A Framework for Associating Mobile Devices to Individuals Based on Identification of Motion Events

by

Madi Zhanbyrtayev

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

The ubiquity of the Internet-of-Things (IoT) devices in everyday life allows various sensors to be utilized in networked systems for solving a number of real-world problems. Models utilizing specific sensing modalities achieve impressive performance in understanding human activity and are used in systems developed for monitoring and improving indoor living conditions. A combination of multiple sensors could even allow a better understanding of the environment. Nevertheless, certain sensing modalities may not have a direct correlation in their measurements, hence, making the fusion of the sensor data quite challenging. This thesis studies the feasibility and design of a sensor fusion system that can associate two unrelated sensing modalities, namely radio frequency and visual domains, by identifying and associating events, human motion, that leaves a signature in both domains.

We present a holistic framework for associating a mobile device unique identifier to an individual holding it during a certain activity. We study different motion detection methods that rely on the analysis of Received Signal Strength Identifier (RSSI) combined with state-of-the-art Computer Vision approaches to object tracking. We run field experiments to evaluate the performance of different motion detection methods and use the proposed framework to associate mobile devices to individuals who hold or carry them. Our results indicate that an accuracy of 75% can be achieved in the device-to-individual association task.

Preface

Some parts of the thesis were a result of a collaboration with students undergoing the High School Internship Program (HIP) at the University of Alberta. The data collection experiments described in Chapter 3 were conducted with help from the students, Celina Sheng, Grace Nguyen, Isabella Ng, Kaitlyn Stark and Ananya Nandiraju. Throughout the thesis, the students helped with the data collection experiments and ground truth annotation described in Chapter 3. The framework structure described in Chapters 4 and 5 was designed by myself, with the assistance of Dr. Ardakanian and Dr. Nikolaidis. Literature review in Chapter 2, results and conclusions in Chapters 6 and 7 are my original work.

Section 4.2.1 of this thesis was published as [55]. I was responsible for data analysis and collection as well as the paper composition. Dr. Ardakanian and Dr. Nikolaidis edited the final paper draft and provided supervision during the project. Other sections in Chapter 4 are to be submitted for a journal publication.

Acknowledgements

First, I would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC) for providing funding for my thesis research through the Discovery program.

I am grateful to my supervisors, Dr. Omid Ardakanian and Dr. Ioanis Nikolaidis, for constant support, guidance and patience throughout this thesis. They motivated me to become a better researcher, question critically every approach and think creatively in solving problems. Their help was essential in reaching a successful thesis defence.

I would like to acknowledge the incredible work of the participants of the HIP program at UofA. I thank Celina, Grace, Isabella, Kaitlyn and Ananya for all the hard work they put in and curiosity they had during our work together.

I would like to thank all of my colleagues and friends from the Networks Group and Sustainable Computing Research Labs. Their feedback and shared knowledge were important to me at all stages of my research. I also appreciate all the support and inspiration they gave me.

My friends from around the world in Edmonton, Astana, Budapest, Menlo Park, Los Angeles, Saint-Petersburg, Moscow, New York, Chicago, Munich, Toronto, Vancouver, Montreal and Almaty always shared their support during these years of studying. Thank you all for that.

Finally, I thank my family for everything they've done for me to have this wonderful opportunity. None of this would be possible without them.

Thank you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The emergence of the Internet-of-Things (IoT) devices has introduced a large number of networked systems with various sensor modalities. Real world problems such as minimizing energy usage, monitoring environmental conditions, food tracking, transportation planning, etc. can be studied and supported using this technology. An IoT device can report measurements from the Inertial Measurement Unit (IMU) – magnetometer and accelerometer – temperature sensor, light sensor, and infrared sensor. Together these measurements can be utilized, for example, in human activity tracking and recognition systems. Various sensors available at an affordable price can be also deployed in an indoor environment for monitoring and efficient control of building subsystems.

Mobile devices equipped with a myriad of such sensors have become prevalent, examples of which are smartphones and wearables which are affordable, sensor-rich, and small in size. These devices can be used to track and localize people, enabling a wide variety of applications, such as activity monitoring, occupancy estimation, and people identification [11], [26], [28], [54]. Sensors can also be used for safety and security applications, such as intrusion detection or natural disaster notification.

While the variety in sensing devices allows to collect measurements at different rates, quality and quantity, it raises the challenge of data fusion as some sensing modalities cannot be trivially fused. A conventional sensor fusion combines various sensor measurements to improve performance or gather additional information about an environment. In this thesis we investigate

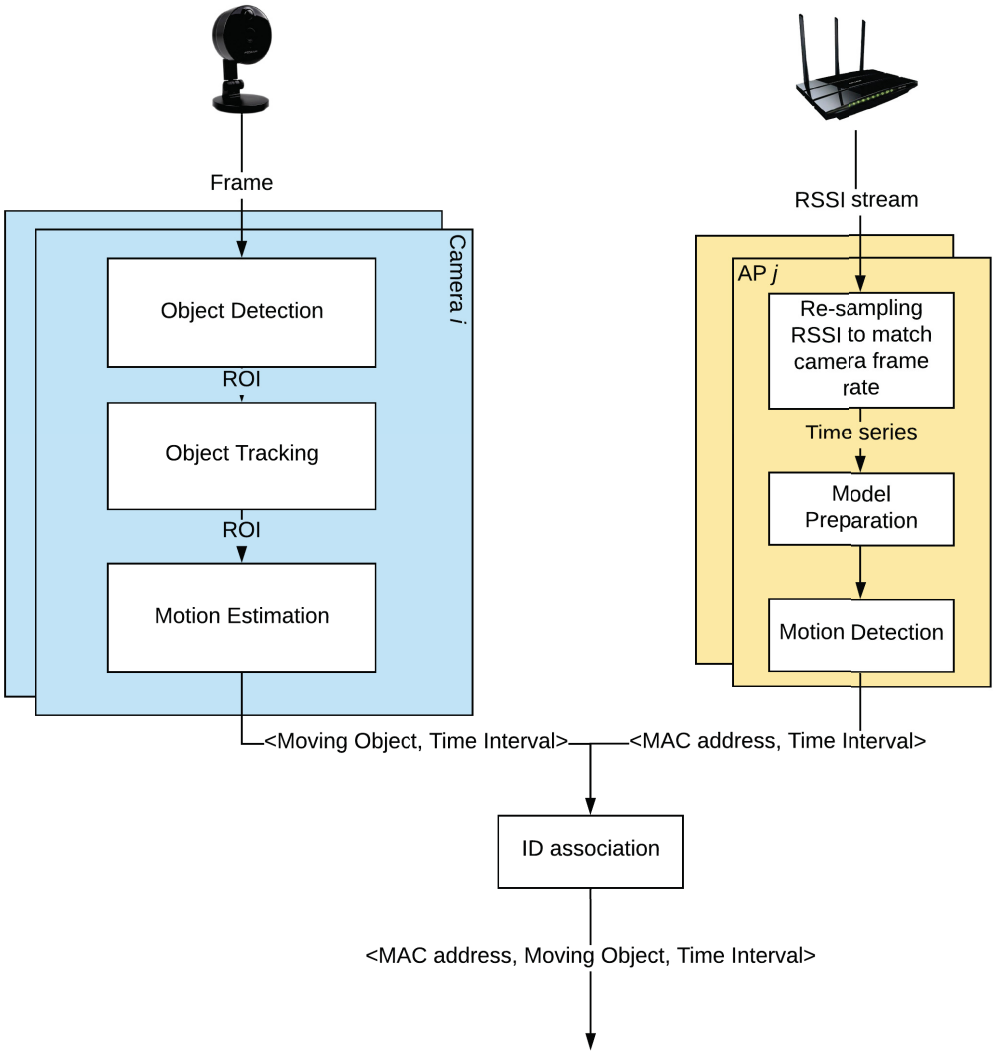

Figure 1.1: General framework overview.

how to reliably detect motion in an indoor environment by fusing video and RSSI data obtained from one or multiple sensing nodes. We build and evaluate a system capable of detecting/tracking WiFi-enabled devices in a room and associating them to the person carrying them. We propose a framework for sensor fusion from different sensing modalities based on commonality of people activities using sensors. A general structure of the framework is depicted in Figure 1.1.

The goal of the framework is to provide standard for inputs and outputs of different sensing modalities to allow "plug and play" type of system usability.

In the presented study we apply the framework on two main sensor analysis components, a radio signal strength based component and a video component, each processing raw data and providing motion detected in each stream. The radio signal processing provides metrics, "features" that may be used by machine learning models or by applying simple schemes such as comparison to a threshold to decide if a device is moving. The result will be a collection of timestamped instants that the device is moving to be compared against moving objects detected through Computer Vision (CV). The CV tracks moving people across multiple frames of the video sequences and reports a set of tracks and their corresponding motion times. In our fusion processing, tracks are then labelled with the devices' unique identifier to associate a particular device to an owner detected from the vision subsystem. The example fusion of this thesis is linking movements of a WiFi transmitter in a device to movements of a visually tracked individual and deducing that the tracked individual must be carrying the device.

1.1 Prototypical Applications

This section elaborates on certain applications, and outlines real-world scenarios and the challenges that need to be solved with sensor data fusion. All use cases presented in this section are for indoor smart environments with multiple participants.

In this thesis, we focus on challenges and possible improvements to the following applications:

- **Indoor Localization and Positioning.** One of the common problems addressed using multiple sensors is indoor positioning and localization of devices or objects in a scene. Being able to precisely localize individuals in an indoor environment may help reduce navigation time in emergencies, such as dispatching emergency services in an unknown building. This application may not be plausible for people unfamiliar with the facility, since they may not be communicating with an indoor system or possess necessary sensor devices.

- **Assisted Living (AL).** An example application domain of great importance is the observation of people's health conditions in a home environment [31], [54] and improving the quality of assisted living [4]. A number of sensors have been discussed in [13] as part of AL systems for the elderly. AL systems may be used by elder population to help in everyday life tasks by gathering sensor data. However, most of these systems are still error prone and still require more sophisticated and generalizable methods for long-term robustness [13].

  Others research the health conditions of the caregivers, due to the high risk of physical injuries related to care provision [31]. In both cases, it is typically assumed that a person being tracked always wears or carries a mobile device; but this may not be possible due to the type of activity exercised [11]. In a related question, a challenge for image based systems is to distinguish whether a device is left at some location or that it belongs to a particular individual, which may be solved, as we will see, by incorporating another sensing modality.

- **Human Activity Recognition (HAR).** Another field of research where sensors play a significant role in enhancing user experience is optimizing building energy consumption by understanding human actions in an environment [2]. A challenge for this kind of systems is to fuse different modalities that can help track people's activity in a comprehensive manner. A recent survey on Body-Sensor Networks (BSN) [18], notes that data fusion needs increasing sensing dimensionality to achieve better performance. For example, an accelerometer can report data as if a person is at rest, when they are really static or simply the device is left on a table. This type of behaviour could be, possibly, resolved through other sensor data, such as video, to disambiguate actions in a scene.

The proposed systems and solutions for the above applications can be divided into device-based and device-free systems, where the former utilizes sensor measurements produced by a mobile device. The second type utilizes "passive" sensing methods, where one does not require possession of the devices and

instead the individuals are tracked using sensors installed in a room. Several systems rely on smartphone integrated sensors, namely Inertial Measurement Units (IMU), which may provide a higher accuracy in localizing people, yet may reveal private information about the users [15], [22]. This type of sensing is also known as crowd sensing [20]. Such an "active" approach relies on the assumption that users collaborate and share their data with a controller or perform computations locally, on a phone. While we can use measurements from the infrastructure mediated sensing for a particular space (conference room, hallway or private office) as an alternative, it may require installation of additional equipment in the existing building infrastructure [32]. As an example, the system described in [15] relies on an array of ultrasonic sensors. Although the method may solve a number of problems, this set up is not common for most commercial facilities, may be expensive to establish and requires complex user participation, such as wearing additional equipment. By contrast most residential and commercial environments include wireless Access Points (APs) and possibly a camera for surveillance purposes; thus, these devices might be used as a means of building networked sensing systems.

There is also a number of systems which rely on mobile device's embedded sensors, such as accelerometers, magnetometers, etc. that aim at understanding human actions in an environment. HAR systems using this type of sensors require a subject's collaboration and might be invasive in terms of sensor placement [50]. Such systems utilize sensor readings to infer the type of activities a device owner is engaged in. The main rationale of identifying certain events is in the similarity of sensor readings to those collected previously [28]. Another approach is the examination of statistical attributes of certain activities, as an example, accelerometer values will have lower variance and standard deviation from its mean for a particular axis when a person is not moving the sensor [11]. A number of systems have been proposed for online subject activity tracking [11], [30], utilizing accelerometers, gyroscopes, and other smartphone built-in sensors to recognize motion and activities, such as walking, running, or sleeping. An alternative to that is the utilization of object and infrastructure mediated sensing, which employs sensors placed as part of a room infrastruc-

ture, such as on a light switch or on a coffee pot. Objects with this type of sensors may provide a deeper understanding of peoples' activity assuming the use of these devices in an individuals' activities. Activities engaging object movement are referred to in the literature as *micro-motions*, and may be computationally expensive or difficult to detect [41].

All applications listed above can be improved with multi-sensor fusion and identifying sources of each stream at a particular point in time, i.e., understanding when an event impacts multiple sensor readings. In this work, we will rely on infrastructure mediated sensing, namely radio and visual sensing modalities, to understand different types of motion that causes notable impact in sensor readings, such as a moving person, referred to as *macro-motion*. Inferring motion from cameras and smart devices could help associate people to devices, thereby increasing accuracy of activity recognition, health tracking and possibly indoor positioning.

1.2 Existing Sensors and Desired Properties

The variety of affordable sensors allows system developers to employ various sensing modalities depending on their budget, purpose and complexity of the systems. While many sensing devices are providing significant services reporting their measurements, they can be grouped into a few broad categories: visual, environmental and subject sensing.

Visual sensors are similar in terms of their output, providing a usually two dimensional grid of values, where each value can be multi-spectral, consisting possibly of spectrum value and depth. A regular three channel camera (RGB) provides 3 such grids with red, green and blue values for each pixel, and it is the most common type of visual sensors used in smart environments. Another kind is infrared (IR) arrays, varying in resolution, frame rate (FPS) and cost, where a high definition temperature array may cost of up to $1000 per sensor. On the other side, a less expensive device is limited to up to 8 FPS [7]. Depth sensors are more precise at object tracking and one of the most successful commercial examples is Microsoft Kinect, although it is on an expensive side

and programmed to keep track of only 2 active subjects. All of these visual sensors can be used interchangeably or in combination with each other in an environment, although RGB cameras usually serve surveillance purposes and are used for, as an example, tracking user identities via face recognition [7]. A recent survey on ambient sensing suggests that visual sensors achieve the highest performance on average and are good at detecting a more complex events [45].

Environmental sensors are a group of devices that are generally better for privacy preservation, yet still providing a lot of useful insights for many indoor applications. Such devices report, e.g., CO2 concentration level, motion detection, light intensity, temperature, humidity etc. According to a recent survey [45], ambient sensors require the least user collaboration, require less computational power, sustain operation for long periods of time, though some might require maintenance. Nevertheless, they are less accurate due to their limitations in identifying individuals or objects which is an essential point in this thesis.

We consider the following factors in selecting sensing modalities in our methodology:

- **Cost.** It is an important issue for a wide variety of applications which can either limit performance of the system or increase financial cost to the stakeholder. Our goal is to utilize existing infrastructure and rely on commonplace of sensors in each environment.

- **Computation Complexity.** Complexity is an indirect cost that may either increase cost or provide less relevant results due to delay in computationally expensive calculations. In this thesis, the proposed framework aims to achieve computation time of less than the frame sampling interval processing, i.e., implementable in near real-time.

- **User Privacy.** Privacy is a concern that may prevent people from integrating sensors into their smart environments. Monitoring home environments and healthcare facilities may reveal a significant amount of

information, even some health conditions. Therefore, the framework should use a lesser amount of private information.

- **Experimental Set Up.** Approaching critically the set up is an indicator of the solution's feasibility in a real world scenario. Whereas a number of researchers have demonstrated outstanding performance of their methodologies for indoor environments, they may not always be applicable because of:
  - **User involvement**, being a deciding factor in keeping track of people's behaviour, may not be acceptable if extra sensors are required on them as part of a system. Ambient sensors allow no, or a smaller amount of, user involvement with the system, so they are naturally preferred.
  - **Performance metrics** used in several solutions label correctness of decisions made by their systems differently, but for our application it is important to understand motion on a frame by frame timing basis to achieve real-time operation. Moreover, the systems should be also compared in the amount of training data required to build a model.

The above factors will be used when evaluating existing solutions for device and object identification. The sensing modalities chosen for this thesis are visual and radio signals and are a better choice in terms of the majority of the above listed points. The sensor data assumed are produced by sensors that are always connected to a power source, possibly, built into the environment, do not require significant user collaboration, do not intrude privacy (beyond their original purpose) and may provide higher precision by fusing data from both sensing modalities.

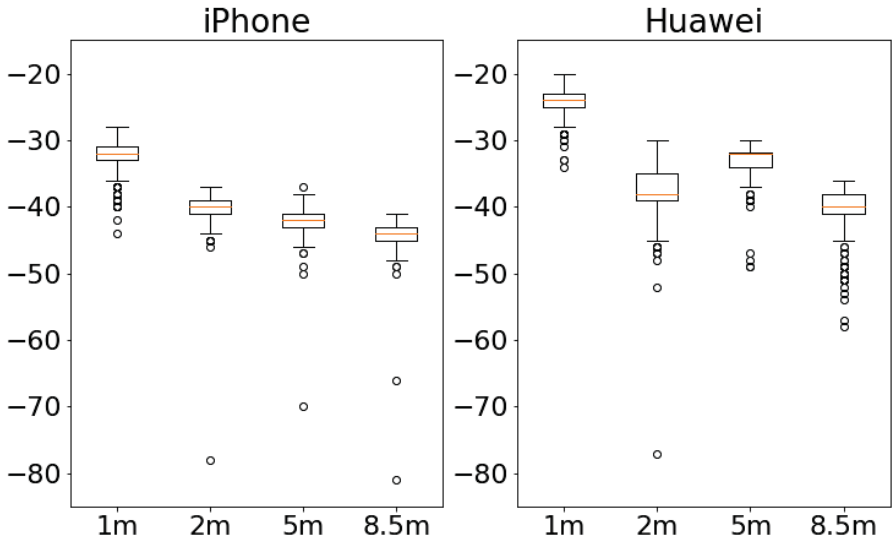

Figure 1.2: RSSI values at various distances for an iPhone Xs and Huawei 6P phone.

1.3 Radio Signal Strength as Distance Estimator

Commodity Wi-Fi APs are essential in many environments and in this thesis serve as another sensing modality. Each frame received by an AP provides information about the signal strength as measured at the receiver, called Received Signal Strength Indicator (RSSI). According to several researchers [21], [42], RSSI values correlate with the distance from the receiver. We usually witness an increase in signal strength when a transmitter is closer to an AP. The rationale behind this is that signal strength can provide approximation of the distance from the wireless AP. However, the relation is not linear and varies drastically due to wireless propagation characteristics and possible signal interference.

There are various techniques for localization and a person's activity recognition based on RSSI, but RSSI values vary with manufacturer and lack precision. As an example, we have conducted several experiments in a conference room to examine the behaviour of RSSI values from an iPhone XS and a Huawei Nexus 6P mobile devices at a height of 1.2 meters from the floor at

various distances from a receiver. The results can be seen in Figure 1.2, where box-whisker plots show the fifth and ninety fifth percentile each calculated for 500 samples of RSSI per distance within line-of-sight (LoS) of receiver, an AC1750 model made by TP-Link. As it can be seen in practice, Figure 1.2, there is a strong relationship between distance and average RSSI value. Nevertheless, there are several cases, as an example when the RSSI is -55 dBm, where one may not be able to tell for sure how far a devices is from an AP. The situation generally deteriorates with obstructions, if the transmitting device is not in LoS.

It is a challenge to determine the variability of RSSI measurements across different manufacturers of mobile devices. As an example, similar tests were performed for a Huawei device in the same room and on the same day, yet the RSSI values at each distance vary from the ones reported in iPhone experiment as seen in Figure 1.2. Since the used receiver(TP-Link AC1750) has multiple antennas, the reported values are from a single, specific antenna. RSSI decreases beyond two meters from an AP, yet measurements of -80, e.g. -50 dBm are observed at each distance further than two meters away from the AP.

The problem can also be attributed to the receiving end, yet in this example we fixed the AP device to a single variety. The used AP is equipped with multiple receiving antennas each reporting different values of signal strength, as expected, which were found to be as much as 10 dBm apart.

To sum up, APs can be utilized to estimate the signal strength of certain wirelessly connected devices and treat this as an indication of distance from the receiver, yet with a significant error. Although the previous work on RSSI based localization has demonstrated low precision, we will be able to detect motion by simply tracking variations in signal measurements as shown in Figure 1.2.

1.4 Visual Domain Analysis

It is known that a visual sensor can help distinguish, with high precision movements in the environment, yet there are still significant challenges to Computer Vision (CV). First, depending on the camera placement, we can either use more sophisticated machine learning models for tracking people's faces or simpler ones which can identify human motion robustly [27]. Second, cameras installed may differ in image definition; thus, reducing tracking performance. These issues can be resolved by tracking only object motion rather than identifying more complex events.

There is a myriad of visual sensors available for visual data collection providing RGB, infrared, depth and thermal readings for analysis. While this variety may create many opportunities, there are several trade offs that must be considered for each. A commonly available component of smart environments is a high definition RGB camera for which there has been significant research on multiple object tracking applications [6]. It is pointed out that thermal sensors may work in varying light conditions without affecting performance, although they cost more than an RGB camera or come at lower resolution or frames-per-second (FPS) rate. Object tracking with RGB camera proved to be efficient while tracking large objects in the scene, yet is still inefficient in terms of detecting small or distant objects, such as a mobile device carried by person [7].

Camera surveillance cannot be effortlessly coupled with other sensor measurements and multiple sensing modalities, such as wireless data transmissions, to increase performance of the aforementioned systems. While this combination of sensors' data may seem unrelated, it could be achieved through performing time-based correlation of "similar" events observed in both domains.

1.5 Contributions

In this thesis the following research questions (RQ) are to be answered:

- **RQ:** Is it feasible to fuse RSSI and Visual domains as explained in our

framework?

We associate individuals to devices (MAC) addresses by identifying similar motion patterns in the visual and RSSI domains.

- **RQ:** Which RSSI-based motion detection method is better considering the system requirements?

  We propose several methods for RSSI-based motion detection, namely Thresholding Coefficient of Variation, Bayesian Inference, Support Vector Machine, and Recurrent Neural Network approaches. We evaluate their relative performance under different conditions and inputs.

- **RQ:** Is the incorporation of multiple receiversAPs beneficial for our framework?

  We propose and evaluate a consensus algorithm over multiple static receivers (in our case, statically placed AP) applying also a moving average filter to the votes cast by different APs. We evaluate performance of single AP based motion detection algorithms compared to the multiple APs case.

- **RQ:** Is it possible to generalize RSSI motion detection models and algorithms across different mobile devices?

  We conduct several experiments with varying conditions in the experimental set up, i.e. different number of individual and devices. We test performance of the methods across different device makes and the environments.

In this chapter we summarized widely used substitutes for sensors in our experiment, their advantages and disadvantages. Then, we continued the discussion by listing key system requirements that we follow while building the framework. We elaborated on the ambient sensors used in our environment, and stated key assumptions and challenges in each domain. Finally we concluded with major contributions of this work. The rest of this thesis is organized as follows. Chapter 2 will provide a taxonomy of previously proposed

solutions, and evaluate each system according to the requirements outlined previously. This thesis utilizes a data set collected specifically for this work; thus, Chapter 3 will explain placement of sensors, the number of participants, the pre-defined scenario, and the structure of each sensor's data. Ground truth annotation for the Experiment I, described in Section **??** was conducted by the HIP summer interns. The methodology, Chapter 4 and 5, will outline the framework for sensor data fusion based on event alignment, then, revealing details on each component. Chapter 6 will discuss performance metrics used to evaluate each approach to motion detection and present the results. We will conclude the thesis in Chapter 7 by highlighting major findings of this work and listing challenges which are yet to be addressed.

Chapter 2

Literature Review

Research on motion detection based on RSSI mostly consists of two main directions: device-based tracking and device-free passive (DfP) analysis. Device-based tracking systems rely on signal processing from a mobile transmitter to a wireless AP. A smartphone is usually connected to a network provided by the AP on a particular radio frequency, where transmitted frames are received, and each frame is associated with RSSI value, as an indicator of signal strength. As it was stated in Section 1.3, there is a correlation to a certain degree of RSSI and distance from an AP; thus, it may be referred to compute either a location of a device relative to an AP or event recognition [48], [58]. Device-free approaches, on the contrary, rely on previous knowledge of a room infrastructure and tend to capture abnormalities or different states for event classification [51].

The latter approach uses fluctuations of RSSI as an estimate of distortion in an environment between transceiver and receiver, i.e. a certain event has happened. Nevertheless, this method requires prior knowledge of the environment and may not perform well under varying circumstances [51], [57], [58]. These circumstances include a different number of occupants in the room, number of transmitting devices and sometimes make and model of both wireless transceiver and receiver, as it was briefly mentioned in Section 1.3.

The majority of research in the area of event detection falls in one of two several main categories: ambient sensing and location-based activity recognition. The following subsections illustrate existing groups of methodologies,

Modality	Device-based	Device-free
RSSI	[47]–[49]	[29], [38], [50]
CSI	None	[14], [19], [44], [50], [52], [56]–[58]

Table 2.1: Taxonomy of related work on RSSI motion detection.

Modality	Model-based	Feature-based
Visual	[5], [10], [16], [30], [39]	[9], [17], [53]

Table 2.2: Taxonomy of related work on CV object detection.

sensor modalities used, challenges and relation to our work. The taxonomy of related work corresponding to RSSI modality and their approach to passive or active device use is displayed in Table 2.1, whereas research in CV on object tracking is grouped into model-based and feature-based and presented in Table 2.2.

2.1 Received Signal Strength Indicator

This section outlines a number of related works based on RSSI with applications to Human Activity Recognition (HAR), motion detection, indoor localization as a motion estimator.

The problem of Motion Detection can be considered as a particular type of HAR, since mobile devices may be sensing user behaviour. HAR has been studied for over a decade using various approaches. Indeed, a number of HAR research articles utilize a variety of sensor readings to understand particular states, such as moving or standing person. A survey [50] on HAR outlines sensor data analysis worn by a subject as well as environmental sensing, where any interaction with a smart device may indicate a certain action by the user. Ambient sensing relies solely on infrastructure installed in rooms rather than more sensor-rich scenarios, such as smart object interaction.

This section is going to describe a number of existing methods for RSSI time series data analysis involving simpler machine learning and deep learning algorithms. A number of solutions that utilize radio frequency domain have been proposed in HAR research, where more recent works suggest that Channel State Information produces better performing models rather than RSSI based

methods [14], [38]. In [38] for example, a comparative study on passive, device-free, RSSI and CSI is performed with the purpose of categorizing sensor data into five core human activities: such as walking, standing, absent (empty room), lying and crawling in a room. They also consider RSSI relevant to environment distortion due to human interference and propose a number of significant statistical features for training a machine learning model to solve a classification problem. One of the main contributions relevant to this thesis is their model and its performance. Nevertheless, they claim that these models cannot be transferred due to changes in a room structure, occupancy etc. Another major concern for us is that in [38] only a single person is tracked in the room, which is not the case in many environments.

One of the oldest methods directly related to our work is [48], which falls into the category of device-based motion recognition. Its authors also relied on RSSI values by multiple APs from mobile devices in indoor environments to estimate static and moving behaviours. A number of devices were positioned randomly throughout a building's floor in different rooms, and laptops were used carried by a person at walking speed in a hallway. The goal was to understand if a transmitter was static or moving anytime within a 30-second window from the point of view of multiple receivers. This is different from our goal, which is trying to understand if a device started moving at a particular point in time.

Another assumption made in the paper, that an object has a constant mobility orbit, may have helped them because the change in RSSI might have been more conclusive for determining the disposition of the device, i.e. moving or static. They used statistical metrics, sample variance, and a threshold, to decide if a device was moving in an observation window from the perspective of a single fixed receiver's stream. Another contribution of their work is varying sample set sizes which demonstrated that they can achieve 90% accuracy regardless of sample size. Their work is related to our work in terms of sensing modality and the general goal of the system, although their performance is not informative.

Recent work on RSSI based event recognition also includes a number of

systems identifying an object's activity based on location attributes of an indoor environment. Such an approach has been discussed in [29] where a localization algorithm was proposed and used to infer an activity from an object's displacement. A total of nine actions were proposed that a user can perform in an environment and related those to a path that an object takes to perform the action. For example, the "prepare food" action will mean that the subject has to go from a desk to a microwave oven. To understand the initial location and final destinations of users, fingerprinting was utilized, ranked RSSI vectors and application of Principal Component Analysis (PCA) to select the best features for their Deep Learning model was used. A fingerprinting technique, that has proven to be more accurate in recent years, is a method of prior knowledge initialization where a vector of RSSI values is collected over time in a particular room, thus, profiling the environment in advance [29]. It is an adequate approach that allows achieving 1-2 meters accuracy, yet it relies on the room environment being constant and may not be able to provide consistently high performance [29]. The events that were targeted in this approach can also be grouped into the two main classes, moving and non-moving, which is the goal of our classification problem. Although it was mentioned in [29] that they apply PCA for choosing better features for model training, we intend to investigate the performance of such models based on raw RSSI values obtained from the experiment. The best performance achieved for activity recognition based on the action path is almost 80% accuracy, which is what we will consider as a performance benchmark for our purposes.

2.2 Channel State Information

An alternative direction is to use Channel State Information (CSI) [19], [44], [52], [57]. Activity tracking with CSI has proven to be more accurate than RSSI, in fact, it can outperform the latter in most cases using just a single device [19]. Nevertheless, devices supporting this type of measurement do still require WiFi chipset [44].

Recent research involving CSI for human activity recognition reported

in [57] proposed a deep neural network approach for classifying basic tasks, such as lying, standing, sitting, i.e. static events, against walking and running, i.e motion. The data collection process involved establishing a transmitting and a receiving CSI enabled Access Point on a 5GHz network without interference from other networks, capturing CSI data on over 100 sub-carriers at 80 times per second. Several data collection rounds were executed in two different environments, on consecutive days, to examine the extent to generalizing the system. According to [57] a CSI "frame" in their work is a 114 sub-carrier amplitude frames captured by the receiving node within the most recent 0.5 seconds, amounting to a 40x114 pixel 'image'. A sequence of Artificial Neural Network (ANN) models is trained for representation learning, feature extraction and sequence learning. The ANN is a type of machine learning algorithm that imitates the structure of a brain, where neurons contribute to an output of the algorithm on different layers. Each CSI frame is passed to a trained Auto-Encoder for reducing dimensionality, then output is passed to a CNN to extract the most informative features which are passed to an LSTM network for learning temporal dependencies to classify human action in that period of time. Whereas the authors reported an average cross-validation accuracy of 97.4% outperforming previously proposed systems in true and false positive rates too, they failed to demonstrate the generalization of their model to other environments.

In this thesis, we are utilizing a similar architecture of ANN for time series classification, but also examine the transferability of such complex models.

A different approach to CSI activity classification was proposed in [51] and was one of the first works to pursue trying to solve localization, motion detection and human micro-activity recognition together. The data collection rate is similar to [57], 80 samples per second for over a hundred sub-carriers, with an environment 'scanning' approach. First, each time series is filtered through low-pass and network quality filtering, which then is categorized into CSI corresponding to a moving or static object. A CSI profile is considered to have a moving object if a cumulative variance of each sub-carrier amplitude is above the empirically determined threshold, the results obtained have 98%

accuracy detecting moving and static objects. It is possible to further identify user behaviour by matching a static CSI frame to previously collected profiles for each activity. The procedure involves calculating a signal distance, such as Earth Movers Distance (EMD) and compare to a pre-determined threshold. This type of classification has been tested in a real environment, of a private apartment, and demonstrated high accuracy, though, it has a number of limitations. All experiments involved only one subject in a smart environment which, though useful for private office spaces, is not representative of many environments.

Gu et al. have examined CSI-based motion detection of people in a room based on abnormality detection [19]. The main idea is to analyze the maximum amplitude of CSI during the absence of movement in the room, and apply it as a threshold for signal distortion caused by people's motion. The algorithm starts by capturing CSI data on 30 different sub-carriers which is then filtered using a variation of a low-pass filtering technique described in [1].

Then, several subcarriers that are a better representation of the event are chosen based on signal distance metric, since it is assumed that those will have less noise in their readings. These streams are used for "silence" detection by measuring the mean and standard deviation of its 'silent' states. Their methodology was tested in a real office space with several students performing regular activity throughout a 24 hour period with video recording serving as a ground truth measure. The goal of the system was to identify macro-motions of people with a duration of up to 15 seconds per motion; however, the authors were not concerned if a duration of the movement is correctly identified by the system. The reported accuracy of motion detection during the experiment was 92% which outperformed other techniques also utilizing CSI data. The work is solving a motion detection problem that is similar to ours, but by detecting if a person has moved in a particular period of time rather than detecting when people moved. Moreover, the author notes that the approach is effective mainly for device-free cases since it 'scans' the environment and could suffer performance degradation if one of the transmitting/receiving nodes were moving.

Another alternative to automate decision making, on both RSSI and CSI data, is to utilize Artificial Intelligence models, such as conventional machine learning and deep learning. Multiple architectures of ANNs were compared for time series classification tasks in [50]. The authors have provided a performance comparison of wide, fully connected (FCN) and deep neural networks for various time series classification tasks. They also illustrate key challenges related to training each type of architecture and important insights into developing this kind of model.

Research in event detection for time series data has evolved from machine learning to deep learning methods with the emergence of ANNs. Conventional machine learning techniques rely on a number of features pre-processed from the data as input to a model, known as a feature vector, then mapped to a labelled output for classification of data. An alternative approach to feature defined machine learning models is to use neural network models, where neurons are trained to learn features necessary for accurate classification [50].

Summarizing CSI-based motion detection systems, the majority of systems does a more thorough work in categorizing human activity in comparison to RSSI based methods. Nevertheless, it remains the case that the environment plays a very significant role in motion detection and mobile device-based tracking is essentially an open problem for CSI-based systems. Moreover, it is essential for our work to identify which devices were in motion, and in particular, during which intervals they could be classified as moving. Experiments conducted for this research contain a varying number of people and devices creating a more representative real-world setting.

2.3 Indoor Localization as Activity Estimation

A significant component of the presented work is whether an object is moving both in the visual and in the radio signal domain, i.e. by implication deciding if a device or a person is moving or static. Several researchers in the field of sensor networks have proposed solutions to indoor localization

problem using RSSI [29], [47], [58]. It is reasonable to think that change in location of an object means that the object has moved, so any localization algorithm automatically may provide a solution to our sub-problem. Nevertheless, many RSSI-based solutions cannot achieve high localization accuracy or require extensive learning of an environment beforehand [47], [56]. Otherwise, localization algorithms may not be as efficient which may result in delayed motion classification due to late displacement detection. Closely related work, in [58], aims at identifying a user by associating a wireless device to the user. One of their main assumptions is that a mobile device, a smartphone, is carried by a user, thus, identifying them as a person. The system distinguishes static, e.g. office computer, and personal mobile devices, to associate user identity obtained through 'known' work PC to a user's smartphone. First, it was essential for them to discard temporarily appearing devices, that do not belong to the building floor; this is implemented through localizing devices overnight. Next, the system distinguishes between mobile and static devices to separate personal and work equipment. Then, since most of the static devices are surrounded by other computers, a better location estimate is obtained by applying a filtering technique. Finally, the standard deviation of location estimate and a distance threshold was used to associate a user to an office PC. The authors proposed the system as an application of an indoor localization service proposed in [56] which is also based on device-free CSI scanning of the environment. Impressive user association accuracy of 95.8% across 24 static devices was reported, yet did not report a precise percentage of correct mobile phone associations. The evaluation experiment was set on a building's floor with a combination of personal offices (one person per room) and shared cubicles of up to 4 people. The indoor localization error of the system proposed in [56] was lower than 2 meter which may allow estimating device motion between rooms more accurately than intra-room motion.

Most indoor localization systems utilizing either RSSI or CSI sensing provide accuracy of up to 2 meters, which is useful only for macro inter-room movement as was shown in [29], [58]. In the state-of-the-art device-free, CSI based HAR systems better accuracy at localizing objects is claimed, yet this

does not help to identify them, i.e. the AP measurements may not be able to distinguish which device is causing a 'distortion' at which point. Moreover, these systems may still underperform in a smaller indoor space room due to shorter motion intervals and distortions caused by the other moving devices; thus, aiming at solving mobility detection problems directly with RSSI values from each device may improve overall motion detection accuracy.

2.4 Motion Detection using Computer Vision (CV)

Motion estimation using CV techniques can also be grouped into several categories. There is a group of methods that compute the difference in pixel values as an indication of a change in the environment, identifying outlying values for a particular pixel, known as background subtraction approach [7]. Other CV algorithms can be grouped as detection based methodologies, where an object is being detected and then motion is calculated based on its track [8], [10]. A technique for multiple object tracking using high-resolution cameras, that has been around for more than a couple of decades is foreground estimation or background subtraction algorithms [53]. The idea behind such an algorithm is to create a binary mask of an input image where a pixel is white, i.e foreground, in case there is a strong deviation from a previous frame value. The output mask is then analyzed to detect the so-called "blobs" of white pixels as objects detected when the area is substantial. Each blob detected throughout a sequence of frames is associated with a track, and a series of detections is a single tracked object. A similar method that can be used to generate a foreground mask is Optical Flow [7], which estimates the motion of each pixel from frame to frame.

A number of object tracking methodologies rely on continuous object detection and handling missing detection in frames by using various filtering techniques, such as Kalman Filter [9]. Possible bottlenecks for object tracking are object detection related issues, such as object occlusion, track mislabelling, etc. [7]. In this thesis, we adopt recent state-of-the-art CV techniques for

identifying people in a room and tracking their motion throughout the scene.

An example of Foreground detection based system was proposed in [17] for monitoring museum visitors and improving the user experience. The basic idea of the paper was to model a background image and subtract it from a current frame and analyze for significant changes to label it as a foreground. A histogram of possible pixel values was collected over a period of 120 frames, for each pixel in an image sequence. These histograms were used for the Gaussian Mixture model fitting for estimating the probability density function (PDF) of each pixel value. Bayesian inference utilized the PDFs to calculate the probability of a pixel belonging to a foreground mask, where one would indicate it as being foreground, and zero as a background pixel. Such a foreground image provides several segments, the "blobs", indicating a moving object. The scope of [17] was to analyze statistically the area and connected components of each blob to identify bounding boxes. The new objects are then initialized with a Kalman Filter, a filtering technique that allows estimating the physical state of a system. The filtering technique follows a constant velocity model, where a moving object, if not detected in the frame, is assumed to proceed with the same speed as in the previous frames. This approach allows us to predict the location of an object and can improve tracking accuracy. The proposed approach demonstrated robustness for a busy period of several days in an indoor environment with a single camera point of view.

A different group of CV algorithms utilizes features, such as color histogram (HOG), edges, corner pixels, morphology etc., extracted from a particular region of interest [7], [24]. These features are detected throughout a video sequence and matched together to create a track of an object. The last group of CV methods also relies on features, but it does not explicitly track those in each frame [8]. There are many deep neural networks and classification models which aim at learning patterns from training on time series data [10]. Most object detection models are based on raw image pixel input to a convolution neural network architecture specific to particular applications, such as outdoor pedestrian detection, object classification, human and face recognition, etc.

A widely used approach for object classification in an image, was proposed

in [16] which allows robust detection and tracking of multiple people in an environment. A form of Fully-Convolutional Networks (FCN), a form of Neural Networks, which is trained to produce a region of interest (ROI) type of output was used [16], trained on top of the VGG16 [39], a deep Neural Network architecture, with 16 layers. The method was able to successfully detect and classify objects, such as people, buses, roads, etc. Nevertheless, the computation rate was five frames per second possibly due to a heavy amount of computations for the particular NVidia GPU to handle.

2.5 Sensor Data Fusion

A key element of our work is properly associating moving objects detected by the CV algorithm to a MAC address of a device that was in motion during the same period of time. There are several approaches to coupling different sensor streams based on the "similarity" of such events [3], [37]. The work in [3] describes coupling of data streams from different sensors having a time drift due to limited computation power. It relies on several known signal processing techniques, such as Dynamic Time Warping (DTW) and Earth Movers Distance (EMD) metrics to estimate information difference in sensor readings. The idea is that DTW determines the closest patterns in a time series and treats them as similar events represented as the nodes in a graph where the distance is a weight of edges. Therefore, solving the shortest path problem in that graph will yield a solution, synchronizing several streams simultaneously. In DTW, it is necessary to choose a starting and a final node the time series that has the most accurate time, since it will serve as a reference to other streams. The work in [3] exposes key challenges and assumptions made in a time series and performs event coupling across similar sensors' data.

Continuing on event coupling, there has been recent research on coupling dissimilar streams, such as camera and accelerometer data worn by a user [37]. The work has continued on the sensor coupling approach, yet they address event classification in a different fashion, such as actual human activities rather than quantifying it with respect to entropy. In [37], activities are first clas-

sified with a Support Vector Machine (SVM) algorithm, then the distance is calculated on those events and then uses a graph shortest path solution similar to [3]. The approach will be more relevant to our work due to the similarity of sensing modalities and scope of research. The prototype of the system built according to the framework proposed in this thesis will be compared against other RSSI and device-based activity detection methods, such as [48] and [49].

Chapter 3

Data Set Collection

We have conducted two preparatory and two realistic, for a total of four, data collection experiments. The preparatory data is collected in a specific contrived way that is more useful for machine learning model training and probability distribution calculation. The realistic data experiments are scripted, but representative of real indoor environments. Each experiment took place at a conference room during off-peak hours, where we assume external interference is minimal. The room layout is shown in Figure 3.1, and an approximate floor plan is shown in Figure 3.2. The dimensions of the room are 8 by 10 meters. There is only one individual in both of the preparatory data collection experiments but various mobile device models are used. In the realistic experiments, the first involved two individuals and three mobile devices (smartphones), and the second involved three individuals and five mobile devices. There are eight terminal (green) and six intermediate (yellow) locations in the room for the individuals to walk between during the realistic experiments, as shown in Figure 3.2. The terminal locations represent the start and end points of a movement trajectory, while the intermediate locations are used as markers for the individuals to follow consistent paths. These paths are designed, specifically, for challenging the visual domain with the object occlusion, i.e. losing an individual in the video.

A total of four APs acting purely as packet sniffers were placed at fixed locations in every corner of the room and four cameras are installed at the same spots. The APs are TP-Link AC 1750 Archer C7, a commercial off-the-shelf

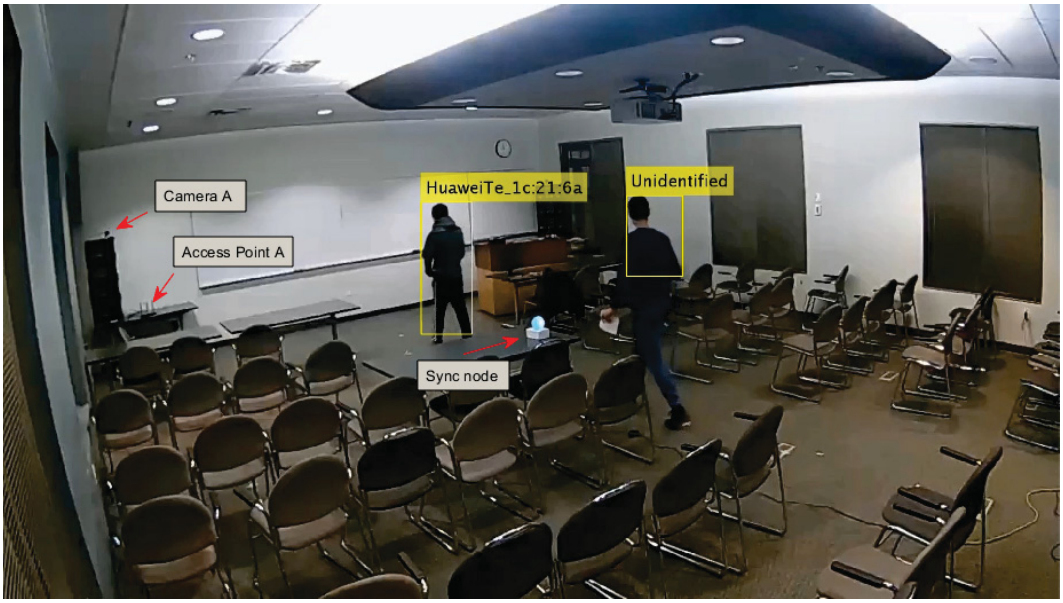

Figure 3.1: Conference Room Layout. (Notice lit up synchronisation beacon.)

device, put to "monitor" mode which allows capturing wireless frames on WiFi channels. The operating system used on the devices was OpenWRT version 18.03 with tcpdump software version 4.9.2, the latest stable release at that time. The tcpdump packet analyzer is a command that can parse the contents of the received wireless packets and report it in a human-readable format. A packet can contain timestamps, source, destination of wireless transmissions, length of packets in bytes, type of wireless frame, data load, RSSI on one or multiple antennas, and several other useful attributes. The routers can operate in 2.4 GHz and 5 GHz frequencies, yet in this experiment, they were set to "listen" transmissions on the lower frequency. Although the majority of recent smartphones support 5GHz WiFi, it is not the case among wearable devices due to the need to keep device costs low. The other component of our setup are high definition (720p) indoor IP cameras; type C1 by Foscam with a 23 fps rate, supporting both wireless and wired connections. For the reported experiments all sniffers and cameras were connected through Ethernet leaving the wireless interfaces unused to reduce potential interference.

The last component, a synchronisation beacon was custom-built and used to ensure that time is synchronised across multiple sensing devices and different

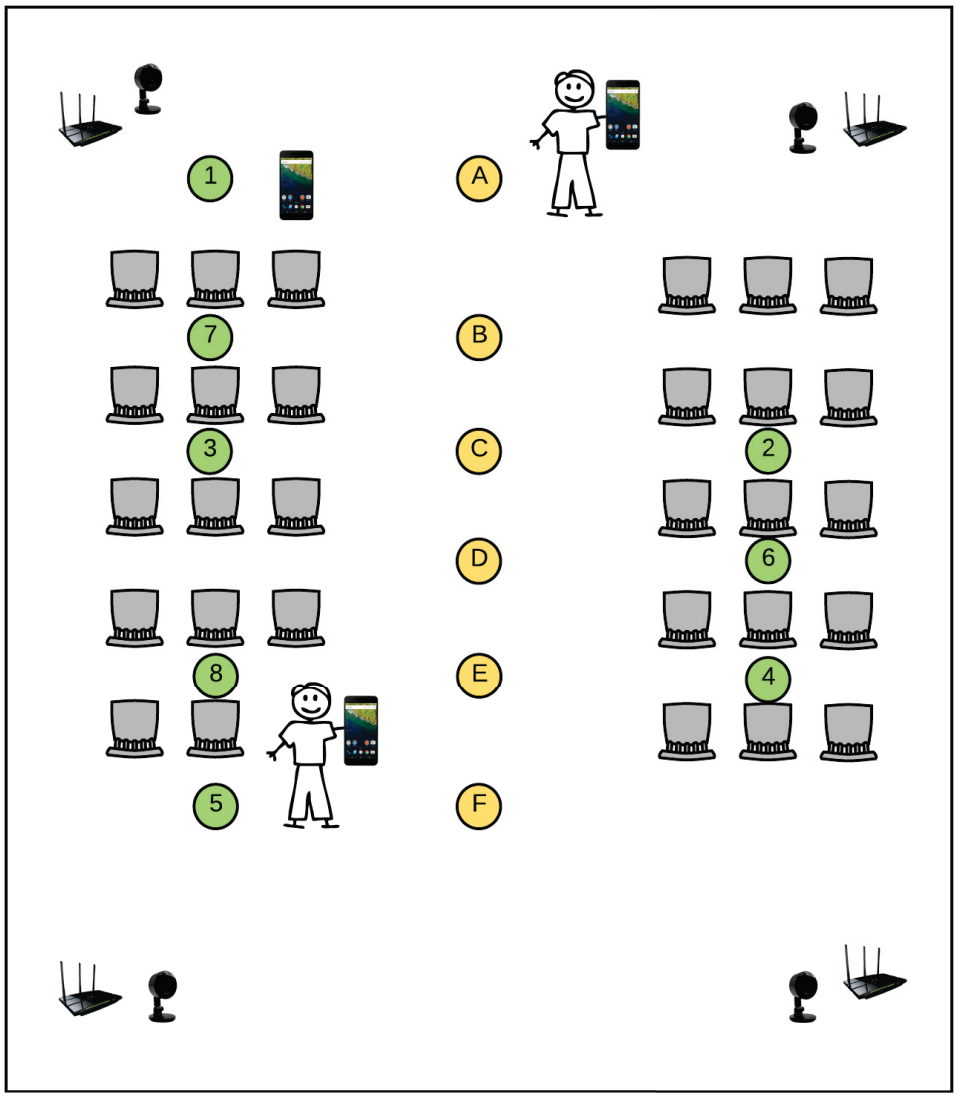

Figure 3.2: Floor plan of the conference room.

data streams. The beacon is put right in the middle of the room and operates as follows: it simultaneously emits a signal observable in both domains, visual and RF domains. The beacon frame transmitted follow an increasing sequence number expressed as SSID. The Beacons are part of the packet capture produced by the sniffers (APs). The synchronisation beacons strobes an LED light inside a spherical diffuser, visible in the visual domain. An example of a visible light emission can be seen in Figure 3.1 in the center of the image. The

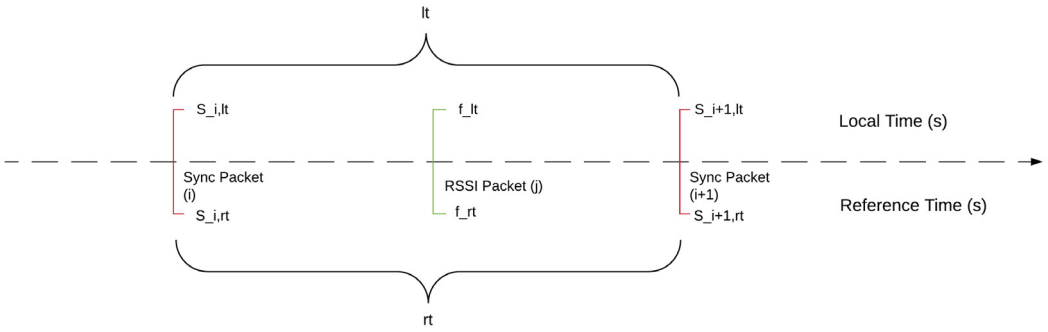

Figure 3.3: Term description for Time Synchronisation. Sync Packet is a SSID beacon from synchronisation node, and RSSI Packet is a transmission from a device.

timestamp, at which a frame is send with lit up synchronisation beacon, was recorded at the cameras, serves as a time reference and allows to correspond the frames to the packet trace captured Beacons.

3.1 RSSI data

Each mobile device is associated with an access point on channel 1, and set to transmit at least five packets per second, relying on a simple ping request to a routerAP, mimicking how a mobile device would connect to the WiFi infrastructure of a building. This is not always the case in a real environment since a device may rest in an idle, energy conserving mode which reduces transmissions. However the majority of connected devices we used produced comparable or multiple of the transmission rate we utilize. The reason for this kind of behaviour is background processes present in many mobile application which send data to the cloud. In our experiment, the devices did not have Internet connectivity and hence the data transmissions were reduced to the one produced by the repeated pings.

A sniffer captures the frames transmitted, each frame is labelled with the router's local estimate of time, and one or multiple RSSI levels depending on the make of the AP. The chosen APs are marking each frame received with four RSSI values, for three different antennas and one for the combined signal

strength. Each RSSI measurement is of 1 dBm granularity and varies across antennas on the same router. Before utilizing the RSSI data, it was necessary to synchronise each router stream using the synchronisation node presented earlier. A beacon frame from the synchronisation node is received every five seconds with an SSID including a sequence number. These beacons receptions serve as reference points for each of the four RSSI streams obtained from each sniffer.

Time adjustment for a captured frame on one stream is calculated according to the following formula:

$$f_{rt} = \frac{rt}{lt}(f_{lt} - s_{i,lt}) + s_{i,rt}, \tag{3.1}$$

where $s_{i,rt}$ is the reference time of the previously received synchronisation beacon received before current frame, $s_{i+1,rt}$ is the next beacon reference time, $s_{i,lt}$ is a sniffer's local timestamp of the beacon frame, and f_{lt} *and* f_{rt} are the received frames local and reference times. That is the synchronisation is performed separately for each set of frames between each pair of successive synchronisation beacons.

We assume that time synchronization errors accumulate evenly between the received synchronisation beacon frames; thus, the time elapsed since last received synchronisation frame in local estimate is proportional to the reference time passed since the last beacon. Thus, the RSSI and video streams cannot get significantly out-of-sync to threaten the validity of our results. The Figure 3.3 provides graphical description for each of the terms introduced in the Equation 3.1.

3.2 RGB frames

The cameras were positioned at the corners of the room at a height of 2.2 meters. An example of a frame and field of view covering the room can be seen in Figure 3.1 which was captured by one of the cameras. Each camera captured 3 channel, Red-Green-Blue (RGB), 1280 by 720 pixel frames at a constant rate of 23 fps, 110 degrees field of view. The cameras did not experience any loss in image frames; thus, every stream with a proper start and

end frame alignment are assumed to be synchronised. The four video streams initially were manually aligned according to the first and the last synchronisation beacon lights seen in the image. We confirmed that the beacon light is captured by the cameras at the same five second interval, corresponding to 115 frames as shown in Figure 3.4, by calculating amount of white foreground pixels at the location on the image where the synchronisation beacon light is located in each frame. The foreground image is obtained with a background subtraction algorithm [46], which examines possible values of each pixel and marks deviations as white or gray pixels. The algorithm was provided with a region of interest (ROI), a particular portion of an image frame, with the synchronisation beacon light, specified beforehand.

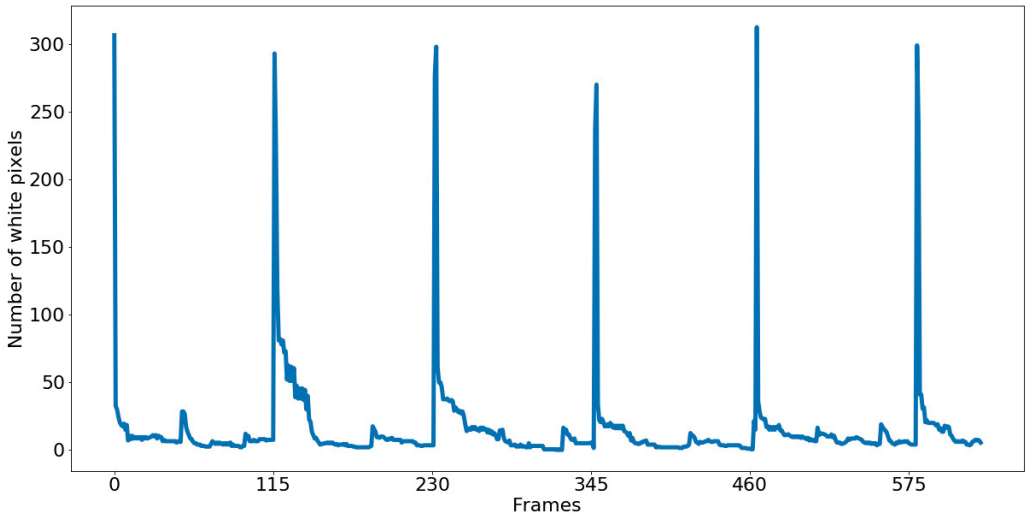

Figure 3.4: Example identification of when the synchronisation beacon lights up (at peaks).

Figure 3.4 is an example of the analysis with a number of equidistant peaks, which are caused by the visible light produced by the synchronisation beacon. However, Figure 3.4 demonstrates only a fraction of an experiment when no individual crossed through the ROI. In that case, the foreground contains a significantly larger number of white pixels and the fluctuations that can be associated with the beacon light are secondary. The first frame of video would correspond to time zero of the experiment and the second is captured at time

0.044 seconds and so on. Thus, we have, for each video stream, a total of 1035 seconds for the first experiment and 1875 seconds for the second.

3.3 RSSI Motion-based Data Collection

The four experiments are labelled and referenced in this thesis as 1, 2, 3, 4, whereas realistic Experiments 3 and 4 are labelled with Roman numerals, I and II, to represent realistic experiments.

The first preparatory experiment, Experiment 1, aims at producing a uniform data set for various smartphone types. The environment used is the same as in the round of experiments depicted in Figure 3.1.

Having five different mobile devices we conduct an experiment with each of the devices separately. For every device, an individual is moving between fixed locations and stopping for the same amount of time, so we can collect results at equal length intervals of moving and stationary dispositions. The individual is always facing the receiver/AP which is collecting RSSI values of the smartphone transmissions.

Each of those baseline data sets captures approximately 3 minutes of a device motion and 3 minutes when there is no motion. The data streams collected for every device are manually labelled on a frame by frame basis, indicating 1 if a device was in motion within the last interval, and 0 otherwise. The intervals are each 0.044 seconds long, reciprocal of frame rate. This data will be used later by the Deep Learning model training. Each device's mobility patterns for the first two experiments were manually labelled from the camera recordings. The data is presented in a form of interval, specifying time when a device was in motion, one, or static, zero.

The collected packet data included synchronisation beacon frames as in the previous experiments, and all packets were synchronised according to Equation 3.1.

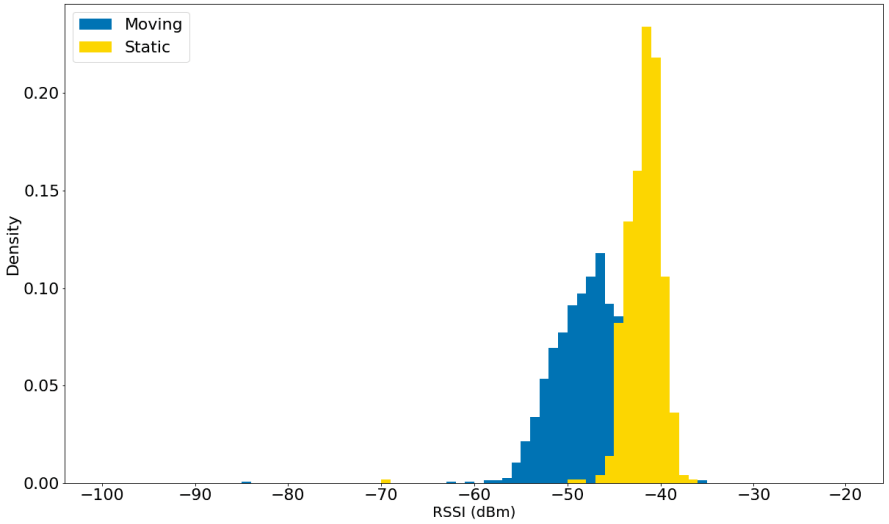

Figure 3.5: Static vs. Moving RSSI histograms for iPhone at 5m distance.

3.4 RSSI Distance-based Data Collection

The second experiment, Experiment 2, took place in the same room, yet for each device individually. Two devices, namely iPhone Xs and Huawei 6P, were chosen to collect data of RSSI fluctuations at various distances and states, such as moving and static. Each device was placed 1, 2, 5 and 8.5 meters away from a sniffer within the line of sight (LOS). Every distance measurement contained 500 RSSI samples gathered on the same day for static and 1500 for a device in motion around that point. The motion is performed by an individual holding the device and exercising constant left to right steps around the specified location with range of 0.5 meters.

These experiments are executed with the purpose of obtaining a probability density function for RSSI values given the state of a device: static or moving. These probability estimates are used as likelihood estimation in a Bayesian Inference equation that is proposed in later sections.

Figure 3.5 demonstrates a normalized histogram of RSSI values both when moving and static at five-meter distance from an AP. As it can be seen, the two cases have very distinct distributions and can be argued to approximately

follow a Gaussian distribution around their means. A similar pattern can be seen for other distances and for other devices. Combining RSSI measurements for devices at various distances, we can see the RSSI distribution is more likely to follow a bimodal distribution, rather than normal, as shown later in Figure 4.4. We used Gaussian Mixture models to fit the probability function into the histogram data. The observed result in Figure 4.4 is likely to be due to a better signal at closer distances, one and two meters distances from the AP.

3.5 Realistic Data Collection Scenarios

There are two real world experiments that were conducted for the system performance evaluation. The two experiments described in this section are conducted in the same room, but with different combinations of mobile devices and individuals in the environment. They were designed to closely represent a real world indoor environment. The mobile devices used in both experiments are Apple iPhone Xs, Apple iPhone 6s, Xiaomi Redmi Note 5 and two Huawei 6P smartphones. Each device is connected to a 2.4 GHz AP on the same channel as the sniffing APs and the synchronisation beacon. The individuals were asked to move in the conference room between terminal positions (green) while passing through intermediate (yellow) positions, as depicted in Figure 3.2. The behaviour of the individuals was natural without any restriction on the length and speed of motion, or how they hold and carry the mobile device.

3.5.1 Experiment I

The first realistic experiment was conducted with a few limitations and a small number of people and mobile devices. The total duration of the experiment was 17.25 minutes (1035 seconds) with three stages each spanning five to six minutes. During the first stage, two individuals start at locations one and eight respectively, remaining static for some amount of time, then moving to the next location. The next location was determined by ascending or descending order, hence it would be location marked as two if a participant started from one, and location marked as seven if started from eight. Each individual executed

a total of three cycles moving between these locations, as an example, the first individual's path was one to eight to one to eight.

Three mobile devices were used, namely Huawei 6P (we label it as Device 1 in each RSSI stream), Xiaomi (as Device 2) and another Huawei 6P (as Device 3). Each of these produces a number of wireless packets, captured by the sniffers, which we label as Traces corresponding to the devices, i.e. Trace 1 is produced by the Device 1. The traces are then used to create RSSI time-series, a sequence of measurements with an equidistant time interval between the readings. As an example, a Time-Series 1 is formed with a time interval, δ, of 0.044 seconds for the duration of the experiment, where $RSSI_t$ is the most recent RSSI measurement from the Trace 1 recorded between t and $t - \delta$. In the case when no RSSI value was within the interval, the $RSSI_t$ is obtained by duplicating $RSSI_{t-\delta}$. The frequency of missing RSSI measurements was insignificant and constitutes approximately 30 samples per five minutes. The two individuals would pick up and put devices as listed in Table 3.1. For example, Xiaomi was held by individual 2 during the first five minutes of the experiment, then it was placed at location one for the next stage. During the last stage Trace 2 was generated by motion of individual 1 who also was holding the device corresponding to Trace 1, i.e. Huawei 6P. The device which produced Trace 3 was placed near location six and remained there for the entire experiment. Descriptive statistics on received RSSI packets are reported in Table 3.2.

Experiment stage	Device 1	Device 2	Device 3
0 - 5 min	Ind. 1	Ind. 2	Static
5 - 10 min	Ind. 1	Static	Static
10 - 15 min	Ind. 1	Ind. 1	Static

Table 3.1: Generation of the traces and corresponding individual.

Data Attribute	Trace 1	Trace 2	Trace 3
Number of Packets (thousands)	7.9	7.7	8
Movement Fraction (percent)	30	20	0

Table 3.2: General description of RSSI data for experiment 1.

3.5.2 Experiment II

In the second experiment, we extended the duration of the experiment from fifteen to thirty minutes. This time the individuals were moving randomly between the locations. The experiment is improved by using a wider variety of smartphones and number of individuals. The duration was increased to test the performance of the proposed methods in a more complex environment. This experiment is assumed to be a more accurate representation of the real world example. Five mobile devices were used, namely Huawei 6P (Device 1), a Xiaomi (Device 2), another Huawei 6P (Device 3), an iPhone XS (Device 4), and a Google Phone Pixel (Device 5). The three individuals would pick up and put devices as it is listed in Table 3.3, where Ind. 1 corresponds to Individual 1, noting that the individuals are not necessarily the same as those completed Experiment I.

Experiment stage	Device 1	Device 2	Device 3	Device 4	Device 5
0 - 5 min	Static	Ind. 2	Ind. 3	Ind. 4	Static
5 - 10 min	Static	Ind. 2	Ind. 3	Ind. 4	Static
10 - 15 min	Ind. 4	Ind. 2	Ind. 3	Ind. 4	Static
15 - 20 min	Ind. 4	Ind. 2	Ind. 3	Ind. 4	Static
20 - 25 min	Ind. 2	Static	Ind. 3	Static	Static
25 - 30 min	Ind. 2	Ind. 2	Ind. 3	Static	Ind. 4

Table 3.3: Generation of the traces and corresponding individual, Experiment II.

3.6 Ground Truth Annotation

All the experiments described previously were recorded with cameras for further ground truth event labeling. The experiments were manually labelled using video-annotation tools, ELAN and ANVIL, proposed in [25], [40] which support frame-based multi-layer annotations. The video annotation software allows users to select intervals and set a label for them. The motion patterns of every device were labelled manually with one, if moving, and zero, if static. Then, for every moving interval, another annotation layer with device-to-individual association was added, where the labels indicate the individual

carrying the device.

Chapter 4

Methodology

4.1 Framework Overview

This chapter proposes a framework for a particular type of sensor fusion based on the idea of motion pattern detection utilizing different sensing modalities. The goal is to provide a data processing procedure with several reusable components, given that they follow certain input and output specifications. Figure 4.1 demonstrates the general structure of the framework, where the two sensor data analysis flows produce collections of moving individuals and/or devices. This section describes various components and parameters of the proposed framework.

We apply a thresholding method based on Coefficient of Variation (CoV), Bayesian Inference, and Supervised Machine Learning algorithms to examine RSSI data. The visual domain is analyzed using state-of-the-art object detection and tracking algorithms. The two input streams of the framework are video sequences and packet traces captured by the sniffers, and the output is in the form of a similar video with individuals labelled with corresponding devices' unique identifiers. The following sections describe the methodologies developed for analysis of these streams separately and completed with a simple association technique of moving visual objects to devices in motion.

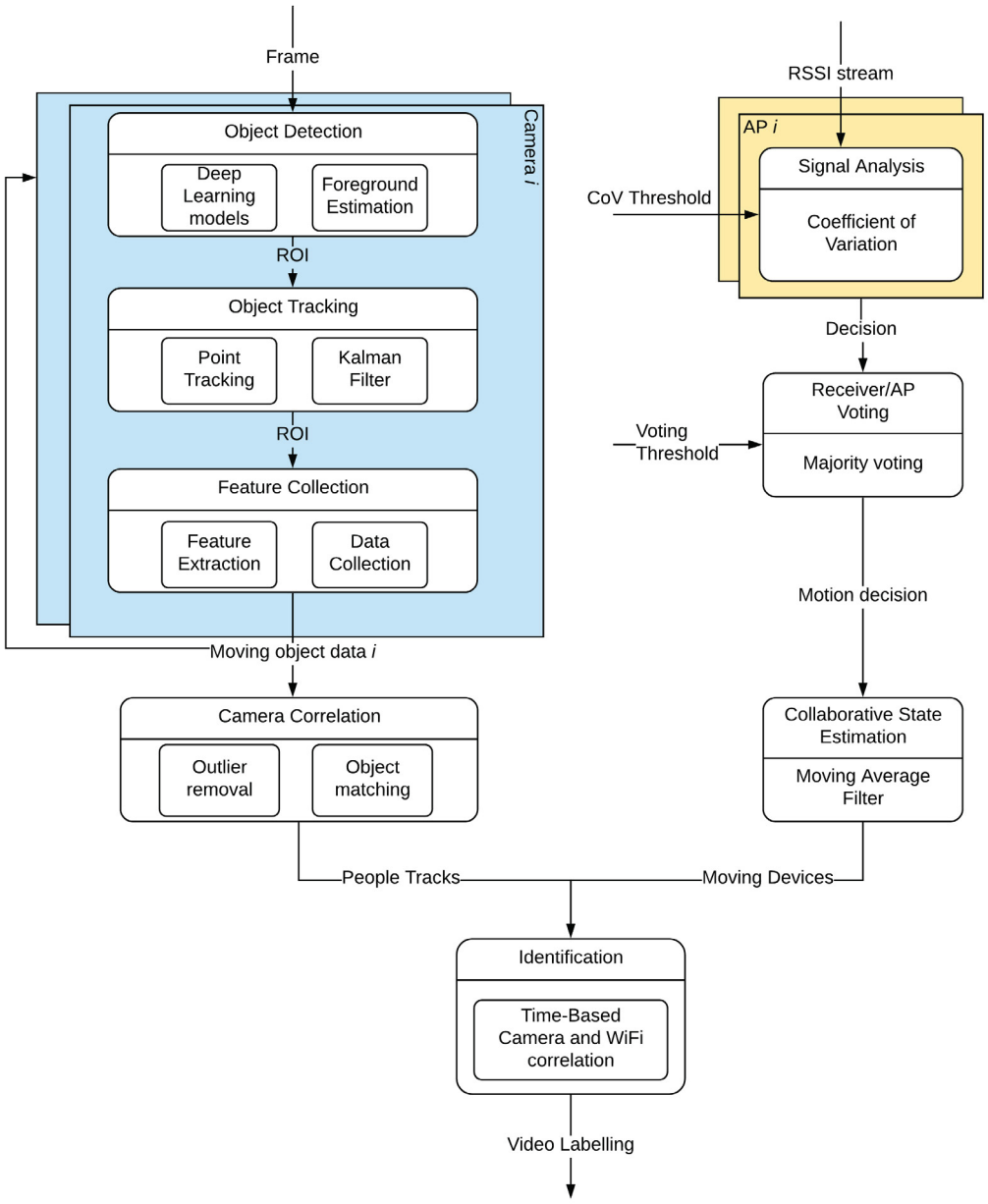

Figure 4.1: Framework Overview.

4.2 RSSI-based Device Motion Detection

Each packet sniffer captures incoming packets from a mobile device and collects a timestamped samples.

4.2.1 Coefficient of Variation

The CoV is a (unitless) statistical measure of the relative standard deviation of samples, the formula for which is shown in Equation 4.1. Consider $S = \{s_k | t_p - t_k < \tau\}$ where t_p is the present time and τ is a predefined threshold, where S is a set of RSSI measurements received within last τ seconds. The CoV and a threshold is used for deciding if a device was in motion during that interval. This method can serve as a benchmark for other approaches since it is the least demanding in terms of computation.

$$C_v(S) = \frac{\sigma(S)}{\mu(S)} \tag{4.1}$$

where τ is 2 seconds in our case since it is sufficient to capture fluctuations in RSSI produced by human motion.

Since we utilize camera time as the reference time for all streams, time is incremented by 0.044 seconds at a frame rate of the visual sensor. For each frame received by the camera, we evaluate the CoV of the RSSI samples obtained from sniffers. Then, the sensor "decides" if a device started moving by applying a constant threshold to each of the computed CoV streams. A motion detection from each sniffer is compared with other streams for collaborative decision. Aggregate vote is obtained by combining the decsions from all the APs point of view on a frame-by-frame basis. We apply a Moving Average Filter on the aggregate voting step's result to smooth the final decision and remove noisy detections. It is necessary to provide the system with a number of votes considered as a majority. As a result we obtain a set of devices, motion states and time intervals.

Figure 4.2 illustrates a calculated CoV stream during a particular stage of experiment for a Huawei Nexus 6P device. From the figure, it can be seen that there is a correlation between CoV and a device motion, which is labelled as

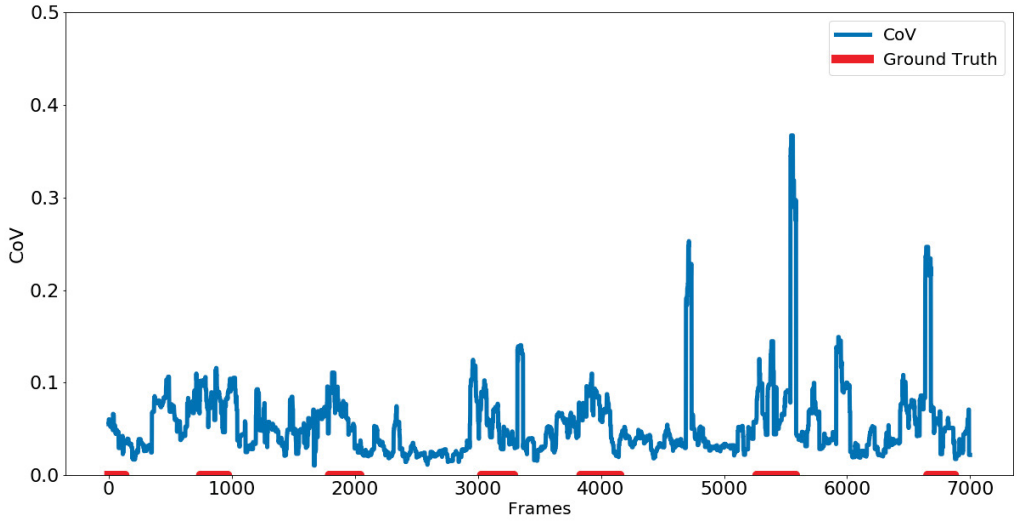

Figure 4.2: CoV time series pertaining to a single RSSI stream vs. the ground truth for a Huawei Nexus 6P.

the ground truth in the image. There is also a number of spikes present for a short period of time which is one of the reasons for applying moving average filter later on. Thresholding this time-series produces a frame-based decision from a single AP point of view.

We have examined various window sizes for the Moving Average filter described previously, and presented the result for the two different devices in Figure 4.3. As it can be seen from the figure, both the decisions on both devices are more accurate with a four second window time interval.

4.2.2 Bayesian inference

An alternative approach that was used for RSSI analysis is Bayesian Inference. This approach provides a probabilistic estimate of a device motion. We calculate the probability of a moving device according to the formula given in Equation 4.2. The Equation 4.2 is derived from the generalized Bayes Theorem, where the goal is to determine probability of a device moving provided that the system observed a sequence of the RSSI value at any time slot $i + 1$ depends on the RSSI at the previous time slot and the conditional probability follows normal distribution. We also assume that The RSSI value at time slot

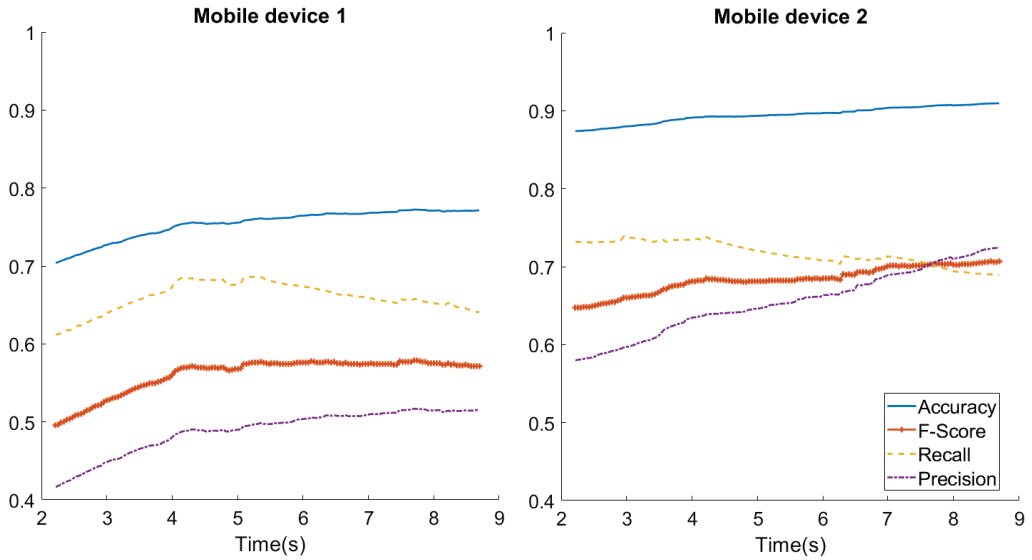

Figure 4.3: Moving Average filter window size vs detection accuracy.

$i+1$ is independent of all previous measured RSSI values given the RSSI at time i. We can write the posterior as

$$P(M|RSSI_i, RSSI_{i+1}) = \frac{P(RSSI_{i+1}|RSSI_i, M) * P(RSSI_i|M) * P(M)}{P(RSSI_i, RSSI_{i+1})} \tag{4.2}$$

where $P(RSSI_i, RSSI_{i+1}) = P(RSSI_{i+1}|RSSI_i, S_j) * P(RSSI_i|S_j)$ and $P(M)$ is the probability of being in the moving state, i.e, the likelihood.

Each term in the above formula can be estimated from the measurements (empirically). In Section 3 we have explained the data collection process for estimating static and moving device's RSSI probability density function (PDF). The PDF explains the probability of an RSSI value given that it is moving or static, $P(RSSI_i|M = Moving)$ in the Equation 4.2. A similar distribution was used for calculating $P(RSSI_i|M = Static)$. The joint probability $P(RSSI_i, RSSI_{i+1})$ is obtained through the sum of probability distributions of the RSSI values in static and moving states that is:

$$P(RSSI_i, RSSI_{i+1}) = \sum_{M \in \{Static, Moving\}} P(RSSI_{i+1}, RSSI_i, M) \tag{4.3}$$

A moving device's RSSI values are more likely to follow a normal distribution as it is evident from Figure 4.5. However, a static device's histogram of RSSI values follows bimodal distribution as it can be seen from Figure 4.4.

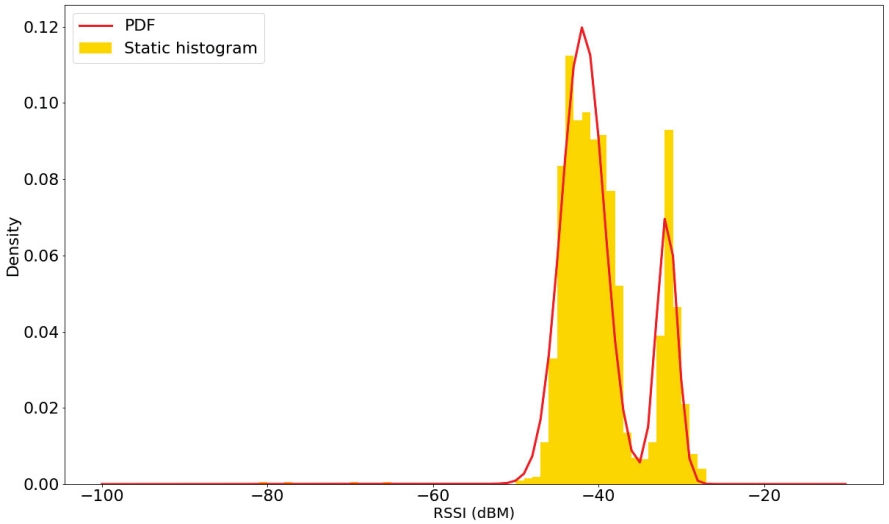

Figure 4.4: Histogram of RSSI values for a static device vs Gaussian Mixture Model fit (Device 1). The similar pattern can be observed in other devices.

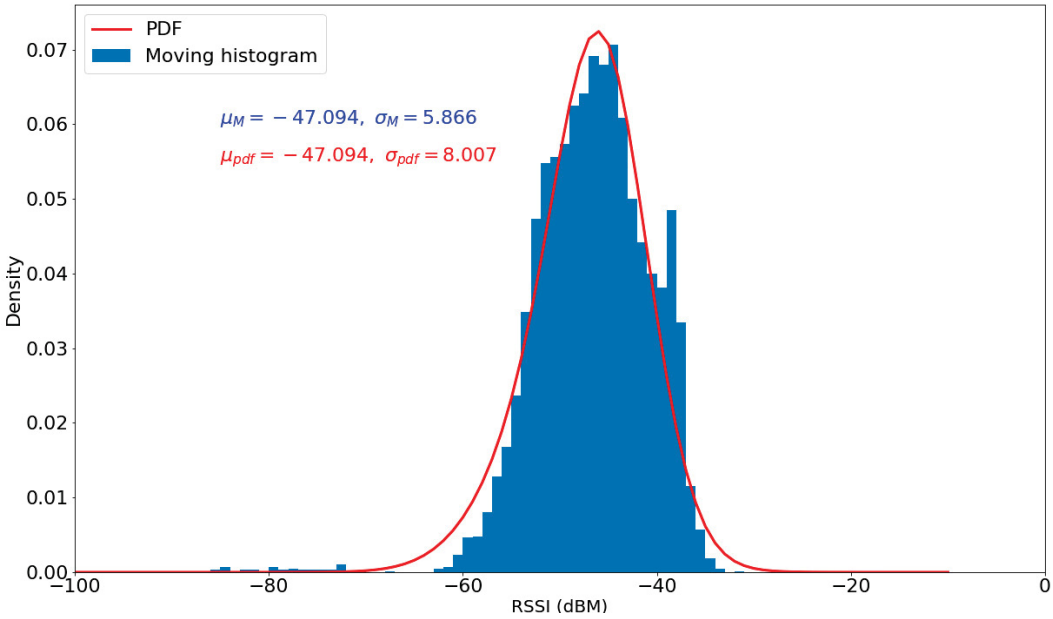

Figure 4.5: Histogram of RSSI values for Device 1 vs Normal distribution fit.

4.2.3 Supervised Learning methods

A class of machine learning algorithms, which "learn" dependencies for classifying inputs according to an expected output, is referred to as *supervised*. Support Vector Machine algorithm is a type of a supervised learning which requires ground truth data to train the model. Since we are looking at time varying data and want to capture changes in signal, we utilize motion indicators such as variance of RSSI, standard deviation and coefficient of variation. There are three features for each sniffer's antenna stream; 36 features to train in total, 3 antennas by 4 APs by 3 features.

We train the model on the data collected from the motion-based experiment data. The data captures proportions of the original data, containing around 50 percent of motion data. The time-series contains the 36 features for each timestep and a device motion label corresponding to that period of time. The 36 features that were computed previously are used as input to the SVM algorithm and motion label as the expected output.

4.2.4 Recurrent Neural Networks

We have collected a data of RSSI from all different mobile device manufacturers as a training set. The data is used to train Neural Network models for each of the devices to classify motion or no motion in the WiFi domain. The deep learning methods had proven to achieve high performance for time series classification as reported in [52].

One of the properties of Recurrent Neural Networks (RNNs) is their ability to "learn" key features for classification and other purposes, so a developer does not have to derive new data from existing source. This can result in better performance due to unlimited data insights obtained from raw data, yet it may require larger volume of training data.

In our work we use data of different length for each of mobile devices with similar motion patterns, taken from the same data set as the testing set, to be used as a learning data set. The training set size is determined by the number of mobile devices present in each experiment, thus, Experiment I

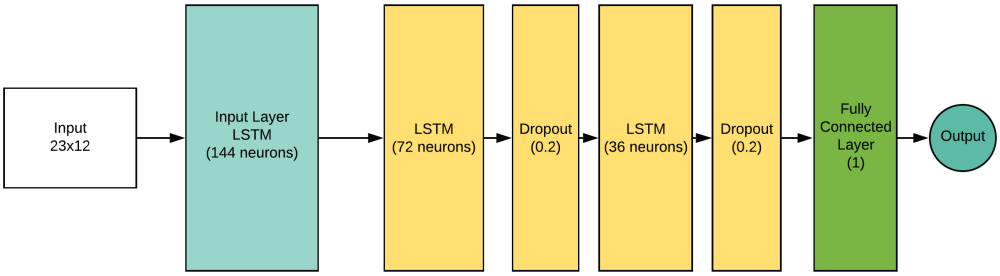

Figure 4.6: Structure of the Recurrent Neural Network used for RSSI time-series classification in moving and static states.

can have up to 70% of it's duration used for training purposes, amounting to 1400 half-second intervals in 1035 second experiment by 23 RSSI samples by 12 AP antennas by 3 mobile devices', total over 4000 samples. Moreover, the data collection experiment described in Section 3.3 can provide us with additional 600 samples per device. The RNN we build consists of a single Long Short Term Memory (LSTM) layer and a Fully Connected layers with Rectified Linear Unit (ReLU) activation function.

Our implementation of the machine learning models relies on TensorFlow and Keras python libraries [12], [43].

Chapter 5

Moving Object Tracking

The goal of Multiple Object Tracking (MOT) systems is to identify a particular object throughout a sequence of frames. The problem is different from a single object tracking since the system may not be informed about the objects to be tracked. Therefore, it is necessary to identify and track objects based on certain properties, such as visual features or motion.

We use state-of-the-art computer vision algorithms to detect moving people in a scene. A video recorded for the experiments is a sequence of RGB frames captured at a regular time interval, every $\frac{1}{23}$ seconds in our case. The general idea of object tracking can be explained through object detection and association in a frame sequence. Detecting moving individuals in the scene can be done using two approaches: looking at the changes in the scene or identifying a person like objects in an image. Both cases have been studied extensively and a number of solutions have been proposed in each case. In this thesis, we utilize the state-of-the-art approaches presented in MOT challenges [5], [36]. The competitors are given a number of video sequences with various length and environment conditions. The goal of the challenges is to propose a solution which can identify and track as many people in the scene as possible, though challenged by occlusion, moving camera and number of individuals.

The first object detection approach we consider relies on fluctuations in pixel values to calculate the foreground mask, i.e. an image with black pixels as a background, and analyze it to detect objects. The second is a more complex

approach to object detection where a machine learning model is trained to "search" for particular objects, such people standing still in a given frame. Then, after each frame is analyzed for objects in the scene, it is necessary to associate the identified objects across multiple frames to create a track of detections. However, an individual may not be identified in a frame causing disturbances in object tracking. The problem can be mitigated, in some cases, with filtering.

This chapter describes a method for multiple individual tracking using RGB cameras in an indoor environment. Here we explain the details of Background Subtraction and state-of-the-art Pedestrian Tracking methods used for object detection. Then, we apply a filtering technique, namely Kalman Filtering, to estimate possible object location when the objects are not detected. The next objective for the system is to track the identified object along the frames by associating detections to known sequence of boxes in different frames. Finally, the tracks historical information is used for object motion detection and combined with the moving devices data to associate one to another. Note that we do not exploit the multiple cameras, and instead focus on processing carried out from a single camera point-of-view.

5.1 Object Detection

The first step in object tracking is to detect the object, in our case an individual, based on either motion or visual features. A representation of object detection in a visual domain is a bounding box, $b_i \in B$, in the form of (x_i, y_i, w_i, h_i). The x *and* y are the coordinates of the top left corner in an image, B is a set of all bounding boxes for a frame and w_i and h_i are corresponding width and height of each box. These bounding boxes are the output of object detection algorithms described below.

Object detection in a frame with motion typically utilizes statistical analysis of the foreground mask, such as the number of connected components, and the area and centroid calculation [33]. We first describe the methodology adopted in this work to detect motion in a room: for each camera stream,

we analyze a sequence of frames to detect moving objects, then track these objects along the sequence of frames by using a Kalman Filter.

5.1.1 Foreground Detection

Gaussian Mixture Models are used to model values for each pixel to classify later as a part of the background following the method proposed in [17]. A moving individual can cause deviations in modelled pixel values, which is labelled as a foreground mask, i.e., white pixels in a black and white image. Figure 5.1 demonstrates an example of object detection done by background subtraction and blob analysis. The mask is then analyzed for connected components and their areas to identify moving objects. A certain threshold value for the area can be set to reduce spurious detections in noisy foreground. If a component has an area larger than the threshold it can be considered as a detection.

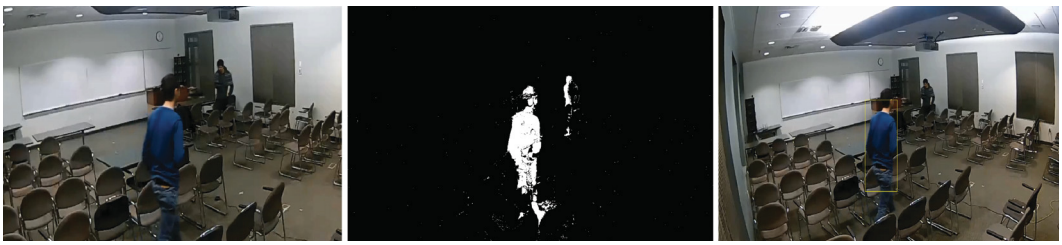

Figure 5.1: **Left:** Raw, **Center:** Foreground Mask, **Right:** Object Detections.

5.1.2 Pedestrian Tracking Model

Another approach to visual object tracking is to utilize pre-trained deep neural networks, such as in the methods mentioned in [16], [39]. State-of-the-art methods are proposed as a part of Multiple-Object Tracking (MOT) challenge where the goal is to detect and track moving individuals in an outdoor environment [34]. There was a number of top performing solutions presented, yet the implementations of several of them were not available at the time of writing this thesis. Therefore, we proceeded with one of the top three contenders, the method presented in [5] which is considered to perform multiple object

tracking in real-time on a node with a specific configuration. The processing in [5] was done with the help of Graphical Processing Unit (GPU), in particular NVidia Titan X GPU. The algorithm processes video frames and provides bounding boxes for the identified objects.

The main idea of this kind of algorithms is to classify objects by their type from an image frame, such as person, dog, cat and etc. An assumption present is that objects follow a particular visual structure that can be recognized by a machine learning algorithm. For example, a distinct structure of upper body can indicate that there is a person in the frame. There are models, such as [39], which are trained to detect and classify a number of objects, yet some are specifically designed to identify standing people. In our work, we utilize one of the pedestrian tracking models for outdoor and indoor environments, since it is more related to the scope of this thesis.

The object detection model takes an RGB frame as input and produces a collection of bounding boxes for each person object. Since the objects can overlap each other and provide duplicate detections, it is necessary to choose the one with the highest confidence level.

5.2 Kalman Filter

Kalman Filter is a method for a system state estimation that can be used to predict the system state in the next time step or identifying outliers, i.e. erroneous associations in object tracking [9].

The state of a dynamical system can be described by Equation 5.1

$$s_i = As_{i-1} + w_{i-1} \tag{5.1}$$

$$m_i = Cs_i + q_{i-1}, \tag{5.2}$$

where, i is the time index, $s_i = \begin{bmatrix} x_i & \dot{x}_i & y_i & \dot{y}_i \end{bmatrix}^\top$ is a vector that contains the coordinates and corresponding velocities of the object, m_i is an observation obtained from sensors set up in an environment.

The matrix A in Equation 5.1 represents the state transition model, i.e., how the system is expected to change between two successive samples. Vectors

w and q are the process and measurement noises assumed to follow Gaussian processes and modelled through normal distribution.

We utilize Kalman Filter on a frame basis which predicts current estimate of the system state based on previous measurements, rather than recalculating the entire sequence of detected objects.

The filter provides a moving object's coordinates in the frame which can be used as a synthetic object detection and for smoothing the trajectory of movement. The predicted location can also be used for track association with objects detected in the current frame.

In the active scene, the motion-based object detection algorithm is used to identify regions in the current frame. A Kalman Filter is used to predict the new location of a previously identified object [9]. The prediction and correction steps are defined in Equations 5.3 and 5.4. The prediction is based on our expectation of the system, A, transition model and the state in the previous time step, s_{i-1}. We utilize a Constant Velocity model which is commonly used in object tracking systems with Kalman Filter [35]. The model is expressed as $A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$, where Δt is the length of the interval between two successive samples.

$$\hat{s_i} = As_{i-1} \tag{5.3}$$

$$s_i = \hat{s_i} + K(m_i - C\hat{s_i}) \tag{5.4}$$

The update Equation in 5.4, essentially, combines predicted and measured states. The rationale behind the formula is to use expected system state to filter possible process and measurement noises. The Kalman Gain, K in Equation 5.4, is a special term which can provide preference, towards either the measurement term or the current state estimate. This parameter can provide more responsive behaviour of the filter and tuned to achieve better results. Nevertheless, an optimal Kalman Gain is calculated through the covariance matrix of errors between estimated and measured states. Object to track association is then performed based on minimizing the distance between

detections and tracks.

5.3 Track Association

A track in the MOT problem is a sequence of bounding boxes, such that each b_i identifies the same object over a sequence of frames. For each object, b_i with i as object index, and track, r_j where j is a track index, we compute a distance matrix where each entry represents a "cost" of association between track and object. If there is no track identified yet, each bounding box serves as an initial entry in the track sequence. The distance is defined as

$$d(b_i, r_j) = \sqrt{(x_{b_i} - x_{r_j})^2 + (y_{b_i} - y_{r_j})^2} \tag{5.5}$$

A cost matrix is computed by Equation 5.6

$$D = \begin{bmatrix} d(b_1, t_1) & d(b_1, t_2) & ... & d(b_1, t_j) \\ d(b_2, t_1) & d(b_2, t_2) & ... & d(b_2, t_j) \\ ... & & & \\ d(b_i, t_1) & d(b_i, t_2) & ... & d(b_i, t_j) \end{bmatrix} \tag{5.6}$$

The D matrix explained in Equation 5.6 represents a "cost" of association of tracks to the detected boxes. The main goal is to determine the "cheapest" association for each of the tracks, solving it with a greedy algorithm, i.e., providing sub-optimal solution for each of the tracks. Since the distance represents the Euclidean distance between detected object and last box of the track, the optimization goal is to match the objects that are closer to each other. The distances are computed from the boxes coordinates in the images. The objective is to minimize the total "cost" of box to track association, such that a bounding box, b_i, is mapped to at most one track, t_i, and no track can have two detections assigned to it. In case there is more detections than current tracks, unassociated boxes are considered as new tracks, whereas unassociated tracks utilize Kalman Filter prediction as a box in current frame.

5.4 Device-to-Individual Mapping

The last component of the identification is to "attach" possession of a device to moving individuals. The two data flows provide similar information in the form of: $\{p_i | p_i \in P,\ if\ p_i\ moving,\ where\ p_i = (MAC_i, st_i, ct_i)\}$ - is a set of moving devices, $\{r_i | r_i \in T,\ if\ t_i\ moving,\ where\ t_i = (Box_i, st_i, ct_i)\}$ - is a set of moving individuals, and st_i, ct_i indicate starting and ending times. The time intervals are used to determine how close the motion events are in the scene, what is the duration of those and how devices' motion patterns might correlate.

If the events ensue with a significant time differnece they are possibly unrelated. The assignment threshold, τ, serves the purpose of disassociating events that occurred in an interval of more than half a second.

Chapter 6

Results

This chapter presents the results of our methodology, thereby demonstrating the performance of each sensor modality. We provide results for the two realistic data experiments conducted in a conference room using the process of data collection presented in Chapter 3.

We use several standard performance metrics, such as accuracy, precision, recall and f-score which we define next. Depending on the sensing modality, visual or RSSI, we define the correct identification of device's motion (RSSI) in a particular time interval as a true positive (TP). A true negative (TN) is the opposite, a device is static and the AP consensus finds it static. We define a false negative (FN) as a decision that a device is deemed static, although it is, actually, in motion. A false positive (FP) is exactly the opposite of that.

CV modality has a different definition for TP, TN, FN and FP which is due to the type of output the algorithms provide. Since our methodology is based on individuals' motion events, it is reasonable to label a TP if the number of moving objects matches the quantity of tracked objects. We note that this is a stricter definition of TP compared to defining it for when there is at least one moving object, but not necessarily the correct number of objects. A TN will be an event when the object is detected, but remains static. We do not consider undetected static objects as TP because of several possible reasons it might be undetected, such as having small "blob" size or being occluded, while being in motion. Therefore, TP for CV is a correct identification of the number of moving people in the scene, and a TN is identification of static

individuals in the scene. FNs are the detections which are considered to be static, whereas they are actually moving, and FP is the oposite, at least one static person is identified as moving.

Equations for the accuracy, precision and recall are shown below:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.3}$$

$$F1 - Score = 2\frac{Recall * Precision}{Recall + Precision} \tag{6.4}$$

Since our methodology derives devices' or individuals' motion on a frame by frame basis, we label the decisions accordingly on a per-frame basis. Below we describe analysis results in each component of the framework, RSSI and CV, and explain how the methodologies differ in terms of data preparation.

6.1 Motion Detection with RSSI

This section describes the results of RSSI time series analysis for device motion detection using a variety of methods. The CoV thresholding and multiple AP voting scheme are used as a benchmark for other methods since they require the least amount of computation. This method requires parameter tuning, in particular, providing thresholds for a number of decision points which is not ideal for a generalised system architecture. It is possible that the threshold values need to be adjusted to perform in different environments; thus, making the solution not applicable to some cases.

We subsequently use a conventional machine learning (ML) classification methods, such as Support Vector Machine (SVM). These methods still depend on the features chosen by user, yet it can be trained with little user involvement. The performance of the model depends on the amount of training data

Table 6.1: Overview of the methods presented in this section for RSSI analysis.

Methodology	Training	Performance	Computation
CoV	Only Tuning	Satisfactory	Low
SVM	Yes	Good	Moderate
RNN	Yes	Great	Heavy
Bayesian Inference	Yes	Inconclusive	Low

and how features correlate with each other, and with the motion label (i.e., the ground truth). These methods were described in Chapter 4.

The limitation of manual feature engineering is that it relies on insights of the model designer. Thus, we adopt with a more complex ML method, Recurrent Neural Networks (RNNs), which can analyze raw data, automatically exrtract meaningful features, infer useful correlations on their own and build the most suitable model. Finally, we compare the results obtained from these methods to the Bayesian Inference approach where a device's motion is deduced based historical conditional distribution of RSSI values.

Table 6.1 demonstrates a brief overview of all the models discussed here for RSSI motion detection. Several algorithms provide "Good" performance for the motion detection of the devices, yet we proceed with SVM, due to a number of reasons. SVM requires significantly less computation resources for training the model, generating features and validation. SVM provides a comparable result for multiple devices and on average requires less training data.

6.1.1 Coefficient of Variation

The Coefficient of Variation (CoV) is a ratio of the standard deviation of a sample set to its mean, which is also known as relative standard deviation. In other words it can show how significantly the RSSI values fluctuate with respect to the average of the recent samples. We utilize the CoV of RSSI values to decide if a device is moving. This gives us a more reliable detection compared to standard deviation because the CoV represents relative fluctuations in the RSSI.

Firstly, we needed to identify CoV thresholds for each sniffer/AP and their

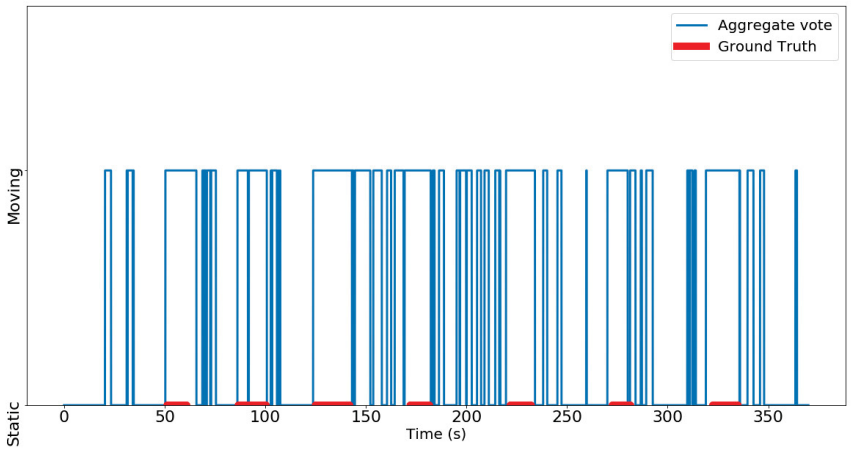

Figure 6.1: An example of a low CoV threshold chosen for motion decision, Device 2. Red line indicates intervals where ground truth indicates movement.

antennas to choose the best performing one out of the three antennas each AP possesses. We utilized the previously mentioned performance metrics to indicate the better performing antenna first and then use those antennas for finding the threshold. The APs are treated separately and only antennas in the sniffer are compared to each other. Then, for each sniffer, there is a custom CoV threshold determined according to the metrics. An example for one such antenna is shown at Figure 6.1. Here a low value for CoV threshold was chosen resulting into multiple FPs, moreover worsened by disturbed moving intervals. The choice of threshold was done experimentally by comparing the output decision to the ground truth, to maximize the overlap of decision made by an AP and the latter.

Combination of multiple APs decisions is done by aggregating "votes" on every device's motion on a frame-by-frame basis. An example of aggregated voting of the APs on the device motions is shown in Figure 6.2, where the y-axis corresponds to the number of APs determining that the device is moving. It can be seen that the aggregate decision also has several disturbed intervals. This motivated us to apply a filtering technique which can smooth out irregularities in the decision and compare with the ground truth.

Thresholding CoV alone resulted in identifying many disjoint intervals of

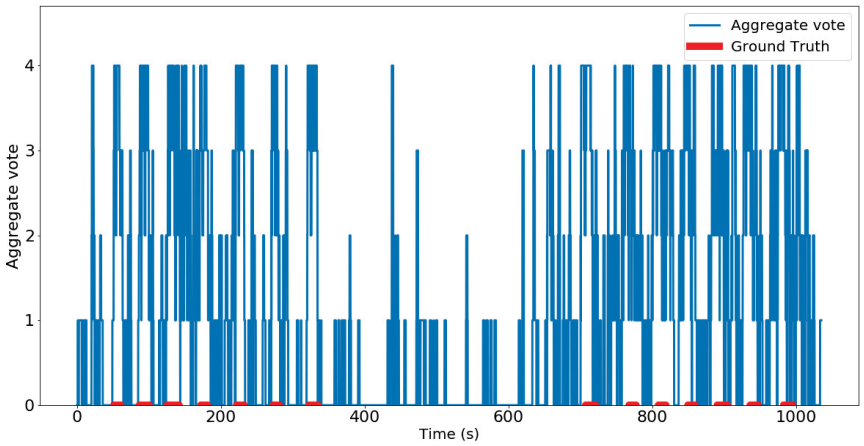

Figure 6.2: An example of unfiltered aggregate voting, Device 2.

device's motion, which was addressed by application of a Moving Average (MA) filter. The filter is applied onto the aggregate vote, such as the one illustrated in Figure 6.2. The size of the sample set used for calculation of the MA filter is determined according to Figure 6.4. It can be seen from the graphs that its performance flattens after the four-second window size, i.e., longer history may not provide significantly better results, and, furthermore, delays AP consensus calculation, if it were to be performed in real-time. The results of determining whether a device is moving are provided in Figures 6.5 and 6.6. There are still several erroneous detections for a short interval of time, which could have appeared due to signal fading. The phenomenon is related to multi-path propagation of signal which deteriorates in more cluttered environments.

Next, we examine an appropriate threshold for reaching consenus among APs for a device motion in a given time step. It was essential to understand if a single AP decision is more efficient than the agreement of multiple APs scheme. The reason is that a single AP may have a "cleaner" view of RSSI fluctuations due to, for example, a proper line-of-sight or a closer proximity of its transceiver. Nevertheless, Figure 6.3 demonstrates that the majority voting is a better option for a collaborative motion detection consensus. The majority AP consensus decides if a device is in motion, otherwise, it is considered static.

Analyzing the performance of AP consensus on device motion detection

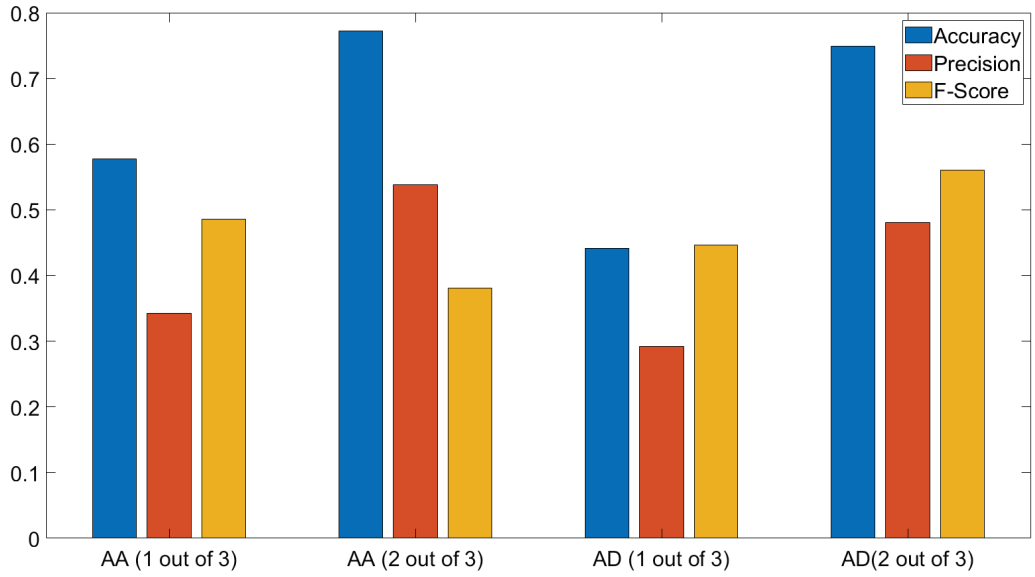

Figure 6.3: Performance of multiple AP consensus with relation to the consensus vote threshold and CoV threshold uniformity. AA represents antenna-agnostic, AD is antenna-dependent CoV threshold.

during different intervals of Experiment I, it was observed that certain intervals produced more favorable results in terms of motion detection than others. Tables 6.2 and 6.3 demonstrate a similar picture in performance of motion detection for both devices. Precision and recall are above average in both devices for the first five minutes, during which each individual was holding only one device. The best performance can be observed for the second five minute interval of the experiment, when only one device was held by an individual and the others were at rest. We remark that for the interval when the device was static, interval 5-10 minutes in Table 6.3, no TP events were possible, yet the TN rate was 0.99. The last interval of the experiment has the worst precision, recall and accuracy, and it is when an individual picked up Device 2 and held it through the rest of the experiment. We believe that this behaviour can be attributed to the close distance between the devices and consequent signal fading before reaching APs. It is also because of this interval that the overall performance drops significantly. Omitting this portion of the experiment would result in the performance comparable to the first five minutes of the experiment.

Table 6.2: Results for various experiment intervals of Experiment I (Device 1).

Interval	Precision	F-Score	Recall
0-5 minutes	0.59	0.62	0.65
5-10 minutes	0.61	0.71	0.87
10-15 minutes	0.30	0.37	0.49
Entire experiment	0.48	0.56	0.67

Table 6.3: Results for various experiment intervals of Experiment I (Device 2).

Intervals	Precision	F-Score	Recall
0-5 minutes	0.92	0.87	0.82
5-10 minutes[1]	0	0	0
10-15 minutes	0.50	0.54	0.59
Entire experiment	0.67	0.69	0.71

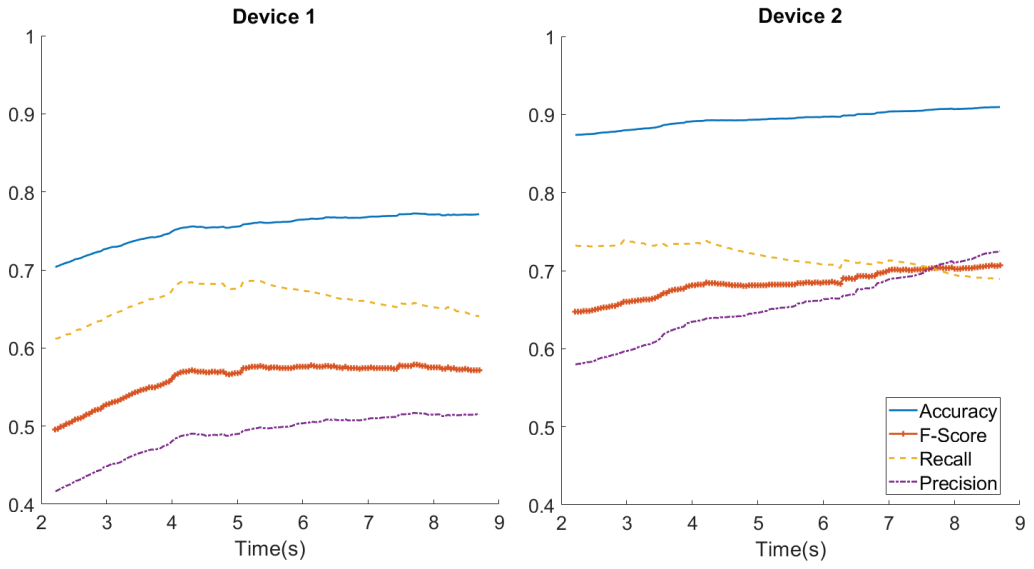

Figure 6.4: The effect of Moving Average Window Size on Performance.

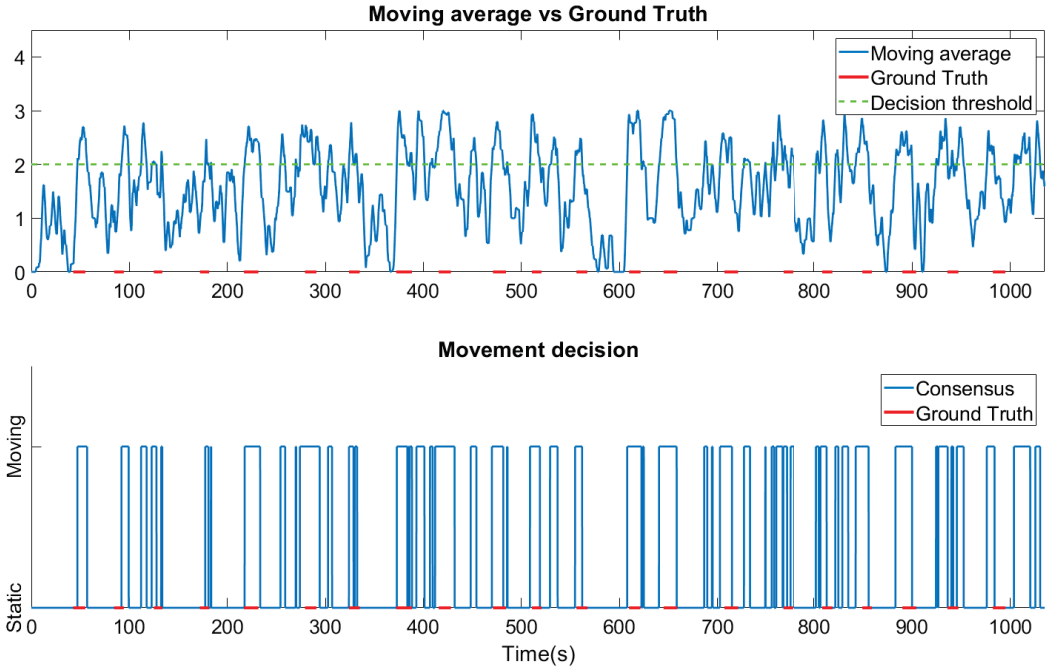

Figure 6.5: AP Consensus vs Ground Truth Motion, Device 1, Experiment I. Red line indicates intervals where ground truth indicates movement.

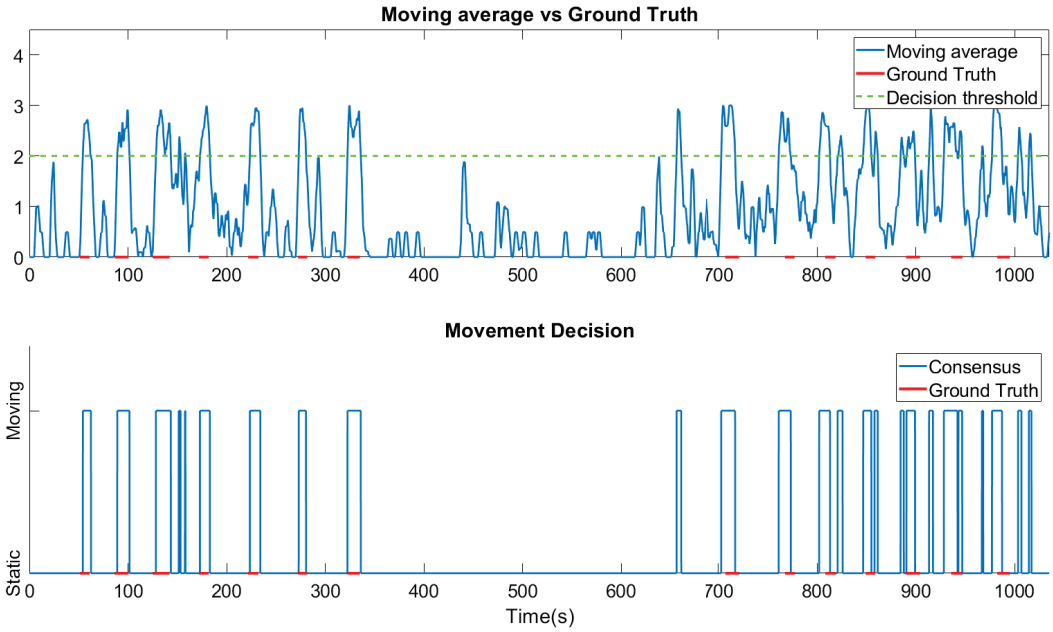

Figure 6.6: Overview of thresholding technique for movement detection based on CoV values, Device 2, Experiment I. Red line indicates intervals where ground truth indicates movement.

Table 6.4: Features engineered from the RSSI time-series.

Feature per time-series	Equation
Standard deviation, σ	$\sqrt{\frac{\sum(x_i-\overline{X})^2}{N}}$
Variance, σ^2	$\frac{\sum(x_i-\overline{X})^2}{N}$
CoV, c_v	$\frac{\sigma}{\mu}$

6.1.2 SVM

The previous methodology required careful choice of thresholds for each of the AP antennas, the number of AP to be considered as a majority, and the MA filter size. In this subsection we describe results of training a supervised ML model, SVM which we introduced in Chapter 4, the features used, and provide the performance against the training set size. The features are displayed in Table 6.4, which are derived from each of the APs' antennas, totaling 12 by 3, 36 features.

Next we provide comparison of the model performance where only measurements of one antenna from each AP were chosen. We tested different combinations of antennas, choosing the same antenna across different sniffers and a randomly selected antenna from each AP.

Moreover, we examine how the model could be generalized to different experiments, and what can be achieved if we have access to the output of one or two devices. In particular, we utilize Experiment I and II in the data set described in Chapter 3 to use as both training and validation sets.

To evaluate the performance of the model on our data set we utilize five-fold cross validation technique and provide average performance observed. The idea of the evaluation approach is to create five different data partitions, where each partition is used as a testing set in each of the different five rounds of data validation [23]. The other four partitions are used as training set for the model for the validation stage and averages of each performance metric is reported.

The 70% of all data will be used for training and the rest for validation, whereas the cross-validation method will divide data into five equivalent-size

[1]No TP were detected since the device was static, the TN rate is 0.99.

segments of training data and validate on the remaining 20%. The larger number of folds and subsequent smaller validation set is justified by a more representative model performance, i.e., certain parts of data may perform significantly poorer than others.

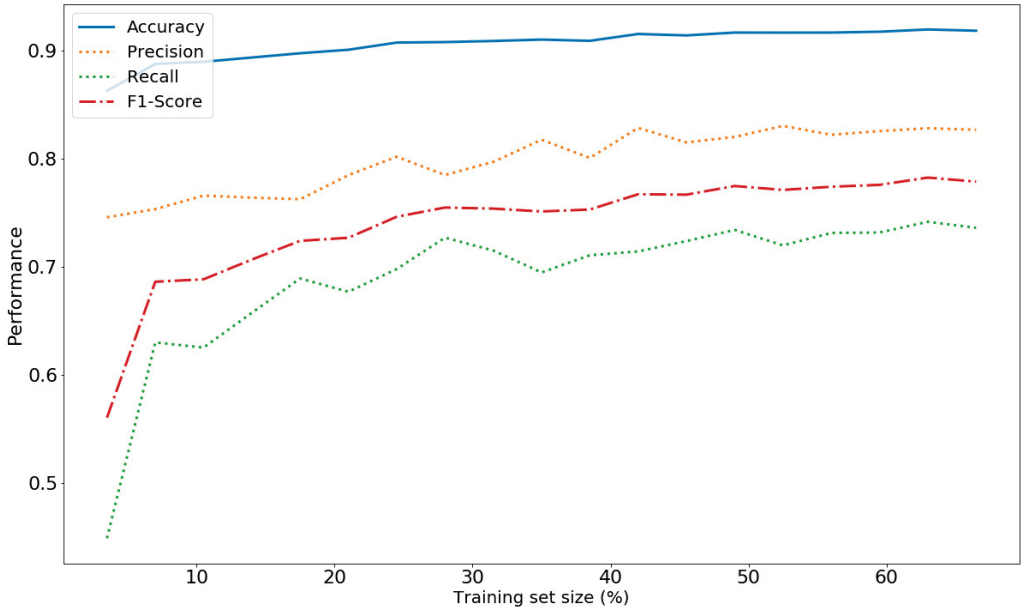

Figure 6.7: Performance vs Training set size.

The results and discussion on various methods of training the models are provided below:

- **Utilizing measurements of different antennas.** Each AP is capable of reporting RSSI measurements from three different antennas and a combined signal strength. Thus, it was necessary to test if it is sufficient to utilize a single antenna or multiple antennas' readings. First, we chose antenna one from each AP and utilized those measurements for the SVM model training. For this type of antenna utilization, average performance of different antenna measurements is reported in Table 6.5. Second, we examined how combination of different antennas across multiple APs affects the performance. Average performance is shown in Table 6.6, where it can be seen that the performance of random and specific antenna set ups are comparable to each other. Nevertheless, it

can be outperformed by applying a larger amount of RSSI data from each AP, such as utilizing all the RSSI measurements.

Precision	0.69
Recall	0.53
F1-Score	0.61
Accuracy	0.80

Table 6.5: Five-fold cross-validation (same antenna at each AP), Experiment II.

Precision	0.64
Recall	0.51
F1-Score	0.57
Accuracy	0.82

Table 6.6: Five-fold cross-validation for randomly chosen per-AP antenna, Experiment II.

- **Examining sufficient training size for the SVM** Figure 6.7 illustrates how precision and recall change with the amount of training data available for the model. From the 70-30% split it can be seen from Figure 6.7 that there is no significant improvement after training on more than 25 % of the sample size, which is about five minutes of Experiment I.

  Five-fold cross validation provides a different result on average possibly due to a single poor performing fold. The results are shown in Table 6.7. From the results it follows that only 25% of the experiment data is sufficient to understand device's motion intervals with accuracy of nearly 90% varying with the device make.

Precision	0.77
Recall	0.81
F1-Score	0.78
Accuracy	0.86

Table 6.7: Five-fold cross-validation, Experiment II.

- **Cross-device training.** This test was designed to evaluate how the model could be generalized across different device manufacturers. The

model training involved computation of signal features, such as standard deviation, variance and coefficient of variation. The features can be calculated for each mobile device RSSI time series which was our next step. It was necessary to apply trained models on other devices' data to understand the possibility of model generalization across devices but in the same environment. A possible reason for these results could be overfitting of the model on the training data which consists only of RSSI features from a single mobile device. From Table 6.8 it can be seen that training the model on only one device data may lead to overfitting, i.e., the model performance on the training set is much higher than that of testing data. Therefore, we utilized data from both devices as training data and run cross-validation test on the rest of data.

	Device 1 -> Device 2	Device 2 -> Device 1
Precision	0.5	0.25
Recall	0.15	0.3
F1-Score	0.23	0.27

Table 6.8: SVM cross-device performance; trained X -> tested Y, Experiment II.

- **Multiple device training.** Proceeding with partial knowledge of motion patterns in multiple devices streams allows a better performing model across all devices. The training and validation process started with splitting the original RSSI streams into discrete, non-overlapping partitions of data, each covering 12 timesteps, time intervals between video frames. Then, features depicted in Figure 6.4 are computed for each of the partitions of data, totalling 36 features. Half of the samples generated from each device stream are combined into one training data set, which is then used for cross-validation. The results for cross-validation of the joint data set are shown in Table 6.9.

SVM, heavily relies on the features used for modelling. This limitation could be solved by applying neural network models, such as Recurrent NN, where historical knowledge is taken into account and features extracted automatically.

	Five-fold cross-validation	Device 1	Device 2	Device 3
Precision	0.88	0.91	0.87	0.54
Recall	0.81	0.88	0.23	0.81
F1-Score	0.84	0.89	0.34	0.65

Table 6.9: Results for five-fold cross-validation for multiple device-based data set. Columns Device X represent performance of the model on the test sets for each device separately, Device 2 was not part of training set and was added for performance comparison, Experiment II.

6.1.3 RNN

The RNN structure is depicted in Section 4.2.4. The number of layers was chosen incrementally, starting with a single LSTM layer and proceeding with 2 and more. The input layer is a Long Short Term Memory (LSTM) layer with ReLU(Rectified Linear Unit) activation function. It accepts input obtained from previous 23 measurements from a one second time window of 12 RSSI values, from three antennas for each of the four APs. We also report the performance of a single antenna per AP as input to the same RNN. Table 6.10 demonstrates five-fold cross-validation result for the RNN with a smaller input, only four RSSI streams per sample.

The output layer with only one neuron is activated with sigmoid function, which is widely used in binary classification tasks. There is a number of LSTM layers in between input and output layers each consisting of varying number of neurons with ReLU activation functions. Each layer is followed by a Dropout layer, which prevents overfitting of the RNN by ignoring a fraction of each layer weights. The number of neurons in the first layer corresponds to the input size and decreases with each layer to extract the most useful features. Three layers were introduced to this problem where the model is optimized with Adam to achieve better classification accuracy. Specifically the model is trained on a batch size of 12 samples for 5 epochs. The low number of epochs is chosen to avoid overfitting of the model to a certain data set.

The RNN performance can be compared with that of SVM using the same performance metrics. The results shown in Table 6.11 represent performance of the RNN using 30% validation data collected from the Experiment II. It can

be seen that RNN performs better with multiple AP's antenna data, although the increase in performance might not be satisfactory with larger input size. It can be seen that RNN requires a larger amount of training data to achieve acceptable performance. Notably, the data that was provided as input to this model was a raw time-series data without any prior feature engineering.

Precision	0.70
Recall	0.68
F1-Score	0.68
Accuracy	0.87

Table 6.10: Five-fold cross-validation on 70% of data, Device 1, Experiment II

Precision	0.78
Recall	0.75
F1-Score	0.76
Accuracy	0.91

Table 6.11: The highest performance of RNN trained on 70% of data, Device 1.

Precision	0.39
Recall	0.42
F1-Score	0.40
Accuracy	0.64

Table 6.12: The best performance achieved after training RNN on a five minute (30%) data, Device 1.

6.1.4 Bayesian Inference

We tested a Bayesian Inference (BI) approach on the collected data experiments, which demonstrated several essential challenges for the methodology to be studied. Bayesian Inference of device motion from its RSSI time-series provides a probability of the device's motion, given a vector of previously observed RSSI measurements. However, it is necessary to set a threshold value for the probability, τ,in order to conclude whether the device is moving, if $P(M|RSSI_i, RSSI_{i+1}) > \tau$, and static otherwise, as explained in Section 4.2.2. This section examines how the method's performance varies while adjusting

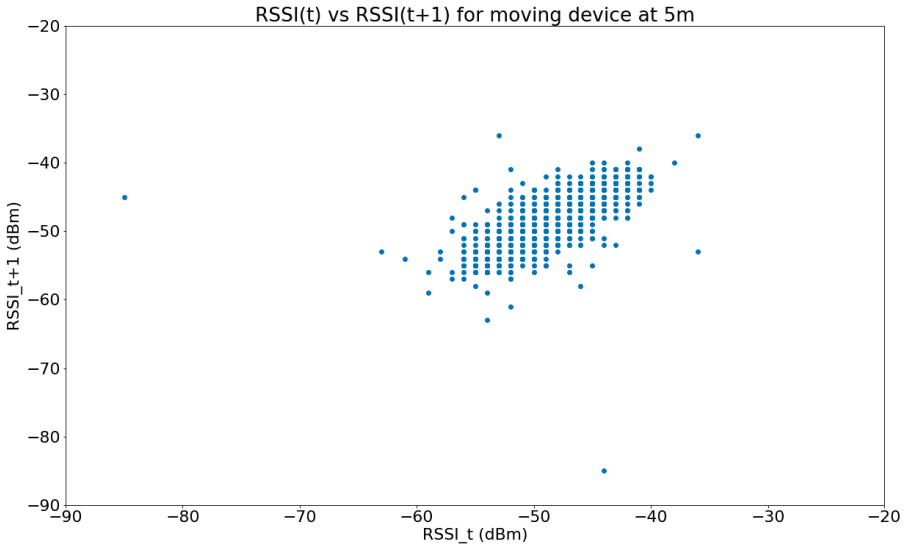

Figure 6.8: RSSI distribution of two related measurements.

the threshold value for the probability, discusses the application of moving average filter and a number of previously observed RSSI values.

One of the main questions for this methodology has been the strong auto-correlation of RSSI time-series terms especially at lag 1. The reason is that two packets transmitted by the same transceiver, within a small time interval, can be assumed to relate to each other. In other words an RSSI value received in a next time-step is more likely to have similar value or within one standard deviation to that observed in the current frame. We have evaluated how RSSI values at time step $t+1$ are distributed with respect to time step t. Figure 6.8 demonstrates RSSI value distribution for a static iPhone device at a five meter distance from the sniffer during Experiment 2. As it can be seen in the figure, the spread can be approximated with a normal distribution. Fitting a curve for each possible RSSI value might be cumbersome, since it is difficult to hold an experiment to cover all combinations of RSSI values. Therefore, we proceeded with a regression neural network model that is trained on the discrete data from the Experiment 2. The model allows to compute the mean and variance of a Gaussian probability distribution, $P(RSSI_i|RSSI_{i-1})$ for Equation 4.2. Using this method we calculated Equation 4.2 for different number of historical RSSI

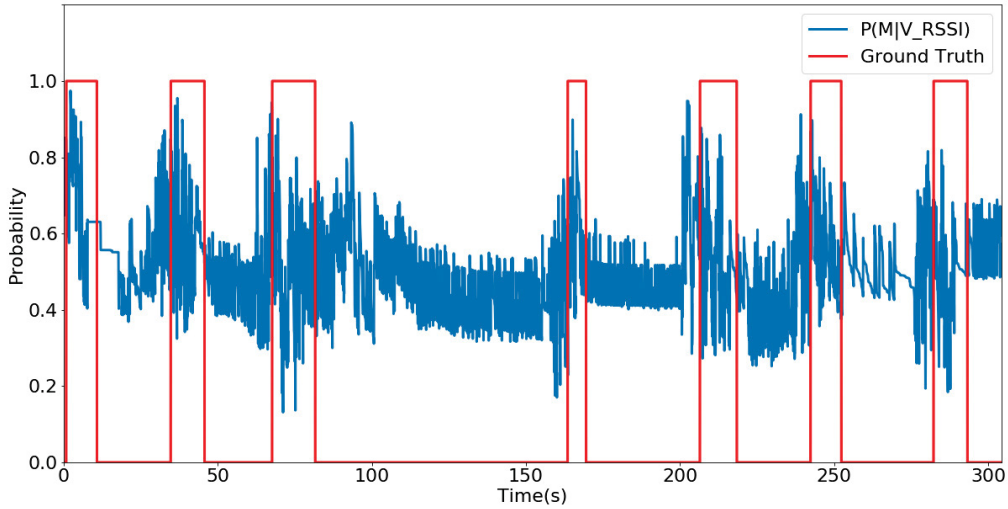

Figure 6.9: Probability of a device moving given an RSSI vector of two samples vs motion ground truth data.

values which is shown in Figure 6.10. The results were obtained with a fixed probability threshold, 0.7, chosen manually, to examine impact of historical terms addition.

Figure 6.9 demonstrates the result of Equation 4.2 for a single RSSI time series. From the figure, a slight correlation in probability and motion ground truth and the PDF peaks close to the beginning of the moving interval can be seen. Nevertheless, the correlation may not be utilized as easily due to frequent fluctuations in probability estimates. The decisions from each RSSI stream can be aggregated together in the same manner as CoV time-series from multiple APs. Since we have the opportunity to utilize multiple antennas from the same AP, we examine the impact of using all the antenna measurements in our probability calculation. Tables 6.13 and 6.14 demonstrate performance of different antenna sensor data usage. While the higher number of antennas per AP results in an increase in the recall, it is still insignificant in all the other metrics. Overall performance of the Bayesian Inference approach is inconclusive and outperformed by all of the previously discussed methodologies. To sum up, we observed increase in performance in most cases with more APs antennas used; nevertheless, there is still a challenge of proper combination of

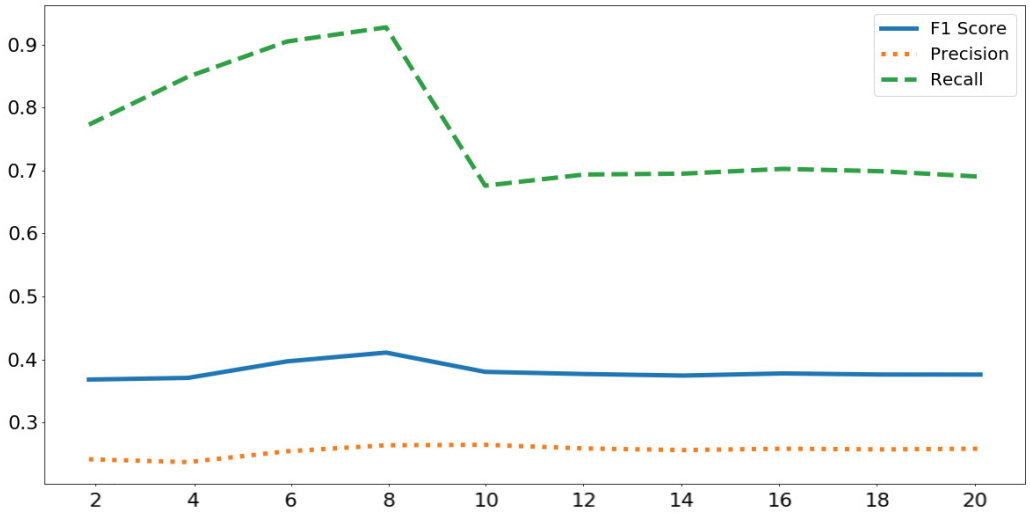

Figure 6.10: Bayesian Inference performance vs. RSSI sample size (unitless).

the measurements rather than simply aggregating the votes.

Precision	0.30
Recall	0.23
F1-Score	0.26
Accuracy	0.75

Table 6.13: Average BI performance for single antenna per AP, Device 1.

Precision	0.29
Recall	0.42
F1-Score	0.34
Accuracy	0.69

Table 6.14: Average BI performance for multiple per-AP antennas, Device 1.

6.2 Motion Detection with CV

Motion detection using CV relies on object detections throughout the video sequences. While some methods designed specifically to capture motion in the environment, others are not as efficient for that purpose [16]. This section examines the efficiency of object detection algorithms by comparing their output to the ground truth.

Labelling the number of moving individuals was done on a frame-by-frame basis to reduce a timing error that could result in reduced performance when we calculate the device-to-individual assignment.

6.2.1 Foreground Mask

Object occlusion occurs because of path intersections caused by multiple moving individuals, but also may be caused by placement of objects in the environment, such as tables and chairs. An example of such occlusion is demonstrated in Figure 6.11, where two individuals mistaken as being one. Another type of missed detection is an object further away from the camera, which reduces the object motion area (blob size) and may not be detected properly. This issue appears also when a foreground mask is segmented into many pieces, all due to the same object. All of these lead to interrupted object tracks, which was the main reason behind applying the Kalman filtering technique.

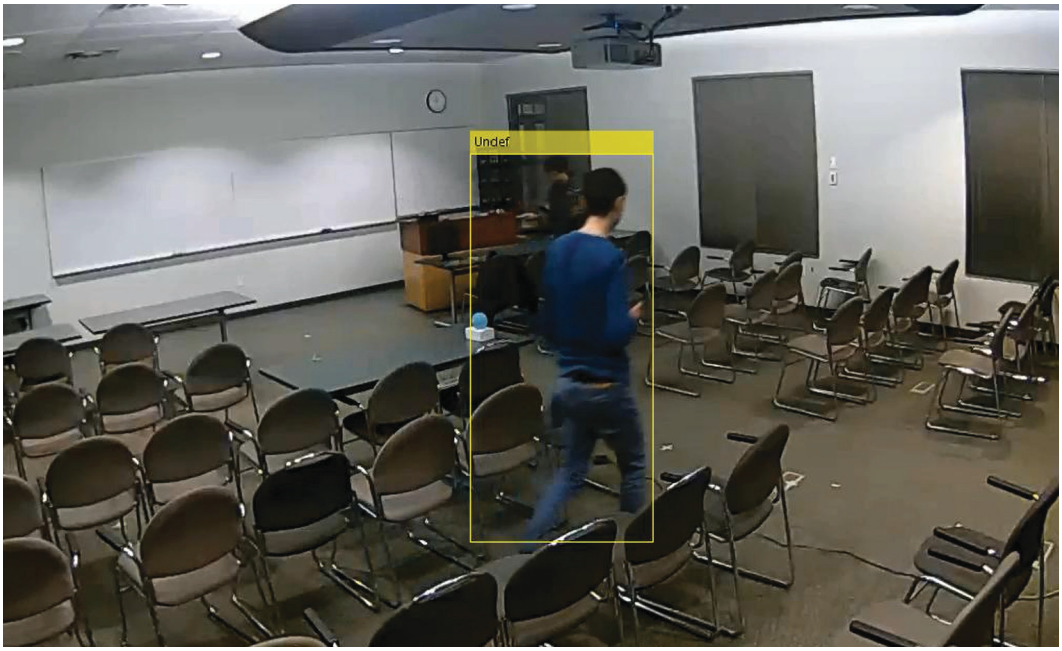

Figure 6.11: Missing object detection caused by partial occlusion.

We ran the algorithm on the Experiment I video to estimate the quality of object detection throughout the sequence. Algorithm in 5.1.1 was able to correctly identify the number of moving objects in the scene, TPs, was only

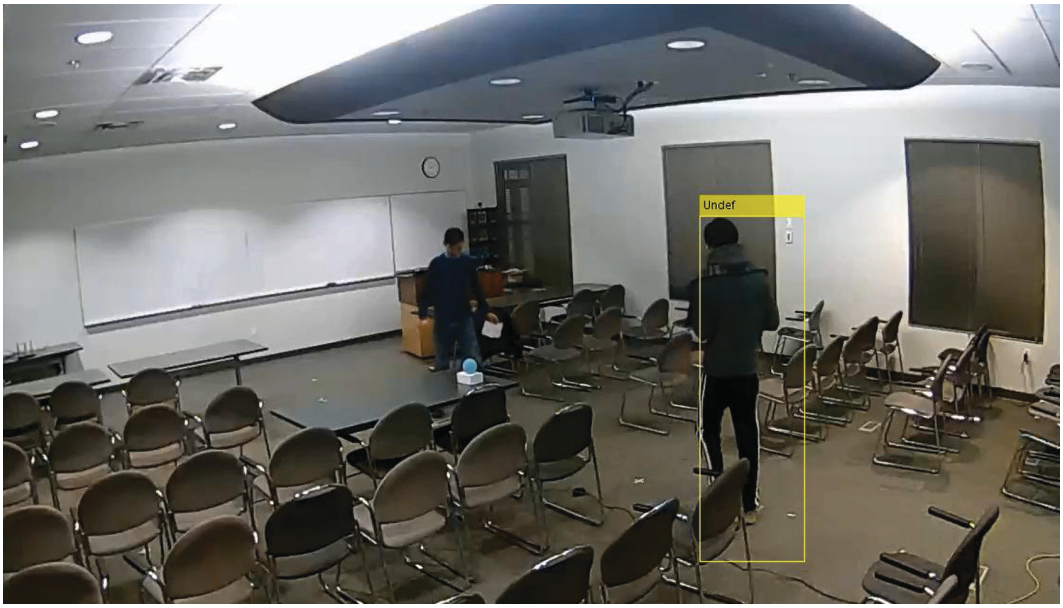

Figure 6.12: Undetected object due to distance from camera.

60% of all frames, yet for motion detection, the TP rate, in almost 85%. While this algorithm is motion-based it still lacks accuracy in terms of object segmentation, which is affected by multiple obstacles and crossing paths of the individuals. Utilizing this methodology can be more useful in large uncluttered environments where individuals have the opportunity to follow a longer path without frequent interruptions in their motions.

6.2.2 Deep Learning models

Similar approach to testing pre-trained state-of-the-art models was used on the experiments where we counted the number of correctly identified moving objects. The algorithm was exposed to a similar type of problems as in the previous subsection, although performed better in terms of detection. This can be seen in the detection of partially occluded individuals, and the approach is less exposed to segmentation errors.

During successful object identification algorithm described in 5.1.2 was able to associate tracks between each other, yet to estimate motion it was essential to set a threshold value for fluctuations in the x, y coordinates of the bounding box top left corner. We evaluate the previous coordinates of the

bounding boxes across set of one second, 23 frames, since the visual domain can be more precise in capturing deviations from the original values. However, it is prone to capturing small motions caused by an individual's limb motion rather than a macro-motion observed when the individual moves to another location.

The algorithm was successful in individual detection with the rate of 87% accross the entire experiment, yet was successful in identifying the number of moving individuals in the scene in only 79% frames with motion.

To sum up, pre-trained models are far more efficient in identifying moving objects in the scene primarily due to a higher precision in object detection.

6.3 Device-to-Identity Association

The results for event association are presented in a similar form to object detection, where a TP is when the MAC address of a mobile device was correctly attributed to an individual holding it and not associated, if otherwise. TNs, on the other hand, are events when the object track was not associated with any label and the individual holding it. FP is observed when a track was assigned a label from another moving device, whereas FN is a track without label, although the tracked individual is holding the device.

Overall performance of the algorithm was 75% for all the individuals combined, whereas it was able to associate multiple devices to one individual correctly less than 30% of time. Algorithm described in 5.4 performs better (on average) when a single object is moving (with or without the device), reaching 83% accuracy. An example of correct identification is shown in Figure 6.13.

During the event alignment process several challenges occurred which degraded performance of the algorithm substantially.

- **Multiple moving devices** is one of the key factors affecting the association performance that appeared throughout the experiment. The issue was observed mostly during periods of experiment when multiple devices were calculated to be moving simultaneously. This caused several tracks to be associated with incorrect labels until the device stopped moving

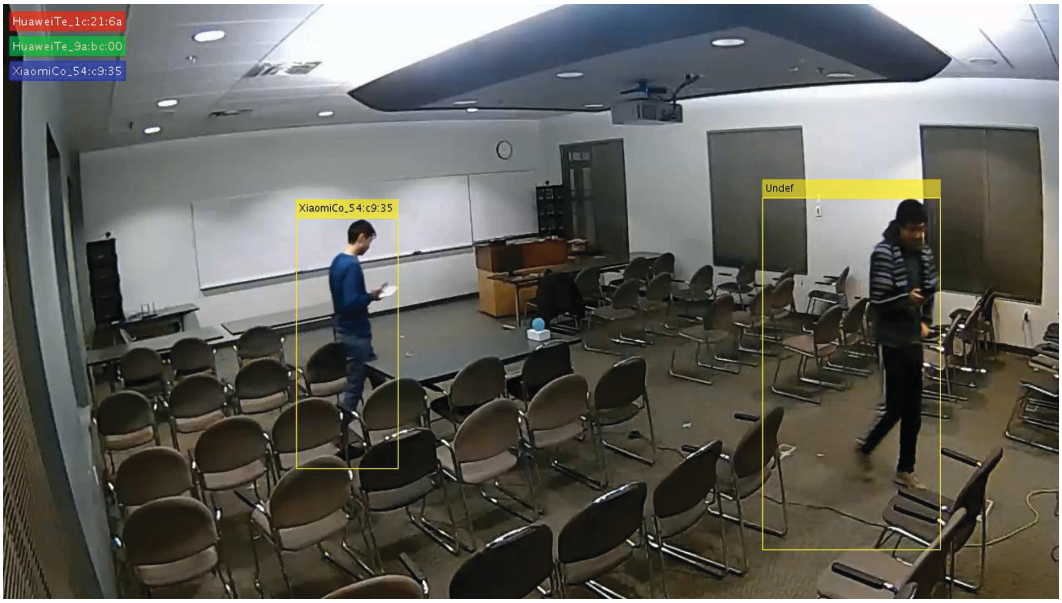

Figure 6.13: Example of device to CV object association.

or the track was lost.

- **Mislabelling of the individuals.** On a few occasions the CV could not detect one of the individuals in the scene, when both individuals were in motion. At the time when one device was held by undetected individual and calculated to be in motion, it was associated to the only identified object in the scene. This lead to incorrect association for both individuals' tracks.

Chapter 7

Conclusion

Throughout this thesis we tried to answer the RQs presented in Section 1.5. While pursuing thorough research for the answers, we observed several interesting challenges that might be of use for future systems of this type. To restate our work, we implemented an object tracking and identification framework based on fusion of different sensor modalities and reported results in Chapter 6. The goal of the framework is to detect and track individuals in the scene and label them with unique addresses, to demonstrate possession of certain devices. The core idea behind this association is the analysis and matching of motion observed under different sensing modalities.

The sensors that were used for this framework produced visual and signal strength readings, which are challenging to fuse. It is difficult to correlate RF signal with what can be captured with cameras, since the signal does not have any visible representation. This is why it was necessary to obtain and utilize a deeper understanding of events and how they relate to the sensor measurements. The event that we believe to be useful for both domains, yet does not require extensive computation and preparation is the motion of individuals and their mobile devices.

Section 1.2 outlined a number of essential system requirements that we followed to build our system. To sum up, we build a system which is inexpensive to set up and utilizes the most common indoor infrastructure present nowadays. The data analysis part did not require significant preparation of prior knowledge of the scene, yet we also examined methodologies which utilize at

least partial knowledge of the experiment output. The set up does not require significant user participation, since the data is generated passively by observing the transmissions from mobile devices and the presence of individuals in a room.

We now summarize the observations made and provide an answer to each of the following RQs:

- **RQ:** Is it feasible to fuse RSSI and video as explained in our framework?

  The results of association of individuals to their mobile devices has shown that it is possible to utilize device and object motion as a common event in both RSSI and Visual domains. Nevertheless, the demonstrated performance was not perfect and may degrade with longer duration experiments.

- **RQ:** Which RSSI-based motion detection methodology is better considering the system requirements?

  Although SVM and Recurrent Neural Networks have shown a higher performance than other methods and can be adapted to multiple device manufacturers, they still require partial knowledge about the experimental set up. We also examine sufficient training set sizes to achieve competitive performance. SVM requires significantly less training data to achieve comparable performance as RNNs. Therefore, we believe that SVM might be a better choice for RSSI-based motion detection given a small amount of data. The ML model outperforms motion detection methods discussed in [29], which is based on initial localization and inferring motion from an individual's displacement. RNN could be a good alternative when training data is abundant. Alternative to these models would be CoV-based motion detection, because it does require only parameter tuning which can be configured prior to usage, unlike Bayesian Inference and the ML models.

- **RQ:** Is the incorporation of multiple APs beneficial for our framework?

Methods that were presented in this thesis and utilize multiple APs have outperformed their counterparts of single AP-based models. We have seen an increase in all the discussed metrics, especially recall and precision, when readings from multiple APs were taken into account.

- **RQ:** Is it possible to generalize RSSI-based motion detection algorithms for different mobile devices?

  Section 6.1.2 has demonstrated that the ML model is capable of learning correlation between RSSI features and motion ground truth. It can also be seen that exposure of the model to more data from various devices improves overall and per-device performance.

7.1 System Feasibility

The main thesis of this work was to examine the feasibility of the framework given limited correlation of the sources. There are several assumptions that make the framework possible, such as connectivity of mobile devices, motion of individuals and sensor placement.

All of the assumptions and their effect is listed below. One of the main assumptions is the connection of mobile devices to a data (in this case, WiFi) network, that is set up on a certain frequency and channel. This may play significant role in motion estimation using RSSI values, since it is the main source of non-visual sensor readings. A device which is not connected to the building network, simply, will not be detected by the system and an individual could be treated as unidentified object. We discuss solution to this as part of a future work later in this chapter.

Another assumption is restriction on individual movement inside a single room and walking within the field of view of all the cameras. A possible scenario in case an individual leaves the field of view is missing object track and creation of a new one, when it reappears. During the data collection experiments described in Chapter 3, we aimed to reconstruct a realistic environment set up with almost no limitation to individual behaviour. We believe that

consistent presence of an individual within the field of view of a camera is a highly possible scenario, since the sensor is, usually, placed within a corner of the room to cover the entire space for reasons of surveilance. However, no experiment took into consideration inter-room movement and an individual entering or leaving the room, subsequently, camera and sniffer loosing line-of-sight with the object. An individual leaving the room might cause confusion in the device motion, since the RSSI values are claimed to deteriorate when there is obstruction in the way. Thus, the RSSI values might fluctuate significantly which can cause issues for all the methods for RSSI evaluation presented in this thesis.

Overall, the system is still heavily dependent on the environment set up and user behaviour. Both factors can affect performance of individual components putting the entire system performance. As an example, changing layout of the room where the measurements take place may increase signal fading which, as reported in Section 6.1.1, obstructs motion estimation. Limited motion paths may lead to short walk patterns made by individuals and less sound RSSI fluctuations and interruptions in motion-based CV algorithms.

Finally, assuming that an individual is in a certain room and moves, at least once, together with a mobile device it is possible to link the device MAC and the visual identity. This scenario raises plausible privacy issues for the individuals. Therefore, users may be unwilling to accept the operation of such a system.

7.2 Future work

The proposed framework yields low accuracy in a number of components, which can be addressed in a number of different ways. The overall performance depends on the accuracy of each individual stream analysis procedure; thus, leaving room for improvement. During our experiments we faced several common issues that were reported for each of the sensing modalities. Thus, improving methodologies or substituting the sensing modality can improve the results.

We believe that there are several directions for future work and are listed below:

- **Data collection.** During our RSSI analysis of devices' motion, we discovered that training the model on a single device's data is highly likely to overfit on a single device's data; thus, performing poorly on other devices. Therefore, a possible improvement to the approach would be collecting RSSI motion patterns accross multiple devices for a particular environment.

- **Association rule.** We acknowledge that the association algorithm described in this work is basic and lacks complexity. It can be significantly improved, if more signal information is extracted from both sensing modalities. Distance-based approaches may be incapable of distinguishing the events using only time difference which can be improved by adding more sophisticated features, as an example, the velocity of objects.

- **User privacy.** It is an important issue for many nowadays, since the devices are ubiquitous and can provide a lot of sensitive information about the owner. In this thesis, we utilize cameras that can be deemed privacy intrusive as they can be used to track people and detect activities and now even the device information makes it possible to link their virtual identity to the real one. Therefore, a significant improvement for this system would be the replacement of cameras with a sensor that reveals less sensitive information, such as a thermal sensor.

- **Multiple channel monitoring.** Throughout the experiments we worked with the assumption that the devices are connected to a specific network on a certain channel. This may be unrealistic with a higher variety of networked devices in an environment. Therefore, one of the directions is to include multiple channel monitoring for transmissions which should allow capturing the majority of traffic from mobile devices. Additionally, non-WiFi communication of the devices can also be observed using

suitable equipment. For example cellular communication uses other frequency bands and channels. However, devices to achieve that, such as Software Defined Radios (SDRs) are not as ubiquitous and are still expensive.

References

[1] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 90–102.

[2] O. Ardakanian, A. Bhattacharya, and D. Culler, "Non-intrusive occupancy monitoring for energy conservation in commercial buildings," *Energy and Buildings*, vol. 179, pp. 311–323, 2018.

[3] T. R. Bennett, N. Gans, and R. Jafari, "Data-driven synchronization for internet-of-things systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, pp. 1–24, 2017.

[4] T. R. Bennett, J. Wu, N. Kehtarnavaz, and R. Jafari, "Inertial measurement unit-based wearable computers for assisted living applications: A signal processing perspective," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 28–35, 2016.

[5] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 941–951.

[6] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[7] L. Čehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1261–1274, 2016.

[8] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4836–4845.

[9] C. K. Chui, G. Chen, *et al.*, *Kalman filtering*. Springer, 2017.

[10] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, 2019.

[11] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 386–403, 2016.

[12] *Device to individual association framework*. [Online]. Available: `https://github.com/sustainable-computing/Device-to-Individual-Association-Framework`.

[13] F. Erden, S. Velipasalar, A. Z. Alkar, and A. E. Cetin, "Sensors in assisted living: A survey of signal and image processing methods," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 36–44, 2016.

[14] Q. Gao, J. Wang, X. Ma, X. Feng, and H. Wang, "CSI-based device-free wireless localization and activity recognition using radio image features," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 346–10 356, 2017.

[15] A. Ghosh, A. Chakraborty, D. Chakraborty, M. Saha, and S. Saha, "Ultrasense: A non-intrusive approach for human activity identification using heterogeneous ultrasonic sensor grid for smart home environment," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–22, 2019.

[16] R. Girshick, "Fast R-CNN," in *International Conference on Computer Vision (ICCV)*, 2015.

[17] A. B. Godbehere, A. Matsukawa, and K. Goldberg, "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation," in *2012 American Control Conference (ACC)*, IEEE, 2012, pp. 4305–4312.

[18] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, "Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges," *Information Fusion*, vol. 35, pp. 68–80, 2017.

[19] Y. Gu, J. Zhan, Y. Ji, J. Li, F. Ren, and S. Gao, "Mosense: An rf-based motion detection system via off-the-shelf WiFi devices," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2326–2341, 2017.

[20] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, IEEE, 2014, pp. 593–598.

[21] J. Huang, S. Chai, N. Yang, and L. Liu, "A novel distance estimation algorithm for Bluetooth devices using RSSI," in *2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017)*, Atlantis Press, 2017.

[22] A. Jain and V. Kanhangad, "Investigating gender recognition in smartphones using accelerometer and gyroscope sensor readings," in *2016 international conference on computational techniques in information and communication technologies (ICCTICT)*, IEEE, 2016, pp. 597–602.

[23] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[24] H. Jiang, J. Li, D. Wang, and H. Lu, "Multi-feature tracking via adaptive weights," *Neurocomputing*, vol. 207, pp. 189–201, 2016.

[25] M. Kipp, L. F. von Hollen, M. C. Hrstka, and F. Zamponi, "Single-person and multi-party 3D visualizations for nonverbal communication analysis.," in *LREC*, 2014, pp. 3393–3397.

[26] S. Lee, Y. Chon, Y. Kim, R. Ha, and H. Cha, "Occupancy prediction algorithms for thermostat control systems using mobile devices," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1332–1340, Sep. 2013.

[27] Y. Lin, J. Shen, S. Cheng, and M. Pantic, "Mobile face tracking: A survey and benchmark," *Proceedings of the Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA*, pp. 18–22, 2018.

[28] Y. Liu, L. Nie, L. Liu, and D. S. Rosenblum, "From action to activity: Sensor-based activity recognition," *Neurocomputing*, vol. 181, pp. 108–115, 2016.

[29] Z. Ma, B. Wu, and S. Poslad, "A WiFi RSSI ranking fingerprint positioning system and its application to indoor activities of daily living recognition," *International Journal of Distributed Sensor Networks*, vol. 15, no. 4, p. 1 550 147 719 837 916, 2019.

[30] M. B. Mendoza, C. A. Bergado, J. L. B. De Castro, and R. G. T. Siasat, "Tracking system for patients with Alzheimer's disease in a nursing home," in *TENCON 2017-2017 IEEE Region 10 Conference*, IEEE, 2017, pp. 2566–2570.

[31] J. Muckell, Y. Young, and M. Leventhal, "A wearable motion tracking system to reduce direct care worker injuries: An exploratory study," in *Proceedings of the International Conference on Digital Health*, ACM, 2017, pp. 202–206.

[32] S. N. Patel, "Infrastructure mediated sensing," PhD thesis, Georgia Institute of Technology, 2008.

[33] V. Popovic, K. Seyid, Ö. Cogal, A. Akin, and Y. Leblebici, "Real-time image registration via optical flow calculation," in *Design and Implementation of Real-Time Multi-Sensor Vision Systems*. Cham: Springer International Publishing, 2017, pp. 199–224.

[34] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision*, Springer, 2016, pp. 17–35.

[35] K. Saho, "Kalman filter for moving object tracking: Performance analysis and filter design," *Kalman Filters-Theory for Advanced Applications*, 2017.

[36] H. Sajid and S.-C. S. Cheung, "Background subtraction for static & moving camera," in *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 4530–4534.

[37] A. Shaabana and R. Zheng, "Cronos: A post-hoc data driven multi-sensor synchronization approach," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 3, pp. 1–20, 2019.

[38] S. Sigg, S. Shi, F. Buesching, Y. Ji, and L. Wolf, "Leveraging RF-channel fluctuation for activity recognition: Active and passive systems, continuous and RSSI-based signal features," in *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, 2013, pp. 43–52.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[40] H. Sloetjes and P. Wittenburg, "Annotation by category-ELAN and ISO DCR," in *6th international Conference on Language Resources and Evaluation (LREC 2008)*, 2008.

[41] A. Srivastava, J. Gummeson, M. Baker, and K.-H. Kim, "Step-by-step detection of personally collocated mobile devices," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, ACM, 2015, pp. 93–98.

[42] Y. Sung, "Rssi-based distance estimation framework using a Kalman filter for sustainable indoor computing environments," *Sustainability*, vol. 8, no. 11, p. 1136, 2016.

[43] *Training & evaluation with the built-in methods : Tensorflow core*. [Online]. Available: `https://www.tensorflow.org/guide/keras/train_and_evaluate`.

[44] H. Tran, A. Mukherji, N. Bulusu, S. Pandey, and X. Zhang, "Improving infrastructure-based indoor positioning systems with device motion detection," in *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*, IEEE, 2019, pp. 176–185.

[45] M. Uddin, W. Khaksar, J. Torresen, *et al.*, "Ambient sensors for elderly care and independent living: A survey," *Sensors*, vol. 18, no. 7, p. 2027, 2018.

[46] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequièvre, "A benchmark dataset for outdoor foreground/background extraction," in *Asian Conference on Computer Vision*, Springer, 2012, pp. 291–300.

[47] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, 2016, pp. 165–178.

[48] M. Wallbaum and S. Diepolder, "A motion detection scheme for wireless LAN stations,"

[49] J. Wang, Q. Gao, M. Pan, and Y. Fang, "Device-free wireless sensing: Challenges, opportunities, and applications," *IEEE Network*, vol. 32, no. 2, pp. 132–137, 2018.

[50] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[51] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, ACM, 2014, pp. 617–628.

[52] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 international joint conference on neural networks (IJCNN)*, IEEE, IEEE, 2017, pp. 1578–1585.

[53] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, "Video object segmentation and tracking: A survey," *arXiv preprint arXiv:1904.09172*, 2019.

[54] Y. Zeng, P. H. Pathak, and P. Mohapatra, "WiWho: WiFi-based person identification in smart spaces," in *Proceedings of the 15th International Conference on Information Processing in Sensor Networks (IPSN)*, Vienna, Austria: IEEE Press, 2016, 4:1–4:12, ISBN: 978-1-5090-0802-5.

[55] M. Zhanbyrtayev, O. Ardakanian, and I. Nikolaidis, "Device mobility detection based on optical flow and multi-receiver consensus," in *2019 IEEE SENSORS*, IEEE, 2019, pp. 1–4.

[56] H. Zou, M. Jin, H. Jiang, L. Xie, and C. Spanos, "WinIPS: WiFi-based non-intrusive IPS for online radio map construction," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2016, pp. 1081–1082.

[57] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, "Deepsense: Device-free human activity recognition via autoencoder long-term recurrent convolutional network," in *2018 IEEE International Conference on Communications (ICC)*, IEEE, 2018, pp. 1–6.

[58] H. Zou, Y. Zhou, J. Yang, and C. J. Spanos, "Unsupervised WiFi-enabled iot device-user association for personalized location-based service," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1238–1245, 2018.