SEQUENTIAL MONTE CARLO LOCALIZATION USING BLUETOOTH LOW-ENERGY BEACONS

by

Fahri Serhan Daniş

B.S., Computer Engineering, Galatasaray University, 2006

M.S., Systems and Control Engineering, Boğaziçi University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2021

ACKNOWLEDGEMENTS

First and foremost, I sincerely thank my advisors Cem Ersoy and Taylan Cemgil for their profound faith in me and my work, as well as their invaluable guidance, practical suggestions and patience that cannot be underestimated.

I express my gratitude to my thesis committee members Arzucan Özgür, Emre Özkan, Tuna Tuğcu and Sinan Yıldırım.

I had a long and confusing journey while completing this thesis. I shared a lot with the people of CmpE on the way. Several of them played a decisive role in the pursuit. Most importantly, I do not think I will ever enjoy such an inspiring and friendly atmosphere. I thank Cem Ersoy again for his contributions to this atmosphere and being the role model of an academic. I would like to extend my deepest gratitude to Orhan Ermiş, Barış Kurt, Can Tunca, Turgay Pamuklu, Cihat Baktır, Alp Kındıroğlu, Mine Öğretir, Nezihe Pehlivan, Sıla Güler, Çetin Meriçli, Barış Gökçe, Tekin Meriçli, Itır Karaç, Akın Günay, Gaye Genç, Yunus Emre Kara, Nadin Kökciyan, Abuzer Yakaryılmaz, Albert Ali Salah and Ergin Özkucur. I will also not forget the great support from my professors Tuna Tuğcu, Suzan Üsküdarlı, Lale Akarun and Pınar Yolum in the desperation of the dark days. I specially thank Hande Alemdar for taking me out of a drifting state and triggering a series of fortunate events. I remember my dear friend İsmail, who deserves a big thanking just for his presence and big smile.

I am grateful to my colleagues at Galatasaray University, Atay Özgövde, Merve Ünlü, Özlem Durmaz İncel, Sultan Turhan, Yeşim Kop, Gülfem Alptekin and Emre Alptekin for their tolerance, encouragement and warm friendship. My special thanks go to Teoman Naskali for his immense contributions in the data collection process and Pınar Uluer for her constructive commentaries and watchful eagerness to help in any case. My respects go to Günce Orman for her initiative to constitute a long-awaited entourage that makes the local academic life more decent.

I owe a gratitude for my friend Arman Boyacı for sharing his valuable ideas.

I am grateful to my parents Çiğdem and Selim, my brother Dilhun, his dear family and my parents-in-law, Semiha and Ferhat, for their love, patience and support.

Last, but not the least, I thank my love, Nihal, for her endless patience, unconditional and unwavering support during my doctoral journey and of course for sharing and making this life meaningful and purposeful for me and our little daughter Lale. Everyday is beautiful with both of you.

This work is supported by the Turkish Directorate of Strategy and Budget under the TAM Project number 2007K12-873.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
LIST OF FIGURES viii
LIST OF TABLES xiv
ABSTRACT xv
ÖZET xvi
LIST OF SYMBOLS xvii
LIST OF ACRONYMS/ABBREVIATIONS xx
1. INTRODUCTION 1
1.1. Problem Definition and Motivation 1
1.2. Indoor Positioning Systems and Technology 2
1.3. Preliminary Analysis 5
1.4. Proposed Work 9
1.5. Related Work 12
1.6. Contributions and Summary 17
2. EMISSION DENSITY ESTIMATION WITH AFFINE WASSERSTEIN COMBINATION 19
2.1. Transition Model 20
2.2. Observation Models 21
2.2.1. Nearest Fingerprint (NF) 21
2.2.2. κ-Nearest Fingerprint Combination (κ-NF) 22
2.2.3. Artificial Neural Networks (ANN) 23
2.2.4. Affine Wasserstein Combination (AWC) 24
2.2.4.1. Wasserstein Interpolation 25
2.2.4.2. Affine Wasserstein Combination 26
2.2.4.3. Pair Selection Strategy (loose alignment) 27
2.2.4.4. Two-position Affine Wasserstein Combination (AWC-B) 28
2.2.4.5. Multiple-position combination (AWC-E) 29
2.3. Inference with SMC Filter 30

3. ADAPTIVE SEQUENTIAL MONTE CARLO FILTER FOR PARAMETER ESTIMATION . . . 31
3.1. SMC Filter for BLE Localization . . . 33
3.1.1. Initialization . . . 34
3.1.2. Importance Sampling . . . 34
3.1.2.1. Static diffusion . . . 35
3.1.2.2. Decaying diffusion . . . 35
3.1.3. Particle Evaluation . . . 36
3.1.3.1. Evaluation with probabilistic RF map estimate . . . 36
3.1.3.2. Evaluation with occupancy map . . . 38
3.1.4. Resampling . . . 39
3.2. Adaptive SMC Filter with Parameter Sampling for BLE Localization . 39
3.2.1. Initialization . . . 41
3.2.2. Importance Sampling . . . 41
3.2.3. Particle Evaluation . . . 42
3.2.4. Resampling . . . 42
4. HIGH PRECISION DATA ANNOTATION . . . 44
4.1. Camera Pose Estimation . . . 45
4.2. Pose Elimination . . . 47
4.3. Kalman Filter for Smoothing . . . 49
4.4. RSSI Data Registration . . . 50
5. EXPERIMENTAL SETUP . . . 53
5.1. Domestic Area . . . 53
5.1.1. Beacons and Sensors . . . 53
5.1.2. Test Area and Fingerprints . . . 53
5.2. Office Area . . . 55
5.2.1. Hardware Setup . . . 55
5.2.2. Markers and Camera Calibration . . . 57
5.2.3. Handling Setup Errors . . . 59
5.2.4. Data Collection Software Setup . . . 61
5.2.5. Position Annotated Dataset . . . 63

6. EXPERIMENTS AND RESULTS 64
6.1. Experiments for Affine Wasserstein Combination 64
6.1.1. Model Parameters 64
6.1.2. Parameter Search 65
6.1.3. Fingerprint Sets 66
6.1.4. Radio Frequency Map Generation Performances 67
6.1.5. Test Trajectories and Observations 68
6.1.6. Summary of Experiments 69
6.1.7. Results 70
6.2. Experiments for High Precision Data Annotation 75
6.2.1. Data collection experiments 76
6.2.2. Results 76
6.3. Experiments for the Adaptive Sequential Monte Carlo Filter 81
6.3.1. Particle Population Size Selection 83
6.3.2. Resampling Strategy Selection 85
6.3.3. Experiment I: Accurately Synthesized RSSI Data 86
6.3.4. Varying the Diffusion Particle Population Size 88
6.3.5. Experiment II: Noisily Synthesized RSSI Data 91
6.3.6. Experiment III: Real RSSI Data 92
6.3.7. Discussion 94
7. CONCLUSION 96
REFERENCES 100
APPENDIX A: SYNTHETIC DATA GENERATION 113
A.1. Trajectory Sampling 113
A.2. Data Sampling 114

LIST OF FIGURES

Figure 1.1. Change in the RSSI histograms with respect to distance. 6

Figure 1.2. Variation in the detected number of packets with respect to distance. 7

Figure 1.3. A possible interpretation of a sample histogram. 8

Figure 1.4. Sample fingerprints and their positions. 9

Figure 2.1. Graphical model for tracking. 19

Figure 2.2. Transition density of the position of a mobile agent at time t. . . . 21

Figure 2.3. Visualization of a convex combination at the position $\mathbf{x}$ in the convex hull of other fingerprint positions. 22

Figure 2.4. Neural network structure for histogram estimation. 23

Figure 2.5. Interpolation ambiguity: We want to move the pile of red histogram onto the green histogram. The figure on the left represents a direct jump of the measures onto the new location, whereas the one on the right shows a gradual movement of the measures towards the new pile location. 26

Figure 2.6. Affine Wasserstein Combination Algorithm. 27

Figure 2.7. Affine combination of two histograms of Gaussian densities for different α and β values. Yellow vertical lines show the original location of the modes. 28

Figure 2.8. A position $\mathbf{x}$ is loosely aligned with positions $\mathbf{f}_i$ and $\mathbf{f}_j$ if it falls in the green area. 28

Figure 2.9. SMC Filter for BLE Localization. 30

Figure 3.1. Proposed HMM with a latent chain for positions that depends on another one for diffusion factors. 32

Figure 3.2. Tools for evaluating the predicted particles. 37

Figure 3.3. SMC Filter for BLE Localization. 40

Figure 3.4. Parameter sampling based Adaptive SMC Filter for BLE Localization. 43

Figure 4.1. The workflow aims to collect maximum information available, filter the most relevant pose, and synchronize it with the RSSI readings. The arrow weights denote the data load coarsely. 44

Figure 4.2. The position vectors and the coordinate frames: w, c and m; for the world, the camera and the marker frames respectively. 47

Figure 4.3. Position elimination strategies: close marker selection (left) and outlier elimination (right). Black points are selected for the next stage, while the red points are eliminated. 48

Figure 4.4. Disturbing the system artificially to synchronize the timestamps of the video frames with RSSI data. 51

Figure 4.5. RSSI data overlaid on the position data: time flows from red to black. 52

Figure 5.1. Bluetooth dongle on a tripod and beacons mounted on the walls. 54

Figure 5.2. Sample calibration images from the area including markers, extension cables, Bluetooth adapters, computers and other obstacles. . 55

Figure 5.3. Hardware setup. 56

Figure 5.4. ArUco markers and their distribution in the test area. 58

Figure 5.5. Data collection and communication architecture. 61

Figure 5.6. RSSI data flow on ROS (left) and the live data visualizer (right). 62

Figure 5.7. AR localization toolkit for estimating, visualizing, synchronizing and logging data. 62

Figure 6.1. Distribution of β for different points with different alignments. . . 66

Figure 6.2. Distribution of fingerprint locations with different sizes. Black dots: measurement locations, colored dots: beacon locations. 66

Figure 6.3. Comparison of the methods with respect to their radio frequency map generation performances. We measure the Wasserstein distances between the estimated fingerprints and the original fingerprints on the same positions. 67

Figure 6.4. Sampled trajectories with different parameters. 69

Figure 6.5. Block diagram of the experimental setup. 69

Figure 6.6. An SMC filter on the run: Yellow trajectory is the computer generated positions, cyan trajectory is an estimation. 70

Figure 6.7. Trajectory estimation performances of different observation models with different sizes of fingerprint sets. The black lines point the medians and the red lines are for the means. 71

Figure 6.8. Localization error distributions and corresponding cumulative error distributions for two fingerprint location configurations: 15 and 8 locations. 74

Figure 6.9. Position data collection and experiments. 76

Figure 6.10. Comparison of isolated strategies. 77

Figure 6.11. Comparison of combinations of selection and outlier detection strategies. 77

Figure 6.12. Comparison of different noise covariance performances of the Kalman filter after preprocessing with selection and outlier detection strategies. 78

Figure 6.13. Comparison of different noise covariance performances and lag penalty application. 79

Figure 6.14. Error comparisons with respect to varying paces. 80

Figure 6.15. Experiments with different strategies: black dots are the estimated positions: (a) raw position estimations, (b) estimations after only the Kalman filter, (c) only elimination strategies, (d) the Kalman filter preceded by elimination strategies. 81

Figure 6.16. Visualizations of SMC filter experiment samples with different diffusion factors. Green dots are the ground truth positions, red dots are the estimated positions. 81

Figure 6.17. Sample error distributions. From left to right: Trajectory errors throughout an experiment, histogram of the corresponding errors, and cumulative distribution of the errors. The green and red lines indicate the median and mean of the distribution respectively. . . 83

Figure 6.18. Duration distributions (left) and corresponding error distributions (right) with respect to different particle population sizes. 84

Figure 6.19. Duration distributions (left) and corresponding error distributions (right) with respect to different resampling strategies sizes. 85

Figure 6.20. Accurately synthesized data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively. . . 86

Figure 6.21. Adaptive behavior of a sampled diffusion factor. The weighted average of the diffusion factor accompanies the error fluctuations of the position estimates. 87

Figure 6.22. Error distributions for different sensitivity values and different particle size configurations. 90

Figure 6.23. Duration (left) and error rate (right) comparison of the different adaptive strategy configurations with the static and decaying models, the first and second distributions respectively from the left. . . 90

Figure 6.24. Noisily synthesized RSSI data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively. 91

Figure 6.25. Real RSSI data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively. 93

Figure A.1. A randomly generated trajectory and associated RSSI data 114

LIST OF TABLES

Table 6.1. Medians of RF map estimation performances (Wasserstein distance). 68

Table 6.2. Means of trajectory estimation errors (m). 72

Table 6.3. p-values of the pairwise one-sided t-tests of the results for the set of 15. 73

Table 6.4. p-values of the pairwise one-sided t-tests of the results for the set of 8. 73

Table 6.5. Medians of trajectory estimation performances. 73

Table 6.6. p-values of one-sided pairwise Wilcoxon tests for the set of 15. . . 74

Table 6.7. p-values of one-sided pairwise Wilcoxon tests for the set of 8. . . . 75

Table 6.8. Best error statistics (m) for accurately and noisily synthesized RSSI measurements on different trajectories. 89

Table 6.9. Best error statistics (m) for real RSSI measurement experiments. 93

ABSTRACT

SEQUENTIAL MONTE CARLO LOCALIZATION USING BLUETOOTH LOW-ENERGY BEACONS

We model the tracking of a Bluetooth Low-Energy (BLE) moving transceiver as a hidden Markov model, and filter the latent positions using sequential Monte Carlo. A novel aspect of our approach is the development of an observation model, specifically tailored for received signal strength indicator fingerprints: a histogram interpolation based on the optimal transport model of Wasserstein distance. The tracking results of the entire system are compared with alternative baseline estimation methods, such as nearest neighboring fingerprints and an artificial neural network. Our results show that highly accurate estimation from noisy data is practically feasible. Another novelty is a three layered hidden Markov model with joint state and parameter estimation. Assuming that a BLE transmitter does not provide any other motion related information, the transition density is designed to be a normal distribution whose noise covariance depends on a parameter, namely the diffusion factor, that is to be estimated alongside the positions. We first show an experimental proof of concept using synthetic data by comparing three parameter estimation approaches: static, decaying and adaptive diffusion factors. We then obtain the results on real data which show that online parameter sampling adapts to the observed data and yields lower error means and medians, but more importantly steady error distributions with respect to a large range of parameters. Thirdly, we introduce a novel technique and an associated dataset for evaluation of wireless indoor positioning algorithms. The technique makes use of an augmented reality based positioning system to annotate the signal data with high precision positions. Video streams captured by the cameras are subjected to a series of marker recognition, subset selection and filtering operations to yield highly precise pose estimations, hence, they can be used as the ground truth positions.

ÖZET

BLUETOOTH DÜŞÜK-ENERJİ İŞARETÇİLERİ İLE ARDIŞIL MONTE CARLO KONUMLAMASI

Hareketli Bluetooth düşük-enerji (BLE) alıcı-verici takip problemini saklı Markov modeliyle modelliyoruz ve örtülü konum bilgisini ardışıl Monte Carlo ile süzgeçliyoruz. Yaklaşımımızın sağladığı yeni bakış açısı, alınan işaret şiddeti göstergelerinden oluşturulmuş parmak izleri için özel olarak biçilmiş bir gözlem modelidir: Wasserstein uzaklığının en iyi taşıma modeliyle çözümüne dayalı bir histogram arakestirimi. Bütün sistemin takip sonuçları, en yakın komşu ve yapay sinir ağları gibi alternatif temel tahminleme yöntemleriyle karşılaştırılmaktadır. Sonuçlarımız gürültülü veriden yüksek kesinlikli tahminlemenin mümkün olduğunu göstermektedir. Başka bir yenilik, tümleşik bir durum ve parametre tahminini sağlayan üç seviyeli saklı Markov modelidir. Bir BLE vericisinin hareketi hakkında bilgimiz olmadığını varsayarak, geçiş yoğunluk fonksiyonunu normal dağılım olarak tasarlıyoruz. Dağılımın gürültü kovaryansının temelini oluşturan yayılım parametresini konumlarla beraber tahminliyoruz. Önce, bu yöntemin geçerliliğini sabit, azalan ve uyarlamalı yayılım parametresi tahminleme yaklaşımıyla gösteriyoruz. Sonrasında canlı parametre örneklemesinin gözlenen veriye uyum sağladığını ve düşük hata ortalamaları ve orta değerleri ortaya koyduğunu gösteren gerçek veri sonuçlarını elde ediyoruz. Bundan önemlisi, geniş bir parametre değer kümesi için istikrarlı hata dağılımlarına ulaşıyoruz. Üçüncü olarak, kablosuz iç ortam konumlama algoritmalarının değerlendirmesi için yeni bir yordam ve bağlantılı bir veri kümesi tanıtıyoruz. Yordamımız, sinyal verilerini yüksek hassasiyetli konum verileriyle eşlemek için bir artırılmış gerçeklik temelli konumlama sistemiden faydalanmaktadır. Kameralardan sağlanan görüntü akışları bir sıra işaretçi algılama, alt küme seçimi ve süzgeçleme operasyonlarına tabi tutulmaktadır. Sonucunda kesin referans konumları olarak kullanılabilecek yüksek hassasiyetli poz verileri elde edilmektedir.

LIST OF SYMBOLS

$\mathbf{A}$	Kalman state transition model
$\mathcal{A}$	Support area for the position estimations
b	Beacon (emitter) index
$\boldsymbol{b}$	Set of available beacons
B	Size of RSSI histogram (bins)
c	Camera entity or coordinate system
$C()$	Cost function
$d()$	Euclidean distance function
d	Size of the position vector
D	Diffusion population size in the adaptive SMC filter
$\mathbf{f}$	Fingerprint or reference position
F	Number of frames to be processed together
$\mathcal{F}$	Set of all fingerprint positions
$\mathbf{G}$	Kalman gain
$\mathcal{G}$	Gamma distribution function
$\mathbf{h}^a$	Histogram or discrete RSSI distribution at position a
$\tilde{\mathbf{h}}_{\mathrm{M}}$	Histogram estimate via method M
h_i^a	Individual histogram measure
$\mathbf{H}$	Matrix of histograms
$\mathbf{I}$	Intermediate transport map
$\mathbf{Id}$	Identity matrix
$\mathbf{J}$	Kalman observation model
k_t	Diffusion factor, noise covariance factor for transition density
$k_t^{(i)}$	Diffusion particle with index i at time t
$\mathbf{k}$	Vector of diffusion particles
$\mathbf{K}$	Noise covariance matrix of the transition density
L	Size of the hidden layer (ANN)
m	Marker entity or coordinate system

$\mathbf{M}$	Transport map or transport matrix
$\mathcal{M}$	Multinomial distribution function
$\boldsymbol{\mathcal{M}}_{\mathbf{h}^a,\mathbf{h}^b}$	Set of all transport matrices between $\mathbf{h}^a$and $\mathbf{h}^b$
n	Input sample size (ANN)
N	Sample population size in the generic SMC filter
$\mathcal{N}$	Normal distribution function
$p()$	Probability density function
$\mathbf{P}$	A posteriori Kalman covariance matrix estimate
P	Position population size in the adaptive SMC filter
q	Proposal distribution
$\mathbf{Q}$	Kalman process (transition) noise covariance
$r()$	Occlusion selector function
$\mathbf{R}$	Kalman observation noise covariance
$\mathbb{R}$	Real set
${}^i_j\mathcal{R}$	Rotation matrix for conversion from frame j to frame i
s	Sensor (or adapter) index
S	Closeness parameter
sigm()	Sigmoid function
t	Time index
$\mathcal{T}_i$ or $\mathcal{T}_i'$	Experimental tracks
U	Outlier elimination parameter
$\mathcal{U}$	Uniform distribution function
$\mathbf{V}$	Output layer weight matrix (ANN)
w	World entity or coordinate system
$\mathbf{W}$	Hidden layer weight matrix (ANN)
WD	Wasserstein distance
$\mathbf{x}_t$	Position vector or latent variables at time t
x_i	Individual coordinates of $\mathbf{x}$
$\mathbf{X}$	Matrix of position vectors, or set of position particles
y_t^b or y_t^s	RSSI reading from beacon b (or sensor s) at time t
$\mathbf{z}$	Noisy position via marker detection

$\mathbf{Z}$	Hidden layer variable matrix
α	Similarity coefficient (AWC)
β	Displacement coefficient (AWC)
δ	Kronecker delta operator
Δ	Sampled difference
ϵ	Noise
$\boldsymbol{\epsilon}$	Noise vector
η	Decay rate
θ	Orientation (yaw)
${}^{i}\boldsymbol{\theta}_{j}$	Orientation vector of entity j defined in the frame i
Θ	Set of generic parameters
ι	Intermediate transport map measures
κ	Number of neighbors
λ_i	Individual affine coefficients
$\boldsymbol{\lambda}$	Affine coefficient vector
μ	Probability measure
ν	Sensitivity value
ρ	Loose alignment threshold
${}^{i}\boldsymbol{\tau}_{jk}$	Position vector from entity j to entity k defined in frame i
$\upsilon_t^{(i)}$	Diffusion importance weights for the adaptive SMC filter
$\boldsymbol{\upsilon}_t$	Set of diffusion importance weights
$\omega_t^{(i)}$	Importance weights for the generic SMC filter
$\omega_t^{(i,j)}$	Position importance weights for the adaptive SMC filter
$\boldsymbol{\omega}_t^{(i)}$	Set of (position) importance weights per diffusion particle

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AOA	Angle of Arrival
AR	Augmented Reality
ANN	Artificial Neural Network
AWC	Affine Wasserstein Combination
AWC-B	Two Position Affine Wasserstein Combination
AWC-E	Multiple Position Affine Wasserstein Combination
APF	Auxiliary Particle Filter
BLE	Bluetooth Low-Energy
BRDF	Bidirectional Reflectance Distribution Function
CCTV	Closed-Circuit Television
CSI	Channel State Information
EKF	Extended Kalman Filter
fps	Frames per second
GPR	Gaussian Process Regression
HMM	Hidden Markov Model
IBIS	Iterated Batch Importance Sampling
IPS	Indoor Positioning System
K	Thousand
KL	Kullback-Leibler
κ-NF	κ-Nearest Fingerprint
KNN	K Nearest Neigbor
LED	Light-Emitting Diode
LOS	Line of Sight
MAC	Media Access Control
MLE	Maximum Likelihood Estimation
NF	Nearest Fingerprint

PDR	Pedestrian Dead Reckoning
PM	Propagation Model
RF	Radio Frequency
ROS	Robot Operating System
RSSI	Received Signal Strength Indicator
SIR	Sequential Importance Sampling
SLAM	Simultaneous Localization and Mapping
SMC	Sequential Monte Carlo
SVM	Support Vector Machine
TOA	Time of Arrival
TDOA	Time Difference of Arrival
TOF	Time of Flight
USB	Universal Serial Bus
RFID	Radio-Frequency Identification
WPL	Weighted Path Loss

1. INTRODUCTION

The majority of the world population has started to dwell in cities and urbanized areas, where people spend most of their time indoors. As the buildings and structures become ever more complex, the need to use positioning and navigation tools gains importance. However, satellite based positioning systems that are originally designed to work outdoors yield highly inaccurate estimations or fail entirely due to the coverage loss of their signals. Therefore, there is a need to build specialized systems to estimate the micro-locations of objects or the people indoors.

1.1. Problem Definition and Motivation

We propose a lightweight, easy deployable and adaptive indoor positioning system (IPS). We address the problem of positioning and tracking in indoor environments with probabilistic sampling methods that use Bluetooth Low-Energy (BLE) signal indicators as measurements. In a closed indoor area decorated with multiple stationary BLE devices, we track the position of a mobile device that transmits or receives only BLE messages with no other information about its movement or its whereabouts. Due to the nature of the radio-frequency (RF) signals, message transfer is affected by the factors like occlusion, scattering, reflections and interference from other devices, resulting with the multipath phenomenon, mostly because of the furnitures and the materials used in the architecture of the building. We present a sequential Monte Carlo (SMC) filter specially tailored for indoor localization applications. The specialization is of two parts: an emission density by a histogram interpolation method to cope with the multipath phenomenon and an adaptive diffusion based transition density to compensate motion variations.

Assessment of an IPS algorithm is also problematic, because high resolution ground truth data for indoor positioning is rare. It is difficult to collect accurate position data on every point the object of interest passes. This difficulty prevents the

researchers from setting a generalized benchmark for IPS [1]. In this thesis, we handle this problem by building a technique for collecting sets of signal parameter data and associated ground truth positions that are automatically generated by the help of an augmented reality (AR) marker based precise positioning.

An accurate lightweight and easy deployable tracking system for indoor environments has many practical applications. Tracking of objects indoors in real-time has become essential in many fields such as retail, logistics, marketing and health. In healthcare, with the aging population of the world, health workers need assistive systems for tracking elderly or sick people in order to mitigate their workload [2]. Elderly people or patients with dementia can be tracked in indoor environments without needing any other caretaking personnel or complex sensors [3]. The Covid-19 pandemic has introduced the contact tracing notion, which can also be resolved with accurate IPSs [4]. In marketing and retail, a reliable IPS deployed in a mall or a shopping center help make targeted and location based promotions, campaigns or advertisements [5,6]. For more effective advertisements, the IPS systems track physical client behaviors [7,8]. In factories, production and industrial environments, indoor positioning has become a need in asset and hardware tracking for inventory analysis or security reasons. These systems are also available for shipment tracking in logistics [9] or livestock tracking in farms [10]. In mobile robotics, the localization is the basis for trajectory planning and smooth and secure navigation. Researchers attempt to solve this problem with dead-reckoning using inertial sensors accompanied with visual/range related environmental sensors like cameras, range scanners and sonars, or using a combination of these methods [11, 12].

1.2. Indoor Positioning Systems and Technology

With the lack of satellite positioning systems, there appeared diverse modalities, various signal parameters, statistics and associated processing technologies adapted to IPS. Visual information provides the most accurate solutions for positioning, but requires high amounts of data to be captured, transferred and processed. The domain

has developed several processing algorithms like visual odometry and scanmatching that use conventional cameras [13] and depth cameras [14]. Visible light communication that use LED beacons is specially designed for indoor positioning [15], but lately, with the introduction of multiple AR detectors, AR tags have also become robust alternatives for visual information processing [16]. Ultrasound based positioning systems provide accurate localization at low cost. Such systems use the time-of-flight (TOF) of sound waves to estimate the distances between a mobile receiver and stationary transmitters. Estimations are then multilaterated to obtain a position [17]. The same phenomenon that uses light modality is commercialized into range finders and lidars. These systems are specialized in finding the occluded areas or objects in their ranges, making them high precision, but expensive indoor positioning alternatives [18].

Whereas vision, inertial navigation and RF based data processing solutions coexist in hybrid systems with the pervasiveness of smartphones [19], low power consumption and low processing requirements make the pure RF based systems preferable. Wi-Fi technology, originally a family of wireless network protocols, is one of the oldest RF technologies used for indoor positioning because of its easy-to-use and already installed infrastructure for accessing local networks. Signal parameters and preprocessing techniques prior to position estimations depend mostly on the experiments performed with Wi-Fi infrastructures in the first decade of this millenium [20]. Radio-frequency identification (RFID) based positioning systems were considered to be easily affordable alternatives. They consist of two types of appareils: readers and transponders or tags, which may in turn be passive or active. Cheap passive tags are powered by the radio waves emitted by the reader [21], hence narrowing the emission distance under one meter, whereas active tags have their own energy sources and can be read from greater ranges up to hundreds of meters [22]. Ultra wideband is one of the emerging technologies used for IPS. With its high data rate which can attain the speeds of up to 100 Mbps, it has become a good solution for near-field data transmission [23].

BLE transmitters are becoming more ubiquitous everyday, because of their low energy consumption, high accessibility and small mass. BLE is a distinctive feature

added to the Bluetooth technology standard at v4.0. It has been introduced to facilitate the short range communication which requires large amounts of data transfer, however, this new feature on the Bluetooth technology made it a good candidate to be a part of the positioning systems as it provides an efficient way of transferring messages with low energy. A properly configured transmitter can stand online from months to years with conventional batteries [24]. These transmitters can communicate easily with contemporary handheld devices, making them readable by any BLE-ready device [25]. Transmitters are also called beacons, as they were first thought as stationary objects that broadcast Bluetooth packets. Recently, thanks to their small volumes and masses, they are becoming the actual mobile objects of interest that are being tracked by the help of the surrounding sensor systems. These cheap devices are attached to items, assets, animals or people that need to be tracked. Neburka *et al.* [26], after extensively analyzing the performance of BLE technology in indoor environments, show that it is a promising technique for indoor positioning even though the signal parameters are inaccurate and highly depend on the BLE module used. Its new abilities like durability, mobility and high reaction time makes the Bluetooth BLE technology replace Wi-Fi for the positioning purposes.

For the RF based positioning systems there exist a handful of approaches for the signal parameters that can be used to predict the position information. The most used parameters are received signal strength indicator (RSSI), time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and channel state information (CSI). The downside of RF signals is that they do not work robustly because of the multipath problems. We focus on the RSSI, which may give an estimate on the flight distance of the signal from the beacon to the receiver [27].

The data formed by the signal parameters need to be preprocessed before the actual filtering or estimations of positions are performed [20]. There are mainly two basic techniques that process the signal parameters used in IPS: trilateration and fingerprinting. Trilateration is the estimation of positions given the distances between the tracked object and static landmarks whose positions are priorly known. The technique

directly depends on the accuracy of these distances, which fluctuates as the signal parameters fluctuate [28, 29]. With BLE, the distances to the beacons are estimated via the RSSI readings according to the RF signal propagation model [30]. Chen *et al.* [31, 32] combine the pedestrian dead reckoning (PDR) with a weighted path loss (WPL) algorithm based on the log-distance path loss model between a router and a client, under an extended Kalman filter. While the technique has a fast training phase, problems arise due to the RSSI-to-distance mapping model, because of the instability of the distance estimation with BLE [33, 34].

On the other hand, fingerprinting relies on a prior scene analysis (or RF map construction [35]), in which measurements are collected from the environment and calibrated into specific features, called fingerprints, which describe a part of the environment [36]. The system estimates the location of a sensor by comparing the online measurements with the fingerprints, and the most similar fingerprint location is regarded as an estimate for the expected position. The accuracy depends on the profiling density of the corresponding parameter. To make the technique robust and precise, fingerprint locations must be as dense as possible. Thus, there arises a trade-off between the installation overhead versus the precision [37]. Besides, the overhead may also include the recalibration after the system is installed [38, 39]. Previous work on IPS shows that fingerprinting based techniques overperform the trilateration technique with respect to reliability and precision [34, 40].

1.3. Preliminary Analysis

In our experimental setups, we estimate and track the position of the appareils that transfer BLE packets. These packets are captured and corresponding RSSI measurements are generated by the sensor(s). Related literature guides us to use the fingerprinting technique. We do a preliminary study to see whether a trilateration technique may be an option with the beacon and sensor sets in hand. As a previous research suggests [34], we attempt to map the data to the distances according to a logarithm based attenuation model, but without performing a fitting, we can see the

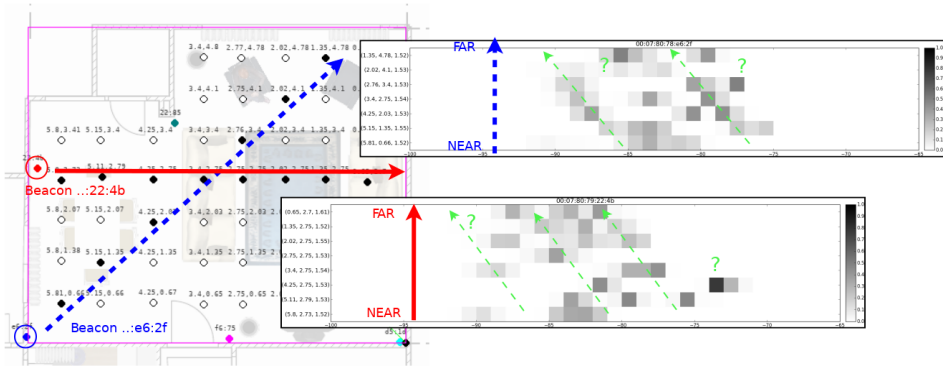


Figure 1.1. Change in the RSSI histograms with respect to distance.

inconsistency in the histograms changing with distance.

Figure 1.1 shows the points that move away from two different beacons and their corresponding histograms of the RSSI data on each position. Whereas the expected behavior of the modes would be to move from lower values to higher values as the point gets farther away from the beacon, we cannot visualize this behavior consistently. The direction of the blue arrow is selected to be aligned with almost linearly positioned fingerprint locations, so that we expect consistently attenuating strength values of the corresponding beacon data. The histograms of the blue beacon on Figure 1.1 do not display the expected behavior. Instead, we obtain multi-mode histograms as we get farther away. The same inconsistency can be visualized with the similar setting for the red beacon. As we cannot map the data to some distance metric, the trilateration method does not seem to be an option in our case. Even when the beacon is in the line-of-sight (LoS) of the sensor, the RSSI values show interesting fluctuations. Moreover, a single reading, which may be a reflection, would lead to incorrect distance mappings, but with fingerprinting the same reading can be explained by another mode of a histogram.

We further analyze the signal behavior with the detection probability of a stationary beacon as seen in Figure 1.2. We expect to see bell shaped histograms whose modes decrease as the distances increase. We indeed observe bell shaped histograms generally, but the variations of the modes and the variances cannot be modeled ap-

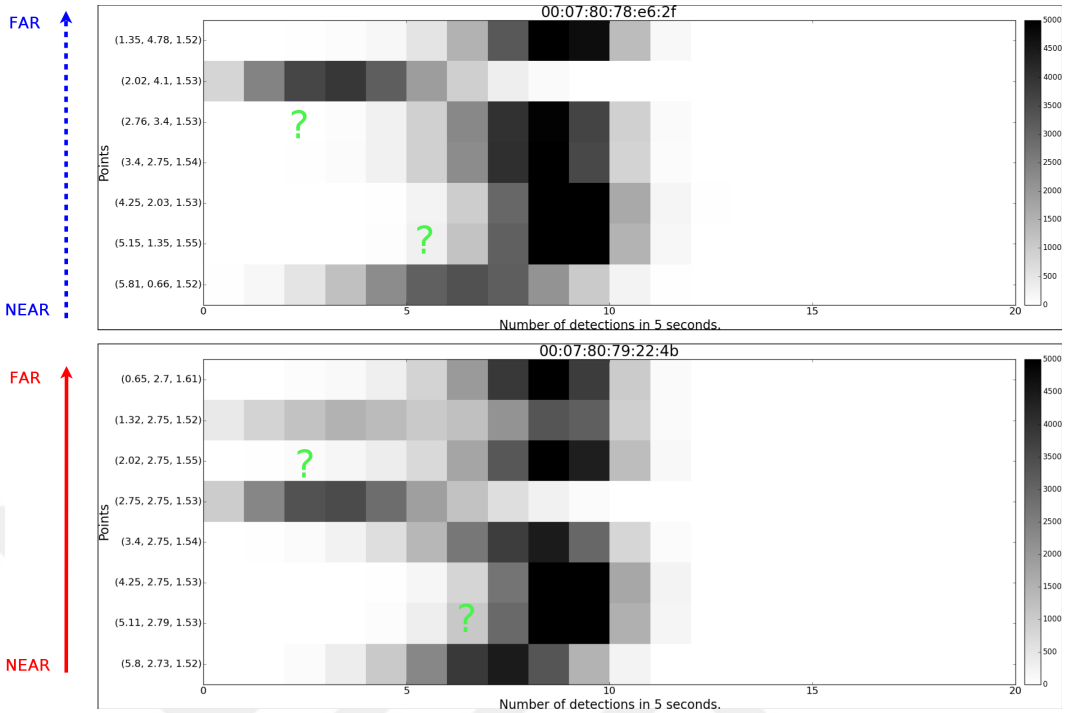


Figure 1.2. Variation in the detected number of packets with respect to distance.

propriately. Finding a mean or mode is not helpful in our research since we require a mapping to distance values in order to make position estimations. The figures do not display gradually decreasing black colored points with respect to distance, that is to say, the expected behavior is not observed as we do not visualize decreasing detection numbers.

To be able to use the detection probability as a statistic for the distance from a beacon, the range of the beacons should be lowered and beacons should be densely mounted, so that we can see the probability changing after a certain threshold which would actually be the range of the beacon. Nevertheless, we deduce that this dense deployment would not meet our expectations.

The formations of the RSSI histograms collected on stationary points reveal interesting features. Some RSSI values do not receive any hits, which results with gaps in the histograms. This feature may be interpreted with gaussian mixtures with extremely different variances, some converging to zero (peaks). We relate the forms of

the histograms to captured RSSI data due to various reflections. A peak with a high RSSI data may be interpreted as a direct reception, and the peaks with lower values may be reflections through different paths (see Figure 1.3). Even though some measure clusters in the histograms resemble bell shaped distributions, most of the data does not reflect this formation, and it would not be efficient to model them with Gaussian mixture models. The sudden gaps in the histograms cannot be interpreted with proper Gaussian forms.

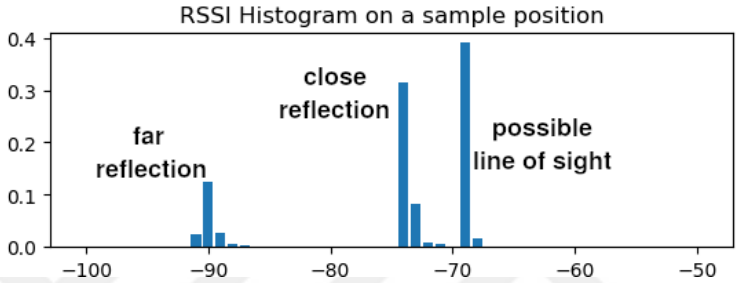
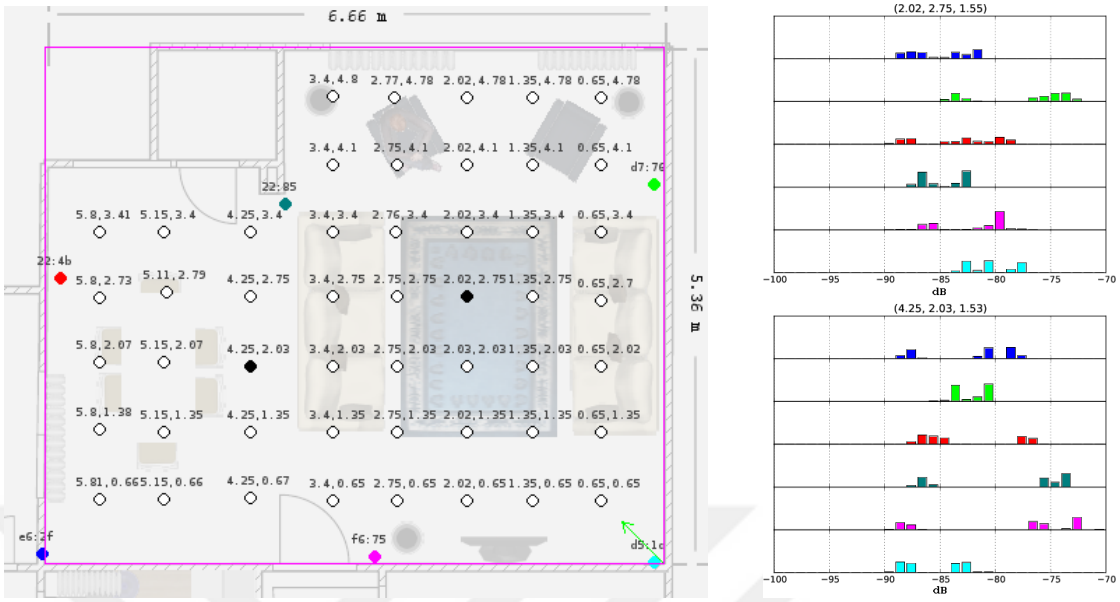


Figure 1.3. A possible interpretation of a sample histogram.

The analyses on the collected data show that the signal-strength-to-distance mapping cannot be performed accurately and efficiently. Therefore we rely on a prior scene analysis based on fingerprinting, and we compute the histograms of RSSI readings on certain positions as fingerprints (see Figure 1.4(a)), but we cannot use the conventional tracking algorithms like the Kalman Filter or its nonlinear-like variations, as the fingerprint histograms do not display any Gaussian-like structure, possibly because of the signal reflections from various objects and surfaces (see Figure 1.4(b)). To try to obtain bell shaped histograms, we could have discarded the data of the lower strengths by using sliding windows, but we tend to keep the multimodality in the histograms, believing that the multimodality has the location information within.

Fingerprinting requires some tedious work of collecting data for long durations from multiple positions in the environment before actual estimation algorithms are executed. Besides, the SMC method needs a dense probabilistic RF map that will be used as the emission density to evaluate the particles on any position on the map. Both of these requirements above can be recovered with a proper histogram combination method. The method takes a fingerprint set of histograms that are not necessarily



(a) A dense fingerprinting example in an indoor environment. Hollow, black and colored dots show measurement positions, sample fingerprint positions (b) and beacon positions respectively.

(b) Sample histograms for different beacons. Color codes are the same with the beacon dot colors in (a).

Figure 1.4. Sample fingerprints and their positions.

dense, but covers the whole map, and yields a high resolution probabilistic RF map.

1.4. Proposed Work

We model the system as a state space model (SSM), in which we observe RSSI readings and estimate the positions. Two important properties of the positioning data restrict us from using linearizable and exact methods: the emission densities that map the position readings to the measured RSSI values are nonlinear and non-Gaussian, and their indices are also discrete integers. Therefore, we attempt to approximate the nonlinear Bayesian filtering using the SMC filter.

SMC filters, or namely particle filters or bootstrap filters, are well known for their efficiency in solving numerically complex and non-standard problems that were previously intractable [41]. Monte Carlo sampling techniques have been long used in positioning and have a popularity in both combining multiple sensor readings like visual

information and inertial information, and perform better against its counterparts like Kalman filter family, especially for nonlinear approximation. They are known to be efficient and easily implemented [42].

An SMC method is composed of three main stages: (i) a prediction step where the particles are moved with respect to a transition density that simulates a motion related input, (ii) an update step where the particles are evaluated according to an emission density, so that their likelihood values or weights are computed, and (iii) a resampling stage where the particles are reselected with respect to how likely they are, to overcome the degeneracy problem.

In the update step of our model, the emission density, that is the distribution of the RSSI values conditioned on the positions, is estimated using the affine Wasserstein combination (AWC) technique [37]. Wasserstein distance is originally a distance metric in optimal transport, defining the distance between two distributions [43]. While optimally transporting measures between distributions, a transport map is calculated and the measures are gradually moved according to this map [44, 45], which corresponds to an interpolation between two distributions.

The prediction step generally tries to model the dynamics of the tracked object by inserting some motion related data. In our model, we contrarily assume that we have very little information on how the object moves: it may only jump onto the neighboring coordinates. This "close" jumping behavior is defined by a normal distribution, which may also be named as a *diffusion*: the particles are diffused into the environment. A diagonal noise covariance is driven by a diffusion factor, which can also be indirectly associated with the velocity of the tracked object, thus becoming very essential for the SMC filter for accurate position estimation. When this diffusion factor is set to a high value, the system tends to be in an exploratory behavior, that is, the predictions are diffused rapidly over a wide area. This behavior increases the possibility of capturing the highest likelihood, but conversely the particle set will be distracted by spuriously similar patches that are highly probable in the RF map, and they will be updated

accordingly, probably increasing the error. If an excessively low diffusion factor is chosen, the predictions tend to lag or get totally stalled and the estimated trajectory cannot catch the true trajectory. These caveats may lead to a divergence in accurate positioning and temporary but relatively highly erroneous position estimations.

In a diffusion based transition density, the diffusion factor should be optimized for each trajectory, but an optimum static parameter can only be properly estimated after at least one full run of the algorithm. Moreover, a *static diffusion* factor optimization approach may be misleading as the velocity of the object or the emission noise can vary throughout an experiment. We also show that a *decaying diffusion* factor may not be as accurate as a fine tuned constant one. We solve this issue by sampling the diffusion factor along with the positions: *adaptive diffusion* factor [46].

When parameter estimation is the purpose of the SMC methods, we generally consider maximum likelihood estimation (MLE) of static parameters [47]. The main objective is to estimate an unknown static parameter from the data in an online or offline manner [48], however, we want the noise parameter to be not only statically estimated, but also to be adapted with respect to the changing conditions or properties, like the velocity of the tracked object or noise fluctuations in the measurement data. A common approach for the SMC filters to adapt to the changing and/or unknown motion of the tracked targets is to augment the state with a parameter under the name augmented particle filters, where the parameter to be estimated is moved to a new process and parameter estimations can be performed through this new process if sufficient data can be supplied [49].

In order to evaluate the proposed SMC modifications, we further propose a framework that make use of an AR-based localization system to synchronize RSSI data with position data. We employ the markers designed for AR applications to label the RSSI data with ground truth positions. AR performs exceptionally well in indoor locations when based on special markers. AR applications are initially developed for adding artificial figures into real world views, its special markers can be detected by a camera to

obtain the precise position and orientation of the marker relative to the camera. This information is transitively used to calculate the location of the camera if the position of the marker is known, so they are preferred in indoor navigation when we aim to track the position of the viewer [50]. The system estimates positions of a specially designed setup that captures video and RSSI data simultaneously. It performs very accurate positioning with an error rate under 0.05 m and it can function alongside the BLE based localization without causing extra interference. With its low error rate, this method provides ground truth positions to evaluate or verify a pure BLE or any other wireless localization system and will greatly speed up the calibration procedure. After the postprocessing, we obtain RSSI data points, each of which is annotated with its precise 3D position. Moreover, with this technique, a fingerprinting or RF map generation workload can be eased and fully automatized.

As an end product, we also provide a publicly available dataset collected using the proposed method. The dataset includes position annotated RSSI data of several trajectories alongside the preprocessed probabilistic RF map estimations in fingerprint or grid formats.

1.5. Related Work

The solutions that use fingerprinting converge in performing the map construction using sparse reference points and path-based approaches. Ai *et al.* [51] construct a fingerprint database using PDR algorithm for spatio-temporal mapping. The full map is then estimated using Gaussian Process Regression (GPR). Their approach is shown to be 300% faster than the conventional approaches. In another study, Li *et al.* [52] perform a fast path-based scene analysis using a step counter to catch the walking speed changes. The positions are approximately estimated and the data are used to train a GPR model. Their algorithm relies on collecting signal parameters and walking parameters separately and tagging the fingerprints with the positions inferred using the walking parameters and the map afterwards.

For fine localization, we require many fingerprints, collection of which costs time and thus is impractical. There exist some attempts to mitigate the time dedicated to the fingerprinting process, which should usually be repeated with the changing environmental conditions. Moreover, for accurate localization, a full scan fingerprint construction should be completed in a certain period of time. Different algorithms are employed over the above mentioned preprocessing techniques. Zhang *et al.* [53] study three typical algorithms in IPS domain: k-Nearest Neighbors (kNN), neural networks and Support Vector Machines (SVM). They show that kNN methods yield better results for localization than the other two. Pelant *et al.* [54] use fingerprint technique and evaluates BLE localization performances based on extensive indoor measurements, mapped by propagation modes. Zhuang *et al.* [55] combine the propagation model (PM) with the extended Kalman Filter (EKF), and reach an error rate of 2.56 m improving the localization accuracy with sparse beacon deployment. Faragher and Harle [56] compare the fingerprinting technique via Wi-Fi with the current fingerprinting via BLE. They use a Bayesian estimator as the main method for positioning with a grid size of 1m, and achieve a tracking accuracy of 2.6 m using a dense beacon distribution and 4.8 m using a sparse distribution, which, as they claim, is a significant improvement over Wi-Fi even using sparse beacon deployment [57]. As another technique, Chen *et al.* [58] study the Bayesian fusion methods using Bluetooth fingerprints and achieve a horizontal positioning accuracy of 4.7 m on the average with respect to Bayesian static estimation and Kalman filter.

Vector interpolation techniques are considered to estimate the histograms associated with arbitrary map positions. A such technique is applied in the works of Zhang *et al.* [59,60] with Wi-Fi strength indicators. They approximate the likelihood of an observation via Nadaraya–Watson kernel regression, which is a computationally complex model, but they achieve accuracies under one meter with Wi-Fi.

Wasserstein distance based histogram interpolation methods have been used for different applications. Bonneel *et al.* [61] use a similar regression intuition in computer graphics under the name of "Wasserstein Barycenter Cooordinates". One of their

applications is recolorizing the images with a transition between the original and the objective color histograms. Likewise, Solomon *et al.* [62] introduce a class of algorithms for tractable optimization problems to solve optimal transportation over geometric domains and apply the idea for shape interpolation, bidirectional reflectance distribution function (BRDF) design, color histogram manipulation, skeleton layout and soft maps.

SMC filters have exclusive use cases in indoor positioning with RF signals. Nurminen *et al.* [63] build a particle filter where PDR predicts the position particles, and Wi-Fi RSSI measurements and the floor plan correct the predictions and achieve a median error rate of 0.8 meters. They show that map information provides a significant improvement on the positioning accuracy. Shubair *et al.* [64] propose a particle filter based technique for Wireless Sensor Networks (WSN) that uses TDOA parameters as measurements. Their TDOA-PF technique is shown to outperform the other well known techniques with respect to both accuracy and robustness. Filipek *et al.* [65] use BLE transmitters as signal sources and estimate the distances between several of these stationary transmitters and use this information for weighing the particles of the generic SMC filter. The tests are performed with previously and discretely selected positions, and the method achieves an error rate of 0.35 meters in average. Zafari and Papapanagiotou [66] use the particle filter over the iBeacon data. They achieve an error rate as low as 0.27 meters using the trilateration technique. In one specific work that use the trilateration technique, Cantón Paterna *et al.* track a mobile sensortag using the surrounding sensing computers [67]. They employ multiple BLE channels to mitigate the effect of fast fading and interferences. The positions are estimated through a trained trilateration method based on weights and smoothed via Kalman filtering. They reach an error rate of 1.82 m in a 54 m^2 room, and 4.6 m in a 290 m^2 room.

One of the methods similar to our work for parameter estimation in SMC is the iterated batch importance sampling (IBIS) method that updates a grid of samples in the parameter space [68], but IBIS is not recursive and requires to compute probability density function of every measurement variable given the previous measurement in closed form. Papavasiliou *et al.* [69], addresses the same problem using two layers of

Monte Carlo methods to solve the previous recursivity issue. The first of these layers generates random grids of parameters, and a particle filter is run with each of these first layer parameters. Chopin *et al.* [70] extend the IBIS method to a grid of samples in the parameter space that are updated over time. At each step, a particle filter is used to process the measurement, and a new grid is generated. These methods involve nested layers of particle filters [71]. The nested methods have not yet been used for tracking of objects, nor for indoor positioning domain.

Auxiliary particle filters (APF) are proposed to solve the problem of adaptation and imprecisement of the sequential importance resampling (SIR) method due to a large variance of the particle weights [72]. Our proposed algorithm may also be associated with APF because of two stage particle assignments, but differs by the type of particle association between two stages. In APFs, firstly auxiliary variable indices are sampled with respect to some auxiliary likelihood values and the second stage particles are sampled using these indices. In our proposed method, we do not analytically link the first stage particles to the second stage, but sample second stage particles using the first stage particles as parameters. When weights are assigned for the second stage particles, they are also propagated to the first stage. This approach filters a trajectory of positions and noise parameters jointly.

Despite their popularity, the SMC techniques have an important disadvantage: it is unclear how to set the noise parameters of the conditional distributions in the first place. Efficient adaptive mechanisms are needed to tune these parameters while performing the actual tracking operation. In [73], an adaptive noise variance selection strategy was designed. They obtain a blindness value that is controlled by two parameters: α for steepness and β for the position of the transition. The noise variances are adjusted according to the blindness estimation at the previous iteration, so that noise in the measurements are never amplified due to the dynamic components. Whereas our approach is similar to this work, we cannot control the emission density noise variances.

Adaptivity in SMC filters may refer to adjusting different parameters, in order to mitigate the estimation errors in changing conditions in the dynamics or in the environment. Lang *et al.* [74] design an adaptive particle filter that is augmented with two resampling steps, one before the prediction step and the other before the the weight update step, so that the number of particles is decreased in the second resampling step which in order decreases the computation time. The tests are performed on a real robot that use sonar readings as measurements. As another example, the size of the samples may be adapted on the fly by measuring the approximation error. In [75], the approach is based on updating the importance weights until the Kullback-Leibler (KL) divergence between the maximum likelihood estimate and the underlying posterior is under a predetermined bound. The efficiency is boosted by avoiding unnecessary resamplings.

Other than the processing techniques, the positioning community lacks a clear standard for evaluating positioning systems. Multiple attempts have been made in order to fill this gap to validate the accuracy of an indoor positioning system. De la Osa *et al.* propose a method to define a ground truth for real life purposes [76]. The technique is constructed upon creating predefined paths with checkpoints which are used as indicators when the tester passes over them. The ground truth trajectory is then estimated by interpolated points between checkpoints. Another one is the skeleton path technique in which the user follows a predetermined path with a constant speed and the positions are sampled. Clearly the techniques based on walking are prone to human errors, incorrectly assuming constant paces and true timings [77]. Moreover, the positions are estimated using inaccurate dead-reckoning data based on step counting [51, 52]. According to [78], this predefined path technique, or namely office walk, is the most preferred ground truth collection method. This inaccurate method is also the cheapest and the most precise ground truth data collection method so far used to validate indoor positioning systems.

In the literature, there exist several valuable datasets formed by wireless signal parameters and corresponding ground truth information, however, such data collections

do not reflect the true nature of the trajectories. Instead, the sets are formed by signal parameters collected with sparsely located reference points [79], or by modifying the paths artificially making the carrier remaining still on predefined reference points [80]. Moreover, the ground truth notion may not even belong to the positions. In the work of Girolami *et al.* [81], the ground truth of social interactions are reported with the BLE RSSI data. Likewise, in the work of Sansano-Sansano *et al.* [82], the ground truth is in terms of the user gate speed.

In the wireless positioning community, AR positioning techniques are used hybridly to enhance the performance of the former systems as wireless signals are prone to yield high error rates [16,83]. Amongst these studies, Byrne *et al.* propose a methodology for collecting RSSI information and accelerometer data using markers placed on the ground and cameras mounted on people to collect the ground truth [84]. Although they function well, camera based devices are usually more expensive than BLE devices and they require the active movement of the camera to determine the location which is not ideal if objects or people need to be passively tracked. Accordingly, La Delfa *et al.* compare different marker technologies for environmental suitability [85]. Xavier *et al.* utilize markers for simultaneous localization and mapping (SLAM) [86], whereas Ujkani *et al.* perform sensor calibration with AR markers in a robotic cell achieving accuracies of 5-10 cm [87].

1.6. Contributions and Summary

This thesis presents an SMC filter method and a data collection method designed for the purpose of indoor positioning based on fingerprinting technique. The methodology is divided into three major chapters that correspond to the three main contributions of this thesis. Chapter 2 describes the first contribution, the affine Wasserstein combination, that is used to estimate a probabilistic RF map, which then can be used as the emission density to evaluate the particles on any position on the map.

Given in Chapter 3, the second contribution is the on-the-fly estimation of the variable diffusion factor through importance sampling alongside the position estimations. This is achieved with a novel version of the nested SMC filter specially tailored for indoor positioning and tracking. We have two layers of particles, one for the positions and the other for the diffusion parameters. The weights evaluated using the measurements at each step are used firstly to estimate a weighted mean of positions. Secondly, the weights are propagated through the diffusion layer and a weighted diffusion factor can also be computed.

In Chapter 4, we describe our ground truth data collection technique that annotates the RSSI data samples with high precision and high resolution position data.

We use two different setups for collecting the fingerprint data and the trajectory data: a domestic setup where we track the position of a mobile sensor that captures data from the surrounding transmitters, and an office environment where the sensors are stationary and we track a mobile transmitter. We illustrate the setup and associated data in Chapter 5.

The experimental design and the results are reported in Chapter 6. We show that the proposed emission density estimation method combined with the adaptive diffusion approach is robust to inaccurate temporary estimations. The results show that an adaptive diffusion factor leads to better position estimations for BLE-based object tracking in indoor environments than the similar works.

The thesis is completed with a conclusion that includes the insights and possible future directions and challenges in Chapter 7.

2. EMISSION DENSITY ESTIMATION WITH AFFINE WASSERSTEIN COMBINATION

We focus primarily on the estimation of the emission densities on arbitrary positions, so that the likelihood of a single position can be computed. Whereas we contribute to the domain with the application of Wasserstein interpolation to estimate an emission density, several other regression or interpolation methods are also discussed. With transition and emission densities in hand, we show how to construct an SMC Filter for BLE localization and tracking at the end of the section.

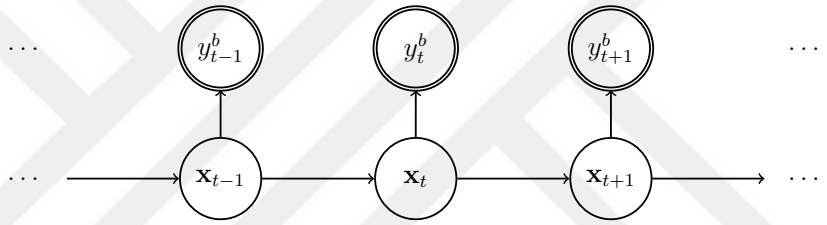


Figure 2.1. Graphical model for tracking.

We begin with a version of the hidden Markov model (HMM) where the diffusion factors are regarded as static. Let $\mathcal{A}$ be a subset of $\mathbb{R}^2$ and $\mathbf{x} = (x_1, x_2)$ be a position in this set $\mathcal{A}$, forming the latent variables. We observe the RSSI values belonging to a discrete set of $\{-120, \ldots, -60\}$, the size of which is adjusted according to the RSSI limit values. We denote each data instance y^b_t, with t being the timestamp of such an instance and b being the index of the beacon that transmits data. According to our model y^b_t depends on the position $\mathbf{x}_t$ and the beacon b [1] . The graphical model is given

[1] In the most general case, bold uppercase letters ($\mathbf{M}, \mathbf{H}$) denote matrices, bold lowercase letters ($\mathbf{x}$) are the vectors. Concordantly, $\mathbf{h}$ will always symbolize the histogram vectors. An element of a vector will be denoted with the unbolded version of the original vector and with an index in the subscript. Index numbering is classically in lower case. Particularly, t always denotes the time points. The superscript on the histogram ($\mathbf{h}^{\mathbf{x}}$) symbolizes the position it is generated on. If the histogram is estimated by a specific method, it will be marked with a tilde ($\sim$), the method will be written in the subscript, and possible hyperparameters will be written as parameters to the method function ($\tilde{\mathbf{h}}^{\mathbf{x}}$). The positions are of two types: $\mathbf{x}$, the arbitrary ones, and $\mathbf{f}$, the fingerprint positions. Fingerprint positions will be indexed in the subscript as they belong to a finite set. $p(\mathbf{x}|y, \Theta)$ denote the conditional probability distributions, and Θ are the problem specific hyperparameters.

in Figure 2.1 and the generative model follows:

$$\begin{aligned} y_t^b &\sim p(y_t^b|\mathbf{x}_t, \Theta) \\ \mathbf{x}_0 &\sim p(\mathbf{x}_0) \\ \mathbf{x}_t &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}). \end{aligned} \tag{2.1}$$

For the transition density, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, we use the motion model given in Section 2.1. In Section 2.2, we construct the estimators for the emission density via multiple methods that use the fingerprints. With the estimated emission densities we compute the likelihood of a new observation sample, $p(y_t^b|\mathbf{x}_t, \Theta)$. The two models are then combined in a tracking filter described in Section 2.3.

2.1. Transition Model

We restrict our agent to reside on a plane, in which, the pose is composed of two cartesian coordinates, $\mathbf{x}_t = (x_1, x_2)_t$ (see Figure 2.2). We assume not to have access to any internal sensory data that enables dead reckoning. The diffusion motion model relies on the assumption that the robot will be in a close position to a previous one, which is thus modeled as a Gaussian distribution with the previous position as the mean value. The covariance of the distribution determines how far the robot can move in a time unit [12]. The next state, $\mathbf{x}_t$, can be modeled by:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \boldsymbol{\epsilon}_t \tag{2.2}$$

where $\boldsymbol{\epsilon}_t = \mathcal{N}(0, \mathbf{K})$. We assume that distributions on individual dimensions are independent and their variances are equal.

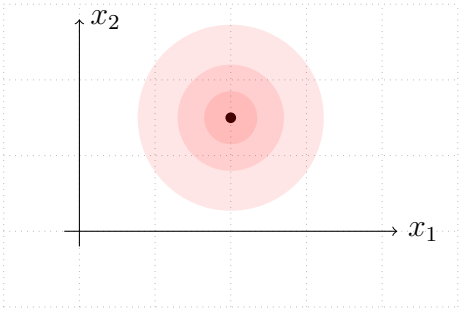


Figure 2.2. Transition density of the position of a mobile agent at time t.

2.2. Observation Models

We present multiple emission density estimations based on different histogram or fingerprint selection or interpolation techniques. We focus on finding an estimate for the likelihood of a measurement given the position on the map.

2.2.1. Nearest Fingerprint (NF)

The most naïve method to estimate the likelihood density on a position $\mathbf{x}$ is to use directly the information on the nearest fingerprint position, corresponding to the κ-nearest neighbor method with $\kappa = 1$. For a given position, $\mathbf{x}$, we find the nearest fingerprint position to $\mathbf{x}$ with respect to the Euclidean distance. With $\mathbf{f}_i \in \mathcal{F}$ and $i \in [1, \ldots, N]$:

$$i^* = \arg\min_i \|\mathbf{x} - \mathbf{f}_i\|. \tag{2.3}$$

This method assumes that the area in which the position estimation will be handled is densely sampled, so that for every position $\mathbf{x}$, there happens to be a close fingerprint position. We use the corresponding histogram on the closest fingerprint location, $\mathbf{h}^{\mathbf{f}_{i^*}}$, as the likelihood:

$$p(y|\mathbf{x}, \Theta) = \tilde{\mathbf{h}}^{\mathbf{x}}_{\mathrm{NF}} = \mathbf{h}^{\mathbf{f}_{i^*}}. \tag{2.4}$$

2.2.2. κ-Nearest Fingerprint Combination (κ-NF)

Alternatively, we can take the convex combination of the surrounding fingerprints. We compute a linear combination of the histograms with respect to their distances to the target position which reside in the convex hull of the input histograms (see Figure 2.3). Furthermore, we generalize the convex combination onto the plane spanned by the fingerprint positions, relaxing the convex hull to an affine hull, making it an affine vector combination.

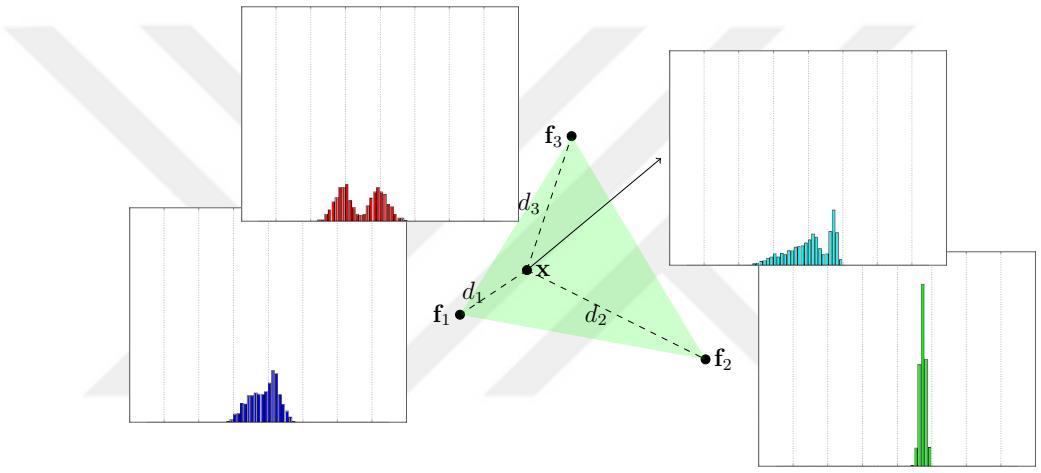


Figure 2.3. Visualization of a convex combination at the position $\mathbf{x}$ in the convex hull of other fingerprint positions.

Let $\mathbf{f}_j$ denote the fingerprint positions on which the histograms are previously computed, $\mathbf{h}^{\mathbf{f}_j}$, and let the position $\mathbf{x}$ reside in the affine hull of a subset of fingerprint positions. Then the histogram on $\mathbf{x}$, $\mathbf{h}^{\mathbf{x}} = \{h_i\}$, is defined as:

$$h_i = \sum_j \lambda_j h_i^{\mathbf{f}_j} \tag{2.5}$$

where $\boldsymbol{\lambda} = \{\lambda_j\}$ are multipliers that are tuned with respect to the distances between the target position $\mathbf{x}$ and the fingerprint positions $\mathbf{f}_j$, and necessarily $\sum_j \lambda_j = 1$. Particularly we use the softmax function to boost the effect of the close fingerprint positions: $\lambda_j = \frac{\exp(-\|\mathbf{x}-\mathbf{f}_j\|)}{\sum_k \exp(-\|\mathbf{x}-\mathbf{f}_k\|)}$. The emission density of a specific position on the map

can then be estimated as:

$$p(y|\mathbf{x},\Theta) = \tilde{\mathbf{h}}^{\mathbf{x}}_{\kappa\text{-NF}}(\kappa) \tag{2.6}$$

where κ is a parameter that decides how many fingerprint positions closest to the target position will be used. Note that the method NF is the special case of κ-NF with $\kappa = 1$.

2.2.3. Artificial Neural Networks (ANN)

Considering the problem as a regression, the intuition leads to tackle it using the neural network approach. A multilayer neural network takes the map coordinates as the input, and returns the histogram measures at that coordinates as the output. We expect the synaptic links to learn the histograms with respect to the positions (see Figure 2.4).

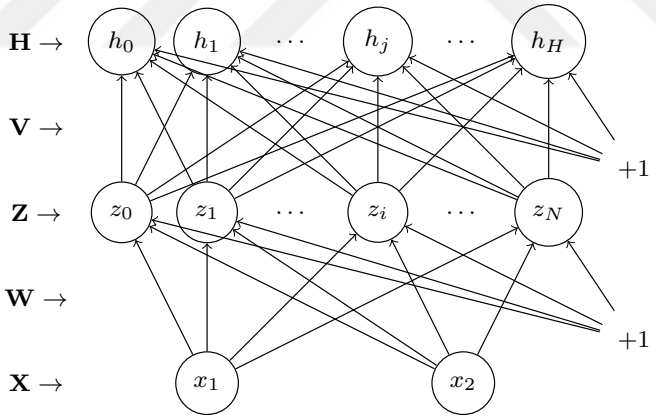


Figure 2.4. Neural network structure for histogram estimation.

Layers are fully connected via nonlinear activation functions (hyperbolic tangent and sigmoid) of the linear perceptron equations. We also add a bias for each node to make the structure learn an offset value if possible. The forward neural equations in matrix notation are given in (2.7):

$$\begin{aligned} \mathbf{Z} &= \tanh\left([\mathbf{X}\ \mathbf{1}]\,\mathbf{W}\right) \\ \mathbf{H} &= \text{sigm}\left([\mathbf{Z}\ \mathbf{1}]\,\mathbf{V}\right) \end{aligned} \tag{2.7}$$

where $\mathbf{X}$ is of size $n \times (d+1)$, $\mathbf{W}$ of $(d+1) \times L$, $\mathbf{Z}$ of $n \times (L+1)$, $\mathbf{V}$ of $(L+1) \times B$ and $\mathbf{H}$ of $n \times B$, and the sizes, n, d, L and B denote the sample size, number of inputs (coordinates), hidden layer size and number of outputs (histogram indices) respectively.

We employ Keras Deep Learning Library [88, 89] for training the neural network model and find the interlayer weights. The algorithm starts with uniformly randomized weights with values in $[-0.05, 0.05]$. The first layer is passed through the hyperbolic tangent and the second layer through the sigmoid function as the relations are considered to be nonlinear. We use the RMSprop, an adaptive gradient descent based learning rate method as the optimizer [90], and the mean squared error as the loss function. With the weights in hand after training, we run the network in the forward direction to get a histogram estimation for the position of interpolation, $\mathbf{x}$:

$$p(y|\mathbf{x}, \Theta) = \tilde{\mathbf{h}}^{\mathbf{x}}_{\mathrm{ANN}}(L). \tag{2.8}$$

2.2.4. Affine Wasserstein Combination (AWC)

Wasserstein distance is originally a distance metric used to compare densities [44, 45]. In the process of calculating the cost to transport the measures between densities, we also obtain a transport function that maps the measures of one density to the measures of the other one. We employ the transport function to produce an interpolation by transporting the measures gradually and to develop an affine combination between two or more histogram positions.

In a discrete state space, we represent the transport function with a transport matrix, $\mathbf{M} = \{\mu_{i,j}\}$, which holds the measure to be transported from a one dimensional vector, $\mathbf{h}^a = \{h^a_i\}$, to another one dimensional vector, $\mathbf{h}^b = \{h^b_j\}$. In our case, these vectors are actually normalized histograms, whose bins are identical and each bin index

corresponds to an integer between the given extremities, as in the following:

$$\begin{matrix} \mathbf{h}^a = \{h_i^a\} \\ \mathbf{h}^b = \{h_j^b\} \end{matrix} \text{ where } \begin{matrix} 0 \leq h_i^a \leq 1 \\ 0 \leq h_j^b \leq 1 \end{matrix} \text{ and } \begin{matrix} \sum_i h_i^a = 1 \\ \sum_j h_j^b = 1. \end{matrix} \tag{2.9}$$

In the discrete case [91], the Wasserstein distance is the minimum cost of transporting one histogram onto another:

$$WD(h_a, h_b) = \min_{\mathbf{M} \in \boldsymbol{\mathcal{M}}_{h_a, h_b}} \sum_{i,j} \mu_{i,j} C(i,j) \tag{2.10}$$

where $\mu_{i,j}$ is the measure that would be transported from the location i to the location j and $C(i,j)$ is the cost multiplier that is usually related to the distance between the locations i, j. If the cost function, C, is a linear function, the Wasserstein transport matrix is efficiently computed by scanning the array and keeping track of the transported weights between bins.

2.2.4.1. Wasserstein Interpolation. We can perform the interpolation operation in multiple ways (Figure 2.5). In fact, we define two parameters, α and β, both in the interval of $[0, 1]$ [45]. The actual interpolation between densities is controlled by the coefficient α, which decides how similar the interpolation would be to the original densities. For the values of α near zero, the interpolation will be similar to the source histogram, whereas for the higher values will make the interpolation resemble the destination histogram. β controls the evolution of the interpolation between a linear interpolation ($\beta = 0$) and a displacement interpolation ($\beta = 1$) [45].

Formally, given two histograms, namely $\mathbf{h}^0$ and $\mathbf{h}^1$, we first compute the Wasserstein mapping, $\mathbf{M}_{\mathbf{h}^0, \mathbf{h}^1}$, which is merely a matrix that demonstrates the transport plan of the measures. According to this transport plan, we calculate two intermediate plans, $\mathbf{I}'_{\alpha,\beta} = \{\iota'_{i,j}\}$ and $\mathbf{I}''_{\alpha,\beta} = \{\iota''_{i,j}\}$, which represent the destination indices the measures,

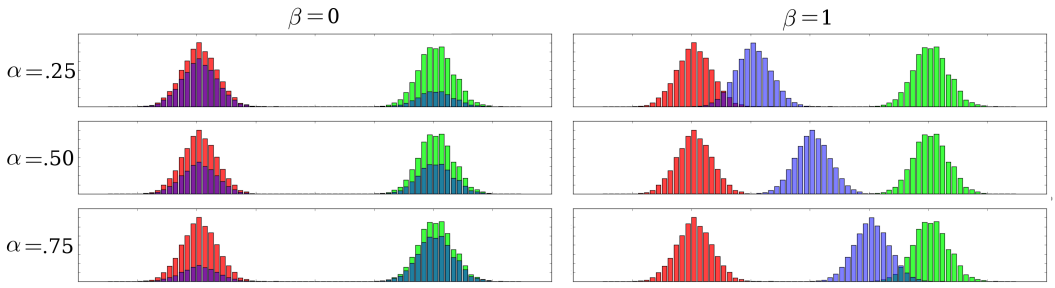


Figure 2.5. Interpolation ambiguity: We want to move the pile of red histogram onto the green histogram. The figure on the left represents a direct jump of the measures onto the new location, whereas the one on the right shows a gradual movement of the measures towards the new pile location.

$\mu_{i,j}$ will be distributed to:

$$\begin{aligned} \iota'_{i,j} &= i + \lceil \alpha\beta(j-i) \rceil \\ \iota''_{i,j} &= j - \lceil (1-\alpha)\beta(j-i) \rceil \end{aligned} \tag{2.11}$$

where $\lceil . \rceil$ is the ceil operator as the indices should be integers. Indices of the final interpolation is computed as follows:

$$h_l = \sum_i \sum_j (1-\alpha)\mu_{i,j}\delta(\iota'_{i,j} = l) + \alpha\mu_{i,j}\delta(\iota''_{i,j} = l) \tag{2.12}$$

where $\delta \in \{0, 1\}$ is the Kronecker delta operator.

Note that a two-position convex combination (κ-NF) of histograms is merely a special case of the Wasserstein combination, in which β is set to 0.

2.2.4.2. Affine Wasserstein Combination. The interpolation of histograms can be generalized to their linear combinations by varying α on the real line, relaxing the limitation of the interval $[0, 1]$ to acquire values out of the interval. With the intermediate

transport plans (2.11), we redefine the interpolation operation at (2.12) as:

$$h_l = \sum_i \sum_j \frac{|1-\alpha|}{|1-\alpha|+|\alpha|}\mu_{i,j}\delta(\iota'_{i,j} = l) + \frac{|\alpha|}{|1-\alpha|+|\alpha|}\mu_{i,j}\delta(\iota''_{i,j} = l). \tag{2.13}$$

The algorithm is given in Figure 2.6 and the combinations for different values of α and β can be seen in Figure 2.7.

1: Initial two histograms $\mathbf{h}^0 = \{h_i^0\}$ and $\mathbf{h}^1 = \{h_j^1\}$.
2: Calculate the discrete Wasserstein mapping $\mathbf{M} = \{\mu_{i,j}\}$.
3: Initialize the interpolation $\mathbf{h}$ with zeros, $h_l = 0$.
4: **for all** i, j **do**
5: Calculate new histogram indices of the measure
$l_0 = i + \lceil \alpha\beta(j-i) \rceil$
$l_1 = j - \lceil (1-\alpha)\beta(j-i) \rceil$
6: Distribute the weight on these indices
$h_{l_0} = h_{l_0} + \frac{|1-\alpha|}{|1-\alpha|+|\alpha|}\mu_{i,j}$
$h_{l_1} = h_{l_1} + \frac{|\alpha|}{|1-\alpha|+|\alpha|}\mu_{i,j}$
7: **end for**
8: **return h**

Figure 2.6. Affine Wasserstein Combination Algorithm.

2.2.4.3. Pair Selection Strategy (loose alignment). For a given position, $\mathbf{x}$, we find the nearest fingerprint positions $\mathbf{f}_i$ from the set of all fingerprint locations, $\mathbf{f}_i \in \mathcal{F}$. If an arbitrary point in a closed area ($\mathbf{x} \in \mathcal{A}$) is *loosely aligned* with two fingerprint locations on which some histograms are defined, we can compute an affine combination of histograms on this point $\mathbf{x}$ using the two fingerprints. A point $\mathbf{x}$ is aligned loosely with two other points $(\mathbf{f}_i, \mathbf{f}_j)$ if

$$d^{\perp}(\langle \mathbf{f}_i, \mathbf{f}_j \rangle, \mathbf{x}) \leq \rho \tag{2.14}$$

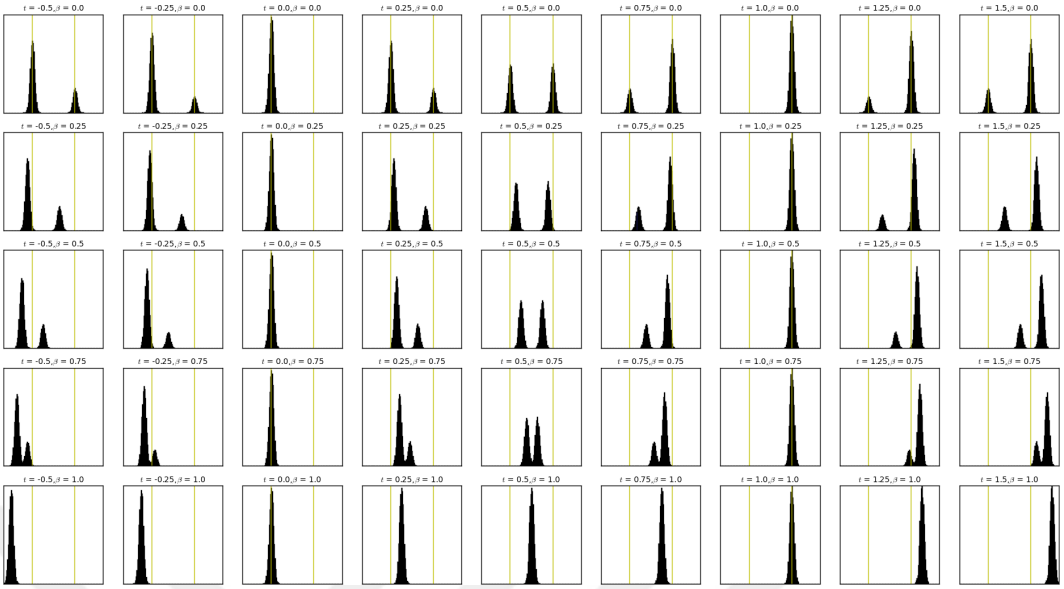


Figure 2.7. Affine combination of two histograms of Gaussian densities for different α and β values. Yellow vertical lines show the original location of the modes.

where $d^{\perp}$ defines the orthogonal distance of a point to a line, $\langle \mathbf{f}_i, \mathbf{f}_j \rangle$ is the set of affine combinations or simply the line passing through the points, $\mathbf{f}_i$ and $\mathbf{f}_j$, and ρ is the *looseness parameter* (See Figure 2.8).

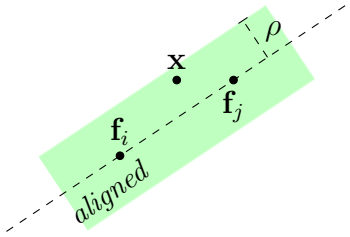


Figure 2.8. A position $\mathbf{x}$ is loosely aligned with positions $\mathbf{f}_i$ and $\mathbf{f}_j$ if it falls in the green area.

Finally, if there exist predefined histograms at positions $\mathbf{f}_i$ and $\mathbf{f}_j$ and if $\mathbf{x}$ is aligned loosely with these positions with respect to a previously defined ρ, we can apply an affine Wasserstein combination of histograms between $\mathbf{h}^{\mathbf{f}_i}$ and $\mathbf{h}^{\mathbf{f}_j}$ at $\mathbf{x}$.

2.2.4.4. Two-position Affine Wasserstein Combination (AWC-B). As an attenuation model of wireless signals does not necessarily behave linearly, we propose multiple

mappings, f_m, that map the distance between map positions to the Wasserstein similarity parameter, α. These mappings are linear, quadratic, cubic, logarithmic and exponential mappings, $\alpha = f_m(\|\mathbf{f}_i, \mathbf{x}\|)/f_m(\|\mathbf{f}_i, \mathbf{f}_j\|)$. With $\mathbf{M}_{\mathbf{f}_i,\mathbf{f}_j}$ being the Wasserstein Mapping from the histogram at $\mathbf{f}_i$ to the histogram at $\mathbf{f}_j$, the affine combination of histograms between $\mathbf{h}^{\mathbf{f}_i}$ and $\mathbf{h}^{\mathbf{f}_j}$ at $\mathbf{x}$ is computed. An estimator for the probability density at the position $\mathbf{x}$ would be:

$$p(y|\mathbf{x}, \Theta) = \tilde{\mathbf{h}}^{\mathbf{x}}_{\text{AWC-B}}(\alpha, \beta). \tag{2.15}$$

The original Wasserstein transport problem is defined to be an interpolation [45], where α is restricted to be in $[0, 1]$, however without this limitation on α, the interpolation generalizes to an affine combination, so that the position $\mathbf{x}$ does not necessarily have to reside in the convex hull of the fingerprints. The method is expected to give a likelihood density estimation out of this hull, as the candidate position $\mathbf{x}$ may exist out of the hull. The definition does not depend on the density of the fingerprints, but the accuracy would depend on the scatter of the fingerprints.

2.2.4.5. Multiple-position combination (AWC-E). The Wasserstein combination method for a single position can be employed using only two fingerprints, but the method is prone to lose any other surrounding information. In order to take advantage of the whole map and be able to compare this method with other estimation algorithms that use more than two fingerprints, we develop an extension on the affine Wasserstein combination method.

According to this extension, for the position of interest, $\mathbf{x}$, we select all of the possible fingerprint pairs that can generate a combination according to the selection model. For each pair we perform the pairwise combination and obtain a histogram estimation on $\mathbf{x}$ with each pair. The final combination, $\tilde{\mathbf{h}}_{\text{AWC-E}}$, is found by taking the affine combination of the available two-position affine Wasserstein combinations, $\tilde{\mathbf{h}}_{\text{AWC-B}}$, and averaging them with the weights, λ_i, inverse exponentially proportional to

the pairwise average distances to $\mathbf{x}$. We will denote the final histogram with $\tilde{\mathbf{h}}_{\text{AWC-E}}$:

$$p(y|\mathbf{x},\Theta) = \tilde{\mathbf{h}}^{\mathbf{x}}_{\text{AWC-E}} = \sum_i \exp(-\lambda_i)\tilde{\mathbf{h}}^{\mathbf{x}}_{\text{AWC-B}}. \tag{2.16}$$

2.3. Inference with SMC Filter

In our setting, we use the generative model defined in (2.1) with the graphical model in Figure 2.1. For the transition density, $p(\mathbf{x}_t|\mathbf{x}^{(i)}_{t-1})$, we employ the diffusion motion model (2.2), assuming that no rotation or translation information is supplied. For the emission density, $p(y_t|\tilde{\mathbf{x}}^{(i)}_t)$, we have six different estimators, listed in Section 2.2 as (2.4), (2.5), (2.8), (2.15) and (2.16). We build up a standard sequential Monte Carlo filter [41, 92] setting these densities into the predict and update steps as given in Figure 2.9.

1: Instantiate

For $i \in [1,\ldots,N]$, sample $\mathbf{x}^{(i)}_0 \sim p(\mathbf{x}_0)$

2: Importance Sampling

Update time index $t \leftarrow t+1$

For $i \in [1,\ldots,N]$, sample $\tilde{\mathbf{x}}^{(i)}_t \sim p(\mathbf{x}_t|\mathbf{x}^{(i)}_{t-1})$

For $i \in [1,\ldots,N]$, evaluate to obtain the importance weights $\omega^{(i)}_t = p(y_t|\tilde{\mathbf{x}}^{(i)}_t)$

Normalize the importance weights $\tilde{\omega}^{(i)}_t = \frac{\omega^{(i)}_t}{\sum_{i=0}^{N}\omega^{(i)}_t}$

3: Select

Resample N particles as $\mathbf{x}^{(i)}_t$ from the particles $\tilde{\mathbf{x}}^{(i)}_t$ according to the importance weights.

4: Recurse with repeating 2.

Figure 2.9. SMC Filter for BLE Localization.

3. ADAPTIVE SEQUENTIAL MONTE CARLO FILTER FOR PARAMETER ESTIMATION

For diffusion factor estimation, we improve the HMM given in Chapter 2. The diffusion parameters constitute a prior latent layer and the positions form a secondary one, and the RSSI measurements captured by the distributed sensors are the observations. We write the generative dependency model as follows:

$$\begin{aligned} k_0 &\sim p(k_0) \\ k_t &\sim p(k_t|k_{t-1}) \\ \mathbf{x}_0 &\sim p(\mathbf{x}_0) \\ \mathbf{x}_t &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, k_t) \\ y_t^s &\sim p(y_t|\mathbf{x}_t). \end{aligned} \tag{3.1}$$

The diffusion layer forms a Markov chain, with a prior distribution at the initial time, $p(k_0)$. The diffusion parameter, k_t, at discrete time, t, is a function of the diffusion parameter at its previous time, $p(k_t|k_{t-1})$. The secondary latent layer is initialized likewise with a prior distribution, $p(\mathbf{x}_0)$, and the transition density at the current time is conditioned on the position at its previous time and the diffusion parameter at the current time, $p(\mathbf{x}_t|\mathbf{x}_{t-1}, k_t)$. We observe only the RSSI measurements, y_t^s, captured by a sensor, s. RSSI data are modeled by the emission density $p(y_t^s|\mathbf{x}_t)$. The problem forms the hidden Markov model shown in Figure 3.1. We want to make a Bayesian approximation to a filtering distribution $p(\mathbf{x}_t, k_t|y_{1:t}^s)$, and jointly find accurate position estimates of the tracked object and the diffusion factor given the measurements, or the RSSI values, so far.

The tracked object is a beacon that emits Bluetooth low energy packets. We restrict our beacon to reside on a plane, in which the position is composed of two cartesian coordinates, $\mathbf{x} = (x_1, x_2)$. We assume that these beacons do not have any

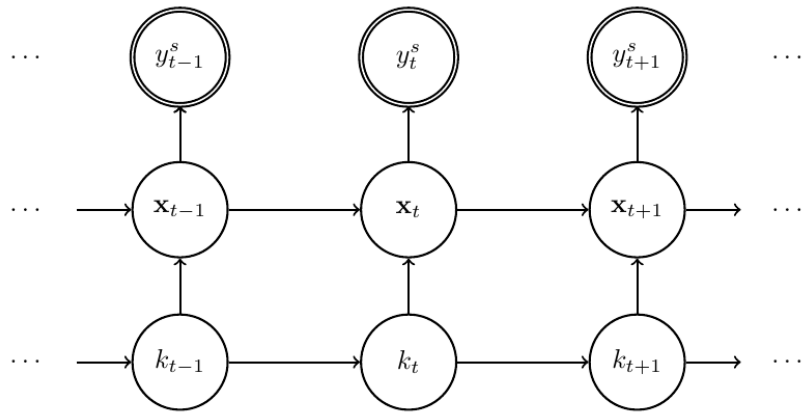


Figure 3.1. Proposed HMM with a latent chain for positions that depends on another one for diffusion factors.

other sensor or processing appliances, hence, we (assume to) have no access to any internal information about the movement of the tracked object, but we also assume that the object moves smoothly in the area so that its current position is directly related to its position at the previous timestamp (no kidnapping). This transition is modeled as a *diffusion* density which is described by a normal distribution with the previous position $\mathbf{x}_{t-1}$ as its mean, and a 2×2 covariance matrix that is driven by the diffusion factor, $\mathbf{K}_t = k_t \mathbf{I}_2$:

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, k_t) = \mathcal{N}(\mathbf{x}_{t-1}, k_t \mathbf{I}_2). \tag{3.2}$$

The diffusion factor serves as a scale for the noise covariance of the position chain. Therefore, this factor should be positive. To guarantee to easily sample positive diffusion factors, we define the transition density by a gamma distribution. The shape and scale parameters of the gamma distribution, a and b, are tuned accordingly to ensure that the diffusion factor at the previous timestamp, k_{t-1} is put at its mode, and

a *sensitivity* value, ν, at its variance:

$$\begin{aligned} p(k_t|k_{t-1}) &= \mathcal{G}(a_t, b_t) \\ \text{with } a_t &= \frac{(k_{t-1})^2}{\nu} + 1 \\ \text{and } b_t &= \frac{\nu}{k_{t-1}}. \end{aligned} \tag{3.3}$$

The system depends solely on the RSSI measurements, y_t^s, captured by a sensor, s, at a discrete time step, t. To obtain a probability density of the RSSI measurements given the positions, namely a probabilistic RF map, we use the histograms of RSSI measurements, or fingerprints, previously captured on several positions in the test environment. Because we require to evaluate any RSSI measurement on any point, we utilize a histogram interpolation method, Affine Wasserstein Combination (AWC) [37]. From the two specific types of these interpolation methods, we choose Two-point Affine Wasserstein Combination which considers the closest and the most aligned two fingerprint positions to compute a histogram on the position $\mathbf{x}_t$, as follows:

$$p(y_t^s|\mathbf{x}_t, \Theta) = \tilde{h}(y_t^s; \mathbf{x}_t, \alpha, \beta) \tag{3.4}$$

where $\tilde{h}()$ denotes an estimate for the histogram of RSSI measurements for a specific sensor, s, at position $\mathbf{x}_t$. The parameters α and β are algorithm specific parameters. The algorithm is summarized in Section 3.1.3 as a part of SMC filters.

3.1. SMC Filter for BLE Localization

We first describe a generic SMC filter for fusing BLE data captured by several sensors. In this generic version, the diffusion factor is set as static and decaying, so the diffusion layer is replaced with a single static value or a deterministic decaying function for the current section. We evolve the setup into the adaptive SMC filter by using the diffusion transition density in Section 3.2.

In the SMC formalization, N will denote the particle population size at each step of the SMC algorithm. The population size will always be constant in an experiment, but may be varied between experiments. We will denote each particle with $(\mathbf{x}_t^{(i)}, \omega_t^{(i)})$ where $\mathbf{x}_t^{(i)}$ is a 2D cartesian vector that stands for the position of the particle $i \in [1, \ldots, N]$ at discrete time step $t \in [0, \ldots, T]$ and $\omega_t^{(i)}$ is its current weight. Time indices will be put in subscripts and particle indices will be in parentheses in superscripts. Vectors will be shown with bold and lowercase characters, matrices are bold and uppercase. The predictions are marked with a tilde, $\tilde{}$.

3.1.1. Initialization

Initially, we have no information on the whereabouts of the transmitter. The particles of the initial population are modeled as continuously uniformly distributed:

$$\mathbf{x}_0^{(i)} \sim \mathcal{U}(\mathcal{A}) \tag{3.5}$$

where $\mathcal{A}$ is the rectangular area of the test environment that supports the uniform distribution.

3.1.2. Importance Sampling

For an appropriate proposal distribution, we set a diffusion based density to sample new particles from:

$$\tilde{\mathbf{x}}_t^{(i)} \sim \mathcal{N}(\mathbf{x}_{t-1}^{(i)}, k_t \mathbf{I}_2) \tag{3.6}$$

where a new particle of the prediction population is sampled from a normal distribution in which the mean is the i^{th} particle from the previous time step, and $k_t \mathbf{I}_2$ is a diagonal matrix driven by a diffusion factor to assure that the scatter scales in both dimensions are equal.

The accuracy of the system depends highly on the diffusion factor, k_t, as it determines the scatter of new particles. If the scatter is too high, the system tends to explore a large area of interest and this may result in high localization errors temporarily while exploring. If the scatter is too low, we restrict the system in a small neighborhood so that it cannot shoot the particles far enough, which prevents exploration of more accurate regions. We use two different diffusion factor strategies:

3.1.2.1. Static diffusion. If the transmitter is assumed to move at a known constant velocity and if the RSSI measurements can discriminate the position accurately, an appropriate diffusion factor can be determined beforehand, by which the proposal density distributes the particles as far as required and the estimations are expected to be accurate:

$$k_t = k_{t-1} = k^*. \tag{3.7}$$

We set the diffusion factor to a predetermined value, k^*, which ensures that the new particle predictions are scattered with respect to a distribution with a static covariance, or a static diffusion factor. A feasible value of this parameter is searched by running a filter with a discrete set of parameters. It is a problem that this factor may only be determined empirically, as we do not also know the velocity of the tracked object.

3.1.2.2. Decaying diffusion. The problem with the static diffusion factor may be overcome by reducing the diffusion factor gradually. As the observations are fed into the system, the estimations are expected to be more accurate. Beginning with a high diffusion factor, k_{max} for high exploration and decaying it gradually down to finer scales is another strategy. We set an initial diffusion factor, $k_0 = k_{max}$ and multiply it with a decay factor, η, at each step. The diffusion factor is reduced by some decay factor

until it reaches a limit value, k_{min}:

$$k_0 = k_{max}$$
$$k_t = \begin{cases} \eta k_{t-1}, & \text{if } k_t > k_{min} \\ k_{t-1}, & \text{otherwise.} \end{cases} \tag{3.8}$$

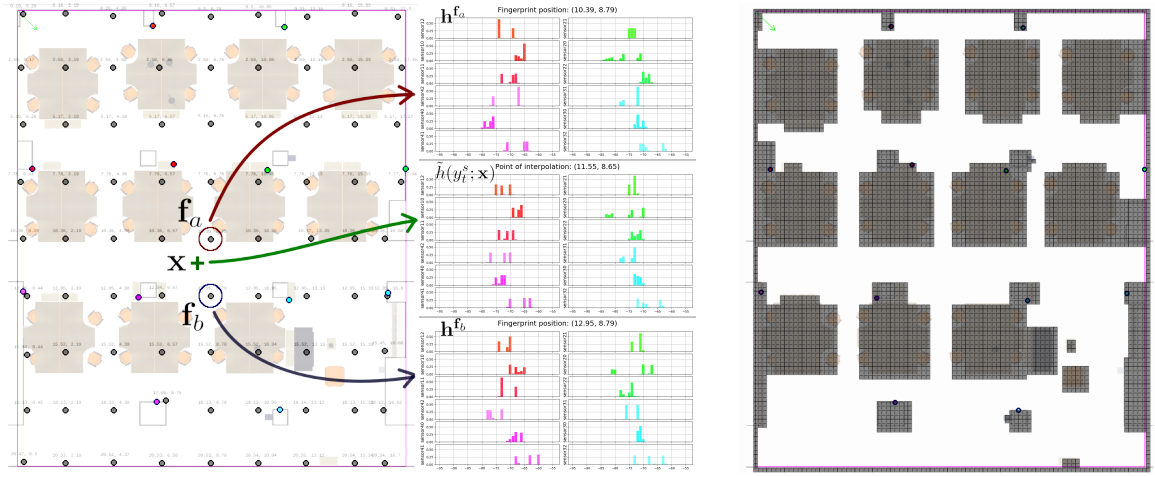
The static diffusion strategy is a special case of the decaying diffusion strategy with $\eta = 1$ and $k_0 = k_{max}$.

3.1.3. Particle Evaluation

The particles are evaluated against a probabilistic RF map to determine their importance weights. A weight is quantified by the likelihood of being in that position with the measured RSSI value, however, we do not have such a map. Instead, we have access to some histograms on some reference points, i.e. the fingerprints. In this section, we first find an estimate for the probabilistic RF map that will emulate the emission density of the state space model, so that we can evaluate the particle predictions. We update it with the architectural information as an occupancy map.

3.1.3.1. Evaluation with probabilistic RF map estimate. To obtain the weight of a particle, we first need to estimate the RSSI distribution on the particle's specific position. We employ the Two-point AWC algorithm from Daniş and Cemgil [37]. The algorithm is used to estimate the corresponding RSSI histograms of a sensor on any position on the map using fingerprints. Note that this section is added to summarize the algorithm and to ensure the integrity of the whole methodology. The validity of the procedure and its relation to real measurements are detailed in the original article [37].

The histogram interpolation is a two-step process: Firstly, for a given position on the map, $\mathbf{x}$, we find the most appropriate pair of fingerprint positions from the set of all fingerprints, $(\mathbf{f}_a, \mathbf{f}_b) \in \mathcal{F}^2$. Secondly, we run AWC with the parameters, $\mathbf{x}$ as



(a) Fingerprint positions (gray) and histogram interpolation with AWC: Histograms on red and blue encircled positions are used to estimate an histogram on green crosshair.

(b) Occupancy map: black cells denote the unpassable regions (columns, furnitures or walls).

Figure 3.2. Tools for evaluating the predicted particles.

the target position and $(\mathbf{f}_a, \mathbf{f}_b)$ as the source positions. Briefly, the algorithm finds a transport plan of the probability measures on discrete positions by minimizing the cost of transferring measures from one histogram onto the other. The interpolation histogram on the specific position is then estimated by realizing this transport plan partially with respect to its distance to the fingerprint pair. An example interpolation is given in Figure 3.2(a).

We have a histogram on each of these positions for each sensor, $\mathbf{h}^{\mathbf{f}_a}$ and $\mathbf{h}^{\mathbf{f}_b}$, and we are to estimate the histograms of RSSI measurements on $\mathbf{x}$, $\tilde{h}(y_t^s; \tilde{\mathbf{x}}_t, \alpha, \beta)$. The algorithm specific parameters, α is a similarity measure that is tuned with respect to the distance of the target position to each of the source positions, and β controls if the interpolation is a linear interpolation or a displacement interpolation.

With an estimate of the probabilistic RF map defined on any point on the map, we evaluate the particles using the measured RSSI value to obtain their importance

weights:

$$\begin{aligned}\omega_t^{(i)} &= p(y_t^s|\tilde{\mathbf{x}}_t^{(i)}) \\ &= \tilde{h}(y_t^s; \tilde{\mathbf{x}}_t, \alpha, \beta).\end{aligned} \tag{3.9}$$

3.1.3.2. Evaluation with occupancy map. Any area that the system is installed in may have unpassable (occluded) regions, such as building columns, walls, cabinets or tables. The transmitter is not expected to pass through these occluded regions. These impassable regions of the area are labeled beforehand, so that the particles that get generated on these regions are evaluated accordingly. The occupancy grid is denoted by $r(\mathbf{x})$:

$$r(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \text{ is in the occupied region,} \\ 1, & \text{otherwise.} \end{cases} \tag{3.10}$$

For particle evaluation with occlusion, we have two choices: (i) these particles are either resampled until they fall in the passable region, or (ii) the weights of such particles are directly set to zero. The former method (i), would be the desired action in order to keep the efficient sample size intact, but this action may lead to several resampling attempts especially if the particles are at the border, resulting in a drastic slow down of the current step. We use the latter method (ii) to keep the evaluation duration stable by setting the particle weights to zero if they are out of the passable area, which means that the particles are sterilized not to produce offsprings in the resampling and nor are they taken into consideration for weighted mean estimation:

$$\omega_t^{(i)} = \tilde{h}(y_t^s; \tilde{\mathbf{x}}_t^{(i)}, \alpha, \beta) r(\tilde{\mathbf{x}}_t^{(i)}) \tag{3.11}$$

where the importance weight of the i^{th} particle, $\omega_t^{(i)}$, is also multiplied with the corresponding occupancy information $r(\tilde{\mathbf{x}}_t^{(i)})$, which sets the weight to zero if the sampled position, $\tilde{\mathbf{x}}_t^{(i)}$, is in an occupied zone, or leaves untouched otherwise. A sample occu-

pancy map is given in Figure 3.2(b).

The importance weights are normalized before the resampling stage:

$$\tilde{\omega}_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{i=0}^{N} \omega_t^{(i)}}. \tag{3.12}$$

3.1.4. Resampling

A resampling strategy is used to overcome the degeneracy problem of the algorithm, that is to avoid a situation where all importance weights but one are zero. We use the systematic resampling strategy [93], whose time complexity is of order $O(N)$, more efficient than the multinomial or stratified resampling techniques [94]:

$$\mathbf{x}_t^{(i)} \sim Syst(\tilde{\mathbf{X}}_t, \tilde{\boldsymbol{\omega}}_t),\ i \in [1, \ldots, N]. \tag{3.13}$$

An evaluation of the three traditional resampling techniques are added for this setup in Section 6.3. The SMC filter for BLE Localization is summarized in Figure 3.3.

3.2. Adaptive SMC Filter with Parameter Sampling for BLE Localization

In this section, we describe the adaptive SMC filter that jointly estimates the position and the unknown variable diffusion factors. Compatible with the proposed hidden Markov model (Figure 3.1), we assume to have two layers of latent variables, one for the position and the other for the diffusion factors.

We modify the generic SMC filter defined in Section 3.1 by first adding the diffusion particles. We will denote the diffusion (factor) particles with the superscripted version of the diffusion factor, $k_t^{(i)}$. The position particles are also modified to have two indices in superscript, $\mathbf{x}_t^{(i,j)}$, with i for the diffusion particle it depends and j for the index in the set of position particles that depend on the same diffusion particle. We

1: **Initialization**

Sample a set of particles:

for $i \in [1, \ldots, N]$, $\mathbf{x}_0^{(i)} \sim \mathcal{U}(\mathcal{A})$

Initialize diffusion factor:

$k_0 = k_{max}$

2: **Importance Sampling**

Update the time index

$t \leftarrow t + 1$

Sample a new set of particles from the current set:

for $i \in [1, \ldots, N]$, $\tilde{\mathbf{x}}_t^{(i)} \sim \mathcal{N}(\mathbf{x}_{t-1}^{(i)}, k_t \mathbf{I}_2)$

Evaluate the proposed set with the RF and occupancy maps to obtain the importance weights:

for $i \in [1, \ldots, N]$, $\omega_t^{(i)} = \tilde{h}_{y_t^s}(\tilde{\mathbf{x}}_t^{(i)}, \alpha, \beta) r(\tilde{\mathbf{x}}_t^{(i)})$

Normalize the importance weights:

for $i \in [1, \ldots, N]$, $\tilde{\omega}_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{i=0}^{N} \omega_t^{(i)}}$

Update the diffusion factor:

$k_t = \eta k_{t-1}$

3: **Resampling**

Resample N particles according to the importance weights

for $i \in [1, \ldots, N]$, $\mathbf{x}_t^{(i)} \sim Syst(\tilde{\mathbf{X}}_t, \tilde{\boldsymbol{\omega}}_t)$

4: **Recursion**

Go back to 2, if not at the end of the trajectory

Figure 3.3. SMC Filter for BLE Localization.

also introduce the similar versions for the position particle weights, $\omega_t^{(i,j)}$, and diffusion particle weights, $\upsilon_t^{(i)}$.

The generic SMC filter is also modified in terms of the number of particles. We assume that multiple position particles can be instantiated by a diffusion particle. We will denote the size of the diffusion particles with D and the size of the position particles generated by each diffusion particle with P. We can also make the algorithm only one

chain dependant by setting the population size of the other chain to 1.

3.2.1. Initialization

We do the initial sampling of the diffusion particles, $k_0^{(i)}$, that are distributed along a uniform distribution. The position particles are initially sampled like in the original SMC filter for localization, but for a different sample size of P per diffusion particle:

$$\begin{aligned} k_0^{(i)} &\sim \mathcal{U}([k_{min}, k_{max}]),\ i \in [1, \dots, D] \\ \mathbf{x}_0^{(i,j)} &\sim \mathcal{U}(\mathcal{A}),\ i \in [1, \dots, D],\ j \in [1, \dots, P]. \end{aligned} \tag{3.14}$$

3.2.2. Importance Sampling

Diffusion particles are sampled along a gamma distribution, aligned with the transition density given in (3.3) with $i \in [1, \dots, D]$. Position particles are sampled as in Section 3.1.2, but for two indices: $i \in [1, \dots, D]$ for the diffusion particle and $j \in [1, \dots, P]$ defining the index in the population sampled by the associated diffusion particle:

$$\begin{aligned} \tilde{k}_t^{(i)} &\sim \mathcal{G}(a_t^{(i)}, b_t^{(i)}),\ i \in [1, \dots, D] \\ &\text{with } a_t^{(i)} = \frac{(k_{t-1}^{(i)})^2}{\nu} + 1 \\ &\text{and } b_t^{(i)} = \frac{\nu}{k_{t-1}^{(i)}} \\ \tilde{\mathbf{x}}_t^{(i,j)} &\sim \mathcal{N}(\mathbf{x}_t^{(i,j)}, k_t^{(i)} \mathbf{I}_2)),\ i \in [1, \dots, D],\ j \in [1, \dots, P]. \end{aligned} \tag{3.15}$$

Small values of the sensitivity make the diffusion estimations rigid, keeping the weighted average of the diffusion factor almost static, whereas with the higher values we observe rapidly varying values for the same statistic, making the diffusion factor

estimations behave aligned with the measurement errors. As a consequence, higher values may seem to be preferred, but we will show that unnecessarily high sensitivity values can also cause undesired fluctuations for the diffusion factor estimations.

3.2.3. Particle Evaluation

The position particles, $\omega_t^{(i,j)}$, are evaluated against the frequency map and the occupancy map to obtain the particle weights as before (see Section 3.1.3). The diffusion particles, $\upsilon_t^{(i)}$, are evaluated by accumulating the corresponding position particle weights:

$$\begin{aligned} \omega_t^{(i,j)} &= \tilde{h}(y_t^s; \tilde{\mathbf{x}}_t^{(i,j)}, \alpha, \beta) r(\tilde{\mathbf{x}}_t^{(i,j)}) \\ \upsilon_t^{(i)} &= \sum_j \omega_t^{(i,j)}. \end{aligned} \tag{3.16}$$

Both set of particles are then normalized as in the following:

$$\begin{aligned} \tilde{\omega}_t^{(i,j)} &= \frac{\omega_t^{(i,j)}}{\sum_{i,j} \omega_t^{(i,j)}} \\ \tilde{\upsilon}_t^{(i)} &= \frac{\upsilon_t^{(i)}}{\sum_i \upsilon_t^{(j)}}. \end{aligned} \tag{3.17}$$

3.2.4. Resampling

With the normalized weights for the set of particles, the resampling stage is the same with the resampling in Section 3.1.4. The SMC filter with parameter sampling is given in Figure 3.4. The bold vectors or matrices, $\tilde{\mathbf{k}}_t$, $\tilde{\boldsymbol{\upsilon}}_t$, $\mathbf{X}_t^{(i)}$ and $\tilde{\boldsymbol{\omega}}_t$, stand for the sets of the corresponding particle variables.

1: **Initialization**

Initialize a set of particles:

for $i \in [1, \ldots, D]$, $k_0^{(i)} \sim \mathcal{U}([k_{min}, k_{max}])$

for $j \in [1, \ldots, P]$, $\mathbf{x}_0^{(i,j)} \sim \mathcal{U}(\mathcal{A})$

2: **Importance Sampling**

Update the time index

$t \leftarrow t + 1$

Sample a new set of particles from the current sets:

for $i \in [1, \ldots, D]$, $\tilde{k}_t^{(i)} \sim \mathcal{G}(a_t^{(i)}, b_t^{(i)})$,

for $j \in [1, \ldots, P]$, $\tilde{\mathbf{x}}_t^{(i,j)} \sim \mathcal{N}(\mathbf{x}_{t-1}^{(i,j)}, k_t^{(i)}\mathbf{I}_2)$

Evaluate the proposed set with the RF and occupancy maps to obtain the importance weights:

for $i \in [1, \ldots, D]$,

for $j \in [1, \ldots, P]$, $\omega_t^{(i,j)} = \tilde{h}_{y_t^s}(\tilde{\mathbf{x}}_t^{(i,j)}, \alpha, \beta) r(\tilde{\mathbf{x}}_t^{(i,j)})$

$v_t^{(i)} = \sum_j \omega_t^{(i,j)}$

Normalize the importance weights:

$\tilde{\omega}_t^{(i,j)} = \frac{\omega_t^{(i,j)}}{\sum_{i,j} \omega_t^{(i,j)}}$

$\tilde{v}_t^{(i)} = \frac{v_t^{(i)}}{\sum_i v_t^{(j)}}$

3: **Resampling**

Resample $D \times P$ particles according to the importance weights

for $i \in [1, \ldots, D]$, $k_t^{(i)} \sim Syst(\tilde{\mathbf{k}}_t, \tilde{\boldsymbol{v}}_t)$

for $j \in [1, \ldots, P]$, $\mathbf{x}_t^{(i,j)} \sim Syst(\tilde{\mathbf{X}}_t^{(i)}, \tilde{\boldsymbol{\omega}}_t)$

4: **Recursion**

Go back to 2, if not at the end of the trajectory

Figure 3.4. Parameter sampling based Adaptive SMC Filter for BLE Localization.

4. HIGH PRECISION DATA ANNOTATION

Data collection and annotation methodology comprises four main consecutive sections as seen in Figure 4.1. The inputs to the system are the collected images or video frames, timestamped RSSI readings from the test area and the stationary marker poses.

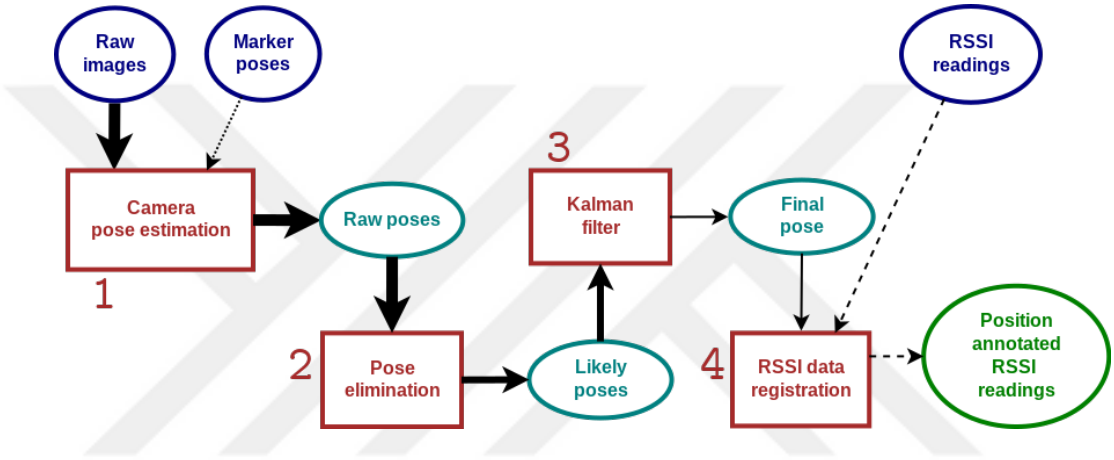


Figure 4.1. The workflow aims to collect maximum information available, filter the most relevant pose, and synchronize it with the RSSI readings. The arrow weights denote the data load coarsely.

Firstly, we employ the ArUco marker detector to estimate marker poses (positions and orientations) relative to the camera(s) [95, 96]. This information is combined with the previously recorded marker poses relative to the world to estimate the pose of the camera setup in the world frame. Secondly, we apply pruning techniques to eliminate the misleading data in a batch of estimated poses, which we observe to be very noisy due to erroneous detections. Thirdly, we apply a Kalman filter to obtain the most relevant pose given a handful of poses. Finally, these accurate camera poses are registered in time domain with the RSSI readings.

4.1. Camera Pose Estimation

The ArUco suite provides detector routines for the markers. With properly calibrated cameras and measured stationary markers, we can detect markers and obtain their positions and orientations both in three dimensions with high accuracy in real time as long as the lighting conditions are appropriate.

In this section, the variables, m, c and w denote the entities, a marker, the camera setup and the world, respectively. We represent the position (translation) and the orientation (rotation) vectors with ${}^{i}\boldsymbol{\tau}_{jk}$ and ${}^{i}\boldsymbol{\theta}_{j}$ respectively. The superscript on the left, i, is for the frame the vector is defined in. The subscript on the right, j, is for the entity the vector belongs to. For the position vectors, a double subscript jk shows the start and the end points. If the start position is the frame origin itself, we omit the first entity, ${}^{j}\boldsymbol{\tau}_{jk} \equiv {}^{j}\boldsymbol{\tau}_{k}$. For example, ${}^{c}\boldsymbol{\theta}_{m}$ defines the orientation vector of the marker m, in the camera frame, c, and ${}^{w}\boldsymbol{\tau}_{cm}$, specifies a vector from the camera c to the marker m in the world frame w.

We use rotation matrices to convert vectors from a predefined coordinate frame to another. A position vector defined in the frame i is multiplied with the rotation matrix, ${}^{k}_{i}\mathcal{R}$, from the left to be converted into the coordinate frame k. Note that with a proper multiplication, the coordinate frames in the diagonals are eliminated, (i in (4.1)). Similarly, if we have a rotation matrix from the frame i to j and another one from j to k, we can obtain transitively the rotation matrix from i to k by multiplying them properly as in (4.1). Note the elimination of the frame j in the following:

$$\begin{aligned} {}^{k}\boldsymbol{\tau}_{j} &= {}^{k}_{i}\mathcal{R}\ {}^{i}\boldsymbol{\tau}_{j} \\ {}^{k}_{i}\mathcal{R} &= {}^{k}_{j}\mathcal{R}\ {}^{j}_{i}\mathcal{R}. \end{aligned} \tag{4.1}$$

If we have an orientation vector of the entity i defined in the coordinate frame j, we can obtain the associated rotation matrix from i to j, using the Rodrigues' rotation

formula, and vice versa (4.2) [97], as in the following conversions:

$$\begin{aligned} {}^{j}\boldsymbol{\theta}_i &= Rodrigues({}^{j}_{i}\mathcal{R}) \\ {}^{j}_{i}\mathcal{R} &= Rodrigues({}^{j}\boldsymbol{\theta}_i). \end{aligned} \tag{4.2}$$

We denote the positions of stationary markers as ${}^{w}\boldsymbol{\tau}_m$, and their orientations as ${}^{w}\boldsymbol{\theta}_m$. When markers are detected and recognized in a camera frame, the pose estimator supplied by the ArUco suite estimates the position vectors pointing to the center of the marker, ${}^{c}\boldsymbol{\tau}_m$, and orientation vector, ${}^{c}\boldsymbol{\theta}_m$, relative to the camera coordinate frame.

We are interested in obtaining the camera pose relative to the world frame, (${}^{w}\boldsymbol{\tau}_c$, ${}^{w}\boldsymbol{\theta}_c$). This is achieved by multiplying the measured marker poses relative to the world, (${}^{w}\boldsymbol{\tau}_m$, ${}^{w}\boldsymbol{\theta}_m$), with the detected marker poses relative to the camera frame, (${}^{c}\boldsymbol{\tau}_m$, ${}^{c}\boldsymbol{\theta}_m$). For positions, we do:

$$\begin{aligned} {}^{w}\boldsymbol{\tau}_c &= {}^{w}\boldsymbol{\tau}_m - {}^{w}\boldsymbol{\tau}_{cm} \\ &= {}^{w}\boldsymbol{\tau}_m - {}^{w}_{m}\mathcal{R}\ {}^{m}_{c}\mathcal{R}\ {}^{c}\boldsymbol{\tau}_{cm} \\ &= {}^{w}\boldsymbol{\tau}_m - ({}^{m}_{w}\mathcal{R})^{T}\ {}^{m}_{c}\mathcal{R}\ {}^{c}\boldsymbol{\tau}_m \end{aligned} \tag{4.3}$$

where the right hand side of the second line is a reformulation using the measured or captured information in hand. ${}^{w}\boldsymbol{\tau}_{cm}$ is the vector from the camera to the marker relative to the world coordinate frame. Using Rodrigues' rotation formula, we first convert the orientation vector of the marker frame in the world frame, ${}^{w}\boldsymbol{\theta}_m$, to the rotation matrix from the world frame to the marker frame, ${}^{m}_{w}\mathcal{R}$, and then the marker orientation vector in the camera frame, ${}^{c}\boldsymbol{\theta}_m$, to the rotation matrix from the camera frame to the marker frame, ${}^{m}_{c}\mathcal{R}$. We also have the marker position in the camera frame, ${}^{c}\boldsymbol{\tau}_m$, and multiplying them appropriately we attain the required position vector, ${}^{w}\boldsymbol{\tau}_{cm}$, hence, the camera position vector in world coordinates, ${}^{w}\boldsymbol{\tau}_c$, for each detected marker (see Figure 4.2).

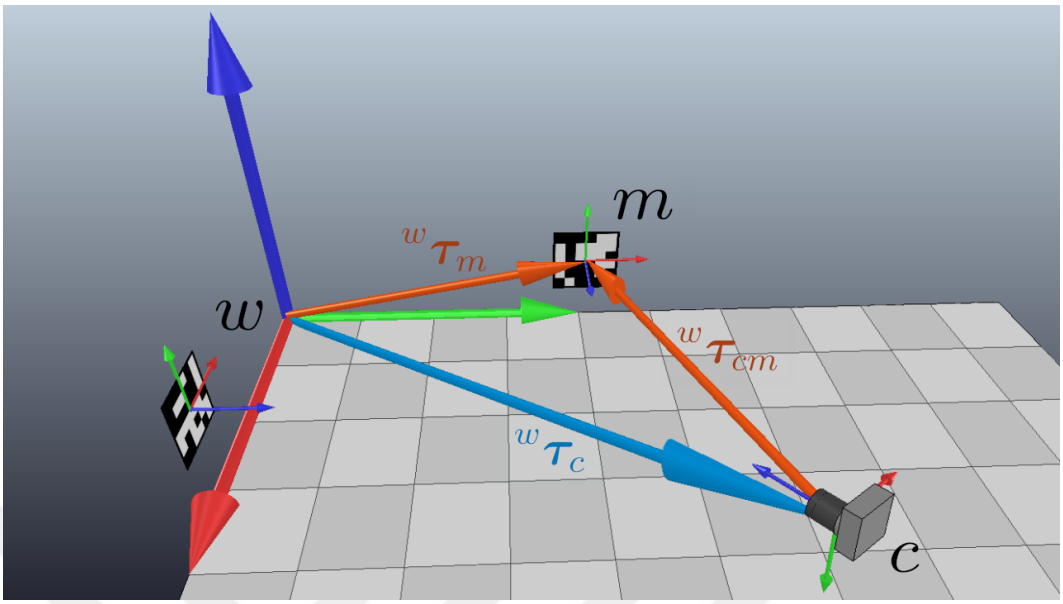


Figure 4.2. The position vectors and the coordinate frames: w, c and m; for the world, the camera and the marker frames respectively.

For orientations, we use the rotation matrix from the camera frame to the world frame, ${}^{c}_{w}\mathcal{R}$. Whereas this matrix can be used to convert a vector from camera coordinate system to the world coordinate system, it also defines how the camera is oriented within the world frame. Employing the Rodrigues' rotation formula, we convert the rotation matrix into the orientation vector of the camera in the world frame (4.4):

$$ {}^{w}\boldsymbol{\theta}_{c} = Rodrigues({}^{w}_{c}\mathcal{R}). \tag{4.4} $$

The conversions above are repeated for each detected marker in an image frame. Since we construct our detection setup aiming to capture several markers at a time, we expect to estimate multiple camera pose vectors at each camera frame.

4.2. Pose Elimination

The variances of the camera pose estimations are observed to be large due to noisy marker pose detections caused by minor errors in manual measurements, faulty camera

matrices and coefficients, motion blur or varying lighting conditions. The combination of these factors lead to inaccurate rotation matrices, which in turn accumulate to varying positions for the camera. Before applying a filtering algorithm for camera poses, we employ two elimination strategies (see Figure 4.3).

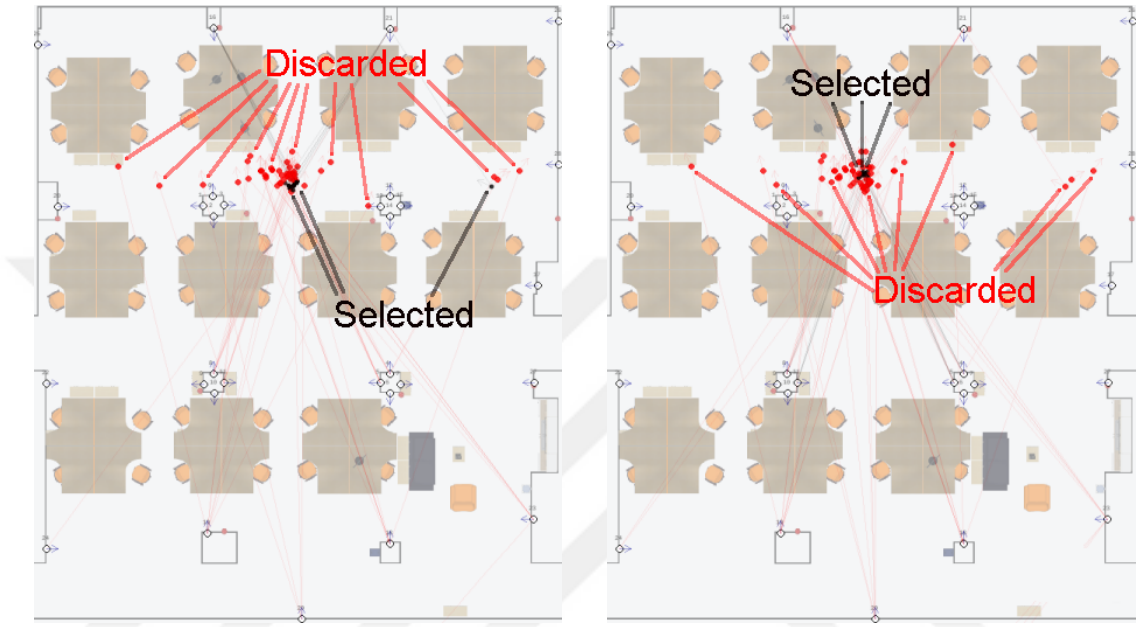


Figure 4.3. Position elimination strategies: close marker selection (left) and outlier elimination (right). Black points are selected for the next stage, while the red points are eliminated.

At each camera frame, we expect to capture multiple marker readings. We use two cameras to expand the field of view, so that if one camera misses the markers, the other one captures possibly more than one marker. Moreover, to make sure that there exist multiple marker readings, we process F camera frames in one shot. For example, with $F = 15$ and navigating at 0.35 m/s, each batch of frames corresponds to about 0.04 meters.

We observe that camera poses from distant markers can be more misleading. This defect is due to the number of pixels and the resolution of the camera. As the distance increases, the lack of pixels lead to larger measurement errors. Therefore, as long as multiple poses are estimated in a time window of F frames, the poses based on distant markers may be discarded. We set a closeness parameter, S, that takes into account

the closest S marker readings. This is a fast technique that sorts the detected markers in a frame with respect to their first coordinates (distance to the camera) and extracts the S poses with the minimum marker distances.

Another elimination is achieved by an outlier detection test. In a set of pose estimations, we eliminate the outliers with respect to the distances to the consensus point [98]. The parameter U, is defined to determine the number of remaining poses after consecutive outlier detection iterations. This technique is merely a slow backward subset selection method that is repeated in a greedy manner.

4.3. Kalman Filter for Smoothing

Even though some marker detections are eliminated, we still receive erroneous estimations. Error fluctuations can be controlled by employing appropriate filtering mechanisms. The current section is linked to the previous section through pose vector estimates of the camera setup, which will be our observations in the filtering design: $\mathbf{z}_t = ({}^c\boldsymbol{\tau}_w, {}^c\boldsymbol{\theta}_w)_t$. We model this new problem as a state space model, in which the observed poses, y_t, are assumed to be generated noisily from the latent poses, $\mathbf{x}_t$. The pose vector of the tracked object, $\mathbf{x}_t$ at time t, is a function of the pose vector at a previous time, $\mathbf{x}_{t-1}$. This function is also called the motion model that stands for the dynamics of the object.

The transition densities are assumed not to follow explicit motion models. Instead, we assume that $\mathbf{x}_t$ resides close to $\mathbf{x}_{t-1}$. Observation densities are designed similarly. $\mathbf{z}_t$ should not be far away from $\mathbf{x}_t$. With a prior distribution at the initial time, $p(\mathbf{x}_0)$, we are to find the filtering distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$. These assumptions can be realized by forming transition and emission densities with Gaussians (4.5), whose

noise covariance matrices will be denoted as $\mathbf{Q}_t$ and $\mathbf{R}_t$ respectively:

$$\begin{aligned} \mathbf{x}_0 &\sim p(\mathbf{x}_0) \\ \mathbf{x}_t &\sim \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{Q}_t) \\ \mathbf{z}_t &\sim \mathcal{N}(\mathbf{z}_t; \mathbf{x}_t, \mathbf{R}_t) \end{aligned} \tag{4.5}$$

The exact inference for the filtering distribution becomes tractable, and can be solved with the standard Kalman filter. The dimensions of the latent space and the measurement space are assumed to be independent. The values of the covariance matrices are chosen to compensate for unexpected high errors in measurements [32,99]. The recursive equations of the Kalman filter follows:

Predict:

$$\begin{aligned} \hat{\mathbf{x}}_t &= \mathbf{A}\mathbf{x}_{t-1} \\ \hat{\mathbf{P}}_t &= \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^T + \mathbf{Q}_t \end{aligned}$$

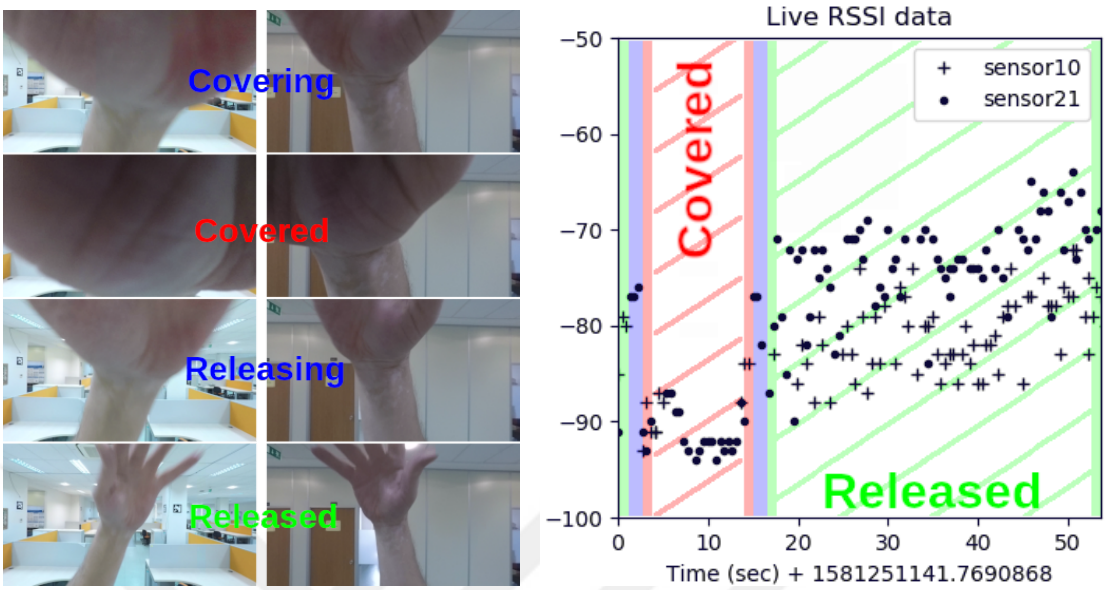
Update: (4.6)

$$\begin{aligned} \mathbf{G}_t &= \hat{\mathbf{P}}_t\mathbf{J}^T(\mathbf{J}\hat{\mathbf{P}}_t\mathbf{J}^T + \mathbf{R}_t)^{-1} \\ \mathbf{x}_t &= \hat{\mathbf{x}}_t + \mathbf{G}_t(\mathbf{z}_t - \mathbf{J}\hat{\mathbf{x}}_t) \\ \mathbf{P}_t &= (\mathbf{Id}_d - \mathbf{G}_t\mathbf{J})\hat{\mathbf{P}}_t \end{aligned}$$

where transition and observation matrices, $\mathbf{A}$ and $\mathbf{J}$, are chosen as identity matrices ($\mathbf{Id}_d$), $\mathbf{P}_t$ are the estimated covariance matrices, and $\mathbf{G}_t$ are the optimal Kalman gains. We obtain $\mathbf{x}_t$ as the pose estimations.

4.4. RSSI Data Registration

After the filtering operation, timestamped pose data are obtained, but there still exist problems in synchronizing the timestamped RSSI data to timestamped position data.



(a) Detecting a covering visually.

(b) Detecting a covering in RSSI readings.

Figure 4.4. Disturbing the system artificially to synchronize the timestamps of the video frames with RSSI data.

The first problem is that the cameras cannot be synchronized to a precise world clock. We overcome this problem by adding artificial disturbances that would affect both the camera views and the RSSI data, so that synchronization may be possible: the setup consisting of a beacon and cameras is covered with hands, easily detectable by the cameras (see Figure 4.4(a)). Covering the setup with hands also creates significant reduction in the RSSI data because body parts absorb electromagnetic signals at 2.4 Ghz, so the artificial disturbance can be detectable in RSSI readings too (see Figure 4.4(b)). Registering the start time of both data into world clock results with a fine synchronization of the RSSI readings and camera frames.

Secondly and obviously, RSSI timestamps and pose timestamps do not match exactly. RSSI data is far less frequent than the pose data. To coincide the two data types in time domain, we perform linear interpolation of the pose data at RSSI timestamps, and we label each RSSI timestamp with the interpolated poses as its ground truth.

After the time registration stage, we obtain a dataset of RSSI data points annotated with the position it is captured on. Two of the trajectories from our dataset are displayed in Figure 4.5.

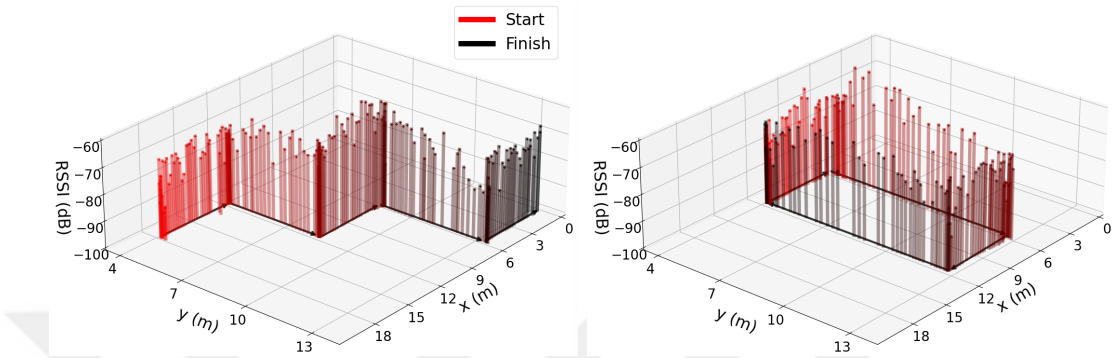


Figure 4.5. RSSI data overlaid on the position data: time flows from red to black.

5. EXPERIMENTAL SETUP

We designed two different experimental setups. The first one is a domestic area, actually an actively used living room of an apartment. We track a mobile sensor that captures data from the stationary beacons attached on the walls. The experiments for emission density estimation are performed on the data collected in this domestic setup.

A second setup is designed to cover a larger area, an office environment of size 340 m^2. The tracked device is replaced with a beacon. Several surrounding sensors capture data from the mobile beacon. We also record visual data in this second setup with a special platform that holds the cameras and the beacon together, so that ground truth positions can also be collected.

5.1. Domestic Area

5.1.1. Beacons and Sensors

We use multiple stationary beacons that serve as message emitters. These are the commercial Bluetooth Low Energy Beacons supplied by Boni working at around 1 Hz each. The placement of the beacons are shown in colored circles in Figure 1.4(a).

A USB Bluetooth dongle receives the BLE messages and reads RSSI values that are used for both fingerprinting and tracking purposes. The Bluetooth dongle is attached to the top of a tripod (1.5 meters of height) to elevate the sensor in order to increase the line-of-sight with the beacons (see Figure 5.1).

5.1.2. Test Area and Fingerprints

We use a living room of a flat actively used by a married couple to collect data. A plan of the room is given in Figure 1.4(a). The area has a length of 6.66 meters



Figure 5.1. Bluetooth dongle on a tripod and beacons mounted on the walls.

and a width of 5.36 meters. To gather data, we log the incoming BLE RSSI data with respect to the beacon MAC address on different points in the test area.

On the selected locations of the area, we set up the Bluetooth dongle to dump the incoming raw BLE RSSI data for 24 hours, corresponding to about 85k data points for each beacon, in order to obtain a data spectrum as redundant and as time invariant as possible. Histograms on each position for each dongle is then generated using these data. In our work, a total of 50 days of BLE data are collected with one location per day making 50 different positions on the map (Figure 1.4(a)). Histograms are normalized to represent a density of the RSSI values on a point on the map. If no data belonging to a beacon can be captured on a certain point, then its corresponding distribution is set to constant zero (see Figure 1.4(b)). The normalized version of the histograms are called the fingerprints of the specific indoor area, which represent a summary of the map in the means of BLE data, or a probabilistic RF map.

This work aims to encode the location (not distance) information with the fingerprints. So a previous scene analysis is surely required. Should a new beacon be added

into the system, its fingerprint information on specific locations should also be added. In our setup, we naturally receive data from other beacons, or other Bluetooth based transmissions. We filter them out as we do not assume to have any previously known "location vs RSSI " based information related to those beacons.

Because measurements are conducted in an actual living room, where interferences due to Wi-Fi from multiple sources, other Bluetooth-based signals, multiple living beings and metallic masses are highly probable.

5.2. Office Area

In the second test setup, we use a set of BLE beacons and adapters, low profile computers, motion cameras and visual markers installed in an indoor area of size 17×20 m^2. Intended originally as office space for fellow graduate students, the area also includes columns, walls and furnitures that can influence the wireless signals (see Figure 5.2). We expect to collect RSSI data easily using the beacons and adapters, and annotate these data with the positions that are accurately estimated by using the cameras and the visual markers.

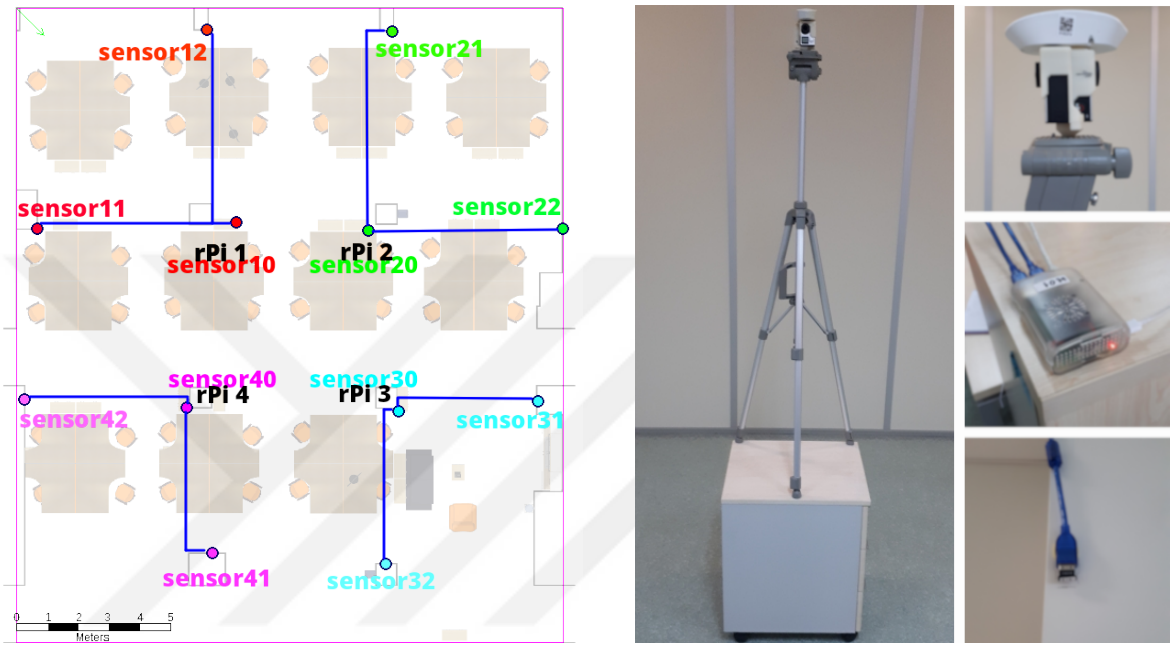


Figure 5.2. Sample calibration images from the area including markers, extension cables, Bluetooth adapters, computers and other obstacles.

5.2.1. Hardware Setup

We use POI Beacons as transmitters, which are constructed upon Nordic Semiconductor multi-protocol SoC and 32-bit Arm Cortex M0 processor with 256 kB flash

and 16 kB on chip. They use the Bluetooth 4.0 BLE protocol and iBeacon protocol, and have a signal range of 70 meters. Beacons send two BLE identity packets per second and have a running time up to 4 years with two standard AA batteries (see Figure 5.3(b)).



(a) Sensor distribution in the plan of the area: colored circles show the sensors and the blue cables are their connections to the computers.

(b) The navigable platform (left) and the custom case of beacon and the cameras, a computer and a sensor (right).

Figure 5.3. Hardware setup.

There are 12 Bluetooth sensors stationed in the test area (see Figure 5.3(a)). We use 8 Logilink Bluetooth 4.0 adapters and 4 onboard Bluetooth embedded chips driven by the computers (see Figure 5.3(b)). The computers are put on the cabinets to ensure that their embedded sensors have a standard elevation of one meter. Two adapters are connected via USB to each computer using extension cables. The cables are attached to the ceiling to minimize possible signal interferences. The adapters are attached on the walls elevated about two meters from the floor. The extension cables are tested to supply enough power to the adapters (see Figure 5.2).

The distributed data capture system is mainly constructed upon four Raspberry Pi 3 Model B+ single board computers, which are based on Broadcom BCM2837B0,

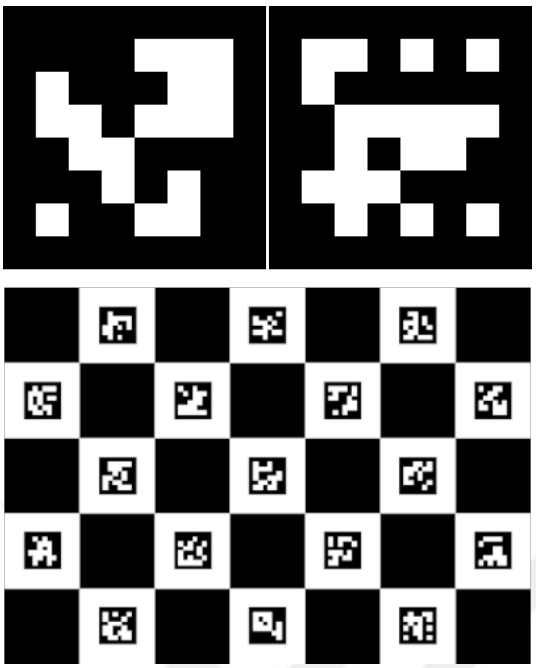
Cortex-A53 (ARMv8) 64-bit quad-core processors. They run Raspbian Stretch operating system, compatible with the Robot Operating System (ROS) Melodic middleware. The computers scan the BLE packets emitted by the transmitters and publish them into the local area network (see Figure 5.3(b)).

The motion cameras are GoPro Hero 4, capturing images in a resolution of 1920×1080 at 60 fps. We use a 3D printed custom case that holds two motion cameras and the BLE beacon together (see Figure 5.3(b)). The cameras are placed back to back, and with their lenses aligned facing opposite directions. One camera stays upside down with respect to the other one. The design minimizes the translations between camera coordinate frames and maximizes the functionality of the cameras by allowing reaching both of the buttons.

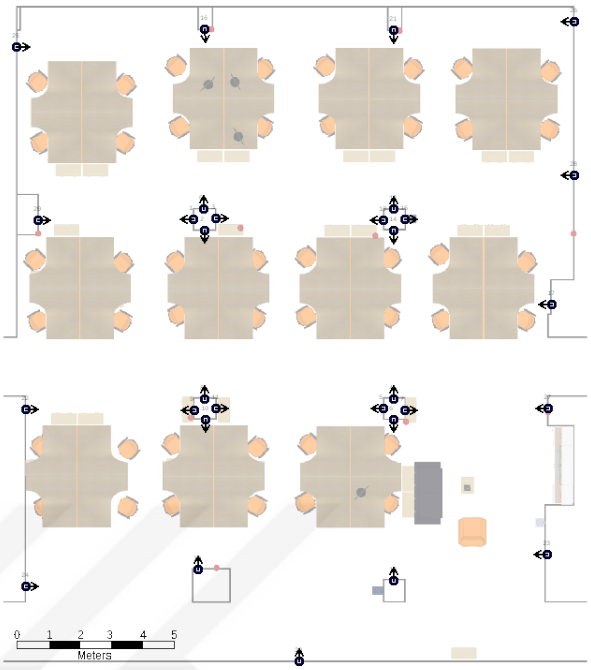
The beacon is put on the custom setup which is attached onto a tripod which in turn is mounted on a wheeled cabinet that allows smooth navigation in the test area (see Figure 5.3(b)). The beacon height is set to be 2 meters to obtain a clear view of the test area in terms of both visual data and RF data. The tripod functionality also allows us to switch to different alignments of the camera setup, from which we prefer horizontal alignment to maximize the view angle, hence, the number of detectable markers, and to minimize the interference on the beacon due to the cameras.

5.2.2. Markers and Camera Calibration

We choose 30 square markers from the dictionary of 6×6 matrix markers of the ArUco marker suite (Figure 5.4(a)), and print them on hard paper with edges of 0.3 meters. The markers are attached onto the walls or columns where they can be viewed from almost any angle in the area (Figure 5.4(b)). We record the label and the pose (of the center) of each marker with respect to the predetermined world coordinate frame $({}^{w}\boldsymbol{\tau}_m, {}^{w}\boldsymbol{\theta}_m)$.



(a) ArUco marker samples (top) and the calibration chart (bottom).



(b) The markers in the test area as black dots and arrows.

Figure 5.4. ArUco markers and their distribution in the test area.

The ArUco suite provides calibration procedures as well as marker generation and detection. Using several views of the Charuco calibration chart [100] (Figure 5.4(a)), the camera matrix and the corresponding distortion coefficients are tuned. Highly accurate camera coefficients are mandatory for precise pose detection of the markers. We prefer the use of planar view setup of cameras to the fish eye mode because of its easy calibration procedure and large enough view angles.

The current setup allows us to have an almost spherical world view with the fish eye mode of the cameras, but we avoid using the fish eye mode of the cameras for three reasons: (i) the calibration procedure is challenging for the fish eye view, (ii) the fish eye distortions have to be corrected before running the AR Marker detection algorithms as they have to be fed with planar markers, and (iii) we do not need to see ceilings and with the planar view setup, the cameras already have a large enough angle of view.

5.2.3. Handling Setup Errors

The errors in marker based localization is due to multiple factors that can be divided into two sets: human errors and environmental errors. Human errors include mostly errors in measuring the marker poses, reference points and unstable navigation. We can also put the camera calibration based errors into this set, which lack the necessary views for the calibration process. Environmental errors are due mostly to ambient light fluctuations.

The markers are installed with the following factors taken into consideration in order to minimize the errors due to human factors: (i) From any position with any orientation in the area, at least one marker should be detectable. Markers should be above objects that can cause occlusion. (ii) Markers are attached in the places illuminated uniformly and sufficiently. (iii) Markers are placed with the help of a jig to ensure that they are at a fixed height and orientation to facilitate the height and rotation measurements.

The ArUco suite provides calibration procedures as well as marker generation and detection. Using several views of the Charuco calibration chart [100] (Figure 5.4(a)), the camera matrix and the corresponding distortion coefficients are tuned. Highly accurate camera coefficients are mandatory for precise pose detection of the markers.

Before running the main set of experiments that will assess the trajectory estimation performance, we collect a preliminary set of controlled images of each marker to evaluate the calibration and detection performance. The navigable platform is put in the area in front of each marker at different angles and distances. Sample images can be seen in Figure 5.2.

By adjusting the camera matrix and the parameters to their best values, we achieve an empirical minimum mean squared distance error of 0.05 meters for all markers installed in the area. A highly controlled experiment can minimize the errors due

to the human factors, but environmental factors still accumulate on the errors. With a controlled error rate under 0.1 meters, marker based localization is still highly better than any other wireless based localization method, and can be used as the ground truth to evaluate the RF RSSI based localization performance.

For the static recording of positions using AR markers; It is observed that modern office installations that have a raised floor are prone to flexing while people are stationary or mobile in the proximity of the image capture device. The tilting or oscillation of the image capture device can have a small effect on the captured position information. For calibration and verification purposes, the authors waited until the capture platform was stabilised and image capture was triggered wirelessly to obtain the best results.

The installation of the AR markers is a tedious and time consuming work, but this procedure can be eased by considering the following points: (i) Instead of sticking the markers on the walls, fixing them on portable stands at predetermined elevations will mitigate the installation time significantly. Polycarbonate stands should be preferred to minimize wireless signal attenuation and interference due to these artificially introduced materials. The portable marker stands can be removed from the area quickly after the data collection stage. (ii) Since markers can be recognized from any angle, choosing common orientations like the multiples of $\frac{\pi}{2}$ will help greatly. With standard elevations and easy orientation labeling, the experimenter is left with measuring the marker positions in 2D coordinates. (iii) We used two motion cameras not to miss any marker at any time interval, but this can be regarded as excessive. A single camera setup will also be sufficient to detect the markers, but the focus of the camera should be static, as variable focal length will disrupt the marker pose estimations. (iv) Instead of mobile beacons, one can use a mobile sensor (i.e. computer or smartphone) to capture the BLE packets emitted by the beacons that furnish the area. Several other sensors can be introduced into the setup, leading to more data that can improve the position estimation accuracy.

5.2.4. Data Collection Software Setup

For sharing data among multiple computers connected through Ethernet, we use the ROS middleware (see Figure 5.5). For the purpose of data collection, we create three main nodes: We have a master node for orchestration and entry point for other nodes. We prefer to dedicate a simpler computer for the master node that does not deal with any data processing. It is responsible for triggering the publishers in the network. The network can easily be expanded with other nodes. A node that wants to enter into the ROS data collection network has to present itself to this master node. Publishers (ii) are responsible for scanning the Bluetooth devices on the computers, generate RSSI parameters from incoming Bluetooth packets and publish these parameters as RSSI topic messages. The published messages include a timestamp, MAC address of the source beacon, MAC address of the sensor that captures the Bluetooth packet and the RSSI value in decibels. Finally, we have a visualizer (iii) that subscribes to the RSSI topic messages and displays them in live windows with respect to time. The visualizer is also responsible for recording the data in log files for each sensor.

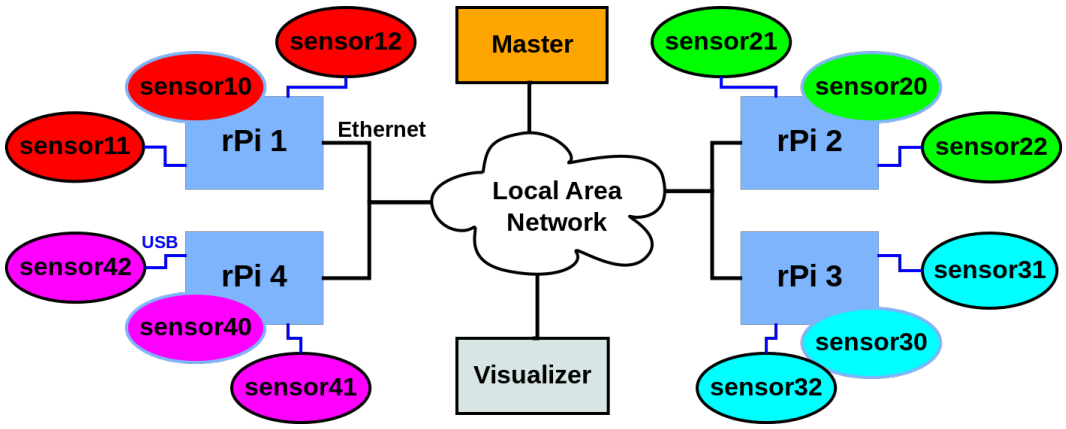


Figure 5.5. Data collection and communication architecture.

The master, publisher and visualizer nodes are running Python-3 scripts. The visualizers have also graphical user interface based on PyQt wrappers (see Figure 5.6).

To ease the synchronization of video and RSSI data, we develop and utilize a custom toolkit written with PyQt. A snapshot of the toolkit can be seen in Figure 5.7.

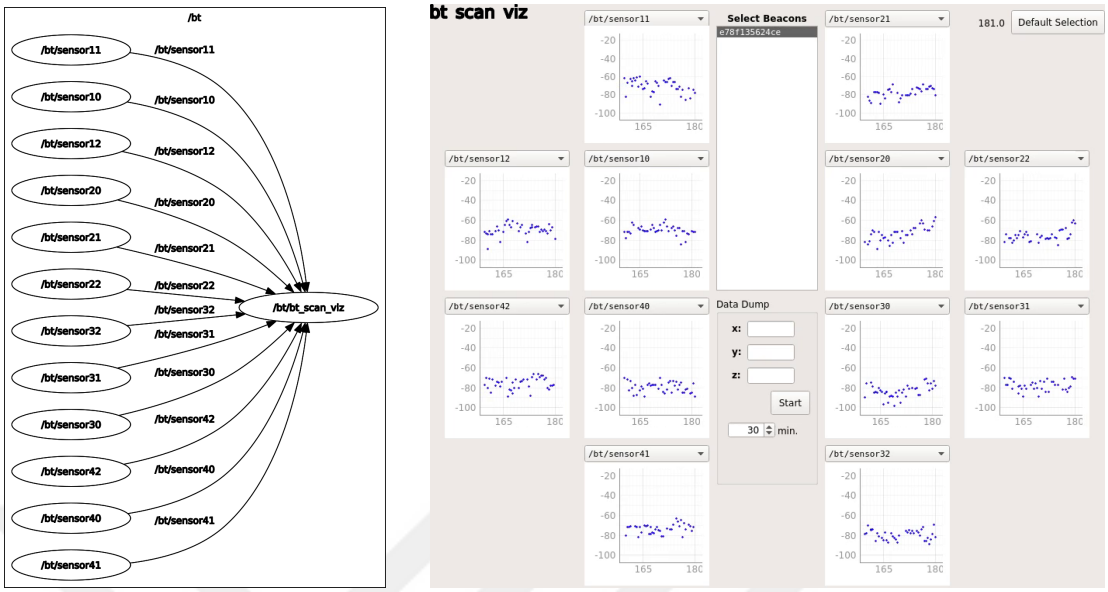


Figure 5.6. RSSI data flow on ROS (left) and the live data visualizer (right).

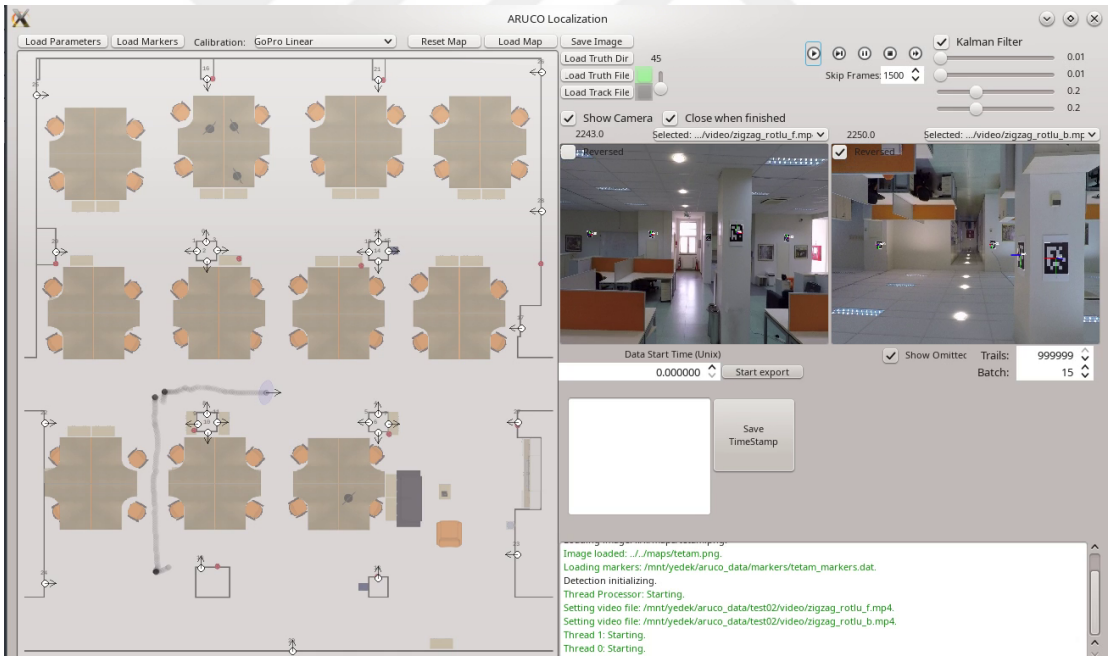


Figure 5.7. AR localization toolkit for estimating, visualizing, synchronizing and logging data.

The toolkit is used for performing several tasks required for calibration, pose estimation and registration processes, such as displaying video frames from two cameras and navigate through the frames freely; detecting markers given the proper calibration parameters and displaying them in video frames; estimating raw positions given the

markers; applying different elimination strategies; toggling Kalman filter and tuning its noise parameters on the fly; displaying raw, eliminated or smoothed position estimations on the map with their related markers; taking snapshots from the map (most of the figures including maps in this article are generated by this toolkit); marking the needed timestamps for synchronization; displaying logged or predetermined trajectories; and logging the obtained position estimations into data files. The framework will be available for the researchers.

5.2.5. Position Annotated Dataset

The dataset provides the trajectory data files that include the position annotated RSSI data collected while navigating the special platform in the test area. There are 15 different trajectory files with varying lengths and forms. Each sample in a file comes with a timestamp, a transmitter address, a sensor address, an RSSI reading, a ground truth position in 3D and its associated orientation matrix. We also supply the map of the test area, the files for the configuration of the sensors and various parameters that are required to convert the measurements in meters into pixels, and vice versa. For the completeness of an indoor localization project, we publish auxiliary data of raw fingerprints, their processed histograms, two sets of estimated probabilistic RF maps in the grid form and two sets of occupancy maps [37]. The dataset will be available for the researchers that work in the field.

6. EXPERIMENTS AND RESULTS

We divide this section into three parts to describe various experiments, corresponding data, search space for different parameters and obtained results. The first section is for the experiments to assess the emission density estimation method using AWC. We use a standard SMC filter in the experiments. The second section describes the experiments on how to collect RSSI data and associated ground truth data. With an emission density estimation model and position annotated RSSI data, we conduct our final batch of experiments which evaluates the adaptive SMC filter that jointly estimates the diffusion factor and the positions.

6.1. Experiments for Affine Wasserstein Combination

We test the RF map models that will be used as the emission density to update the particles given the RSSI readings. We compare the performances of the estimated maps formed by different interpolation methods using different fingerprint configurations.

6.1.1. Model Parameters

Emission density estimators have their own model parameters (represented previously as Θ). Before employing the tracking algorithms with different estimators, we search for their optimal values by trial-and-error using the leave-one-out cross-validation strategy [101, 102] using the full set of fingerprints (50 measurement locations) and search for the parameter configurations that minimizes the error between the estimated histogram and the original fingerprint. We report these parameters as a list:

- Nearest Fingerprint (NF):
  - No parameters
- k-Nearest Fingerprints (κ-NF):

  - Number of nearest fingerprints: $\kappa = 8$
- Artificial Neural Networks with one layer (ANN):
  - Number of hidden layer nodes: $N_h = 40$
  - Hidden layer activation function: tanh
  - Output layer activation function: sigmoid
- Affine Wasserstein Combination with the best two positions (AWC-B):
  - Distance to decibel mapping function: linear
  - Evolution type parameter: $\beta = 0.36$
  - Looseness parameter: $\rho = 0.63$
- Affine Wasserstein Combination with multiple positions (AWC-E):
  - Distance to decibel mapping function: linear
  - Evolution type parameter: $\beta = 0.38$
  - Looseness parameter: $\rho = 0.53$

We also search for the optimal diffusion factors (k) of the predictive distribution for the diffusion model (Section 2.1) that maximizes the localization performance. For this purpose, we take tests of the whole tracking system with different observation density estimators using different diffusion factors for the predictive distribution. With an exhaustive search, we find and set a near-optimal diffusion factor as $k = 0.38$.

6.1.2. Parameter Search

The parameter β in $[0, 1]$ of the Wasserstein interpolation defines how far the measures will be distributed. If β is high, the measures make a gradual movement towards the destination distribution. If low, then the measures jump directly onto the destination distribution. In other words, β defines a metric from a smooth transition to a convex combination between two distributions.

With the fingerprints collected from the scene, we search the most likely β parameter using the LOO-CV approach, in which we take out a point and find the points that can be used to estimate the histogram on the removed point with different orien-

tations. The distribution of β can be seen in Figure 6.1. The votes for the likely values of β seems to pile up towards 1. This leads us to take $\beta = 1$ to generate the most accurate histograms using the neighboring histograms, which can be interpreted as that the interpolation tends to be a displacement interpolation instead of a stationary one (not a convex combination).

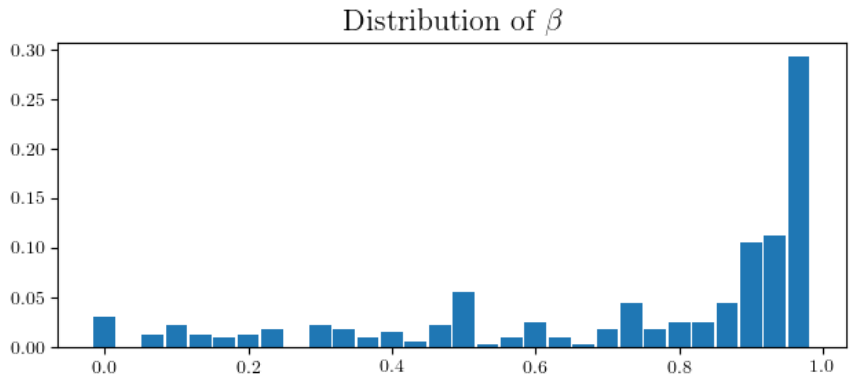


Figure 6.1. Distribution of β for different points with different alignments.

6.1.3. Fingerprint Sets

We test the observation density estimators on different subsets of fingerprint locations. Beginning with the whole set of 50 fingerprint locations, we run simulations with the estimation algorithms using the sets of 32, 21, 15 and 8 fingerprint locations. The configurations are shown in Figure 6.2.

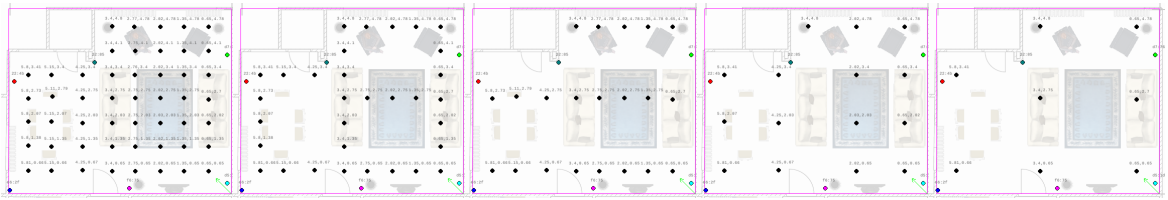


Figure 6.2. Distribution of fingerprint locations with different sizes. Black dots: measurement locations, colored dots: beacon locations.

As the complexity of the algorithms are also similar, we do not mention their complexities. Only ANN takes too long to train, while it passes an actual training procedure compared to the others. Moreover, we use a fine grid structured probabilistic

RF map, on which histograms for every beacon every grid center are previously computed, so that for the actual runs the likelihood estimation of the particles is merely a lookup table operation with the trade-off of high memory usage.

6.1.4. Radio Frequency Map Generation Performances

Before running the SMC experiments, we first compare the performance of the histogram combination methods in terms of their abilities to estimate the unknown fingerprints. We use the fingerprint configurations defined in the previous section as the training set. We estimate the full set of fingerprints (50 FLs) given this subset of fingerprints. At each iteration, from the full set of fingerprints, we first take out the specific fingerprint if it exists in the subset. Then the combination methods are employed to estimate a fingerprint (set of histograms) on the target position. The estimated histograms are compared with the originally collected histograms. We use the Wasserstein distance metric to evaluate the similarity of the histograms. We obtain 300 distance measures, because the methods are repeated for 50 fingerprints with 6 beacon histograms on each position. Each method is run using their best parameter sets.

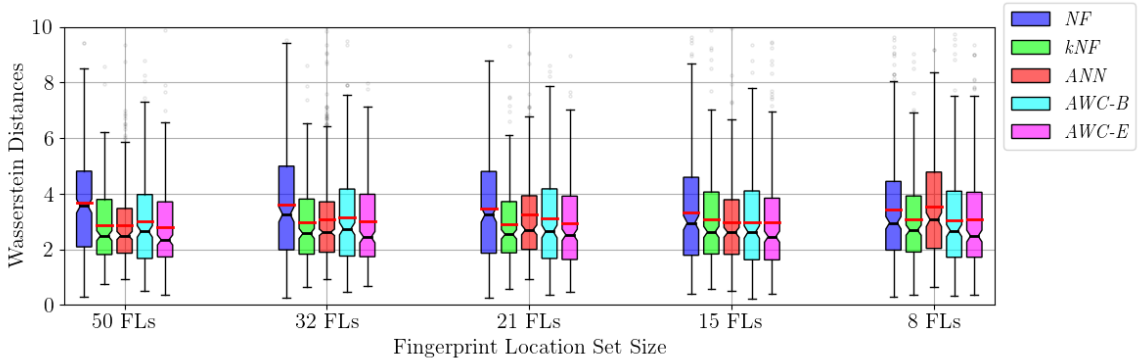


Figure 6.3. Comparison of the methods with respect to their radio frequency map generation performances. We measure the Wasserstein distances between the estimated fingerprints and the original fingerprints on the same positions.

In Table 6.1, we report the means of these distances, and Figure 6.3 shows the distance distributions as boxplots. We regard that the radio frequency map is gener-

ated better if the corresponding Wasserstein distance is lower. The results show that using a fingerprint (NF) instead of its neighboring fingerprints results with high estimation errors, but with convex combination of these fingerprints (κ-NF) improves the performance. We visualize the same phenomenon for the AWCmethods. With AWC-B, Using a surrounding pair to generate an interpolation yields comparable results with the previous methods. Smoothing the AWCmethod with multiple pairs, as AWC-E, we obtain significantly better performances for all of the fingerprint sets, and we conclude that the method AWC-Eestimates the radio frequency maps better than its counterparts.

Table 6.1. Medians of RF map estimation performances (Wasserstein distance).

Medians	50 FLs	32 FLs	21 FLs	15 FLs	8 FLs
NF	3.557	3.258	3.247	2.956	2.955
κ-NF	2.492	2.568	2.539	2.623	2.689
ANN	2.497	2.613	2.700	2.609	3.066
AWC-B	2.643	2.739	2.642	2.621	2.674
AWC-E	**2.338**	**2.440**	**2.520**	**2.456**	**2.479**

6.1.5. Test Trajectories and Observations

For testing purposes, we sample synthetic trajectories that imitate the movement of a person that navigates at 0.5 m/s with low tendency to rotate and receives beacon data similar to the BLE RSSI data generated in the setup area (see Figure 6.4). The details of the trajectory generation is given in APPENDIX A.1.

Observational data are sampled afterwards on the trajectory points. The details of the data generation is given in APPENDIX A.2.

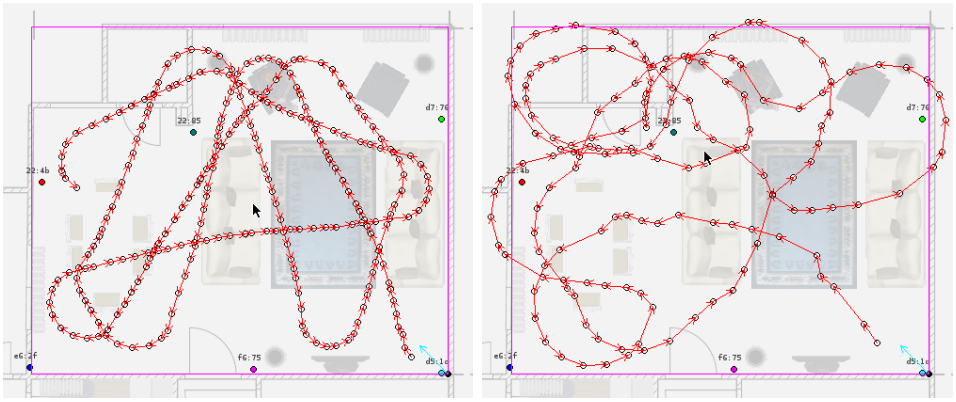


Figure 6.4. Sampled trajectories with different parameters.

6.1.6. Summary of Experiments

We supply a brief summary of the experiments for the readers (see Figure 6.5). (a) We first collect long hours of BLE RSSI data from densely selected points in the area. (b) We convert these data into fingerprints (normalized histograms) that can be queried for the likelihood of a RSSI value given 2D point and beacon ID. (c) With data from multiple fingerprint locations in hand, we search for the model parameters of the individual location estimators like κ in κ-NF, N_h in ANN or β and ρ in AWC based estimators. (d) We select a subset of fingerprint locations and (e) we then run the estimators with these parameters on a fine grid with points of 0.1 meters apart, in order to construct a grid of observation densities for different beacons, or namely an estimate of the probabilistic RF map of the area.

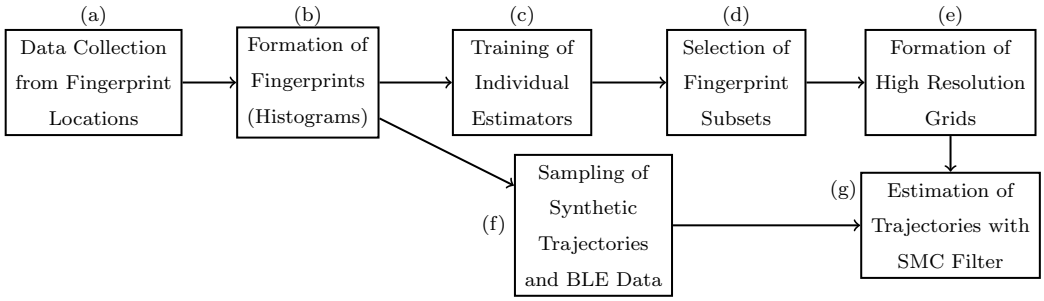


Figure 6.5. Block diagram of the experimental setup.

In a different track, (f) we sample a trajectory of a virtual agent that navigates in the test area and generate RSSI data from the previously collected dense fingerprint

locations. Finally, using the estimate of the probabilistic RF map as the observation density, sampled RSSI data on the trajectory points as the observations and the motion model as the prediction density, (g) we employ the SMC filter to estimate the position estimations for the observations and measure the tracking performance of the specific method.

6.1.7. Results

We defined five different estimator methods to estimate the observation densities: NF, κ-NF, ANN, AWC-B and AWC-E. The estimations are performed using 50, 32, 21, 15 and 8 measurement locations. The estimators estimate the observation densities for particle evaluation in the update step of the SMC algorithm. For the prediction step, we employ the motion model defined in Section 2.1, and the recursive loop of the SMC algorithm is closed. The simulations are designed with Python-3.4 and are run with a particle size of 10k on an Intel Xeon 2630. The particle evaluation step is parallelized by 32 processes. Each iteration takes about 0.8 seconds. A snapshot of the running SMC algorithm is given in Figure 6.6.

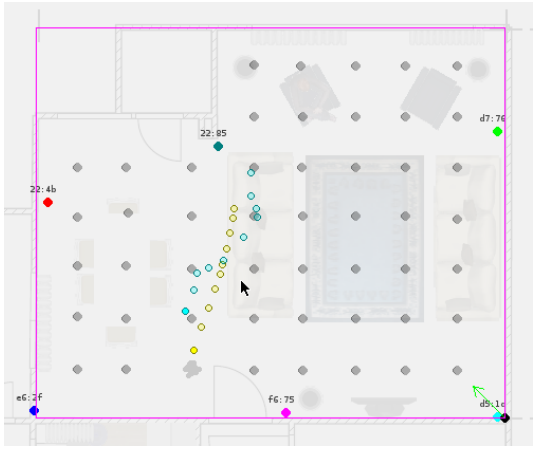


Figure 6.6. An SMC filter on the run: Yellow trajectory is the computer generated positions, cyan trajectory is an estimation.

The SMC filter was run for the same trajectory of 232 points, with 32 of them discarded as the burn-in period, leaving us with 200 points of error for each run. We

log the distances of the predicted positions to the original positions as errors. Each combination (estimator and fingerprint set size) is repeated for 30 times, which makes 6000 position estimations or error measurements per combination. The box plot of the statistics is given in Figure 6.7. We both report mean and median errors.

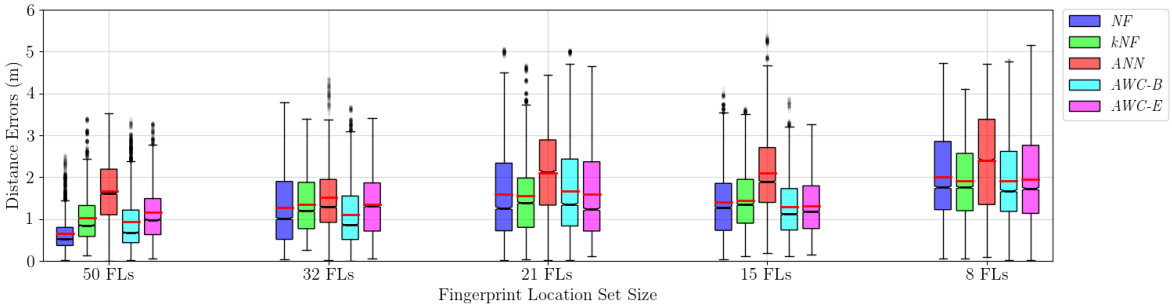


Figure 6.7. Trajectory estimation performances of different observation models with different sizes of fingerprint sets. The black lines point the medians and the red lines are for the means.

It can be seen from the results that high resolution fingerprint information (50 FLs) defines the map the best for the tracking purposes with any of the applied methods. Without employing any complicated method, that is, using only NF in estimating the observation density gives the best result: an error rate of 0.66 meters, with the lowest variance. This is an expected result but with an impractical setting. With more than 50 fingerprint locations, this error rate will surely get lower, but making the scene analysis much more a burden at the same time. Collecting data from many locations is obviously not practical. Reduced number of fingerprints would facilitate the installation and calibration procedure in BLE positioning and tracking infrastructure, so that the results for such scenarios are substantial and realistic for us.

As the configurations lose fingerprint locations, we see that NF is unable to keep its success with the highest resolution and Wasserstein interpolation based techniques stand out. Even though the error rates over 1m, especially AWC-B looks promising with and error rate of 1.29 m with 15 fingerprint locations. The same algorithm makes an error rate of 1.9 meters with only 8 fingerprint locations.

Neural network approach (ANN) is both inconvenient as we do not have many samples for training and impractical as the training iterations take too long. Moreover, the results show that ANN yields consistently higher error rates compared to the other methods. These results is expected whilst the neural networks require many samples beforehand, and we also try to reduce the required fingerprint size, which makes ANN naturally inappropriate for our purposes.

Table 6.2. Means of trajectory estimation errors (m).

Means	50 FLs	32 FLs	21 FLs	15 FLs	8 FLs
NF	**0.663**	1.266	1.602	1.414	2.012
κ-NF	1.033	1.359	**1.561**	1.445	**1.908**
ANN	1.661	1.515	2.098	2.094	2.395
AWC-B	0.932	**1.098**	1.662	**1.294**	1.910
AWC-E	1.157	1.355	1.588	1.314	1.956

Table 6.2 summarizes the mean localization errors for different fingerprint configurations with respect to the applied estimator algorithms. Concentrating on the results with lower number of fingerprints (15 and 8), we see that Wasserstein based methods race head to head with the κ-NF method. We also run two t-tests for significance analysis on the last two configurations. In the significance test tables we report the p-values for the one-sided hypothesis if the value on the row is less than the value on the column. According to the p-values, for 15 fingerprint locations (see Table 6.3), AWC-B has the best localization performance significantly, but for the configuration of 8 fingerprint locations (see Table 6.4), there is not a best estimator, because AWC-B and κ-NF cannot outperform each other.

Even though the box plots in Figure 6.7 give a hint about error distributions, we also supply the explicit distributions and corresponding cumulative error distributions. In Figure 6.8, we encode the methods with the same color codes with Figure 6.7.

Table 6.3. p-values of the pairwise one-sided t-tests of the results for the set of 15.

$<$	NF	κ-NF	ANN	AWC-B	AWC-E
NF	-	0.0000	0.0000	1.0000	1.0000
κ-NF	1.0000	-	0.0000	1.0000	1.0000
ANN	1.0000	1.0000	-	1.0000	1.0000
AWC-B	0.0000	0.0000	0.0000	-	0.0000
AWC-E	0.0000	0.0000	0.0000	1.0000	-

Table 6.4. p-values of the pairwise one-sided t-tests of the results for the set of 8.

$<$	NF	κ-NF	ANN	AWC-B	AWC-E
NF	-	1.0000	0.0000	1.0000	1.0000
κ-NF	0.0000	-	0.0000	1.0000	0.0000
ANN	1.0000	1.0000	-	1.0000	1.0000
AWC-B	0.0000	1.0000	0.0000	-	0.0000
AWC-E	0.0000	1.0000	0.0000	1.0000	-

The figures obviously show that the error distributions are not normally distributed, which is probably due to sudden incorrect far position estimations in the SMC filter. We believe that such results are better shown with box plots and medians, as the errors have skewed distributions, where means would be misleading. So we also report the error medians in Table 6.5.

Table 6.5. Medians of trajectory estimation performances.

Medians	50 FLs	32 FLs	21 FLs	15 FLs	8 FLs
NF	**0.519**	1.008	1.259	1.267	1.754
κ-NF	0.847	1.194	1.385	1.342	1.753
ANN	1.607	1.289	2.127	1.889	2.403
AWC-B	0.676	**0.859**	1.349	**1.118**	**1.672**
AWC-E	0.984	1.309	**1.232**	1.174	1.724

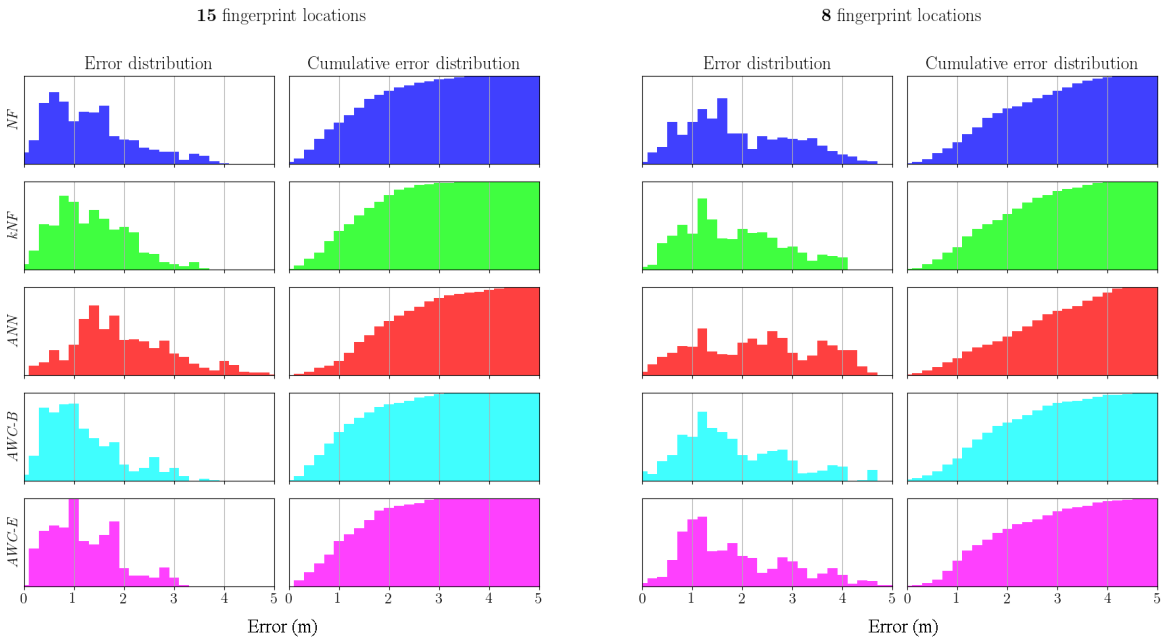


Figure 6.8. Localization error distributions and corresponding cumulative error distributions for two fingerprint location configurations: 15 and 8 locations.

For the significance test, because the error results are not necessarily normally distributed, we apply one-sided Wilcoxon signed-rank tests on the error rates of the configuration pairs. We report the two low sized fingerprint sets' values with the confidence value of $\alpha < 0.05$.

Table 6.6. p-values of one-sided pairwise Wilcoxon tests for the set of 15.

$<$	NF	κ-NF	ANN	AWC-B	AWC-E
NF	-	0.0000	0.0000	1.0000	0.9998
κ-NF	1.0000	-	0.0000	1.0000	1.0000
ANN	1.0000	1.0000	-	1.0000	1.0000
AWC-B	0.0000	0.0000	0.0000	-	0.0001
AWC-E	0.0002	0.0000	0.0000	0.9999	-

Table 6.6 shows that for the configuration of 15 fingerprint locations, the methods proposed, AWC-B and AWC-E, perform significantly better than any other method we compared. For a smaller configuration of 8 locations, we almost see the same results. Table 6.7 shows that the proposed methods perform significantly better than the ap-

Table 6.7. p-values of one-sided pairwise Wilcoxon tests for the set of 8.

$<$	NF	κ-NF	ANN	AWC-B	AWC-E
NF	-	1.0000	0.0000	1.0000	1.0000
κ-NF	0.0000	-	0.0000	0.9014	0.5156
ANN	1.0000	1.0000	-	1.0000	1.0000
AWC-B	0.0000	0.0986	0.0000	-	0.1205
AWC-E	0.0000	0.4844	0.0000	0.8795	-

plied techniques, except against κ-NF, in which we can say that our methods perform slightly better than κ-NF. This leads to the claim that, for the smalls sized fingerprint sets (8 and 15), Wasserstein-based interpolation techniques reduce the errors significantly, except for only one case, compared to the applied techniques so far. Moreover, as the complexity of the methods are similar, Wasserstein interpolation techniques are preferable for small sized fingerprint sets. Amongst the proposed methods, AWC-B is seen to perform slightly better than its counterpart AWC-E.

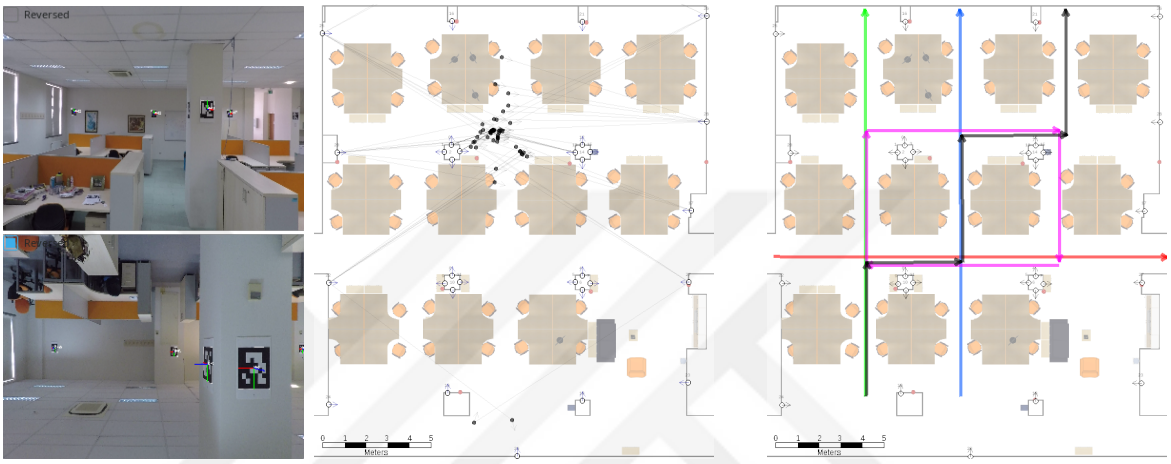
Thus, we conclude that that two methods AWC-B and AWC-E are the two candidates to be used in observation density estimation in the SMC filter for tracking purposes with BLE beacon information for lower number of fingerprints if the positions of these fingerprints are scattered evenly in order to perform better histogram interpolations.

6.2. Experiments for High Precision Data Annotation

We conduct the experiments for collecting high resolution and high precision position data using the AR markers stationed in the area. The experiments are evaluated using both time difference and distance to the predetermined paths.

6.2.1. Data collection experiments

We collect two types of data at the same time: timestamped RSSI data emitted by a BLE beacon sensed by BLE receivers, and position estimations through the videos captured by the two camera system (see Figure 6.9(a)).



(a) Detected markers in camera views (left) and corresponding raw pose estimations (right).

(b) Predetermined paths of the experiments.

Figure 6.9. Position data collection and experiments.

We prepare five experimental paths for testing the performance of our methodology: three distinct linear paths, a rectangular path and a zigzag one. The camera setup is driven along these paths trying to respect a linear and smooth motion, simulating human movement. The predetermined paths are also given in Figure 6.9(b). We repeat the experiments with our equipment in the area with different values of the parameters for the elimination strategies, the closest selection, S, and the outlier elimination, U; and with the observation covariance matrix of Kalman filter. We predefine the transition covariance matrix as $\mathbf{Q} = 0.01\mathbf{Id}$, which corresponds to the case that the pose at the next time step should be close to the current one.

6.2.2. Results

Visualizing the raw estimations, we hypothesize that the pose estimations display two types of faults: (i) they may be positioned very far from the true position because

of incorrect marker axis estimations (e.g. two points in the lower part of the map in Figure 6.9(a)), or (ii) they may fluctuate around a true point because of pixelwise problems (e.g. the ones around the dominant cluster). We believe that these types of errors can be controlled first by discarding the estimations via the detections of the farthest markers, and then by making an outlier based pruning with the rest.

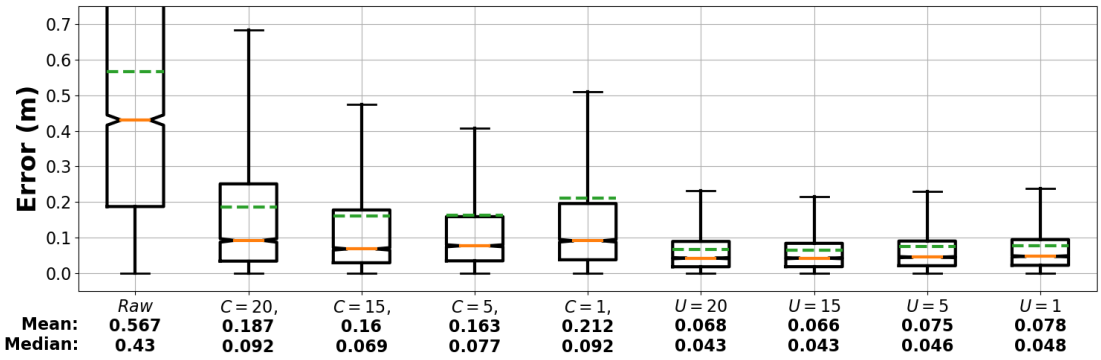


Figure 6.10. Comparison of isolated strategies.

In the figures, we report the deviations from the ideal paths. The isolated versions of these two elimination strategies are compared with the errors of raw estimations (without elimination). According to Figure 6.10, it is obvious that an elimination method is mandatory, and between these isolated elimination strategies, the outlier elimination method with $U = 15$ performs better than others, giving an error median of 0.043 meters. Moreover, the error distributions for the selection method display undesired skewed distributions with high numbers of outliers.

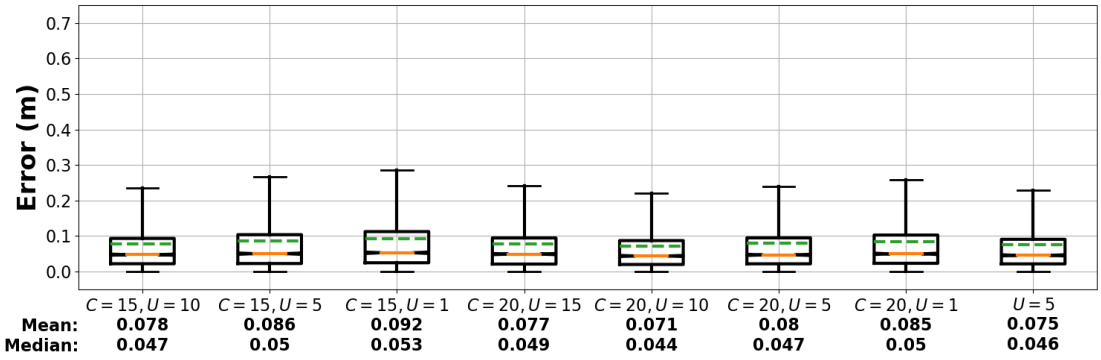


Figure 6.11. Comparison of combinations of selection and outlier detection strategies.

We select two of these experimental methods: (i) the isolated method with $U = 15$, and (ii) a combination of a selection method with $S = 20$ and a following outlier elimination with $U = 10$ would work best among the experiments performed (see Figure 6.11). However, at each step of the greedy outlier elimination strategy, a new mean calculation is required. This calculation makes the strategy computationally intensive with respect to the selection strategy, so it would be intuitive to precede it with the quicker close marker selection method, since the far markers usually yield misleading estimations. Moreover, since there is not a significantly best one, we prefer the latter one ($S = 20$, $U = 10$) of the selected methods.

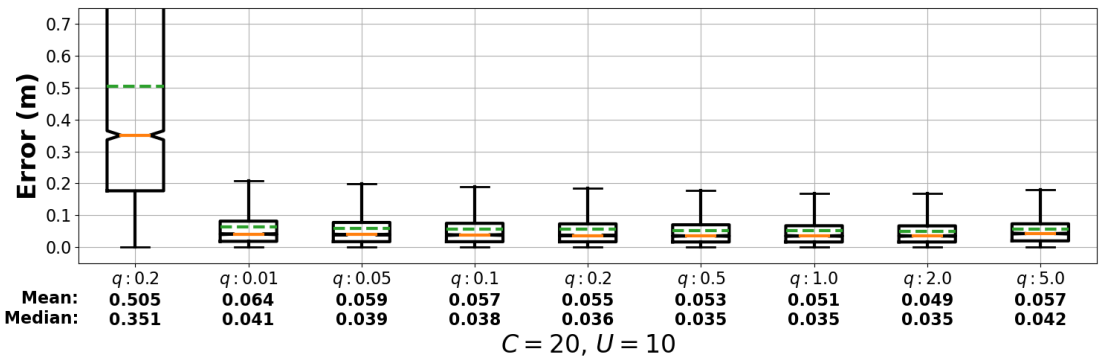


Figure 6.12. Comparison of different noise covariance performances of the Kalman filter after preprocessing with selection and outlier detection strategies.

On the selected position estimations after the elimination techniques, we employ a Kalman filter to attain smoother and more accurate trajectories. Figure 6.12 summarizes the parameters for the covariances and the errors, after employing the Kalman filter on the preprocessed position estimations. We decide that the deviation medians can be reduced down to 0.035 meters and the means under 0.05 meters with a covariance matrix of $\mathbf{R} = 2.0\mathbf{Id}$.

Applying the Kalman filter does not mean that we achieve more accurate trajectories. Up to this point, we measure and compare the deviations of the estimations from the predefined ideal paths. Smoother paths yield lower deviations, however, when

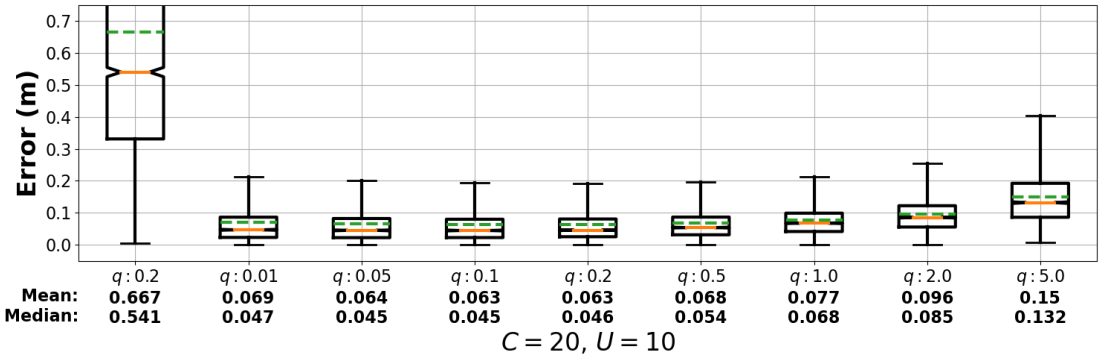


Figure 6.13. Comparison of different noise covariance performances and lag penalty application.

the Kalman filter is applied on the position estimations, dragging is also possible if the noise covariance of the Kalman filter is chosen to be too high. If this covariance is excessively high the estimated trajectory will surely be smooth, yielding very low deviances, but failing to estimate the real positions in the forward direction. This dragging can be evaluated by measuring the distance between a real known end point of a segment and the estimated points at the same timestamp. To avoid high dragging situations, we penalize the experiment by accumulating these drag distances as errors on the deviations, and we obtain Figure 6.13. The parameters with $q > 0.2$ are highly penalized, meaning that estimated paths do not lead to accurate localization, and with $q < 0.1$, less smoothing is applied. We conclude that choosing $q \in [0.1, 0.2]$ for these experiments results with the best performances.

We consider a final situation where Kalman parameters are also affected by the speed of the navigation. We also test the same experiment at three different paces. The results in Figure 6.14 show that choosing $q = 0.2$ and $q = 0.1$ for slow navigations (at about 0.3 m/s) and for quicker navigations (at about 0.4 m/s) respectively will yield better performances.

One may propose to directly filter the raw position estimations without applying any elimination method (see Figure 6.15(b)), however, as the raw estimations are highly erroneous because of mainly misdetection of marker orientations, the Kalman

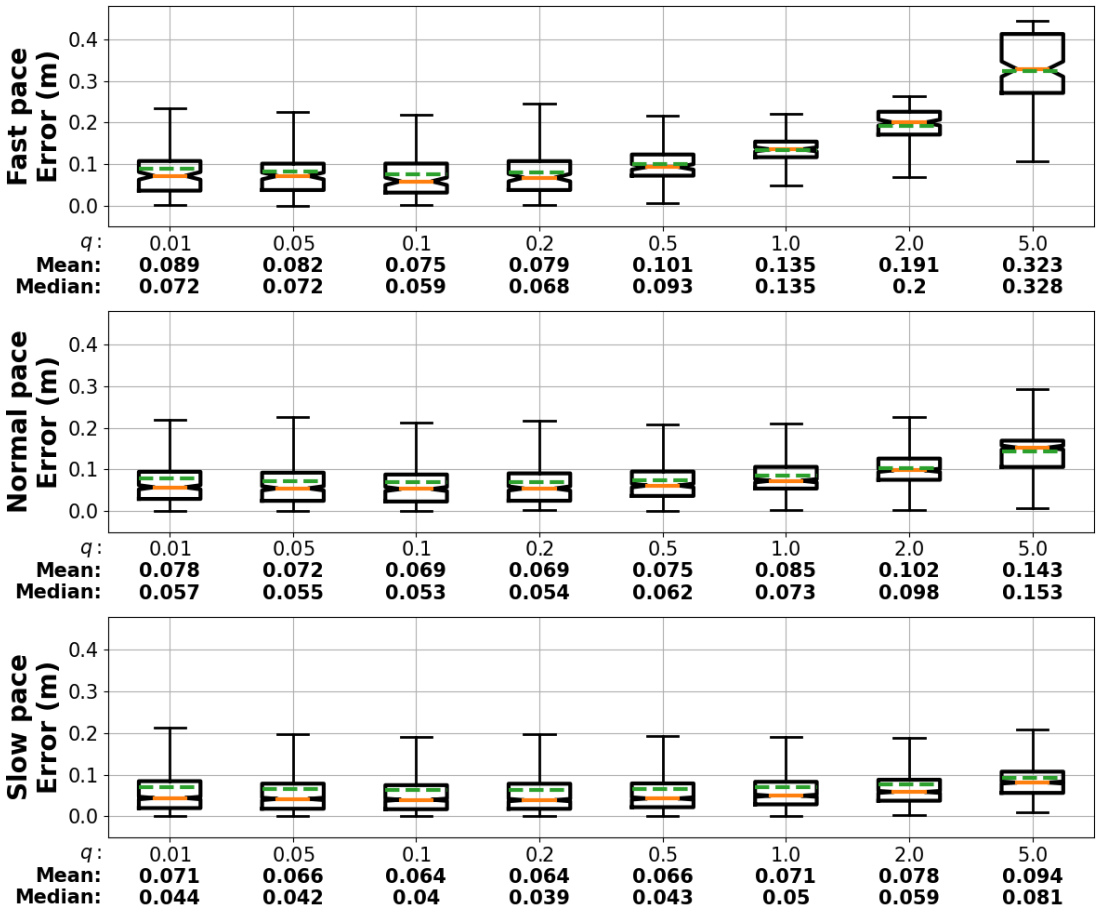
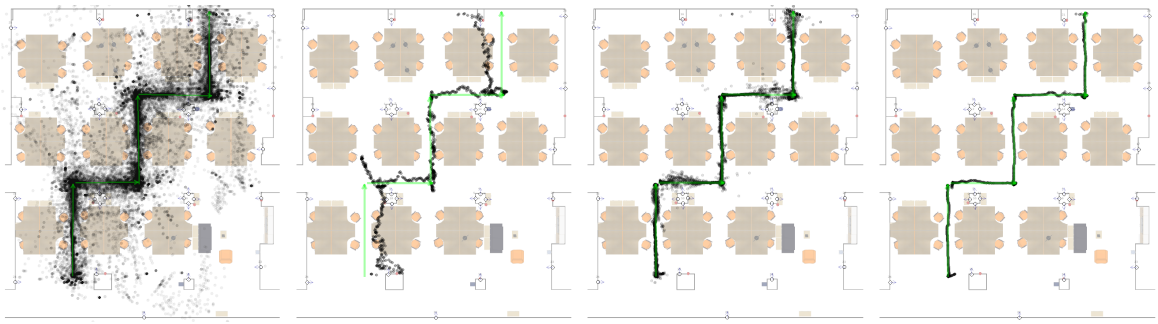


Figure 6.14. Error comparisons with respect to varying paces.

filter is unable to cope with raw estimations. An elimination method is mandatory. The positions after the elimination strategies still draw fluctuating trajectories as seen in Figure 6.15(c), but the selected position estimations are distributed around the true paths. These fluctuations may be eliminated with online filtering mechanisms. Figure 6.15(d) is the result of first applying the aforementioned elimination strategies on the positions estimated by the ArUco marker detection algorithm, and then smoothing the intermediate results using the Kalman filter with proper noise covariance parameters. We see that green predetermined path can also be visually matched with the filtered positions.



(a) (b) (c) (d)

Figure 6.15. Experiments with different strategies: black dots are the estimated positions: (a) raw position estimations, (b) estimations after only the Kalman filter, (c) only elimination strategies, (d) the Kalman filter preceded by elimination strategies.

6.3. Experiments for the Adaptive Sequential Monte Carlo Filter

The experiments are performed on an Intel Xeon E5-2697A running at 2.60GHz. For the experiments, we run the SMC filters of different models with different configurations. A sample visualization of a running SMC filter is given in Figure 6.16.

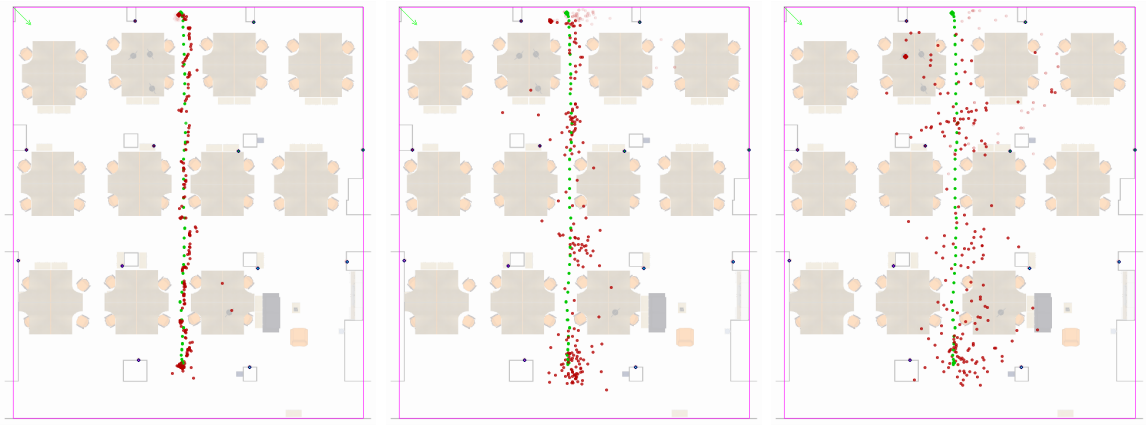


Figure 6.16. Visualizations of SMC filter experiment samples with different diffusion factors. Green dots are the ground truth positions, red dots are the estimated positions.

To statistically assess the results, each configuration is run for 50 times in parallel and repeated for different trajectories. In the results, we occasionally use K as

an abbreviation for the multiples of 1000 (e.g. 1K = 1000). We vary the following parameters in the following sets for different models:

- Static diffusion model
  - static diffusion factor: $k \in [0.001, 2.0]$
  - particle population size: $N \in \{1\text{K}, 2\text{K}, 5\text{K}, 10\text{K}\}$
- Decaying diffusion model
  - decaying diffusion factor: $\eta \in \{0.9, 0.95, 0, 99\}$
  - diffusion factor higher limit: $k_{max} = 5.0$
  - diffusion factor lower limit: $k_{min} \in [0.01, 2.0]$
  - particle population size: $N = 1K$
- Adaptive diffusion model
  - initial diffusion particles: $k_0 \in [0.00001, 5.0]$
  - sensitivity: $\nu \in [10^{-8}, 10^{-2}]$
  - particle population sizes: $(D, P) \in \{(500, 1),$
    $(333, 2), (250, 3), (200, 4), (100, 9), (50, 19),$
    $(40, 14), (20, 49), (10, 99), (5, 199), (4, 249),$
    $(2, 499), (1, 999)\}$

We evaluate each experiment by the accuracy of the estimated trajectory. We measure this accuracy by the mean and median of the Euclidean distances between the weighted position means, or estimations, and their corresponding ground truth positions. In Figure 6.17, we show a sample of trajectory errors throughout an experiment, the distribution and cumulative distribution of these errors. The very first estimations of a trajectory yield high errors naturally, and as more data are fed, the estimations tend to give lower errors. We rather rely on the medians of errors than the means, because the errors obtained after an experiment form right skewed distributions. This skewness is mostly due to the temporary far divergences of the particle clouds from the true positions.

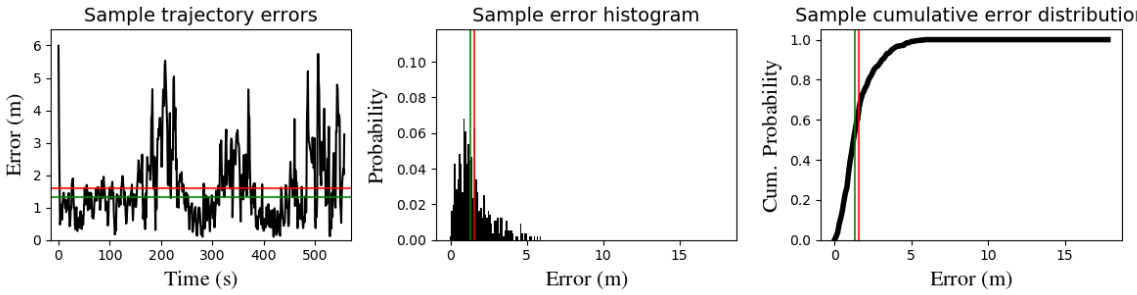


Figure 6.17. Sample error distributions. From left to right: Trajectory errors throughout an experiment, histogram of the corresponding errors, and cumulative distribution of the errors. The green and red lines indicate the median and mean of the distribution respectively.

We perform three types of main experiments: The first set of experiments aims to find the best strategy along with the most appropriate parameters for the SMC algorithm on synthetically and very accurately sampled trajectories. The first set is also for setting a proof of concept for the proposed methods and determining the parameters that are to be used in the experiments that follow. For the second set of experiments, we generate noisy RSSI data using closest fingerprint information. This second set is used for finding the generalized parameters for the different models of SMC filters. The third set of experiments evaluate the performance of the SMC filter on the trajectories with real RSSI measurements collected by navigating the test setup in the environment. Narrowing down the parameter set in the first and second, we compare and focus on the model parameters that can be used in real world conditions where the data is degenerated due to various factors.

6.3.1. Particle Population Size Selection

According to the nature of the SMC algorithms, with larger particle population sizes the algorithms approximate better, but higher particle sizes do not suit to the real world conditions because of high processing power requirements. Before elaborating

the error comparisons, we first select a reasonable particle population size by comparing the effect on the experiment run times with the frequency of the captured.

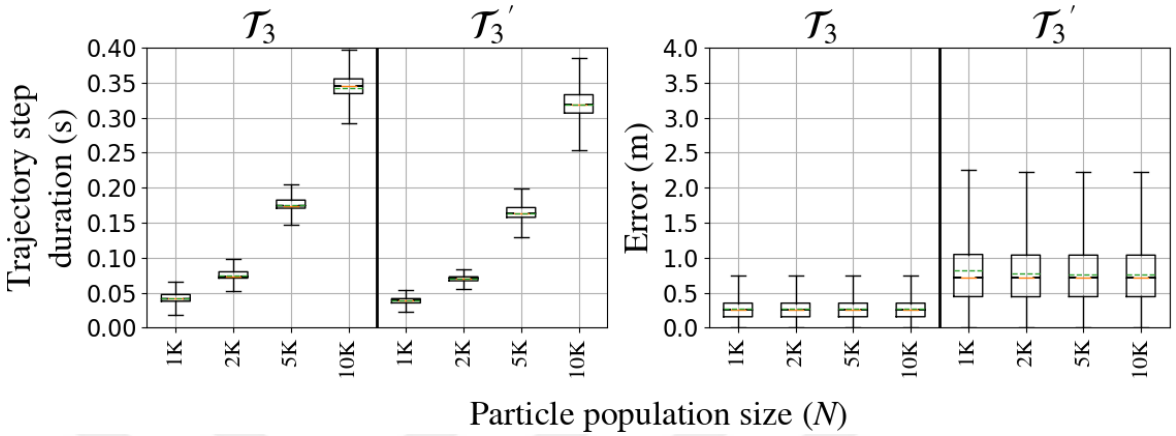


Figure 6.18. Duration distributions (left) and corresponding error distributions (right) with respect to different particle population sizes.

In Figure 6.18, we show the step processing durations throughout the experiments and corresponding trajectory error distributions. For the experimental trajectories $\mathcal{T}_3$ and $\mathcal{T}_3'$, we set safe static diffusion factors of 0.2 and 0.3 respectively. Safe parameters tend to explore sufficiently and does not get stalled. We see that varying the population size from 1K to 10K does not significantly reduce the error for the specific experiments. Hence, we choose a safe particle population size of 1K particles, which allows the system to process about 25 data points per second. With the current setup, if the processing were to be performed in real time, such a configuration would rarely miss any captured data.

Since there exist two layers that sample particles in the adaptive strategy, a comparable particle population size can be computed with $N_{ada} = D + DP$, because after D diffusion particles are sampled, for each sampled diffusion particle P position particles are sampled. The following sections use compatible population sizes of diffusion particles and position particles.

6.3.2. Resampling Strategy Selection

Resampling strategy is also very crucial in SMC. We compare three traditional resampling strategies: Multinomial, systematic and stratified. Whereas multinomial resampling strategy is known to approximate better as it simulates the real particle weight distribution, it has a higher computational complexity compared to its counterparts. We repeat the experiments for different resampling strategies with the same static diffusion factors, 0.2 and 0.3, for the trajectories, $\mathcal{T}_3$ and $\mathcal{T}_3'$. In Figure 6.19, we compare three traditional resampling techniques with respect to their running times and error distributions.

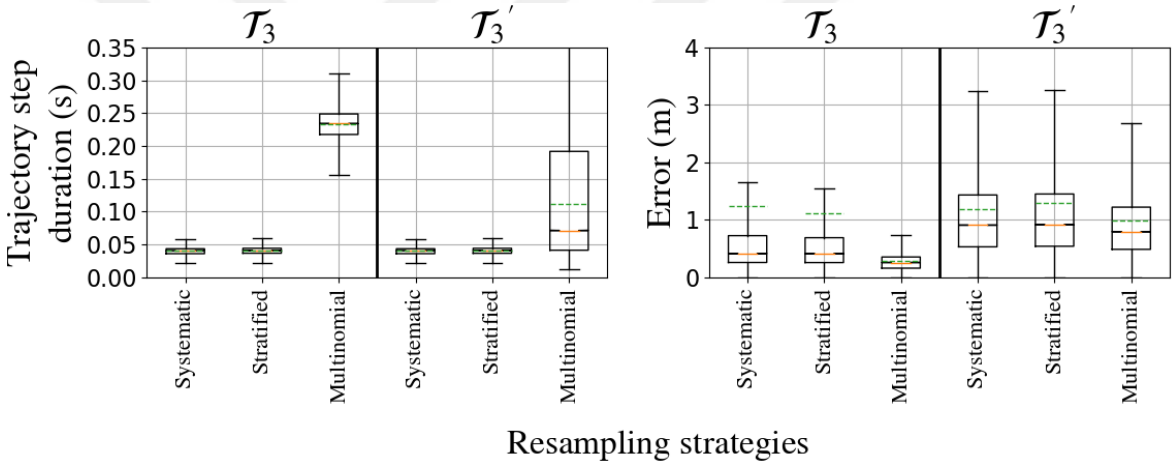
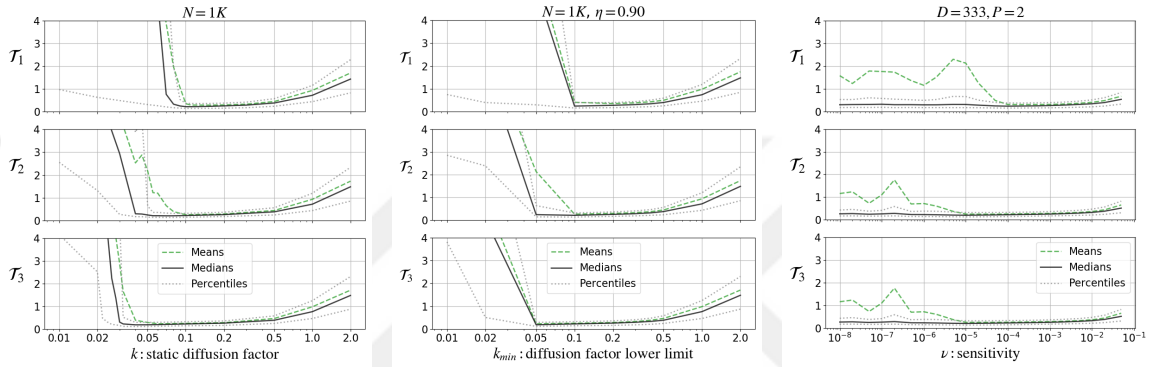


Figure 6.19. Duration distributions (left) and corresponding error distributions (right) with respect to different resampling strategies sizes.

The results show that varying the resampling strategy does not significantly improve the error statistics, but running times differ considerably. Employing a multinomial resampling strategy boosts the running times, however, corresponding error distributions have lower variances. Nonetheless, other strategies are preferable for their lower running times. There is no significant difference between stratified and systematic resampling strategies. Because the systematic resampling is also known to be the best with respect to the computational complexity and random numbers used [94], we prefer it in the following experiments.

6.3.3. Experiment I: Accurately Synthesized RSSI Data

For an exhaustive parameter search, we regenerate synthetic RSSI data on the collected trajectory positions. We aim both to show a proof of concept of an efficient adaptive parameter sampling method for the SMC filter in BLE based indoor localization, and find the best filter parameters for the given trajectories.



(a) Static diffusion (b) Decaying diffusion (c) Adaptive diffusion

Figure 6.20. Accurately synthesized data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively.

Among the three different diffusion factor strategies, the static diffusion factor performs better than others as expectedly if an optimal diffusion factor is given beforehand. We report the error means and medians with respect to both different static diffusion factors, keeping the population size at 1K. As shown in Figure 6.20(a), we achieve the lowest medians around the static diffusion factors of 0.1, 0.07 and 0.04, compatible with the paces of the trajectories. Moreover, the trajectory estimates are very accurate (best errors around 0.22 meters) and thus the method is very appropriate as the RSSI data are synthesized using the radio frequency map grids of size 0.2 meters.

These results show that in order to achieve accurate estimations, we should set the correct values for the static diffusion factors before the actual filter is run, but these correct values cannot be known beforehand as velocities or observation noises are unknown, or moreover the velocity can vary throughout an experiment. For the paces

comparable with the walk pace of a person, setting a value between 0.1 and 0.5 would be the best for the static diffusion factor, but for the values just under the best factor, we risk of falling into a stalled state where the estimations cannot keep up with the actual trajectory, because of the lack of exploration tendency. This strategy shows that a high static diffusion factor should be chosen to cope with the varying and unknown velocities, however, we see that for higher values the SMC filter loses the focus and tends to explore with largely scattered particle clouds.

Selecting a decaying diffusion factor that behaves exploratorily in the beginning and more exploitatively through the end, shows that we reach a more generalized system for some of the experiments, but we still require to select a covariance factor lower limit where the decaying should not surpass. With higher decay rates we attain more generalized error distributions, but with lower decay rates lower errors are obtained. Setting appropriate lower limits makes the system no different than selecting static diffusion factors (see Figure 6.20(b)).

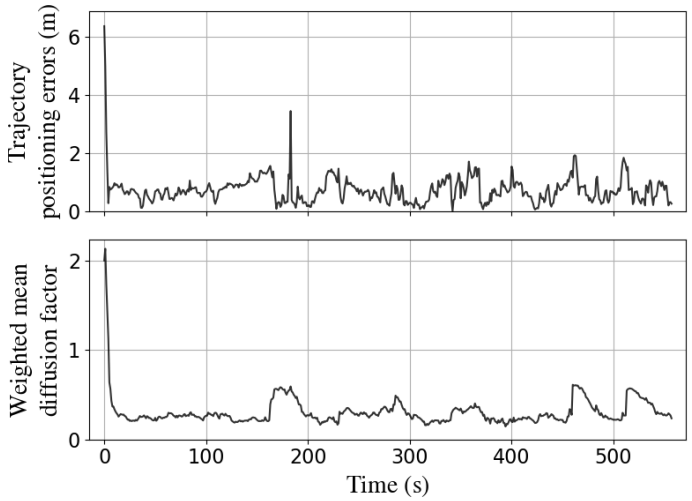


Figure 6.21. Adaptive behavior of a sampled diffusion factor. The weighted average of the diffusion factor accompanies the error fluctuations of the position estimates.

The proposed approach, the adaptive diffusion factor, however, yields more generalized results by avoiding falling into a stalled state. Sampling the diffusion factor helps the system maintain a generalized behavior, that is, when the error gets higher

exploration is favored, and exploitative behavior otherwise. Figure 6.21 shows a gist of this behavior on a sample run. The temporary peaks of the weighted mean diffusion factor can be seen when the positioning errors are boosted. The diffusion factor adapts itself to the changing conditions (i.e. loss of particle focus).

The diffusion factor is not set too low or too high, both of which would result with high positioning errors. In Figure 6.20(c), the error mean and medians are plotted against different sensitivity values. We see that even varying the sensitivity dramatically does not boost the errors like in the static or decaying cases. The medians of the errors tend to stay around a constant value, which is to say that we have a more generalized algorithm that can tune its diffusion factor, k, according to the requirements. Selecting a sensitivity value about 2×10^{-3} is appropriate for our algorithm to achieve low, but more importantly consistent error distributions.

For each set of experiments we also show the lowest errors achieved and the corresponding parameters. For the synthetic data, these statistics are given in Table 6.8. According to the tested parameters, the static diffusion strategy performs the best. This is an expected behavior, since we note that these experiments have constant velocity. Besides, an adaptive diffusion strategy shows comparable performances with the two other strategies.

6.3.4. Varying the Diffusion Particle Population Size

Figure 6.22 summarizes the error distributions with respect to the sensitivity. We see that by distributing the particle sizes to the layers, we reach lower error rates for specific sensitivity values ($D = 200, P = 49$), but narrower thus highly stable plots are obtained with increased diffusion particle size ($D = 500, P = 1$). We set the total sampled particle size to 1K, but in order to decide how many samplings each layer should perform, we examine the error distributions with varying diffusion and position particle sizes.

Table 6.8. Best error statistics (m) for accurately and noisily synthesized RSSI measurements on different trajectories.

		Static	Decaying	Adaptive
		$N = 1K$	$N = 1K$	$D = 333$ $P = 2$
$\mathcal{T}_1$	Par.	$k^* = 0.1$	$k_{min} = 0.2$ $\eta = 0.90$	$\nu = 1.0 \times 10^{-4}$
	Median	**0.227**	0.239	0.235
	Mean	0.356	0.349	**0.312**
$\mathcal{T}_2$	Par.	$k^* = 0.07$	$k_{min} = 0.1$ $\eta = 0.90$	$\nu = 2.0 \times 10^{-5}$
	Median	0.202	0.217	**0.198**
	Mean	0.248	0.267	**0.247**
$\mathcal{T}_3$	Par.	$k^* = 0.04$	$k_{min} = 0.05$ $\eta = 0.90$	$\nu = 1.0 \times 10^{-5}$
	Median	**0.183**	0.185	0.187
	Mean	**0.222**	0.230	0.229
$\mathcal{T}_1'$	Par.	$k^* = 0.4$	$k_{min} = 0.5$ $\eta = 0.90$	$\nu = 2.0 \times 10^{-4}$
	Median	0.748	0.777	**0.675**
	Mean	**0.780**	0.836	0.803
$\mathcal{T}_2'$	Par.	$k^* = 0.3$	$k_{min} = 0.2$ $\eta = 0.90$	$\nu = 5.0 \times 10^{-4}$
	Median	0.671	0.675	**0.654**
	Mean	0.802	0.830	**0.734**
$\mathcal{T}_3'$	Par.	$k^* = 0.03$	$k_{min} = 0.2$ $\eta = 0.90$	$\nu = 1.0 \times 10^{-4}$
	Median	0.717	0.737	**0.688**
	Mean	0.812	0.866	**0.763**

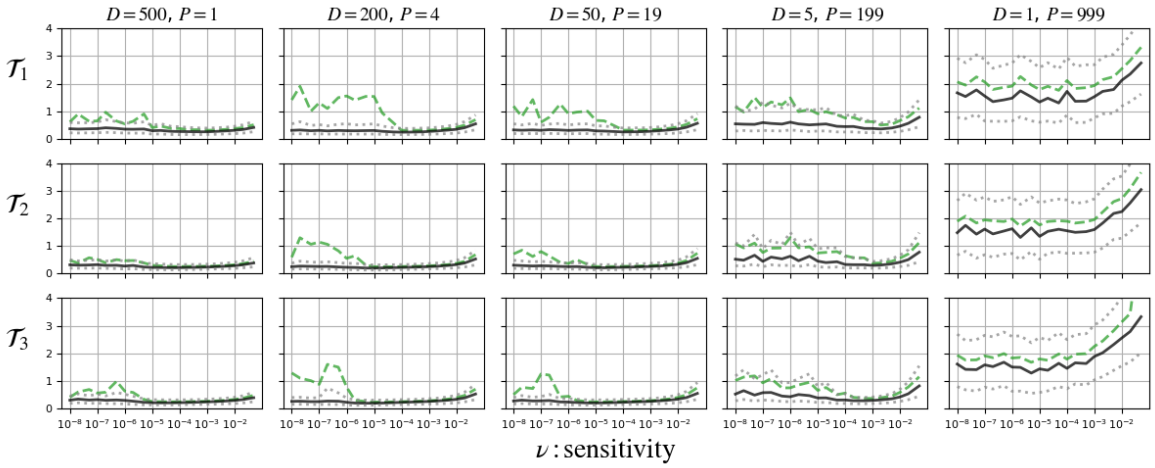


Figure 6.22. Error distributions for different sensitivity values and different particle size configurations.

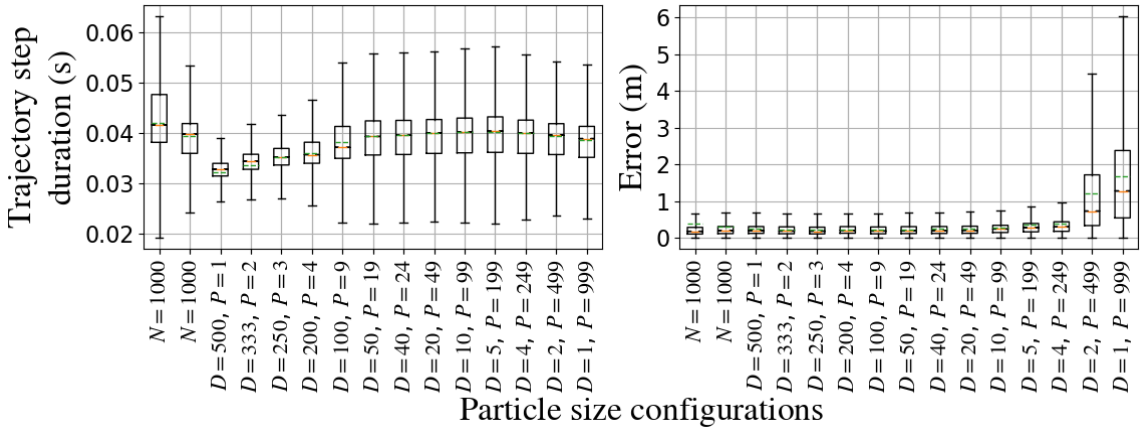


Figure 6.23. Duration (left) and error rate (right) comparison of the different adaptive strategy configurations with the static and decaying models, the first and second distributions respectively from the left.

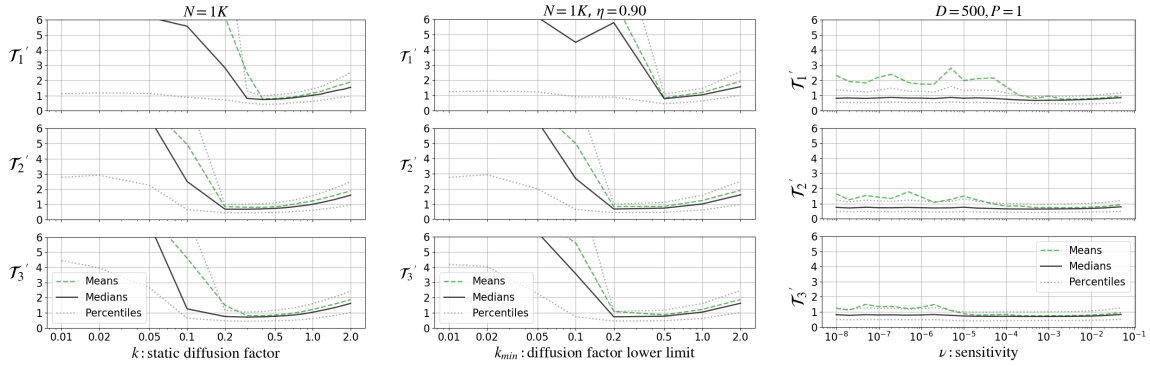
To determine an appropriate particle size tuple, we look finally at the timings of each option. Figure 6.23 compares the step durations and errors of static and decaying diffusion models at 1K with the sampling models of various configurations. The adaptive model with a diffusion particle size of 500 has the lowest run times with error rates comparable to the opposing configurations. This is due to that sampling a position particle is more time consuming than sampling a diffusion particle. The parameters $(D, P) \in \{(500, 1), (333, 2), (250, 3)\}$ for the adaptive strategy are favorable,

because they have lower run times and highly stable error rates.

6.3.5. Experiment II: Noisily Synthesized RSSI Data

The measurement model of the first set of experiments is designed to be very accurate, and does not project to real life conditions. For the second set of experiments, we disrupt the generation model to synthesize highly noisy RSSI data for the trajectories. We mark the identifiers of the trajectories with the primed versions of the same trajectories in the first set. ($\mathcal{T}_1'$, $\mathcal{T}_2'$ and $\mathcal{T}_3'$). The aim of this set is to show whether the findings after the first set is still valid for a more generic setup.

These data are synthesized using the fingerprints. The RSSI data point on a trajectory position is resampled upon the closest corresponding fingerprint histogram, which makes the generated RSSI data highly noisy with respect to the previous data set. Considering the gaps between the fingerprint positions, estimation errors up to 1 m is acceptable.



(a) Static diffusion (b) Decaying diffusion (c) Adaptive diffusion

Figure 6.24. Noisily synthesized RSSI data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively.

Figure 6.24 displays a similar behavior for the strategies, but with higher error rates. For the static and decaying diffusion, we obtain U-shaped graphics, left of which points the space of stall, and the right shows the space of exploratory behavior. The

space of the adaptive strategy errors is far from a U-shape. Though, if the search space is enlarged, we surely experience some deteriorated error distributions, but with a large enough search space, we obtain stable error distributions for the adaptive diffusion strategy. This is interpreted as the diffusion rate is automatically tuned for better estimations wherever it was initialized.

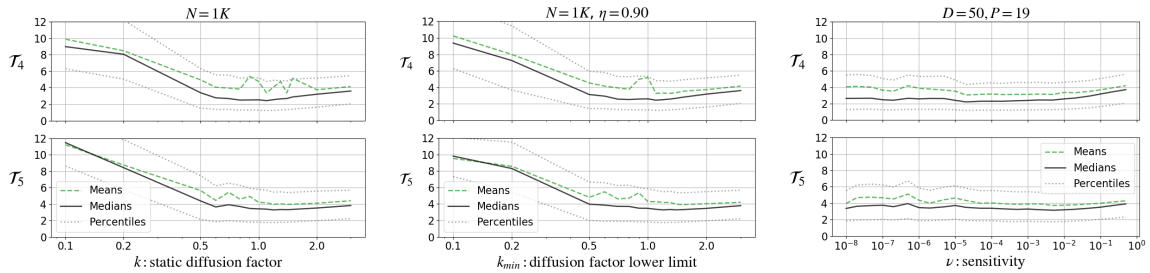
The adaptive strategy performs also better when the best parameters are compared. We observe in Table 6.8 that for all of the experiments, the adaptive strategy reaches the lowest error medians and means. We conclude that, if the measurements are noisy, an adaptive strategy based on parameter estimation by sampling leads to the better estimations.

6.3.6. Experiment III: Real RSSI Data

For the real data, we use two different trajectories whose RSSI measurements are captured with the navigable platform detailed in Section 5.2. The RSSI data are collected simultaneously with the ground truth positions. The position estimations by the SMC filters are evaluated with these ground truth positions.

In these experiments, the nature of the RF signal and the lack of dead reckoning information prevent the system from making accurate estimations. Different from the first two sets of experiments, here the emission density is very noisy and the navigation velocity is variable.

Even with the handicapped conditions, Figure 6.25 shows that the adaptive diffusion strategy keeps its steadiness for a large range of sensitivity values. We remark also that, for the real data set, we choose particle parameter sizes as $(D, P) = (50, 19)$, that is, at each step, we sample 50 particles for diffusion factor and 19 position particles for each diffusion factor. Even though the best parameters are found empirically, we can generalize that choosing a larger size for the diffusion particle population produces more accurate estimations (see Section 6.3.4).



(a) Static diffusion (b) Decaying diffusion (c) Adaptive diffusion

Figure 6.25. Real RSSI data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively.

Table 6.9. Best error statistics (m) for real RSSI measurement experiments.

		Static	Decaying	Adaptive
		$N = 1K$	$N = 1K$	$D = 50$ $P = 19$
$\mathcal{T}_4$	Par.	$k^* = 1.1$	$k_{min} = 1.1$ $\eta = 0.90$	$\nu = 5.0 \times 10^{-4}$
	Median	2.434	2.45	**2.223**
	Mean	3.362	3.284	**3.064**
$\mathcal{T}_5$	Par.	$k^* = 1.2$	$k_{min} = 1.4$ $\eta = 0.95$	$\nu = 5.0 \times 10^{-3}$
	Median	3.288	3.253	**3.148**
	Mean	3.953	3.867	**3.73**

We compare finally the best parameter configurations for each strategy. The adaptive diffusion strategy outperforms the other strategies with respect to stability and error performance. Figure 6.25 shows that the static and decaying strategies are prone naturally to fall into a stalled state if the diffusion parameters are selected to be too low, but the adaptive diffusion strategy does not get stalled for a comparatively large space of sensitivity values. Moreover, Table 6.9 shows that the adaptive diffusion strategy has the best median and mean error rates.

We report the best sets of parameters that are found empirically. It can be seen from the results that the best set of parameters may vary even for different trajectories in the same area. One may think that for only the specific parameters the strategies perform better, but the performance of the algorithm is similar for a large range of its sensitivity values. We can generalize that selecting an arbitrary sensitivity value from the ranges will perform acceptably without ever knowing these empirically best values.

6.3.7. Discussion

We compare our best error statistics that use real RSSI measurements with the state of art results that use mobile transmitters and stationary sensors in indoor experimental areas with similar sizes:

The work of Cantón Paterna *et al.* [67], is installed in an area of 270 m^2. They achieve an error rate of 4.6 m during 90% of the time using 4 computers and one beacon. They measure the performance of their work using interpolated points with respect to timestamps on predefined paths.

We do another comparison with a livestock tracking system [10]. The tests are conducted in a zone that houses 31 dairy Holstein cows with a size of 390 m^2. BLE transmitters are attached to the collars of the cows and RSSI data are captured using battery powered receivers. Moreover, they verify their results using a closed-circuit television (CCTV) system with three cameras. They measure an error median and mean of 3.3 m and 4.3 m respectively.

In terms of positioning performance, our error rate in the worst case scenario (3.148 m) is lower than the very similar setups. We observe that some of the experiments yield higher error rates naturally, but we deduce that these samples are not numerous, considering the low variances of the adaptive strategies (see Figure 6.25(c)). Moreover, we use a set of RSSI data annotated with its ground truth position using an AR based truth detection system, which makes the evaluation of our work more

accurate compared to its counterparts.

The only downside of our work is the prior scene analysis stage based on fingerprinting. The performance depends on the number of reference points and durations, which makes this stage tedious. However, we also bring a solution to mitigate this workload with the Wasserstein histogram interpolation using a lower number of reference points. Moreover, the AR based ground truth collection facility helps to automatically annotate the reference points without measuring them manually. The CCTV system used in the cow barn [10] does a similar online ground truth positioning of the cows, which is an indispensable requirement for measuring the performance correctly.

All of the works use low cost and affordable transmitters and sensors. Canton Paterna *et al.* [67] use multiple low cost computers to expand the coverage, because higher number of sensors relates to better positioning performance, but usage of stationary computers for sensing brings extra cost. Trogh *et al.* [10] propose low cost and battery powered receivers. Likewise, we bring forth a scalable solution to expand the sensing coverage using low cost USB Bluetooth adapters and extension cables. The cables can be extended more as long as they can supply enough power, and a lower number of computers can host a swarm of Bluetooth adapters.

In summary, when compared to the works with similar setups, the results show that the proposed work achieves better positioning performance with accurate evaluation. Moreover, we do not only compare our localization algorithm with its accuracy. We also show that it is scalable for space, hardware and complexity situations, as new sensors and low cost computers can be added easily and the complexity may be tuned with different particle population sizes.

7. CONCLUSION

In this work, we develop methods to render indoor localization and tracking to be practical using only the BLE technology. Bluetooth sensors that share the same environment with the transmitters extract RSSI parameters from the captured packets. The RSSI data can be used as position indicators, but they are very bad indicators because of both the multipath problem of the RF signal nature and the instability of the signal indicators through time. We deal with the tracking problem using BLE technology in three parts.

Firstly, we design an SMC technique that relies on the estimation of a probabilistic RF map from a small set of RSSI fingerprint measurements. The estimated RF map is subsequently used as the observation model of a dynamical system where we do target tracking by a sequential Monte Carlo algorithm. Not surprisingly, there is a direct relationship between the accuracy of the RF map and accuracy of localization and tracking. We show first that when RSSI fingerprints are collected on a very dense grid, a RF map can be accurately estimated and a satisfactory tracking performance can be obtained despite the high variability of the actual measurements. However, dense fingerprint sampling is often not practical or at least not desired in real applications as this requires careful measurement and a long data collection process. In this paper, we develop methods to obtain accurate RF maps from far fewer and sparsely sampled fingerprints and show empirically that tracking accuracy is still acceptable. We propose variations of the Wasserstein interpolation method, which is also originally derived from the Wasserstein distance computation method. The results of these Wasserstein variations are compared with the results of the well known regression methods, namely the nearest neighbor approach, the affine combination and the neural networks. We can generalize from the results that we estimate the unknown RF information on the arbitrary positions by an interpolation using the RF strength densities on previously measured positions. Moreover, Wasserstein combination variations are good candidates for histogram estimation purposes, as the error rates of these metods

increase consistently while the number of fingerprints are reduced, but they perform better compared to the other applied techniques.

Secondly, we model the motion of the tracked object as a diffusion, because we have no information on how the object moves. Considering that velocity of the object may vary or the emission density model may be inaccurate, we extend the standard state space model to include another latent layer for online variable parameter sampling. The diffusion particles are sampled with their own error covariance, a sensitivity value. We show that this sensitivity value can be chosen from a large space compared to the point estimate of the static diffusion factor. With an almost arbitrary sensitivity value, the diffusion factors get adapted to the problem and more accurate position particles can be achieved. We also show that this method is efficient for both performance and accuracy with respect to the classical SMC filter and the similar works of the state-of-art.

We finally develop a technique for annotating the RF signal parameters with high precision position data in both time and space domain using AR markers. Considering the inaccurate and various evaluation techniques of the indoor positioning algorithms, we see that the community lacks standardized evaluation procedures and benchmarks. This need makes high frequency ground truth data very valuable. Besides, we bring forth not only a method for fine grained ground truth data collection, but also a method for collecting data practically. We also make the data publicly available for the researchers who want to test their algorithms based on RSSI data. With error rates lower than 0.05 m, the position data can be regarded as the ground truth for the less accurate wireless positioning systems, especially for RSSI based localization.

The current design of this indoor positioning system takes several issues into consideration at the same time. The first issue is the affordability. We use cheap sensors, beacons and processing units that are easily affordable. The sensing setup can be constructed on low budget. Even though the data collection setup is similarly affordable and can be used repeatedly, we regard it as a one time and problem specific

construction.

Another issue is the practicality, which is attained through the easy installation of the sensors and the beacons and the use of ROS as the message passing system. When the system is set, any other processing/sensing unit can be inserted easily into the working system. This practical growth leads to scalability of the sensing system. With several cheap beacons multiple objects can be tracked. The only bottleneck is the required processing power, which can be tuned through the number of particles processed per iteration or the number of servers that are responsible for processing.

For the accuracy, there are systems and associated modalities that work definitely more accurately than our system design that uses beacons and Bluetooth adapters. The position annotation work is an example of a very accurate indoor positioning system. However, these systems either require high amounts of data to process at a time or rely on very expensive sensors, which prevent the researchers from installing them on any object of interest.

Our final system design is not as accurate as a vision based indoor positioning system, but our work shows that with a proper SMC tuning, it is possible to form a wireless based indoor positioning system that is sufficiently accurate, easily affordable, highly scalable and practical at the same time. The system can also be run in real-time.

For the future insights on RF map estimation methods, an immediate research would be to make an analysis on the appropriate fingerprint locations. Having the practicality of the method in mind, a researcher can find the number of fingerprint locations with respect to the area size and the required fineness. Additionally, if a method can give confidence for a position in the area, it may urge the researcher to gather data from the locations with low confidence values. A proper Gaussian Process Regression for multiple outputs would be a good future work to apply on this data as the algorithm provides the confidence values by its nature. Expectedly, with such small data, neural networks are unable to learn the nonlinear histogram on a planar

surface without over learning, but more features about positions or RF data can be used to improve neural network based histogram interpolation.

The contribution of the adaptive SMC has potential to be applied to the localization environments where propagation conditions change significantly over time. Localization algorithms with fixed parameters fail in such environments, however the proposed solution is based on fingerprinting, which may bring a bottleneck due to the empirical parameters. We believe that this bottleneck can also be overcome with the application of an histogram interpolation algorithm and a fast fingerprinting technique.

In the proposed IPS, captured data by the distributed sensors are transferred to a single computer where the SMC filter algorithm is run. As a future work, the system can be made decentralized by running the SMC filter on each sensor unit. This work will require to determine the particle sizes the sensor units can process in real time, and for better approximation the posterior information should be efficiently shared between the processing nodes in terms of particles.

We regard the AR based data annotation technique as a one-time prior assistive tool for the calibration of a wireless indoor positioning system. It is intended for collecting position annotated data which can later be used for parameter estimation of the tracking algorithms, fingerprinting and more importantly evaluation of the whole system in terms of position estimation accuracy. For future work, this method can be used for automatic, online and fast fingerprinting for wireless signal based localization, which is substantially a challenging task for the researchers working in the domain. The setup may be navigated to collect data from any possible point in the area, and these data can be processed to form the fingerprints fast and easily. Moreover, a self navigating robot can be utilized for real automatization of fingerprinting.

REFERENCES

1. Eyobu, O. S., A. Poulose, and D. S. Han, "An Accuracy Generalization Benchmark for Wireless Indoor Localization based on IMU Sensor Data", *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, pp. 1–3, 2018.

2. Kim, S.-C., Y.-S. Jeong, and S.-O. Park, "RFID-Based Indoor Location Tracking to Ensure the Safety of the Elderly in Smart Home Environments", *Personal Ubiquitous Comput.*, Vol. 17, No. 8, p. 1699–1707, 2013.

3. Komai, K., M. Fujimoto, Y. Arakawa, H. Suwa, Y. Kashimoto, and K. Yasumoto, "Elderly person monitoring in day care center using Bluetooth Low Energy", *ISMICT*, pp. 1–5, IEEE, 2016.

4. Chowdhury, M. J. M., M. S. Ferdous, K. Biswas, N. Chowdhury, and V. Muthukkumarasamy, "COVID-19 Contact Tracing: Challenges and Future Directions", *IEEE Access*, Vol. 8, pp. 225703–225729, 2020.

5. Li, F., C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A Reliable and Accurate Indoor Localization Method Using Phone Inertial Sensors", *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pp. 421–430, ACM, New York, NY, USA, 2012.

6. Kuo, Y.-S., P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light", *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pp. 447–458, ACM, New York, NY, USA, 2014.

7. Brena, R., J. García-Vázquez, C. Galván Tejada, D. Muñoz, C. Vargas-Rosales, J. Fangmeyer Jr, and A. Palma, "Evolution of Indoor Positioning Technologies:

A Survey", *Journal of Sensors*, Vol. 2017, p. 2630413, 2017.

8. Dickinson, P., G. Cielniak, O. Szymanezyk, and M. Mannion, "Indoor positioning of shoppers using a network of Bluetooth Low Energy beacons", *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, 2016.

9. Bisio, I., A. Sciarrone, and S. Zappatore, "A new asset tracking architecture integrating RFID, Bluetooth Low Energy tags and ad hoc smartphone applications", *Pervasive and Mobile Computing*, Vol. 31, pp. 79–93, 2016.

10. Trogh, J., D. Plets, L. Martens, and W. Joseph, "Bluetooth low energy based location tracking for livestock monitoring", *Proceedings of the 8th European Conference on Precision Livestock Farming*, pp. 469–475, 2017.

11. Siegwart, R. and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, Bradford Company, Scituate, MA, USA, 2004.

12. Thrun, S., W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, 2005.

13. Xiao, A., R. Chen, D. Li, Y. Chen, and D. Wu, "An Indoor Positioning System Based on Static Objects in Large Indoor Scenes by Using Smartphone Cameras", *Sensors*, Vol. 18, No. 7, p. 2229, 2018.

14. Biswas, J. and M. Veloso, "Depth camera based indoor mobile robot localization and navigation", *2012 IEEE International Conference on Robotics and Automation*, pp. 1697–1702, 2012.

15. Simon, G., G. Zachár, and G. Vakulya, "Lookup: Robust and Accurate Indoor Localization Using Visible Light Communication", *IEEE Transactions on Instrumentation and Measurement*, Vol. 66, No. 9, pp. 2337–2348, 2017.

16. Koc, I. A., T. Serif, S. Gören, and G. Ghinea, "Indoor Mapping and Positioning using Augmented Reality", *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 335–342, 2019.

17. Medina, C., J. Segura, and A. Torre, "Ultrasound Indoor Positioning System Based on a Low-Power Wireless Sensor Network Providing Sub-Centimeter Accuracy", *Sensors (Basel, Switzerland)*, Vol. 13, pp. 3501–26, 2013.

18. Kumar, G. A., A. K. Patil, R. Patil, S. S. Park, and Y. H. Chai, "A LiDAR and IMU Integrated Indoor Navigation System for UAVs and Its Application in Real-Time Pipeline Classification", *Sensors*, Vol. 17, No. 6, p. 1268, 2017.

19. Simões, W. C. S. S., W. A. E. Silva, M. M. D. Lucena, N. Jazdi, and V. F. D. Lucena, "A Hybrid Indoor Positioning System Using a Linear Weighted Policy Learner and Iterative PDR", *IEEE Access*, Vol. 8, pp. 43630–43656, 2020.

20. Gezici, S., "A Survey on Wireless Position Estimation", *Wirel. Pers. Commun.*, Vol. 44, pp. 263–282, 2008.

21. Diallo, A., Z. Lu, and X. Zhao, "Wireless Indoor Localization Using Passive RFID Tags", *Procedia Computer Science*, Vol. 155, pp. 210 – 217, 2019, the 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology.

22. Xu, H., Y. Ding, P. Li, R. Wang, and Y. Li, "An RFID Indoor Positioning Algorithm Based on Bayesian Probability and K-Nearest Neighbor", *Sensors*, Vol. 17, p. 1806, 2017.

23. Alarifi, A., A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. Al-Ammar, and H. Al-Khalifa, "Ultra Wideband Indoor Positioning Technologies:

Analysis and Recent Advances", *Sensors*, Vol. 16, No. 5, p. 707, 2016.

24. Cinefra, N., *An adaptive indoor positioning system based on Bluetooth low energy RSSI*, Master's thesis, Politecnico di Milano, 2013.

25. Kriz, P., F. Maly, and K. Tomáš, "Improving Indoor Localization Using Bluetooth Low Energy Beacons", *Mobile Information Systems*, Vol. 2016, pp. 1–11, 2016.

26. Neburka, J., Z. Tlamsa, V. Benes, L. Polak, O. Kaller, L. Bolecek, J. Sebesta, and T. Kratochvil, "Study of the performance of RSSI based Bluetooth Smart indoor positioning", *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 121–125, 2016.

27. Čabarkapa, D., I. Grujić, and P. Pavlović, "Comparative analysis of the Bluetooth Low-Energy indoor positioning systems", *2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, pp. 76–79, 2015.

28. Han, G., H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization Algorithms of Wireless Sensor Networks: A Survey", *Telecommun. Syst.*, Vol. 52, No. 4, p. 2419–2436, 2013.

29. Martin, P., B.-J. Ho, N. Grupen, S. Muñoz, and M. Srivastava, "An iBeacon Primer for Indoor Localization: Demo Abstract", *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, BuildSys '14, pp. 190–191, ACM, New York, NY, USA, 2014.

30. Boukerche, A., H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Localization systems for wireless sensor networks", *IEEE Wireless Communications*, Vol. 14, No. 6, pp. 6–12, 2007.

31. Chen, Z., H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of WiFi, Smartphone Sensors and Landmarks Using the Kalman Filter for Indoor Local-

ization", *Sensors*, Vol. 15, No. 1, pp. 715–732, 2015.

32. Chen, Z., Q. Zhu, and Y. C. Soh, "Smartphone Inertial Sensor-Based Indoor Localization and Tracking With iBeacon Corrections", *IEEE Transactions on Industrial Informatics*, Vol. 12, No. 4, pp. 1540–1549, 2016.

33. Mahiddin, N. A., N. Safie, E. Nadia, S. Safei, and E. Fadzli, "Indoor position detection using WiFi and trilateration technique", *The International Conference on Informatics and Applications (ICIA2012)*, pp. 362–366, The Society of Digital Information and Wireless Communication, 2012.

34. Dahlgren, E. and H. Mahmood, *Evaluation of indoor positioning based on Bluetooth Smart technology*, Master's thesis, Chalmers University of Technology, 2014.

35. Muñoz-Salinas, R., M. J. Marín-Jimenez, E. Yeguas-Bolivar, and R. Medina-Carnicer, "Mapping and localization from planar markers", *Pattern Recognition*, Vol. 73, pp. 158–171, 2018.

36. Pritt, N., "Indoor location with Wi-Fi fingerprinting", *IEEE Applied Imagery Pattern Recognition Workshop, AIPR 2013, Washington, DC, USA, October 23-25, 2013*, pp. 1–8, 2013.

37. Daniş, F. S. and A. T. Cemgil, "Model-Based Localization and Tracking Using Bluetooth Low-Energy Beacons", *Sensors*, Vol. 17, No. 11, p. 2484, 2017.

38. Liu, H., H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems", *Trans. Sys. Man Cyber Part C*, Vol. 37, No. 6, pp. 1067–1080, 2007.

39. Farid, Z., R. Nordin, and M. Ismail, "Recent Advances in Wireless Indoor Localization Techniques and System", *Journal of Computer Networks and Communications*, Vol. 2013, p. 12, 2013.

40. Bisio, I., F. Lavagetto, M. Marchese, and A. Sciarrone, "Smart probabilistic fingerprinting for WiFi-based indoor positioning with mobile devices", *Pervasive and Mobile Computing*, Vol. 31, pp. 107–123, 2016.

41. Doucet, A., N. de Freitas, and N. Gordon, *An Introduction to Sequential Monte Carlo Methods*, pp. 3–14, Springer New York, New York, NY, 2001.

42. Cappé, O., E. Moulines, and T. Ryden, *Inference in Hidden Markov Models (Springer Series in Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2005.

43. Villani, C., *The Wasserstein Distances*, pp. 93–111, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

44. Benamou, J.-D. and Y. Brenier, "A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem", *Numerische Mathematik*, Vol. 84, No. 3, pp. 375–393, 2000.

45. Papadakis, N., G. Peyré, and E. Oudet, "Optimal Transport with Proximal Splitting", *SIAM Journal on Imaging Sciences*, Vol. 7, No. 1, pp. 212–238, 2014.

46. Daniş, F. S., A. T. Cemgil, and C. Ersoy, "Adaptive Sequential Monte Carlo Filter for Indoor Positioning and Tracking With Bluetooth Low Energy Beacons", *IEEE Access*, Vol. 9, pp. 37022–37038, 2021.

47. Yildirim, S., S. S. Singh, T. Dean, and A. Jasra, "Parameter Estimation in Hidden Markov Models With Intractable Likelihoods Using Sequential Monte Carlo", *Journal of Computational and Graphical Statistics*, Vol. 24, No. 3, pp. 846–865, 2015.

48. Kantas, N., A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On Particle Methods for Parameter Estimation in State-Space Models", *Statist. Sci.*, Vol. 30, No. 3, pp. 328–351, 2015.

49. Yun, J., F. Yang, and Y. Chen, "Augmented Particle Filters", *Journal of the American Statistical Association*, Vol. 112, No. 517, pp. 300–313, 2017.

50. Kim, J. and H. Jun, "Vision-based location positioning using augmented reality for indoor navigation", *IEEE Transactions on Consumer Electronics*, Vol. 54, No. 3, pp. 954–962, 2008.

51. Ai, H., K. Tang, W. Huang, S. Zhang, and T. Li, "Fast fingerprints construction via GPR of high spatial-temporal resolution with sparse RSS sampling in indoor localization", *Computing*, Vol. 102, No. 3, pp. 781–794, 2020.

52. Li, C., Q. Xu, Z. Gong, and R. Zheng, "TuRF: Fast data collection for fingerprint-based indoor localization", *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, 2017.

53. Zhang, L., X. Liu, J. Song, C. Gurrin, and Z. Zhu, "A Comprehensive Study of Bluetooth Fingerprinting-Based Algorithms for Localization", *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pp. 300–305, 2013.

54. Pelant, J., Z. Tlamsa, V. Benes, L. Polak, O. Kaller, L. Bolecek, J. Kufa, J. Sebesta, and T. Kratochvil, "BLE device indoor localization based on RSS fingerprinting mapped by propagation modes", *2017 27th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pp. 1–5, 2017.

55. Zhuang, Y., J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons", *Sensors*, Vol. 16, No. 5, p. 596, 2016.

56. Faragher, R. and R. Harle, "An analysis of the accuracy of bluetooth low energy for indoor positioning applications", *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+*

2014), Tampa, FL, USA, Vol. 812, 2014.

57. Faragher, R. and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons", *IEEE Journal on Selected Areas in Communications*, Vol. 33, No. 11, pp. 2418–2428, 2015.

58. Chen, L., L. Pei, H. Kuusniemi, Y. Chen, T. Kröger, and R. Chen, "Bayesian Fusion for Indoor Positioning Using Bluetooth Fingerprints", *Wireless Personal Communications*, Vol. 70, No. 4, pp. 1735–1745, 2013.

59. Zhang, V. Y. and A. K.-S. Wong, "Kernel-based Particle Filtering for Indoor Tracking in WLANs", *J. Netw. Comput. Appl.*, Vol. 35, No. 6, pp. 1807–1817, 2012.

60. Zhang, V. Y., A. K.-S. Wong, and K. T. Woo, "Histogram Based Particle Filtering with Online Adaptation for Indoor Tracking in WLANs", *International Journal of Wireless Information Networks*, Vol. 19, No. 3, pp. 239–253, 2012.

61. Bonneel, N., G. Peyré, and M. Cuturi, "Wasserstein Barycentric Coordinates: Histogram Regression Using Optimal Transport", *ACM Trans. Graph.*, Vol. 35, No. 4, p. 71, 2016.

62. Solomon, J., F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, "Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains", *ACM Trans. Graph.*, Vol. 34, No. 4, pp. 66:1–66:11, 2015.

63. Nurminen, H., A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization", *International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–10, 2013.

64. Shubair, R. M. and H. Elayan, "Enhanced WSN localization of moving nodes using a robust hybrid TDOA-PF approach", *2015 11th International Conference*

on Innovations in Information Technology (IIT), pp. 122–127, 2015.

65. Filípek, P. and A. Kovarova, "Indoor Localization Based on Beacons and Calculated by Particle Filter", *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, CompSysTech '16, p. 269–276, Association for Computing Machinery, New York, NY, USA, 2016.

66. Zafari, F. and I. Papapanagiotou, "Enhancing iBeacon Based Micro-Location with Particle Filtering", *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, 2015.

67. Cantón Paterna, V., A. Calveras Augé, J. Paradells Aspas, and M. A. Pérez Bullones, "A Bluetooth Low Energy Indoor Positioning System with Channel Diversity, Weighted Trilateration and Kalman Filtering", *Sensors*, Vol. 17, No. 12, p. 2927, 2017.

68. Chopin, N., "A sequential particle filter method for static models", *Biometrika*, Vol. 89, No. 3, pp. 539–552, 2002.

69. Papavasiliou, A., "Parameter estimation and asymptotic stability in stochastic filtering", *Stochastic Processes and their Applications*, Vol. 116, No. 7, pp. 1048 – 1065, 2006.

70. Chopin, N., P. E. Jacob, and O. Papaspiliopoulos, "SMC 2 : an efficient algorithm for sequential analysis of state space models", *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, Vol. 75, No. 3, pp. 397–426, 2013.

71. Crisan, D. and J. Míguez, "Nested particle filters for online parameter estimation in discrete-time state-space Markov models", *Bernoulli*, Vol. 24, No. 4A, pp. 3039–3086, 2018.

72. Pitt, M. K. and N. Shephard, "Filtering via Simulation: Auxiliary Particle Filters", *Journal of the American Statistical Association*, Vol. 94, No. 446, pp. 590–

599, 1999.

73. Bagdanov, A. D., A. Del Bimbo, F. Dini, and W. Nunziati, "Adaptive uncertainty estimation for particle filter-based trackers", *14th International Conference on Image Analysis and Processing (ICIAP 2007)*, pp. 331–336, 2007.

74. Lang, H., T. Li, G. Villarrubia, S. Sun, and J. Bajo, "An Adaptive Particle Filter for Indoor Robot Localization", Mohamed, A., P. Novais, A. Pereira, G. Villarrubia González, and A. Fernández-Caballero (editors), *Ambient Intelligence - Software and Applications*, pp. 45–55, Springer International Publishing, Cham, 2015.

75. Fox, D., "KLD-Sampling: Adaptive Particle Filters", *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, p. 713–720, MIT Press, Cambridge, MA, USA, 2001.

76. De la Osa, C. M., G. G. Anagnostopoulos, M. Togneri, M. Deriaz, and D. Konstantas, "Positioning evaluation and ground truth definition for real life use cases", *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, 2016.

77. Poulose, A. and D. S. Han, "Hybrid Deep Learning Model Based Indoor Positioning Using Wi-Fi RSSI Heat Maps for Autonomous Applications", *Electronics*, Vol. 10, No. 1, p. 2, 2021.

78. Adler, S., S. Schmitt, K. Wolter, and M. Kyas, "A survey of experimental evaluation in indoor localization research", *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–10, 2015.

79. Mendoza-Silva, G. M., M. Matey-Sanz, J. Torres-Sospedra, and J. Huerta, "BLE RSS Measurements Dataset for Research on Accurate Indoor Positioning", *Data*, Vol. 4, No. 1, p. 12, 2019.

80. Baronti, P., P. Barsocchi, S. Chessa, F. Mavilia, and F. Palumbo, "Indoor Bluetooth Low Energy Dataset for Localization, Tracking, Occupancy, and Social Interaction", *Sensors*, Vol. 18, No. 12, p. 4462, 2018.

81. Girolami, M., F. Mavilia, and F. Delmastro, "A bluetooth low energy dataset for the analysis of social interactions with commercial devices", *Data in Brief*, Vol. 32, p. 106102, 2020.

82. Sansano-Sansano, E., F. J. Aranda, R. Montoliu, and F. J. Álvarez, "BLE-GSpeed: A New BLE-Based Dataset to Estimate User Gait Speed", *Data*, Vol. 5, No. 4, p. 115, 2020.

83. Alnabhan, A. and B. Tomaszewski, "INSAR: Indoor Navigation System Using Augmented Reality", *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, ISA '14, p. 36–43, Association for Computing Machinery, New York, NY, USA, 2014.

84. Byrne, D., M. Kozlowski, R. Santos-Rodriguez, R. Piechocki, and I. Craddock, "Residential wearable RSSI and accelerometer measurements with detailed location annotations", *Scientific Data*, Vol. 5, No. 1, p. 180168, 2018.

85. La Delfa, G. C., S. Monteleone, V. Catania, J. F. De Paz, and J. Bajo, "Performance analysis of visualmarkers for indoor navigation systems", *Frontiers of Information Technology & Electronic Engineering*, Vol. 17, No. 8, pp. 730–740, 2016.

86. Xavier, R. S., B. M. F. da Silva, and L. M. G. Goncalves, "Accuracy Analysis of Augmented Reality Markers for Visual Mapping and Localization", *2017 Workshop of Computer Vision (WVC)*, pp. 73–77, 2017.

87. Ujkani, E., J. Dybedal, A. Aalerud, K. B. Kaldestad, and G. Hovland, "Visual Marker Guided Point Cloud Registration in a Large Multi-Sensor Industrial

Robot Cell", *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pp. 1–6, 2018.

88. Chollet, F., "Keras", `https://github.com/fchollet/keras`, 2015.

89. Chollet, F., *Deep Learning with Python*, Manning Publications Co., USA, 1st edition, 2017.

90. Tieleman, T. and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude", COURSERA: Neural Networks for Machine Learning, 2012.

91. Ferradans, S., N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol, "Regularized Discrete Optimal Transport", Kuijper, A., K. Bredies, T. Pock, and H. Bischof (editors), *SSVM 2013 - International Conference on Scale Space and Variational Methods in Computer Vision*, Vol. 7893 of *Lecture Notes in Computer Science*, pp. 428–439, Springer, Schloss Seggau, Leibnitz, Austria, 2013.

92. Gordon, N., D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEEE Proceedings F, Radar and Signal Processing*, Vol. 140, No. 2, pp. 107–113, 1993.

93. Kitagawa, G., "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models", *Journal of Computational and Graphical Statistics*, Vol. 5, No. 1, pp. 1–25, 1996.

94. Li, T., M. Bolic, and P. M. Djurić, "Resampling Methods for Particle Filtering: Classification, implementation, and strategies", *IEEE Signal Processing Magazine*, Vol. 32, No. 3, pp. 70–86, 2015.

95. Romero-Ramirez, F. J., R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers", *Image and Vision Computing*, Vol. 76, pp. 38–47, 2018.

96. Garrido-Jurado, S., R. Munoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using Mixed Integer Linear Programming", *Pattern Recognition*, Vol. 51, pp. 481–491, 2016.

97. Liang, K. K., "Efficient conversion from rotating matrix to rotation axis and angle by extending Rodrigues' formula", *CoRR*, Vol. abs/1810.02999, 2018, `http://arxiv.org/abs/1810.02999`.

98. Fischler, M. A. and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Commun. ACM*, Vol. 24, No. 6, p. 381–395, 1981.

99. Wang, J., A. Hu, X. Li, and Y. Wang, "An Improved PDR/Magnetometer/Floor Map Integration Algorithm for Ubiquitous Positioning Using the Adaptive Unscented Kalman Filter", *ISPRS International Journal of Geo-Information*, Vol. 4, No. 4, p. 2638, 2015.

100. An, G. H., S. Lee, M.-W. Seo, K. Yun, W.-S. Cheong, and S.-J. Kang, "Charuco Board-Based Omnidirectional Camera Calibration Method", *Electronics*, Vol. 7, No. 12, p. 421, 2018.

101. Alpaydin, E., *Introduction to Machine Learning*, The MIT Press, 2nd edition, 2010.

102. Arlot, S. and A. Celisse, "A survey of cross-validation procedures for model selection", *Statist. Surv.*, Vol. 4, pp. 40–79, 2010.

APPENDIX A: SYNTHETIC DATA GENERATION

A.1. Trajectory Sampling

The generative model from which we sample the trajectory points follows:

$$
\begin{aligned}
t_0 &= 0 \\
\mathbf{x}_0 &\sim \mathcal{U}(\mathcal{A}) \\
\theta_0 &\sim \mathcal{U}([0, 2\pi]) \\
\tilde{\Delta}_t^t &\sim \mathcal{G}(a_t, b_t) \\
\tilde{\Delta}_t^l &\sim \mathcal{G}(a_l, b_l) \\
\tilde{\Delta}_t^\theta &\sim \mathcal{N}(0, s) \\
t_t &= t_{t-1} + \tilde{\Delta}_t^t \\
x_{1t} &= x_{1t-1} + \tilde{\Delta}_t^l \cos(\theta_{t-1} + \tilde{\Delta}_t^\theta) \\
x_{2t} &= x_{2t-1} + \tilde{\Delta}_t^l \sin(\theta_{t-1} + \tilde{\Delta}_t^\theta) \\
\theta_t &= \theta_{t-1} + \tilde{\Delta}_t^\theta.
\end{aligned}
\tag{A.1}
$$

We sample an initial location in $\mathcal{A}$ and an initial orientation between 0 and 2π, and set the initial timestamp to 0. At each step, a time interval, $\tilde{\Delta}_t^t$, and a distance value, $\tilde{\Delta}_t^l$, are sampled from gamma distributions, and a rotation value, $\tilde{\Delta}_t^\theta$, from a normal distribution. The parameters of the distributions, a_t, b_t, a_l, b_l, s are experimentally set to generate values similar with the real data. To prevent our virtual robot from leaving the area, sampled rotation values are deliberately manipulated by an additional value of $\frac{\pi}{8}$ according to the current orientation. This helps the robot return smoothly into the area when it approaches the borders. Sample trajectories can be seen in Figure 6.4.

A.2. Data Sampling

The generative model from which we sample the RSSI data on the trajectory points follows:

$$\begin{aligned} b_t &\sim \mathcal{M}(\boldsymbol{b}, p_t) \\ i^* &= \arg\min_i d(\mathbf{x}_t, \mathbf{f}_i) \\ y_t &\sim \mathcal{M}(\mathbf{h}_{b_t}^{\mathbf{f}_{i^*}}). \end{aligned} \tag{A.2}$$

We first sample a beacon label from the set of all available beacons $\mathbf{b}$, according to a probability distribution p_t, which reshapes at each time step inversely proportional to the use counts of the beacons, in order to let the beacons be sampled uniformly in small time windows. We then find the closest fingerprint index (i^*), and sample an RSSI value, y_t, using the corresponding beacon histogram.

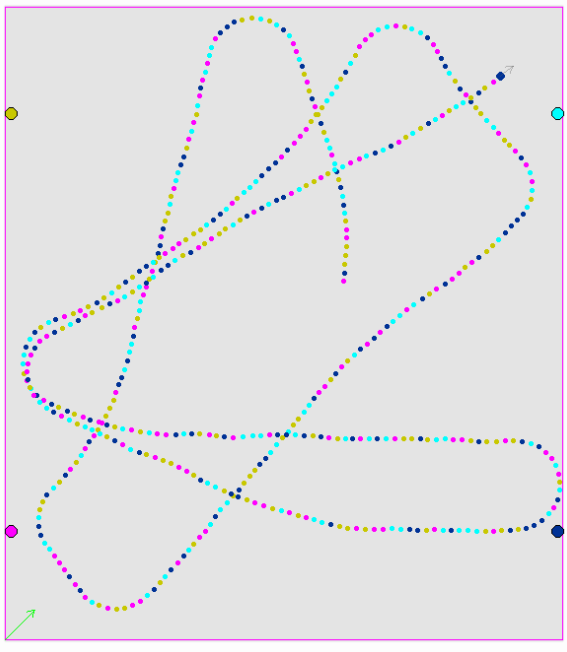


Figure A.1. A randomly generated trajectory and associated RSSI data