UNIVERSITY OF CALIFORNIA,
IRVINE

Internet of Things-Based Collaborative Position-Sensing Systems for Cardiopulmonary Resuscitation and Indoor Localization

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Hsinchung Chen

Dissertation Committee:
Professor Emeritus Pai H. Chou, Chair
Professor Nader Bagherzadeh
Professor Rainer Dömer

2022

DEDICATION

This dissertation is dedicated to my parents and my family for their support and love.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Pai Chou, for giving me support over the years. His care, patience, kindness, and enthusiasm has inspired me and helped me go through many difficulties and challenges. His immense knowledge and guidance have strengthened my skills of doing scientific research and developing technical projects. It is a great honor for me to work with him. I may not have enough words to show my appreciation, but I will never run out of my gratitude to my advisor, Dr. Pai Chou.

I would like to thank all members of EPL lab, Subramanian Meenakshi Sundaram, Ali HeydariGorji, and Seyede Mahya Safavi, for the research projects and papers we work with and for all the joyful time and fun talks we have in the lab.

I would also like to thank Dr. Ruey-Kang Chang, for all his great ideas, help and support.

This work was supported by a Broadcom scholarship and NIH Grants 1R41GM113463-01A1 and 2R42GM113463-02.

VITA

Hsinchung Chen

EDUCATION

Doctor of Philosophy in Electrical and Computer Engineering **2022**
University of California Irvine *Irvine, California*

Master in Computer Science and Engineering **2009**
National Chiao Tung University *Hsinchu, Taiwan*

Bachelor in Computer Science **2007**
National Chiao Tung University *Hsinchu, Taiwan*

RESEARCH EXPERIENCE

IoT Medical Device Development **2016–2020**
University of California, Irvine *Irvine, California*

Multi-core Embedded System Design **2016–2017**
National Chiao Tung University *Hsinchu, Taiwan*

TEACHING EXPERIENCE

Teaching Assistant **2016–2017**
University of California Irvine *Irvine, California*

REFEREED CONFERENCE PUBLICATIONS

BlueBox: A Complete Recorder for Code-Blue Events in Hospitals **2019**
Hsinchung Chen, Subramanian Meenakshi, Ali HeydariGorji, Seyede Mahya Safavi, Pai H. Chou, Cheng-Ting Lee, Ruey-Kang Chang.

Proceedings of the International Symposium on VLSI Design, Automation and Test.

EcoLoc: Toward Universal Location Sensing by Encounter-Based Collaborative Indoor Localization **2017**
Hsinchung Chen, Yi Lin Chen, Chia Hsun Wu, Mohammad Abdullah Al Faruque, Pai H. Chou

The ACM/IEEE International Conference on Internet of Things Design and Implementation.

Security and Privacy Challenges in IoT-Based Machine-to-Machine Collaborative Scenarios **2016**
Hsinchung Chen, Mohammad Abdullah Al Faruque, Pai H. Chou

Proceedings of the International Conference on Hardware Software Codesign and System Synthesis.

PAIS: Parallelization Aware Instruction Scheduling for Improving Soft-error Reliability of GPU-based Systems **2016**
Haeseung Lee, Hsinchung Chen, Mohammad Abdullah Al Faruque

Proceedings of Design Automation and Test in Europe.

ABSTRACT OF THE DISSERTATION

Internet of Things-Based Collaborative Position-Sensing Systems for Cardiopulmonary Resuscitation and Indoor Localization

By

Hsinchung Chen

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2022

Professor Emeritus Pai H. Chou, Chair

The ubiquitous IoT devices have promoted the growth of applications that require target positioning. While precise positioning remains a challenge for IoT devices, existing methods mainly focus on optimizing the positioning algorithms or expanding the sensing modalities. One opportunity area of improvement is to leverage their wireless communication capability to exchange sensing data of different modalities with nearby devices. This work proposes collaborative sensing frameworks to enhance the accuracy of position inference.

Our approach to collaborative sensing is divided into selection of reliable devices, the data-exchange mechanism, and the data-fusion algorithm. We demonstrate our approach on two different position-sensing systems: one for chest compression-depth estimation and one for indoor pedestrian localization. The key components used in the frameworks include quality assessment of sensor data using mutual information, data-fusion algorithms based on the chest-compression model and the pedestrian-encounter model, and the data-exchange mechanism using Bluetooth Low Energy (BLE). A real-time position-estimation method based on our chest-compression model is proposed to remove the noise and handle the cumulative error. A collaborative conditional random field algorithm is developed to reduce the convergence distance in the localization estimation. Experimental results show our collaborative-sensing approach achieves higher rate of convergence than the existing solutions in terms of real-time estimated position.

This dissertation proposes an alternative approach to optimizing the position-sensing systems through collaborative sensing. Our work represents the first step towards the universal positioning on IoT devices and provides research opportunities for further improvement and exploration.

Chapter 1

Introduction

The advances in wireless communications and the ubiquity of mobile IoT (Internet of Things) devices have enabled collaborative position sensing. This chapter first motivates our work on collaborative position sensing with two positioning applications, followed by the contributions and an outline of this dissertation.

1.1 Collaborative Position Sensing on IoT Devices

Location has become one of the fundamental features for applications that provide location-based services (LBS) [47]. A positioning process can be divided into detecting target actions and estimating the target positions based on the characterization of the scenario and the sensing modality. Positioning systems have been widely studied and deployed, such as the use of Global Positioning System (GPS) for outdoor localization. While positioning based on a combination of GPS and cellular tower triangulation is relatively mature and well understood, positioning based on other sensing modalities remains a challenge especially for indoor environments [63].

Positioning is one of the main objectives in the IoT [5]. The advances in semiconductor and wireless communication technologies for integrating different sensing, processing, and wireless connectivity hardware components into a stand-alone system have made possible the IoT position-sensing and distance-measuring systems [26]. The IoT positioning systems sense their physical surroundings and send the corresponding response to service providers to offer users ubiquitous and mobile IoT services [69].

The communication capabilities of IoT devices enable not only data transmission but often also collaborative sensing or data sharing among nearby devices to provide IoT services [51]. The existing approaches have explored various sensing modalities to enable collaborative sensing [24, 41]. However, it is impractical to deploy a variety of sensing systems and process large amounts of heterogeneous data that is received from all the IoT devices. Specifically, the estimated position typically has a higher correlation with the data measured by the nearby sensing systems. In addition, the validity of sensing data is also dependent on the position and its physical surroundings. Therefore, a framework that assists system designers with the deployment and selection of devices and evaluation of the quality of shared data is needed to make the best inference [71].

To highlight the core problems in collaborative positioning on mobile IoT devices, we consider two real-world applications: chest compression depth (CCD), the measurement of how deep the sternum is compressed during CPR, and pedestrian dead reckoning (PDR). What these two problems have in common are that both make use of inertial sensors, and the positioning algorithms must be lightweight and real-time to run on resource-limited IoT devices.

Two collaborative algorithms covering sensor selection, data fusion algorithms, and data exchange mechanisms for the CCD and PDR applications are proposed. We demonstrate the feasibility and practicality of our solutions by actual implementations that exploit the communication capability on real-world IoT devices.

1.2 Dissertation Contributions

Our research proposes the collaborative positioning system design based on the short-range communication capability of IoT. Our system design principle, which extends the sensing modality from an individual IoT device to a cluster of IoT devices, is a major step toward enabling position sensing universally by all mobile IoT devices. We study two position-sensing systems for chest compression depth estimation and pedestrian indoor localization to demonstrate the advantages of collaborative sensing on IoT devices. The contributions of our research are as follows:

1. A collaborative position-sensing framework for real-time CCD estimation that includes
    - A novel chest-compression model for a data fusion algorithm to suppress cumulative error in depth estimation.
    - A mutual-information-based evaluation method for IoT device placement to obtain qualified sensor data.
2. A collaborative indoor localization system among mobile IoT devices, as enabled by
    - A power-efficient encounter model to discover nearby devices and exchange data
    - A collaborative data fusion algorithm to enable universal location sensing by all mobile IoT devices

1.3 Dissertation Outline

This dissertation is organized into two parts: Part I covers the chest compression problem, and Part II investigates collaborative indoor localization. In Part I, Chapter 2 provides a background and related work on chest compression depth estimation in general, and Chapter 3 presents our technical approach to the chest compression problem, including our model suitable for IoT devices, with an evaluation in Chapter 4. Part II is organized also as three chapters, where Chapter 5 surveys related work on indoor localization, Chapter 6 presents our technical approach based on an encounter-based model, and Chapter 7 provides our experimental results. Finally, Chapter 8 concludes this dissertation with directions for future research.

Part I

Chest Compression Depth Estimation

Chapter 2

Background and Related Work

Cardiopulmonary resuscitation (CPR) is a emergency life-saving procedure performed to revive a patient who is unconscious and not breathing. This chapter provides a background on CPR and the existing electronic devices that were developed to assist in performing CPR. Related work is surveyed on chest compression and a table is provided to summarizes the techniques and algorithms for comparison with our work.

2.1 Background

This section elaborates the CPR procedure and the metrics to evaluate the quality of performed CPR and motivates the need for a CPR data logger. Specifically, the data logger should capture chest compression rate and depth as a main requirements.

2.1.1 CPR

CPR, for *cardiopulmonary resuscitation*, is an emergency procedure performed on a person who is in a cardiac or respiratory arrest. It is useful in many different scenarios, including a heart attack or drowning. CPR entails chest compression followed by assisted breathing. By keeping the blood flowing and carrying oxygen to the body, CPR can keep the patient's vital organs alive before the regular heart rhythm is restored. It can be performed on adults and children by trained and even untrained individuals while being coached live over the phone.

2.1.2 CPR Quality

The quality of CPR can directly impact the survival rate of the patients. In 2020, the American Heart Association (AHA) updated the *Guidelines for CPR and Emergency Cardiovascular Care* [8] to include an algorithm to handle adult cardiac arrest as shown in Fig. 2.1. The algorithm quantified the quality metrics of critical components of the resuscitation procedures [8] including proper chest compression depth (minimum depth of 2 inches or 5 cm), fast chest compression rate (100-120 per minute), minimizing interruptions in chest compressions, and avoiding excessive ventilation. In addition, the assessment of cardiac rhythm determines when to perform defibrillation and administer medication.

2.1.3 Improving CPR by Instant Feedback

Real-time feedback is useful to correct deviation from proper CPR. Specifically, instant feedback on chest compression depth (CCD) and chest compression rate (CCR) can be provided in resuscitation education by selecting mannequins or task trainers on the basis of the availability of chest compression features [15]. The assumption is that by repeated practicing, the trainee will more likely be able to perform CPR correctly. However, without a clear indication, there is a high

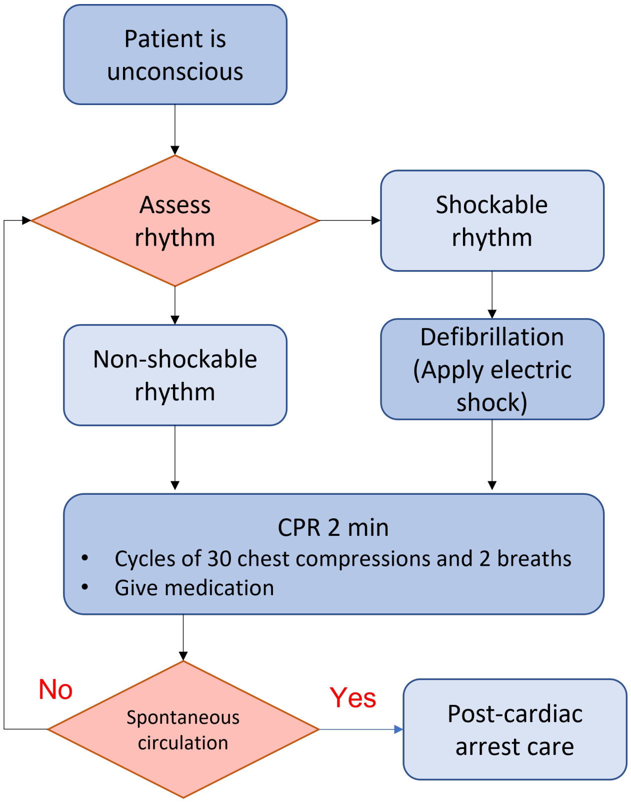


Figure 2.1: CPR Flow for Adult Cardiac Arrest [8]

chance that those criteria will not be met such that the survival rate decreases [52, 53, 49]. Significant variability was observed that a survival rate of only 9% at night was achieved compared to nearly 37% survival in operating rooms or postanesthesia care units during the day [56]. Even in hospitals, the survival rate is over 20% if the arrest occurs between 7 am and 11 pm, but only 15% between 11 pm and 7 am [56]. Therefore, instant feedback should be considered during the CPR procedure to address these issues. The feedback is also important for identifying errors and improving performance in post-CPR debriefings [66, 20, 18, 81, 4]. The high survival variability in different conditions suggests the need of feedback for CPR reviews [30].

2.1.4 Need for CPR Sensors and Loggers

Several electronic devices have been developed to assist rescuers in performing CPR [42, 83]. However, they have not been in wide use, as they need precise placement to be accurate, and they can easily interfere with the CPR procedure. In addition, since the integration of various sensors into one miniature device can enable physiological sensing [31], this type of usage may impede the possibility of multimodal sensing and analysis.

2.2 Related Work

This section surveys existing work on chest compression depth estimation. Related work can be classified into non-inertial sensor-based and inertial sensor-based solutions. The latter is further elaborated based on the used technique and proposed algorithms. Finally, this section summarizes the existing works in a table.

2.2.1 Non-Inertial Sensor-Based Solutions

A varietiy of devices and sensors are used for estimating CCD along with proposed methods. Computed tomography by Jin et al. [34] can provide the thorax scan to estimate CCD, but it is for studying purpose and is not applicable to out-of-hospital or real-time use. The use of force and pressure sensors [50] for CCD estimation are based on the correlation between compression force and displacement of chest. However, the research in evaluating the consistency among individuals is too limited to support the applicability. Thoracic impedance (TI) is studied to identify chest compression [9], but a recent study has shown that TI is unreliable as a predictor of CCD [6].

Other sensing modalities such as vision-based or radio signal-based sensors are adopted for CCD estimation. Mini-VREM by Semeraro et al. [62] is a vision-based system implemented on a PC and uses cameras to estimates CCD by using image processing techniques. The CCD estimation flow of Mini-VREM, which first detects chest compression and then estimates the CCD, is similar to other sensor-measurement methods, but the algorithms are developed based on image-processing techniques. QCPR cam-app [84] is an app that evaluates if it is feasible to provide feedback in real-time by using a smartphone camera. The study only provides chest compression rate but lacks CCD estimation. The radio signal-based method that uses transmission losses of RF [29] during chest compression is evaluated, but the CCD is not provided. Ultrasound and ulra-wideband (UWB) sensors are adopted in Dolgov et al. [19] and Yu et al. [79], respectively, to estimate the CCD by the product of the signal travel time and the signal transmission speed in human body. To obtain the radio signal measurement, a pair of transceiver and receiver must be placed on the chest and the back of the patient correspondingly for the signal traveling through the thorax. The complex setup required by the above mentioned methods is less favorable in real rescue processes.

2.2.2 Inertial Sensor-Based Solutions

Inertial sensors such as accelerometer and gyroscope are more popular than the mentioned sensors due to their low cost, miniature form factor, and ease of deployment. However, the evaluation in Kowalczuk and Merta [39] indicates that the use of inertial sensors to estimate position suffers from the sensor noise. Another evaluation by Pang and Liu [55] shows that the accelerometer is acceptable as a short duration distance-measuring device, though a periodic calibration is required to correct the sensor drift.

Double Integration and Digital Signal Processing Techniques

Two accelerometers are used to handle the random noise in the measured acceleration signal. In Aase and Myklebust [3], the second accelerometer is placed on the floor next to the mannequin and used as the reference of external acceleration components that is subtracted from the acceleration measured on chest. The calculated acceleration signal is then processed by a high-pass filter to remove the gravity components and the CCD is estimated by double integration of the processed acceleration. The cumulative error is eliminated by the boundary-reset mechanism that resets the acceleration, velocity, and displacement to zero at the end of each compression period. However, the proposed reset mechanism may result in the loss of real sensing data and impact the accuracy of estimation. In addition, the proposed method is not designed for providing real-time feedback and require the pressure sensor to identify the chest compressions. Another work by Oh et al. [54] placed the second acceleration on the back of mannequin and apply a *de-trending* method to the integrated signal in each individual compression period to remove the cumulative error. The lowest point in a compression is regarded as the offset component and is removed from the whole signal that is measured in the period of compression. The CCD is then estimated by taking the difference between the highest and the lowest points. The de-trending process is an improved mechanism compared to Aase and Myklebust [3] and also handles the problem of overestimated CCD, which is caused by performing the chest compression on the patients lying on a soft surface. However, the misalignment of sensors may worsen the accuracy and the de-trending process needs to be preformed after each integration. In addition, the work has shown that attaching the second accelerometer on the subject's back may interrupt the chest compression and reduce the quality of the CPR.

A spectral techniques based on Fourier series decomposition (FSD) for CCD is proposed by Ruiz de Gauna et al. [61] and González-Otero et al. [23]. Drift compensation or additional sensors are not required, but only one accelerometer is used. The FSD technique assumes that all chest compressions within the operation interval are similar. Therefore, the acceleration and CCD are regarded as periodic signals and could be approximated by using sinusoidal function. Fast Fourier transform is performed in the selected interval in the acceleration signal, and the first three harmonics and their fundamental frequency are picked up and used to reconstruct the depth signals. The CCD is estimated by taking the difference between the highest and the lowest points of the reconstructed depth signal. In reality, the periodicity assumption may not hold true and thus negatively affect the validity of the reconstructed depth signal. In addition, their experimental results show the estimation is accurate for analysis intervals of a few seconds during continuous chest compressions. The feasibility and reliability of the proposed method in a real resuscitation scenario remain to be evaluated.

Kalman Filter-Based Technique

The positioning algorithms developed based on inertial sensors can be leveraged to estimate CCD. An inertial navigation algorithm based on Kalman filter [11] generates the pseudo-measurements of acceleration to remove the noise in acceleration. The pseudo-measurements are generated by recording the acceleration signal in a static state in which the inertial sensors are placed in a certain tilted angle. In reality, the assumption may not hold true universally, since the noise may vary in different times and places. Thus, the algorithm only works in the specific scenario.

Statistical Model-Based Techniques

The use of statistical models represents an alternative approach to CCD estimation. Unlike the kinematic equations-based methods [39] that suffer from the cumulative error due to the use of double integration, the statistical methods directly predict the CCD based on the features extracted from the sensor data. Lu et al. [46] proposes a statistical model to estimate the CCD using a polynomial function, two handcrafted features, compression rate, and a statistical value M. The feature M was defined as the average of acceleration during a specific time interval considering the correlation between the magnitude of acceleration and CCD. Although their results show that the mean of difference is close to zero, the confidence interval is over 1 cm.

Deep neural networks have achieved great success in extracting features and providing promising results in various applications [1]. Convolutional neural network (CNN) is a type of the deep neural networks commonly used for identifying spatial patterns for computer vision and image processing [37]. While CNN is typically adopted to deal with two-dimensional data, recent studies also demonstrate the outstanding capability to process one-dimensional time-series or sequential data such as ECG biometric recognition [2], stock price forecasting [13], and step-length estimation [33]. An advantage of using 1D CNN is the low computational cost, because a 1D convolution uses scalar multiplication and addition rather than matrix operation.

CNN has been adopted to estimate the displacement based on the features extracted from inertial sensor data. A pedestrian step model based on CNN is applied to the inertial sensor data for gait-feature extraction and step-length estimation [64]. In addition, a step-length model based on CNN achieves better results than the other methods using double integration [25]. A 1D CNN is proposed for CCD feedback [82]. Instead of directly estimating the CCD, the proposed method turns the regression problem into binary classification, and the feedback of normal and abnormal chest compression is provided to indicate if the CCD is within the recommended range. Although this work performs well in identifying most CCDs, misjudgements happen on the boundaries of recommended CCD, which are 2 and 2.4 inches. Therefore, a better solution may be to directly provide the real-value output of CCD with rescuers to assist them in performing CPR.

Table 2.1: Related Work on CCD Estimation

System/Author Name - Year	***Sensor***	***Technique***	***Device Placement***	***Positioning Algorithm***
Jin et al. - 2016 [34]	Tomography	Chest Computed Tmography (CT)	-	-
Shinnosukekun - 2016 [50]	Force and pressure sensors	-	Chest	-
Mini-VREM - 2013 [62]	Camera	Image Processing	Remote Sensing	-
Dolgov et al. - 2011 [19]	Ultrasound	-	Chest and back	Mutiply signal travel velocity by travel time
Yu et al. - 2017 [79]	UWB	-	Chest and back	Mutiply signal travel velocity by travel time
Aase et al. - 2002 [3]	Accelerometer	Double Integration	One on chest and one on floor	Reset mechanism for cumulative error
Oh et al. - 2012 [54]	Accelerometer	Double Integration	Chest and back	De-trending approach
Boussen et al. - 2016 [11]	Accelerometer	Kalman filter	Chest	The use of artificial measurement
González-Oteor et al. - 2014 [23]	Accelerometer	Kinematic equations	Chest	Fourier series decomposition
Lu et al. - 2018 [46]	Accelerometer	Statistical model	Wrist	Author defined polynomial function
Zhao et al. - 2021 [82]	Accelerometer	Deep learning	Chest	1D CNN

Chapter 3

Technical Approach

This chapter presents our technical approach to building a displacement-measuring system for estimating the depth of chest compression during CPR using inertial-sensing data. The challenge is the high precision in real-time feedback requirements. We divide the problem into compression-event detection and compression-depth estimation. A low-computational-complexity solution is proposed based on our motion model that suppresses the cumulative error in the displacement-measuring process by fusing inertial-sensing data measured by two IMUs. An information-theoretic metric called mutual information is used to evaluate the accuracy improvement by using data fusion. We compare an implementation of our approach with existing work and discuss the experimental results.

3.1 Problem Statement

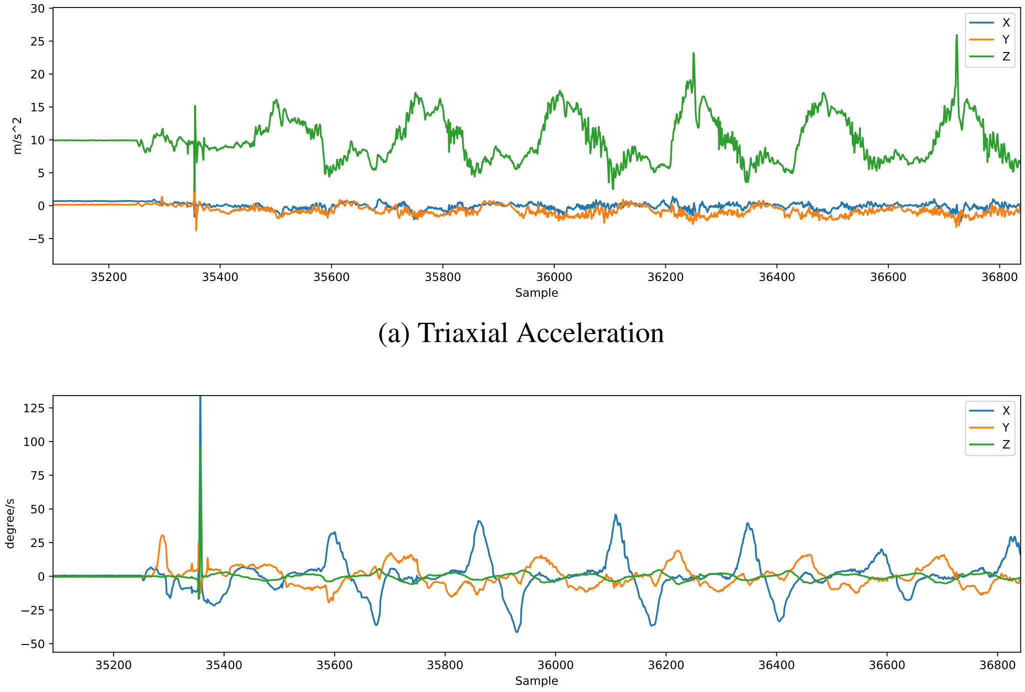
The problem to be addressed by our work is the cumulative error resulting from applying double integration method to noisy acceleration signal in the CCD estimation process. Without error handling methods, the estimated CCD is inaccurate and cannot be used to assist rescuers in performing CPR. Our solution uses two IMUs to suppress the cumulative error. More specifically, one IMU is used to measure the vertical movement of chest and the other one is placed away from the chest compression center to measure the rotational motion. The chest compression rate and chest compression depth are estimated by our proposed data-fusion algorithm and provided in real time.

3.2 IMU Data Characterization

This section presents our observation on the collected inertial sensor data recorded during chest compression. Two IMUs, each of which contains an accelerometer and a gyroscope, are placed on different locations of the chest, namely the chest center point and a spot two inches away from the compression point along the nipple line. The collected IMU data are evaluated by using mutual information to exploit the information content, and the characteristics of IMU data are explored by our proposed rotational-motion model to enable data-fusion techniques.

3.2.1 Estimation of Chest-Compression Depth using IMU

A major disadvantage of using IMUs to estimate the displacement or attitude of an object by using the kinematic equations is the cumulative error. The error is accumulated in the estimated result due to the continuous integration of the measured noisy signal with respect to time. Without removing the noise in the measured signal, the continuous integration results in the growth of linear error in velocity and angle of rotation and quadratic error in displacement. Therefore, we propose a data-fusion algorithm that uses two IMUs to suppress the noise. In our solution, the first IMU is placed on the chest-compression point, and the measurements are used by a depth-prediction model that uses kinematic equations to estimate the displacement. On the other hand, the second IMU is placed away from the chest-compression point and used as a reference to remove the noise from the predicted depth estimated by the prediction model.



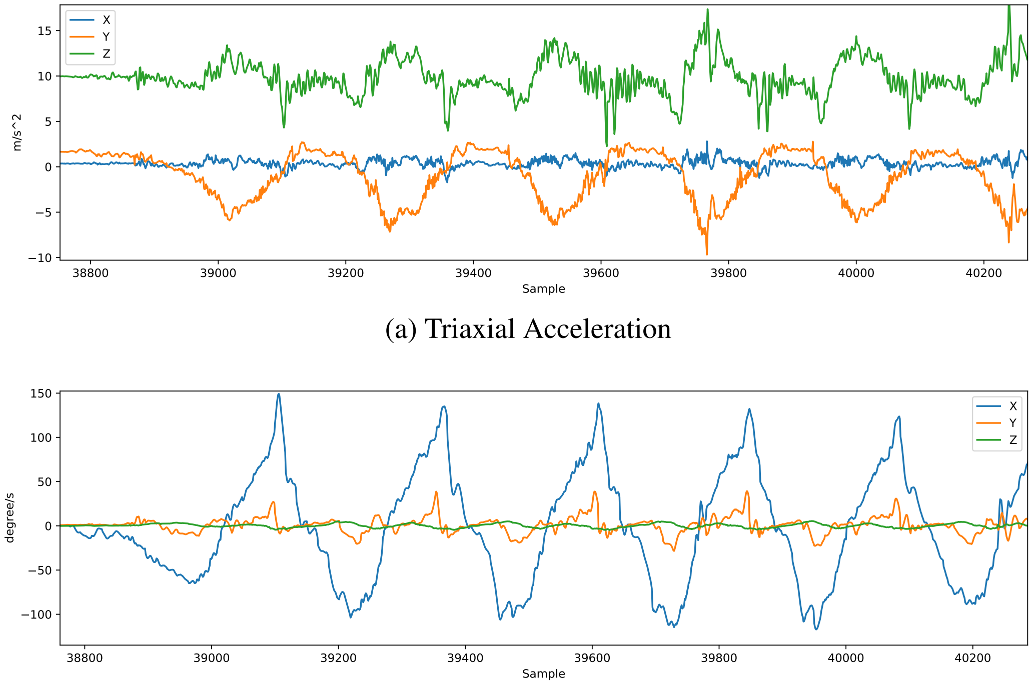
(a) Triaxial Acceleration

(b) Triaxial Angular Velocity

Figure 3.1: Inertial Sensor Data measured at the Compression Point during Chest Compressions

3.2.2 IMU Data Patterns

The acceleration and angular velocity data measured with the first IMU placed on the compression point during chest compressions are used to calculate the vertical acceleration for depth estimation. Fig. 3.1a shows the measured triaxial acceleration starting from the static state over several compressions. The non-zero triaxial offsets in the beginning are generated by gravity and should be removed to obtain the vertical acceleration generated by chest-compression force. The Z-axis acceleration in the chest-compression region shows a repetitive pattern that is a rough acceleration and deceleration trend and can be used to split the triaxial acceleration into segments, each of which represents an individual chest compression. The compression rate can be calculated by the period of chest compression, which is the product of sample period and number of samples in the segment. The angular velocity as shown in Fig. 3.1b shows the orientation of IMU keeps changing in chest compressions so that the effect of gravity on the three axes also varies. Therefore,



(a) Triaxial Acceleration

(b) Triaxial Angular Velocity

Figure 3.2: IMU Data Measured in Different Placements during Chest Compressions

the orientation of the IMU during chest compressions is required in order to extract the vertical acceleration of chest movement from the raw acceleration.

Different patterns are observed in the acceleration and angular velocity when the IMU is placed on the different location that is two inches away from the compression point. As shown in Fig. 3.2, the Y- and Z-axis acceleration, and X- and Y-axis angular velocity measurements have more regular waveforms, since the IMU placement leads to greater change of orientation, and the gravity is distributed more to the Y-axis acceleration. In addition, the kinematic equation is no longer working for compression-depth estimation by using the second IMU measurement, because the IMU was not moving vertically with the compression point. The pattern of sinusoidal-like waves in the angular velocity is a better feature to use for splitting the signal into segments compared to the use of vertical acceleration. By taking the zero-crossing points in the angular velocity, we can find the boundary between compression events and the time point of the deepest position.

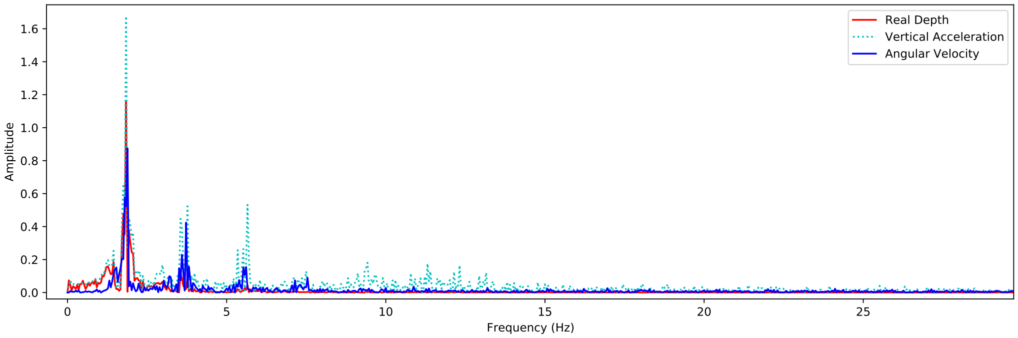
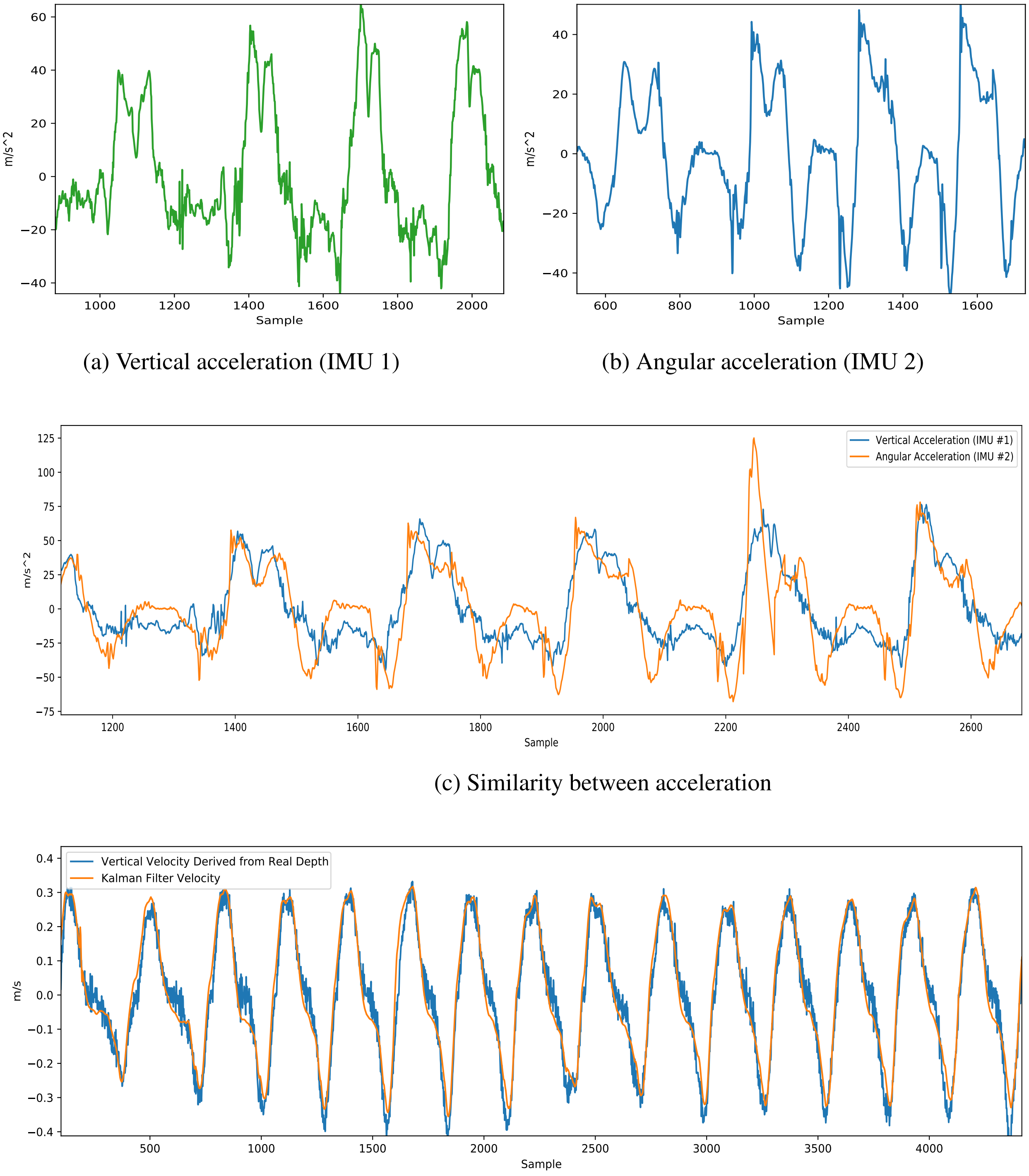


Figure 3.3: Spectral analysis of real depth, vertical acceleration, and angular velocity

The real depth and inertial-sensor data collected during chest compression can be regarded as periodic signals since the repetitive patterns are observed. We apply the FFT spectral analysis to the real depth signal, vertical acceleration from the first IMU, and X-axis angular velocity from the second IMU for comparison in frequency domain. As shown in Fig. 3.3, the real depth and the data from both inertial sensors can be approximated as a periodic signal consisting of the fundamental frequency component and the corresponding harmonic components at the same frequencies. The fundamental frequency component is the dominant frequency and carries the most energy. The similar result from spectral analysis motivates our proposed sensor-fusion algorithm by using two IMUs placed at different locations.

3.3 Rotational Motion

We propose a chest compression model that follows the rotational motion to fuse the data from two IMUs. In the rotational motion, the tangential velocity is the product of rotational radius and angular velocity as shown in Eq. (3.1). The tangential acceleration is the derivative of tangential velocity with respect to time and is calculated by multiplying the rotational radius by the angular acceleration α as shown in Eq. (3.2). We first calculate the vertical acceleration by removing the gravity from the Z-axis acceleration measured by the first IMU. We found the vertical acceleration



(a) Vertical acceleration (IMU 1)

(b) Angular acceleration (IMU 2)

(c) Similarity between acceleration

(d) Comparison between velocity derived from real depth and estimated velocity by Kalman filter

Figure 3.4: Rotational Motion

as shown in Fig. 3.4a to have high similarity with α, i.e., the derivative of the angular velocity from the second IMU as shown in Fig. 3.4b. Therefore, we assume that the tangential acceleration can be represented by the vertical acceleration and can be calculated by using the angular acceleration if the rotational radius is given. Fig. 3.4c shows the similarity between the vertical acceleration and the tangential acceleration that is the product of angular acceleration and estimated rotational radius by our algorithm, which will be provided in the Section 3.5. Our proposed algorithm uses the Kalman filter to fuse the two acceleration streams to estimate the vertical velocity. The chest-compression velocity is derived from the gold standard for the real depth, which is measured using the displacement sensor. Fig. 3.4d shows the similarity between the vertical velocity and the real velocity.

$$v_t = r\omega \tag{3.1}$$

$$a_t = \dot{v}_t = r\dot{\omega} = r\frac{d\omega}{dt} = r\alpha \tag{3.2}$$

3.4 Information-theoretic Metric for Sensor Data Evaluation

To better evaluate the effect of different sensor placements on sensor data as shown in Section 3.2, an information-theoretic metric, called mutual information, is used to quantify the correlation between the real CCD, which is measured from displacement sensor, and the sensor data collected from IMU. In addition, to validate our rotational model proposed in Section 3.3, the dependency between the real CCD and vertical velocity or depth estimated by our proposed rotational model are also discussed in the remaining section.

The IMU-based chest-compression sensing system can be represented as a simple communication system as shown in Fig. 3.5. In this representation, the real CCD data is generated by successive chest compressions and processed through an encoder for data transmission. Different locations on chest for IMU placement can be regarded as noisy communication channels that are used to

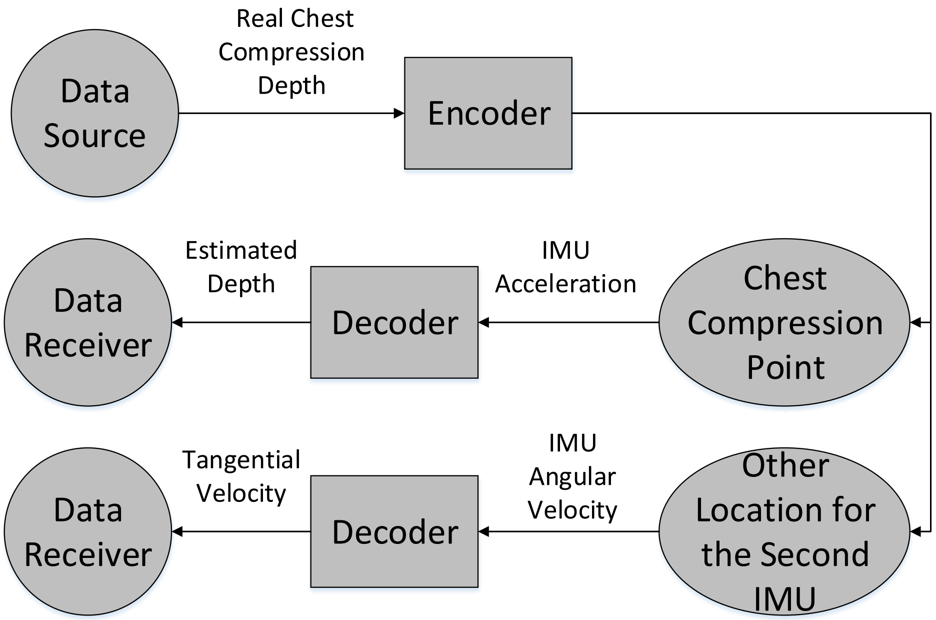


Figure 3.5: Chest compression represented by a communication system

transmit the CCD data to the receivers, which are the two IMUs used in our system. The received data, including both measured acceleration and angular velocity, is processed through decoders, which are the distance-measuring techniques, to estimate the CCD.

Suppose that a discrete random variable X is defined for chest compression and that the depth of a chest compression is an event drawn from a probability distribution function $p(x)$. The *information of an event*, which is regarded as uncertainty, is then defined by Eq. (3.3), and the *entropy of the random variable* X, which quantifies the average uncertainty, is the expectation of possible outcomes of the random variable defined by Eq. (3.4). Likewise, the entropy of random variable A and that of G are defined by Eq. (3.5) and Eq. (3.6) for acceleration and angular velocity, respectively.

The *conditional entropy* quantifies the uncertainty of random variable X given that the information of random variable A is provided. The joint and conditional probability of the random variables are used to calculate the entropy of X conditional on A and G as defined by Eq. (3.7) and Eq. (3.8), respectively. The larger the conditional entropy, the more information the random variable provides to describe the outcomes of chest compression. Thus, the mutual information as defined in Eq. (3.9) is interpreted as how much uncertainty is removed from random variable X after obtaining the information of random variable A.

$$h(x) = -\log_2 p(x) \tag{3.3}$$

$$H(X) = -\sum_{x\in X} p(x)\log_2 p(x) \tag{3.4}$$

$$H(A) = -\sum_{a\in A} p(a)\log_2 p(a) \tag{3.5}$$

$$H(G) = -\sum_{g\in G} p(g)\log_2 p(g) \tag{3.6}$$

$$H(X|A) = -\sum_{a\in A}\sum_{x\in X} p(x,a)\log_2 p(x|a) \tag{3.7}$$

$$H(X|G) = -\sum_{g\in G}\sum_{x\in X} p(x,g)\log_2 p(x|g) \tag{3.8}$$

$$I(X;A) = H(X) - H(X|A) \tag{3.9}$$

$$I(X;A) \geq I(X;D), \quad D = f_d(A) \tag{3.10}$$

$$I(X;T) \geq I(V;T) = H(V) - H(V|T), \quad V = f_v(X), T = f_t(A,G) \tag{3.11}$$

The data processing inequality in Eq. (3.10) states that the information content cannot be increased by any post-processing techniques. Let D be the discrete random variable representing the CCD estimated by using IMU acceleration. By the data processing inequality, the estimated CCD will not provide more information about the real chest compression depth than the IMU acceleration. Thus, the mutual information between the real depth and raw acceleration provides the upper bound to evaluate the accuracy of the data processing techniques by comparing the mutual information between real and estimated depth with the upper bound. In addition, by placing the second IMU at different locations and calculating the mutual information $I(X;G)$, the location where the highest mutual information is calculated can be chosen as the best to place the IMU.

Our proposed algorithm can be evaluated by mutual information of the compression velocity, which is derived from CCD, and the velocity estimated by our algorithm. According to Eq. (3.11),

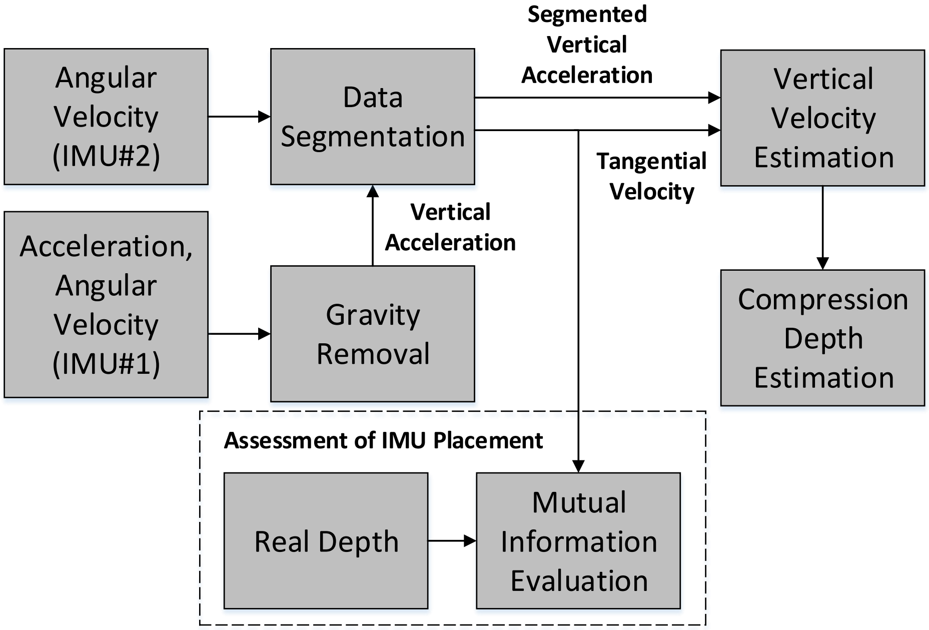

Angular Velocity (IMU#2)
Data Segmentation
Segmented Vertical Acceleration
Vertical Velocity Estimation
Tangential Velocity
Vertical Acceleration
Acceleration, Angular Velocity (IMU#1)
Gravity Removal
Compression Depth Estimation
Assessment of IMU Placement
Real Depth
Mutual Information Evaluation

Figure 3.6: Flow for Depth Estimation

the upper bound of our algorithm is constrained by $I(X,T)$ and the conditional entropy $H(V|T)$ indicates the dependency between the compression velocity and estimated velocity. $H(V|T)$ is used to validate the feasibility of our proposed rotational model. In addition, $I(X;T)$ can be used as the metric to compare estimated accuracy among different algorithms. The experimental results on the IMU placement and the comparison between data-processing algorithms are given in Section 4.2.1.

3.5 Framework

This section presents our framework for estimating chest-compression depth. It is composed of gravity removal, data segmentation, vertical velocity estimation, and CCD estimation. The structure of this framework is shown in Fig. 3.6.

3.5.1 Gravity Removal from Acceleration

The gravity-removal stage uses the estimated IMU orientation to remove the gravity component in acceleration. Our framework considers the scenario where the patient lies on a flat surface and assumes the earth coordinates. Therefore, when the IMU is tilted, the surface coordinates deviates from the IMU coordinates, since the gravity is distributed to all three axes. To remove the gravity from the vertical acceleration, which is perpendicular to the patient's chest, the orientation is needed to transform the IMU measurement from IMU coordinates to earth coordinates.

The Madgwick filter is a quaternion-based sensor-fusion algorithm that fuses IMU data to estimate the orientation [48]. A rotation quaternion for three-dimensional space is a four-dimensional complex number and can represent any rotation or sequence of rotations by using an Euler axis $\vec{u}$ and an angle β as shown in Equation (3.12).

$$\begin{aligned} q &= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \\ &= \begin{bmatrix} \cos\frac{\beta}{2} & u_x \sin\frac{\beta}{2} & u_y \sin\frac{\beta}{2} & u_z \sin\frac{\beta}{2} \end{bmatrix} \end{aligned} \tag{3.12}$$

The orientation of a static object in three dimensional space can be regarded as the result of successive rotations around corresponding coordinate axes. For example, the extrinsic rotations define the sequence of rotations using Euler angles ϕ, θ, ψ with respect to x, y, and z axes, respectively. The Euler angles can be found from the rotation quaternion using Equation (3.13) and are used to define a rotation matrix. The acceleration with respect to the earth coordinates, a_{rawearth}, can be calculated by multiplying the IMU acceleration by the rotation matrix and the a_{mov}, which is the acceleration generated by the force of chest compression, is extracted from a_{rawearth} by removing the gravity. Our model uses ϕ and θ and sets ψ to zero to generate the rotation matrix, since the Z axis is parallel to gravity.

$$
\begin{aligned}
\phi &= \arctan\left(\frac{2q_3q_4 - 2q_1q_2}{2q_1^2 + 2q_4^2 - 1}\right) \\
\theta &= -\arcsin\left(2q_2q_4 + 2q_1q_3\right) \\
\psi &= \arctan\left(\frac{2q_2q_3 - 2q_1q_4}{2q_1^2 + 2q_2^2 - 1}\right)
\end{aligned}
\tag{3.13}
$$

3.5.2 Data Segmentation and Chest Compression Rate

The IMU data is segmented by chest compression, and the chest compression rate can be directly calculated by the inverse of the segmented chest compression interval. The segmentation algorithm is developed based on the local peak detection algorithm. When the chest is compressed to the deepest point, the displacement or the tilt angle of IMU is the local maximum within the period of the chest compression. The local minimum between two successive local maxima is the boundary between chest compressions, that is, where the current chest compression ends and the next one starts. Each successive pair of local-minimum points locates a chest compression in the IMU data.

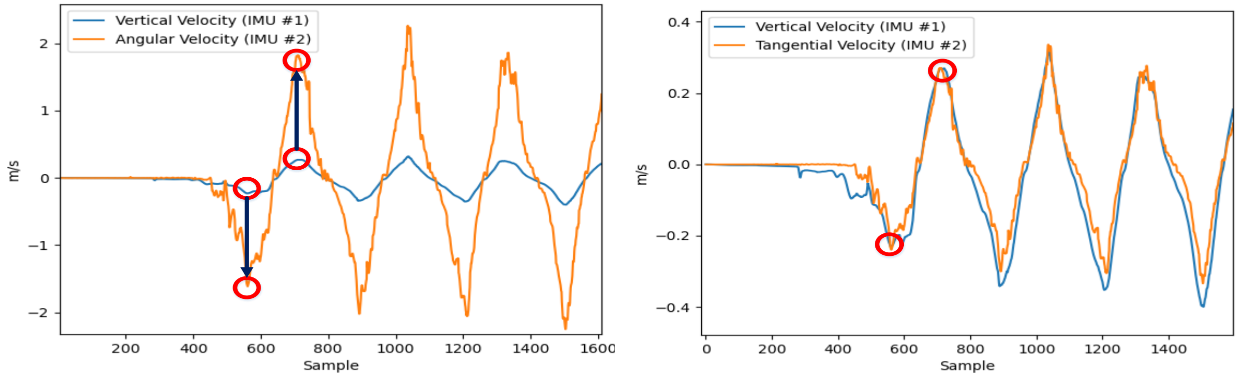
Our framework develops an online algorithm for IMU data segmentation by using the angular velocity measured from the second IMU. As described in Section 3.2.2, the zero-crossing points in angular velocity can be used to segment the chest compressions. However, we found the pattern of fluctuation around zero in the data due to the very short amount time in static state when the chest is at the deepest point or back to the originally decompressed position. Therefore, we transform the problem of finding zero-crossing points in angular velocity into the peak detection problem in the estimated angle of rotation. The vantage of using the peak detection solution is that the peak points and the valley points represent the deepest compression point and the boundary of chest compressions respectively so that the extra pattern recognition for classifying zero-crossing points into deepest point or boundary between compression point is not needed.

The segmentation process in our algorithm is designed to be activated when chest compression is performed. When the change of angular velocity is over the predefined threshold, the last zero-crossing point is annotated as the beginning of a chest compression, and the peak-detection algorithm is applied to the sliding window, which stores the measured data of a certain period, to find the local maximum. The AHA guidelines suggests a chest-compression rate of 100 to 120 per minute, so that the size of the sliding window in our algorithm is set to 250, which is half of the sampling rate. When the local maximum is found, the sliding window is relocated and starts from the local-maximum data point, and the segmentation process applies the valley-detection algorithm to find the local minimum. The segmentation process keeps running until chest compression stops, when the IMU remains static and observes minor fluctuation of angular velocity around zero. In addition, local maximum found after the pause of chest compression is also close to zero. Therefore, the termination condition is satisfied when the difference between the local minimum and local maximum is smaller than the designed termination threshold. The segmentation process restarts upon meeting the activation condition, that is, the change of angular velocity exceeds the predefined threshold.

Algorithm 1: Real-time Data Segmentation and Compression Rate Estimation

```
Input: Angular Velocity $\overrightarrow{Gyr}$, Acceleration $\overrightarrow{Acc}$, Sample Rate $Rate$
Output: Compression event parameters $\overrightarrow{cpParam}$
Function PEAKVALLEYDETECTION($\overrightarrow{Gyr}, \overrightarrow{Acc}, Rate$):
    $Diff \leftarrow$ CHANGEOFDATA($\overrightarrow{Gyr}$)
    $T_{Start} \leftarrow$ FINDLASTZEROCROSSING($\overrightarrow{Gyr}$)
    while $Diff > ActivationThreshold$ do
        $\overrightarrow{WinBuf} \leftarrow$ ANGLEFROMMADGWICKFILTER($\overrightarrow{Gyr}, \overrightarrow{Acc}, T_{Start}, Rate$)
        $T_{Maximum}, Peak \leftarrow$ PEAKDETECTION($\overrightarrow{WinBuf}, \overrightarrow{Gyr}$)
        $\overrightarrow{WinBuf} \leftarrow$ RELOCBUF($\overrightarrow{Gyr}, \overrightarrow{Acc}, T_{Maximum}, Samp$)
        $T_{Minimum}, Valley \leftarrow$ VALLEYDETECTION($\overrightarrow{WinBuf}, \overrightarrow{Gyr}$)
        $\overrightarrow{cpParam} \leftarrow$ APPENDCPRPARAM($T_{Start}, T_{Maximum}, T_{Minimum}$)
        $cpm \leftarrow (T_{Minimum} - T_{Start}) \div Rate \times 60$
        $T_{Start} \leftarrow T_{Minimum}$
        $Diff \leftarrow Peak - Valley$
    return $\overrightarrow{cpParam}$
```

Algorithm 1 provides the pseudo code to segment the data by chest compressions and to calculate



(a) Z-axis Velocity and Angular Velocity

(b) Z-axis Velocity and Tangential Velocity

Figure 3.7: Z-axis Velocity and Tangential Velocity

the chest-compression rate in real-time. When the change of angular velocity is over the pre-defined threshold, the algorithm is activated, and the timestamp of the last zero-crossing point in angular velocity is annotated as the start of chest compression. The Madgwick filter is used to estimate the rotational angles, and the angles are kept in the sliding window. The peak-detection method is applied to the sliding window until the local maximum is found. The sliding window is relocated to the timestamp of the found local maximum, and the valley-detection algorithm is applied for the local minimum. The local minimum is at the end of current chest compression and is used as the start of the next chest compression for the sliding window. The three time-stamps are used to locate the chest compression and to calculate the chest-compression rate. If the difference between the maximum and minimum is smaller than the threshold, then the segmentation algorithm is stopped until the next chest compression is detected.

3.5.3 Vertical Velocity Estimation

The rotational radius is first calculated by matching the angular velocity with the vertical velocity in order to obtain the tangential velocity. The velocity estimated by using the Z-axis acceleration from the first IMU in the first chest compression is chosen to represent the tangential velocity.

We assume the estimated velocity has minimum drift error, and an empirical scaling factor can be obtained by matching the amplitude of tangential velocity with the one of angular velocities, as shown in Fig. 3.7a. The scaling factor is then applied to the following angular velocity signal to compute the tangential velocity as shown in Fig. 3.7b.

The states, state-transition model, observation model, input, and observation matrices are defined as shown in Eq. (3.14) for parameters used in Kalman filter. The state vector X is composed of position and velocity. The prediction matrix A and the control vector B follow the kinematic equations to predict position and velocity by using the vertical acceleration u as input. The vector H is the observation model that maps the state space to the measurement space, and the z is the tangential velocity, which is the product of rotational radius and angular velocity.

Definition of Matrices:

$$\begin{aligned} x &= \begin{bmatrix} P \\ V \end{bmatrix}, \quad A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \quad H = [0 \quad 1] \\ u &= \begin{bmatrix} Vertical\ Acceleration \end{bmatrix} \\ z &= \begin{bmatrix} R \times Angular\ Velocity \end{bmatrix} \end{aligned} \tag{3.14}$$

Predict:

$$\begin{aligned} \hat{x}_{t|t-1} &= A\hat{x}_{t-1|t-1} + Bu_t \\ P_{t|t-1} &= AP_{t-1|t-1}A^T + Q \end{aligned} \tag{3.15}$$

Update:

$$\begin{aligned} y_t &= z_t - H\hat{x}_{t|t-1} \\ S_t &= HP_{t|t-1}H^T + R \\ K_t &= P_{t|t-1}H^T S_t^{-1} \\ \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t y_t \\ P_{t|t} &= (I - K_t H)\, P_{t|t-1} \end{aligned} \tag{3.16}$$

The *adaptive Kalman filter* is used to fuse the data from two IMUs to estimate the vertical velocity. A conventional Kalman filter can be implemented based on Equations (3.15) and (3.16). An online estimation approach [73] is adopted to adaptively determine the covariance matrix of the measurement noise R. The measurement filtering residuals V_{z_t} is calculated using Eq. (3.17):

$$\begin{aligned} V_{z_t} &= Z_t - H\hat{x}_{t|t} \\ &= Z_t - H(\hat{x}_{t|t-1} + K_t y_t) \\ &= (E - HK_t)y_t \end{aligned} \tag{3.17}$$

$$Q_{z_t} = R_t - HP_{t|t}H^T \tag{3.18}$$

By applying the error propagation law to Eq. (3.17), the covariance matrix Q_{z_t} is computed as shown in Eq. (3.18) and the covariance matrix R is then calculated using Eq. (3.19).

$$\begin{aligned} R_t &= Q_{z_t} + HP_{t|t}H^T \\ &= \frac{1}{m}\sum_{i=0}^{m-1} V_{z_{t-1}} V_{z_{t-1}}^T + HP_{t|t}H^T \end{aligned} \tag{3.19}$$

In Eq. (3.19), the m is the number of samples used to calculate the residuals V_{z_t}. Extra computation is required for V_{z_t} and $HP_{t|t}H^T$ since neither is generated in the process of conventional Kalman filter.

```
Algorithm 2: Chest Compression Depth Estimation
Input: Angular Velocity $\overrightarrow{Gyr}$, Acceleration $\overrightarrow{Acc}$, Compression event parameters $cpParam$
Output: Chest Compression Depth $CCD$
Function DepthEstimation($\overrightarrow{Gyr}, \overrightarrow{Acc}, cpParam$):
    $\overrightarrow{verticalAcc} \leftarrow$ RemoveGravity($\overrightarrow{Gyr}, \overrightarrow{Acc}$)
    $\overrightarrow{segment} \leftarrow$ DataSegmentation($\overrightarrow{Gyr}, \overrightarrow{verticalAcc}, cpParam$)
    $rotationalRadius \leftarrow$ estimaterotationalRadius($\overrightarrow{verticalAcc}, \overrightarrow{Gyr}$)
    $\overrightarrow{verticalVelocity} \leftarrow$ AdaptiveKalmanFilter($\overrightarrow{segment}, rotationalRadius$)
    $\overrightarrow{CCD} \leftarrow$ ComplementaryFilter($\overrightarrow{verticalVelocity}, \overrightarrow{Acc}$)
    return $\overrightarrow{CCD}$
```

3.5.4 Estimating Chest-Compression Depth

Our proposed method for CCD estimation assumes the depth can be approximated by the tilt angle of the IMU. The CCD can be calculated by using the vertical velocity from Kalman filter and the kinematic equation in Eq. (3.20). However, as discussed earlier, this approach results in the linear growth of cumulative error in the estimated displacement. To handle the error, we transform the depth-estimation problem into an orientation-estimation problem. The vertical velocity can be regarded as scaling angular velocity in Eq. (3.21) and the depth estimation equal angle estimation in Eq. (3.23). To estimate the tilt angle, the acceleration measured from the second IMU is used. In our IMU placement, we assume the pitch angle is required to represent the tile angle, which can be calculated by using Eq. (3.24).

$$CCD_{\text{t}} = CCD_{\text{t-1}} + V_{\text{KF}} T_{\text{sampleperiod}} \tag{3.20}$$

$$\sim CCD_{\text{t-1}} + r\,\omega_{\text{t}} T_{\text{sampleperiod}} \tag{3.21}$$

$$= r\,Angle^{\text{kf}}_{\text{t-1}} + r\,\omega_{\text{t}} T_{\text{sampleperiod}} \tag{3.22}$$

$$= r\,Angle^{\text{kf}}_{\text{t}} \tag{3.23}$$

$$Angle^{\text{acc}}_{\text{t}} = \arctan \frac{Acc_{\text{y}}}{Acc_{\text{z}}} \tag{3.24}$$

A complementary filter is used to fuse the calculated angle to constrain the growth of error. In the short period, the estimation using the kinematic equation is accurate but suffers from the cumulative error. On the other hand, the angle calculated from acceleration does not drift and provides very accurate angle estimation from the gravity component as long as the IMU is in static state. Complementary filter is a simple and typical sensor fusion technique that consists of a low-pass and a high-pass filter. To handle the cumulative error, the angle calculated from acceleration is processed through high-pass filter and the low-pass filter is applied to the angle estimated from gyroscope measurement. As shown in Eq. (3.25), we choose 0.993 and 0.007 for the α and β

respectively to estimate the CCD.

$$
\begin{aligned}
CCD_{\mathrm{t}} &= r\,Angle_{\mathrm{t}} \\
&= r\,\alpha\,Angle_{\mathrm{t}}^{\mathrm{kf}} + r\,\beta\,Angle_{\mathrm{t}}^{\mathrm{acc}} \\
\alpha + \beta &= 1
\end{aligned}
\tag{3.25}
$$

Algorithm 2 provides the pseudo code for CCD estimation. The algorithm first calculates the vertical acceleration by removing the gravity from the acceleration data measured by the first IMU. With the chest compression parameters, the inertial data is segmented by chest compression. The vertical velocity is estimated by segments, and the depths are calculated as scaled angle by using the complementary filter that fuses the vertical velocity and the calculated tilted angle obtained from acceleration.

3.6 Summary

In this chapter, we observed different patterns in sensor data when IMUs are placed at different location. The observed patterns motivate our rotational model and two algorithms for estimating vertical velocity and CCD by using two IMUs. Our proposed algorithms work in real time and are low computational cost. In addition, an informational-theory metric, call mutual information, is used to validate our proposed rotational model. The experimental results will be provided in the next chapter.

Chapter 4

Evaluation

This chapter presents and evaluation of our CCD estimation algorithm. We first describe our experimental setup and the test scenario, followed by a comparison of sensor placement for accuracy and complexity.

4.1 Experimental Setup

We collected chest-compression data using a 3rd-generation prototype of a medical device called Bluebox (BB). BB is designed as a data logger to record vital signs such as body and ambient temperature, 1-lead ECG, breathing and ambient sounds, and blood-oxygen concentration (SpO2) in a code blue event. We use only the data from the 9-DoF IMU on BB for the purpose of evaluating our CCD algorithm. We program the BB firmware to store the sensor data in the internal flash storage during CPR, and the data can be offloaded through USB interface afterwards.

A mannequin for CPR training as shown in Fig. 4.1 is used for the simulation of chest compression. To measure chest movement during CPR, two BBs are placed at two different locations, one on the compression point and the other 2 inches away. A displacement sensor is attached to the chest of

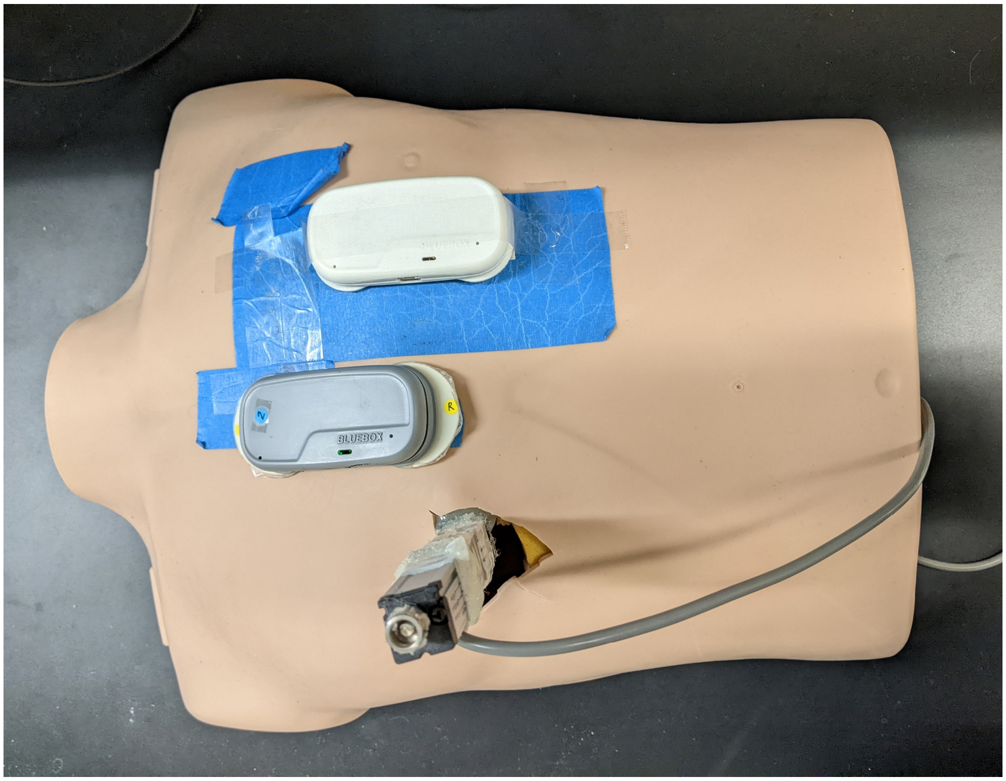


Figure 4.1: Experimental Setup

mannequin to generate the ground truth for evaluating the estimated accuracy.

Two test scenarios are performed to evaluate the accuracy of the chest compression model and the proposed real-time algorithm. In the first test scenario, we tried placing the second BB at different locations of chest: 1.5, 2, 3, and 4 inches away from the compression point. The results are used to evaluate the impact of different displacement on the rotational motion model. In the second test scenario, the second BB is placed on the location 2 inches away from the compression point to evaluate the accuracy. In both test scenarios, the first BB is always placed on the compression point for acceleration measurement.

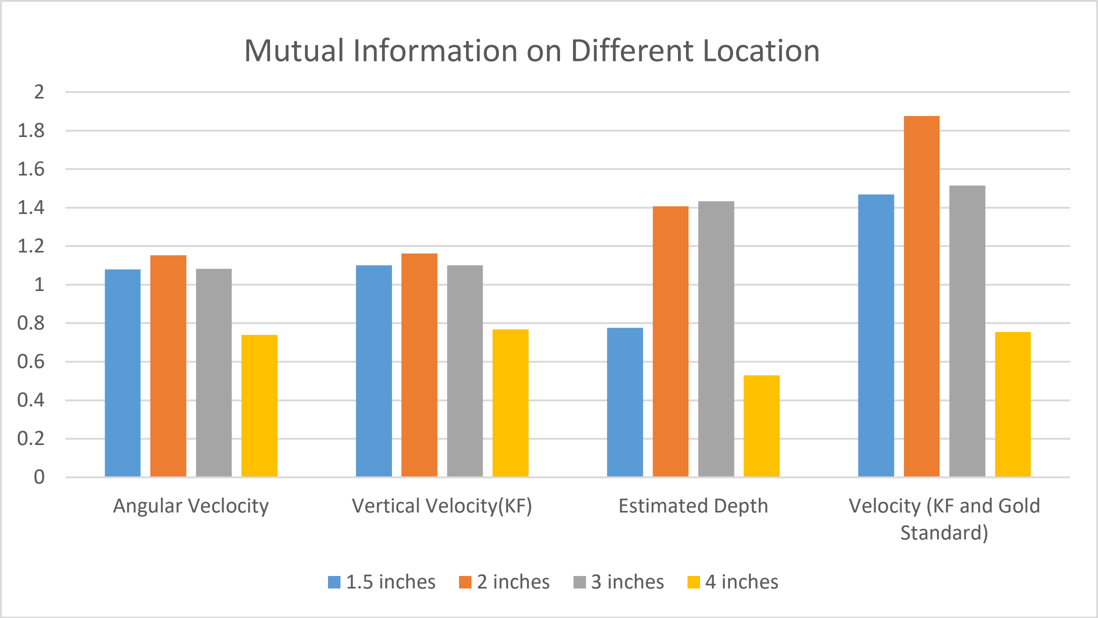


Figure 4.2: Placement Evaluation using Mutual Information

4.2 Experimental Results

The section evaluates the impact of sensor placement on estimated accuracy. Our proposed framework first uses mutual information to demonstrate the best location selected for vertical velocity estimation. The estimation accuracy is compared between our proposed method and the Fourier series decomposition (FSD) method [61, 23]. The complexity of the algorithms are also studied to demonstrate the feasibility of real-time feedback.

4.2.1 Evaluation of Second IMU Placement

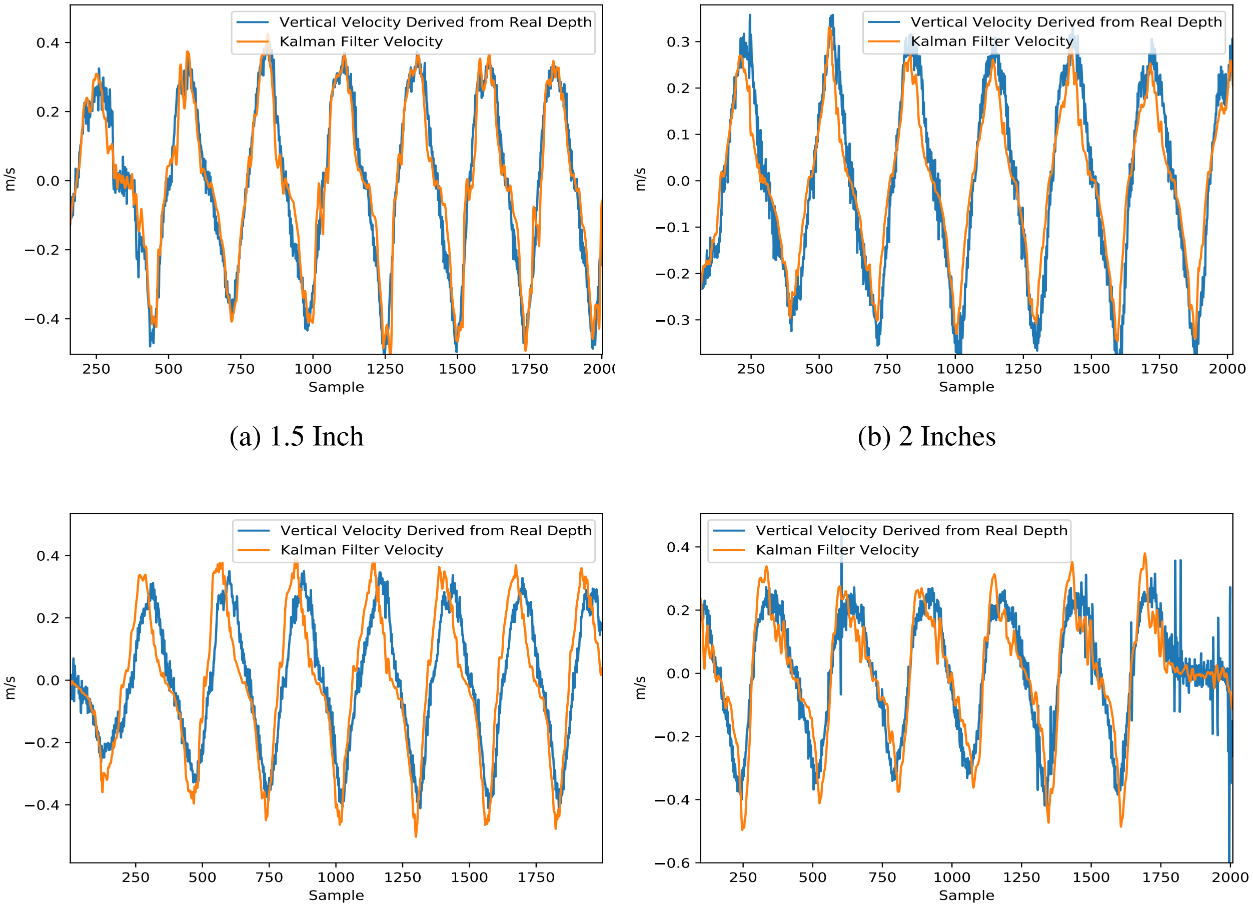
The mutual information between the real depth and the signals measured by the sensor placed at four different locations are provided in Table 4.1. The selected signals include angular velocity, vertical velocity estimated by KF, and estimated depth. In addition, the mutual information of the

Table 4.1: Mutual Information for Sensor Placement Evaluation

Placement Evaluation using Mutual Information				
Distance (Inch)	1.5	2	3	4
Angular Veclocity	1.079	1.153	1.083	0.739
Vertical Velocity(KF)	1.101	1.162	1.100	0.769
Estimated Depth	0.775	1.407	1.434	0.530
Velocity (KF and Gold Standard)	1.469	1.876	1.515	0.754

vertical velocity and the velocity derived from the gold standard is also calculated. The results are visualized as shown in Fig. 4.2. First, from the results of angular velocity, we can find that placing the sensor 2 inches away from the chest compression point achieves the highest mutual information. The fourth location results in the lowest mutual information, because because it is at the outskirt of the compression region and sensor rotates within a smaller range. The mutual information of estimated vertical velocity is approximately the same as the mutual information of angular velocity. This is because the measurement model of Kalman filter is based on rotational motion. The angular velocity is used to generate the tangential velocity that is used to constrain the cumulative error in the vetical velocity estimaion. The mutual information between estimated vertical velocity and the real velocity derived from the gold standard also shows the second location to be the best among the four locations to estimate the vertical velocity. Fig. 4.3 compares the estimated velocity and real velocity. Our sensor-fusion algorithm is able to suppress the cumulative error in vertical velocity estimation. However, in Fig. 4.3d, the waveform of the estimated velocity is less smooth and much more noisy comparing to the other three.

Although the first three locations can provide high-quality vertical velocity estimation, the mutual information between CCD and vertical velocity estimated at the second and third locations are much higher than that at the first location. However, high mutual information does not imply high accuracy. Our depth-estimation algorithm assumes the device placement location is flat so that the product of rotational radius and angle of rotation can be regarded as the displacement. The depth estimation of the first location is shown in Fig. 4.5. Since the complementary filter uses the



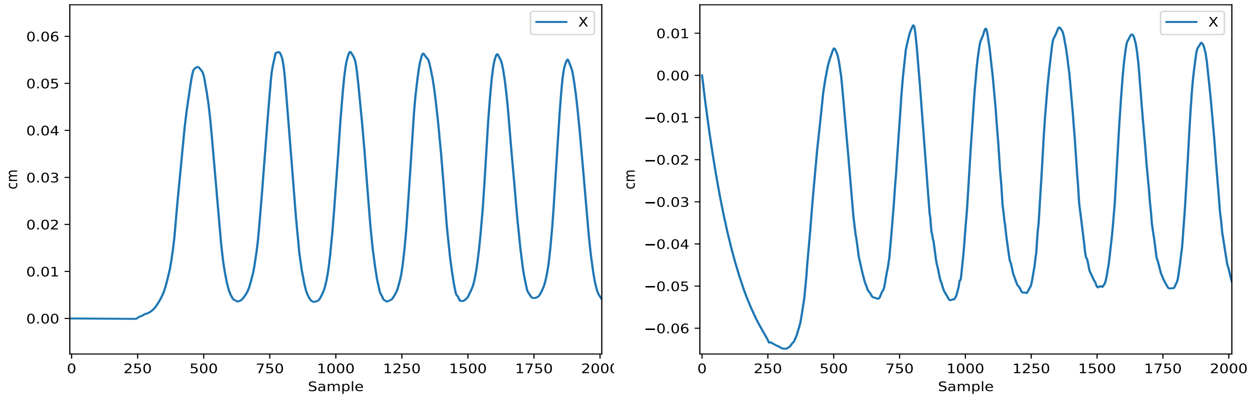
(a) 1.5 Inch

(b) 2 Inches

(c) 3 Inches

(d) 4 Inch

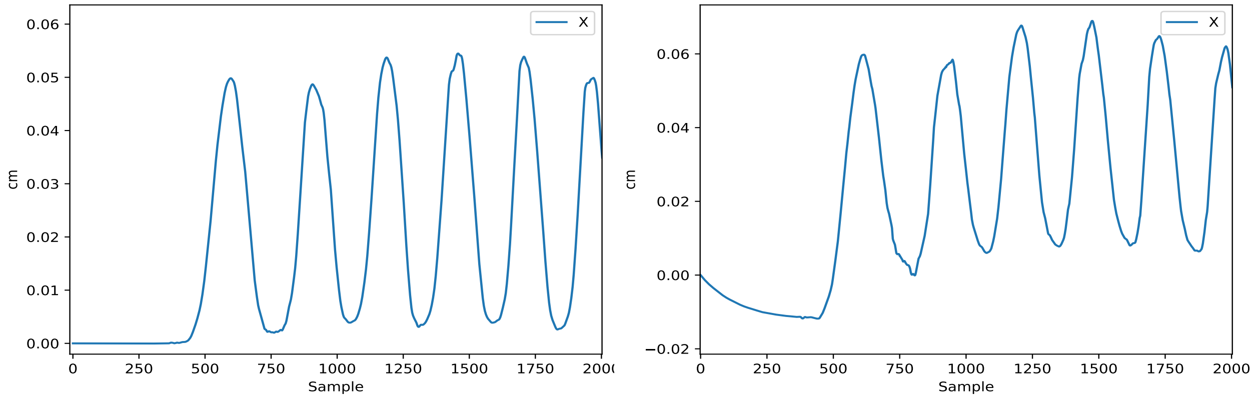
Figure 4.3: Comparison between KF Velocity and Velocity derived from Gold standard



(a) Estimated Depth without Complementary Filter

(b) Estimated Depth with Complementary Filter

Figure 4.4: Estimated Depth on Third location



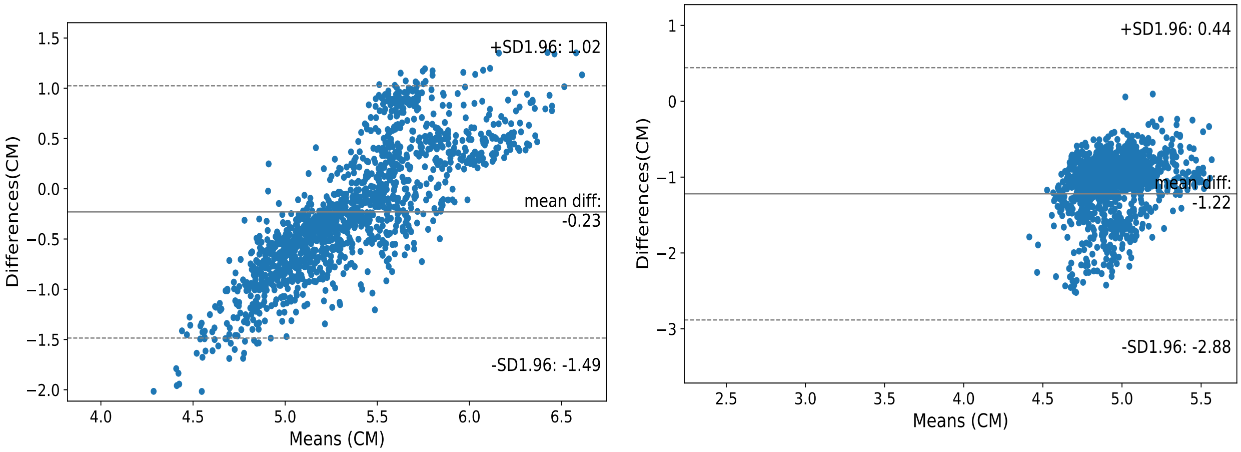
(a) Estimated Depth without Complementary Filter (b) Estimated Depth with Complementary Filter

Figure 4.5: Estimated Depth on First Location

angle of rotation to handle the cumulative error in depth estimation, the uneven surface results in a negative displacement added into the initial position as shown in Fig. 4.5b. Compare to the results in Fig. 4.5a without using complementary filter, the CCD is overestimated by around 2cm with using complementary filter. As shown in Fig. 4.4b, the initial tilt angle on the third location also generates a negative displacement of around 6.5 cm in compression depth. In addition, the tilt angle is calculated by Eq. (3.24) using acceleration signals. The acceleration may be interfered with chest compression force so that the value of mutual information on the first location is decreased.

4.2.2 Accuracy Comparison

Fig. 4.6 shows the accuracy comparison between our proposed method and FSD method over eleven sessions, which produced 1363 CCD records. Our proposed method is able to provide more accurate estimation with a mean of difference of 0.23 cm, compared to 1.22 cm using the FSD method. In addition, the range of 95% limits of agreement $(-2.88, 0.44)$ in FSD method is also wider than our method. The results using the FSD method are shown in Fig. 4.6b, as the differences between estimation and gold standard are aggregated in the center of mean difference region. This is because the FSD method assumes the signal is periodic, and the Fourier-series

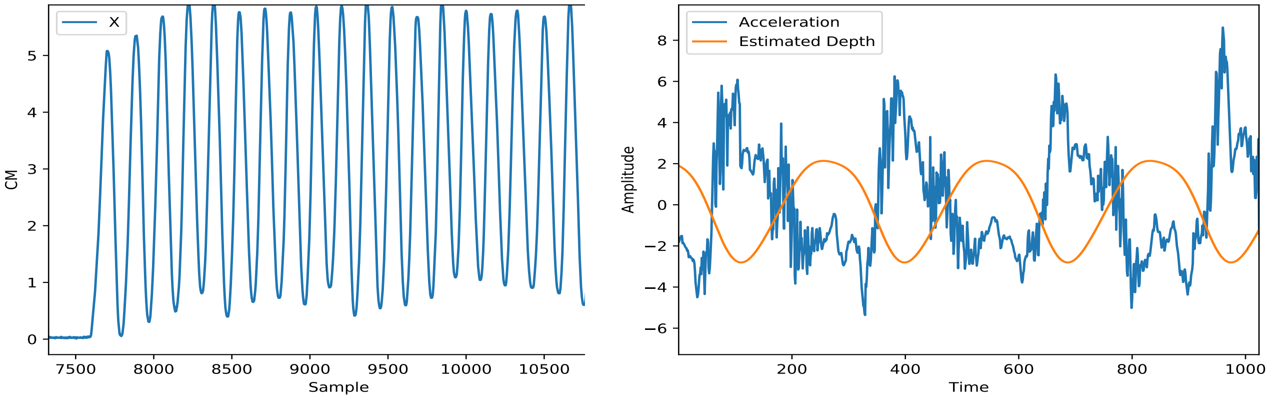


(a) Accuracy of Our Proposed Method

(b) Accuracy of Fourier Series Decomposition Method

Figure 4.6: Bland-Altman Plot of the Mean Difference

representation is unable to handle the DC component. Thus, the CCD is estimated by the difference between the highest and lowest data points in the interval of a chest compression as shown in Fig. 4.7b. However, in the decompression stage, the chest may not return to the original position, and the incomplete chest compression leads to an offset between the original position and the decompression position as shown in Fig. 4.7a. The FSD method is unable to find the offset, so that the estimation is inaccurate. Our proposed method estimates the depth by complementary filter that directly estimates the depth rather than taking the difference between the peak and valley points. Thus, it is more accurate without losing the offset component. However, an oblique ellipse is shown in Fig. 4.6a from our estimated results. We observe that this results from the inconsistent rotational radius calculated by our amplitude-matching method. Our method uses the difference between the peak and valley points to find the scale factor, but it is not guaranteed as the best real value to match the two signals. The rotational radius has high impact on the accuracy of our vertical velocity estimation. Inaccurate radius estimation results in the deviation in the successive velocity and depth estimation. The smaller and larger estimated rotational radius result in lower and higher depth estimations, respectively, such that the oblique ellipse shape is observed.



(a) Incomplete Decompression

(b) Fourier Series Decomposition

Figure 4.7: Imperfect Chest Compression

4.2.3 Computation complexity comparison

Our proposed framework is light-weight and real-time. We utilize the rotational motion and complementary filter to estimate the depth. Since our observation is assumed to be the tangential velocity, the Kalman filter used in vertical velocity estimation does not require to compute the position so that our depth estimation framework can be further simplified as scalar operations. On the other hand, the other proposed methods such as the FSD method use Fourier series representation and the computationally expensive FFT operation is required. The other methods use band-pass filters to remove the cumulative error, but the filtering cost is still expensive.

4.3 Discussion

In this section, we discuss the limitations of our proposed method and the possible optimizations for improvement of system accuracy and robustness.

Rotational Radius

The rotational radius is the key to enabling the sensor-fusion algorithm. Although in the accuracy experiments we placed our sensor on a flat surface of the chest, from the placement evaluation experiments, we found the sensor-fusion algorithm to able to provide accurate estimation for vertical velocity as long as the sensor is in rotational motion during the chest compression process. This finding suggests that the behavior of rotational motion in the chest compression scenario is two-dimensional, and that the rotational model can be applied if the sensor is placed in the compressed region of the chest. Typically, in an emergency, the doctors have very limited time to set up the equipment to perform resuscitation. A more thorough study on the compression region can improve the convenience of the BB device. At the same time, a more robust rotational radius-estimation method is definitely required to provide accurate velocity estimation.

Tilt-Angle Compensation

Our experimental results have shown that compensation is required to account for the uneven surface for sensor placement. We adopted the complementary filter for its simplicity. To deal with the orientation problem, one possible solution is the attitude initialization in sensor system before the start of chest compression. A more sophisticated orientation-estimation technique such as the Madgwick filter, which we used to remove the gravity, can be adopted to estimate the rotation angle for the purpose of tilt-angle compensation.

Part II

Encounter-Based Collaborative Indoor Localization

Chapter 5

Background and Related Work

Positioning and localization are two of the key enablers of location-based services and applications. A variety of techniques based on different sensing modalities have been proposed. This chapter first provides a background on the stages of the positioning and localization process and categorizes the commonly used techniques. After a summary of indoor localization, we survey related works that serve as baseline of comparison with our work.

5.1 Overview of Positioning and Localization

Object positioning and localization by our definition are shown in Fig. 5.1. Object localization is the process that compares the current estimated position with the spatial information obtained from prior investigation of environmental space in order to determine the target location. The object localization process can be divided into three stages: data processing, positioning, and map-matching. For example, the received Global Positioning System (GPS) signals are used to measure the distances between the receiver and the satellites, and the target position can be estimated with respect to the positions of satellites. The target location can be determined by matching the target

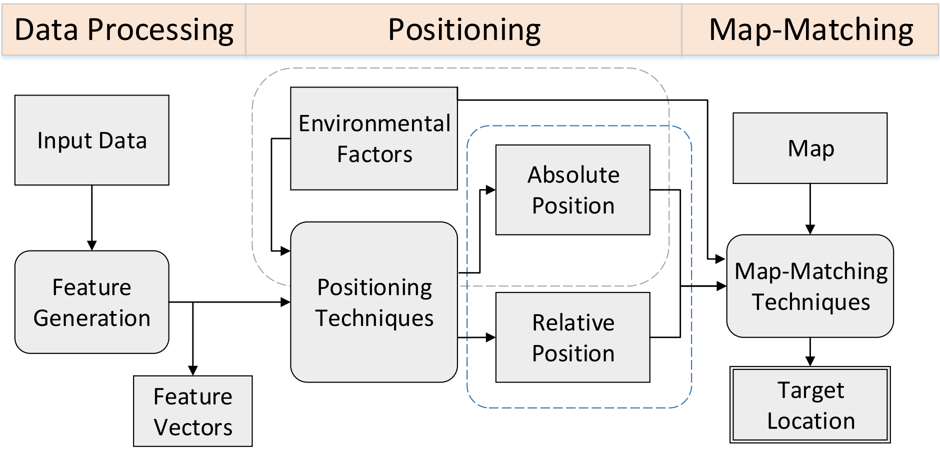


Figure 5.1: Stages of Object Localization

position at the specific location on the map.

The data processing stage takes as input a certain type of measurements based on the sensing modality and transforms the metadata into feature vectors that can better represent the target state in the environment. We classify the sensing system as either active or passive based on the criterion that if the device used in the sensing system can obtain the input data independently, then the system is considered an active system. Thus, GPS is classified as a passive system, since the receiver requires satellites to transmit the GPS signals. This classification of the sensing system is used to organize the following positioning techniques.

Positioning techniques are classified as absolute or relative based on whether the estimated target position is related to known environmental factors. The environmental factors are typically the position of the reference nodes such as beacons, landmarks, or anchor nodes that keep broadcasting their positions through wireless signals. In general, the absolute position is usually estimated through a passive sensing system, since receiving input data from reference points usually implies the positions of the reference points are also known. Taking GPS as an example, GPS signals carry the position information of satellites so that the estimated position using GPS is absolute. On the contrary, a relative target position is estimated with respect to the origin of the coordinates at the center of the target's starting position. When the target moves, the latest target position can be obtained by updating the previous target position with the estimated displacement. Taking kinematic equations as an example, the displacement and velocity can be derived if measurement time and acceleration are available. The displacement can be derived from data from passive sensors such as an accelerometer. Therefore, unlike those absolute positioning techniques that rely on the positions of the reference nodes in the infrastructure, relative positioning techniques are more flexible in deployment.

The last stage of localization is to match the target's estimated position on the map to obtain the corrected target location. Unlike absolute positioning, which considers environmental factors, relative positioning requires information such as the initial target location or landmark positions obtained during the target movement to match the relative position on the map.

In the following sections, we organize the commonly used positioning techniques based on our classification of positioning techniques.

5.2 Positioning Techniques

This section presents a taxonomy of positioning techniques, as shown in Fig. 5.2. Positioning techniques can be classified into relative and absolute ones. Motion measurement-based and radio-based techniques are representative of relative positioning and absolute positioning, respectively. Sensor fusion techniques are used to combine data measured by more than one sensing modality to make estimation. Training model-based techniques use abundant data to train probabilistic models for estimation. The sensor fusion and training model-based techniques can be used for both relative and absolute positioning. Table 5.1 summarizes these techniques by listing the sensing modality, features, and the algorithms used.

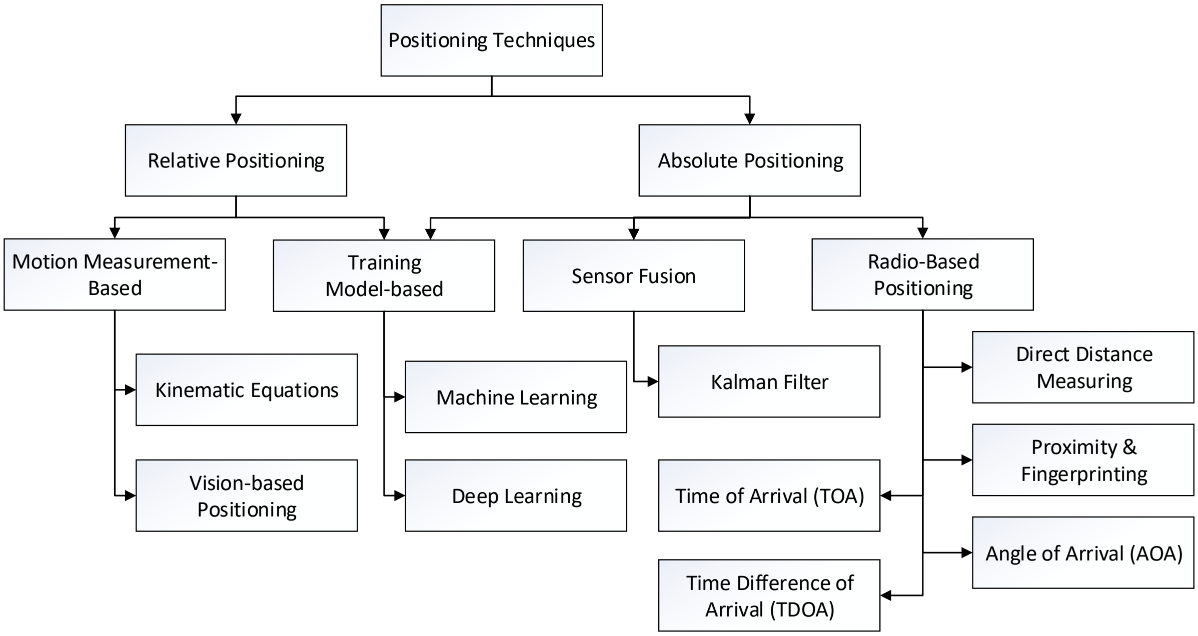


Figure 5.2: Taxonomy of Positioning Techniques.

5.2.1 Radio-Based Positioning

Radio-based positioning techniques exploit the characteristics of the received radio wave signals transmitted from the reference points to estimate the target position. The data processing stage generates a variety of features such as connectivity, distance, angle, and signal strength based on the sensing modality. This section reviews how radio-based techniques use different sensing modalities for object positioning.

Proximity and Fingerprinting

Proximity techniques rely on the broadcast signals sent from the reference points deployed in the infrastructure. The coverage of each reference point defines the area where the transmitted signal can be received or the connectivity can be established. The position of the reference point is assigned to the target position if the target is located within the coverage. The accuracy of estimation is dependent on the number and the density of the reference points, and the estimated target position can be anywhere in the coverage of the reference nodes. Due to the simplicity, the estimated position is usually used as error-constrained information and assists other positioning techniques in achieving higher accuracy of estimation[80].

Fingerprinting techniques are similar to proximity positioning but use quantified measurements such as *received signal strength indicator* (RSSI) [65] or *channel state information* (CSI) [74] as fingerprints to estimate the target position. The implementation of fingerprinting techniques is divided into offline and online phases. In the offline phase, infrastructure deployed with reference points is required to construct fingerprint maps. The fingerprint map is composed of fingerprint vectors, each of which is a vector of fingerprints with respect to the reference points and is measured at each possible position in the environment. In the online phase, the measured fingerprint vector is associated with the closest one on the pre-built fingerprint map to find the most probable target position. Fingerprinting techniques based on RSSI can be easily implemented due to the popularity of smartphones and other smart wearable devices. However, as a result of the cost of infrastructure setup and fingerprint map construction, the fingerprinting techniques are usually implemented as part of sensor-fusion techniques similar to proximity techniques.

Direct Distance Measurement

Direct distance measurement techniques use the travel distance of radio signal and the position of the transmitters to estimate the receiver position. Based on the inverse square law, the intensity of signal is inversely proportional to the square of the distance from a transmitter. Therefore, RSSI, which is the strength of the received signal measured by a receiver, is leveraged to build the signal propagation model for different RF modules such as wireless sensor networks (WSN) [78] or Bluetooth Low Energy (BLE) [68]. The target position can be determined by applying trilateration techniques to the estimated distance and the positions of the reference nodes, which are the position of transmitters in the signal propagation model. The line-of-sight (LOS) condition is assumed in the signal propagation models. Therefore, in reality, the accuracy of estimation is subject to environmental noise and the obstacles in the signal-transmission pathway.

Time of Arrival

Time of Arrival (ToA) techniques use the signal-transmission time to estimate the distance between two nodes. With two timestamps that indicate the times at which the signal was sent and received, respectively, and the signal transmission speed, the distance between the transmitter and the receiver can be estimated by multiplying the signal transmission speed by the transmission time. Similar to direct distance measurement (Section 5.2.1), the estimated distance can be further used to estimate the target position by geometry-based approaches. For example, the WLAN infrastructure is leveraged to estimate the target position with respect to the routers' position [32]. ToA uses signal transmission time so that the precise time synchronization between the transmitter and the receiver is required to obtain accurate distance estimation.

Time Difference of Arrival

Time Difference of Arrival (TDoA) is a variation of ToA. Instead of requiring the timestamp at which signal is transmitted, the target broadcasts the signal to the reference nodes and uses the difference of signal transmission time to estimate the target position [21]. Unlike ToA, which directly measures the distance between two nodes, TDoA measures the difference in distance between the target and each reference node to obviate time synchronization.

Angle of Arrival

Angle of Arrival (AoA) computes the angle of the received signal and uses triangulation to estimate the target position [40]. Compared to ToA based methods, AoA does not require precise time synchronization but requires directional antennas or antenna arrays to obtain the angle of the received signals. However, the cost of setting up the infrastructure for AoA is greater than that for RSSI. In addition, unlike the RSSI techniques that can use existing mobile devices, extra hardware is required for AoA techniques. Thus, the AoA techniques are less frequently used compared to RSSI techniques.

5.2.2 Motion Measurement-Based Positioning

Motion measurement-based methods use physical features associated with a moving object to estimate the target position. Physical features such as acceleration, angular velocity, or images that contain the target position information related to the environment are commonly used as input to the existing approaches.

Kinematic Equations

Kinematic equations (5.1) relate acceleration, velocity, displacement, and time to describe the motion of an object. An accelerometer is an electronic device that can provide acceleration measurement every sample period. By kinematic equations, the velocity and displacement can be sequentially obtained using the measured acceleration and the corresponding sample period.

$$\begin{aligned} \vec{v} &= \vec{v_0} + \vec{a} \cdot t \\ \vec{d} &= \frac{\vec{v_0} + \vec{v}}{2} \cdot t \\ &= \vec{v_0} \cdot t + \frac{1}{2}\vec{a} \cdot t^2 \end{aligned} \tag{5.1}$$

Dead reckoning (DR) is an iterative technique that estimates the current position by the the previous estimated position and the current measured data such as speed, direction, time, the number of pedestrian steps, or the number of turns of the wheels. DR is a simple and fundamental option for positioning when GPS signal is not available. Combining DR with kinematic equations using accelerometers [27], the current position is the sum of the previous position and the displacement, which is computed using equations (5.1). The position estimated by the combination of kinematic equations and DR is a relative position, since the initial position or environmental factors are not required. However, the noise in the acceleration measurement will lead to quick drift of the estimation in a few seconds since the displacement is calculated by double integration, which results in a quadratically growing error in the displacement estimation. The error keeps accumulating in the new estimated results since the new estimation is based on the previous estimation, and DR itself has no explicit correction mechanism. Therefore, an error handling mechanism such as the boundary condition, which uses zero velocity when the target is static, is required to suppress the growth of error [3].

Vision-based Positioning

Vision-based techniques use image processing for position estimation. Similar to radio-fingerprinting techniques, an image database is built to cover all possible positions in the specific environment, and a photo taken by the target is compared with the database to find the most probable position. The cost of vision-based techniques is the environment investigation for image databases and the intensive computation capability for image processing so that central servers are usually leveraged to solve the image-matching problem [67].

5.2.3 Kalman Filter

Kalman filter is an iterative sensor fusion technique, which takes as input different types of sensor data to compensate for the insufficiency of individual sensors. A vector of states is maintained and iteratively processed along with the new input data for new estimation. Kalman filter is divided into two steps: prediction and update.

In the prediction step, a state-transition model and a control-input model are developed to predict the current state based on the previous state vector and the current input data. Meanwhile, a covariance matrix is used to indicate how accurate the current prediction is based on previous covariance matrix and process noise, which models unknown external uncertainties and follows a zero-mean multivariate normal distribution.

The update step aims at improving accuracy of the predicted state vector by the extra information from the observation, the other input data that provides information about the current state. An observation model is developed and the observation noise is assumed to be zero-mean Gaussian white noise.

In order to find the optimal estimation, the Kalman gain, which is the measure of how reliable the prediction and observation are, is calculated based on the covariance of process noise and observation noise. The predicted state is then updated by adding the proportion of difference between predicted state and observation determined by the Kalman gain.

Taking the IMU and fingerprinting data as an example [65], the prediction step develops the state-transient model and control-input model based on the kinematic equations and DR. The update step uses the fingerprinting measurement as observation to associate the possible position. The Kalman gain is determined by the hyper-parameters for the covariance noises, and the final target position is estimated by following the update steps of Kalman filter.

The Kalman filter assumes the system being modeled is linear dynamic system (LDS). Variations such as Extended Kalman filter and Unscented Kalman filter are developed to handle the nonlinear system. The LDS assumption and the use of Gaussian distribution make Kalman filter less flexible. However, compared to other alternatives such as particle filter or hidden Markov model, Kalman filter is less computationally intensive, and the state-transient model is well developed for positioning applications such that Kalman filter is still one of the most popular approaches.

5.2.4 Statistical model-Based Positioning

A statistical model is a mathematical model that takes as input a random variable and estimates the possible output a probability distribution. This section elaborates the popular techniques that use machine learning and deep learning to construct the statistical models for object positioning.

Hidden Markov Models

Hidden Markov model (HMM) is a probability model that defines a joint probability distribution based on Markov process over a sequence of states and a sequence of corresponding observations. The hidden states are those states that are not observable but can be inferred through the observations. HMM models system behavior by two parameters: the state-transient probabilities matrix and emission probability matrix. The *state-transient probabilities matrix* provides the probability of a state transition. The *emission probability matrix* represents the probabilities of observations in each state. A joint probability distribution of states and observations is defined by state-transient distribution and emission distribution. HMM makes the assumption that the observations are conditionally independent of all other given states, and the property of Markov process indicates that the next possible state is only dependent on the current state. By applying the assumption and property, the joint probability distribution can be interpreted as the distribution over all possible states by taking as input the current state and the measured observation. The work [28] solves the positioning problem by using an HMM that considers the fusion of two sensing modalities. A set of states is created to represent the possible positions on the floor plan, and the sequence of observations are composed of pairs of RSSI fingerprint and step-detection information. The HMM can be used for position estimation after it is trained with the collected observations under the definition of states. The disadvantage of HMM is the cost of estimation, since all the states have to be paired with observations to find the maximum of the joint probability distribution.

Particle Filter *Particle filter* is a probabilistic model based on Bayesian filtering and importance sampling. It also assumes that the system is modeled as a Markov process. Similar to Kalman filter, particle filter is implemented in an iterative algorithm that can be divided into two steps: prediction and update step. In the prediction step, the current state is estimated based on the previous state. The update step takes the current measurement to update the predicted estimation. However, unlike Kalman filter that assumes linear system and Gaussian distribution, particle filter uses numerical methods to approximate the expectation of the posterior probability distribution. By importance sampling techniques, the samples, also called particles, are drawn from the state transition probability distribution, which is acquired in model training. In addition, the sample weights are repetitively computed by multiplying the probabilities of observations by previous sample weights. The expectation of state estimation is the sum of the product of the normalized sample weights and the corresponding samples. Particle filters can be implemented in the way of sensor fusion using IMU and fingerprint measurements [76]. In the prediction step, the IMU data is used with previous estimation to predict the current state. The predicted state and the state estimated from fingerprint observations are used in importance sampling to compute the weights of particles. The weight can be regarded as the probability of the fingerprint observed from the predicted state using IMU data. The weights are multiplied by the drawn particles, which represent the predicted states, to find the most probable position. The estimation of particle filters relies on the particles and the corresponding weights. Therefore, the quality of sampled particles have a direct impact on the accuracy. In addition, the disadvantage of the particle filter is it requires many particles to achieve good performance, which can be computationally expensive.

Conditional Random Fields *Conditional random fields* (CRFs) are a class of statistical models used for structured prediction, which makes the prediction considering the correlations in the sequence of input data. Linear-chain CRF is one that defines a conditional probability of a sequence of output estimation given a sequence of input observation in which the sequential dependencies of input are explored. The Linear-chain CRF can be regarded as the implementation of a set of feature functions along with the corresponding weights. The feature functions are defined as the joint probability distribution of the estimated state and the input observations. However, they can also be defined as the joint probability distribution of sequential state transition and the observations. The weights of features are learned in the model training process. By summing up the product of feature function outputs and the corresponding weights, the linear chain CDFs output the most probable sequence of states based on the sequence of input. The work [77] uses linear-chain CRFs to solve the positioning and map-matching problem based on the IMU data and RSSI fingerprint measurements. Two feature functions are implemented by using the moving distance and the orientation estimated from IMU data. These two feature functions are defined to find the most probable pair of states that best fit the change of position resulting from the object movement. Another feature function uses the previous output estimations and the orientations of displacement in a recent time window to correct the current orientation and find the most probable consecutive states. The last feature directly uses the fingerprints to find the most probable state. Combining all the features, the estimation turns out to be the object trajectory describing the relative position and object movement.

Neural Network Artificial neural networks (ANNs), usually called neural networks (NNs), have become one of the dominant models of applications. The structure of an ANN is composed of three layers: input, hidden, and output layer. The input layer is the interface that transforms the input data into a certain format that can be processed by the hidden layer. The hidden layer is the core of an NN in that it discovers patterns in the data and generates feature vectors. The feature vectors are fed into the output layer to generate the predicted result based on the type of problem being solved. In addition, the NNs are very powerful tools to extract the features, which can be used to improve the capabilities of other machine learning models compared to using the raw features. Thanks to the powerful data processing and prediction capabilities, deep learning is developed and variations of NNs are proposed to better explore the dependency between input data and predicted outputs. Variations such as the convolutional neural networks (CNNs) are proposed to explore the spatial patterns, and recurrent neural networks (RNNs) are used to explore temporal dependency in input data. Architectures of NNs and their variations are on-going research.

Table 5.1: Positioning Techniques

Positioning Techniques	Sensing Modality	Features	Positioning Algorithm	Type of Localization
Radio-Based				
Proximity	Wireless Sensing	Connectivity	Database Association	Absolute
Fingerprinting	Wireless Sensing	RSSI, CSI	Database Association	Absolute
Direct Distance Measuring	Wireless Sensing	RSSI	Trilateration	Absolute
TOA	Wireless Sensing	Signal Transmission Time	Trilateration	Absolute
TDoA	Wireless Sensing	Difference of Signal Transmission Time	Trilateration	Absolute
AOA	Antenna Array	Angle of Received Signals	Triangulation	Absolute
Motion Measurement-Based				
Kinematic Equations	IMU	Acceleration	Dead-Reckoning	Relative
Vision-based	Camera	Object Recognition	Image Processing	Relative
Sensor Fusion				
Kalman Filter	IMU, RSSI	Acceleration	Linear System Model, Gaussian Distribution	Relative
Training Model-Based				
Hidden Markov Model	Wireless Sensing, IMU	Acceleration, RSSI	Joint Probability Model, Markov Process	Relative

5.3 Indoor Localization

Indoor localization generally infers the target object located in an indoor environment without the access of GPS signals. Unlike GPS, which has been the dominant mechanism for outdoor localization, indoor localization is realized through a variety of techniques considering the diverse scenarios of applications. For example, the firefighters require accurate location information to provide assistance at the scene of an emergency. The information of floorplan or infrastructure is usually unavailable such that those techniques that depend on the reference points are inadequate. However, another application such as parking can adopt sensor fusion that takes as input the IMU measurement and RSSI fingerprint to estimate the user location and plans the route for the user to reach the target parking spot. Our work elaborated in chapter 5 assumes the floorplan and the information of the reference points are available. The following section surveys related works, and their proposed techniques will be implemented for comparison with our solution.

5.4 Related Work

UPTIME [7] is a user movement tracking system that generate a pedestrian trajectory based on *Pedestrian dead reckoning* (PDR) using only the inertial sensors embedded in user's mobile phone. The proposed system uses the inertial sensor data to develop a step detection method that activates the state transition, which is the change of position, when the pedestrian is moving. A gait type classification model estimates the displacement of pedestrian movement by mapping the identified gait type with the predefined step length. Similar tracking systems are proposed and enhanced by modeling the step pattern to dynamically estimate step length. The work [58] evaluates the impact on accuracy by using three different step length models. SmartPDR [36] further improve a existing step model [75] that assumes the vertical impact occurred from walking activity is proportional to step length. However, the step models still use predefined parameters to estimate the step length by measured acceleration. The vantage of the mentioned system is only inertial sensors are required but the accuracy highly depends on the estimation of user step length and the error is accumulated in the estimation since only the inertial sensors is used and the lack of error correction mechanism.

A sensor fusion algorithm [45], Inertial Sensors and WiFi Fingerprint (ISWF) algorithm, is developed to handle the cumulative error resulted from DR. Unlike other works using the reference information to correct the error in the estimation, ISWF uses the distance measured by AoA technique to improve the step length estimated by using inertial sensor data. Another work, UnLoc

[72], uses the step count method to estimate the step length. UnLoc classifies the inertial sensor data by patterns and use the identified patterns to infer the possibly encountered landmarks on the map. By using the knowledge of floor map and the step count, the step length can be estimated. To obtain accurate results, these methods require extra information like fingerprint map or the capability of pattern recognition for landmark identification.

Other existing works propose KF-based sensor fusion algorithms that can directly optimize the estimated position. Various sensors are used to get the measurement data. [16] deploys cameras around corners in indoor environment to measure the user position. [57] uses a distance sensor, LIDAR, to detect the wall on corridors and the detected walls are used as constraint to correct the heading of the pedestrian path. The barometer [38] is used with the accelerometer for height estimation. The RSSI of received wireless signals is also leveraged to measure the distance between the target devices and the deployed beacons. The wireless technologies such as WiFi [43, 14] and RF [60] are broadly adopted. The measurement model for observation is required to enable KF and the performance of KF will be decreased if the model is not accurate such as the system is not linear or the covariance matrices of noise do not follow the normal distribution in reality.

The Zee [59] solution adopts particle filter to fuse inertial sensor data and RSSI fingerprints to estimate a four-dimensional state vector that were used to calculate the user position. The state vector contains the estimations of step length and direction. The map constraints are also reviewed to remove unqualified particles such as the change of position is overlapped with the wall on the map. The PF can provide better result than KF but is computationally expensive in terms of the number of particles used. In addition, the work [10] compares the performance of the step counting-based DR combining with the map-matching algorithm with PF that only use the inertial sensor for estimation without considering the map constraint. The developed algorithm leverages the pattern of pedestrian walking in a corridor that the step direction is quasi-straight line to calibrate the sensor orientation. The result shows the proposed DR algorithm can provide more accurate estimation than PF.

MapCraft [77] provides an affordable solution based on *Conditional random field* (CRF) for devices with lower computation capability. The feature functions in MapCraft are defined by using step length, step direction, RSSI fingerprints respectively. The map constraint is considered while defining the state. Therefore, MapCraft does not spend extra cost to remove unqualified state as the removal of particles in PF resampling step. In addition, the estimated results are the normalized sum of the feature function output and the number of feature function is far less the number of particles required in PF. Therefore, CRF is adopted in our system that will be demonstrated in chapter 5.

Researchers have proposed techniques that use shared information to enhance localization. The location can be estimated through the information exchanged between known location reference nodes like deployed beacons or other devices [70]. The centralized design that use servers to coordinate devices have also been proposed to estimate the location of devices systematically [17, 35]. However, these techniques involve a third party to manage and process the information for IoT devices and suffer from the constraints of flexibility and real-time response. A decentralized information exchanged mechanism is proposed and described in chapter 5.

Chapter 6

Technical Approach

6.1 Background

This section presents the sensing modality and the generation of pedestrian trajectory, which is exchanged when users are in an encounter. The pedestrian trajectory is generated by Pedestrian Dead Reckoning (PDR) using an inertial sensor. This section first reviews the sensing modalities for pedestrian trajectory, followed by step detection, which triggers the estimation of heading and step length.

6.1.1 Sensing Modalities

Pedestrian dead reckoning (PDR) can be done using IMU sensors and RF-based proximity sensors, both of which are found on virtually all wearables, smartphones, and tablets.

An IMU sensor typically consists of an accelerometer, a gyroscope, and optionally a magnetometer. IMUs are miniature in size and incur minimal power consumption. The IMU data can be used to model the step behavior of pedestrians, and the pedestrian trajectories can be generated by

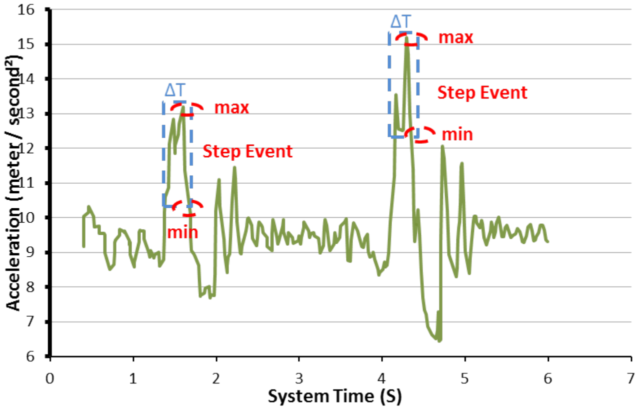


Figure 6.1: Peak-and-Valley for Step Detection

collecting the sequential step data.

Virtually all wearables and mobiles support Bluetooth Low Energy (BLE) for low-power, short-range communication. One key feature of BLE is that it can also be used for proximity sensing between two BLE nodes to detect encounter events. Section 6.2 provides the details on the use of BLE for our encounter model.

6.1.2 Step Detection

Step detection is the problem of inputting pedestrian motion data and outputting events that delimit the steps. The *peak-and-valley* detection method [44] can monitor pedestrian activities based on the measured accelerometer data shown in Fig. 6.1. The patterns of acceleration are evaluated to determine the user movement behavior. We first define the acceleration change threshold and the time window and then continuously monitor the magnitude of acceleration. If the difference between the maximum and minimum acceleration values within the time window is larger than the threshold, then a step event is triggered.

6.1.3 Heading Estimation

Heading estimation is the problem of computing the orientation. The heading can be estimated by the product of the gravity vector and the magnetic vector provided by the accelerometer and magnetometer, respectively. Accurate orientation estimation is a challenge due to the interference by the user movement and the physical environment to the accelerometer and magnetometer. On the other hand, a gyroscope suffers much less interference and can provide more accurate measurement for the heading orientation. The rotated angles can be calculated by multiplying the sensor output by the sample period. The current orientation can then be obtained by summing up the successive calculated angles over time. In order to remove the accumulated error in the estimated orientation, a low-pass filter is applied to the calculated orientation data, and the output is used as trajectory for data exchanged in an encounter.

6.1.4 Step-Length Estimation

The estimation of step length relies on the accelerometer. However, the estimation is influenced by individual factors such as step frequency, walking speed, age, etc. Step-length estimation is outside the scope of this work. Without loss of generality, we assume the step length is fixed, and a better step-length estimator can be plugged in to improve the accuracy of this work.

6.2 Encounter Model

The encounter among multiple devices is modeled to enhance the efficiency of indoor localization by sharing the trajectories. A low-power, device-to-device (D2D) RF interface such as Bluetooth Low Energy (BLE) serves as the mechanism for both communication and proximity sensing. A proximity threshold is defined to trigger the encounter event, which enables the sharing of trajectories among devices.

6.2.1 Proximity Threshold

The encounter model uses the capability of detecting the proximity of other users to define an encounter event in terms of the distance threshold between two devices. The detection is similar to the pairing process where two devices discover each other and establish the connection. The proximity sensing is defined by the BLE protocol and is widely used thanks to the low power consumption and low cost. Therefore, our encounter model utilizes the pairing process of BLE protocol to implement the proximity feature based on Bluetooth 4.1, in which the BLE stack supports simultaneous advertising and scanning. During the pairing process, a device, which acts as a scanner, periodically scans the advertising packets sent from other BLE devices, which act as advertisers. The scanner can estimate the distance between itself and the advertiser based on the RSSI.

$$\log_{10} D = \frac{A - RSSI[dBm]}{10n} \tag{6.1}$$

An encounter event is triggered if the packets from the advertisers are scanned and detected in a certain range. Due to the use of omnidirectional antennas in most BLE-based systems, it is difficult to determine the direction. Thus, in our system design, the encounter event happens only if the distance between scanner and advertiser is within the step length. Eq. (6.1) is used to measure the distance between two devices. To detect that the encounter happens, we set the value of distance as the step length, which is the maximum distance between two encountering users, and the RSSI is measured to find the proximity threshold, which is the minimum RSSI value that two nodes can measure from the received packets within a step length.

Fig. 6.2 shows our RSSI measurements of scanned advertising packets between two devices. Based

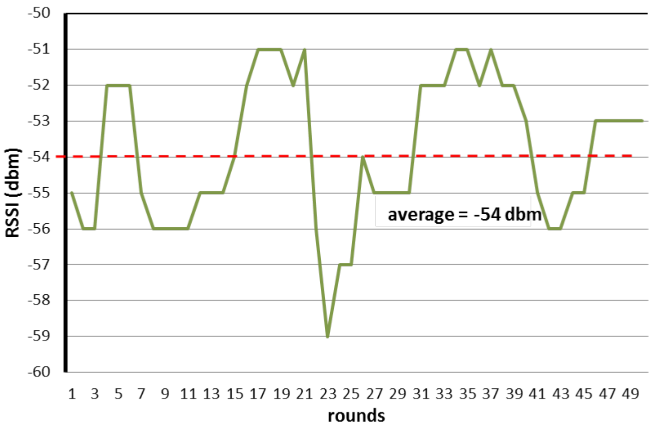


Figure 6.2: The Proximity Threshold of RSSI

on our experiment, for $n = 2$ and $A = 54$, the average RSSI value of -54 dBm corresponds to our chosen step length D. Thus, an encounter event occurs whenever the RSSI is -54 dBm or higher.

6.2.2 Trajectory Exchange

We extend the proximity sensing with trajectory exchange to enable the collaborative localization. BLE allows data to be piggy-backed onto its advertising payload. Therefore, it is not necessary to send the trajectory data in a separate transaction or to go through an extra connection procedure, which would incur time and power overhead. However, due to the limited size of the payload, a node sends only a portion of its trajectory instead of the entirety. The original size of the orientation data is 32 bits floating point format.

To represent the trajectory in a compact format, we assume that the subject takes one step at a time and can move in only one of eight possible directions (i.e., in 45° increments). This enables us to encode the state transition using only 3 bits using only about 1/10 the original trajectory size. This translates into about 42 steps of trajectory in BLE 4.0 and 4.1, and BLE 4.2 increases the payload size by nearly an order of magnitude.

6.3 Collaborative Indoor Localization

The core of EcoLoc is to combine the user trajectory with the exchanged ones to enhance the estimation capability of positioning. As with any DR, the initial location is required to match the trajectory on the map for user location estimation. Since our PDR mainly relies on the fidelity of inertial sensor data, the sensor noise can severely influence the accuracy of our PDR. The supplementary information and available constraints, such as the beacon signal or the corridor walls on the floorplan that users cannot penetrate, can be exploited to correct the error during estimation. In EcoLoc, the collaborative CRF is proposed to manage the trajectories and a real-time tracking algorithm is developed to estimate the user's location.

6.3.1 Background Models

A collaborative CRF (CCRF) is implemented in EcoLoc to exploit the trajectories obtained from the opportunistic encounter. We first introduce the Hidden Markov Model and Maximum Entropy Model, and the CRF is derived by integrating both models. The feature functions for CCRF are defined to fuse the exchanged trajectories to estimate the user location.

Hidden Markov Model (HMM)

HMM models the distribution of a sequence of observations without the knowledge of a sequence of observed states and is able to estimate the most likely states to generate the same observations we have. For sequence reasoning, two assumptions are made: (1) the probability of going to a state at time t depends only on the state at time $t-1$, and (2) the probability of the observation at time t depends only on the current state at time t. Therefore, given a set of states $X = \{s_0, s_1, ..., s_m\}$ and a set of observations $Y = \{z_0, z_1, \ldots, z_n\}$, the HMM can be formulated as follows:

$$p(\vec{S},\vec{Z}) = \prod_{t=1}^{T} p(s_t|s_{t-1})p(z_t|s_t) \tag{6.2}$$

where $\vec{S}$ and $\vec{Z}$ represent the estimated sequence of states and the sequence of observations we have, respectively. The joint probability distribution of chosen states and observations points out the possibility of chosen states so that we can explore the most likely series of states. However, several shortcomings make HMM less attractive considering the accuracy and efficiency. The probability joint distribution is not efficient enough for HMM since the effort to model the observations is unnecessary and only the output states provide the information we want.

The dependency assumption between the estimated states and the corresponding observations limits the capability to consider the correlation of observations. Given a sequence of observation, the conditional probability cannot be maximized globally, thereby undermining the accuracy of estimation.

Maximum Entropy Model (MaxEnt)

MaxEnt is able to handle the local optimum problems suffered by HMM since MacEnt assumes nothing about what is unknown. In other words, MaxEnt does not assume the dependency between state and observation so that output state is estimated based on entire observation data.

The main idea of MaxEnt is to provide a most uniform distribution (highest entropy) model if incomplete information about the probability distribution is given. Given the input random variable set X and output variable set Y, conditional entropy quantifies the uncertainty of described situation as:

$$H(y|x) = -\sum_{(x,y)\in(X,Y)} p(y,x)\log p(y|x) \tag{6.3}$$

The goal of MaxEnt is to find the set of model $p^*(y|x)$ that is *consistent* with the training set and outputs the model that provides the largest conditional entropy. A model is consistent with the training set only if the expected value of each feature function in the empirical distribution as modeled by Eq. (6.4) is equal to its expected value in the model distribution in Eq. (6.5). Here, the feature functions are those that link the observation from the training set with the output states we want to estimate.

$$E_{\widetilde{p}}(f_i) = \sum_{X \times Y} \widetilde{p}(y,x) f_i(y,x) \tag{6.4}$$

$$\begin{aligned} E_p(f_i) &= \sum_{X \times Y} p(y,x) f_i(y,x) \\ &= \sum_{X \times Y} \widetilde{p}(x) p(y|x) f_i(y,x) \end{aligned} \tag{6.5}$$

$$E_{\widetilde{p}}(f_i) = E_p(f_i) \tag{6.6}$$

With these constraints, the expected value must be the same between the model and the training set as shown in Eq. (6.6), such that the MaxEnt can be treated as solving an optimization problem by using Lagrange multipliers, λ_i, for constraints. The MaxEnt model can be formulated as:

$$p^*(s|z) = \frac{1}{Z_{\vec{\lambda}}(z)} \prod_{i=1}^{m} \exp(\lambda_i f_i(s,z)) \tag{6.7}$$

$$Z_{\vec{\lambda}}(z) = \sum_{s \in S} \exp(\lambda_i f_i(s,z)) \tag{6.8}$$

6.3.2 Conditional Random Field

CRF can be understood as a sequential extension to the MaxEnt model and represented by various feature functions accompanied by weight λ_i formulated as shown in Eq. (6.9), where j denotes the position in the observation sequence and m is the number of feature functions. CRF combines the benefits of HMM and MaxEnt. It is capable of realizing the probability of state transition by defining complex features such that CRF can take context into account in training and testing phase to enhance the accuracy of estimation. The feature functions f_i represent the constraints provided by the collected observations such as floor map or trajectory in the scenarios of localization.

$$p(\vec{S}|\vec{Z}) \propto \prod_{j=1}^{n} \exp\left(\sum_{i=1}^{m}(\lambda_i f_i(s_{j-1}, s_j, \vec{Z}, j)\right) \tag{6.9}$$

In EcoLoc, CRFs consist of feature functions that describe the possibility of location transition by using the corresponding observation [77]. The step detection decides when to activate the CRF estimation. The heading orientation, Z_t^{θ}, is used as the observation to define our first feature function as shown in Eq. (6.10). We assume the heading orientation is a log-normally distributed random variable so that the probability density of the log-normal distribution can be leveraged to formulate our feature function. The σ_{θ}^2 is the heading variance of observations Z_t^{θ}, and $\theta(S_{t-1}, S_t)$ is the heading orientation between the last state S_{t-1} and the state S_t that we estimate for the current step.

$$f_1 = \ln\left(\frac{1}{\sqrt{2\pi\sigma_{\theta}^2}}\right) - \frac{(Z_t^{\theta} - \theta(S_{t-1}, S_t))^2}{2\sigma_{\theta}^2} \tag{6.10}$$

The second feature function is defined by using the RSSI observation. It is optional and is considered only when the beacon signal is available in the indoor environment. Similar to our first feature function, we use the RSSI observation to calculate the distance, Z_t^d, between the user and the beacon B_i to formulate the feature function as follows:

$$f_2 = \ln\left(\frac{1}{\sqrt{2\pi\sigma_{d}^2}}\right) - \frac{(Z_t^{d} - D(B_i, S_t))^2}{2\sigma_{d}^2} \tag{6.11}$$

where the σ_d^2 is the distance variance of the observations Z_t^d and $D(B_i, S_t)$ is the distance between the beacon and the estimated state.

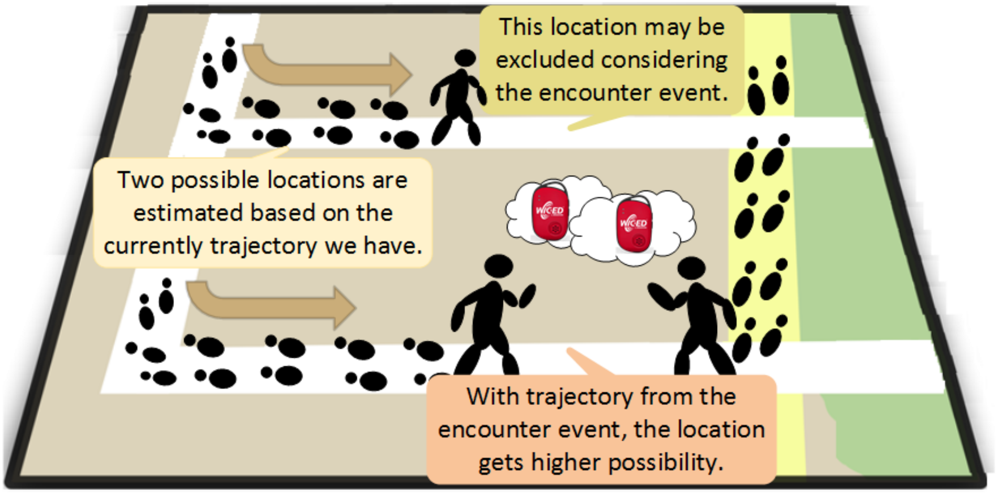

This location may be excluded considering the encounter event.
Two possible locations are estimated based on the currently trajectory we have.
WICED
WICED
With trajectory from the encounter event, the location gets higher possibility.

Figure 6.3: Encounter Event in our CCRF

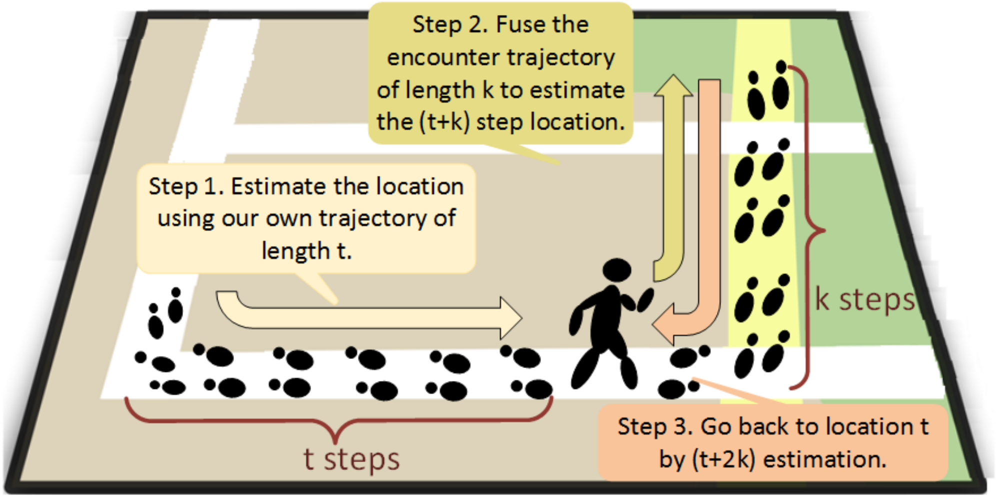

Step 2. Fuse the encounter trajectory of length k to estimate the (t+k) step location.
Step 1. Estimate the location using our own trajectory of length t.
k steps
t steps
Step 3. Go back to location t by (t+2k) estimation.

Figure 6.4: Collaborative Conditional Random Fields

6.3.3 Collaborative Conditional Random Field

Our CCRF merges the shared trajectory with our own trajectory to improve the convergence distance of localization. As a simple example shown in Fig. 6.3, two possible locations are estimated by ordinary CRFs. After combining the trajectories, our CCRF eliminates the upper location but selects the lower one because that is the only feasible location that allows the combined trajectory to satisfy the floor-plan constraints.

Our CCRF is illustrated in Fig. 6.4. It extends the current trajectory with the acquired trajectory from the encounter events. Suppose the length of the acquired trajectory is k steps and our trajectory is t steps, the merged trajectory may not be used to estimate the current location since the estimated location is at step $(t+k)$ instead of step t. Therefore, our CCRF reverses the acquired trajectory and estimate the location at step $(t+2k)$. The CCRF can be formulated as follows:

$$p(\vec{S}|\vec{Z}) \propto \prod_{j=1}^{t+2k} \exp\big(\sum_{i=1}^{m}(\lambda_i f_i(s_{j-1}, s_j, \vec{Z}, j)\big) \tag{6.12}$$

6.3.4 Real-Time Tracking Algorithm

Algorithm 3 shows our real-time tracking algorithm based on Viterbi's [22]. It provides real-time update for IoT devices without estimating location from the scratch.

Algorithm 3: Real-time tracking algorithm

```
Input: The length of output $T$, observation $\vec{Z}$, state $\vec{S}$
Output: The most likely hidden states sequences $\vec{X}$
Function HeuristicRealtime ($T$, $\vec{Z}$, $\vec{S}$)
begin
    // Calculate the score
    foreach $s_x \in \vec{S}$ do
        $score[s_x, j] \leftarrow \max_{y \in Valid}(score[s_y, j-1] \times \Psi_j(s_x|s_y,\vec{Z}))$ ; // Estimate the score of states at step j
        $path[s_x, j] \leftarrow \arg\max_{y \in Valid}(score[s_y, j-1] \times \Psi_j(s_x|s_y,\vec{Z}))$ ; // Store the path of states at step j
    $max\_state \leftarrow \arg\max_{s_i} score[s_i, T]$ ; // Find the highest score of state at step T
    foreach $i \in \langle 0, 1, \ldots, T-1 \rangle$ do
        $X_i \leftarrow path[max\_state, i]$ ; // Output the path
    return $X$ ;
```

When a user takes a step, each possible locations on the map, which are the pre-defined states, will get a score that represents the probability of walking to it from other possible states. Given the observation $\vec{z}$ at step j, the score of s_x transferred from state s_y is calculated through the following equations:

$$score(s_x, j) = score(s_y, j-1) \times \Psi_j(s_x|s_y, \vec{z}) \tag{6.13}$$

$$\Psi_j(s_x|s_y, \vec{z}) = \exp\left(\sum_{i=1}^{m} (\lambda_i f_i(s_y, s_x, \vec{Z}, j)\right) \tag{6.14}$$

Suppose there are possible states $S = \{s_0, s_1, ..., s_n\}$ and the length of observation is T. The sequence of the highest score at each step is the trajectory we want. Compared to the exhaustive Viterbi search, the time complexity of our tracking algorithm is $O(|S|^2 T)$ instead of $O(|S|^T)$. Furthermore, because only eight states per transition are possible, the time complexity per step is optimized to $O(|S| \times 8)$ = $O(|S|)$.

Our algorithm works as follows. First, each state is assigned a score, which is the highest score among eight calculated results, by using the observation and the score of neighbor states at the previous step (Lines 3-5). The current location is then determined as the state with the highest score (Line 6) and the trajectory is generated as well (Line 8).

Chapter 7

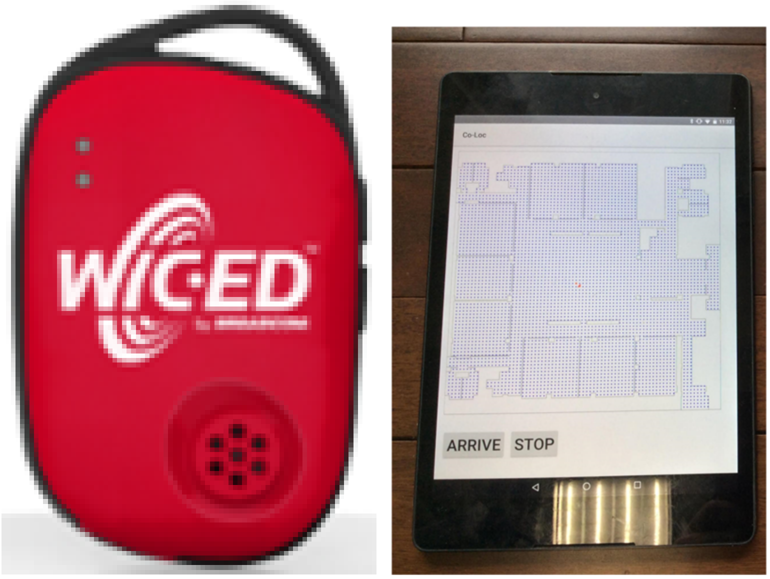
Evaluation

7.1 Experimental Validation

We have implemented our proposed work on two system platforms and set up a number of test environments to evaluate our work. The same setup is used to evaluate a number of competing techniques.

7.1.1 System Prototypes

To show the applicability of our proposed technique to a wide range of hardware, we implemented our proposed technique on two platforms: a tablet and a sensor tag. The tablet shown in Fig. 7.1b we use is the Nexus 9, which contains inertial sensors (accelerometer, gyroscope, and magnetometer) as most tablets do and runs Android 6 Marshmallow OS with Bluetooth 4.1 (dual-mode) support. On the lower end, we use the Broadcom WICED Sense Bluetooth Smart Sensor Tag [12] shown in Fig. 7.1a, which contains not only inertial sensors but also a barometer, temperature and light sensors, a buzzer, and USB slave controller, all controlled by the BCM20737S, the Broadcom



(a) Broadcom WICED Sense Tag

(b) Android Nexus 9 Tablet

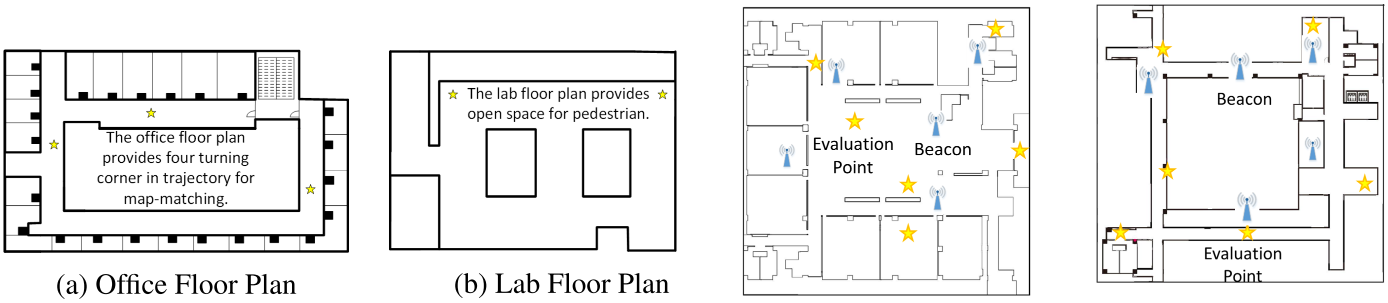
Figure 7.1: Two Platforms used for Our Evaluation

System-in-Package (SiP) with ARM Cortex-M3 MCU with an integrated BLE transceiver running the 4.1 stack.

The implementation on WICED Sense validates EcoLoc's applicability to resource-constrained IoT devices. WICED Sense provides the sensing functions and requires a connection to a central device such a smartphone, tablet, or PC with a BLE transceiver. The central device retrieves the results from the WICED Sense tag and reports the estimated location to the user. However, due to the lack of computation capability, only CRF and CCRF could be running, as PF requires too much memory and computation power for mobile devices.

7.1.2 Experimental Setup

The ground truth is measured to generate the map for location estimation. Several evaluation points are decided on the floor plan, and the untrained participants can choose any route they like to reach the evaluation points. The user trajectories would then be collected for the comparison


The office floor plan provides four turning corner in trajectory for map-matching.
The lab floor plan provides open space for pedestrian.
Evaluation Point
Beacon
Beacon
Evaluation Point

(a) Office Floor Plan (b) Lab Floor Plan (c) 1st Floor Plan (d) 5th Floor Plan

Figure 7.2: Floor Plans for Our Experiments.

with competing works over several real-world floor plans, as shown in Fig. 7.2. The tablet-based devices are tested in two different places, but the sensor tags are tested only with the smaller floor plan due to the limited physical memory for storing the map information. The improvement from the assistance of radio fingerprinting is also evaluated. The deployments of the beacons are shown on the same smaller floor plan for the tablet but not the sensor tag, due to the limited computing capability.

The results of the location estimation are collected and compared with the ground truth when the participants reach the evaluation points. Another result we collect is the *convergence distance*, in this case defined to be the number of steps required before the localizer outputs a location within a 5-meter radius, which is the precision of GPS, from the ground truth. The overhead of running CCRF is also compared in terms of the execution time on the smartphone and power on the sensor tag. Finally, when running the CCRF on the tablet, the influence of different sampling rates of the sensors on power consumption is evaluated.

7.1.3 Comparison

We implemented a number of related techniques for the purpose of comparison with our proposed *CCRF*. They include (1) ordinary *CRF*, (2) *Particle Filter*, (3) and *encounter-based collaborative Particle Filter (CPF)*. The ordinary *CRF* does not consider encounter but just uses the same feature functions as *CCRF*, and the result can show how the CRF benefits from the encounter mechanism.

Particle Filter is another popular model for indoor localization. In our implementation, a total of 200 particles are used for estimation. Meanwhile, the dead reckoning and fingerprinting techniques are the same as those used in *CRF*. The result is used to compare the advantage of *CRF* considering the computation overhead and accuracy. The last one is the *encounter-based collaborative Particle Filter*. We add the encounter mechanism to *Particle Filter* to fuse the standard deviation and the central, instead of all samples particles, so that the particles in the intersection of the standard deviations are resampled to improve the computation efficiency.

7.2 Results

EcoLoc is evaluated using two Broadcom IoT devices, the WICED Sense tag, in two indoor environments and using two Android devices, Nexus 9 tablet, in other two indoor environments to demonstrate its universal applicability. On lower-end IoT devices, EcoLoc provides basic localization functions, but on more resource-rich devices, it can be extendable to take advantage of existing infrastructure such as beacons to enhance the convergence distance and accuracy.

7.2.1 Convergence Distance

The *convergence distance* we present here is the displacement required in the estimation to provide a location within a 5-meter radius from the ground truth. Note that convergence distance is a metric mainly for beaconless operation, i.e., (P)DR under floor plan constraints but without the starting location. Convergence distance is less of a concern for reference-based localization. EcoLoc can still take advantage of beacons when available. A short convergence distance means EcoLoc can operate either without beacons or can operate well while requiring much lower density of deployed beacons.

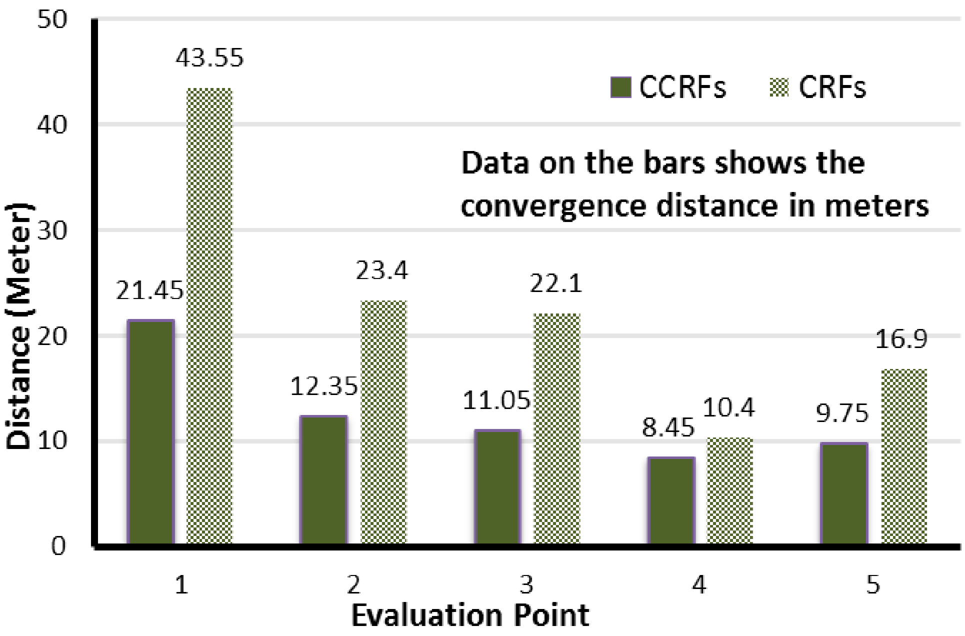


Figure 7.3: The Comparison of Convergence Distance for WICED-Sense among Five Evaluation Points

Improvement from Encounter

We first evaluate how encounter affects the CCRF on the IoT device and the applicability of EcoLoc on the IoT devices. Fig. 7.3 shows the convergence distance is significantly shortened by up to 50% compared to the non-encounter CRF in first three evaluation points. This is because the indoor space is limited and the evaluation points we put are closer to the middle of hallway. The two participants walk from either end of the corridor to reach the evaluation points. The convergence distance measured from the rest of two evaluation points did not get improved significantly because of the small indoor space such that the shorter distance is enough to estimate location. We show that the convergence distance can benefit from the trajectories exchanged upon encounter.

The experimental results on the tablet conducted on the 1st and 5th floors of the campus Building are shown in Fig. 7.4. We compare the results of PF, CRF, CPF, and CCRF. The convergence distances on the 5th floor of all four localization techniques are less than 50 meters, which are all shorter than the results on the 1st floor, because the trajectory on the 5th floor are more constrained

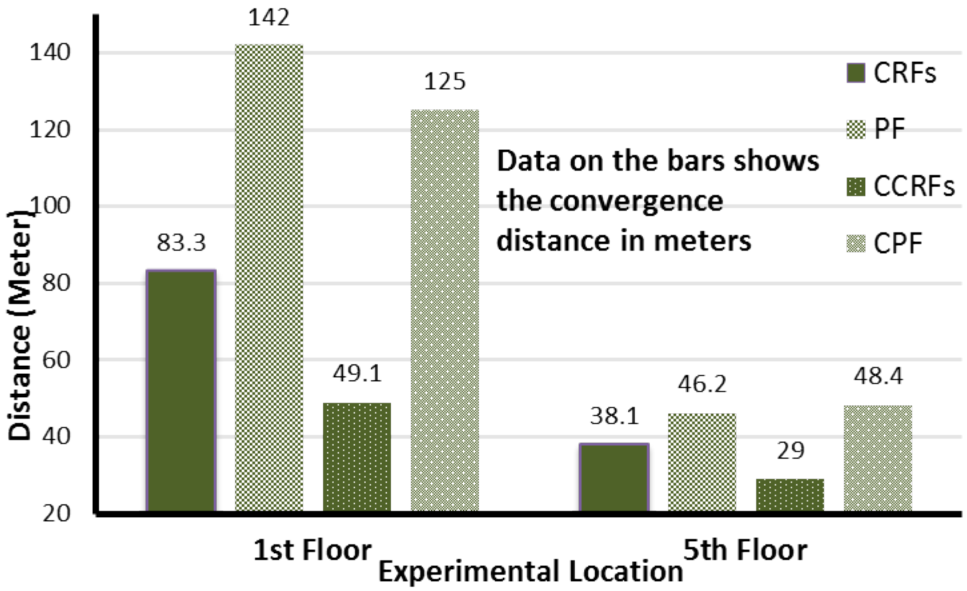
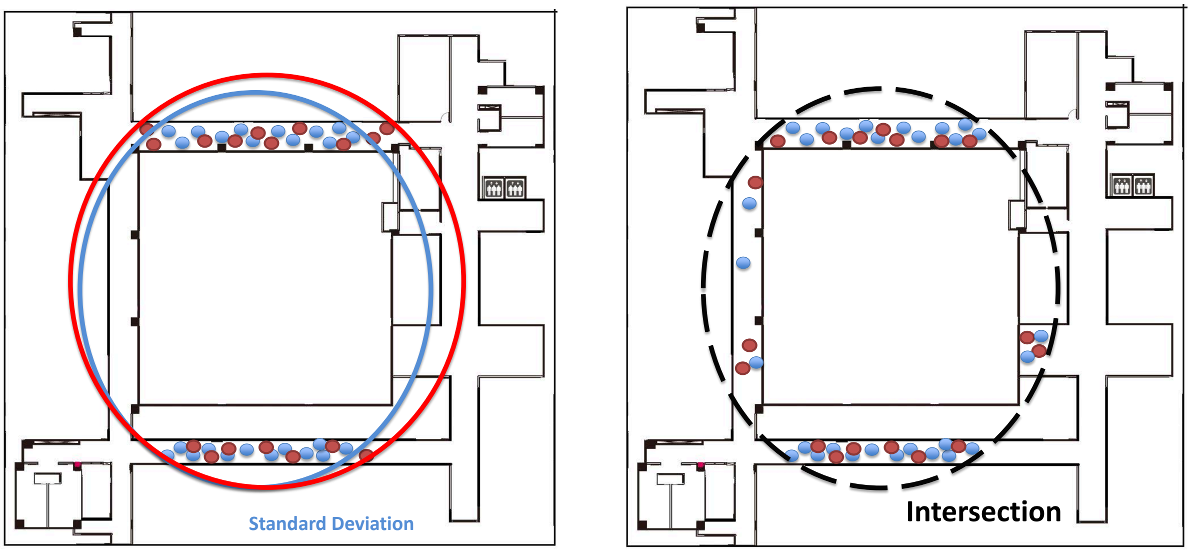


Figure 7.4: The Comparison of Average Convergence Distance for Tablet on the 1st and 5th Floors

by the floor plan by a turn in the corner. In addition, both CRF and CCRF have shorter convergence distances than PF and CPF do. This is because CRF captures the constraints and is able to provide the most probable location immediately based on the prebuilt states; in contrast, PF iteratively generates the new particles to explore the possible locations. In PF, each particle is assigned a weight to represent the probability of that particle staying at the user ground truth location. In each iteration, the weight of each particle is updated, and the weight of a particle may decrease due to constraint violation such as walking through a wall. PF gradually discards the low-weight particles and generates a new set of particles for the next iteration. Therefore, the resampling process of PF results in longer distance to obtain the most probable location.

Interestingly, CPF has longer convergence distance than PF does. This is because the resampling process of PF that we used for encounter also considers unreachable locations. As shown in Fig. 7.5, the area of the the standard deviation are calculated upon encounter. The new set of particles would then be spread in the intersection area. The right and left corridors are inside the intersection area and some particles are spread. These particle cause the extra cost for the calculation of convergence distance.



(a) Before encounter, the areas of the standard deviation are large and almost overlapped.

(b) After encounter, the particles are resampled in the intersection area.

Figure 7.5: The bad situation of encounter event for CPF.

Encounter in Open Space

Fig. 7.4 shows the convergence distance conducted on the 1st floor. The difference between the 5th and 1st floors is that the 1st floor is an open indoor space with much fewer floor-plan constraints, so that convergence distance increases for all experiments. However, the results in Fig. 7.4 show the convergence distance of CCRF is significantly reduced comparing to CRF since the encounter provides opportunities for sharing trajectories among users. In contrast, the encounter events did not provide much more assistance since PF-based systems work poorly in open-space environments such as an atrium. In summary, the CRF-based systems need shorter convergence distance than the PF-based systems do. Meanwhile, the encounter mechanism significantly help CRF-based localization to reduce the convergence distance.

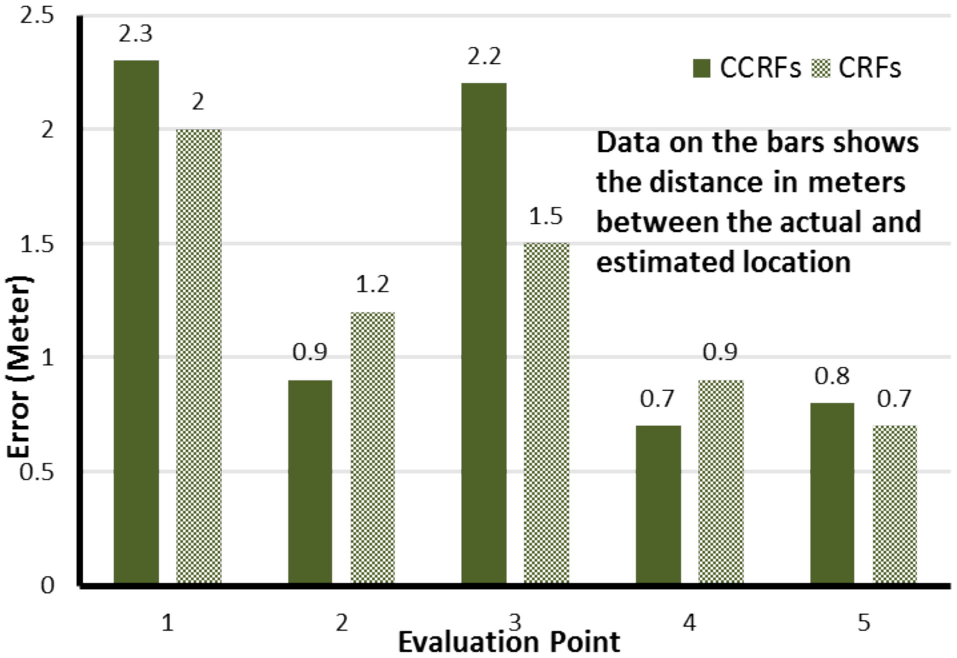


Figure 7.6: The Comparison of Accuracy for WICED-Sense among Five Evaluation Points.

7.2.2 Accuracy

We quantify accuracy by measuring how close to the ground truth location each technique can estimate. Our experiments are conducted in different indoor environments with multiple participants to evaluate the accuracy of CRF-based and PF-based localization.

On IoT Platform

The results we collected for IoT devices are shown in Fig. 7.6. The results for CCRFs and CRFs are similar, where the error difference is within one step length. The accuracy of CCRF is sometimes worse than CRF, primarily due to the noisy sensor data that cannot be removed by exchanging trajectories. In addition, without sufficient observations such as RSSI fingerprints or locations of beacons, the merged trajectories may accumulate more errors that result from PDR and worsen the accuracy of estimation.

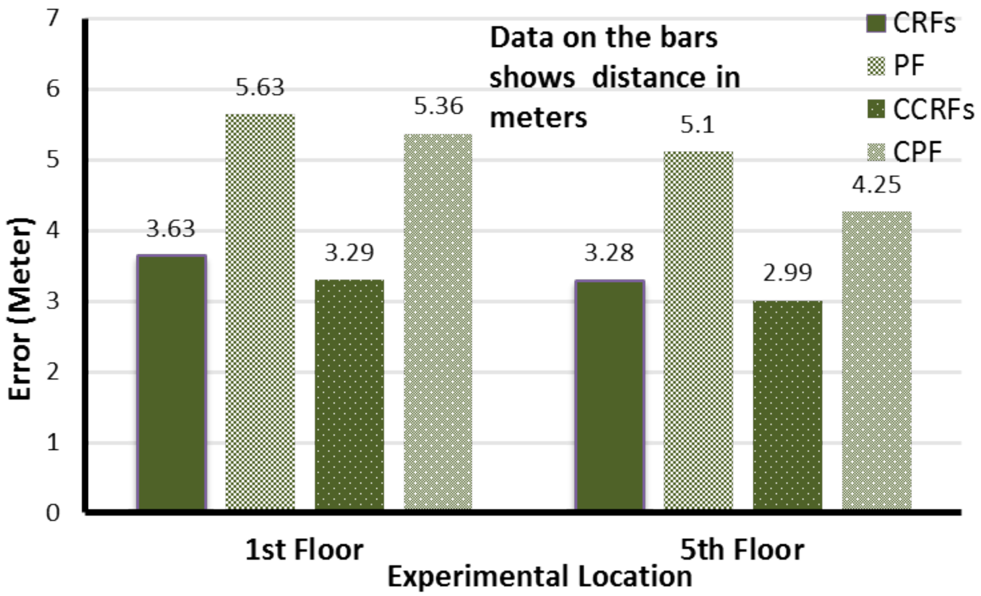


Figure 7.7: The Comparison of Average Accuracy for Tablet on the 1st and 5th Floors without Beacons

On Tablet

Fig. 7.7 compares the average accuracy of these localization techniques on the tablet on the 1st and 5th floors of the campus Building. Overall, the CCRF and CRF can provide the same level of accurate results and similar results as we have seen on the IoT platform. The CRF-based systems are also more accurate than the PF-based systems.

Noise Tolerance

The results of individual evaluation points on the 5th floor are shown in Fig. 7.8. We observe that at the evaluation point 6, PF is more accurate than CRF. This is because most of the participants walk diagonally to reach this evaluation point. Since we assume the step length is a fix value when generating the state map, the diagonal distance is longer than the distance between two states. The error is thus accumulated and worsens the accuracy. PF is not influenced by the diagonal walking because it is not restricted to the state transition, and it utilizes the sampling particles to explore the

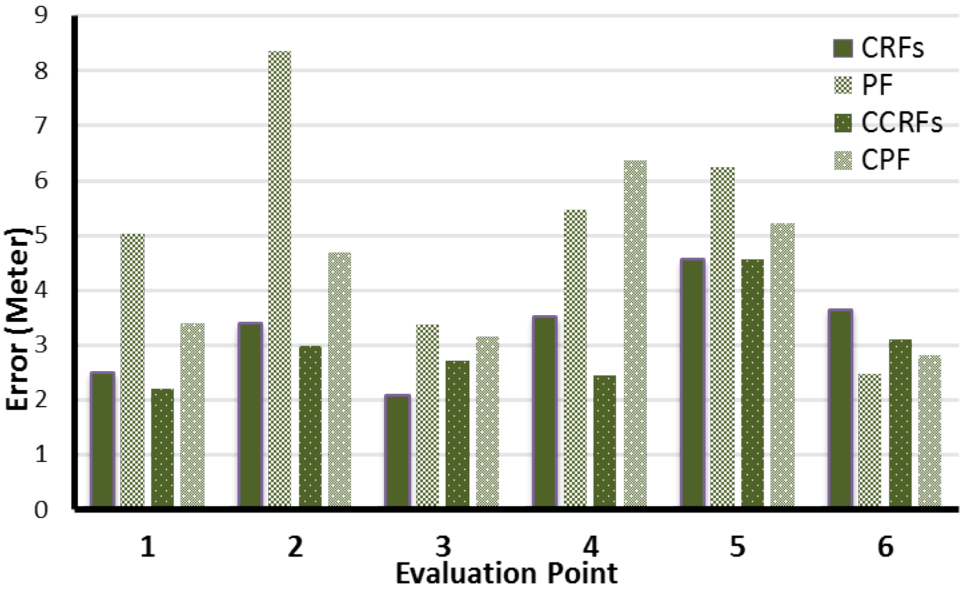


Figure 7.8: The Comparison of Accuracy for Tablet on the 5th Floor without Beacons

possible location. At evaluation points 2 and 5, due to the magnetic field noise caused by the metal building materials, the magnetometer sensor data worsens the accuracy of all systems, especially PF. It shows that CRF-based systems have higher tolerance to the sensor noise than PF-based ones.

The results of individual evaluation points on the 1st floor are shown in Fig. 7.9. It is a relatively open, unconstrained space. In the experiment, the participants walk randomly to reach the evaluation points. Similar to the 5th floor, the CRF-based systems provide less accurate estimation due to the frequently diagonal movement. In addition, the results of PF are worse because we use Gaussian distribution at the Importance Weighting step. When we resample the particles, a small subset should be chosen and assigned different heading orientations based on the Gaussian distribution for error modeling in each iteration. Without floor-plan constraints in the open space, every particle is always active such that more particles accumulate errors after several iterations.

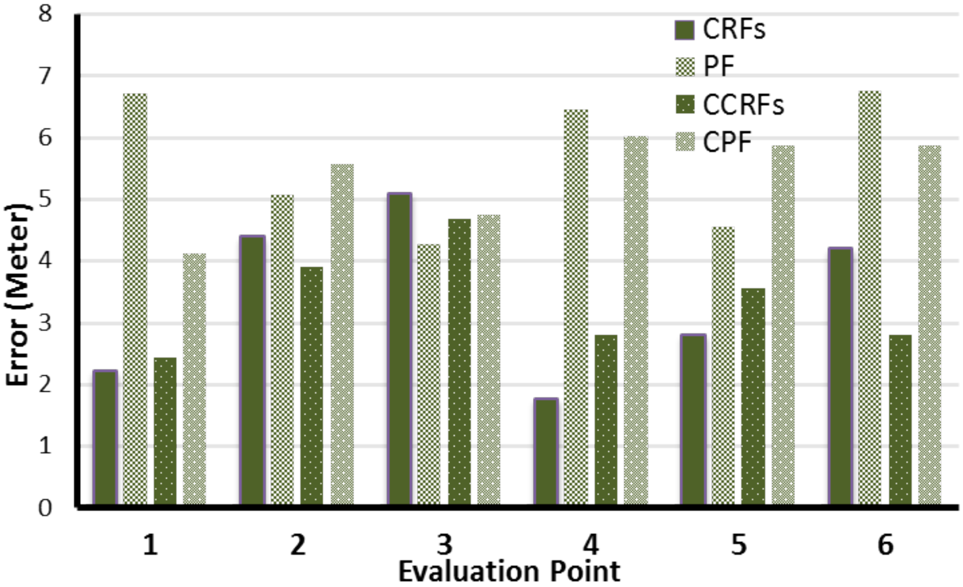


Figure 7.9: The Comparison of Accuracy for Tablet on the 1st Floor without Beacons

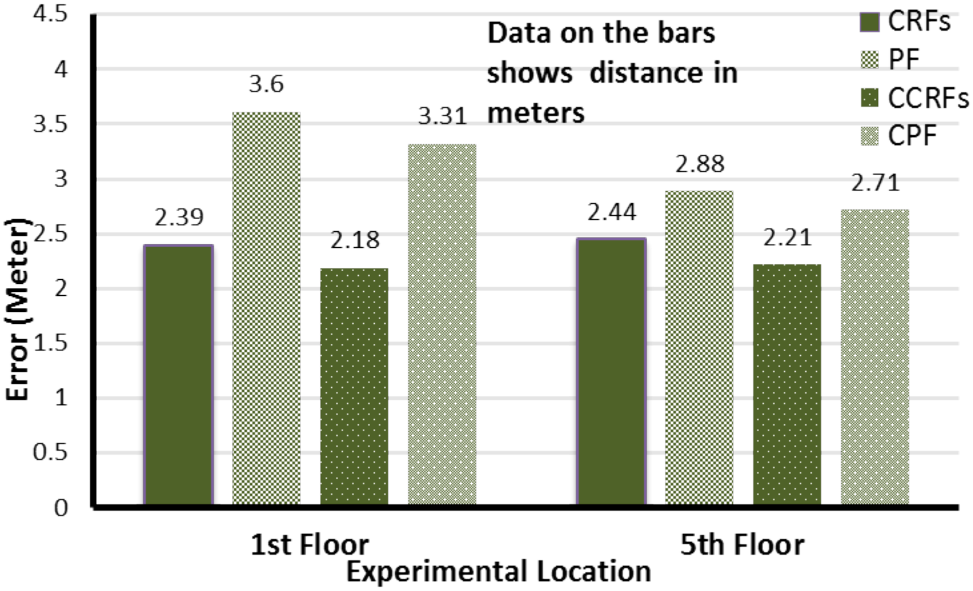


Figure 7.10: The Comparison of Average Accuracy for Tablet on the 1st and 5th Floors with Beacons

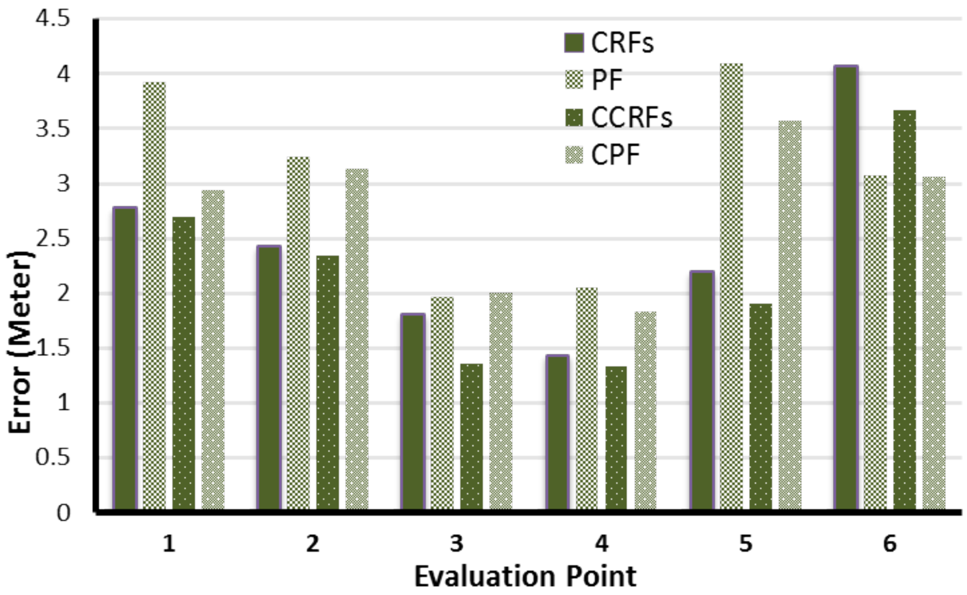


Figure 7.11: The Comparison of Accuracy for Tablet on the 5th Floor with Beacons

With Beacons

The results of the techniques with beacon assistance are shown in Fig. 7.10. As explained in the section on the encounter model, due to the limited packets size, the beacon information is used only by the individual device for estimation but not shared with other users.

The estimated errors of all of the systems are all improved, since the beacons provide the extra reference for error correction. Again, we check the result at evaluation point 6 on the 5th floor shown in Fig. 7.11, which suffers from the error caused by the diagonal walking. Though the beacons can improve the accuracy, it does not work perfectly to remove the accumulated error from the diagonal walking. On the other hand, with the help of beacons, the estimated results at the evaluation point 5 is significantly improved since the system can infer that the user location is in the conference room. Fig. 7.12 shows the experimental result collected on the 1st floor, and the accuracy is also improved by the reference of the beacons. Note that only 5 beacons are used in the experiment, and we expect the accuracy to be further improved by deploying more beacons.

In summary, a CRF-based system can provide more accurate estimated results than a PF-based sys-

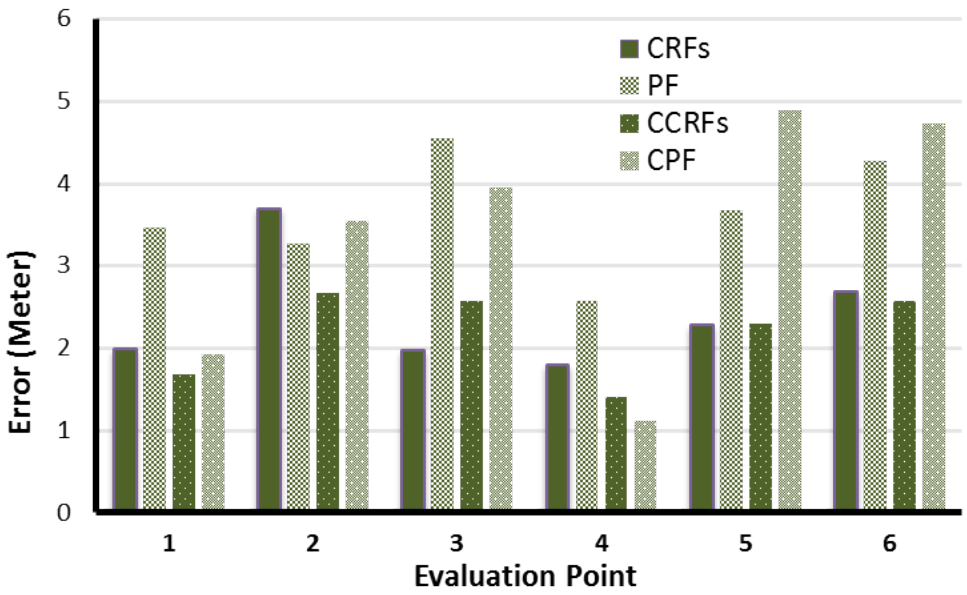


Figure 7.12: The Comparison of Accuracy for Tablet on the 1st Floor with Beacons

tem can. A CRF-based system has better performance with or without the assistance of constraints. Although the diagonal movement decreases the accuracy, the error can be bounded using the fingerprinting technique such as the beacons we use in the experiment. Meanwhile, a CRF-based system has higher resistance to the sensor noise compared to a PF system in the same environment. However, the information estimated from the dead reckoning technique directly influences the accuracy of all the system.

7.2.3 Algorithm Overhead

The overhead is evaluated by measuring the power consumption of our implementation, since the APIs for timing measurement are not open-source, and we are unable to instrument the system software. We observe the overhead mostly coming from the pairing behavior (scanning and advertising).

We compare the time complexity of the CRF-based system and the CPF system running on the tablet. The CPF requires 35.36 ms for the processing in each steps, while the CCRF spends only

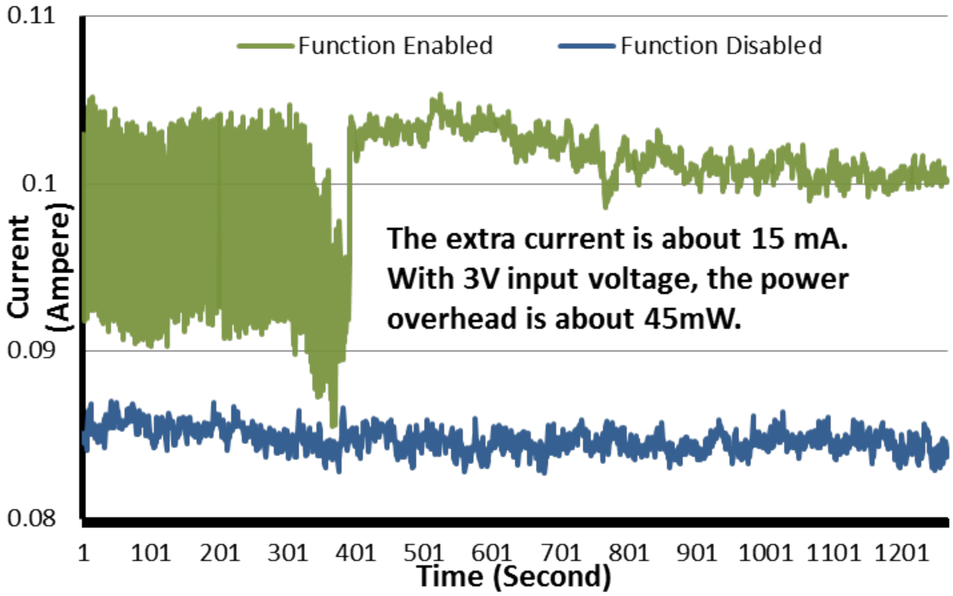


Figure 7.13: Overhead of Power Consumption for WICED-Sense.

4.56 ms. Fig. 7.14 shows that the execution time when the encounter events occur. Although the CPF can execute the process within 20 ms, which is much lower than CCRF (215.44 ms), it sacrifices the accuracy and convergence distance that we evaluate in the previous sections for the lower time complexity.

7.3 Summary

The knowledge of location is fundamental to intelligent behavior of interconnected devices. Currently, location is a privilege afforded mainly by higher-end systems equipped with GPS, cameras, or acoustic, or other ranging devices that are bulky, costly, or consume relatively high power. Low-end sensor nodes rely on either specialized devices or infrastructure support.

To make location knowledge a universal right of all IoT devices, not just the resource-rich ones, we propose EcoLoc, which requires only low-end processing capabilities and low-power inertial sensors in packages as small as 2×2 mm^2. It exploits sharing of trajectories between devices upon encounter as a way to estimate the location on the map while requiring up to 50% shorter con-

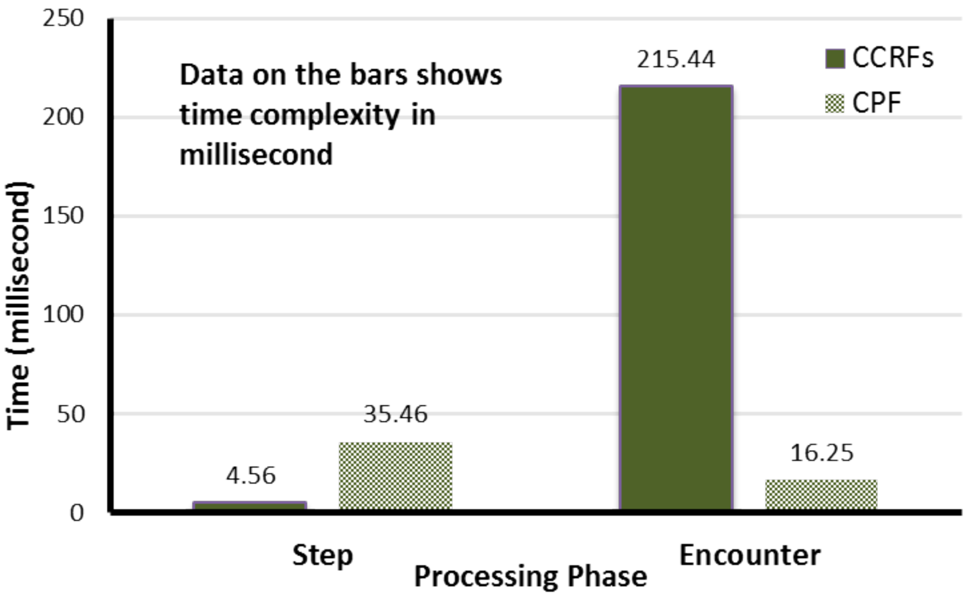


Figure 7.14: Comparison of Time Complexity between CCRF and CPF

vergence distance than previous techniques. The computation effort is also reduced significantly, thanks to the use of our CCRF model. The use of BLE makes it practical because it uses the existing RF interface for not only communication but also proximity sensing and encounter exchange. The overhead we measured on IoT device in terms of current consumption of 15 mA.

Chapter 8

Conclusions

This dissertation has presented collaborative sensing approaches to solving the position inference problems. This chapter concludes this dissertation with a summary and research directions for future work.

8.1 Summary

Collaborative sensing approaches are proposed and implemented in two position sensing systems. This dissertation consists of two parts, Part I presents a framework for chest compression depth estimation, and part II proposed a collaborative approach for indoor localization.

8.1.1 Chest Compression Depth Estimation

A collaborative sensing system is proposed to provide real-time chest compression depth feedback. We exploit the rotational motion to enable the collaborative sensing by using two IMUs. One IMU is placed on the center of chest to measure the acceleration and the other one is placed 2 inches away from the center to measure the angular velocity. Kalman filter and complementary filter are used for depth estimation. The experimental results show our proposed compression depth estimation method provides more accurate results than the comparing method. Our proposed method is lightweight and designed for resource-limited IoT devices.

The contribution of our proposed collaborative position-sensing framework for real-time CCD estimation includes

- A novel chest-compression model for a data fusion algorithm to handle cumulative error in depth estimation.
- A mutual-information-based evaluation method for IoT device placement to obtain qualified sensor data.

8.1.2 Collaborative Indoor Localization

A collaborative indoor localization system is proposed to enable universal location sensing among IoT devices by utilizing the encounter events. The BLE advertising protocol is exploit for proximity sensing and the advertising packet contains the user trajectory for data exchange. A collaborative conditional random field is proposed to merge the received trajectories and generate the most probable locations while significantly shortening the convergence distance compared to the state-of-the-art techniques. Our collaborative position sensing runs in real-time and can be distributed to resource-limited IoT devices.

The contribution of our collaborative indoor localization system includes

- A power-efficient encounter model to discover nearby devices and exchange data
- A collaborative data fusion algorithm to enable universal location sensing by all mobile IoT devices

8.2 Future Work

Many research opportunities still remain for further improvement and exploration in our proposed position sensing system. Research directions for future work based on our proposed system are provided.

8.2.1 Chest Compression Depth Estimation

Our rotational model assumes that the z-axis velocity can represent the tangential velocity. An orientation estimation model to find the tangential component of the velocity can be developed to improve the accuracy of rotation radius estimation. Our depth estimation method assumes that the sensor is placed on a flat surface. The algorithm can be further improved by integrating an advanced orientation estimation system. The statistical model can be used to replace our rotational model but the acquirement of training data may be a challenge. The clinical trial is essential to the medical system design and also provides the opportunities to collect the training data. However, CPR is not a general procedure can be perform on real people. Therefore, the study on data collection is a potential research topic to develop the statistic-based method.

8.2.2 Collaborative Indoor Localization

Our work currently assumes the step length is fixed and the encounter event happened when the distance between two users are within one step length. A more general step-length estimation model can be added and the encounter event can be further defined in term of distance to improve the accuracy. In addition, since the trajectory is broadcast using advertising packet, the privacy and security issues should be considered. User nehavior can be infereed by analyzing the shared trajectory. Thus, a randomly generated, single-use identifier such as randomized MAC address used in iBeacons can be considered to protect the privacy.

Bibliography

[1] A survey of deep neural network architectures and their applications. *Neurocomputing*, 234: 11–26, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.12.038. URL `https://www.sciencedirect.com/science/article/pii/S0925231216315533`.

[2] Deep-ecg: Convolutional neural networks for ecg biometric recognition. *Pattern Recognition Letters*, 126:78–85, 2019. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2018.03.028. URL `https://www.sciencedirect.com/science/article/pii/S0167865518301077`. Robustness, Security and Regulation Aspects in Current Biometric Systems.

[3] S. O. Aase and H. Myklebust. Compression depth estimation for CPR quality assessment using DSP on accelerometer signals. *IEEE Transactions on Biomedical Engineering*, 49(3): 263–268, 2002. doi: 10.1109/10.983461.

[4] B. Abella. The importance of cardiopulmonary resuscitation quality. *Current opinion in critical care*, 19:175–80, 2013.

[5] F. Al-Turjman. Positioning in the internet of things era: An overview. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–5, 2017. doi: 10.1109/ICEngTechnol.2017.8308188.

[6] E. Alonso, D. González-Otero, E. Aramendi, S. R. de Gauna, J. Ruiz, U. Ayala, J. K. Russell, and M. Daya. Can thoracic impedance monitor the depth of chest compressions during out-of-hospital cardiopulmonary resuscitation? *Resuscitation*, 85:637–643, 2014.

[7] M. Alzantot and M. Youssef. UPTIME: Ubiquitous pedestrian tracking using mobile phones. In *IEEE Wireless Communications and Networking Conference, (WCNC'12)*, pages 3204–3209, 2012.

[8] American Heart Association (AHA). 2020 American Heart Association Guidelines for Cardiopulmonary Resuscitation and Emergency Cardiovascular Care. `https://professional.heart.org/en/science-news/2020-aha-guidelines-for-cpr-and-ecc`.

[9] E. Aramendi, U. Ayala, U. Irusta, E. Alonso, T. Eftestøl, and J. Kramer-Johansen. Suppression of the cardiopulmonary resuscitation artefacts using the instantaneous chest compression rate extracted from the thoracic impedance. *Resuscitation*, 83:692–698, 2012.

[10] H. Bao and W.-C. Wong. An indoor dead-reckoning algorithm with map matching. In *International Wireless Communications and Mobile Computing Conference, (IWCMC'13)*, pages 1534–1539, 2013.

[11] S. Boussen, H. Ibouanga-Kipoutou, N. Fournier, Y. G. Raboutet, M. Llari, N. Bruder, P. J. Arnouxa, and M. Behr. Using an inertial navigation algorithm and accelerometer to monitor chest compression depth during cardiopulmonary resuscitation. *Medical Engineering & Physics*, 38:1028–1034, 2016.

[12] Broadcom. WICED Sense Bluetooth Smart sensor tag / experience kit. `https://www.broadcom.com/products/wireless-connectivity/bluetooth/wiced-sense`, 2015.

[13] J. Cao and J. Wang. Stock price forecasting model based on modified convolution neural network and financial time series analysis. *International Journal of Communication Systems*, 32(12):e3987, 2019.

[14] Z. Chen, 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 localization. *Sensors*, 15(1):715–732, 2014.

[15] A. Cheng and et al. Resuscitation education science: Educational strategies to improve outcomes from cardiac arrest: A scientific statement from the American Heart Association. *Circulation*, 138:e82–e112, 2018.

[16] A. Colombo, D. Fontanelli, D. Macii, and L. Palopoli. Flexible indoor localization and tracking based on a wearable platform and sensor data fusion. *IEEE Transactions on Instrumentation and Measurement, (TIM'14)*, 63(4):864–876, 2014.

[17] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see Bob?: Human localization using mobile phones. In *International Conference on Mobile Computing and Networking, (MobiCom'10)*, pages 149–160, 2010.

[18] C. Dine, R. Gersh, M. Leary, B. Riegel, L. Bellini, and B. Abella. Improving cardiopulmonary resuscitation quality and resuscitation training by combining audiovisual feedback and debriefing. *Crit Care Med*, 36:2817–22, 2008.

[19] M. Dolgov, N. Lentz, S. Fernsner, and A. Bolz. Detection of chest compression depth with intracorporeal ultrasound travel time measurement. *5th European Conference of the International Federation for Medical and Biological Engineering*, 37:1031–1034, 2011.

[20] D. Edelson, B. Litzinger, V. Arora, D. Walsh, S. Kim, D. Lauderdale, T. Vanden Hoek, L. Becker, and B. Abella. Improving in-hospital cardiac arrest process and outcomes with performance debriefing. *Archives of internal medicine*, 168:1063–9, 2008.

[21] V. Filonenko, C. Cullen, and J. D. Carswell. Indoor positioning for smartphones using asynchronous ultrasound trilateration. *ISPRS International Journal of Geo-Information*, pages 598–620, 2013.

[22] G. D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[23] D. M. González-Otero, J. Ruiz, S. R. de Gauna, U. Irusta, U. Ayala, and E. Alonso. A new method for feedback on the quality of chest compressions during cardiopulmonary resuscitation. *BioMed Research International*, 2014, 2014.

[24] X. Guo, N. Ansari, F. Hu, Y. Shao, N. R. Elikplim, and L. Li. A survey on fusion-based indoor positioning. *IEEE Communications Surveys Tutorials*, 22(1):566–594, 2020. doi: 10.1109/COMST.2019.2951036.

[25] J. Hannink, T. Kautz, C. F. Pasluosta, K.-G. Gaßmann, J. Klucken, and B. M. Eskofier. Sensor-based gait parameter extraction with deep convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*, 21(1):85–93, 2017. doi: 10.1109/JBHI.2016.2636456.

[26] M. Hardegger, D. Roggen, S. Mazilu, and G. Tröster. Actionslam: Using location-related actions as landmarks in pedestrian slam. In *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10, 2012. doi: 10.1109/IPIN.2012.6418932.

[27] R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys & Tutorials*, 15(3):1281–1293, 2013.

[28] M. Hoang, J. Schmalenstroeer, C. Drueke, D. Tran Vu, and R. Haeb-Umbach. A hidden markov model for indoor user tracking based on wifi fingerprinting and step detection. pages 1–5, 2013.

[29] M. Hofmann, J. C. Edelmann, A. Bolz, R. Weigel, G. Fischer, and D. Kissinger. Rf based feedback system for cardiopulmonary resuscitation. *IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireleSS)*, 2012.

[30] E. Hunt, J. Jeffers, L. McNamara, H. Newton, K. Ford, M. Bernier, E. Tucker, K. Jones, C. O'Brien, P. Dodge, S. Vanderwagen, C. Salamone, T. Pegram, M. Rosen, H. Griffis, and J. Duval-Arnould. Improved cardiopulmonary resuscitation performance with CODE ACES2: A resuscitation quality bundle. *Journal of the American Heart Association*, 7, 2018.

[31] iRhythm Technologies. iRhythm Technologies, Inc, Zio monitor. `https://www.irhythmtech.com/products-services/zio-at/`.

[32] F. Izquierdo, M. Ciurana, F. Barcelo, J. Paradells, and E. Zola. Performance evaluation of a TOA-based trilateration method to locate terminals in WLAN. *International Symposium on Wireless Pervasive Computing*, pages 1–6, 2006.

[33] H. Jin, I. Kang, G. Choi, D. D. Molinaro, and A. J. Young. Wearable sensor-based step length estimation during overground locomotion using a deep convolutional neural network. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 4897–4900, 2021. doi: 10.1109/EMBC46164.2021.9630060.

[34] S. Y. Jin, S. B. Oh, and Y. O. Kim. Estimation of optimal pediatric chest compression depth by using computed tomography. *Clin Exp Emerg Med*, 3:27–33, 2016.

[35] J. Jun, Y. Gu, L. Cheng, B. Lu, J. Sun, T. Zhu, and J. Niu. Social-Loc: improving indoor localization with social sensing. In *ACM Conference on Embedded Networked Sensor Systems, (SenSys'13)*, pages 1–14, 2013.

[36] W. Kang and Y. Han. SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal*, 15(5):2906–2916, 2015.

[37] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53(8):5455–5516, 2020.

[38] Y.-K. Kim, S.-H. Choi, H.-W. Kim, and J.-M. Lee. Performance improvement and height estimation of pedestrian dead-reckoning system using a low cost MEMS sensor. In *International Conference on Control, Automation and Systems, (ICCAS'12)*, pages 1655–1660, 2012.

[39] Z. Kowalczuk and T. Merta. Evaluation of position estimation based on accelerometer data. In *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*, pages 246–251, 2015. doi: 10.1109/RoMoCo.2015.7219743.

[40] R. Kronberger, T. Knie, R. Leonardi, U. Dettmar, M. Cremer, and S. Azzouzi. UHF RFID localization system based on a phased array antenna. *2011 IEEE International Symposium on Antennas and Propagation (APSURSI)*, pages 525–528, 2011.

[41] J. Kunhoth, A. Karkar, S. Al-Maadeed, and A. Al-Ali. Indoor positioning and wayfinding systems: a survey. *Human-centric Computing and Information Sciences*, 10(1), 2020.

[42] Laerdal Medical. Laerdal Medical, CPRmeter 2. `https://www.laerdal.com/us/products/medical-devices/cprmeter-2/`.

[43] H. Leppäkoski, J. Collin, and J. Takala. Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP'12)*, pages 1569–1572, 2015.

[44] R. W. Levi and T. Judd. Dead reckoning navigational system using accelerometer to measure foot impacts, December 1996.

[45] K. Lin, W. Wang, Y. Bi, M. Qiu, and M. M. Hassan. Human localization based on inertial sensors and fingerprint in industrial internet of things. *Computer Networks*, 15(1):715–732, 2015.

[46] T.-C. Lu, Y. Chen, T.-W. Ho, Y.-T. Chang, Y.-T. Lee, Y.-S. Wang, Y.-P. Chen, C.-M. Fu, W.-C. Chiang, M. H.-M. Ma, C.-C. Fang, F. Lai, and A. M. Turner. A novel depth estimation algorithm of chest compression for feedback of high-quality cardiopulmonary resuscitation based on a smartwatch. *Journal of Biomedical Informatics*, 87:60 – 65, 2018.

[47] D. Macagnano, G. Destino, and G. Abreu. Indoor positioning: A key enabling technology for iot applications. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 117–118, 2014. doi: 10.1109/WF-IoT.2014.6803131.

[48] S. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25:113–118, 2010.

[49] P. Meaney, B. Bobrow, M. Mancini, J. Christenson, A. de Caen, F. Bhanji, B. Abella, M. Kleinman, D. Edelson, R. Berg, T. Aufderheide, V. Menon, and M. Leary. Cardiopulmonary resuscitation quality: [corrected] improving cardiac resuscitation outcomes both inside and outside the hospital: a consensus statement from the American Heart Association. *Circulation*, 128:417–35, 2013.

[50] K. Minami, Y. Kokubob, I. Maeda, and S. Hibino. A flexible pressure sensor could correctly measure the depth of chest compression on a mattress. *The American Journal of Emergency Medicine*, 34:899–902, 2016.

[51] H. Mora, M. T. Signes-Pont, D. Gil, and M. Johnsson. Collaborative working architecture for iot-based applications. *Sensors*, 18(6), 2018. ISSN 1424-8220. URL `https://www.mdpi.com/1424-8220/18/6/1676`.

[52] L. Morrison, R. Neumar, J. Zimmerman, M. Link, L. Newby, P. McMullan Jr., T. Hoek, C. Halverson, L. Doering, M. Peberdy, and D. Edelson. Strategies for improving survival after in-hospital cardiac arrest in the United States: 2013 consensus recommendations: a consensus statement from the American Heart Association. *Circulation*, 127:1538–63, 2013.

[53] R. Neumar, C. Otto, M. Link, S. Kronick, M. Shuster, C. Callaway, P. Kudenchuk, J. Ornato, B. McNally, S. Silvers, R. Passman, R. White, E. Hess, W. Tang, D. Davis, E. Sinz, and L. Morrison. Part 8: adult advanced cardiovascular life support: 2010 American Heart Association Guidelines for Cardiopulmonary Resuscitation and Emergency Cardiovascular Care. *Circulation*, 122:S729–67, 2010.

[54] J. Oh, Y. Song, B. Kang, H. Kang, T. Lim, Y. Suh, and Y. Chee. The use of dual accelerometers improves measurement of chest compression depth. *Resuscitation*, 83:500–504, 2012.

[55] G. Pang and H. Liu. Evaluation of a low-cost mems accelerometer for distance measurement. *Journal of Intelligent and Robotic Systems*, 30:249–265, 2001.

[56] M. Peberdy, J. Ornato, G. Larkin, R. Braithwaite, T. Kashner, S. Carey, P. Meaney, L. Cen, V. Nadkarni, A. Praestgaard, and R. N. R. o. C. R. I. Berg. Survival from in-hospital cardiac arrest during nights and weekends. *JAMA*, 299:785–92, 2008.

[57] D. D. Pham and Y. S. Suh. Pedestrian navigation using foot-mounted inertial sensor and LIDAR. *Sensors*, 16(1):120–126, 2016.

[58] A. R. Pratama, Widyawan, and R. Hidayat. Smartphone-based pedestrian dead reckoning as an indoor positioning system. In *International Conference on System Engineering and Technology, (ICSET'15)*, pages 1–6, 2012.

[59] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *International Conference on Mobile Computing and Networking, (Mobicom'12)*, pages 293–304, 2012.

[60] A. R. J. Ruiz, F. S. Granja, J. C. P. Honorato, and J. I. G. Rosas. Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements. *IEEE Transactions on Instrumentation and Measurement, (TIM'12)*, 61(1):178–189, 2012.

[61] S. Ruiz de Gauna, D. M. González-Otero, J. Ruiz, and J. K. Russell. Feedback on the rate and depth of chest compressions during cardiopulmonary resuscitation using only accelerometers. *PLOS ONE*, 11(3):1–17, 03 2016.

[62] F. Semeraro, A. Frisoli, C. Loconsole, F. oBannò, G. Tammaro, G. Imbriaco, L. Marchetti, and E. L.Cerchiari. Motion detection technology as a tool for cardiopulmonary resuscitation (cpr) quality training: A randomised crossover mannequin pilot study. *Resuscitation*, 84: 501–507, 2013.

[63] M. Serrano, H. N. M. Quoc, M. Hauswirth, W. Wang, P. Barnaghi, and P. Cousin. Open services for IoT cloud applications in the future Internet. In *IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks, (WoWMoM'13)*, pages 1–6, 2013.

[64] M. Sharifi Renani, C. Myers, R. Zandie, M. Mahoor, B. Davidson, and C. Clary. Deep learning in gait parameter prediction for oa and tka patients wearing imu sensors. *Sensors*, 2020. URL `https://doi.org/10.3390/s20195553`.

[65] Y. Shu, Y. Huang, J. Zhang, P. Coué, P. Cheng, J. Chen, and K. G. Shin. Gradient-based fingerprinting for indoor localization and tracking. *IEEE Transactions on Industrial Electronics*, 63:2424–2433, 2016.

[66] J. Soar, D. Edelson, and G. Perkins. Delivering high-quality cardiopulmonary resuscitation in-hospital. *Current opinion in critical care*, 17:225–30, 2011.

[67] W. Sui and K. Wang. An accurate indoor localization approach using cellphone camera. pages 949–953, 2015.

[68] A. Thaljaoui, T. Val, N. Nasri, and D. Brulin. BLE localization using RSSI measurements and iRingLA. *IEEE International Conference on Industrial Technology (ICIT)*, pages 2178–2183, 2015.

[69] M. Thoma, S. Meyer, K. Sperner, S. Meissner, and T. Braun. On iot-services: Survey, classification and enterprise integration. In *2012 IEEE International Conference on Green Computing and Communications*, pages 257–260, 2012. doi: 10.1109/GreenCom.2012.47.

[70] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Ström. Cooperative received signal strength-based sensor localization with unknown transmit powers. *IEEE Transactions on Signal Processing, (SP'13)*, 61(6):1389–1403, 2013.

[71] H. Wang, K. Yao, and D. Estrin. Information-theoretic approaches for sensor selection and placement in sensor networks for target localization and tracking. *Journal of Communications and Networks*, 7(4):438–449, 2005. doi: 10.1109/JCN.2005.6387986.

[72] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: unsupervised indoor localization. In *International Conference on Mobile Systems, Applications, and Services, (MobiSys'12)*, pages 197–210, 2012.

[73] J. Wang. Stochastic modeling for real-time kinematic gps/glonass positioning. *NAVIGATION, Journal of The Institute of Navigation*, 46(4):297–305, 1999.

[74] X. Wang, L. Gao, S. Mao, and S. Pandey. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66:763–776, 2017.

[75] H. Weinberg. Using the adxl 202 in pedometer and personal navigation applications. 2002.

[76] Z. Wu, E. Jedari, R. Muscedere, and R. Rashidzadeh. Improved particle filter based on wlan rssi fingerprinting and smart sensors for indoor localization. *Computer Communications*, 83: 64–71, 2016.

[77] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. Lightweight map matching for indoor localisation using conditional random fields. In *Information Processing in Sensor Networks, (IPSN'14)*, pages 131–142, 2014.

[78] J. Xu, W. Liu, F. Lang, Y. Zhang, and C. Wang. Distance measurement model based on RSSI in WSN. *Wireless Sensor Network*, 2(8):606–611, 2010.

[79] B. G. Yu, J. H. Oh, Y. Kim, and T. W. Kim. Accurate measurement of chest compression depth using impulse-radio ultra-wideband sensor on a mattress. *PLoS One*, 12:1–8, 2017.

[80] F. Zafari, I. Papapanagiotou, M. Devetsikiotis, and T. Hacker. An iBeacon based proximity and indoor localization system. *CoRR*, abs/1703.07876, 2017.

[81] C. Zebuhr, R. Sutton, W. Morrison, D. Niles, L. Boyle, A. Nishisaki, P. Meaney, J. Leffelman, R. Berg, and V. Nadkarni. Evaluation of quantitative debriefing after pediatric cardiac arrest. *Resuscitation*, 83:1124–8, 2012.

[82] L. Zhao, Y. Bao, R. Ye, A. Zhang, and Y. Zhang. A classification method for judging the depth of chest compression based on cnn. 06 2020. doi: 10.21203/rs.3.rs-38347/v1.

[83] Zoll Medical Corporation. Zoll Medical Corporation, PocketCPR. `http://www.pocketcpr.com/iphone.html/`.

[84] Øyvind Meinich-Bache, K. Engan, T. S. Birkenes, , and H. Myklebust. Real-time chest compression quality measurements by smartphone camera. *Journal of Healthcare Engineering*, 2018:1–12, 2018.