurig, and N. Patwari, “Spatial imaging using a communication system’s channel state information”, in *USNC-URSI Radio Science Meeting*, 2016, pp. 41–42. DOI: 10.1109/USNC-URSI.2016.7588502.

[279] M. S. Mercan and E. Öztürk, “Through wall imaging based on MIMO UWB radar with a fast image reconstruction method”, in *2013 European Radar Conference*, 2013, pp. 29–32.

[280] B. Jameson, Y. T. J. Morton, D. Garmatyuk, A. Curtis, and R. Ewing, “Target scene reconstruction in indoor environment with cognitive OFDM radar”, in *International Waveform Diversity Design Conference (WDD)*, 2012, pp. 079–084. DOI: 10.1109/WDD.2012.7311259.

[281] L. Li, X. Chen, and C. Parini, “Development of the UWB radar system with rotating antenna array and two imaging reconstruction algorithms for concealed metallic targets imaging in multipath environments”, in *International Conference on Electromagnetics in Advanced Applications (ICEAA)*, 2015, pp. 759–762. DOI: 10.1109/ICEAA.2015.7297217.

[282] L. Jofre, A. Broquetas, J. Romeu, S. Blanch, A. P. Toda, X. Fabregas, and A. Cardama, “UWB tomographic radar imaging of penetrable and impenetrable objects”, *Proceedings of the IEEE*, vol. 97, no. 2, pp. 451–464, 2009, ISSN: 0018-9219. DOI: 10.1109/JPROC.2008.2008854.

[283] T.-J. Li, M.-M. Ge, and G.-W. Yuan, “Human activity recognition using UWB radar and cameras on a mobile robot”, in *4th IEEE Conference on Industrial Electronics and Applications*, 2009, pp. 3038–3042. DOI: 10.1109/ICIEA.2009.5138706.

[284] R. Zetik, J. Sachs, and R. S. Thoma, “UWB short-range radar sensing - The architecture of a baseband, pseudo-noise UWB radar sensor”, *IEEE Instrumentation Measurement Magazine*, vol. 10, no. 2, pp. 39–45, 2007, ISSN: 1094-6969. DOI: 10.1109/MIM.2007.364960.

[285] P. M. Holl and F. Reinhard, “Holography of Wi-fi Radiation”, *Phys. Rev. Lett.*, vol. 118, p. 183 901, 18 2017. DOI: 10.1103/PhysRevLett.118.183901. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.118.183901.

[286] T. Deissler and J. Thielecke, “Feature based indoor mapping using a bat-type UWB radar”, in *2009 IEEE International Conference on Ultra-Wideband*, 2009, pp. 475–479. DOI: 10.1109/ICUWB.2009.5288802.

[287] F. Guidi, A. Guerra, and D. Dardari, “Personal Mobile Radars with Millimeter-Wave Massive Arrays for Indoor Mapping”, *IEEE Transactions on Mobile Computing*, vol. 15, no. 6, pp. 1471–1484, 2016. DOI: 10.1109/TMC.2015.2467373.

[288] *Navvis company website*, https://www.navvis.com/, Accessed: 27-09-2021.

List of Figures

List of Tables