Research on Sound-based Context Recognition Methods for User's Surroundings using Smart Devices

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2023

DINGIRI BANDAGE Thilina Madushan Dissanayake

List of Publications

1. Journal Paper

1. T. Dissanayake, T. Maekawa, D. Amagata, and T. Hara: Detecting Door Events Using a Smartphone via Active Sound Sensing. *Proceedings of ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4), pp.1-26 (December 2018) (Presented at *UbiComp 2019*)

2. T. Dissanayake, T. Maekawa, T. Hara, T. Miyanishi, and M. Kawanabe: IndoLabel: Predicting Indoor Location Class by Discovering Location-specific Sensor Data Motifs, *IEEE Sensors Journal*, 22(6), pp.5372-5385 (August 2021)

2. International Conference Paper

1. T. Dissanayake, T. Maekawa, and T. Hara: Joint Estimation of the Distance and Relative Velocity of Obstacles via Smartphone Active Sound Sensing for Pedestrian Safety. *IEEE International Conference on Pervasive Computing and Communications (PerCom 2023)* (To appear)

2. T. Dissanayake, T. Maekawa, D. Amagata, and T. Hara: Preliminary Investigation of Detecting Events of Indoor Objects with Smartphone Active Sound Sensing, *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp.500-503 (March 2018) (WiP paper)

3. F. Changzeng, T. Dissanayake, K. Hosoda, T. Maekawa, and H. Ishiguro: Similarity of Speech Emotion in Different Languages Revealed by a Neural Network with Attention, *IEEE International Conference on Semantic Computing (ICSC 2020)* (February 2020).

3. Domestic Conference Paper (with peer-review)

1. T. Dissanayake, T. Maekawa, D. Amagata, and T. Hara: Preliminary Investigation on Recognizing Environment-dependent Human Actions using a Fusion of Wi-Fi Based Place Clustering and Motif Detection from Accelerometer Data, Multimedia, Distributed, Cooperative, and Mobile (DICOMO2019) Symposium (July 2019)

4. Domestic Conference Paper

1. T. Dissanayake, T. Maekawa, D. Amagata, and T. Hara: アクティブサウンドセンシングを用いた屋内日常物のイベント検知に関する検討, 第57 回情報処理学会ユビキタスコンピューティング研究会 (2018年2月) (優秀論文賞受賞)

2. T. Dissanayake, T. Maekawa, T.Hara, T. Miyanishi, and M. Kawanabe: スマートウォッチによるアクティブ・パッシブセンシングを用いた屋内位置ラベル推定, 第68 回情報処理学会ユビキタスコンピューティング研究会 (2020年12月)

3. T. Dissanayake, T. Maekawa, and T. Hara: Preliminary investigation of obstacle recognition via smartphone active sound sensing for pedestrian safety, 第71 回情報処理学会ユビキタスコンピューティング研究会 (2021年8月) (学生奨励賞受賞)

4. T. Dissanayake, T. Maekawa, and T. Hara: Preliminary Investigation of Predicting the Distance and the Relative Velocity of the Obstacles via Smartphone Active Sound Sensing for Pedestrian Safety, 第74 回情報処理学会ユビキタスコンピューティング研究会 (2022年6月)

5. T. Dissanayake, T. Maekawa, and T.Hara: Preliminary Investigation of Employing Smartphone Active Sound Sensing to Predict Distance and Relative Velocity of the Roadside Obstacles to Aid Distracted Pedestrians, 第76 回情報処理学会ユビキタスコンピューティング研究会 (2022年11月) (学生奨励賞受賞)

Abstract

Smart devices such as smart watches and smartphones are commonly used in every aspect of the user's daily life. These versatile smart devices are equipped with high quality sensors, that can be employed to sense contextual information of the user. Contextual information about the user's body is generally collected by means of devices placed on the user's body, i.e., wearable devices. Contextual information about the user's surroundings can be collected either by distributing sensors in the user's environment or by remotely sensing the user's surroundings employing a probing device placed on the user's body.

On-body sensors such as accelerometers, provide information regarding the user's activities that can be employed to develop context-aware applications. However, the user's surroundings also provide a rich stream of information regarding the user's context. For example, information regarding the user's environment and the objects in close proximity is abundantly available in the user's surroundings. Capturing such information makes it possible to estimate information regarding attributes of such objects and the user's interactions with them, and characteristic attributes of the user's environment. This information can then be used to recognize the user's context. In this thesis, we focus on capturing contextual information about the user's surroundings. Specifically, we employ the user's smart devices, i.e., smartphones and smartwatches, and sound sensing to capture contextual information from the user's surroundings. In recent studies, sound sensing is widely being adopted in capturing contextual information of a smart device user's surroundings. Furthermore, the characteristics of sound sensing allow us to employ cost-effective, easy-to-maintain methods to sense the user's surroundings with a less number of sensor nodes. Moreover, high-quality microphones and speakers are already included in the off-the-shelf smart devices that further increase the adaptability of such methods.

In this thesis, we first propose a method to capture information regarding the user's interactions with door objects in the surroundings. The user's interactions with door objects in an indoor environment provide vital information that can be employed in applications in the security, infrastructure, and healthcare sectors. We use a single disused smartphone placed in an environment (room) to recognize the user's interactions with

multiple door objects. Our method not only captures the open/close events of the door objects but also captures the opened/closed states of the door objects at any given time which increases the adaptability of our method to applications such as HVAC control. Furthermore, our method also can capture information regarding human movements in the environment which is an essential factor in applications such as surveillance. We evaluated our method using sensor data from five household environments. Our method could recognize door events/states of multiple door objects in an environment with a single probing device with high accuracy.

Next, we propose a method to capture the user's indoor location attributes using sensor data from the user's smartwatch. We employ both the acoustic data as well as acceleration data from the user's arm to capture location-specific sensor data segments (location-specific acoustic characteristics and acceleration data segments) and estimate the user's indoor location class. We propose a novel matrix manipulation method that automatically extracts short segments of location-specific data from time-series sensor data. Furthermore, we describe a method to automatically build a location classifier without requiring handcrafted rules and templates. We collected data from four different household environments and evaluated our method. Our method could estimate seven different location classes without requiring training data collected from the target environment.

Finally, we propose a method that employs sound sensing via a smartphone to estimate the distance and the relative velocity of the obstacles in front of the user, which can be used to aid distracted pedestrians to avoid a collision with an oncoming obstacle. We employ a novel acoustic probing signal that facilitates distance and velocity estimation of both static and mobile roadside obstacles. Furthermore, our novel neural network architecture considers the mobility of the obstacle to efficiently estimate its distance and relative velocity. We evaluated our method using acoustic data from 327 obstacle encounters. Our method achieved an obstacle mobility estimation with a 94% F-measure. The mean absolute error (MAE) of distance and relative velocity estimation was as low as $20.6\,\mathrm{cm}$ and $0.19\,\mathrm{ms}^{-1}$.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Context recognition, which is employing sensor data to capture the user's real-time contextual information, is one of the most studied fields in the area of ubiquitous computing. The user's contextual information can be divided into two sections, based on the proximity where the information is contained, compared to the user; (1) contextual information from the user's body, and (2) contextual information from the user's surroundings. Contextual information from the user's body is collected using wearable devices such as smartwatches [10, 97], smart glasses [56], on-body cameras [61], smart clothes [17], sports trackers [23], and heart-rate meters [114, 19]. The contextual information attributes that these wearable devices collect include the user's activities, location, interactions with objects, and vital signs, which can be employed to develop user context-aware applications. Contextual information contained in the user's surroundings can also supplement the user's context.However, it is difficult to capture surrounding contextual information by distributing sensors in the environment or placing them on objects of interest. In this study, we focus on designing remote sensing methods to capture contextual information from the user's surroundings based on sound sensing.

The user's surroundings can be divided into indoor environments and outdoor environments. The indoor surrounding context of the user is consisted of the user's interactions with indoor objects [78, 130, 131, 132, 136, 125, 81, 112], states of indoor

objects [77], conditions of the indoor environment [40], and user's indoor location attributes [11, 99]. The context of the user's outdoor surroundings includes attributes of the objects in outdoor surroundings [14, 95], and the user's outdoor location attributes [18, 12].

The user's interactions with most indoor objects can be identified as "events". For example, openings and closings of doors in indoor environments are known as door events. These events might change the "state" of the object, which in turn changes the environmental attributes. When considering the user's interactions with indoor objects, door objects take a prominent place as these interactions highly influence the user's environmental attributes. For example, the user's interactions with a window at a "closed" state (open event), might open an enclosed room to the outside world, transiting the state of the window to "opened" state, which may be a security concern. Recognizing the user's interactions with doors in the environment can be incorporated into applications related to (1) home security, such as intruder detection, (2) infrastructure, such as automation of HVAC control, or (3) healthcare, such as surveillance of independently living elderly people. Therefore, door event recognition has been extensively studied in the ubiquitous computing research community [78, 130, 131, 132, 136, 125]. Therefore, **in this study we focus on capturing contextual information related to user's interactions with doors in an indoor environment.**

The user changes his indoor location continuously during his daily life. Therefore, capturing the indoor location attributes of the user is useful to investigate user's lifestyle. One of the most important attributes corresponding to the user's indoor location is the semantic label of the location, i.e., user's indoor location class. There is a strong relationship between the user's activities and his location. For example, a user sleeps in the "bedroom", and cooks in the "kitchen". Therefore, when a semantic label of the user's current indoor location can be estimated, it can be used as prior knowledge to recognize the user's activities. Furthermore, such information can also be used to label lifelog data and for adaptive and automatic management of lifelogging sensors, i.e., switching off a wearable camera when the user visits the restroom. Therefore, **in this study we also focus on capturing contextual information related to user's indoor location class.**

Furthermore, when the user visits outdoor environments, he is always surrounded by a multitude of different objects which may affect him in numerous ways. For example,

the user may have to change his walking speed, and direction or perform a maneuver to avoid a collision with an obstacle in the user's trajectory. Specifically, smartphone users that walk on the sidewalk while viewing their smartphones can get easily distracted by their devices and can bump into various roadside obstacles. This has become a major concern for pedestrian safety. Therefore, capturing attributes of the objects in the user's surroundings is an important task. Specifically, capturing obstacle attributes such as type, distance, and relative velocity regarding oncoming obstacles provides rich information for developing context-aware applications such as pedestrian safety systems, automation of managing alert levels, and assisting visually impaired people with navigation. Therefore, **in this study we also focus on capturing contextual information related to attributes of the objects in the user's outdoor surroundings.**

1.2 Problems

In this section, we will discuss major problems related to the current methods of capturing contextual information from the user's surroundings. As previously mentioned, capturing contextual information from the user's surroundings has numerous real-world applications. To this end, several studies as well as commercially available methods have attempted to achieve this using different sensor modalities and approaches.

1.2.1 Problems Regarding Capturing User's Interactions with Doors

One of the most common methods of capturing the user's interactions with doors in an environment is by connecting sensor nodes (contact sensors) to each door in the environment, which is known as the distributed sensor method. However, this method requires a large number of sensor nodes to be attached to every door object in the environment, which increases the installation cost. Furthermore, each sensor node is equipped with a separate battery, resulting in a high maintenance cost. Even with high installation and maintenance costs, the information related to door events that can be acquired by these sensors is very limited.

For example, there is information regarding door events that are associated with the user's movements. A user might open the door of an enclosed and unoccupied room,

enter it, and close the door behind him. He also might open the door and close the door without entering it. During the first scenario, the room attribute changed from "unoccupied" to "occupied" due to the combination of door-related events, while in the second scenario, the room attribute remain "unoccupied". Such knowledge can be incorporated into applications such as automation of HVAC control where it is important to recognize if a room got occupied before switching on the air conditioning system. Therefore, it is important to capture the user's movements related to door events as well. IR-based (infrared) motion sensors can be employed to capture the user's presence inside a room. However, this further increases installation and maintenance costs and increases the burden on the sensor network. Therefore, a low-cost system that captures events/states of multiple doors in an environment, as well as the user's movements related to them with a minimum number of sensor nodes, is a necessity.

1.2.2 Problems Regarding Estimating User's Indoor Location Class

Estimating a user's indoor location class using sensor data from different onboard sensor modalities in smart devices is a popular topic in the ubiquitous computing research area. The most common onboard sensor modalities that have been exploited in this area include barometers, magnetometers, accelerometers, and microphones. Data from these sensor modalities are used to extract "location-specific" sensor data. Location-specific sensor data can be defined as sensor data segments that only appear when a user is in a particular location class. For example, acceleration data from his hand related to brushing action can only be found when the user is in a bathroom. Similarly, acceleration data related to cutting actions can only be found when the user is in a kitchen. Furthermore, location-specific acoustic characteristics can also be captured using impulse responses of short excitement signals [101]. For example, when the user is sleeping on the bed in the bedroom, distinguishable acoustic characteristics attributed to the materials in abundance in his surroundings, such as pillows, blankets, and mattresses can be observed. Similarly, when the user is in the bathroom, acoustic characteristics attributed to waterproof tiles, porcelain washbasin, and flowing water can be captured. Such location-specific sensor data can be linked to the location class that they have been collected from and this link between the data and the location class can be used as

a "prediction rule" to estimate the location of the user.

However, some of the previous methods that attempt to estimate the indoor location class of the user assume that location-specific sensor data can be continuously and abundantly observed while the user is in the location-of-interest [101]. For example, when the user is in locations such as elevators, features related to air pressure difference due to elevation and magnetic field difference due to electrical components of the elevator, captured by barometers and magnetometers are predominant in such locations. However, considering the user's visits to the kitchen, acceleration data related to cutting action only lasts for a very short amount of time compared to the time he spends in the kitchen. Therefore, previous methods cannot leverage such short sensor data segments to estimate the user's indoor location class. Furthermore, some studies employ classic machine learning approaches with hand-crafted features and prediction rules to estimate the user's indoor location class [29]. Hence, these methods require a specifically tailored set of features and prediction rules for each environment where such a system is deployed. Therefore, these methods have low generalizability across different environments.

1.2.3 Problems Regarding Estimating Attributes of Outdoor Objects

When the user moves in an outdoor environment, objects in his surroundings, i.e., obstacles, may collide with him if he did not concentrate on his trajectory. However, such a level of concentration cannot be always expected when the user is walking outside. Specifically, smartphone users tend to use smartphones for activities such as watching movies, playing games, and using social media while walking. To ensure the safety of such distracted pedestrians, estimating attributes such as the type, distance and relative velocity of surrounding objects is an important task as they are directly related to the risk factor that a particular object imposes and the possible collision time. To this end, some of the studies have been conducted on placing sensor nodes on the obstacles themselves, i.e., approaching vehicles on the road [62], to detect them using a smartphone. However, such methods target a specific type of obstacle which reduces the generalizability of the method. Furthermore, attaching sensor nodes to every obstacle in an

outdoor environment is not a plausible method.

To solve this problem, sensing attributes of the obstacles using proximity sensors such as ultrasonic sensors [124], placed remotely to the obstacle (carried by the user), has been introduced. However, this method requires additional expensive hardware and increases the burden on the user.

1.3 Approach

To solve the aforementioned problems, we recognized the necessity of a proper probing medium. This probing medium must be able to capture contextual information from the user's surroundings, with a sensing range that facilitates our method's adaptability to the previously mentioned applications. Here, we selected sound as our probing medium. Recently, sound has become a popular choice to sense different attributes of the user's surrounding [54, 55, 90, 110, 80, 50] as well as the gestures performed by the user himself [122, 75, 37, 20, 92, 118]. Furthermore, due to characteristics of sound propagation and reflection, sound can be employed to actively probe for contextual information from the user's surroundings remotely. Sound sensing also provides a good sensing range which is sufficient for the aforementioned applications. Furthermore, microphones and speakers, which are the main component for both active and passive sound sensing, are embedded in most smart devices such as smartphones and smartwatches. Therefore, we propose sound-based methods that can be employed to capture contextual information using smart devices.

The ultimate goal of this study is to propose sound-based sensing methodologies to capture the surrounding context of a smart device user. Specifically, our methods do not require placing multiple sensors in the environment of the user to capture contextual information from the surroundings. Furthermore, all our proposed methods can be implemented on off-the-shelf smart devices and do not require additional specialized sensors, which can increase the burden on the end user.

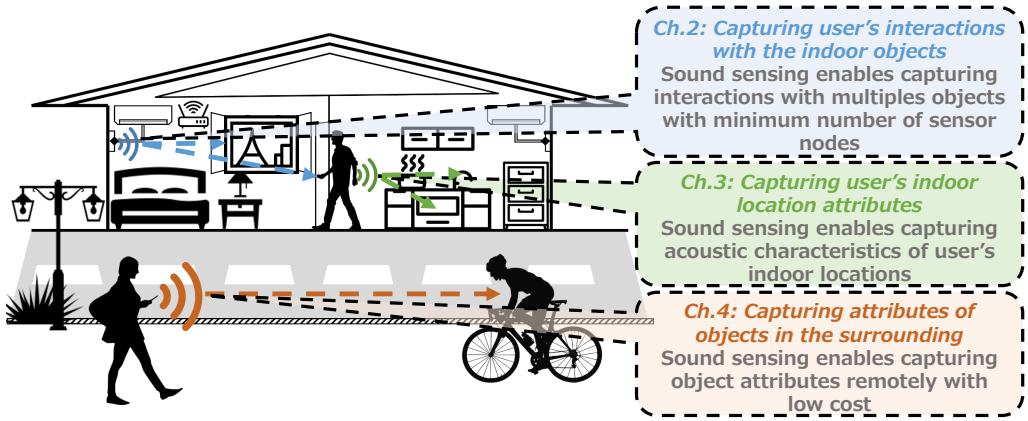


Figure 1.1: An overview of this study

1.4 Research Content

In this research, we address the problems related to capturing contextual information from a user's surroundings using smart devices. Specifically, in this study, we address the issues mentioned in Section 1.2. The overview and the structure of this research are shown in Figure 1.1.

1.4.1 Recognizing Events and States of the Doors

In Chapter 2, we address the problems related to available methods that capture the user's interactions with doors mentioned in Section 1.2.1. Here, we introduce a single acoustic probing device per room that emits inaudible sound waves from its speaker and captures the reflected waves using a microphone. When a door event occurs, the emitted sound waves bounce off the moving door, creating a Doppler shift. Thus, the frequency of the reflected waves is different from that of the emitted waves. We use these frequency differences to detect open/close events of multiple doors using a single probe. Furthermore, Doppler shifts are also caused by the movements of the user in the environment. We leverage these characteristic Doppler shifts to detect if the room of interest is occupied or not. This is a major contribution of our method over the distributed contact sensor-based method that overlooks this attribute. Furthermore, we capture the room's acoustic characters using reflections of short excitement signals emitted from

the probe. These environmental characteristics are attributed to the state of the doors in the surrounding. We use the captured information to recognize the opened/closed states of the doors, which is an essential factor for applications such as HVAC control and surveillance. Our method does not require attaching separate sensor nodes to each door in the room. Instead, our method can recognize events/states of multiple doors using a single device, largely reducing the installation and maintenance cost of our method. By employing a disused smartphone as the acoustic probe, we further reduce the deployment cost of our method.

1.4.2 Predicting Indoor Location Class of User

In Chapter 3, we answer the problems related to predicting a user's indoor location class stated in Section 1.2.2. Here, we employ the sensor data, such as acoustic data from the smartwatch of the user to capture location-specific data related to the user's location class. Specifically, we employ the user's smartwatch to emit short excitement acoustic signals and record the reflected waves. Acoustic characteristics specific to different location classes can be captured using these reflected waves. These acoustic characteristics are attributed to abundantly available materials in each location class. Furthermore, we use a domain adversarial approach to extract environment-independent features from these characteristics and automatically construct a location classifier to estimate the location class of the user. To increase the accuracy of our method, we simultaneously use the acceleration data recorded from the user's hand and captured location-specific acceleration data segments. We use a novel data fusion method to leverage both acoustic and acceleration data to predict the location class of the user with high accuracy.

1.4.3 Estimating Mobility, Distance and, Relative Velocity of Roadside Obstacles

In Chapter 4, we take on the problems described in Section 1.2.3 regarding estimating attributes of the obstacles. Here, we employ the user's smartphone on his hand to remotely and actively probe for obstacles in front of the user. We emit a combination of

high-frequency sound waves from the smartphone and employ the reflected waves to capture information related to the mobility, distance, and relative velocity of oncoming obstacles. Here, we propose a novel probing signal that overcomes the difficulty of probing for both distance and relative velocity attributes of static and mobile obstacles simultaneously. Furthermore, we propose a novel neural network architecture that uses the estimated mobility of the obstacle to effectively select the most appropriate method to estimate its distance and relative velocity.

1.5 Research Contribution

In the broad scope of ubiquitous computing, the concept of smart environment has been thriving and proposes employing computers and other smart devices to ease the burden of the end user in everyday settings and tasks. To make the life of the user more comfortable, smart devices continuously acquire, prepare, process, and analyse sensor data collected from the user's surroundings regarding different aspects of the user's life. Studies have been conducted to propose novel sensing approaches, processing pipelines, and recognition architectures to improve the accuracy, cost effectiveness, plausibility, and generalizability of the existing methods.

Within this context, we propose sound-based approaches to collect contextual information from the user's surroundings. In this study, we focus on employing off-the-shelf smart devices as probes without employing specialized sensors, easing the burden on the end user. Furthermore, our sensing methodologies are designed to minimize the effect on the user's daily life, employing a narrow inaudible frequency band for sound sensing. Our proposed methods can be easily adapted to numerous other applications and settings and significantly contribute to strengthen sound sensing as a powerful tool to sense the surroundings of smart device users. The major contribution of this thesis can be listed as follows:

- We focus on collecting contextual information from both user's indoor and outdoor surroundings. Specifically, we capture contextual information related to user's interactions with doors, user's indoor location class, and attributes of obstacles in user's outdoor surroundings. These aspects can be incorporated into

numerous real-world applications.

- We design novel probing signals and deploy them on off-the-shelf smart devices that can probe for different types of contextual information contained in the user's surroundings. We have included in depth explanation of the different types of probing signals that can be employed to capture (1) motion of different objects relative to the probing device, (2) environmental acoustic characteristics, (3) indoor location class-specific acoustic characteristics, (4) distance between a probing device and an object, and (5) relative velocity of an object with respect to the probing device. To the best of our knowledge, this is the first study that combine multiple fundamental probing signal components to create composite probing signals to sense multiple attributes of surrounding contextual information simultaneously.

- In contrast to the existing methods, our methods employ a minimum number of sensor nodes to probe for multiple surrounding contextual attributes, reducing the installing and the maintenance costs of the probing devices.

- We evaluate our methods using actual data obtained from a semi-naturalistic data collection protocol that mimic real-life scenarios. We conduct additional experiments under different environmental conditions and confirm the robustness of our methods. We further discuss the limitations the methods and make suggestions to overcome them.

1.6 Structure of the Thesis

This thesis is structured as follows:

In Chapter 1, we first discuss the background related to capturing contextual information from smart device user's surroundings. Here, we stress the importance of the contextual information from the user's surroundings and its uses to construct context-aware applications.

In Chapter 2, we introduce a method to recognize open/close events, opened/closed states of doors, and walking events of the user in the environment. In Section 2.1, we introduce the problems regarding the existing door event/state recognition methods and

present the contributions of our proposed method. We further discuss the limitations of the existing methods in Section 2.2 and state the features of our method that attempt to overcome them. In Section 2.3, we conduct an in-depth investigation regarding active sound sensing methods that can be employed to capture information regarding door-related events and states. We explain the proposed method in Section 2.4 and evaluate our proposed method in Section 2.5. We further discuss the performance of our method under different environmental conditions in Section 2.6. In Section 2.7, we conclude our work and discuss future work. This chapter is based on our work published in [25].

In Chapter 3, we introduce a method to estimate the user's indoor location class by leveraging the location-specific sensor data from the user's smartwatch. In Section 3.1, we introduce room-level indoor location classification and methods that are being used in indoor location class prediction. We further discuss the features of our method in contrast to previous works in Section 3.2. In Section 3.3, we present our proposed method for indoor location class prediction, IndoLabel. We evaluate our method in Section 3.4 using data collected in four household environments. We conclude our work and discuss future work in Section 3.5. The study in this chapter is based on our work published in [26].

In Chapter 4, we introduce a method to employ a user's smartphone to perform active sound sensing to estimate the distance and the relative velocity of the oncoming obstacles on the sidewalk. In Section 4.1, we discuss the background of estimating different obstacle attributes using sound sensing. We further discuss the previous attempts at obstacle detection, and distance and relative velocity estimation in Section 4.2. Next, we investigate probing signals that are being used in obstacle attribute prediction and propose a novel probing signal that facilitates obstacle mobility estimation, distance estimation, and relative velocity estimation in Section 4.3. In Section 4.4, we present our proposed method of obstacle distance/relative velocity estimation, named ObsSense. We evaluate our method in Section 4.5 using data collected from real-world roadside obstacles. Finally, we conclude our work and discuss future work in Section 4.5. This chapter is based on our work published in [27].

In Chapter 5, we summarize our work and present the contributions this thesis makes. Furthermore, we further discuss possible directions for our future work.

Chapter 2

Detecting Door Events Using a Smartphone via Active Sound Sensing

In this section, we present our work on recognizing door-related events and states using active sound sensing.

2.1 Introduction

Indoor context recognition has been a major technology behind applications such as context-aware applications and lifelogging. Therefore, indoor context recognition has been the subject of numerous studies in the ubiquitous computing research community. Specifically, contextual information related to movements of indoor objects, such as when a door opens or closes in an indoor environment, can be incorporated into applications such as intruder detection, automation of HVAC control as well as surveillance of independently living elderly people. Therefore, many studies have been conducted employing distributed sensors, i.e., contact sensors, to capture such indoor events. In most studies, sensors such as state-change sensors (e.g., reed switches and magnets), magnetometers, and accelerometers have been employed to achieve indoor event recognition.

However, the major drawback of this approach is that it requires attaching separate sensor nodes to each object in the environment. Therefore, such systems usually have

higher installation and maintenance costs. For example, replacing batteries in sensor nodes and replacing faulty sensor nodes is expensive and time-consuming. This places a huge burden on the user. When the number of sensor nodes placed in an environment gets high, the possibility of sensor nodes becoming faulty also gets high. Due to this, frequent visits from repair personnel are required to replace batteries or sensor nodes. Citing the public database of IoT-enabled houses, Kodeswaran et al. [47] showed that, a house equipped with 14–100 sensor nodes requires a maintenance personnel visit every 18 days. Additionally, attaching sensor nodes to indoor objects affects the aesthetic values of the artifacts [9].

Furthermore, contact sensors attached to a door object will only provide information regarding its events and states, i.e., whether open/close event happens or if the door is in opened/closed state. Therefore, a system based on distributed sensors will not be able to capture human movements related to the door event. As an example, contact sensors connected to a door will not be able to predict if the person who is using the door is entering or exiting the room, making it inapplicable for applications such as HVAC control and surveillance of independently living elderly people that require information regarding both the door event and the human presence in the environment. Even though this problem can be solved by adding an infrared-based motion sensor to the environment, it imposes a further burden on the sensor network by increasing its installation cost while being prone to breakdowns. Furthermore, it also causes problems related to sensing range as indoor environments can be quite complex and the motion sensor will have to deal with a large number of obstacles.

Due to the aforementioned reasons, a method that is reliable and works well with a small number of sensors is necessary to detect events/states of doors in an indoor environment. To this end, Ohara et al. [78] employed Wi-Fi channel state information (CSI) to detect door-based events. However, this method requires to setup a PC with a specially modified Wi-Fi driver and a special Wi-Fi module in the environment of interest which increases the hardware cost. Furthermore, manual extraction of features related to door events from Wi-Fi CSI is a difficult task. Hence, this method relies on deep learning, which requires a large amount of training data, about 100 instances of open/close events per door in the environment. Mahler et al. [71] proposed a smartphone-based home security system based on door event recognition. Smartphone

has become a commodity device in the current world. Therefore, smartphone companies release new smartphone models every year. This results in an increasing number of disused smartphones that are left unattended. These smartphones often contain high-quality onboard sensors which make it possible to employ such devices for indoor event recognition. In the method proposed by Mahler et al. [71], the smartphone is mounted on a door or on the wall near the latching mechanism of a door. When the smartphone is mounted on the door, the sensor data from the magnetometer is employed to recognize open/close events of the door. When the smartphone is mounted on the wall, three-dimensional acceleration data is used to detect the vibrations caused on the wall by the door events. However, this method is difficult to be adapted to an environment with multiple doors due to limited sensing range, as this method requires multiple smartphones to be mounted on/next to each door in the environment.

Due to the aforementioned reasons, we focus on employing disused smartphones as sensor nodes of a door events/states recognition system. The average upgrade cycle of a smartphone is only 32 months in the U.S.[1] Therefore, we can assume that a family of four buys a new smartphone every eight months on average, indicating the plausibility of reusing disused smartphones with high-quality sensors to deploy intelligent context-aware systems. In contrast to Mahler et al's method, we use active sound sensing to probe for door events/states in the environment. This considerably increases the sensing range of the door event/state recognition. Therefore, we propose to install a disused smartphone in a fixed position in a room to recognize the events/states of doors in it.

Doors have fixed states (opened/closed states) and fixed events (open/close events) with which the state transitions occur. In our method, the speaker of the smartphone emits an inaudible high-frequency sound wave with a constant frequency (a sinusoidal wave). The frequency of the sound wave was deliberately selected so that it avoids disturbing users. The inbuilt microphones of the smartphones record the reflected waves. This allows us to capture Doppler shifts created by door-related events. In addition, short sweep signals are periodically emitted from the smartphone. The impulse responses of the reflected waves of the sweeps can be employed to capture acoustic characteristics of the environment, that contain information regarding the states of doors in

[1] https://www.npd.com/wps/portal/npd/us/news/press-releases/2018/the-average-upgrade-cycle-of-a-smartphone-in-the-u-s–is-32-months—according-to-npd-connected-intelligence/

the surrounding [101]. Specifically, the acoustic characteristics of the environment differ according to the open/close states of the doors in the environment. By employing the impulse responses of the reflected sweep waves, we attempt to capture these states of the doors in the surrounding.

To achieve both the event and state recognition of doors, the smartphone emits sine waves and sweeps repeatedly (each 10 sec sine wave will be followed up by a 0.05 sec sweep signal). Using this method we capture information related to Doppler shifts and impulse responses. Furthermore, our method is robust against daily life indoor noises such as voices as we employ a narrow high-frequency band for sound sensing. Moreover, our method is designed to recognize the walking events of a person related to door events using Doppler shifts as this information is necessary for applications such as HVAC control.

We employ the following information to distinguish between events in different door objects:

- The frequency and amplitude of the reflected sound change over time and these reflections contain information regarding the Doppler shifts which differs according to the relative locations of the door objects with respect to the smartphone. (We comprehensively explain the theory of operation in Section 2.3.)

- The directionality of the sound emitted by a speaker changes with the frequency of the sound [94]. Therefore, by employing a composite sine wave composed of two different sine waves with different frequencies (high and low-frequency components), we capture information related to the relative location of the door object with respect to the smartphone.

- Commonly, smartphones are employed with multiple microphones. Here, we employ two inbuilt microphones to capture information related to the direction of the door objects.

We employ the aforementioned information to recognize the events/states of the door objects. As mentioned above, the frequency offset of Doppler shifts changes over time. Hence, we employ hidden Markov models (HMMs) [123] to model events/states of door objects, that are widely used to model time-series data. Specifically, we employ

left-to-right HMM to model events/states of each door. Furthermore, we know that a door at the "closed" state can only expect an "open" event, which transits the state of the door into the "opened" state. Similarly, a door at the "opened" state can only expect a "close" event which transits the door state to the "closed" state. Employing the above prior knowledge regarding door events/states, we establish a grammar that decodes the inputs by the HMMs using the Viterbi algorithm [88].

However, in the presence of multiple doors in an environment, audio signals recorded related to door events/states of different doors can possibly be mixed together. Therefore, there is a possibility of the recognition accuracy of door event/state recognition deteriorates when the feature vector sequences extracted from the mixed signals are directly fed into a recognition model that is designed to recognize events/states of each door object separately (a set of HMMs). Hence, we separate the input signals that contain information regarding events/states of multiple doors into the information of events/states of each door before they are fed into HMMs tailored for each door object. To separate the audio signal captured by microphones into feature vectors related to events/states of each door object, we employ a discriminative classifier. Here, the discriminative classifier is designed to transform the time-series audio signals captured by the microphone into time-series probabilities of the occurrences of each event/state. Therefore, by feeding these time-series probabilities of occurrences of events/states of each door object into HMMs designed for each door, we can recognize the events and states of these doors. Furthermore, by incorporating the generative model, i.e., HMMs, we attempt to capture the temporal regularity of events as well as to eliminate sporadic noises contained in the recorded acoustic signals.

The contribution of this study can be listed as follows.

- We employ active acoustic sensing to propose a novel approach to recognize the states/events of indoor objects. In addition, we propose a two-tier recognition model that recognizes events/states of multiple doors in an environment. This is the first study that uses a smartphone to recognize events/states of doors using active sound sensing.

- We employ the following information to recognize events/states of multiple doors with high precision: (1) the time-series analysis of Doppler shifts created by

open/close events of door objects, (2) the reflections of a composite sine wave composed of two sine waves with different frequencies. (3) the acoustic signals captured by two inbuilt microphones of the smartphone, and (4) the acoustic characteristics of the environment, which are attributed to the states of the doors in the surrounding, are captured using impulse responses.

- Our method can recognize walking events related to door events which are necessary information for the automation of HVAC control and surveillance.

- We collected data from several different environments under different conditions and confirmed the effectiveness and validity of our method.

The remainder of this study is structured as follows: First, we introduce previous studies that are related to indoor context recognition. Next, we conduct a preliminary investigation regarding active sound sensing using a smartphone. We then design a method to recognize events/states of doors using active sound sensing. Finally, we evaluate our method using sensor data collected from real environments.

2.2 Related Work

In this section, we introduce studies related to indoor context recognition.

2.2.1 Monitoring Indoor Events Using Distributed Sensors

A large number of distributed sensors such as accelerometers, RFID tags, switch sensors, and vibration sensors can be employed to detect events of indoor objects [106, 86, 115, 69, 70]. Specifically, these sensors are attached to each object of interest in the environment, and their events are detected. Even though high accuracy and fine-grained sensor data readings are guaranteed in the distributed sensor-based methods, the maintenance (e.g., battery and faulty sensor node replacement) costs are usually high. Several studies have attempted to solve this problem by employing energy-harvesting sensors to monitor buildings. Campbell et al. [15] proposed a method that employs a piezo-film to develop a vibration detector. This vibration detector transmits a data packet to report

Table 2.1: Features of door event recognition methods

	Distributed	**Barometer**	**Camera**	**RF**	**Sound**
installation cost	high	-	-	-	reuse
maintenance cost	high	low	low	low	low
coverage	narrow	wide	wide	wide	wide
privacy issue	no	no	yes	no	yes
dark	work	work	not work	work	work
train	no	yes	yes	yes	yes
power	battery	AC/battery	AC	AC	AC
simultaneous	yes	no	yes	no	no
person	no	no	yes	yes	yes
multiple persons	no	no	yes	no	no

the vibration event and simultaneously generates an electrical current using the vibrations. This sensor can be used to detect open/close events of a door, cabinet, window, or refrigerator door.

Table 2.1 summarizes the features of the approaches based on the distributed door sensors. The price (installation costs) of current commercially available devices is high (approximately 100 USD per device). Furthermore, current systems mainly focus on the main entrance doors as detecting their events can be used for intruder detection. However, these systems are not capable of detecting the movements of a person related to door events. Therefore, such systems are not applicable to other applications such as HVAC control and surveillance. Furthermore, as mentioned in the previous sections, installation costs and maintenance costs are high when the number of sensor nodes in the system is high.

2.2.2 Monitoring Indoor Events Using a Small Number of Sensors

Barometric pressure sensors have also been used in previous studies to detect open/close events of the doors. Patel et al. [81] proposed a method to detect pressure variations caused by open/close events of doors and room-to-room transitions using pressure sen-

sor units attached to HVAC air filters. Wu et al. [125] proposed a method to use the barometer in a smartphone to detect a sharp change in indoor pressure when a door event occurs with an HVAC system. However, these methods only work for environments with HVAC systems.

Shi et al. [98] proposed a method to employ FM-radio signal receivers to recognize indoor situations such as “empty room”, “opened room”, and “walking person”, based on the fact that changes in the environment affect the propagation of radio waves. Wi-Fi channel state information (CSI) can also be used to detect events and states of indoor objects. Ohara et al. [78] proposed a method to detect indoor events using a commodity Wi-Fi access point and a computer with a special Wi-Fi module, placed inside a room to detect the changes in Wi-Fi signal propagation during indoor events. In contrast, we use disused smartphones to recognize indoor events via active sound sensing.

Table 2.1 summarize the features of our approach (sound column) with other approaches that employ a small number of sensor nodes. Note that the approach based on supervised learning requires training data to recognize events of multiple doors. We implement a smartphone application that supports efficient collection and labeling of training data. For example, the application provides clear instructions to the user to use different doors in the environment, e.g., reading out “Please open door number one”, and the user performs the relevant actions. However, to implement our system in the real-world, our system should work with a minimum amount of training data.

Door events recognition approaches based on the barometer, camera, RF, and sound require a continuous AC power supply. To reduce the costs related to the installation and maintenance of the power supply, we can supply electricity via an easily accessible ceiling power outlet or a bulb socket in any room [63, 66].

However, the barometer-based approach is unable to capture the user’s movements related to the door events. In contrast, the RF- and sound-based methods can be used to detect a person in the environment, they can be applied to a variety of applications such as HVAC control and monitoring independently living elderly people, even though these methods are unable to distinguish between multiple people. However, these approaches cannot detect door events when a person inside the room moves simultaneously.

While the camera-based methods can detect multiple persons in an environment, this approach has several critical issues such as privacy problems, and low accuracy in

dark environments. Although the barometer-, sound-, and RF-based approaches cannot detect door events that occur at the same time, the probability with which multiple doors are opened/closed at the same time is considerably low. While the sound-based approach also suffers from privacy issues, our method works using only sound features of the inaudible high-frequency band. In order to further reduce the problems regarding privacy, we suggest extracting the relevant features from the audio data before sending the data to a server where the recognition process runs.

As detailed above, the RF- and sound-based approaches do not have critical drawbacks and can be applied to a variety of applications. However, as mentioned in the introduction, the RF-based approach requires a specially modified Wi-Fi driver-installed PC with a special Wi-Fi module.

2.2.3 Context Recognition Using Active Sound Sensing

Many researchers make use of active sound sensing for context recognition. Gupta et al. [37] propose a method of recognizing hand gestures using the Doppler shift where they employ a tone in the range of 18 kHz as a pilot tone. Fu et al. [33] propose a method of using a 20 kHz sound wave to track exercises using the Doppler shift caused by the movements of the body.

Several mobile computing studies have employed sound beaconing to estimate relative positions to other devices [137, 22]. Active sound sensing has also been used to locate a smartphone user. Rossi et al. [90] propose a method to locate a smartphone based on active sound fingerprinting. The authors measure impulse response at each indoor reference point to train a classifier that estimates a user's current coordinates using observed impulse response. Tung et al. [110] also make use of active sound probing for indoor location tagging. Tachikawa et al. [101] employ active sound sensing as well as passive sound sensing by smartphones to estimate location semantics such as toilet and restaurant.

Similar to the above approaches, we also employ a sound wave with a constant frequency to detect the events of indoor objects. Note that our method has several features, which are mentioned in the introduction section including making use of composite signals and a grammar that defines state transitions of an object, to achieve accurate door

event/state recognition.

2.2.4 Context Recognition Using Passive Sound Sensing

Passive sound sensing has also been used for context recognition. Tarzia et al. [108] used passive sound fingerprinting to locate a smartphone user by extracting sound fingerprints based on the acoustic background spectrum of rooms. Maekawa et al. [68, 64] employed a wrist-worn device containing a camera and microphone to recognize object-based activities. Clarkson et al. [21] used a camera and a microphone attached to a chest strap to detect location-related events such as entering an office, kitchen, or courtyard. Korpela et al. [48] employed a smartphone microphone to evaluate toothbrushing activities by analyzing sound events of toothbrushing.

2.3 Investigation of Active Sound Sensing

We investigated the characteristics of signals obtained via active sound sensing, and based on this investigation, we designed an indoor event recognition method in the next section.

2.3.1 Theory of the Operation of Event Detection Using Doppler Shift

To detect the open/close events of indoor objects, such as doors, we employed a well-known and well-understood phenomenon known as the "Doppler effect" or the "Doppler shift." This effect is defined as the shift of the observed frequency of a wave when the observer is moving relative to the wave source. In this research, the smartphone becomes both the observer and the wave source and the Doppler shift is caused by the moving doors in the environment. Imagine a scenario wherein a smartphone installed inside a room emits a wave with a constant frequency while recording the reflected wave at the same time. When a door event occurs in the environment, the resulting reflected wave then has a different frequency than the original frequency that the smartphone emitted. This frequency shift can be observed as a distortion in the FFT spectrum of

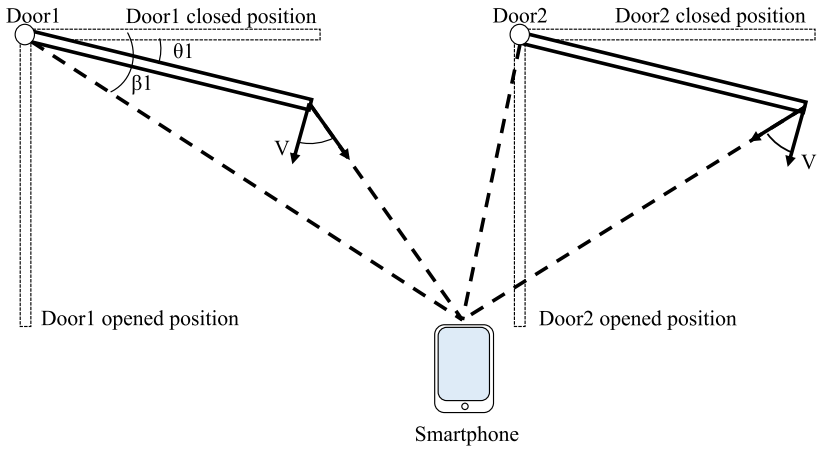
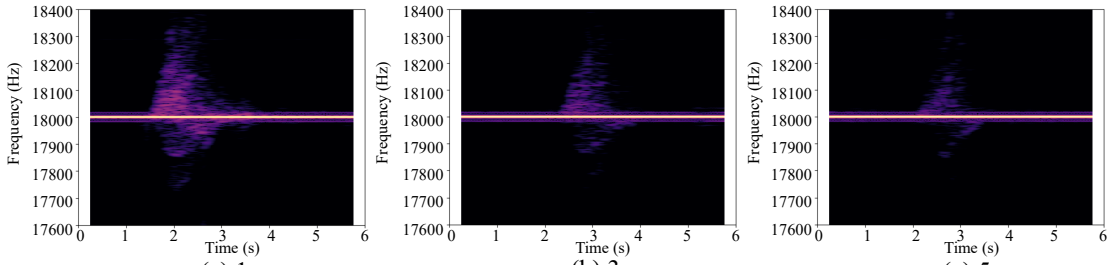


Figure 2.1: Relation between the relative location of a moving door and its velocity component towards the smartphone

the recorded sound. This distorted spectrum can then be analyzed to detect the door event and differentiate between the door events of multiple doors. In other words, when the door rotates around its hinge and moves toward the smartphone, it causes an increment in the recorded frequency, creating a positive shift in the FFT spectrogram. Similarly, when the door is moving away from the smartphone, it causes a decrement in the recorded frequency, creating a negative shift in the FFT spectrogram. By utilizing this characteristic of the frequencies, we can distinguish between the open/close events of the door.

Next, we will look at how can we differentiate between the door events of multiple doors. Figure 2.1 shows a situation wherein a smartphone is placed in a room with two doors. The open event of Door1 shows that there is a gradually diminishing positive velocity component from the door toward the smartphone since the door starts moving from the closed position until the angle between the door frame and the door ($\theta 1$) becomes larger than that between the door frame and a line connecting the smartphone and the hinge ($\beta 1$), resulting in an increment in the observed frequency. From then, until the door reaches its opened position, the velocity component from the door toward the smartphone remains negative, resulting in a decrement in the observed frequency. When Door2 is opened, the velocity component from the door toward the smartphone



(a) 1 m (b) 3 m (c) 5 m

Figure 2.2: FFT power spectrograms for a door-opening event when the smartphone is placed (a) 1 m from the door, (b) 3 m from the door, and (c) 5 m from the door

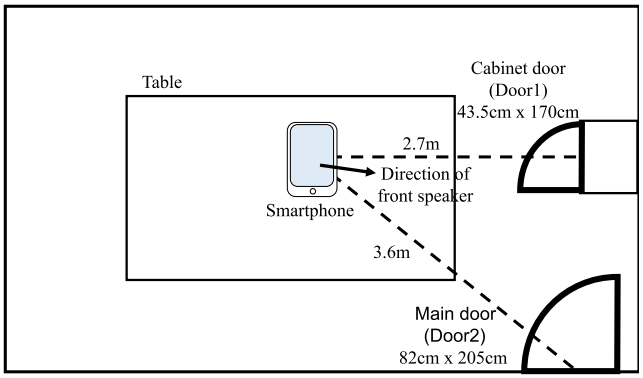


Figure 2.3: Environment of the preliminary experiment (1). The smartphone emits a sinusoidal sound wave with a frequency of 20 kHz. The directions of the front speaker and microphone are identical.

remains positive all the way from the closed position to the opened position. This indicates that only an increment in the observed frequency can be expected. By utilizing these characteristic differences in the frequency shifts, we can distinguish between the door events of Door1 and Door2.

2.3.2 Impact of Distance between Smartphone and Object

The open/close events of doors can be detected utilizing the Doppler shift, as shown above. We first investigated the effect of the distance between the smartphone and a door

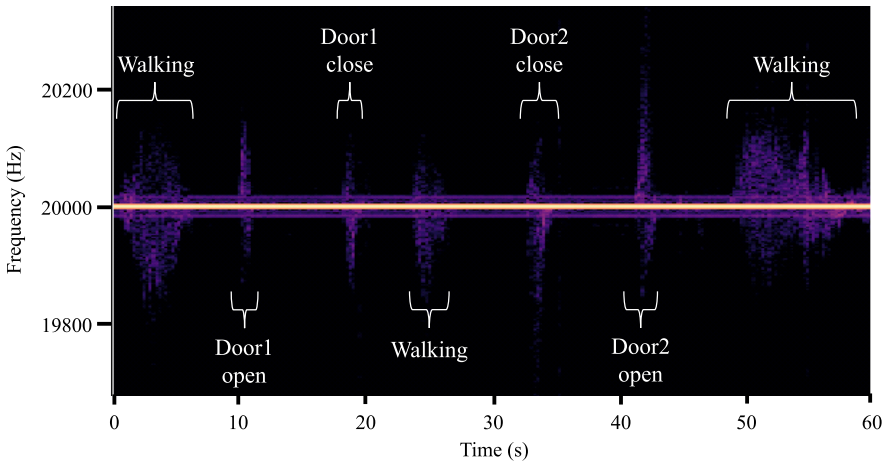


Figure 2.4: FFT spectrogram of the front channel when the smartphone is placed 2.7 m from Door1 and 3.6 m from the Door2, as shown in Figure 2.3

using the proposed method. We recorded acoustic signals emitted from the smartphone placed at various distances from the door. We used the Google Nexus 6P smartphone for active sound sensing. The smartphone emitted a sinusoidal sound wave with a frequency of 18 kHz from its speaker. The two inbuilt microphones (one in the front and the other at the back of the smartphone) recorded stereo sounds at a sampling rate of 44.1 kHz. Figure 2 shows the visualized FFT power spectra of the recorded sound signals that identify an open event when the phone was placed 1, 3, and 5 m from the door. Note that the front microphone and the speaker of the smartphone were positioned in the direction of the door. From these spectrograms, we can confirm that the frequency shift mostly occurred in the positive direction during the open event of the door. In addition, the Doppler shift triggered by the door event could be observed by the smartphone even if the door was 5 m from it.

2.3.3 Time Variance of the Doppler Shift

Figure 2.3 shows the floor plan of our experimental environment, and Figure 2.4 shows an FFT power spectrum obtained when the open/close events of Door1 and Door2 occurred in the environment. During an open event for Door2, the door first moved toward

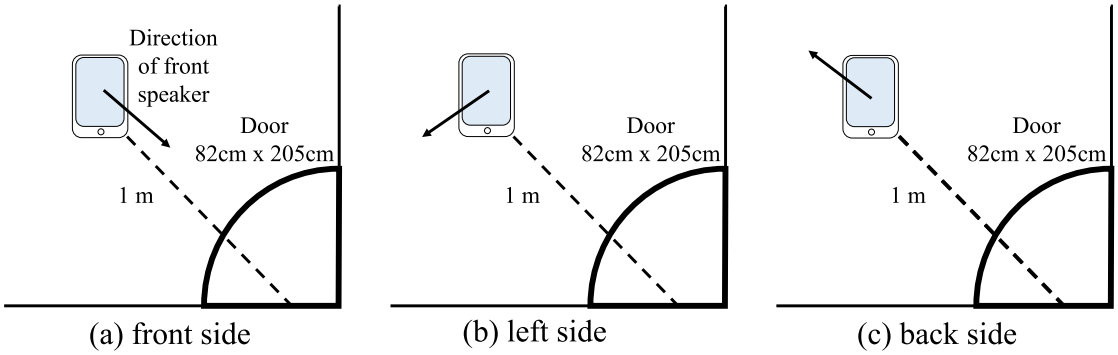


Figure 2.5: Situations wherein a door is located (a) in front of the smartphone, (b) to the left-hand side of the smartphone, and (c) behind the smartphone. The smartphone emits a composite wave of two sine waves with frequencies of 18 and 20 kHz, respectively.

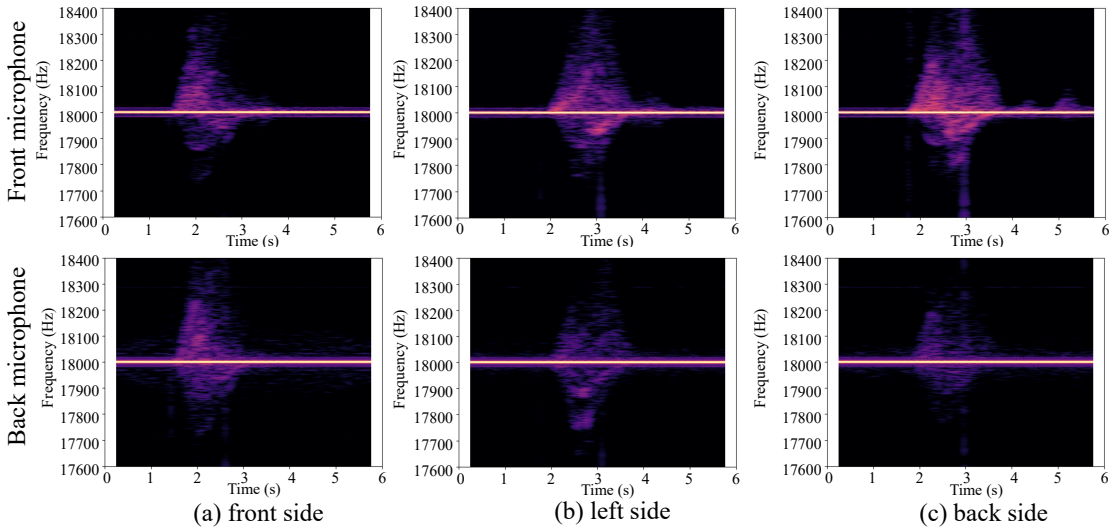


Figure 2.6: Spectrograms of the door-opening events when the door is located (a) in front of the smartphone, (b) to the left-hand side of the smartphone, and (c) behind the smartphone. Spectrograms of the audio data from the front microphone are in the top row and the back microphone are in the bottom row.

the smartphone and then moved away from the smartphone. In this case, the time taken by the door to move toward the smartphone was relatively longer than the time taken by

the door to move away from it. Therefore, we observed that the frequency shift caused by an open event first resulted in a strong shift toward the positive direction (higher than 20 kHz) and then resulted in a relatively weak shift toward the negative direction (lower than 20 kHz) of the FFT power spectrum. Conversely, during a close event of Door2, it moved toward the smartphone for a short time after it moved away from the smartphone for a longer time. This created a frequency shift in a direction opposite to that of an open event, creating a strong shift toward the negative side after creating a weak shift toward the positive side of the FFT power spectrum. In contrast, during the open/close event of Door1, we observed a short-duration frequency shift because Door1 (cabinet door) was smaller than Door2. Additionally, during an open event of Door1, we observed a frequency shift that mainly comprised a shift toward the positive side. As above, because the observed frequency shift depends on the doors, it is necessary to capture the temporal patterns of the Doppler shift to recognize the open/close events.

Moreover, because the speed of walking is similar to that of the door event, the frequency shift caused by the walking events is also similar to that caused by the door event. However, since the duration of the walking event is usually longer than that of the door event, we believe that utilizing the temporal patterns of the frequency shift also enables us to distinguish between the walking and door events.

2.3.4 Stereo Recording Using the Two Inbuilt Microphones

Figure 2.5 shows the three scenarios in which we collected sound data by varying the relative location of the door with respect to the smartphone. Figure 2.6 shows the FFT power spectrograms of an open event of the door for each situation. As shown in Figure 2.6, the sound features obtained from the front and rear microphones of the smartphone were different. Even when the microphone did not face the door, the sounds of the microphone appeared to include a strong shift. This might have been caused by the reflected sounds from the surrounding walls.

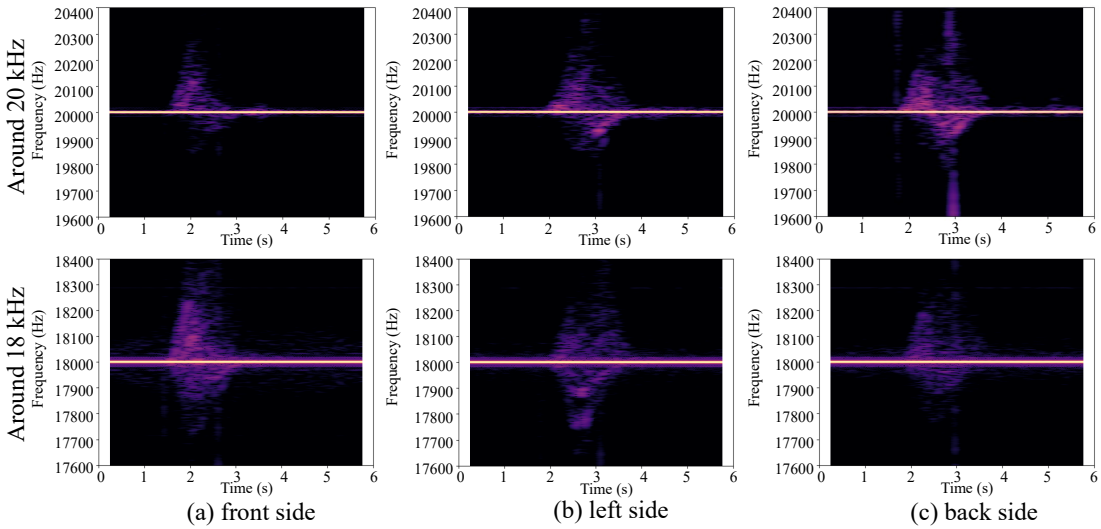


Figure 2.7: Spectrograms of the audio data obtained by the front microphone when the door is located (a) in front of the smartphone, (b) to the left-hand side of the smartphone, and (c) behind the smartphone. The spectrograms in the top row show a frequency range of around 18 kHz, and the spectrograms in the bottom row show a frequency range of around 20 kHz.

2.3.5 Composite Signals Consisting of Two Sinusoidal Waves with Different Frequencies

The characteristics of the sound waves emitted from the speaker differed according to the frequency of the wave. For example, a lower frequency wave tended to have a higher amplitude than that of a higher frequency wave depending on the hardware properties of the speaker. In addition, sound waves with higher frequencies appeared to be more directional in comparison with sound waves with lower frequencies [94]. To obtain information from sound waves with both high and low frequencies, our method utilizes a composite wave. Figure 2.7 shows the spectrograms of the audio data obtained from the front microphone when a composite wave comprising 18- and 20-kHz sinusoidal waves was emitted when the smartphone was placed in three different configurations, as shown in Figure 2.5. As shown in the spectrograms of Figure 2.7, we could confirm that information obtained from the 20-kHz wave and that of the 18-kHz wave are different,

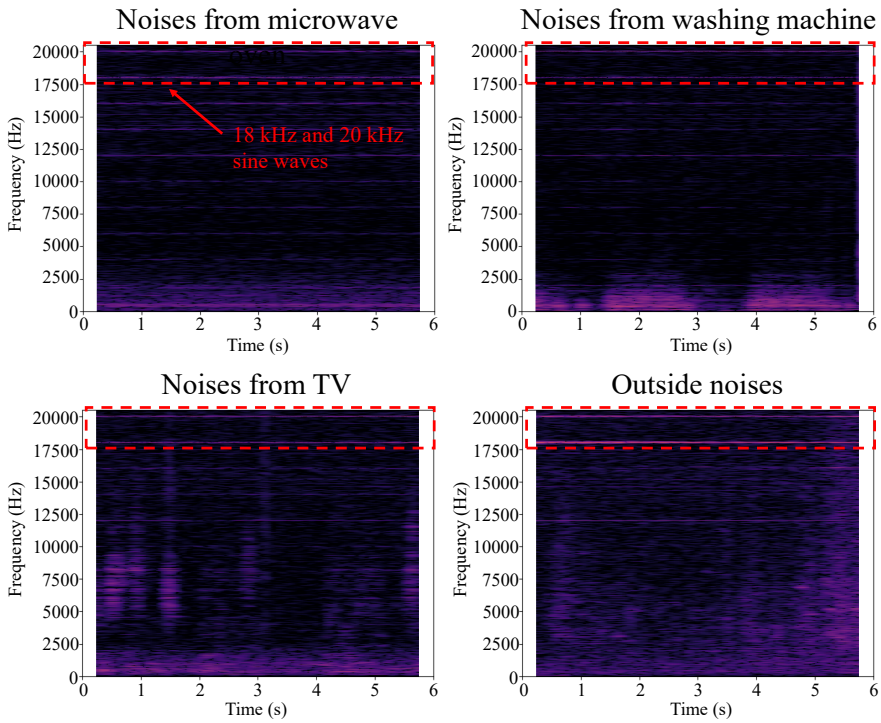


Figure 2.8: Environmental noises recorded by a smartphone when a composite wave comprising 18- and 20-kHz sinusoidal waves was emitted. The distance between the smartphone and the microwave oven was approximately 3 m. The distance between the smartphone and the washing machine was approximately 3 m. The distance between the smartphone and the television was approximately 3 m. The distance between the smartphone and the entrance door was approximately 1 m.

indicating that utilizing the composite wave enables us to obtain information related to the direction of a door of interest.

2.3.6 Daily Life Noises

Because our method focuses only on a high-frequency band, i.e., around 18- and 20-kHz, our method is robust against daily life noises. Figure 2.8 shows spectrograms of audio data containing daily life noises. As shown in the figure, the frequencies of noises caused by a microwave oven and washing machine are low. However, the television

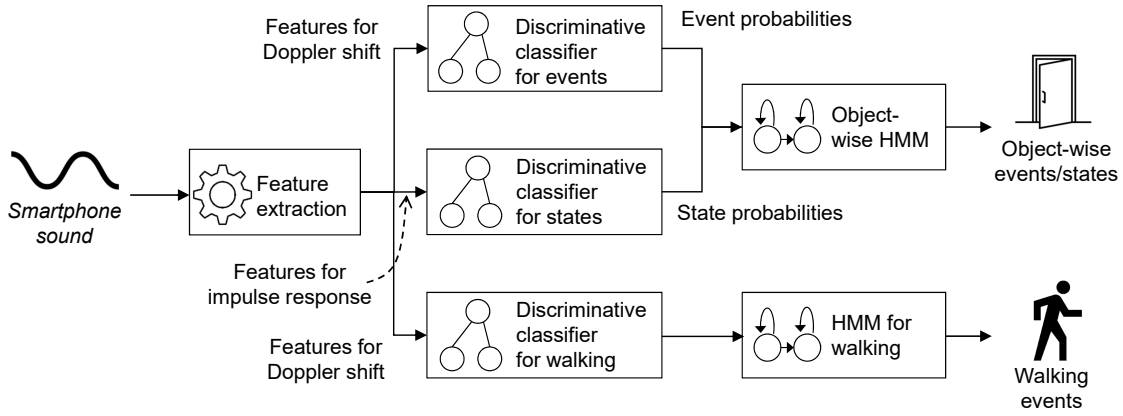


Figure 2.9: Overview of the proposed recognition method

seems to suddenly emit high-frequency noises, which can deteriorate the performance of our method. While it is meaningless to emit inaudible sounds from televisions, we confirmed that the high-frequency noises are sometimes included in television sounds. In addition, when an entrance door is opened, outside noises are observed as shown in Figure 2.8, which can also deteriorate the recognition performance. To cope with these sporadic noises, our method is designed to contain generative models (HMMs), which are robust against such noises. Furthermore, we evaluated the performance of our method in such situations in Section 2.6.1.

2.4 Method

2.4.1 Overview

We assume that each door object has two events and two states, i.e., “open” and “close” events and “opened” and “closed” states. In addition, we regard that there are two events/states regarding a person in an environment of interest; “walking” and “not-walking.” We recognize events/states of each object in the environment of interest as well as the walking events for each time slice. In the proposed method, a smartphone emits a continuous composite sinusoidal wave and also emits occasional sine sweeps. We then use acoustic signals recorded from the front and back inbuilt microphones

to recognize the events/states of the objects in the environment of interest. Figure 2.9 shows an overview of our recognition method. After extracting features, a feature vector sequence related to the Doppler shift is fed into a discriminative classifier that recognizes the events of each object in the environment. In addition, a feature vector sequence related to the impulse response is fed into a discriminative classifier that recognizes the states of each object. After that, the class probabilities from the discriminative classifiers are concatenated and fed into HMMs tailored to each object. As for the walking event recognition, the feature vector sequence related to the Doppler shift is fed into a discriminative classifier for walking event detection. The class probabilities from the discriminative classifier are then fed into HMMs tailored for walking event detection.

As above, we apply the hybrid discriminative and generative approach for event recognition. While discriminative classifiers are reported to outperform generative models, we attempt to deal with sporadic noises with the generative models, i.e., HMMs, which can capture the temporal regularity of the door/walking events in order to filter out the sporadic noises.

2.4.2 Sound Emission

Sine waves (with long durations) and sine sweeps (with short durations) were repeatedly emitted from the front speaker of the smartphone. A sine wave was emitted to obtain information about the Doppler shifts created by the moving objects. This can be utilized to detect door and walking events. A sine sweep was emitted to obtain acoustic information about the environment, which can be utilized to recognize the states of the objects.

The sine wave used in our method was a composite wave comprising two sine waves with frequencies of 18 and 20 kHz. These frequencies are not audible to the human ear. To generate the sine sweep, an excitation signal was emitted by the smartphone sweeping the frequencies from 18 to 20 kHz for 0.05 s. To avoid missing any events of the objects, the sine wave was continuously emitted for 10 s before emitting a sweep for 0.05 s. This process was repeated, and the timestamps of the emitted sweeps were saved in the smartphone.

2.4.3 Feature Extraction

We can extract the features related to the Doppler shift and impulse response from the acoustic signals obtained by both microphones.

Features Related to Doppler Shift

We extracted the features for event detection using the FFT power components around the 18- and 20-kHz frequencies. In the proposed method, we set up a 0.5-s sliding window with 96% overlap and applied a Blackman function to the window to calculate the FFT power spectrum of the audio data. We then extracted 4000 frequency bins from either side of the bandwidths of the 18- and 20-kHz sine waves. This produced a 16000-dimensional frequency vector sequence.

Next, we reduced the dimensionality of the vectors using time frame-wise averaging. Here, we used a 100-dimensional 50% overlapping sliding window along the frequency axis of each data frame, taking the average of the FFT power components within the window as one dimension. With this procedure, we can reduce the dimensionality of the vectors from 16000 to 320 dimensions while still retaining most of the characteristics of the original feature vectors.

Features Related to Impulse Response

We obtained the acoustic characteristics of the environment by analyzing the impulse responses of the emitted sine sweeps, as mentioned above. First, we calculated the FFT power spectrum of the audio data by applying a Blackman function to a 0.01-s sliding window with 85% overlap. This window size is considerably small and can be justified by the fact that the sweep length is 0.05 s.We used the FFT power spectrum components corresponding to the frequency range of the sweep as features for recognizing the states of the objects.

Cowling et al. [24] revealed that the Mel-frequency cepstral coefficient (MFCC)-based feature extraction technique is one of the most appropriate approaches for environmental sound recognition systems, which has a recognition accuracy of 70%. Chen et al. [16] used MFCC features for recognizing bathroom activities, including showering, flushing, and urinating with high accuracy. Therefore, we decided to calculate

12-order MFCC features from the extracted FFT power spectrum components, which are also used as features.

2.4.4 Discriminative Classifier for Events

For each time slice, a feature vector consisting of features related to the Doppler shift is fed into a discriminative classifier for events. The discriminative classifier is a random forest [13] that is used to compute class probabilities of open/close event classes of each object in the environment of interest and a class belonging to neither of the above classes. Therefore, the random forest is a $(2N + 1)$-class classifier (this includes a class named "other" where no door event occurs), where N is the number of door objects in the environment. With this classifier, we can separate input signals containing information regarding events/states of multiple doors into a time-series of class probabilities for each class.

Here our method assumes that the smartphone periodically emits sine sweeps. Since the amplitudes of the sweeps are much larger than those of frequency shifts caused by the Doppler shift, the small frequency shifts might be ignored when the observed signals (or features extracted from the signals that include both the frequency shifts and sine sweeps) are directly fed into a generative model such as an HMM that learns a distribution of the input signals. In contrast, since a random forest classifier, which is the first-tier of our two-tier architecture, performs classifications based on thresholds, the scale of the input features does not affect the accuracy.

2.4.5 Discriminative Classifier for States

We prepared a discriminative classifier tailored to each door object that recognizes the states of the objects using information obtained by sine sweeps. For each time slice, a feature vector consisting of features related to impulse response is fed into the discriminative classifier (random forest). As mentioned above, we repeatedly emitted sine waves and sine sweeps from the smartphone (after each 10-s sine wave signal, the smartphone emits 0.05-s sine sweep signals). Therefore, we can obtain the state information of the objects only during the considerably short period of time when a sine sweep is emitted. If we aim to gather training data without separating the instances where the sweep was

emitted and where the sweep was not emitted, there is a strong possibility of incorrect and inefficient detection of the states of the objects. Since the time when the sine sweep signal is not emitted is much longer than the time when a sine sweep signal is emitted, a trained classifier can ignore the data corresponding to the time when the sweep is not emitted because the amount of the data is much smaller.

To solve this issue, we prepared a discriminative classifier for each door that separates the sweep information obtained when the door is in the opened state, the sweep information obtained when the door is in the closed state, the information obtained between the two sweeps when the door is in the opened state, the information obtained between two sweeps when the door is in the closed state, and the information that belongs to neither of the above states.

Therefore, the classifier is trained as a five-class classifier. Here we explain how to prepare a label for a feature vector at time t for the classifier. The feature vector is labeled as "opened@sweep" when the door was in the "opened" state and a sine sweep was emitted at time t. When the door was in the "closed" state and a sine sweep was emitted at time t, the feature vector is labeled as "sweep@closed." In contrast, when the door was opened and a sine sweep was not emitted at time t, the feature vector is just labeled as "opened." When the door was closed and a sine sweep was not emitted at time t, the feature vector is just labeled as "closed." Otherwise, the feature vector is just labeled as "others." The outputs of the classifier are a class probability for each class for each time slice.

2.4.6 Object-wise HMM

We prepared a set of HMMs tailored to each object, consisting of a left-to-right HMM for each event/state of the object, i.e., "open" event, "close" event, "opened" state, and "closed" state. A sequence of class probability vectors obtained by the discriminative classifier for events of the object and a sequence of class probability vectors obtained by a discriminative classifier for states tailored for the object are concatenated and then fed into the HMMs. Therefore, the values of the observed variables of the HMMs correspond to the concatenated probability vectors, and output distributions of each state in the HMM are represented by using a Gaussian mixture model (GMM). The hidden

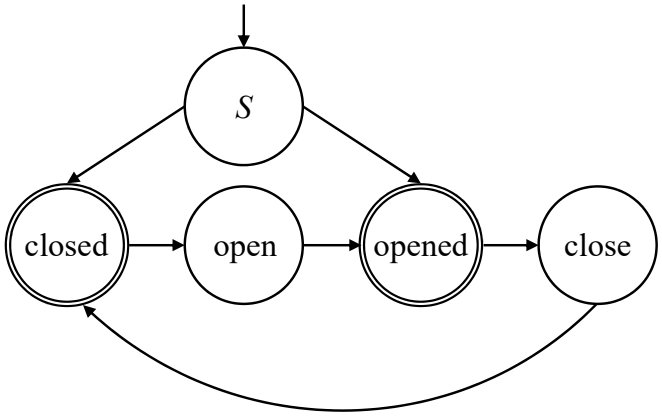


Figure 2.10: State transition diagram of a handcrafted grammar describing state transitions across HMMs

state of the HMM shows the internal state of the event/state of the door. In the model, the hidden state at time t depends only on the previous hidden state at time $t-1$, i.e., a Markov process. Moreover, the observed variable at time t depends only on the hidden state at time t. Therefore, using the HMMs to decode the input sequence enables us to capture the temporal regularity of events/states. We used the Baum-Welch algorithm [88] to train the HMMs. When we decoded test data (probability vector sequence) by employing the trained HMM set, we used the Viterbi algorithm to estimate the most probable state sequence in/across the HMMs [88]. The recognition results show which HMM (event/state) a probability vector at time t corresponds to.

Decoding with Grammar

We prepared a left-to-right HMM for each event/state, and the Viterbi algorithm was used to find state transitions across the HMMs. This means that, when we decoded HMM inputs, we took into account a state transition from the last state of an arbitrary HMM to the first state of another HMM. This corresponds to, for example, a state transition from an "opened" HMM to a "close" HMM (and to all other HMMs of the door).

Here we restrict the state transitions among HMMs by employing a grammar handcrafted based on prior knowledge about state transitions of a door object. For instance,

we can specify that a "close" event of a door occurs only when the door is in an "opened" state. We constructed such a grammar for door objects and Figure 2.10 shows the grammar represented in a state transition diagram. In Figure 2.10, the state "*S*" is the initial state and "closed" and "opened" are the final states. The state "*S*" is a dummy state introduced to limit the possible initial state of the door to be either "closed" or "opened."

After the state is transited to the "closed" state, for example, "open," "opened," "close," and "closed" events/states occur sequentially and are repeated. The state transition probabilities across the HMMs are determined based on the grammar (Figure 2.10). For example, the transition probabilities from the last state of the "opened" HMM to the first states of all other HMMs except for the "close" HMM are set to be 0. In contrast, the transition probability to the "close" HMM is defined as 1. Moreover, we specify the possible final state to be either "opened" or "closed." When we decode the HMM inputs using the Viterbi algorithm, we refer to the grammar to obtain the transition probabilities among HMMs. Note that, because the Viterbi algorithm finds the most probable state transitions, it works even when the initial state of the door is not specified.

2.4.7 Discriminative Classifier for Walking Events

For each time slice, a feature vector consisting of features related to the Doppler shift is fed into a discriminative classifier for walking events. The discriminative classifier is a random forest that is used to compute class probabilities of "walking" and "not-walking" classes in the environment of interest. Therefore, the random forest is a binary classifier.

2.4.8 HMMs for Walking Events

We prepared a set of HMMs, consisting of a left-to-right HMM for each walking event/state, i.e., "walking" and "not-walking." The training and decoding procedures are identical to those of the object-wise HMMs. Note that, because there is no restriction related to state transitions of the walking events, we assume that the transition probabilities between "walking" and "not-walking" are uniform.

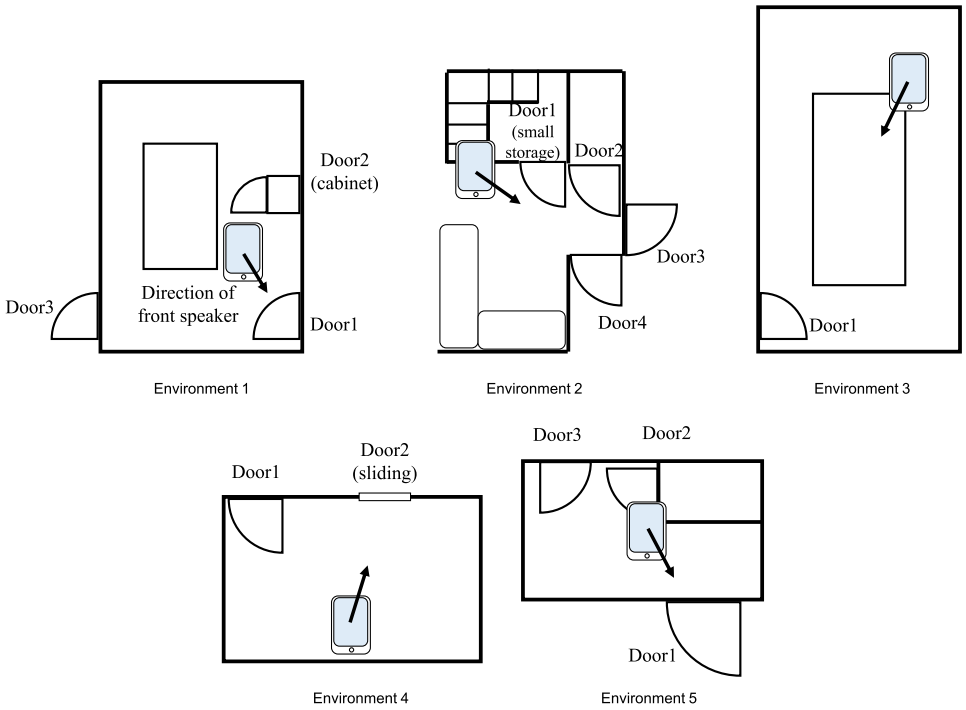


Figure 2.11: Our experimental environments. The sizes of environments 1, 2, 3, 4, and 5 are 3.5 m $\times$ 4.4 m, 3.2 m $\times$ 4.4 m, 4.4 m $\times$ 6.8 m, 3.3 m $\times$ 2.2 m, and 2.6 m $\times$ 1.8 m, respectively.

2.5 Evaluation

2.5.1 Data Set

We collected sensor data in real five environments. Figure 2.11 shows our five experimental environments and their settings. We installed Google Nexus 6P smartphone as shown in the figure. We recorded 44.1 kHz stereo audio using the front and back microphones of the smartphone. The smartphone emitted inaudible sinusoidal sound waves and sine sweeps during the experiment as described in the proposed method section. The smartphone also recorded time stamps when it emitted sine sweeps.

Environment 1 is a storage room with two doors and a cabinet. There are several discarded desktop PCs and shelves in the room. Environment 2 is a kitchen of a house with four doors. There are a refrigerator, a microwave oven, and other utensils. En-

vironment 3 is a meeting room with one door. The distance between the door and the smartphone is approximately 5 meters. In this room, we verified the performance of our method when the distance between a door and the smartphone is long. There are four tables and 11 chairs in the room. Environment 4 is a room of a house with two doors including a sliding door. In this room, we verified the performance of our method when events/states of a sliding door are recognized. Environment 5 is a room of a house with three doors. Because Door1 in the room is an entrance door, outside noises were observed when the door was in the opened state.

In each environment, a participant conducted 8 sessions of data collection. To obtain ground truth, we recorded the sessions with a web camera. Each object has two events and two states, i.e., "open" and "close" events and "opened" and "closed" states. Throughout a session, the participant used all objects so that each event of the objects occurred twice in an arbitrary order. That is, in a session, for example, a door can be opened while a cabinet door can be closed. In another session, the door can be closed while the cabinet door is opened. As above, each object is used under different conditions in our experiment. Because the participant walked at random in the room and used the objects in a random sequence, the doors were used from different sides and in different positions. The duration of a session was approximately 240 seconds. The average duration of "walking" within a session was approximately 15.5 seconds. The "walking" durations entirely depend on the size of the environment and the preference of the user. For example, Environment 5 (Figure 2.11) is smaller than the other environments of interest. Furthermore, to simulate the natural activities of a user, the participant must exit the room while closing the door behind him or her. These are the reasons why the average value derived from the "walking" data is only approximately 6% of the total duration of the session.

To investigate the effects of noises on the recognition performance, we collected additional three sessions of data in Environment 3 when another person was available (using a laptop PC). In addition, we collected additional three sessions of data in Environment 2 when a television was on. Furthermore, to investigate the effects of the ways of using doors on the recognition performance, we collected additional ten sessions of data in Environment 3 when each session was performed by a different participant. Moreover, to investigate the effects of a newly installed object in an environment, we

collected additional three sessions of data in Environment 2 after a chair was installed. These investigations will be described later.

2.5.2 Evaluation Methodology

We evaluated the performance of our method for each environment with leave-one-session-out cross-validation. We prepared the following methods to investigate the effectiveness of the composite sine waves, two microphones, our two-tier architecture with discriminative classifiers, and a handcrafted grammar.

- *Proposed*: This is the proposed method.
- *w/o back MIC*: This method does not use sounds recorded from the back microphone in the feature extraction process. The other procedures are identical to those of *Proposed*.
- *w/o composite*: This method does not use composite sine waves. Instead, this method uses only 20 kHz sine waves. The other procedures are identical to those of *Proposed*.
- *w/o grammar*: This method does not use a handcrafted grammar for each object. That is, the transition probabilities from one HMM to the other HMMs are uniform. The other procedures are identical to those of *Proposed*.
- *HMM*: This method does not use discriminative classifiers for events/states. Extracted audio features for Doppler shift and impulse response are directly fed into an HMM set tailored for each object.

Recognition accuracy for each of the above methods is evaluated using the precision, recall, and F-measure, calculated based on the recognition results per window of data. The number of states of each HMM and the number of Gaussians in each state are 10 and 8, respectively.

Table 2.2: Classification accuracies for the proposed method in Environment 1

Object	Event/state	Precision	Recall	F-measure
Door1	open	0.542	0.886	0.672
	close	0.632	1	0.774
	opened	0.988	0.924	0.955
	closed	1	0.938	0.968
Door2	open	0.650	0.998	0.787
	close	0.59	0.935	0.723
	opened	0.987	0.940	0.963
	closed	1	0.899	0.947
Door3	open	0.474	1	0.643
	close	0.541	0.914	0.680
	opened	1	0.975	0.987
	closed	0.995	0.938	0.966

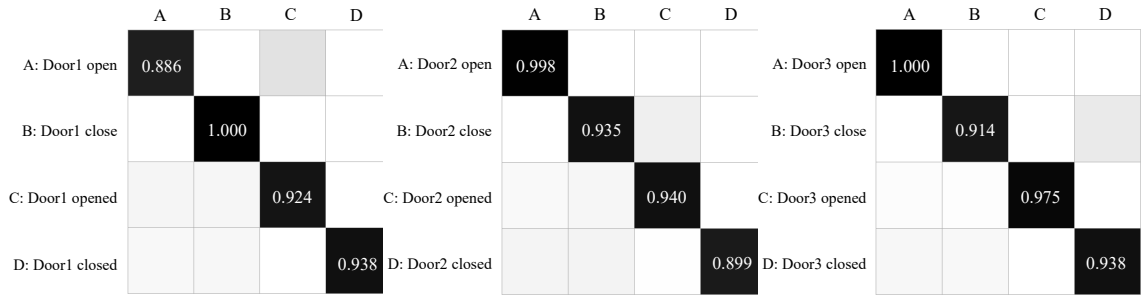


Figure 2.12: Visual confusion matrices for the proposed method in Environment 1

2.5.3 Result of Door Event/State Recognition

Performance of the Proposed Method

Table 2.2 shows the classification accuracies for *Proposed* in Environment 1. In addition, Figure 2.12 shows visual confusion matrices for *Proposed* in Environment 1. As can be seen in the results, *Proposed* accurately recognized door events/states in Environment 1 even though the recorded sounds include noises caused by the participant (like Figure 2.4) because the participant walked around the room. However, the ac-

Table 2.3: Macro-averaged precision, recall, and F-measure for the proposed method in the five environments

Env.	Object	Precision	Recall	F-measure
1	Door1	0.790	0.937	0.842
	Door2	0.807	0.943	0.855
	Door3	0.752	0.957	0.819
2	Door1	0.838	0.963	0.884
	Door2	0.796	0.952	0.850
	Door3	0.834	0.972	0.886
	Door4	0.791	0.953	0.848
3	Door1	0.970	0.978	0.974
4	Door1	0.820	0.939	0.865
	Door2	0.681	0.738	0.705
5	Door1	0.844	0.975	0.895
	Door2	0.834	0.981	0.891
	Door3	0.869	0.976	0.912

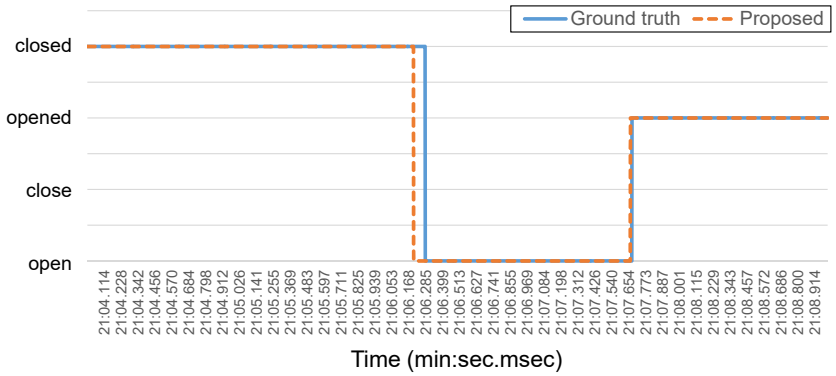


Figure 2.13: Estimates of the proposed method for Door1 in Environment 1. The graph also shows the ground truth, which was obtained from video recordings.

curacies for the “open” and “close” events are somewhat poor compared to those of the “opened” and “closed” states. Figure 2.13 shows an example of the outputs of our

method related to Door1 in Environment 1. As shown in the figure, the estimated start time of the "open" event has a small error (approximately 100 msec). Since the number of instances (feature vectors) belonging to the "open" and "close" classes is small because the durations of these events are short, these small errors greatly affected the classification accuracies for these classes. However, because our method could capture the state changes of objects, the accuracies for the "opened" and "closed" states are almost perfect. We believe that most of the applications based on door event recognition can work well even when the detected starting/ending time of an event has a difference of 100 msec from the ground truth.

Table 2.3 shows the classification accuracies for *Proposed* in the five environments. Even though Door2 in Environment 1 is considerably smaller and located behind the smartphone, the accuracies for the door are just as good as the accuracies for the larger doors that are located in front of the smartphone. Similarly, as shown in the results of Door2 and Door3 in Environment 5, *Proposed* could precisely recognize events/states of doors behind the smartphone. In addition, even though the distance between the smartphone and Door1 in Environment 3 is about 5 m, *Proposed* could accurately recognize events/states of the door. In contrast, the accuracies for Door2 (sliding door) in Environment 4 are somewhat poor because it did not create a large enough Doppler shift to be captured by our smartphone. This is one limitation of the proposed method. However, we confirmed that the recorded sound sometimes captured weak frequency shifts when the sliding door was opened/closed even if the participant opened the door from outside the room. This is the reason why the accuracies for Door2 are not very poor.

Impact of Stereo Sound

Figures 2.14 - 2.18 show the macro-averaged F-measure of *w/o back MIC* for each door object in each environment. The accuracy of *w/o back MIC* for Door2 in Environment 1 is somewhat poorer than that of *Proposed*. This could be because the door is located behind the smartphone. In Environment 3, *Proposed* achieves a 17.2% higher accuracy than *w/o back MIC*. It can be assumed that the back microphone captured sound waves reflected by the wall behind the smartphone. In addition, because the back microphone seems to be more sensitive, which is used for noise cancellation, using the

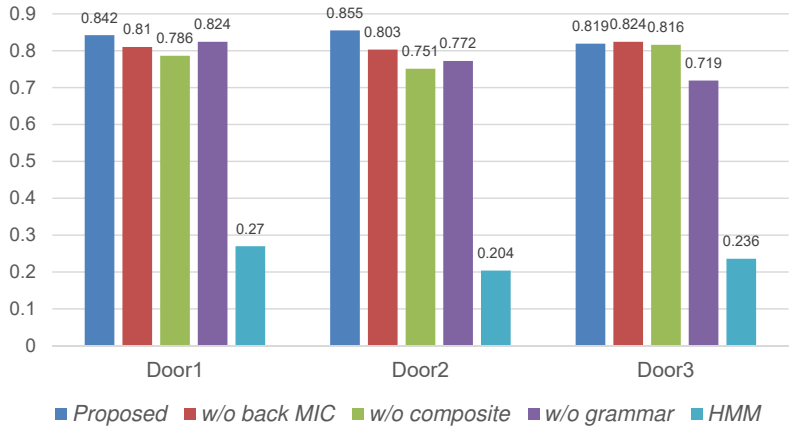


Figure 2.14: Macro-averaged F-measures of the five methods in Environment 1

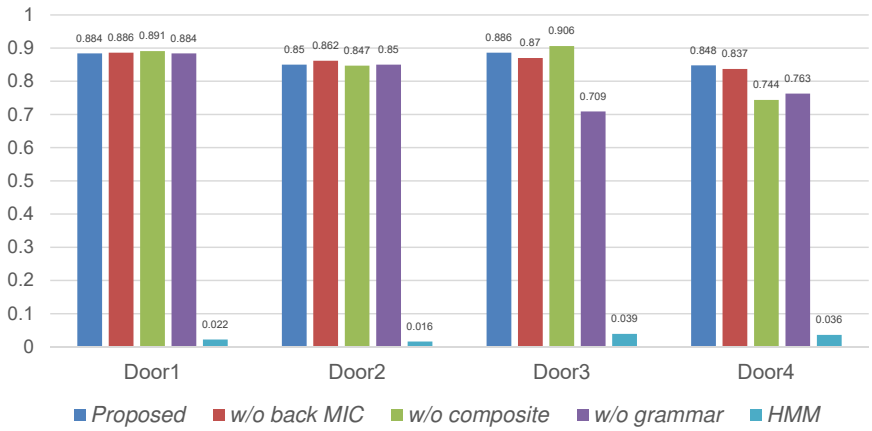


Figure 2.15: Macro-averaged F-measures of the five methods in Environment 2

back microphone is also effective when recognizing events of objects located far from the smartphone. Interestingly, *w/o back MIC* achieves a 77.6% accuracy for the sliding door in Environment 4. In contrast, the F-measure for *Proposed* is 70.5%. Since the smartphone was very close to a wall as shown in Figure 2.11, we consider that the back microphone did not record any useful acoustic information because it did not capture sound waves reflected from the walls behind.

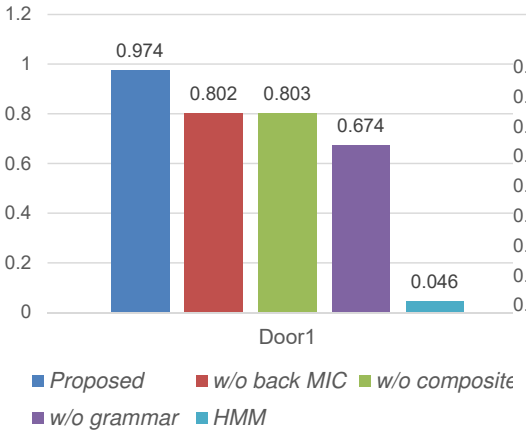


Figure 2.16: Macro-averaged F-measures of the five methods in Environment 3

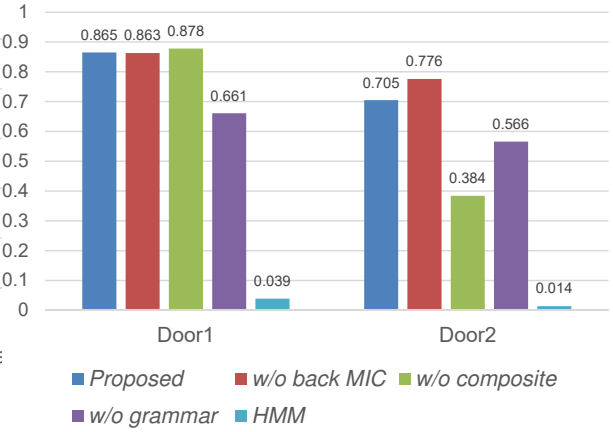


Figure 2.17: Macro-averaged F-measures of the five methods in Environment 4

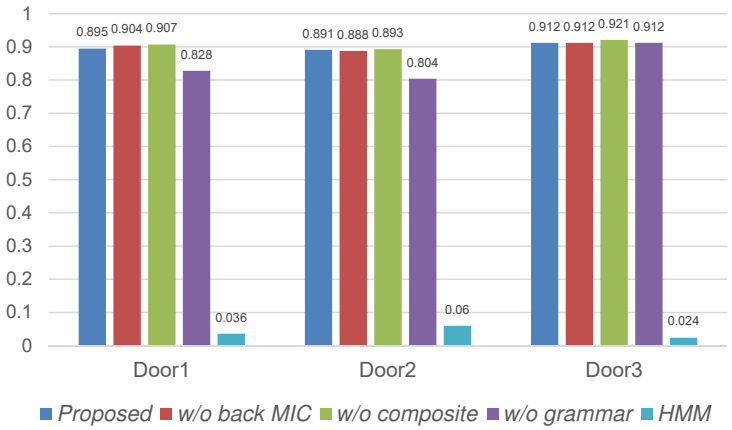


Figure 2.18: Macro-averaged F-measures of the five methods in Environment 5

Impact of Composite Wave

Figures 2.14 - 2.18 also show the performance of *w/o composite*. The accuracies of *w/o composite* for Door1 in Environment 1 and Door 1 in Environment 3, which are located in front of the smartphone, are much poorer than those of *Proposed*. This could be because, when the smartphone faces the direction of a door, the door events do not cause strong frequency shifts for 20 kHz signals as shown in Figure 2.7 (a), which can

relate to the directionality of the high-frequency signals.

The accuracy of *w/o composite* for Door2 in Environment 4 is 38.4%. This is because we could not recognize any frequency shifts around a frequency range of 20 kHz when the sliding door was opened/closed. In contrast, we recognized small frequency shifts around a frequency range of 18 kHz.

Contribution of Grammar

Figures 2.14 - 2.18 also show the performance of *w/o grammar*. As can be seen in the results, using the grammar significantly improved the recognition accuracies of many objects. In particular, the accuracy for Door1 in Environment 3 improved by 30%. This could be because it is difficult to estimate the states of a door located far from the smartphone. Figure 2.19 shows an example of the state transition of Door1 in Environment 1 estimated by *w/o grammar* in comparison to *Proposed*. As shown in the figure, *w/o grammar* outputs impossible state transitions, e.g., the transition from the "open" event to the "opened" state. In contrast, because *Proposed* decodes input signals based on the grammar, *Proposed* can output correct estimates even when it is difficult to estimate the states of a door completely on the acoustic characteristics.

Contribution of Two-Tier Architecture

Figures 2.14 - 2.18 also show the performance of *HMM*. As shown in the results, it is impossible to recognize door events using the simple HMM architecture. As mentioned above, because the smartphone periodically emits sine sweeps whose amplitudes are much higher than those of frequency shifts caused by the Doppler shifts, the small frequency shifts are ignored when the extracted sound features are directly fed into the HMMs, which are generative models that learn distributions of observed variables. The accuracies in Environments 2, 3, 4, and 5 are extremely poor because almost all of the instances were mistakenly classified into open/close classes. In contrast, our method, which is based on hybrid discriminative/generative models, is designed to detect the small frequency shifts using the discriminative classifier, which is robust against the problems regarding the scale.

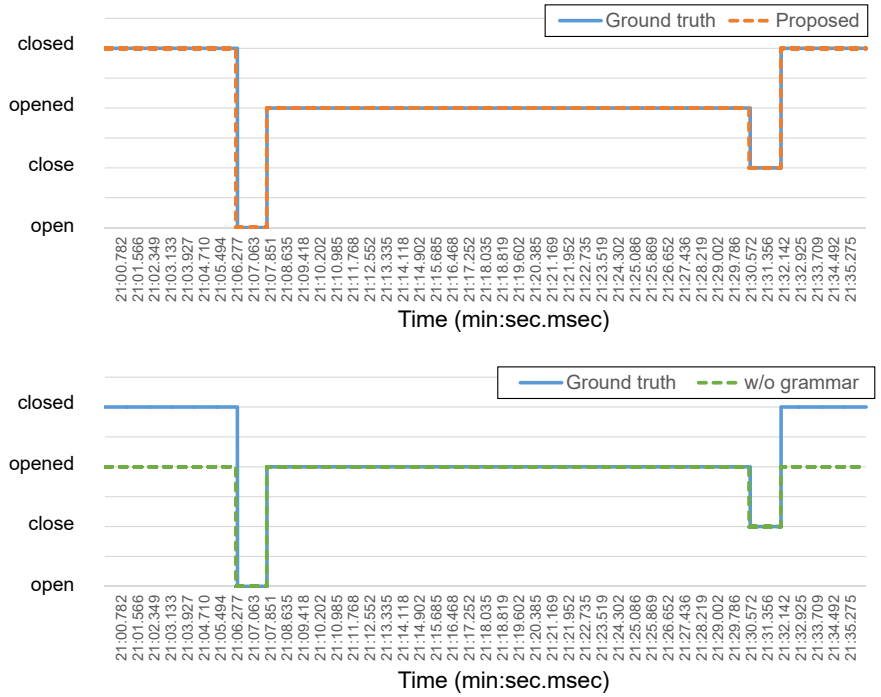


Figure 2.19: Estimates of *Proposed* and *w/o grammar* for Door1 in Environment 1

2.5.4 Results of Walking Event Recognition

Table 2.4 shows the recognition performance of the walking events. As shown in the results, our method could achieve highly accurate walking event detection. The accuracy for "walking" is somewhat poorer than that for "not-walking" because the durations of "walking" events are shorter than those of "not-walking" events. The proposed method failed to recognize the starting and ending parts of "walking" events as shown in Figure 2.20. In addition, because of the limited viewing angle of the video camera, it was difficult to precisely make the ground truth labels. The accuracies in Environment 1 were somewhat poorer than those in the other environments. This environment is a compact environment where Door3 opens into another very small storage space. When Door3 was opened and the participant moved inside that storage space, we observed unexpected Doppler shifts caused by his movements even if he was not present in Environment 1, causing false positives in the classification. We believe that the reflected sound was emphasized by the walls in the storage space since the storage space is very

Table 2.4: Accuracies of walking detection for the proposed method. (The results of Environment 3 are unavailable. Because there is only one door in the environment, the walking events did not occur.

	Event	Precision	Recall	F-measure
Env. 1	walking	0.637	0.876	0.738
	not-walking	0.976	0.910	0.942
Env. 2	walking	0.797	0.850	0.823
	not-walking	0.978	0.968	0.973
Env. 4	walking	0.887	0.814	0.849
	not-walking	0.941	0.966	0.953
Env. 5	walking	0.777	0.842	0.808
	not-walking	0.955	0.932	0.943

Table 2.5: Accuracies of Door2 and Door3 in Environment 5 when the entrance door (Door1) was in the "opened" and "closed" states.

	Door	Avg. precision	Avg. Recall	Avg. F-measure
Door1 opened	Door2	0.833	0.983	0.891
	Door3	0.867	0.977	0.911
Door1 closed	Door2	0.836	0.980	0.891
	Door3	0.856	0.976	0.903

small.

2.6 Discussion

2.6.1 Environmental Noises

We collected additional three sessions of data in Environment 2 when a television was on and a television program was played with medium volume. We trained a recognition model of *Proposed* on data collected when the television was off (8 sessions of data).

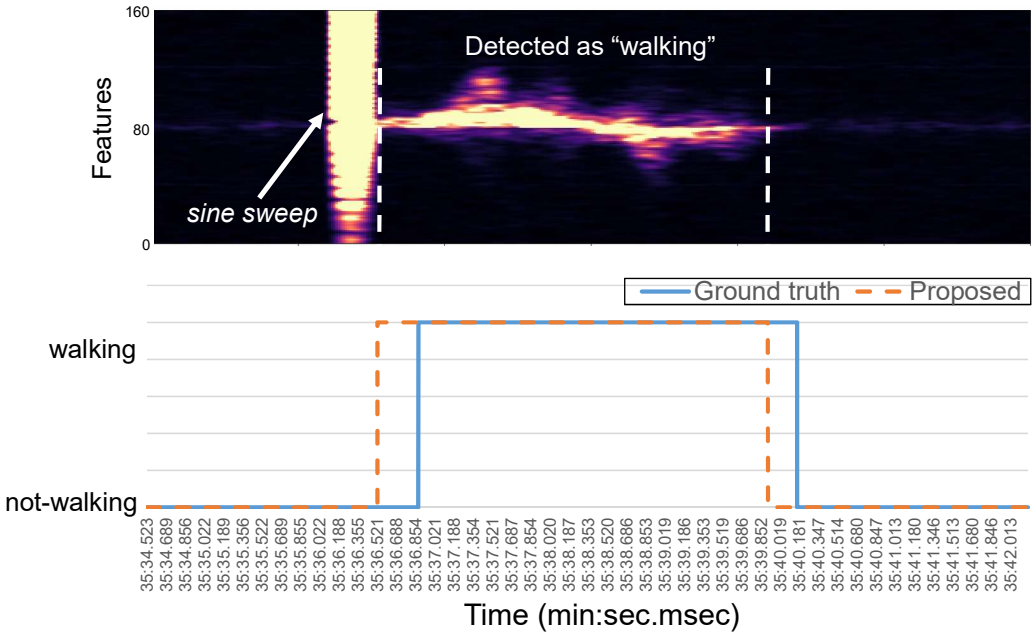


Figure 2.20: Estimates of the proposed method (lower panel) and spectrogram related to frequencies used for classification features (upper panel)

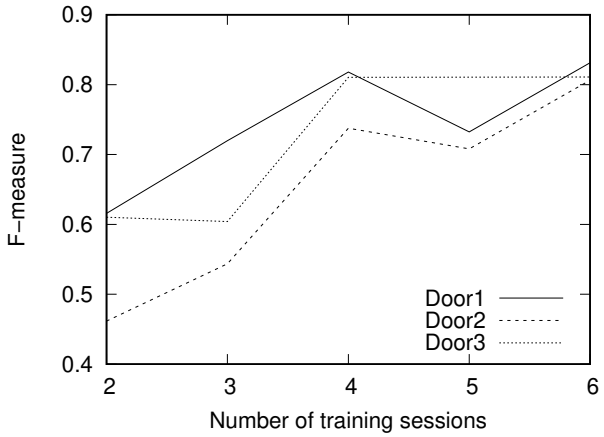


Figure 2.21: Transitions in macro-averaged F-measure when the number of training sessions is varied

The macro-averaged F-measures of Door1, Door2, Door3, and Door4 for the additional sessions were 0.885, 0.856, 0.715, and 0.798, respectively. As can be seen in the results, the F-measure of Door3 significantly decreased (see Table 2.3). A possible reason for this result is that the television programs in the authors' country (Japan) contain audio frequencies up to 20 kHz (see Figure 2.8), which can interfere with the composite sine wave emitted by the smartphone, hence worsening the recognition accuracy of the proposed method. We confirmed that the estimated starting/ending times of the "open" events of Door3 had large errors. However, the F-measures of the "opened" and "closed" states are still very high; 0.992 and 0.906, respectively.

As mentioned in Section 2.3.6, when an entrance door is in the opened state, outdoor noises can be observed by a smartphone. In Environment 5, we calculate the event recognition accuracies for Door2 and Door3 when Door1 (entrance door) was in the opened or closed state. When Door1 was in the opened state, outdoor noises were recorded as shown in Figure 2.8. The recognition accuracies for Door2 and Door3 were presented in Table 2.5. As shown in the results, even when Door1 was in the opened state, the proposed method, which consists of discriminative and generative models, achieved good recognition accuracies.

2.6.2 Effect of Additional Person

We collected additional three sessions of data in Environment 3 when a person was using a laptop PC, sitting between the smartphone and Door1. The macro-averaged precision, recall, and F-measure of these sessions for *Proposed* are 0.828, 0.974, and 0.883, respectively, when we train a recognition model on data collected when there was no person in the environment (8 sessions of data). The precision for the additional sessions was somewhat poorer than that of when there was no additional person in the environment as shown in Table 2.3 since the precisions for the open and close events are 0.670 and 0.643, respectively. This may be caused by small fluctuations of frequency caused by the person between the smartphone and Door1. However, the F-measures for the opened and closed states are as high as 0.974 and 0.979, respectively.

2.6.3 Way of Using Door

We collected eight sessions of data in Environment 3 when each session was performed by a different participant to investigate the effects of the ways of using a door. The way of using the door was different from participant to participant. For example, a participant entered the room while opening the door. In contrast, another participant opened the door while standing still. We evaluate the additional data with leave-one-session out cross-validation using *Proposed*. The macro-averaged precision, recall, and F-measure of Door1 were 0.935, 0.949, and 0.942, respectively. While the accuracies slightly decreased from those in Table 2.3, the F-measures for the door states are still high; 0.985 and 0.991 for the opened and closed states, respectively.

2.6.4 Additional Object

To investigate the effects of a newly installed object in an environment, we collected additional three sessions of data in Environment 2 after a chair was installed in front of Door3 and Door4. We trained a recognition model using *Proposed* on data collected before the installation (8 sessions of data). The macro-averaged F-measures of Door1, Door2, Door3, and Door4 for the additional sessions were 0.903, 0.857, 0.775, and 0.857, respectively. Since the chair was placed between the smartphone and Door3, the F-measure for Door3 dropped down to 0.775. We found that the output probabilities of the discriminative classifier for events after the chair was introduced were lower than the output probabilities for training data. Because the HMMs were trained on the output probabilities of the discriminative classifier for the training data, the trained HMMs could not deal well with the output probabilities for the test data.

To achieve robust recognition, we computed simulated pseudo-output probabilities of the discriminative classifier for events and used them as additional training data for the HMMs. We simply reduced the actual output probabilities for the training data by multiplying each output probability by d (randomly selected from [0.3,0.9]). The macro-averaged F-measures of Door1, Door2, Door3, and Door4 when we used the robust method were 0.894, 0.847, 0.865, and 0.843, respectively. These results prove that, by using our robust method, we can reuse training data collected under a given condition of an environment to achieve a high recognition accuracy when test data are

collected after placing a small obstacle, i.e., a chair, between the smartphone and a door.

2.6.5 Amount of Training Data

To investigate the amount of training data required to achieve accurate predictions using our recognition model, we performed the following test using eight sessions of data collected in Environment 1. We selected n sessions randomly as our training data and employed the remaining sessions as test data to obtain the recognition accuracy of the trained model. We repeated this process three times and calculated the average F-measure for each door. Then we performed the above test while changing the value of n from 2 to 6.

Figure 2.21 shows the relationship between the number of training sessions and the macro-averaged F-measure. As shown in the graph, we could achieve an average F-measure greater than 0.8 for Door1 (0.82) and Door3 (0.81) when we used 4 training sessions. However, the average F-measure for Door2 (0.74) is comparatively lower than the F-measures of Door1 and Door3 when 4 training sessions were used. The main reason for this is that Door2 in Environment 1 is a cabinet door that is considerably smaller than Door1 and Door3. Therefore, the door events of Door2 created comparatively smaller Doppler shifts, hence making it harder for the recognition model to differentiate from Doppler shifts created by other movements inside the room using a small number of training sessions.

2.6.6 Structure of the System

In our method, we proposed to set up one smartphone per environment, where an environment represents a room in a house. Therefore, multiple smartphones have to be employed in order to implement this system for the entire house. In contrast to the distributed sensor method where sensors have to be mounted on every door or window of the house, the number of probes required by our method is equal to the number of rooms in the house. The reduced number of sensors is one of the main advantages of our method over the distributed sensor method. However, this raises the problem of how to communicate between these smartphones so that the entire network of the smartphones works coherently as a single system. This negatively affects the simplicity of our pro-

posed method. Therefore, we consider each smartphone as a separate system, and each of them connects to Wi-Fi separately. While probing, audio data from each time slice are fed into the feature extraction process described in Section 2.4.3. Next, the extracted features will be sent to a server for event recognition via the Wi-Fi network.

2.6.7 Limitations of the Proposed Method

The evaluation revealed that it was difficult to accurately recognize events/states of a sliding door because our method is based on the Doppler shift created by the door movement. However, our experiments revealed that weak frequency shift was caused by events of the sliding door. In addition, the evaluation revealed that walking events in a room next to a room of interest were mistakenly detected. We believe that such false detection can be removed if another smartphone is installed in the next room.

2.7 Conclusion

We proposed a method to recognize door events via active sound probing using a smartphone. Our method achieved accurate recognition employing a composite sine wave, stereo recording, a two-tier discriminative/generative recognition model, and a grammar that describes state transitions of a door. We collected data from different environments with different conditions and confirmed the effectiveness and validity of the proposed method.

At present, smart speakers such as Amazon Echo and Google Home have proliferated. As a part of our future work, we plan to implement our method on such a device to recognize events of the surrounding doors more accurately as they are equipped with more than two microphones. (Amazon Echo and Echo Dot are equipped with a seven-piece microphone array.)

Chapter 3

IndoLabel: Predicting Indoor Location Class by Discovering Location-specific Sensor Data Motifs

3.1 Introduction

Because of the recent proliferation of smart devices such as smartphones, and smartwatches, context recognition technologies such as indoor positioning that use sensors mounted on these smart devices are actively studied. Indoor positioning, i.e., estimating the indoor coordinates of a smart device user or estimating the location class (location semantics) of a smart device user primarily relies on signaling technologies such as infrared [121], ultrasound [87], active sound probing [30, 110], bluetooth [119], and Wi-Fi [53]. Particularly, the estimated location classes are useful for investigating a user's daily life because the semantic information of a user's current location is strongly related to his/her activities. For example, if the estimation is that a user is located in a room with a semantic label of "restroom," this information can be used as prior information to estimate the user's activity (e.g., using toilet). Additionally, the estimated location classes can be used to label lifelog data and to adaptively manage lifelogging devices, for example, turning off a wearable camera if a user enters a restroom [30].

This study focuses on location class prediction. Particularly, we develop a method

called, IndoLabel, to predict room-level location classes, i.e., semantic label, using sensors such as accelerometers and microphones on off-the-shelf smartwatches. We aim to estimate the *type* of geographic locations (e.g., kitchen, bedroom, and smoking area classes). We will not estimate locations from a defined set of locations in an environment of interest. From a sensor data from a smartwatch, we determine the inherent sensor data features for each location class. For example, we can observe knife chopping actions in kitchens using body-worn accelerometers. In addition, we can observe similar sound features by active sound probing in bathrooms because of their water-resistant walls. Because these sensor data features can be observed in almost all types of environments (e.g., kitchens in any house), these features can be considered inherent sensor data used to predict location classes. We automatically extract these location-specific sensor data and aggregate them to estimate the location classes. Note that because we build a location classifier that captures inherent features of location classes, **we do not require *any* labeled sensor data collected in a target environment**. Figure 3.1 shows an example of location class prediction by our method. We use labeled sensor data from different locations in training environments to train a location classifier. Thereafter, we collect sensor data from a target environment and use the location classifier to predict location labels of locations in the target environment.

Previous studies on location class prediction use labeled training data collected in a target environment. Although location class prediction methods that do not require training data from a target environment have been studied, they use one of the following assumptions. The first assumption is a location-specific sensor data features are continuously and abundantly observed in locations of interest [101]. However, in reality, location-specific sensor data are not abundantly and continuously observed when a user is in a particular location. For example, a user will perform the chopping action in a kitchen for a significantly short time compared to the duration of his stay in the kitchen. These prior methods perform efficiently only in an environment where a user spends almost the entire duration on performing cutting actions (or other kitchen-specific actions) if the user is in the kitchen. The second assumption is, a set of location-specific features is predefined to enable prediction [29]. Therefore, a prediction rule is handcrafted for each location class in advance.

Unlike as in previous studies, IndoLabel automatically detects sensor data motifs

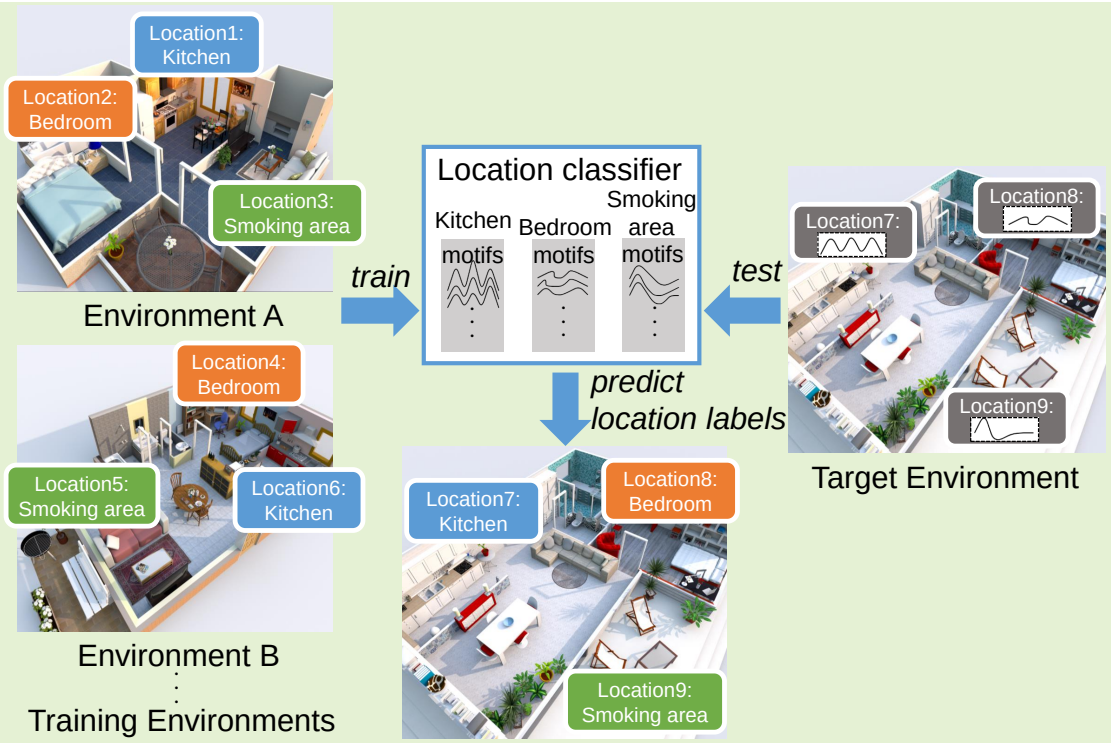


Figure 3.1: Example of location class prediction by IndoLabel in an unseen target environment using the proposed method. Location-specific sensor data templates (motifs) are extracted from the labeled sensor data collected in several locations in training environments. A location classifier trained by these motifs is thereafter used to estimate the location labels of locations in the target environment by using the collected data. IndoLabel does not use any labels in the target environment.

(short data segments) that are observed in a location instance belonging to a specific location class and predicts the location class of the location instance by leveraging the sensor data motifs. This approach enables predicting location classes based on sensor data features that are not continuously observed in a specific location class. In particular, handcrafted rules for each location class are not required. Assume that a user washes dishes in a kitchen and subsequently moves to a washroom to brush his/her teeth. Herein, hand movements corresponding to washing dishes can be considered kitchen-specific motions, and hand movements associated with tooth brushing can be

considered washroom-specific motions. In contrast, hand movements related to walking in the kitchen and washroom are not location-specific motions because they are observed in both locations and are not useful for predicting location classes. To address this problem, this study proposes a novel matrix manipulation that automatically detects location-specific data segments (motifs) from time series sensor data collected in multiple training environments by calculating a score that represents "location specificity" of each segment in the time series based on the concept of Gini impurity [35], that is used to determine significant classification features. After motif discovery, we build a location classifier based on detected location-specific motifs. Herein, to address sensor data differences in different environments, we introduce a domain-adversarial neural network, which is an unsupervised domain adaptation method, into motif detection. Note that, unlike standard domain-adversarial neural networks, our network is designed to capture environment-independent features of location-specific motifs as well as achieve high recognition accuracy of the location classifier. Because motif extraction and classifier training are performed using labeled training data (sensor data with location labels) collected in training environments, our method does not require labeled training data collected from a target environment.

In our experiment, we assume that acceleration, sound, and Wi-Fi data are collected using a smartwatch in a target environment. Additionally, we assume that data collected for a significantly long period, (e.g., one day) is processed. The proposed method initially detects places where a user stayed for a long time by clustering Wi-Fi scans. Using this clustering, Wi-Fi scans collected in the same room are grouped into one cluster. Subsequently, a location classifier estimates the location class of each location cluster using aggregated acceleration and sound data collected in a room (location cluster). (We can obtain acceleration and sound data collected in the same location cluster via timestamps of Wi-Fi, acceleration, and sound data.) As described, we can obtain a location class label for each location cluster. Because a set of Wi-Fi scans is associated with each cluster, we can estimate the location class of a user in *real-time* using only a Wi-Fi scan from the user's device once a location class label is predicted for each location cluster.

The primary contributions of this study are summarized as follows: To the best of our knowledge, IndoLabel is the first method that estimates location classes by au-

tomatically finding location-specific sensor data motifs. Furthermore, this is the first study to introduce the concept of Gini impurity into a method that scans time series to find location-specific templates (motifs). In addition, we use domain-adversarial neural networks to eliminate environment-dependent components in the discovered motifs. Unlike standard domain-adversarial neural networks, the proposed network used in IndoLabel is designed to achieve high motif detection accuracy and accuracy of location label prediction. The technical contributions of this study are i) a method for discovering location-specific motifs, ii) a method for detecting motifs that also maintains high location prediction performance, and iii) a novel location label prediction pipeline composed of the above techniques. We collected sensor data in real home environments and evaluated the proposed method using the leave-one-environment-out cross-validation. We showed that our method can accurately predict location labels in a target environment using labeled training data collected in other environments.

3.2 Related Work

3.2.1 Location Class Prediction

Predicting a semantic label (location class) of a user's current position using smartphone/wearable sensors is actively studied in the ubicomp community. Azizyan et al. [5] used acceleration, image, Wi-Fi, and light features from smartphone sensors and acoustic features to estimate the logical location labels of stores such as Walmart and Starbucks. However, several methods require training data collected from the target environment. Fan et al. [30] conducted a study similar to ours and investigated estimating the location class of a user using active sound probing. However, they only focused on the detection and recognition of the restroom class to adaptively turn a wearable camera on and off to protect the privacy of a user.

Bao et al. [8] estimated room-level location semantics using user behaviors such as frequency of visits, and maximum, and average duration of stay. In contrast, we aim to extract inherent sensor data motifs for each indoor location class. Tachikawa et al. [101], which is our prior work, estimated room-level location semantics by extracting inherent sensor data features for each location class using a modified decision tree. They

used a magnetometer, barometer, and microphone to extract location-specific features and assumed six location classes in four laboratory/office environments. The differences between our method and that of Tachikawa et al. [101] is summarized as follows.

1. They assumed that location-specific features were continuously observed in a target environment because they used sensors such as magnetometer and microphone to record magnetic, air-pressure, and acoustic features that are predominantly observed in locations such as an elevator, and restroom. Unlike our method, this method cannot leverage short sensor data segments that characterize location classes such as acceleration segments corresponding to cutting in a kitchen. As mentioned in the Introduction section, such actions only last for a significantly short amount of time, compared to the entire duration of the user's stay at the location.

2. They employed a classic machine learning approach (modified decision tree) for location class prediction based on handcrafted features. In contrast, we adopt the feature learning approach, i.e., deep learning, using a domain-adversarial neural network tailored to location class prediction.

Elhamshary et al. [29] estimated location semantics in a train station. They used a microphone, accelerometer, gyroscope, magnetometer, and barometer to extract location-specific features and assumed nine location classes. Their method relied on handcrafted rules and sensor data templates tailored to each location class. In contrast, our method automatically extracts location-specific sensor data motifs (templates) based on our matrix manipulation technique.

3.2.2 Landmark-assisted Indoor Positioning

Indoor landmarks are used to correct accumulated errors in simultaneous localization and mapping (SLAM) techniques [1]. Hardegger et al. [38] performed SLAM based on the fact that certain daily activities were performed at particular places (e.g., sleeping in a bedroom). Indoor objects such as pillars and electrical appliances show high magnetic field values. Wang et al. [116] and Gozick et al. [36] detected such magnetic fingerprints using a magnetometer in a smartphone to locate the smartphone user. Detecting

a user's location-specific events such as climbing and descending stairs was achieved using barometers [6, 4]. In contrast, our study uses location-specific sensor data segments to predict location classes. Although most studies rely on handcrafted rules or templates, our method automatically discovers location-specific sensor data segments.

3.2.3 Motif Detection

Vahdatpour et al. [113] proposed a novel method for locating multi-dimensional motifs to discover activities and events in multi-dimensional time series data. They addressed a common problem in activity monitoring applications, i.e., time and value irregularities of motifs. Jain et al. [39] proposed a method for discovering frequent motifs in time series data that significantly performed in input data with noise. They used a greedy alternating minimization-based approach to arrive at the solution after formulating the problem of discovering motifs in time series data as a large optimization problem. Linardi et al. [59] proposed a scalable motif discovery algorithm that aids the discovery of all motifs in a given segment in time series data and ranking of the motifs using the length-normalized distance. Maekawa et al. [65] measured the duration of each operation period of a factory worker who wears an acceleration sensor by tracking a motif that appears only once in each operation period using a particle filter. Xia et al. [128, 127] estimated the starting and ending times of each operation in an operation period based on the tracked motif using information derived from process instructions. In contrast, our motif discovery approach is based on calculating the similarity between sensor data segments with location labels to discover motifs specific to a location class based on the concept of the Gini impurity. To the best of our knowledge, this is the first study to discover motifs specific to a location class by introducing the Gini impurity and detects the occurrences of the motifs using domain-adversarial neural networks.

3.3 Location class estimation method

3.3.1 Preliminaries

We assume that we observe unlabeled sensor data and Wi-Fi scans from a target user wearing a smartwatch in a target environment. (Our experiment uses acceleration data and sound impulse response.) We also assume that we observe labeled sensor data from training users who wear smartwatches in training environments. Therefore, for each time slice t, a Wi-Fi scan and a sensor data segment $\{w_t, \mathbf{s}_t\}$, where w_t is a Wi-Fi scan at time t and $\mathbf{s}_t$ is a sensor data segment at time t, is given. In our experiment, $\mathbf{s}_t$ corresponds to a segment of acceleration data extracted from a time window of acceleration data between time $t-w+1$ and time t, where w is a window size, or a feature extracted from the sound impulse response at time t, which is a multidimensional time series with a length of 1 (this will be subsequently explained in detail). Using the Wi-Fi scans, we determine place clusters where a user spends long periods. Using the sensor data, we predict a location class for each place cluster. We also assume that the smartwatch is paired with a smartphone on the user. The smartwatch transmits the sensor data and Wi-Fi scans to the smartphone via Bluetooth Low Energy (BLE). The smartphone also collects acceleration data using an onboard sensor. Such acceleration data are used to detect locomotion in place clustering. A location class label is associated with a training sensor data segment at each time slice.

3.3.2 Overview

Figure 3.2 shows the overview of the proposed method. Because this study aims to predict location labels in a target environment using a location classifier trained in training environments, the proposed method consists of two main phases: training and testing. In the training phase, we extract location-specific motifs using labeled training data, i.e., sensor data such as acceleration data, from training environments. Using the extracted motifs, we build a motif classifier that is used to detect occurrences of location-specific motifs from the time series of test data. To address environmental differences in motifs (e.g., toothbrushing actions can be different in different environments), we introduce domain-adversarial training to the motif classifier. Finally, we train a location classifier

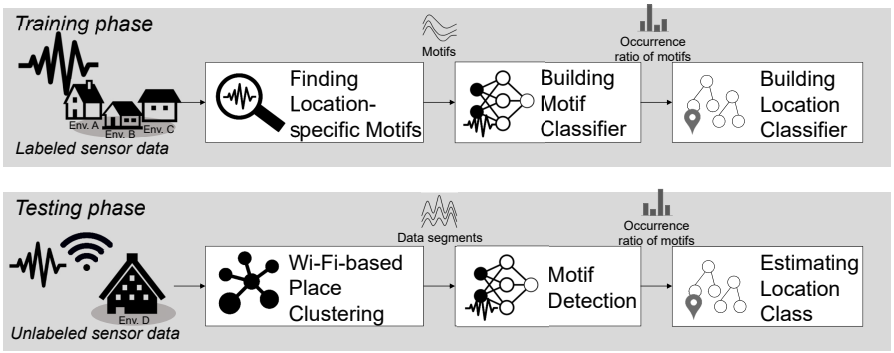


Figure 3.2: Overview of IndoLabel

that uses detected motifs to predict a location class.

In the testing phase, we collect sensor data associated with Wi-Fi scans over a long duration from a target user in a target environment. We initially cluster Wi-Fi scans to obtain place clusters. Wi-Fi scans collected in the same room are aggregated into a place cluster. For each cluster, we detect the occurrences of motifs and estimate the location class of the cluster using the trained location classifier. Once a location class label is predicted for each location cluster, we can estimate the location class of a user in *real-time* using only a Wi-Fi scan from the user's device because a set of Wi-Fi scans is associated with each cluster.

After introducing sensor data preprocessing, we present details of the training and testing procedures.

3.3.3 Preprocessing

Acceleration Data

To reduce the noise while preserving the most significantly unique variations in the acceleration data, we apply Principal Component Analysis (PCA) [83] to the acceleration signals, thereby reducing the dimensionality from the three-dimensional data to one-dimensional data. This step significantly reduces the computational cost of the steps to follow and provides an advantage in locating location-specific data segments. Figure 3.3 (a) and (b) shows examples of three-axis acceleration data and the signals converted

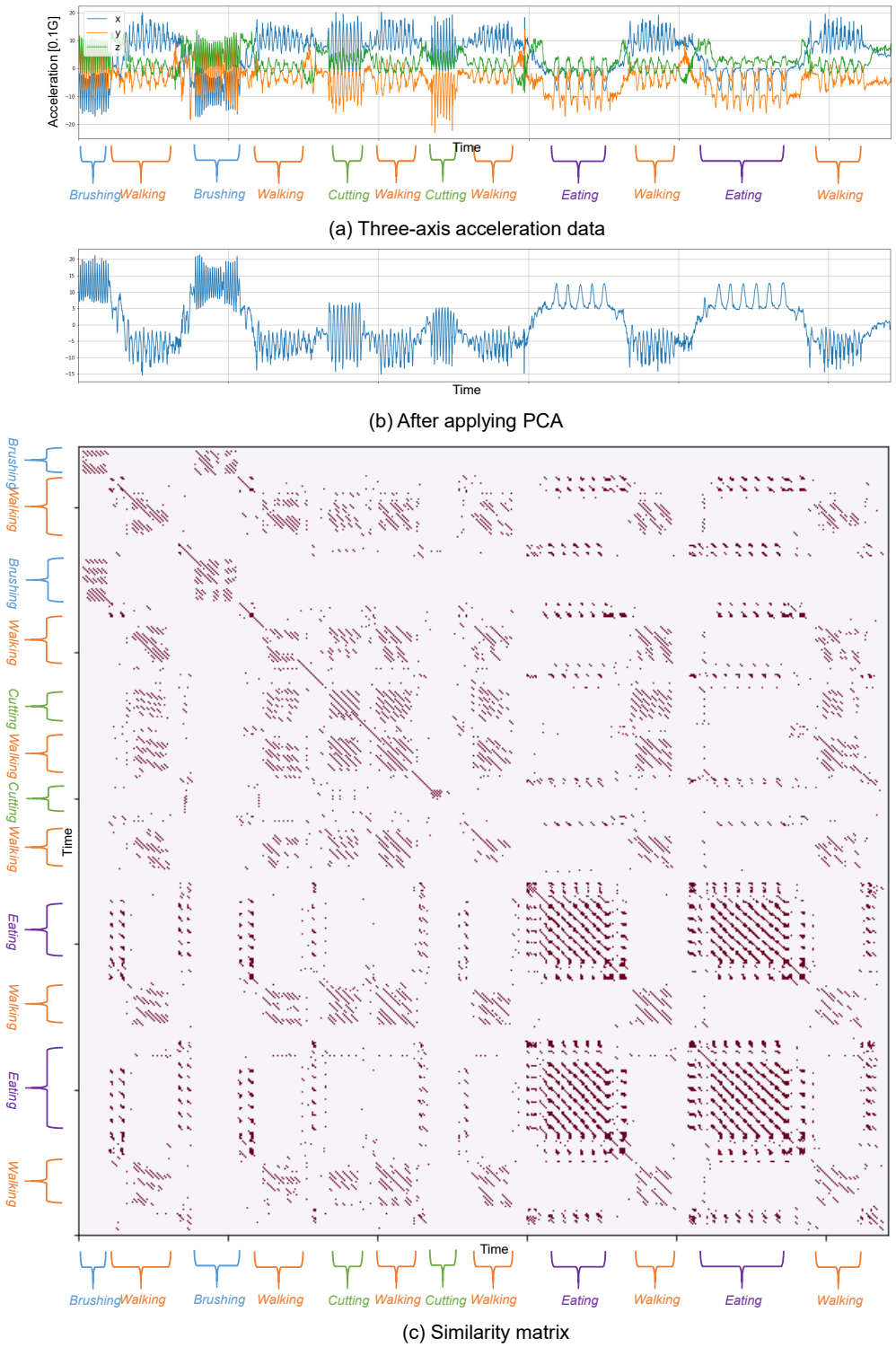
by PCA, respectively. Then we extract the segment in a time window that corresponds to $\mathbf{s}_t$ in Section 3.3.1.

Impulse Response

The acoustic characteristics of a space can be captured using impulse responses. An impulse is a signal that is one at time zero, however, is zero otherwise and equally contains all the frequencies in a frequency domain. This enables an impulse response to contain varieties of acoustic information about spaces, such as the space between an audio source and a receiver, and a room where a probing device is positioned. Impulse responses, in particular, contain time-domain information related to sound propagation, such as reflections, reverberations, and echoes. Factors in an indoor environment such as construction materials, shape, size, and furnishings affect observed impulse responses [52, 91]. This ensures that impulse responses are useful fingerprints of environments of interest.

This study aims to capture and extract environment-independent inherent features for each location class of interest. Environmental factors of the same location class are strongly correlated because instances belonging to the same location class serve the same purpose. For example, because a house's den is typically used to perform document work, this environment contains books, papers, and computers. In contrast, a toilet contains lavatory basins, water-resistant floors, and washbasins. Therefore, impulse responses observed in the same class of locations have similar fingerprints.

However, generating a strong, yet significantly short pulse, i.e., an impulse, in real-world situations is difficult. Several studies have used alternate methods for calculating impulse responses without generating an actual impulse [3, 96]. This study uses a sine sweep that does not require tight synchronization between the signal generator sampling clock and the digitizing unit of the probing device that is used to capture the response. This aids in conducting the measurements using a commercially available off-the-shelf smartwatch [31, 32]. In our method, an excitement signal sweeping frequency between 18 kHz and 20 kHz is emitted by a smartwatch for 1 s.



(a) Three-axis acceleration data

(b) After applying PCA

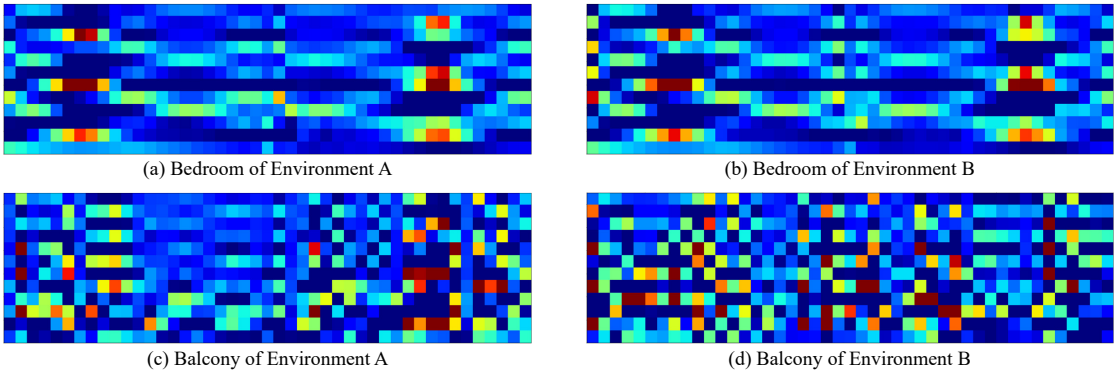
(c) Similarity matrix

Figure 3.3: An example of acceleration data and a similarity matrix computed from the data. The darker cells in the similarity matrix indicate data segments with higher similarity.

After our smartwatch application initiates recording sounds using the smartwatch microphone, the application generates an inaudible sine wave sweep using the smartwatch speaker. After the sine sweep stops, the application continues to record for 0.5 s to record the reverberation. Subsequently, we obtain the impulse response of the signal by calculating the convolution of the recorded audio with the time-reversal mirror of the emitted signal. We ignore impulse responses obtained during hand movements (detected by the variance of acceleration data) because the impulse responses are dominated by information about hand movements. Thereafter, we extract 12-dimensional features, i.e., the MFCC coefficients, by calculating the 12-order MFCC for each sliding time window.

The normal MFCC algorithm that is commonly used as a feature extraction method in speech recognition primarily focuses on the lower frequency bands when extracting audio features. This means that the Mel scale is more discriminative at lower frequencies, and less discriminative at higher frequencies. However, this study aims to capture location-specific audio characteristics by using a sweep that is inaudible to the human ear (18 kHz–20 kHz). Therefore, the MFCC algorithm must be optimized to the frequency range of our probing signal. We achieve this by adjusting the frame size (1 ms) and overlapping of the frames (90%) in calculating the Discrete Fourier Transform (DFT) and by adjusting the Mel-filter banks to obtain considerably significant resolution in the 18 kHz–20 kHz frequency range.

Figure 3.4 shows examples of time series of MFCC computed from the impulse responses from different locations in two different environments, i.e., Environment A and Environment B. From Figure 3.4, the MFCC features of the balconies are considerably different in these environments. This is owing to the high-frequency noise observed in outside environments [25]. In contrast, MFCC features in the bedrooms are similar in these environments. However, these MFCC features in different environments are slightly different, for example, there are small shifts of peaks (red cells) in the time and frequency axes. To obtain robust features from the MFCC features, we extract local binary patterns (LBP) [79, 133]. LBP features are often used in image processing and recognition to extract features regarding patterns in images that are robust against fluctuations in pixels. We adapt this method to extract LBP features from a time series of 12-order MFCC features for each impulse response. Consequently, we obtain multidi-



(a) Bedroom of Environment A (b) Bedroom of Environment B

(c) Balcony of Environment A (d) Balcony of Environment B

Figure 3.4: Examples of MFCC features at different locations in different environments; bedroom (while lying on a bed) and balcony (while smoking). Red and blue cells indicate large and small amplitudes, respectively. The horizontal axis represents the time axis and the vertical axis represents the MFCCs.

mensional data with a length of 1 for each impulse response that corresponds to $\mathbf{s}_t$ in Section 3.3.1.

3.3.4 Finding Location-specific Motifs

Herein, we explain the process of discovering location-specific motifs in the training phase. This process comprises of two procedures: (i) similarity matrix calculation and (ii) calculation of the location specificity measure (LSM) of each sensor data segment.

Similarity Matrix Calculation

This step calculates a similarity matrix using sensor data segments (i.e., a sequence of $\mathbf{s}_t$) collected in each training environment for each sensor modality. This is aimed at determining similar segments in the sensor data. We initially compute the Euclidean distance between each pair of segments (i.e., $\mathbf{s}_i$ and $\mathbf{s}_j$) and form a distance matrix. For each segment, before calculating the distance, we standardize the data in the segment by subtracting the average from each component in the segment and dividing by the standard deviation. This causes its static segments with small noise to yield a large distance and causes segments with similar waveforms to yield a smaller distance in the

distance matrix calculation.

Thereafter, we convert the distance matrix into a normalized similarity matrix by dividing each element by the maximum value in the distance matrix and subtracting them from one. Because the low similarity components are less important in frequent motif discovery, we replace the components showing less than a threshold value in the similarity matrix with 0.0. Herein, an element value of the matrix $\mathbf{S}(i, j)$ shows the similarity between a segment at time i and that at time j. In addition, a row vector $\mathbf{S}_{(i)}$ shows a time series of the similarity between a segment at time i and each segment.

Calculating Location Specificity Measure (LSM)

This step calculates the degree of location specificity, i.e., the LSM for each segment of data based on the concept of Gini impurity, which is computed using the following formula to calculate the statistical dispersion:

$$G = 1 - \sum_{i=1}^{C} p_i^2,$$

where p_i is the proportion of instances belonging to the i-th class, and C is the total classes [134]. Therefore, a smaller Gini coefficient indicates a smaller dispersion of instances. This concept can be used to find a data segment specific to a particular location class. In our case, we compute p_i for the segment that represents the ratio of the occurrences of the segment at the i-th place. Therefore, using p_i, the location specificity of the segment is calculated (i.e., if the segment is observed only at a specific location) based on the idea of the Gini impurity.

Note that there is another metric that can be used to calculate the statistical dispersion of a dataset, i.e., information gain. Raileanu et al. [89] revealed that there is no significant difference between the Gini impurity and information gain if considering splitting a dataset. Furthermore, the computational cost of the Gini impurity is significantly lower than that of the information gain[1]. Because our method calculates

[1]Comparison of the calculation times of Gini impurity and information gain using 100,000 data points with Python version 3.6.8, running on Intel Xenon CPU ES-2620 at 2.10 GHz.
Gini impurity calculation time = 20.938 msec
Information gain calculation time = 31.592 msec

the location specificity based on a large matrix, we select the Gini impurity over the information gain to calculate the LSM.

We assume that a similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ and binary time series of location labels with length n (i.e., ground truth labels) are given. The binary time series $\mathbf{b}_c$ is prepared for each location class c, where its element value at time t is 1 if the training user is at the location class at time t, that is defined as follows:

$$b_{c,t} = \begin{cases} 1 & (\text{the training user is at } c \text{ at time } t) \\ 0 & (\text{otherwise}) \end{cases}$$

For a row vector of $\mathbf{S}$ at each time slice t, i.e., $\mathbf{S}_{(t)}$, we compute $s_{c,t}$ as follows:

$$s_{c,t} = \frac{\mathbf{b}_c \cdot \mathbf{S}_{(t)}^{\mathrm{T}}}{\sum\limits_{b_{c,t'} \in \mathbf{b}_c} b_{c,t'}}.$$

Because $\mathbf{S}_{(t)}$ is the similarity time series for a time window (segment) at time t, $s_{c,t}$ shows the frequency of the occurrences of segments similar to the segment at place c. Furthermore, we compute $p_{c,t}$ as follows:

$$p_{c,t} = \frac{s_{c,t}}{\sum_{i=1}^{C} s_{i,t}}.$$

Therefore, $p_{c,t}$ shows the ratio of the occurrences of the data segment at time t at place c. Using $p_{c,t}$, we compute the LSM of the segment based on the idea of the Gini impurity as follows:

$$\mathrm{LSM}_t = \sum_{i=1}^{C} p_{i,t}^2,$$

A large LSM value shows greater location specificity of the data segment. Figure 3.5 shows an example of time series of LSM values. From the example, location-specific actions (e.g., eating in a dining room) yield high LSM values. In addition, walking actions that are observed in multiple locations, yield low LSM values. Furthermore, because some cutting actions are similar to walking actions, their LSM values are small; thus, these actions cannot be location-specific motifs. Some meaningless actions, for example, a sitting action prior to eating in a dining room, have high LSM values because the length of the example data in Figure 3.5 is significantly short. If such actions

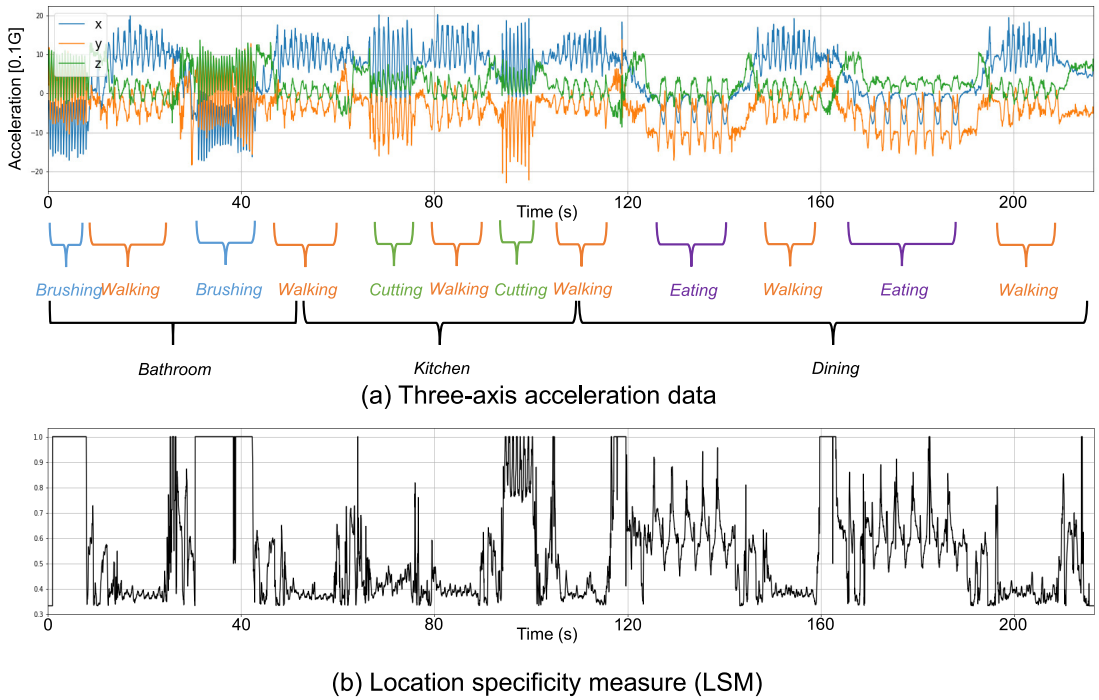


Figure 3.5: An example of an acceleration data and time series of LSMs computed from the data

are also observed in other locations, the LSM values of the actions will be lower. In this study, segments with LSM values larger than a threshold value were considered location-specific motifs.

3.3.5 Building Motif Classifier

In the above procedures, we obtain location-specific motifs from each training environment. Herein, we build a classifier that is used to detect occurrences of location-specific motifs in a target environment for each sensor modality. A sensor data segment is input into the classifier, and a location-specific motif class associated with the segment is the output. For example, a segment corresponding to a chopping action can belong to a "kitchen-specific motif" class or a "kitchen motif class" in short. The total possible output classes of the classifier are $C + 1$, where C is the total location classes, i.e.,

location-specific motif classes, and the plus one denotes the "others" class that corresponds to training segments randomly sampled from non-location-specific motifs. We train the classifier with the extracted location-specific motifs from the training environments. In our method, training instances corresponding to multiple different actions observed in the same location class can be assigned to the same location-specific motif class. For example, chopping and blending actions, that can be observed in a kitchen, are assigned to a kitchen-specific motif class, i.e., kitchen motif class because the aim of the classifier is to detect an action specific to a particular location class, not to distinguish fine-grained actions.

Note that in building the classifier, we consider the variability in sensor data motifs in different environments. For example, a chopping action can be significantly different in different environments. To address this problem, we build a motif classifier based on domain-adversarial neural networks [34]. This network uses domain-adversarial training that learns a domain-independent representation by learning the classifier that simultaneously predicts correct class labels and inaccurately predicts the domain's labels. We use this method to learn environment-independent features and develop a classifier that is robust against environmental differences.

Our proposed method aims to predict the location label of a place cluster of interest. In the next procedure (location classifier in Section 3.3.6), we aggregate the output probabilities from the motif classifier from the same place (place cluster) to predict the location label of the place cluster. Therefore, unlike the standard domain-adversarial neural network, our motif classifier (domain-adversarial neural network) is trained to maximize the performance of the location classifier and maintain the motif classification accuracy.

Figure 3.6 shows the structure of the neural network used in this study. The network has two outputs; motif class and environment labels. Herein, the motif class corresponds to the location-specific motif class, and the environment corresponds to the domain in domain-adversarial learning. The feature extractor in Figure 3.6 that is comprised of 1D convolutional layers, extracts domain-independent features from a sensor data segment. The motif label predictor in Figure 3.6 outputs a motif class label, i.e., a location-specific motif class, using the features extracted by the feature extractor. To ensure that the neural network cannot distinguish between the domains, the gradient reversal layer [34] is

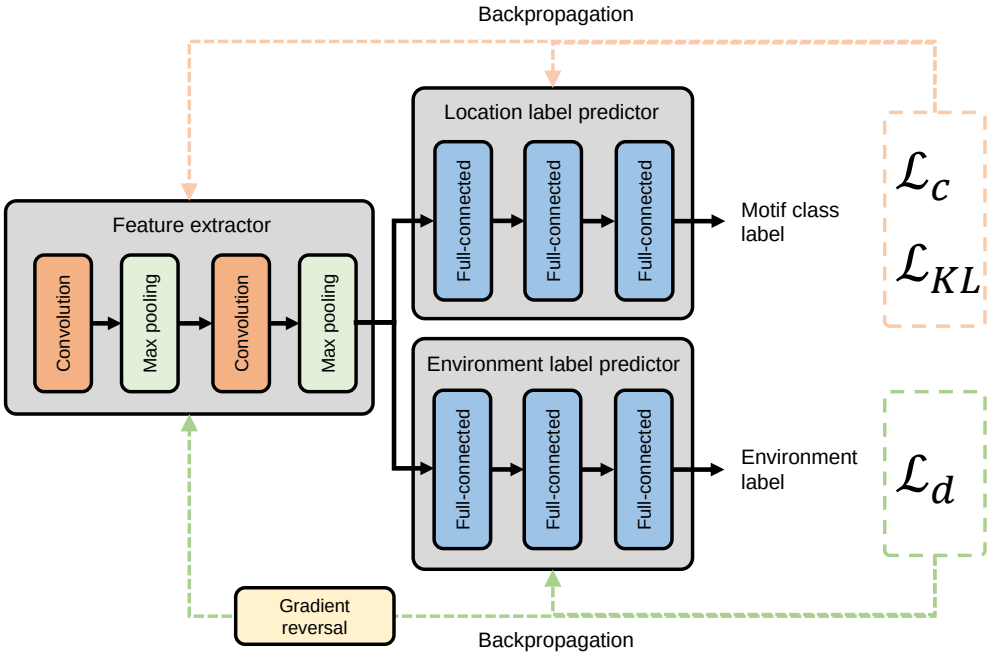


Figure 3.6: Structure of motif classifier based on domain-adversarial learning. $\mathcal{L}_c$, $\mathcal{L}_d$, and $\mathcal{L}_{KL}$ show the loss functions for motif classification, domain classification, and distributions of output probabilities, respectively.

introduced as shown in Figure 3.6. When the network is trained using the backpropagation algorithm [93], the gradient reversal layer multiplies the gradient with a negative constant value, resulting in a feature extractor that cannot estimate domains but classes.

When we train the network, we minimize the following loss function using backpropagation based on Adam [46].

$$\begin{aligned} E(\theta_f, \theta_c, \theta_d) =& \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_c^i(\theta_f, \theta_c) - \lambda_1 \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_d^i(\theta_f, \theta_d) \\ & - \lambda_2 \mathcal{L}_{KL}(\theta_f, \theta_c), \end{aligned} \tag{3.1}$$

where θ_f, θ_c, and θ_d represent the network parameters of the feature extractor, motif label predictor, and environment label predictor, respectively; n is the number of training instances (time series), $\mathcal{L}_c^i(\theta_f, \theta_c)$ denotes the loss of motif class prediction, and $\mathcal{L}_d^i(\theta_a, \theta_d)$ is the loss of domain prediction. $\mathcal{L}_c^i(\theta_f, \theta_c)$ and $\mathcal{L}_d^i(\theta_a, \theta_d)$ calculate the cross

entropy loss. $\mathcal{L}_{KL}(\theta_f, \theta_c)$ is introduced to maximize the performance of the location classifier. The hyperparameters λ_1 and λ_2 are used to trade-off between the loss functions. Since the scales of the loss components $\mathcal{L}_c^i(\theta_f, \theta_c)$, $\mathcal{L}_d^i(\theta_a, \theta_d)$, and $\mathcal{L}_{KL}(\theta_f, \theta_c)$ were similar, we set that $\lambda_1 = \lambda_2 = 1$ in our experiment.

Herein we present details of $\mathcal{L}_{KL}(\theta_f, \theta_c)$. In the following procedure, we aggregate output probabilities for motif prediction from the motif classifier from feeding sensor data segments observed at the same place cluster. The output probabilities (output probability vectors) were averaged and fed into the location classifier. Therefore, the location classifier predicts a location label of a place cluster by referring to all the sensor data segments observed at the place (place cluster). Therefore, the location classifier can achieve high accuracy if the distributions of output probabilities differ between location-specific motif classes. From this idea, we design $\mathcal{L}_{KL}(\theta_f, \theta_c)$ such that the distance between distributions of output probabilities between different location-specific motif classes is large. Because a training batch contains labeled training instances, we aggregate the output probabilities of the training instances for each location-specific motif class in each training iteration. Thereafter, we calculate the distance between the distributions of each pair of location-specific motif classes. Finally, we update the neural network parameters to maximize the distances. Therefore, $\mathcal{L}_{KL}(\theta_f, \theta_c)$ is defined as follows.

$$\begin{aligned}\mathcal{L}_{KL}(\theta_f, \theta_c) &= \frac{1}{{}_2\mathrm{P}_C} \sum_{i,j \in C} D_{KL}(P_i || P_j) \\ &= \frac{1}{{}_2\mathrm{P}_C} \sum_{i,j \in C} \int_x^x P_i(x) \log \left(\frac{P_i(x)}{P_j(x)} \right) dx,\end{aligned} \tag{3.2}$$

where ${}_2\mathrm{P}_C$ shows the permutation. The i and j show the i-th and j-th location-specific motif classes, respectively. The i-th and j-th classes are different classes. P_i denotes the distribution of the output probabilities for training instances belonging to the i-th class. $D_{KL}(P_i||P_j)$ calculates the distance, i.e., the Kullback-Leibler divergence [49], between P_i and P_j. The proposed model is trained to discriminate between location-specific motif classes using a set of sensor data segments observed in the same location cluster.

3.3.6 Location Classifier

We develop a location classifier to predict the location class of a place cluster in a target environment. We estimate the location class of a place cluster based on the occurrence frequencies of location-specific motifs that are the outputs of the motif classifier. Therefore, the input of the classifier is the occurrence ratios of location-specific motif classes, and the output is a location class. Herein, we explain how we compute the input for each place cluster. Initially, we compute the output probabilities of each segment in the cluster using the motif classifier. The motif classifier outputs a probability vector consisting of $C + 1$ probability values. Thereafter, we calculate the average vector over the segments in the place cluster and normalize after removing the element corresponding to the "others" class. The average vector is an input of the location classifier. We manipulate the multimodal sensor data (e.g., acceleration data and impulse response data) by calculating the probability values for each sensor modality and concatenating the probability values for all sensor modalities. For example, if using s sensors and C classes, the total probability values to calculate is $s \times C$.

The quantity of training data for the location classifier is limited because we can construct only one training instance for each place. Therefore, this study uses the nearest centroid [74], where each class is simply represented as a centroid of instances belonging to the class, as the location classifier.

3.3.7 Wi-Fi-based Place Clustering

In the testing phase, we cluster Wi-Fi scans collected by a target user. A single Wi-Fi RSSI scan comprises pairs of a unique MAC address of an access point (AP) and the received signal strength from that AP. Figure 3.7 shows differences in the received signal strength of each AP at the two places.

The observed scans at the same place are similar; thus, we can cluster these scans to form Wi-Fi location clusters. In this procedure, we initially use smartphone acceleration data to detect stable time segments (i.e., non-movement time segments) and thereafter perform Wi-Fi-based clustering using the Wi-Fi scans collected during stable segments. For each sliding time window, we calculate the variance of the acceleration data and consider windows, where the variance values are smaller than a threshold, as stable

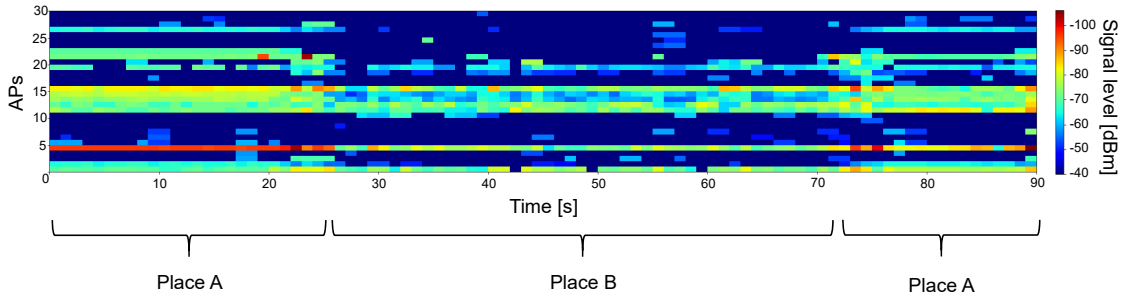


Figure 3.7: An example of signal strength from different APs over time as a user moves from Place A to Place B and thereafter back to Place A

segments. In Wi-Fi-based clustering, we perform a distance calculation between two Wi-Fi scans based on [101, 67]. Herein, assume that we compute the distance between Wi-Fi scans w_i and w_j. If the received signal strength from AP a_n is present in scan w_i although not in scan w_j, we consider the signal strength of AP a_n in scan w_j as -100 dBm that corresponds to the minimum strength value.

Figure 3.7 shows that some APs only appear in several scans. For example, AP no. 30 is only detected in the 39th scan (it does not appear in any other scan). Therefore, we assume that this AP does not have a significant influence on the clustering of scans into different places. To eliminate such ineffective APs in scans, we calculate the standard deviation of the scans for each AP and eliminate APs with standard deviations less than 1.0.

Thereafter, the distance between scans w_i and w_j is calculated as follows:

$$d(w_i, w_j) = \frac{1}{|A|} \sum_{a_n \in A} |w_i(a_n) - w_j(a_n)|, \tag{3.3}$$

where $d(w_i, w_j)$ is the distance between scans w_i and w_j, A is the set of APs in scans w_i and w_j, and $w_i(a_n)$ is the received signal strength from AP a_n included in w_i. Finally, we use agglomerative hierarchical clustering [43] to cluster scans in stable segments into different places using the Wi-Fi distances. Herein, each cluster is a collection of Wi-Fi scans observed at a particular place by the target user's smartphone.

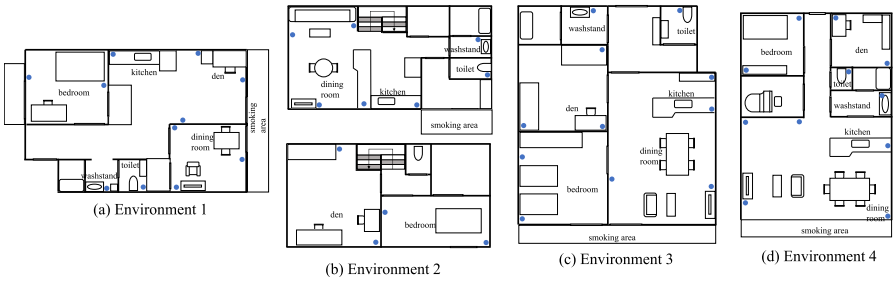


Figure 3.8: Experimental environments. The blue dots denote the locations of the Wi-Fi APs installed inside each environment.

3.3.8 Estimating Location Class

Finally, we estimate the location class of each location cluster in the target environment. Herein, each Wi-Fi scan w_i in the location cluster is associated with a set of $\mathbf{s}_i$, i.e., a set of data segments. We feed each data segment in $\mathbf{s}_i$ in the location cluster to the motif classifier to determine the probability vector corresponding to the location cluster by averaging the output probabilities of the data segments.Thereafter, we input this probability vector into the location classifier. The output of the location classifier is the estimated location class of the location cluster.

3.4 Evaluation

3.4.1 Dataset

To evaluate our method, we collected data from four different home environments (Figure 2.11). Each environment contained eleven Wi-Fi APs. The location of each AP is denoted by a blue dot in Figure 2.11. Note that we could also detect APs from nearby houses and RSSI from these APs were also used in Wi-Fi-based place clustering. The average numbers of APs detected at Environment 1, 2, 3, and 4 were 35, 57, 35, and 36, respectively. In each environment, a participant collected sensor data using an ASUS ZenWatch3 smartwatch attached to his/her dominant wrist in seven locations (kitchen, dining room, restroom, washstand, den, smoking area (outdoor resting place),

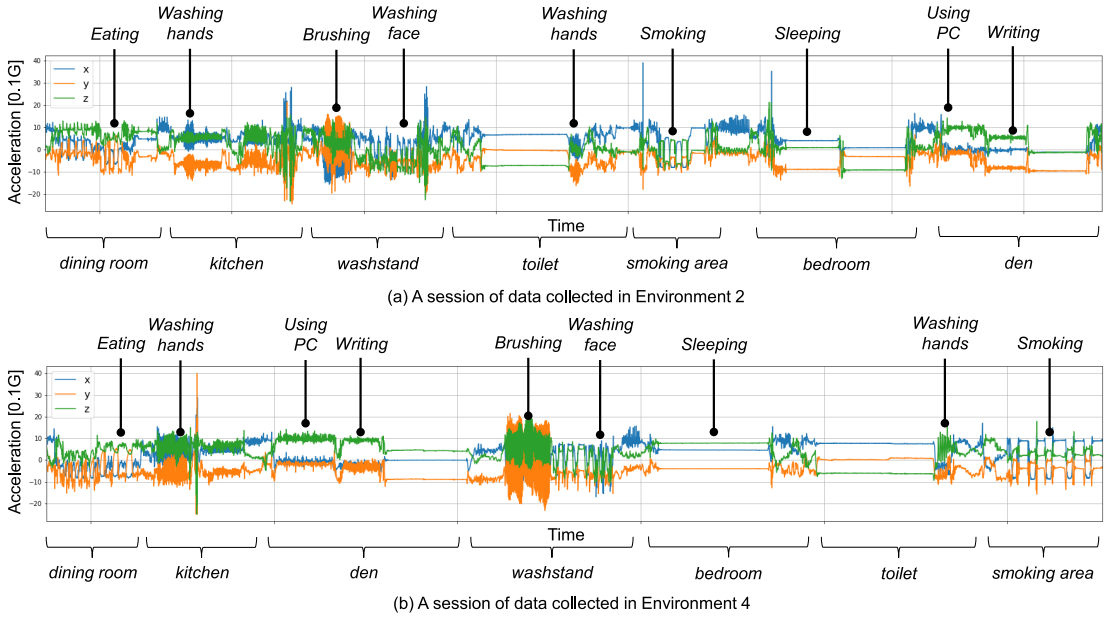


Figure 3.9: Two sessions of real data collected in Environment 2 and Environment 4

and bedroom). Herein, we considered seven location classes; therefore, this problem is a seven-class classification problem. We selected these location classes based on studies [30, 90, 101] and places that were typically frequented by participants in their daily routine. We collected sensor data using a semi-naturalistic collection protocol [7] that permits significant variability in participant behavior than laboratory data. In this protocol, participants performed a random sequence of activities at corresponding locations. Table 3.1 presents a list of the activities performed at each location. The participants sometimes moved between locations before finishing the predefined activities at each location. For example, a participant sometimes performed the chop and wash dishes activities at the kitchen, then went to the dining room, performed sit, eat and, drink activities, and returned to the kitchen to perform wash dishes and wash hands activities. Figure 3.9 shows two sessions of data collected by two participants from Environment 2 and Environment 4. In real situations, there are specific actions that are performed in several locations. For example, washing hands actions can be performed in locations such as the kitchen and washstand, where there is access to water. Therefore, washing hands actions are performed in kitchen, washstand, and toilet location classes to

Table 3.1: List of locations and specific actions performed

Location	Activity
kitchen	chop, wash dishes, wash hands
washstand	brush teeth, wash face, wash hands
bedroom	lie
toilet	sit, use toilet paper, wash hands
dining room	eat, drink, sit
den	use PC, write on a notebook, sit
smoking area	smoke

Table 3.2: Physical features of participants

	Dominant hand	Height (cm)	Sex	Age	No. of sessions
1	right	175	male	26	6
2	right	177	male	27	7
3	right	168	male	30	8
4	right	n/a	female	39	7

simulate an action that can be performed in multiple locations. Consequently, we can evaluate the robustness of our method. The smoking areas are outdoor places; therefore, outdoor noise such as that from automobiles is recorded by the microphone in the smartwatch. In addition, if a participant washed the hands, dishes, and faces, the microphone also captured noise from running water.

Table 3.2 lists the physical features of the participants. Each participant performed multiple data collection sessions, with each session comprising a random sequence of activities. The sampling rates of the accelerometer and microphone on the smartwatch were 30 Hz and 44.1 kHz, respectively. Additionally, we collected camera images to obtain ground truth. The duration of each session was approximately 12 min.

3.4.2 Evaluation Methodology

To evaluate the proposed environment-independent location classifier, we performed our evaluation using “leave-one-environment-out” cross-validation, where sensor data from one environment were used as test data, and sensor data from the remaining en-

vironments were used as training data. We predicted the location classes for each test session. We prepared the following methods to evaluate the effectiveness of the proposed method.

- **IndoLabel**: This is our proposed method that uses acceleration data and impulse responses collected using active probing.
- **RF-ACC**: This method only uses acceleration data and is designed based on Tachikawa et al. [101] that assumed location-specific features are always observed at a location of interest. Therefore, this method assumes a sliding time window and extracts sensor data features in that window. Thereafter, this method forms a feature vector that concatenates the feature values that is input to the random forest (RF) classifier that classifies a feature vector into a location class. The extracted features are the minimum, maximum, mean, standard deviation, etc. by the tsfresh library (v 0.12.0).
- **RF-MIC**: This method only uses impulse responses. The procedures of this method are identical to those of RF-ACC.
- **RF-C**: This method is an improved version of RF-ACC and RF-MIC. It uses acceleration data and impulse responses. It initially clusters Wi-Fi scans in the same manner as the IndoLabel method. Thereafter, it aggregates the location class estimation results for time windows in each location cluster by majority vote to determine the location class of the cluster.
- **Tachikawa et al.**: This method is also designed based on the method proposed by Tachikawa et al. [101]. This method uses the acceleration data and audio impulse responses, as in RF-C method. Instead of the random forest used in RF-C, this method employs an environment-independent decision tree classifier proposed in [101]. A standard decision tree classifier only uses the class labels of the data segments to calculate the information gain and creates splits in the dataset. Alternatively, the method of Tachikawa et al. subtracts the information gain calculated based on labels of environments where the data segments were collected, from the information gain calculated based on the class of the data

segments, and uses the modified information gain to create splits in the decision tree.

- **Only-ACC**: This is a variant of the proposed method that only uses acceleration data.

- **Only-MIC**: This is a variant of the proposed method that only uses impulse response.

- **DANN**: This is a variant of the proposed method that does not use the loss function for distributions of output probabilities in the motif classifier (i.e., does not use $\mathcal{L}_{KL}$ in Equation 3.1).

- **CNN**: This is a variant of the proposed method that does not use domain-adversarial training and the loss function for distributions of output probabilities in the motif classifier (i.e., does not use $\mathcal{L}_{KL}$ and $\mathcal{L}_{d}$ in Equation 3.1).

We cannot compare our method to the method of Elhamshary et al. [29] because they exclusively recognize locations of the train stations with specific sensor data templates tailored for each location. Therefore, we cannot adapt this method for indoor location class prediction in household environments.

The classification accuracy of each method was evaluated using the micro-averaged F-measure that was calculated based on the classification results of location clusters from each session. For the RF-ACC and RF-MIC methods, these metrics were calculated based on the classification results of the time windows.

Herein we introduce the network parameters used in our motif classifier that were empirically selected. The first and second convolutional layers in the feature extractor have six and sixteen kernels using the ReLU activation function, respectively, with a kernel size of two and stride length of two. The fully connected layers in the motif label predictor have 120, 84, and 8 nodes, respectively, with the ReLU activation function. The fully connected layers in the environment label predictor have 120, 84 and, 2 nodes, respectively, with the ReLU activation function.

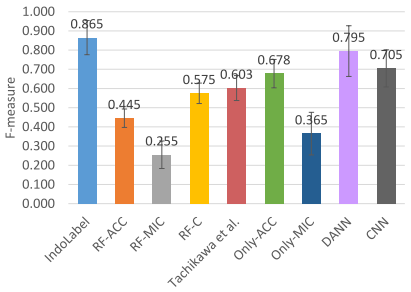


Figure 3.10: Average F-measures of the methods

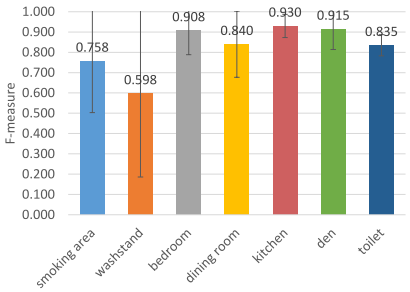


Figure 3.11: Average F-measures of IndoLabel for each location class

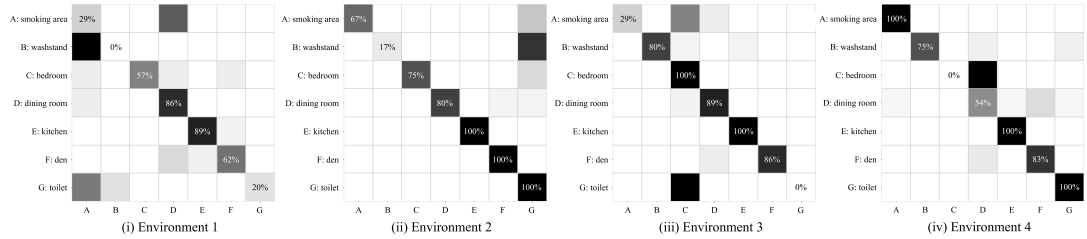
3.4.3 Results

Classification Accuracy

Figure 3.10 shows the average F-measure of IndoLabel. Although IndoLabel does not use training data collected in a test environment, it achieved approximately 87% F-measure on average. Figure 3.11 shows the F-measure of IndoLabel for each location class. Figure 3.12 (c) shows the visual confusion matrices for IndoLabel. From the results, the proposed method accurately estimated the location classes of the test instances using location-specific acceleration and sound motifs in several cases, although we do not use labeled training data collected in a test environment.

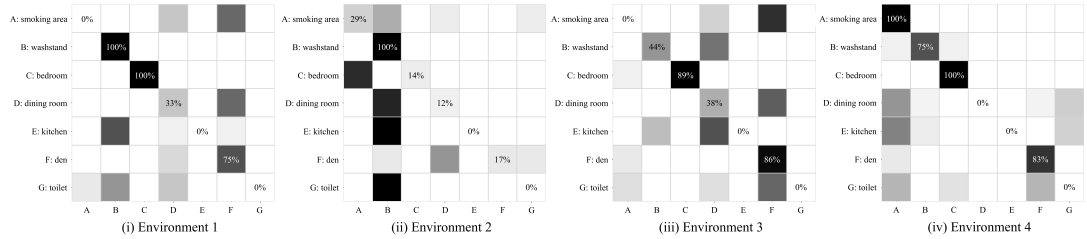
The accuracy for the washstand class was poor compared to that of the other classes, as shown in Figure 3.11. From Figure 3.12 (c), several washstand instances were incorrectly classified into the toilet or bedroom class. The toilet instances are similar to the washstand instances because the participants performed the washing hand actions in both locations. In addition, the ways of using toothbrushes were different between the participants. This can be a reason why the washstand instances were incorrectly classified into the bedroom class that does not have distinguishing hand motions in several cases.

The accuracy of the smoking area class was also somewhat poor compared to that of the other classes. From Figure 3.12 (c), several smoking area instances were incorrectly classified as the dining room class because we could not identify location-specific ac-



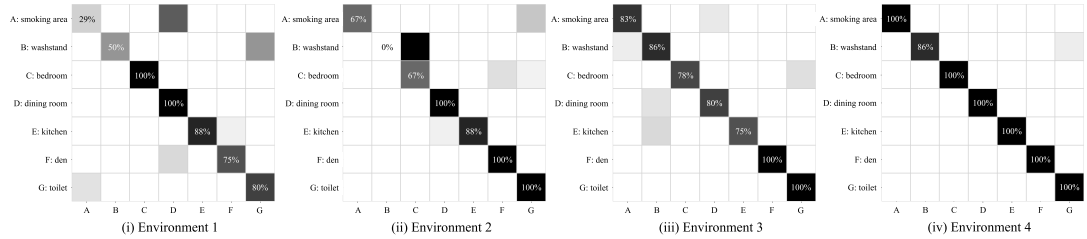
(i) Environment 1 (ii) Environment 2 (iii) Environment 3 (iv) Environment 4

(a) Visual confusion matrices of the Only-ACC method



(i) Environment 1 (ii) Environment 2 (iii) Environment 3 (iv) Environment 4

(b) Visual confusion matrices of the Only-MIC method



(i) Environment 1 (ii) Environment 2 (iii) Environment 3 (iv) Environment 4

(c) Visual confusion matrices of the IndoLabel method

Figure 3.12: Confusion matrices of classification results of Only-ACC, Only-MIC and IndoLabel

tions and impulse responses in the smoking areas. From Figure 3.4 (c) and (d), obtaining significant sound features in outdoor places was difficult because of ambient noise. In addition, the hand movements of smoking actions were insignificant and considerably similar to eating actions. Therefore, predicting location labels of smoking areas was difficult in some environments.

Comparison with Prior Methods

Figure 3.10 also shows average F-measures of the RF-ACC, RF-MIC, RF-C, and Tachikawa et al. methods. RF-ACC, RF-MIC, and Tachikawa et al. methods are based on the idea of the prior approach (i.e., location-specific features are always observed) and estimate

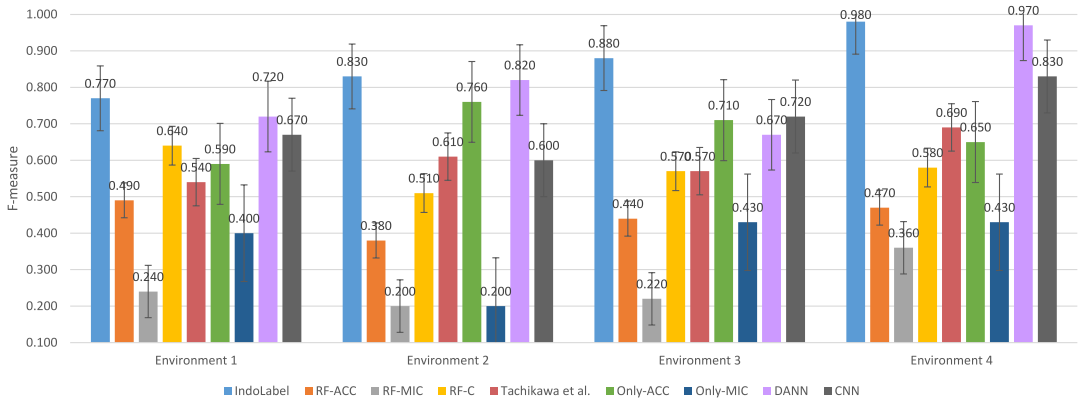


Figure 3.13: Classification F-measures in each environment. Each whisker shows the standard deviation.

location class labels for each time window. The RF-C method that aggregates the results of RF-ACC and RF-MIC for each location cluster obtained from Wi-Fi scans, outperformed RF-ACC and RF-MIC. Tachikawa et al. outperformed RF-C because Tachikawa et al. extracted environment-independent features using a modified decision tree. However, as indicated by the results, the performances of these methods were considerably poor compared to that of IndoLabel. While the participants performed location-specific actions in each location, these methods could not detect these locations because the durations of the actions were significantly short. Moreover, the participants performed the same activities in different rooms, e.g., “wash hands” in the toilet, kitchen, and washstand, resulting in increasing the difficulty of precisely predicting location classes. These results indicate the effectiveness of our approach for predicting location classes based on short sensor data motifs. Specifically, the F-measure of IndoLabel is higher than that of Tachikawa et al. by 30%, indicating that the usability and precision of applications implemented based on IndoLabel will significantly improve compared to the prior studies.

Classification Accuracy in Each Environment

Figure 3.13 shows the average F-measures of the methods for each environment. From the results, our method achieves significant F-measures in all environments. The performance in Environment 1 was marginally poor compared to that of other environments because the accuracies for the washstand and smoking area classes were somewhat poor in this environment. In particular, several instances of the smoking area were misclassified into the dining room class, as shown in Figure 3.12 (c) because the smoking actions are similar to the eating and drinking actions. The accuracies for the washstand class in Environments 1 and 2 were poor because the ways of using toothbrushes by the participants were different from the other participants.

Here, we focus on the relationship between the number of detected APs and the accuracy of each environment. The F-measures of Environments 1, 2, 3, and 4 were 77%, 83%, 88%, and 98% (Figure 3.13), respectively. The average number of detected APs of those environments were 35, 57, 35, and 36, respectively (See also 3.4.1). As can be seen in the results, there is no clear correlation between the number of detected APs and the prediction accuracy.

Sensor Contributions

Figure 3.10 also shows the F-measures obtained by the Only-ACC and Only-MIC methods. From the results, the accelerometer was the best contributor because the Only-ACC method achieved an F-measure value of approximately 68%. Figures 3.12 (a) and (b) show visual confusion matrices for the Only-ACC and Only-MIC methods. From Figure 3.12 (a), the Only-ACC method achieved good accuracy for the dining room, kitchen, and den classes because these classes have characteristic hand motions. From Figure 3.12 (b), the Only-MIC method achieved relatively significant accuracy in estimating the location classes of the bedroom and washstand instances. When the participants were lying on beds, we observed characteristic impulse responses that contained information about blankets. Moreover, the washstands are considerably smaller than the other rooms; therefore, we believe that the impulse responses observed in these rooms contain information about the small-sized rooms. Interestingly, the Only-MIC method successfully recognized the washstand instances although the restrooms were

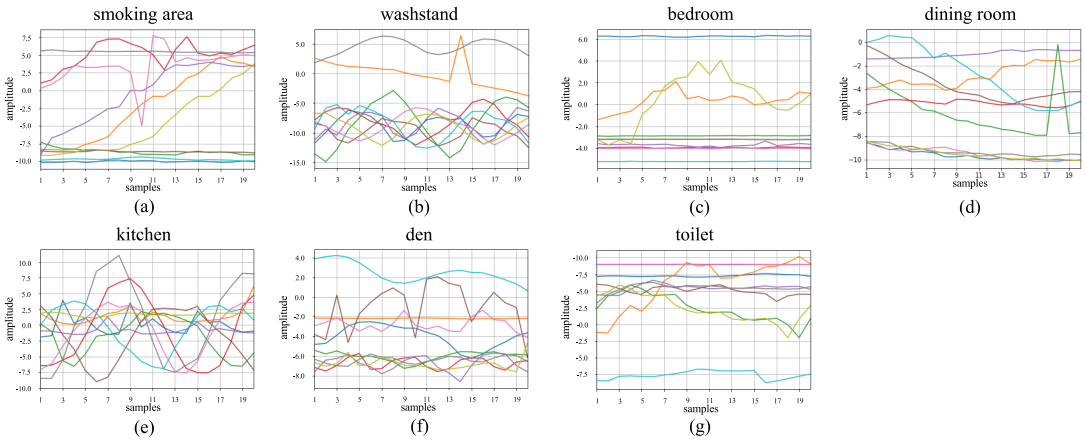
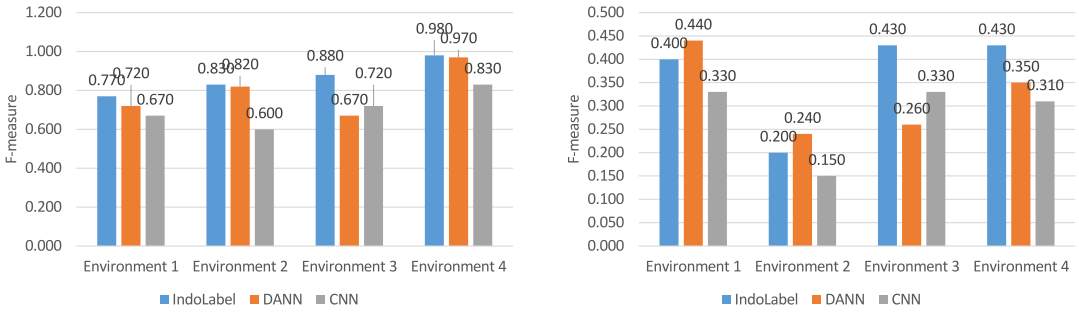


Figure 3.14: Examples of acceleration data motifs (after PCA) extracted from each location class in an environment. The horizontal axis shows the time-axis and vertical axis shows the amplitude.

also small, as shown in Figure 2.11. This can be because the Only-MIC could capture acoustic information related to objects in the room, e.g., the sink.

From Figure 3.12 (a) all the bedroom instances of Environment 4 have been misclassified into the dining room classes when using the Only-ACC approach. In contrast, Only-MIC achieved 100% accurate recognition in the bedroom class for Environment 1 as shown in 3.12 (b). This can be because it easily captured location-specific acoustic characteristics while the hand of the user showed minimum movement in sleep.

Figure 3.14 shows examples of location-specific motifs extracted from acceleration data (after applying PCA) by our method. From the figure, the motifs extracted from the smoking area, washstand, dining room, kitchen, and den show visually identifiable and distinguishable waveforms. For example, the motifs from the washstand class contain high-frequency waveforms that can correspond to the tooth brushing action. Furthermore, the motifs from the smoking area and dining room contain waveforms that correspond to the hand motions of moving the cigarette to the mouth while smoking and moving the spoon from the mouth to the plate while eating, respectively. However, most motifs extracted from the bedroom do not show high amplitudes because the sleeping activity does not necessarily create hand movements. This can make it difficult for our



(a) Classification F-measures of different motif classifiers in different environments when using acceleration data

(b) Classification F-measures of different motif classifiers in different environments when using impulse responses

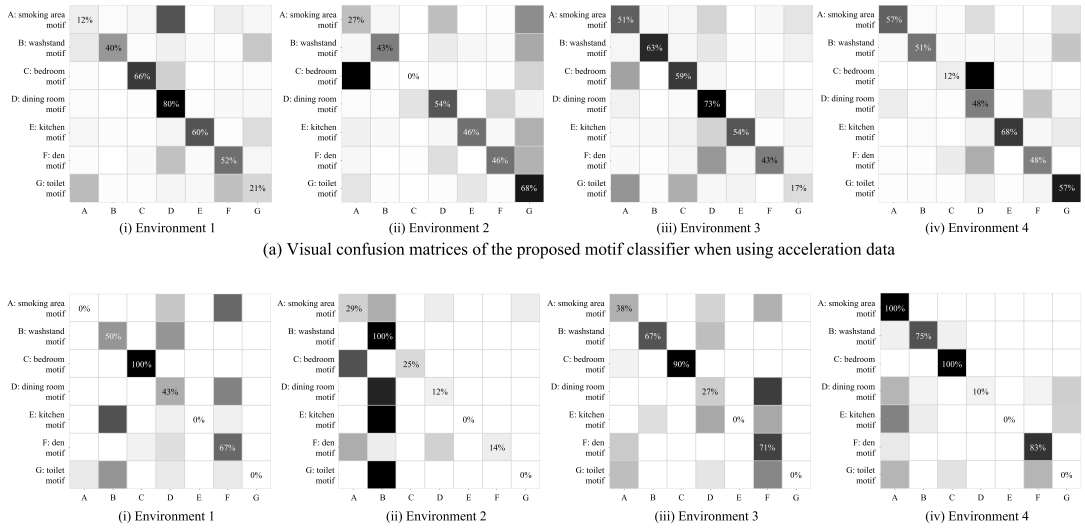
Figure 3.15: F-measures of different motif classifiers

motif classifier to accurately recognize such locations, resulting in poor accuracy of the location classifier. Note that because we can obtain distinguishing information from the impulse responses in the bedroom, IndoLabel could accurately recognize the class. As mentioned above, we confirmed that IndoLabel precisely predicts location classes by fusing acceleration and acoustic motifs.

Contributions of Motif Classifier

Figure 3.10 also shows the F-measures of the CNN and DANN methods. From the results, IndoLabel outperforms DANN by approximately 7%. Particularly, the accuracies for the smoking area and toilet classes were improved by introducing $\mathcal{L}_{KL}$ (the loss function for the distributions of output probabilities). Because these classes have several hand motions similar to other classes, IndoLabel recognized these classes by leveraging the hand motions (see below in detail). In addition, as shown in Figure 3.10, DANN outperformed CNN by approximately 9% by introducing the domain-adversarial training. Specifically, the accuracies for the bedroom and toilet classes were improved. This can be because the original acoustic features in the bedrooms and toilet rooms contain environment-dependent components.

Figure 3.15 shows the classification results of the motif classifier of IndoLabel, DANN, and CNN for acceleration data and audio impulse responses. In many cases, the methods based on the domain-adversarial training, i.e., IndoLabel and DANN, outper-



(i) Environment 1 (ii) Environment 2 (iii) Environment 3 (iv) Environment 4

(a) Visual confusion matrices of the proposed motif classifier when using acceleration data

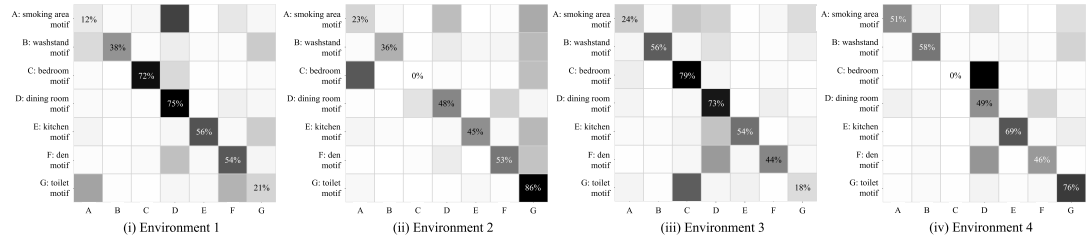
(i) Environment 1 (ii) Environment 2 (iii) Environment 3 (iv) Environment 4

(b) Visual confusion matrices of the proposed motif classifier when using impulse responses

Figure 3.16: Confusion matrices of classification results of the motif classifiers for IndoLabel

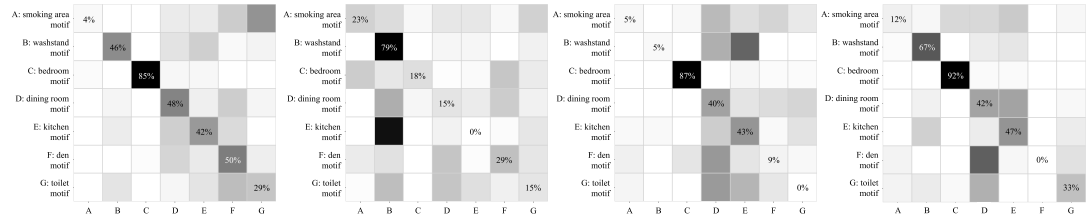
form CNN. Particularly, the accuracies of IndoLabel and DANN for impulse responses are much higher than the accuracy of the CNN. This result indicates that discovering latent acoustic features is pivotal in the location class prediction. Because the room sizes and floor materials are considerably different in each environment, it is difficult to precisely predict location classes using raw impulse responses.

Figures 3.16, 3.17, and 3.18 show the confusion matrices of classification results of the motif classifiers for IndoLabel, DANN, and CNN, respectively. The confusion matrices do not show the results of the "others" class. The accuracies of DANN for the bedroom, kitchen, and dining classes are higher than those of CNN, as shown in the confusion matrices for impulse responses in Figures 3.17 and 3.18. This indicates that domain-dependent components are dominant in these places. Interestingly, the performance of IndoLabel for the toilet class was poor compared to than that of DANN in the confusion matrices of impulse responses, whereas the location classification accuracy of IndoLabel for the toilet class was higher than that of the DANN. From the confusion matrix of the impulse responses in Figure 3.16, some instances belonging to the toilet class were inaccurately classified into the washstand or den class in several environ-



(i) Environment 1 (ii) Environment 2 (iii) Environment 3 (iv) Environment 4

(a) Visual confusion matrices of the DANN motif classifier when using acceleration data



(a) Environment 1 (b) Environment 2 (c) Environment 3 (d) Environment 4

(b) Visual confusion matrices of the DANN motif classifier when using impulse responses

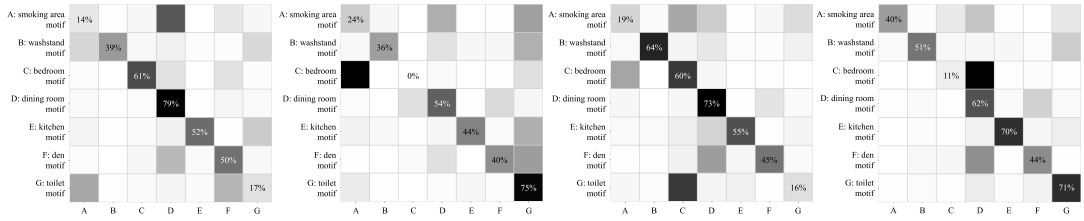
Figure 3.17: Confusion matrices of classification results of the motif classifiers for DANN

ments. This means that the distributions of output probabilities for the toilet instances in the environments are similar to each other. In other words, an averaged output probability vector for the toilet class has high probability values for the washstand and den classes. Therefore, the location classifier could recognize the toilet class by leveraging this feature of the toilet instances. In contrast, as shown in the confusion matrix of the impulse response for DANN in Figure 3.17, classified classes of the toilet instances in the environments differ in different environments. Our proposed neural network (motif classifier) enables accurate recognition of location classes that do not have distinguishing motifs by leveraging motifs for other location classes.

3.4.4 Discussion

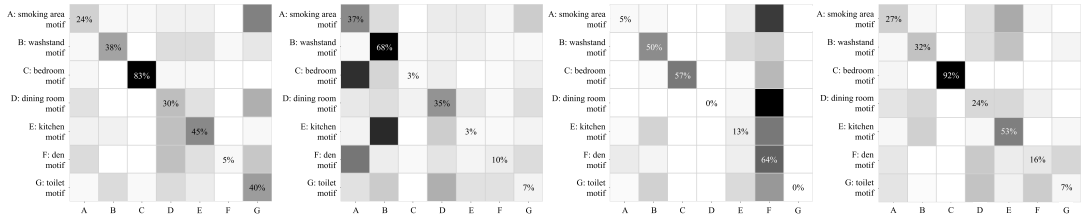
Amount of Training Data

Herein, we discuss the relationship between the amount of training data (i.e., the number of training environments) and classification performance in a test environment. Figure



(a) Environment 1 (b) Environment 2 (c) Environment 3 (d) Environment 4

(a) Visual confusion matrices of the CNN motif classifier when using acceleration data



(a) Environment 1 (b) Environment 2 (c) Environment 3 (d) Environment 4

(b) Visual confusion matrices of the CNN motif classifier when using impulse responses

Figure 3.18: Confusion matrices of classification results of the motif classifiers for CNN

3.19 shows the transition of average F-measures for the methods when we changed the number of training environments. Note that we have different numbers of sessions collected from each environment (Table 3.2). To make the number of training instances consistent, we randomly deleted the excessive number of instances from those environments to match the lowest number of instances among the environments. We used Environment 4 as the target environment. From the results, even though we increased the amount of training data, the F-measure values for RF-C did not improve, indicating the difficulties in capturing location-specific features using this approach. When the number of training environments is two, IndoLabel yields the upper bound of the accuracy.

Energy Consumption

Because we assumed that the accelerometer, speaker, and microphone on a smartwatch were always on, the smartwatch could function for up to approximately 3 h. To reduce the energy consumption of the smartwatch, we can reduce the total executions of sound probing because impulse responses obtained during hand movements are not used in

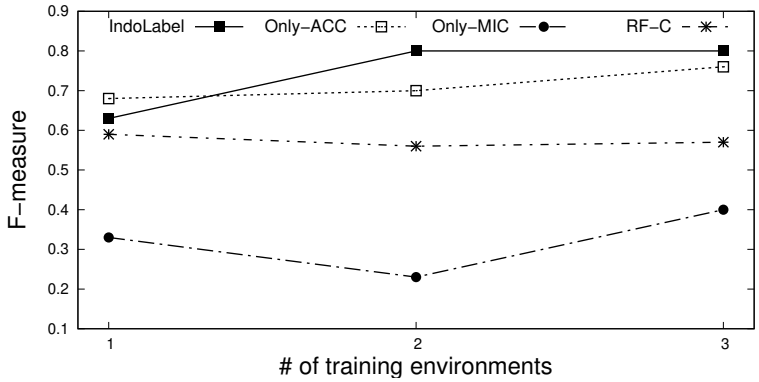


Figure 3.19: Transitions of average F-measures when the number of training sessions is varied

our method. (When only the accelerometer is turned on, the smartwatch functions up to approximately 4.5 h.) Moreover, our smartphone application transmits observed data in real-time, resulting in energy consumption related to wireless transmission. Because our method does not require real-time data, we can delay transmission to save energy (i.e., transmit stored data only once during charging). Furthermore, once the location class of a Wi-Fi cluster is determined, the acceleration and sound data are not needed because we can estimate the user's current location using a Wi-Fi scan.

Generalizability of the Method

We have conducted our experiments using four different participants and confirmed that our method is generalizable across different sexes, ages, and body heights (Table 3.2). However, all the participants were right-handed. To overcome the problems regarding the difference in the dominant hands, we could define a reference coordinate system for the smartwatch on the right-hand [51] and use that to transfer the acceleration data into the corresponding coordinate system for a smartwatch on the left hand. However, the activities performed by different users can further be inconsistent due to their physical conditions such as fitness level and injuries. We believe that our method can further be generalized across such personal characteristics using an online personalization scheme of the recognition model as discussed in [100]. Tachikawa et al. [101] have investigated

the possibility of cross-device location class prediction using two different smartphones. They have discovered that the accuracy of the location class prediction significantly degraded in the case of cross-device recognition. This could be because the sensitivity and the frequency response of the microphones are different between the different devices. Even though this means that device-dependent training is required for our model, Thachikawa et al. also described a method to eliminate the device-dependency by calculating a transformation matrix per device via a calibration phase, where the user is required to record a predefined sound signal for calibration. This transformation matrix can then be used to transform sensor data collected by one device to those by another device.

3.5 Conclusion

We presented a method for predicting the location class of a room where a user was located by discovering location-specific sensor data motifs in sensor data from the user's smartwatch. Additionally, we proposed a novel matrix manipulation that automatically detected location-specific motifs from time series sensor data by calculating a score that represented the location specificity of each segment in the time series based on the concept of the Gini impurity. To the best of our knowledge, this is the first study to introduce the concept of Gini impurity in a method that scans time series data to identify location-specific templates (i.e., motifs). In addition, our method can detect location-specific motifs by eliminating environment-dependent components. We evaluated the proposed method in real house environments and achieved state-of-the-art performance. As part of our future work, we intend to evaluate our method in working environments such as hospitals and factories.

Chapter 4

Joint Estimation of the Distance and Relative Velocity of Obstacles via Smartphone Active Sound Sensing for Pedestrian Safety

4.1 Introduction

4.1.1 Background

Due to the recent proliferation of smartphones and related applications, the number of pedestrians that use such devices for activities other than voice calls while walking has increased, and uses such as browsing social media, sending and receiving text messages, watching videos, playing games, and navigation are common. These pedestrians tend to concentrate on their smartphones rather than the road, and may collide with various roadside obstacles, which could lead to serious injuries and even death. As an example, Nasar et al. [76] showed that 1506 injuries were estimated to have occurred in the US in 2010 due to distracted pedestrians using their smartphones. Furthermore, the American Academy of Orthopaedic Surgeons (2015) found that 26% of respondents to a survey of 2000 adults had experienced accidents due to using their smartphones while walking, ranging from bumping into something without injury, to falls, sprains or fractures [58].

In the field of pervasive computing, studies have been actively conducted on detecting oncoming obstacles using data from different onboard smartphone sensor modalities. The footage from the rear-camera of a smartphone has been used to recognize obstacles in front of the user [117, 14, 82, 107, 85, 44]. However, this method involves some inherent problems, such as limited detection range due to the orientation of the device in use, poor performance in dark environments, and privacy issues. To address these problems, more recent works on obstacle detection have focused on methods based on acoustic sensing. Some previous studies have aimed to detect the presence of obstacles in front of the user with which they may collide [124, 109, 120, 111]. Recent works have not only detected obstacles, but also estimated other properties, such as their types [126], distance from the user [120, 111], and relative velocity [126]. In this study, we also focused on estimating properties of obstacles, particularly the distance and relative velocity, because this information is crucial to estimate the possible timing of collisions and the level of risk of a given obstacle (e.g., high-speed obstacles exhibit higher risk), which is used to issues warnings to user.

4.1.2 Problems

Existing methods designed to estimate properties (distance and relative velocity) of obstacles involve some limitations, including i) limitations in types of obstacles and ii) limited types of properties that can be estimated. As for the first limitation, many previous studies have focused only on static or slow-moving obstacles, such as walls, dustbins, donation boxes, and signs. However, sidewalks often include numerous mobile obstacles, such as pedestrians and bicycles, which can be considered as high-risk obstacles. As for the second limitation, most of the existing methods have focused only on either the distance or relative velocity, although both are crucial information for pedestrian safety, as mentioned above.

These limitations are attributed to the types of probing signals used in these studies. Each of these works relied on either sweep or sinusoidal signals. Although the short pulses of sweep signals have been widely used to estimate the distance between the user and a static obstacle by analyzing reflected sounds [73, 120, 84, 72, 103], estimating the velocity of mobile obstacles with sweep signals is difficult. Some studies have attempted

to detect mobile obstacles and estimate their velocity using time-series analysis of sweep signals, i.e., by tracking the distance to an obstacle within each time window [41, 126]. However, to track the mobility of high-speed obstacles with high granularity, the number of sweeps generated by the smartphone per second should be increased. Due to the inherent characteristics of commercial speakers, when a large number of sweeps are generated per second, the amplitude of the sweeps falls, resulting in the limited sensing range [126].

By contrast, continuous sinusoidal signals (sine waves) have been employed to capture the Doppler shifts created by mobile obstacles to estimate their velocity [135, 60]. However, this approach is not suitable to estimate the distance to an obstacle. In addition, estimating the relative velocity of static obstacles with this approach is difficult because they do not create characteristic Doppler shifts.

As noted above, to the best of our knowledge, no study has attempted to estimate both the distance and velocity of both static and mobile obstacles in a single framework.

4.1.3 Approach

To overcome the abovementioned limitations, we proposed a new method named **ObsSense**, which is designed to estimate the distance and relative velocity of both static and mobile obstacles using an off-the-shelf-smartphone. The main features of ObsSense include i) novel inaudible probing sound signals composed of sweep signals and sine waves, which enable joint estimation of the distance and velocity of both static and mobile obstacles, and ii) a novel neural network architecture that efficiently fuses features of the reflected sounds of the probing signal to estimate both the distance and velocity of a given obstacle.

As for the first feature, the probing signal was designed such that i) the signal composed of sine wave and sine sweep is continuous in the frequency domain, which prevents audible sound noises in the signal due to signal power leakage caused by discrete frequency generation by the speaker [129], and ii) our probing signal maintains a constant volume over both sweep and sine wave components, which facilitates capturing distant obstacles.

As for the second, the most appropriate strategy to estimate distance and velocity

depends on the movement status of an obstacle, i.e., whether it is static or mobile. For example, reflected sine-wave signals, which contain information about Doppler shift, are more useful in predicting the velocity of a mobile obstacle. By contrast, reflected sweep signals are more effective in predicting the velocity of a static obstacle. Therefore, our proposed neural network is equipped with a motion detection module that identifies the movement status of an obstacle as static or mobile. By leveraging the intermediate outputs of the module, we accelerate the training of the neural network so that it learns an appropriate strategy to estimate the distance and velocity of obstacles depending on their movement status. In addition, our network is designed to estimate the distance and velocity by fusing reflected signals of sweep signals and sine waves. For example, reflected signals of sweep signals are useful to predict the distance to an obstacle while they are sparse and noisy. Our network is designed to leverage reflected signals of sine waves, i.e., Doppler shift, as a supplement to predict the distance because the distance at time t can also be estimated based on the distance at time $t-1$ and the velocity at that time, i.e., using integration, which enables us to improve the distance prediction performance.

4.1.4 Contributions

i) This study is the first to estimate the distance and relative velocity of both static and mobile obstacles using active acoustic sensing on a smartphone. ii) We proposed a novel probing signal that facilitates both acoustic ranging and velocity estimation. iii) We proposed a novel neural network architecture designed to estimate the mobility of the detected obstacles and used this knowledge to effectively fuse sound features to estimate their distance and relative velocity.

4.2 Related Work

4.2.1 Static Obstacle Detection and Distance Estimation

Low-cost ultrasonic sensors have been employed to detect and estimate the distance to obstacles [105, 124, 104]. However, these methods require additional attachments

to the smartphone or special devices. Recently, active sound sensing has been widely adapted to perform static obstacle detection and distance estimation on smartphone devices. Some studies have detected obstacles with a possibility of collision by employing the camera as well as active sound sensing [111] and active sound sensing with two onboard microphones [120]. In addition, the distance between a smartphone user and a static obstacle was estimated using sound probes. Specifically, Tung et al. [111] used a smartphone speaker to emit short pulses of sine waves and record the reflections with the microphone. They then calculated the impulse response of the reflections to estimate the distance to obstacles. This was performed by accurately calculating the traveling time of the emitted signal between the user and the obstacle using a threshold-based method to detect the peaks created by the obstacle in the calculated impulse response. More recently, Wang et al. [120] used a smartphone to emit inaudible sweep signals and estimate the distance to obstacles in front of a user by calculating the correlation between the emitted and received signals. Similarly, they employed a threshold-based method to detect the peaks in the impulse response. By contrast, our study focused on *both* the distance and velocity. Furthermore, rather than using a threshold-based method, we utilized a neural network to extract features related to the distance from the calculated impulse responses.

4.2.2 Mobile Obstacle Detection and Velocity Estimation

Passive sensing with smartphones has been employed to detect mobile obstacles, such as approaching vehicles [102, 57, 45]. Some recent works have also aimed to estimate both the distance and relative velocity of the obstacles at the same time [42] by employing acoustic signals emitted by vehicle-mounted speakers. However, these methods only work for moving vehicles that actively emit characteristic noise, and are not suitable to detect other mobile obstacles or static obstacles.

In the field of active sound sensing via smartphones, acoustic ranging techniques using excitement signals, such as sweep signals, have been adapted to detect and estimate the velocity of mobile obstacles. Wu et al. [126] emitted short sweep signals from a smartphone and captured the reflections of vehicles. They estimated the speed of vehicles by capturing the change in the distance between the user and the vehicle over

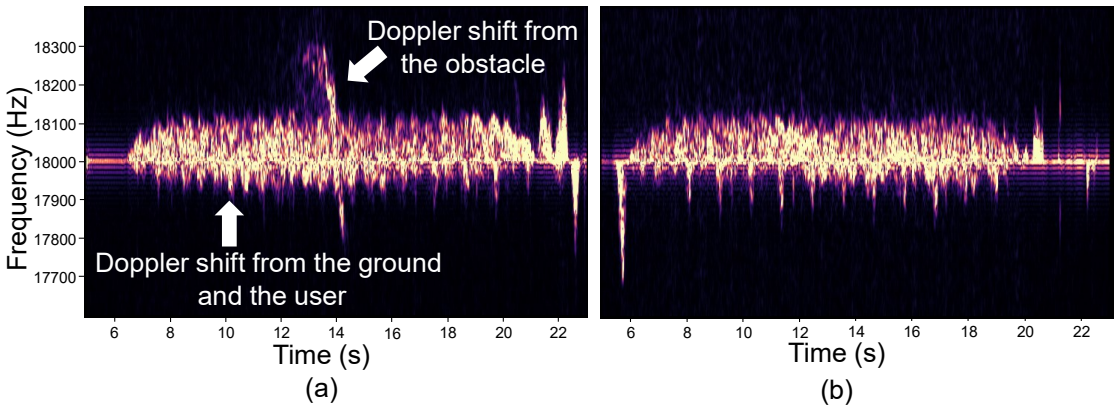


Figure 4.1: FFT spectrograms of reflected sounds of sine waves recorded when (a) a smartphone user was walking toward a person while the person was also walking toward the user and (b) a user was walking toward a stationary person.

time. Jin et al. [41] used a COTS speaker connected to a smartphone to detect right turns of vehicles and alert cyclists prior to potential accidents. They emitted periodic sweeps from the speaker and analyzed time-series reflections to detect approaching vehicles. By contrast, we designed a new probing signal composed of sweep signals and sine waves, and fuses them to estimate the relative velocity.

Sine waves have also been widely used to capture Doppler shifts created by the movements of obstacles and to estimate their velocities. Zhang et al. [135] utilized an inaudible sine wave emitted by a smartphone to capture the Doppler shifts created by an approaching person. They extracted Doppler profiles to identify trajectories of people approaching each other. Similarly, Liu et al. [60] proposed a method to detect indoor pedestrian encounters and estimate their velocities using a sine wave. However, this method is not suitable to detect static obstacles, as they do not produce characteristic Doppler shifts. By contrast, our proposed method was designed to estimate the velocity of both static and mobile obstacles.

4.3 Investigation and Probing Signal Design

ObsSense was designed to estimate the distance and relative velocity of both static and mobile obstacles. To this end, we investigated the characteristics of the different probing

signals that can be used to estimate different properties of static and mobile obstacles. Considering the outcomes of our investigations, we proposed a novel probing signal design that facilitates both distance and velocity estimation.

4.3.1 Obstacle Mobility Detection

Obstacles on the sidewalk can be divided into two major groups according to their mobility; static obstacles such as walls, signposts, and parked vehicles, and mobile obstacles such as walking pedestrians and bicycles. As mentioned in the introduction, the proposed neural network was designed to detect the mobility of an obstacle. To do so, we can employ the well-known Doppler shift. Figure 4.1 shows the Doppler shift created on an 18 kHz sine wave during two different encounters with obstacles. Here, a smartphone user walked towards two different obstacles while the smartphone emitted an 18 kHz sine wave. Simultaneously, the top microphone of the device recorded the reflected waves. Figure 4.1 (a) shows the characteristic Doppler shift created by a pedestrian (obstacle) who walked towards a user. However, when the user walks towards a stationary person (Figure 4.1 (b)), a characteristic Doppler shift cannot be observed. This is because the Doppler shifts created by a static obstacle are not clearly visible against the Doppler shifts created by the moving body parts of the user and the ground. These characteristic Doppler shifts can be used to differentiate between mobile and static obstacles.

4.3.2 Obstacle Distance Estimation

To estimate the distance of obstacles, we can employ a short excitement signal emitted by the smartphone's speaker, known as a sweep signal. In the proposed approach, sweep signals are emitted from the speaker periodically and the reflected signals are captured. In this experiment, the length of the sweep length was 0.05 sec and the duration between the sweeps was 0.25 sec. Next, we calculated the impulse response of the reflected waves and extracted their signal envelopes (IR envelopes; [120, 41], Section 4.4.3 in detail). These IR envelopes contain the intensity of reflections from an obstacle for each distance.

Figure 4.2 (a) shows the reflections from a wall recorded on the IR envelopes when

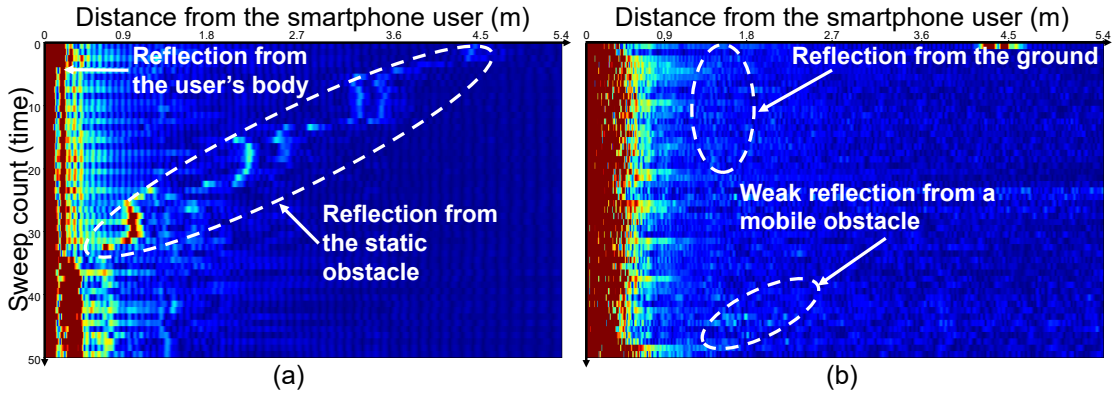


Figure 4.2: (a) Reflections of a static obstacle (a wall) and (b) reflections of a mobile obstacle (a jogging person), recorded on IR envelopes. Each IR envelope is 6000 data points long (x-axis). Considering the speed of sound as 340 m/s and sampling rate as 192 kHz, the distance resolution of each IR envelope sample corresponds to $\frac{340\text{m/s}}{192\text{kHz}} \approx 0.18\text{cm}$. Considering the round-trip time, the maximum distance from which the reflections can be captured is $(0.18\text{cm} \times 6000)/2 = 540\text{cm} = 5.4\text{m}$

the user walks towards a wall. This image was created by time-series stacking of 50 separate IR envelopes. As may be observed, when the user moved towards the obstacle, the reflection appeared closer. This information can be used to estimate the distance to obstacles. However, mobile obstacles with high speeds do not produce reflections with high granularity, as the number of emitted sweeps within a second is limited (Section 4.1.2). Figure 4.2 (b) shows the reflections from a jogging person recorded on the IR envelopes, which were measured with low granularity because the person was moving rapidly. Hence, we proposed a neural network architecture that incorporates both the IR envelopes and Doppler features to estimate the distance of obstacles, because we can observe clear Doppler shift of the mobile obstacle as shown in Figure 4.1 (a).

4.3.3 Obstacle Velocity Estimation

The frequency offset created by a mobile obstacle is directly proportional to its relative velocity with respect to the user. Figure 4.3 (a,b,c) shows the Doppler shifts created by a pedestrian walking toward the user at different speeds. As may be observed, the higher the velocity of the pedestrian, higher the frequency offset of the Doppler shift.

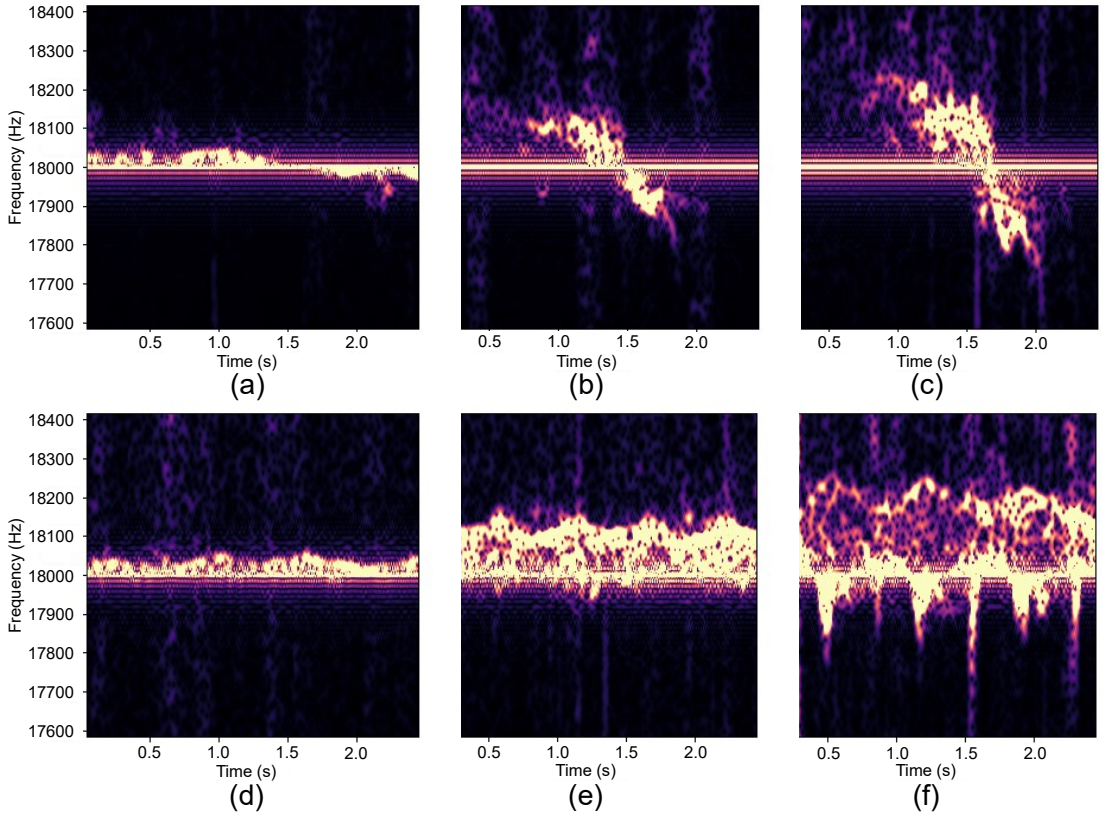


Figure 4.3: FFT spectrograms of reflected sounds of sine waves recorded when a pedestrian walked towards a static user at (a) slow, (b) medium, and (c) fast speeds and a user walking towards a wall at (d) slow, (e) medium, and (f) fast speeds.

However, when an obstacle is static, estimating its relative velocity is difficult, as shown in Figure 4.1 (b). Although the velocity of a static obstacle can be predicted from the time-series of the distance to the obstacle obtained by sweep signals, reflected sounds of the sweep signals are sparse and noisy, as shown in Figure 4.2. Here, note that the relative velocity of the static obstacle is equal to the velocity of the user, meaning that we can predict the relative velocity from the velocity of the user. As shown in Figure 4.1 (b), the reflected sounds contain the Doppler shifts created by the human body as well as the ground, and they can change depending on the velocity of the user. Figure 4.3 (d,e,f) shows the difference in Doppler shifts created by the user's body and the ground depending on their walking speed. The result indicates that reflected sounds of

sine waves can be used to predict the relative velocity of a static obstacle in addition to reflected sounds of sine sweeps.

4.3.4 Probing Signal Design

To implement ObsSense in real-world applications, the probing signal should satisfy the following requirements: i) As shown in the above investigation, to capture the distance and relative velocity of static and mobile obstacles, a probing signal composed of both sweep signal and sine wave components is required. ii) To estimate the distance and relative velocity of high-speed mobile obstacles, each signal component should either be emitted continuously or in rapid succession. iii) To alert the user prior to a collision, oncoming obstacles should be detected from as far away as possible. iv) The probing signal should be inaudible to the human ear. We considered frequencies above 18 kHz as meeting this requirement. However, we found in our preliminary investigations that the frequency response of the smartphone's microphone decreased with frequency (frequencies up to 21 kHz exhibited reliable frequency response). Therefore, using higher frequencies adversely affected the sensing range.

Figure 4.4 (a) shows an FFT spectrogram of reflected sounds of an ideal probing signal that emits a continuous sine and periodic sweep signals simultaneously in two separate frequency bands. The frequency of the sine wave was 22 kHz and sweep range was 18–21 kHz. Both components were separated by a 1 kHz frequency band to prevent Doppler shifts from the sine wave from overlapping with the sweep signals. However, this probing signal involves two major drawbacks. First, at the beginning and end of each sweep signal, the speaker emits a distinct and audible noise that has a frequency lower than 18 kHz (denoted by white arrows). Furthermore, these noises interfere with the emitted sine wave as well. This issue is related to commercial speakers, and is known as signal power leakage due to frequency-hopping, which is caused by emitting on discrete frequencies consecutively [129]. Second, as periodic sweeps are emitted simultaneously with the sine wave, the speaker power is divided between the two signal components. Hence, each component is emitted at half the maximum power of the speaker, resulting in the reduced maximum sensing range of our method. As may be observed in the rectangle outlines in white dots in Figure 4.4 (a), the sine wave segments

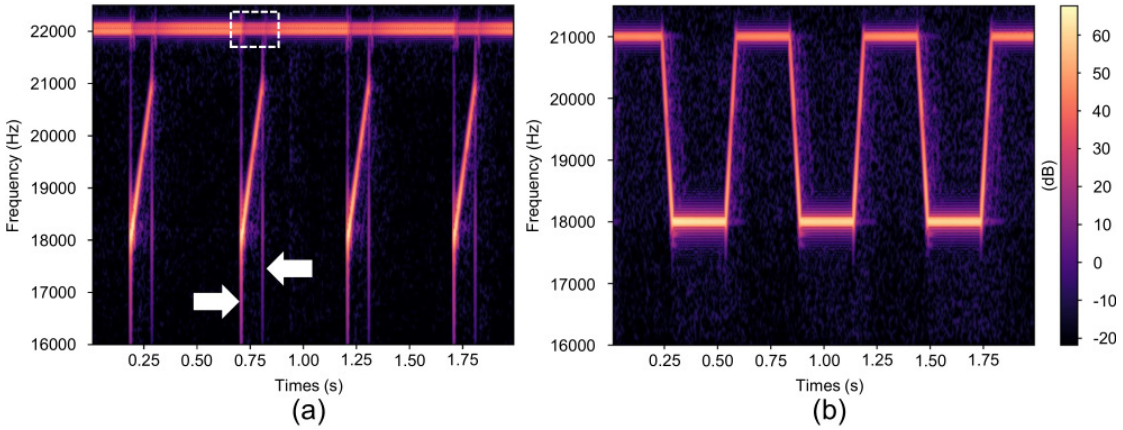


Figure 4.4: FFT spectrograms of two probing signal designs

that are being simultaneously emitted with the sweep signals exhibit significantly low amplitudes compared to the remainder of the signal.

To overcome these problems, we proposed a novel probing signal that emits a combination of sweep signals and sine waves in quick succession. Figure 4.4 (b) shows the FFT spectrogram of reflected sounds of the proposed probing signal. In this signal, every 18 kHz sine wave is followed by an 18–21 kHz sweep signal, followed by a 21 kHz sine wave, which is then followed by a 21 kHz–18 kHz inverse sweep. As may be observed, the proposed probing signal is continuous in the frequency domain, without consecutive discrete frequencies, which eliminates noises due to signal power leakage. Furthermore, this approach solves the speaker power division problem by emitting one signal at a time at full power, which increases the sensing range. In our implementation, each sine wave and sweep/inverse sweep were 0.25 and 0.05 sec long, respectively. The length of the sweep/inverse sweep was selected to minimize system detection delays and multi-path distortions [120]. Furthermore, the interval of sweep/inverse sweep signals (i.e., the length of the sine wave) was determined based on our preliminary experiments. When the number of sweeps emitted per second increased, the overall amplitude of the emitted sweeps decreased, resulting in limited sensing range. Therefore, we selected the interval that did not affect the amplitude of the sweeps. A similar interval between the sweep signals was also used in [41].

Because the probing signal emits sine waves between sweep signals, and ObsSense learns connections between the Doppler and IR features to jointly estimate the distance

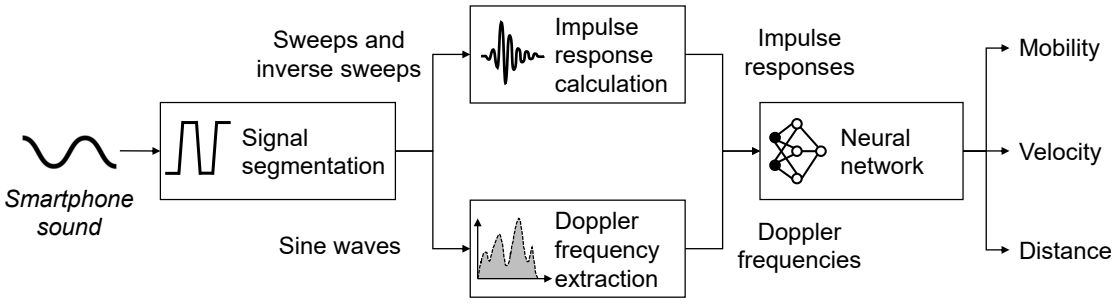


Figure 4.5: Overview of ObsSense

and velocity, our method can estimate the distance between the sweeps with high granularity (at 100 Hz in our implementation).

4.4 ObsSense Method

4.4.1 Overview

In this study, we assumed that the user holds the smartphone in their hand such that they can view the content of the screen of the phone clearly while walking on the sidewalk. A probing signal designed in the previous section is emitted from the top speaker of the smartphone and the top microphone records the reflected waves at a sampling frequency of 192 kHz.

Figure 4.5 shows an overview of the proposed approach. The recorded audio is first fed into the signal segmentation module, which separates the reflected signal into reflections of sine waves, sweeps, and inverse sweeps. Next, the reflections of sine waves are fed into the Doppler frequency extractor and the reflections of sweeps and inverse sweeps are fed into the impulse response calculator. The resulting impulse responses and Doppler frequency features are then fed into a specially tailored neural network that jointly estimates the distance and the relative velocity as well as the mobility of the obstacle. Note that the velocity can be calculated from the distance through derivation. However, because the information about the velocity is important in the distance prediction when sine sweeps are not emitted, this architecture, which explicitly predicts the velocity, can achieve precise distance prediction with high granularity. In addition, we assumed a module that detects an obstacle in front of the user with the possibility

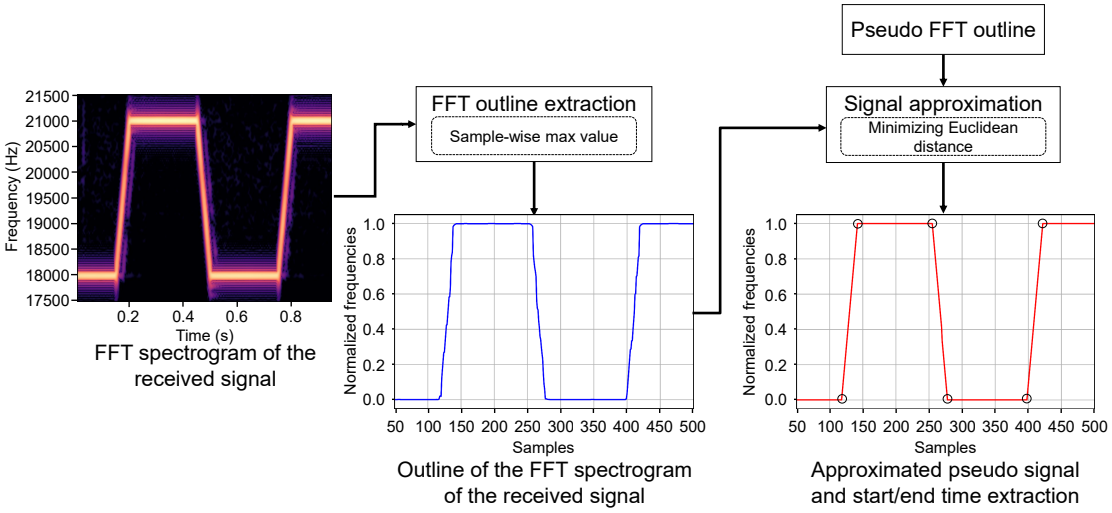


Figure 4.6: Pipeline of identifying start/end times of sine waves, sweeps, and inverse sweeps in the signal segmentation module

of collision, which has been extensively studied in prior works based on probing signals, such as sweep signals and sine waves [124, 109, 120, 111]. When an obstacle is detected, ObsSense predicts its attributes. We explain the modules of ObsSense in the following sub chapters.

4.4.2 Signal Segmentation

Here, we segmented the recorded reflected sound into sine waves, sweeps, and inverse sweeps because sound features are extracted from each of the segments in the following procedure. We extracted segments corresponding to reflections of sine waves, sweeps, and inverse sweeps from the FFT spectrogram of the received signal (as shown in Figure 4.4 (b)) calculated using a Hanning function on a 90% overlapping sliding window with 16,384 samples. Ideally, the smartphone can record the starting and ending times of the sine waves, sweeps, and inverse sweeps because the smartphone generates and emits the probing signals. However, due to several factors, such as noise and delays in sound emission and reflection, the actual starting and ending times of the reflections differed slightly from the times recorded by the smartphone. To acquire exact starting and ending times, we generated a pseudo-FFT spectrogram of the probing signal by

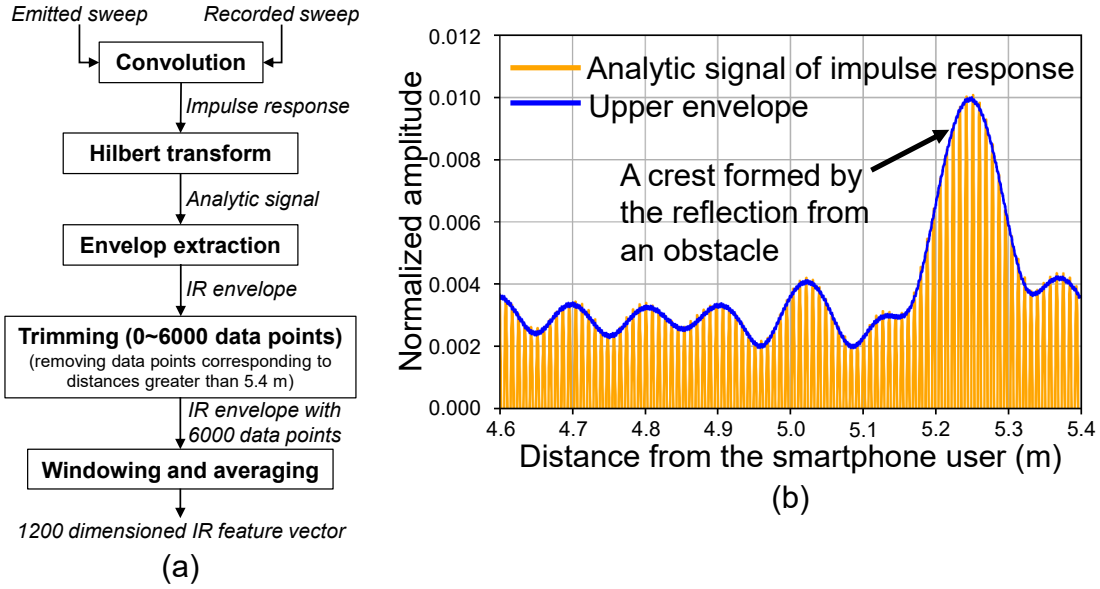


Figure 4.7: (a) IR feature extraction pipeline (b) extracted analytic signal and its upper envelope

using the recorded starting and ending times of the sine waves, sweeps, and inverse sweeps, and then fine-tuned the starting/ending times to find the times that yield the best match between the pseudo-spectrogram and recorded spectrogram.

This is done by simply shifting the starting and ending times of the pseudo-spectrogram until the Euclidean distance between the two spectrograms is minimized. Note that when we calculated the Euclidean distance, the distance for each time step was calculated and the sum over all the time steps within the time window was used as the value of the distance metric. Based on the predicted starting and ending times, the reflected signal can be separated into sweeps, inverse sweeps, and sine waves. The procedures of this process are also illustrated in Figure 4.6.

4.4.3 Impulse Response Calculation

Figure 4.7 (a) shows the IR feature extraction pipeline. Here, we calculated the impulse response using the segmented sweep (or inverse sweep) using the convolution between the recorded signal and the time-reversed transmitted signal [120]. Subsequently, we calculated the analytic representation of the impulse response using the Hilbert trans-

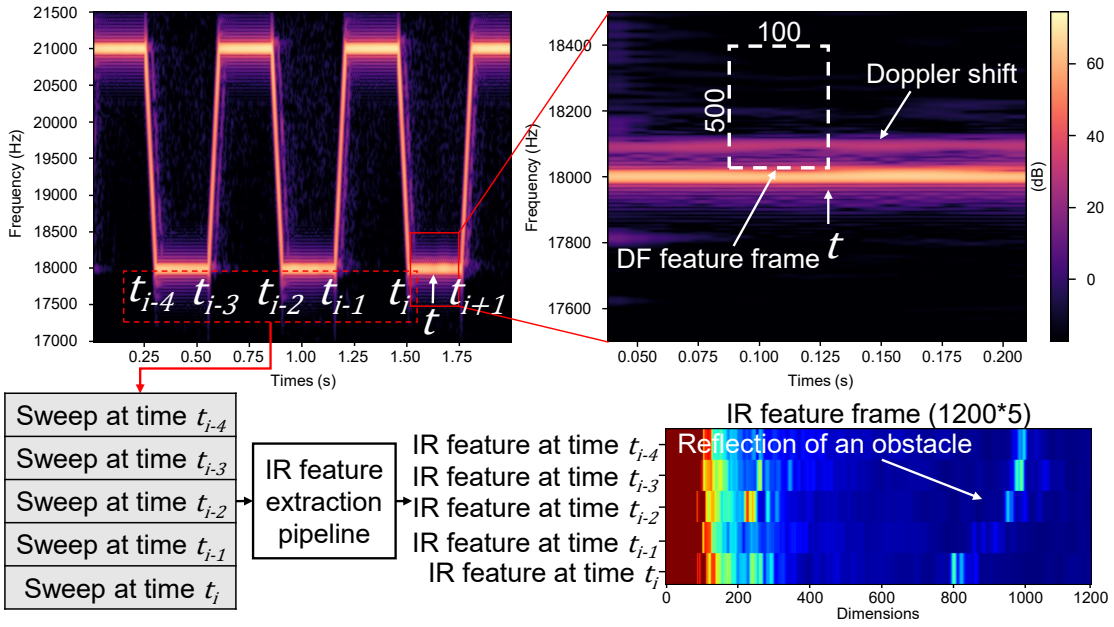


Figure 4.8: DF and IR feature preparation for ObsSense neural network

form and extracted the positive amplitudes of the analytic signal. Figure 4.7 (b) shows the analytic signal of the impulse response and the extracted upper envelope of the analytic signal. Thereafter, we separated the first 6000 data points from the envelope. Using the extracted 6000 data points, we can acquire obstacle distance data from obstacles situated 5.4 m away at most (Figure 4.2), which would be sufficient to achieve the goal of this study. We further reduced the dimensionality of this envelope up to 1200 by employing a rolling window with 100 samples with a stride of 5 samples and replaced the values of each window with the average value within the window, weighted by the standard deviation. This amplified the crests contained in the envelopes, caused by the reflections of the obstacle. As above, a vector of 1200 dimensions was computed from each envelope. We used these vectors as inputs to the neural network.

4.4.4 Doppler Frequency Extraction

We extracted 500 frequency bins from the top of the bandwidth of the 18 or 21 kHz sine wave from the FFT spectrograms, as shown in the top right panel of Figure 4.8. In this case, we were only interested in the positive Doppler shifts that occur when an obstacle

approaches the user. We used these Doppler shifts as inputs to our neural network.

4.4.5 ObsSense Neural Network Architecture

The input for the ObsSense neural network is composed of two features: i) time-series Doppler frequency features (DF features) extracted from the sine waves, and ii) the upper envelopes of the impulse responses (IR features) extracted from periodic sweep signals. Figure 4.8 shows the preparation of DF and IR feature frames. For a given time t, we use the previous 100 Doppler samples to form a DF feature frame of size 500×100. We further stacked the 5 closest previous IR envelopes to form an IR feature frame of size 1200×5. When sweeps were emitted at times t_{i-4}, t_{i-3}, t_{i-2}, t_{i-1}, and t_i and will be emitted at time t_{i+1}, to estimate the distance and the relative velocity of the obstacle at time t ($t_i < t < t_{i+1}$), we employed the envelopes of the sweeps emitted at times t_{i-4}, t_{i-3}, t_{i-2}, t_{i-1}, and t_i. By time-series stacking of the IR features, we captured information regarding the mobility of the obstacles.

The structure of the ObsSense neural network is shown in Figure 4.9. The Doppler extractor and IR extractor are composed of 2D convolution layers that process the DF features and IR features, respectively. The outputs of the Doppler extractor and IR extractor are then concatenated and fed into the mobility estimator. The mobility estimator has one output, which is the mobility label of the obstacle, i.e., static or mobile. Next, the intermediate output of the mobility extractor (4th layer output) is aggregated with the DF and IR features and fed into the distance/velocity estimator. By aggregating the intermediate output of the mobility detector, we allowed the neural network to select the most suitable method to predict the distance and the relative velocity of the obstacle based on its mobility. The distance/velocity estimator has two outputs, including the distance between the user and the obstacle and the relative velocity of the obstacle.

When the neural network is trained, we minimized the following loss function using backpropagation based on the Adam optimizer. [46].

$$\begin{aligned} E(\theta_D, \theta_I, \theta_m, \theta_{d,v}) =& \mathcal{L}_m(\theta_D, \theta_I, \theta_m) + \mathcal{L}_d(\theta_D, \theta_I, \theta_{d,v}) \\ &+ \mathcal{L}_v(\theta_D, \theta_I, \theta_{d,v}) + \mathcal{L}_{int}(\theta_D, \theta_I, \theta_{d,v}), \end{aligned} \tag{4.1}$$

where θ_D, θ_I, θ_m, and $\theta_{d,v}$ represent the network parameters of the Doppler extractor,

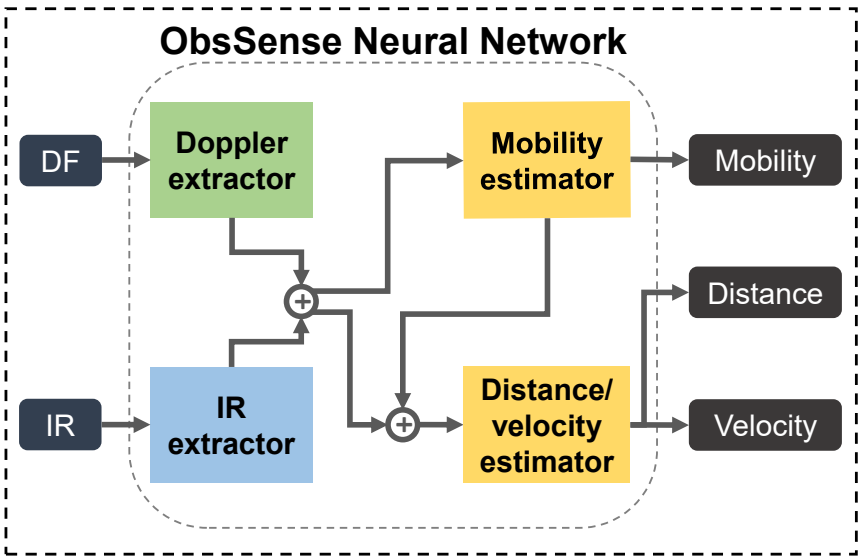


Figure 4.9: Structure of ObsSense neural network. Both the Doppler extractor and the IR extractor include three convolutional layers, each followed by max-pooling layers with a kernel size and step size of 3. Six kernels are included in all 3 convolutional layers in the Doppler extractor, with sizes of 11, 3, and 3, respectively. The number of kernels of the IR extractor are 6, 16, 6 and their step size is 3. Both the mobility estimator and the velocity/distance estimator include five fully connected layers with 500, 100, 50, 30, and 2 nodes, respectively. Each fully connected layer uses the ReLU activation function, except for the last fully connected layer of the mobility detector, which uses the sigmoid function.

IR extractor, mobility estimator, and distance/velocity estimator, respectively. $\mathcal{L}_m()$ denotes the loss of the mobility estimation, and calculates the binary cross-entropy loss of the prediction of the mobility classes, i.e., static or mobile. $\mathcal{L}_d()$ and $\mathcal{L}_v()$ denote the losses of the distance estimation and relative velocity estimation, respectively. $\mathcal{L}_d()$ and $\mathcal{L}_v()$ calculate the RMSLE (Root Mean Squared Log Error) losses of distance and velocity estimations, respectively. $\mathcal{L}_{int}()$ is introduced to leverage the relationship between the distance and velocity to supplement the estimations provided by ObsSense. Particularly, we employed the following relationship between the distance and velocity.

$$d_{t-1} = d_t + v_t \Delta t, \tag{4.2}$$

where d_t is the distance to the obstacle at time t, v_t is the relative velocity at time t, and Δt is the time difference between t and $t-1$. $t-1$ is the timestamp of the closest previous DF and IR feature frames. $\mathcal{L}_{int}()$ calculates the RMSLE loss of d_{t_i-1} prediction as follows.

$$\mathcal{L}_{int}() = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(d_{t_i-1}^{truth}+1) - \log(\tilde{d}_{t_i-1}+1))^2}. \tag{4.3}$$

Here, t_i denotes the time of the i^{th} instance, $\tilde{d}_{t_i-1}$ is a calculation result of d_{t_i-1} using Eq. 4.2 with distance/velocity estimates at t_i, $d_{t_i-1}^{truth}$ is the ground-truth of d_{t_i-1}, and n is the number of training instances.

4.5 Evaluation

4.5.1 Dataset

We collected data from five different static obstacle classes and three mobile obstacle classes that can be commonly found on sidewalks, as shown in Table 4.1, selected based on prior studies. Data were collected from obstacles situated inside the campus and in residential areas around the campus that were relatively busy during the daytime. The participant (smartphone user) was asked to walk toward an obstacle, starting 5 m from the obstacle, and randomly varying their walking speed while looking at the smartphone. The mobile obstacles started from much further (to build speed) and moved toward and past the user as close as possible to simulate a near-miss scenario. For this experiment, we used a Samsung S20 smartphone. The smartphone emitted the above probing signal and recorded the reflected waves. Simultaneously, the participant carried a TFMini Plus LiDAR module connected to an M5Stack Basic model to collect ground-truth data on the distance to obstacles at 1kHz. We used this time-series distance information to calculate the ground truth of the relative velocity of the obstacles through derivation. Table 4.1 shows the number of encounters of each obstacle class.

Table 4.1: Number of Encounters Per Obstacle Type

	Label	Encounters
static	parked vehicle	38
	billboard	36
	telephone pole	57
	wall	44
	standing person	27
mobile	walking person	30
	jogging person	30
	bicycle	65
Total		327

4.5.2 Evaluation Methodology

To evaluate the performance of ObsSense, we first randomly separated our dataset (encounters) into training and testing datasets using 70% and 30%, respectively, of the total data. No data from the training encounters (obstacles) were used to test the model. We prepared the following methods to investigate the effectiveness of ObsSense.

- ObsSense: This was our proposed method.
- w/o integration loss: This was a variant of the proposed method that did not employ the loss term $\mathcal{L}_{int}()$.
- w/o mobility detector: This method did not employ the intermediate-layer output of the mobility detector.
- Only IR: This was a variant of the proposed method that did not employ the Doppler features.
- Only Doppler: This method did not employ the IR features.
- DeepRange Doppler: This method was designed based on the state-of-the-art deep learning method [73]. The method was originally designed to recognize hand gestures and output the distance to the hand by processing the raw reflected sounds of the sine sweeps. However, because it exhibited poor performance in our preliminary experiment when using raw sounds, we used the Doppler features as inputs. The network architecture used in this experiment was identical to [73], except for the output layer, where our

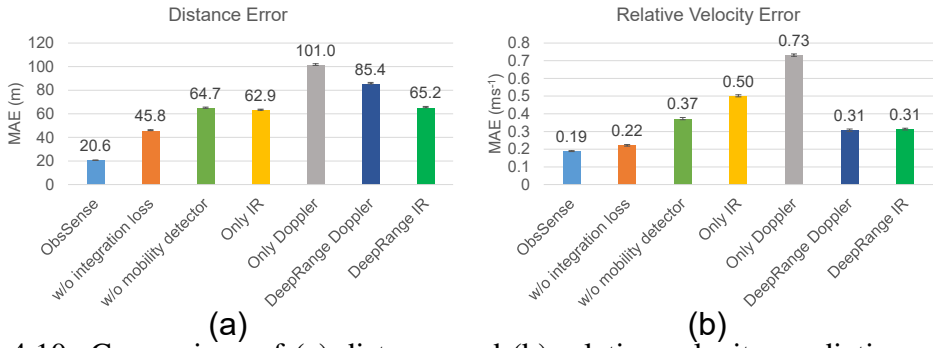


Figure 4.10: Comparison of (a) distance and (b) relative velocity prediction errors of ObsSense and the other methods

version had two output nodes to estimate the distance and relative velocity.

- DeepRange IR: This method was also based on [73]. This method uses IR features as the input.

The performance of the distance and relative velocity estimations was measured in terms of the mean absolute error (MAE) between the predictions and the ground truth.

4.5.3 Results

Performance of ObsSense

Figure 4.10 shows the distance and velocity prediction performances of ObsSense and the other methods. The MAEs of the distance and velocity prediction by ObsSense were significantly smaller than those of the other methods, and ObsSense achieved MAEs of only 20.6 cm and 0.19 ms^{-1}. As shown in Figure 4.11 (a,b), ObsSense precisely predicted the distance and velocity, even though the participant randomly changed the walking speed. In addition, as shown in Figure 4.11 (c), the recorded IR envelopes are very noisy, and the reflections from the wall are not clearly visible, indicating the difficulties of applying threshold-based distance prediction to noisy real-world data. (The performance of threshold-based distance prediction is investigated below.)

Figure 4.12 shows the MAEs of ObsSense for the different obstacle types. The distance errors for persons are relatively larger than those for the other obstacles because the size of person is smaller than that of the other obstacles. Figures 4.13 and 4.14 show the MAEs of ObsSense for different ground truth distances and relative velocities. As

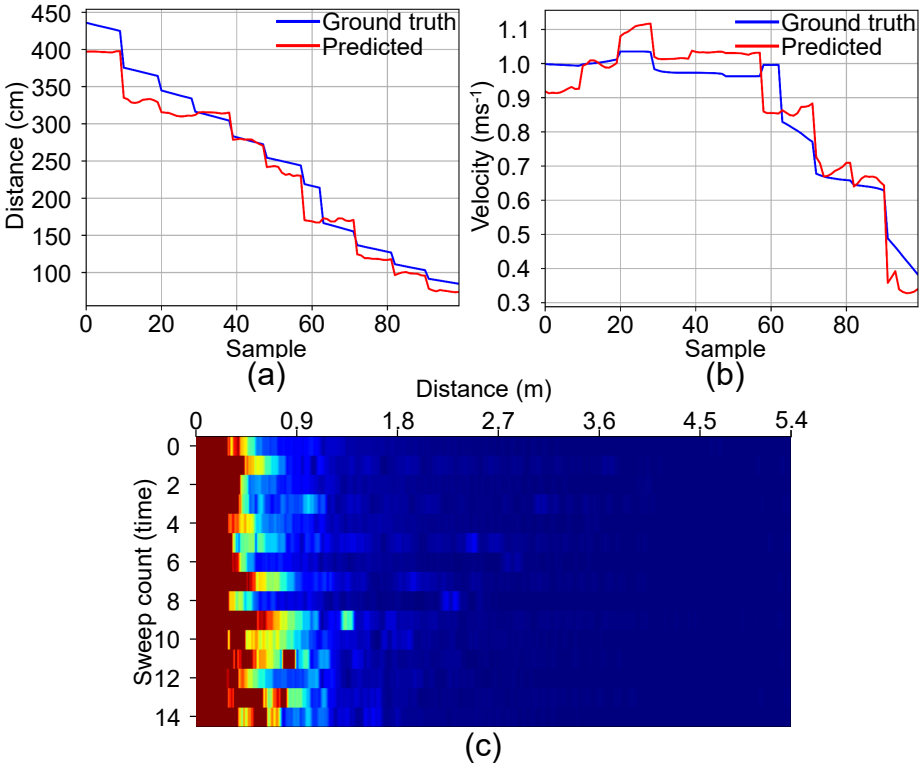


Figure 4.11: Comparison of (a) predicted relative velocity and (b) predicted distance against the ground-truth when the participant walked towards a wall. (c) the reflections by the wall recorded on the IR envelopes

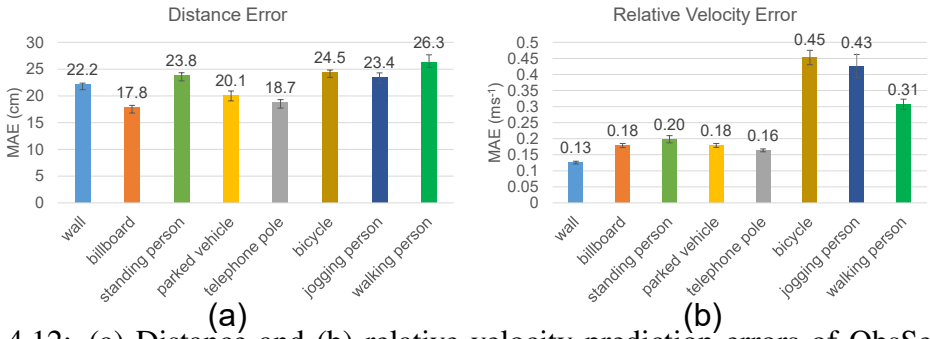


Figure 4.12: (a) Distance and (b) relative velocity prediction errors of ObsSense for different obstacle types

can be seen, when the distance from the obstacle increased, the distance and relative velocity prediction errors increased. This is because when the distance to the obstacle increases, the amplitudes of the reflections from the obstacles become weak. Further-

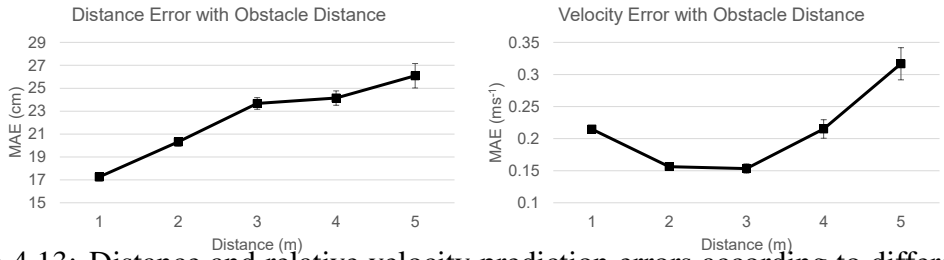


Figure 4.13: Distance and relative velocity prediction errors according to different obstacle distances

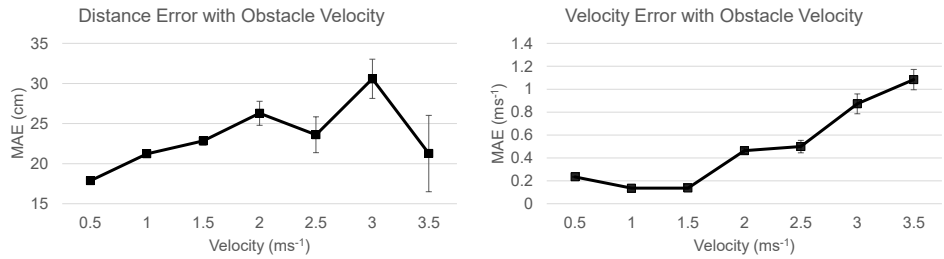


Figure 4.14: Distance and relative velocity prediction errors according to different obstacle velocities

more, when the relative velocity increased, the distance and relative velocity prediction errors increased. This result is natural because the intervals between the sweeps are relatively large compared with the speed. At a distance of 3 m, the relative velocity error was approximately 0.15 ms^{-1}. When the relative velocity of the obstacle was 2 ms^{-1}, the collision time estimation produced an average error of 0.1 s. This accounted for only 6.6% of the total collision time.

Moreover, Figure 4.10 shows the performance of **DeepRange Doppler** and **DeepRange IR**. While the neural networks of these methods were designed based on state-of-the-art methods, the MAEs of these methods were much poorer than those of ObsSense, indicating the effectiveness of our proposed probing signals and the design of our neural network.

Contributions of $\mathcal{L}_{int}()$ and Mobility Detector

Figure 4.10 shows the performance of **w/o integration loss** and **w/o mobility detector**. The distance error of **w/o integration loss** is much larger than that of ObsSense. Specifically, as shown in Figure 4.15, the distance estimation error in **w/o integration loss** for

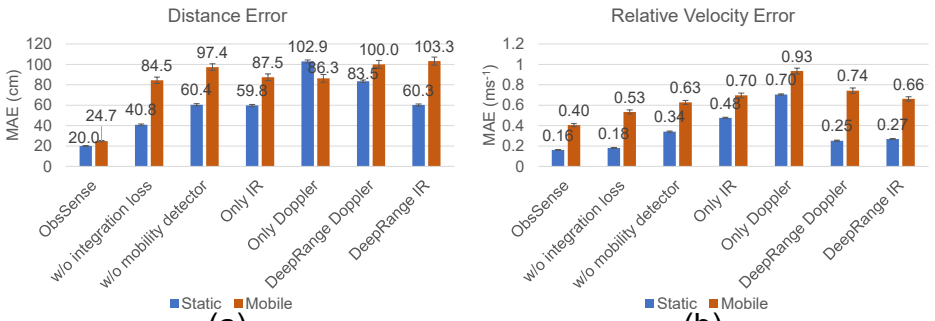


Figure 4.15: Comparison of (a) relative velocity and (b) distance prediction errors of different methods according to obstacle mobility

mobile obstacles is significantly higher than that of ObsSense, indicating the effectiveness of $\mathcal{L}_{int}()$ by enhancing the distance prediction with the velocity information.

As suggested by the performance of **w/o mobility detector**, the contribution of the mobility detector was significant. As shown in Figure 4.15, the mobility estimator improves the accuracy of the distance and relative velocity estimations in both static and mobile obstacles, indicating the effectiveness of knowledge regarding the mobility of the obstacle. Because the classification performance of the mobility detector is high (F-measure: 94%), ObsSense seems to adaptively predict the distance and velocity by referring to the intermediate outputs of the mobility detector, which precisely describes the mobility status of an obstacle.

Contributions of DF and IR Features

Figure 4.10 shows the performance of **Only IR** and **Only Doppler**, indicating the significant contribution of IR features. Contrary to our expectations, the velocity error of **Only Doppler** was larger than that of **Only IR**. This can be because $\mathcal{L}_{int}()$ negatively affects velocity prediction. Because it is impossible to predict the distance by using only the DF features, erroneous distance prediction can cause a conflict regarding the distance-velocity relationship.

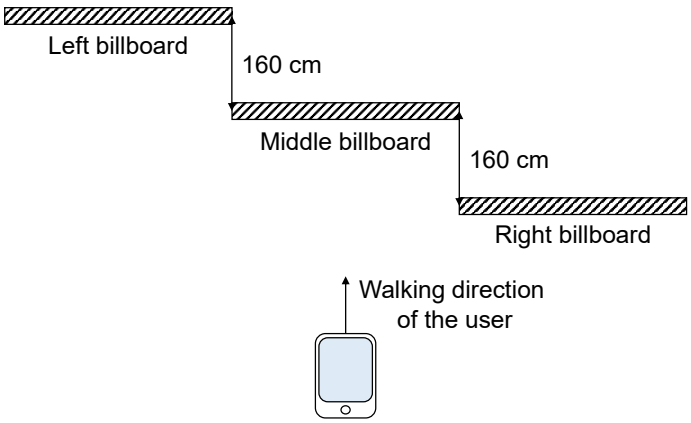


Figure 4.16: Experimental setting for multiple obstacle encounters

4.5.4 Discussion

Testing on Other Users

When we predict the relative velocity of a static obstacle, ObsSense relies on sound reflections from moving body parts of the user. To test the generalizability of our method across different users, we collected data on fifteen encounters from three additional participants. We fed the extracted features into a trained ObsSense neural network. Note that no data from the above participants were used to train ObsSense. The MAE values of the distance and relative velocity predictions were 34.1 cm and 0.29 ms^{-1}, respectively. The MAEs of the distance and velocity predictions of the data of the additional participants are somewhat poorer than that of the primary user. We believe that the reason for this is the body sizes of the additional users were different from the body size the primary user. However, ObsSense also achieved reasonable performance in this setting.

Multiple Obstacle Encounters

To observe the behavior of our method in the case of a multiple obstacle encounter, we used three billboards situated side-by-side to each other, the rightmost one being the closest to the user and the left being the furthest. The experimental setting is shown in Figure 4.16. The distance between the right and middle billboards was 160 cm, and

the distance between the middle and left billboards was 160 cm towards the walking direction of the user. We collected data from five encounters where a participant walked towards the center of the billboards, i.e., the middle billboard, holding the smartphone, and the LiDAR module was directed towards the right billboard to acquire the ground truth of the closest obstacle. We tested the collected data using ObsSense. Although we assumed that ObsSense outputs the distance to the closest billboard, i.e., the rightmost billboard, ObsSense seems to output the distance to the middle billboard. Therefore, the MAE of the distance estimation was 191.4 cm. As the distance to the middle billboard was 160 cm greater than the distance to the ground truth, i.e., the distance to the right billboard, the MAE of the distance estimation of the middle billboard was as 31.4 cm. This could be because the smartphone was directed toward the middle billboard. ObsSense seems to output the distance to an obstacle situated in front of the user because sound reflections from such obstacles could be stronger than from other obstacles.

Performance Under Different Noise Levels

To test our method under different noise levels, we artificially injected noise from urban areas under different (10, 6, 3, and 1 dB) signal-to-noise ratios (SNRs) [2] into five sessions of test data. We used noise from the MAVD-traffic dataset [138] that contains noise from different types of vehicles recorded in urban environments. The MAEs of the distance predictions for different SNRs were 16.7 cm, 16.7 cm, 16.8 cm, and 16.8 cm, respectively. The MAEs of velocity predictions for all SNRs were stable at 0.12 ms^{-1}. When we tested the above five sessions without adding any noise, the MAEs of distance and velocity predictions were 16.6 cm and 0.12 ms^{-1}, respectively. Similar to prior studies using inaudible sounds [2, 42, 111], the effect of the noises was limited.

Amount of Training Data

Figure 4.17 shows the effect of the amount of training data on ObsSense. As the amount of training data increased, the MAE of the distance and relative velocity predictions decreased. With 50% of the training data, ObsSense surpasses the performance of all other methods, achieving a distance error of 30 cm and a relative velocity error of 0.2 ms^{-1}.

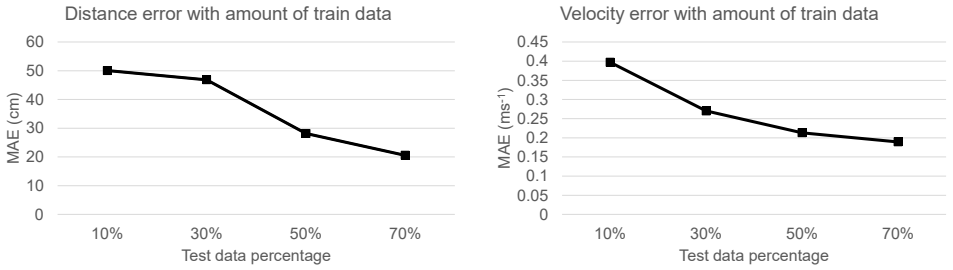


Figure 4.17: Distance and relative velocity prediction errors of ObsSense according to different amounts of train data

Performance of Threshold-based Method

Here, we investigated a threshold-based distance estimation method using IR features. The threshold-based method employs peaks (crests) that are greater than the threshold created on the IR features to estimate the distance to the obstacle, as shown in Figure 4.7 (b). However, because of the semi-naturalistic data collection protocol that we followed, multiple crests created by multiple small obstacles in front of the user are unavoidable. Because it was not possible to distinguish the crest created by the targeted obstacle from that of other obstacles, we calculated the MAE between the ground truth and the largest crest created on the IR envelope. The MAE of the distance estimation of the threshold-based method was 154.8 cm, which is much higher than that of ObsSense. This is because, as Figure 4.11 (c) demonstrates, the data we have collected for this experiment while the user was walking is very noisy.

Comparison with Other Studies

A prior work related to threshold-based methods [120] achieved distance estimation error of 3 cm from a distance of 5 m. This error was smaller than that of ObsSense. However, unlike experiments of distance prediction in [120], the ObsSense experiment used noisy data collected while the user walked. As can be seen from the evaluation of the threshold-based method, the average distance estimation error was significantly larger.

Note that ObsSense requires training data with distance labels, which are not required in threshold-based methods. We collected the ground truth using the TFMini

Plus LiDAR module, which uses a laser to measure distance. Hence, distance ground-truth errors due to the location of the obstacle at which the laser is pointed are inevitable. This is a limitation of our proposed method.

CycleGuard [41] employed a threshold-based method with an external speaker connected to a smartphone to estimate the distance between a cyclist and an approaching vehicle. This method achieves an average distance error of 10 cm. However, this study does not clearly describe the method in which the ground truth is acquired. If ObsSense is to be used to measure the distance between a user and a car, the obtained ground truth of the distance could change from more than 10 cm according to the location where the laser of the LiDAR hits the car, for example, the bumper or windscreen. However, ObsSense, which is based on supervised learning, yields reasonable performance even when we used noisy reflected signals such as Figure 4.11 (c) obtained by off-the-shelf smartphone speakers.

4.6 Conclusion

We presented ObsSense, a method for estimating the distance and relative velocity of oncoming obstacles using smartphone active sound sensing. We proposed a novel probing signal that facilitates the distance and relative velocity estimation of both static and mobile obstacles. Additionally, we proposed a novel neural network architecture that considers the mobility of obstacles to jointly predict the distance and relative velocity of obstacles. As part of our future work, we plan to extend our method to audio-based obstacle class prediction for precise risk-level assessment.

Chapter 5

Conclusion and Future Work

5.1 Summary of the Thesis

Proposing sound-based sensing methodologies to capture surrounding contextual information of a smart device user, without distributing sensors in the environment, was the ultimate goal of this study (Section 1.3). Towards this end, we presented sound sensing techniques that can be employed in both indoor and outdoor settings. Our proposed methods do not use specialized sensors and can be deployed using off-the-shelf smart devices. Compared to the distributed sensor method that capture context information from the user's surroundings, our methods use a minimum number of sensor nodes. This makes our methods easy to implement and maintain.

To achieve our goal, we investigated different probing signals components that can be employed to probe for different attributes of contextual information. We also proposed novel probing signals that combine the characteristics of multiple individual signal components, that can be employed to capture multiple contextual information attributes simultaneously. We also investigated the technical difficulties that arise when implementing these sensing techniques on off-the-shelf smart devices and proposed solutions to them.

Our sound sensing techniques can be employed to capture the following contextual information contained in the user's surroundings:

1. Movements of door objects as well as the movements of the user relative to a

stationary probing device.

2. Environmental acoustic characteristics attributed to the open/close states of the doors in the surrounding.

3. Acoustic characteristics of a location attributed to abundantly available materials in the surrounding.

4. Distance between a probing device and an object and relative velocity of an object with respect to the probing device.

The aforementioned contextual information is the foundation of numerous applications. Hence, this thesis was able to greatly expand the application areas of sound-based context aware techniques.

Specifically, the contributions made in each chapter can be listed as follows:

- **Recognizing events and states of the doors**

  Events/state changes related to door objects in an indoor environment change the environmental attributes. Hence, when capturing the user's surrounding context, recognizing events of the door objects of the user's environment is an important task. In this study, we proposed a sound-based low-cost door event/state recognition method that employs a single disused smartphone to capture the event/states of multiple doors in an environment. Our method does not require attaching separate sensor nodes to each door object in the environment, reducing the number of sensor nodes needed, hence, lowering the installation and maintenance costs. In contrast to door event/state recognition methods that are based on contact sensors attached to each door, our method also can recognize walking events related to door objects, which is an important factor for the applications such as automation of HVAC control and surveillance.

- **Predicting indoor location class of the user**

  The indoor location class of the user has a deep relationship with the user's activities. Therefore, recognizing the user's indoor location class is an important task as it helps to understand the user's daily life. In this study, we proposed a

method that employs sensor data, such as audio data, from the user's smartwatch to estimate the indoor location class of the user. Furthermore, we proposed a novel matrix manipulation method that extracts location-specific short sensor data segments that are attributed to location-specific activities or acoustic characteristics. Moreover, we proposed a method based on a domain adversarial approach to extract environment-independent features from the extracted location-specific sensor data and to automatically construct a location classifier. We also used a novel data fusion method that employs both acoustic and acceleration data from the user's smartwatch to estimate the user's indoor location.

- **Estimating mobility, distance, and relative velocity of roadside obstacles**

  Roadside obstacles can pose a threat to any pedestrian who walks on the sidewalk. This threat gets increased by several folds if the pedestrian is distracted. Specifically, pedestrians that walk on the sidewalk, distracted by their smartphones, can bump into roadside obstacles and can cause accidents. Hence, we proposed a method to employ the smartphone to perform active sound sensing and estimate the distance and relative velocity of the oncoming obstacles. In comparison to the existing methods, our method can estimate the distance and relative velocity of both static and mobile obstacles. In this study, we proposed a novel probing signal architecture that facilitates distance and velocity estimation and a novel neural network architecture that estimates and considers the mobility of the obstacle to jointly estimate its distance and relative velocity.

5.2 Future work

The ultimate goal of this study is to build a platform for smart device users' surrounding context recognition. To this end, we discuss possible directions that our work can be improved on.

- **Capturing additional information regarding door events/states**

  In this thesis, we employed a disused smartphone to capture information regarding the events/states related to door objects. While this is a cost-effective option

as disused smartphones can be found in abundance at present, the number of in-built microphones on the majority of smartphones varies from two to three. However, devices such as Amazon Echo and Google Home, which are common smart speakers at present, contain more than two microphones that can be employed to capture rich information regarding door events/states. As an example, a microphone array can be employed to capture information regarding the direction of the door, considering the difference in the reflected signal's time of arrival (ToA) at each microphone in the array. Similarly, information regarding the user's walking direction in the environment can also be captured using Doppler shifts captured by each microphone. Implementing such systems using a smart speaker is an important task in our future work.

- **Location class estimation methods for hospitals and factories**

  Even though this thesis presented a method to estimate the user's indoor location using the sensor data from the user's smartwatch, we believe that our method can be implemented in numerous different settings to assign semantic labels to different locations. For example, hospital workers perform different activities in different rooms in large hospital facilities. Specific sensor data segments, such as acoustic information from different instruments (noises from the centrifuge in the laboratory and sounds from heartbeat monitors at the intensive care unit), from smart watches placed on nurses' wrists can be used to estimate semantic locations to these rooms. Furthermore, we believe that we can implement our method in a factory setting to extract machine-specific noises from different machines, and actions from the user's wrist when operating them in a factory. This information can then be employed to estimate a semantic label for different areas in large-scale factories. These semantic labels can then be used to understand the bottlenecks of the work floor and production system.

- **Audio-based obstacle class prediction**

  In this thesis, we presented a method to estimate three important attributes of roadside obstacles, i.e., mobility, distance, and relative velocity. We believe that we can employ sound sensing to also predict the class of the oncoming obstacles. The frequency response of the impulse responses of the reflected sweep signals

contains information regarding the class of the obstacle, attributed to the materials that the obstacles are made of. The basic idea is that some materials absorb sounds at certain frequencies well, while other materials reflect them. We believe that we can employ such frequency characteristics to predict the class of the oncoming obstacles. Our preliminary investigation [28] showed that this method can be employed to recognize obstacle class of six static obstacle classes with high accuracy.

Acknowledgment

It has been ten years since I first visited Japan as an undergraduate student. Upon completion of my PhD thesis, I would like to reflect back at my student life at Osaka University and thank a number of very special people, without whom I could not have possibly achieved this feat.

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Takuya Maekawa, for giving me the opportunity to conduct my research under his guidance. I was lucky enough to be his student since I was an undergraduate student and it was a privilege and an honor to be studying from him. Professor Maekawa taught me the importance of dissecting a large problem into a manageable segments and tackle them individually. He was always receptive to my ideas and concerns and patiently guided me to the right path to achieve my goals. During the research work, we had countless discussions and arguments regarding the challenges we faced and he could always envision several steps ahead, which always inspired me to become a better researcher. I would not have completed this thesis without his immense support and commendable effort and guidance. I am ever grateful to him for proposing research topics, methods and techniques and helping me to write research papers and even helping me with my career choices.

I am also grateful to Professor Takahiro Hara, for accepting me to his laboratory to pursue my dreams in research. He was always ready to share his time and expertise with me and his comments and advises improved the quality of this research. I always appreciate his generous help in guiding me through difficulties in my research career.

I also like to extend my gratitude to the committee members for my thesis, professors Toru Fujiwara, Makoto Onizuka, Yasuyuki Matsushita, Shinji Shimojo, and Norihiro Hagita for giving me insightful comments and advises to improve the qual-

ity of this thesis. My sincere gratitude also goes to professors Daichi Amagata, Atsuo Inomata, Kyosuke Yamashita, Yuki Arase, Chuan Xiao, Fumio Okura, Susumu Date, Takahiro Miyashita, and Satoru Satake for your valueble comments and advises during examinations, mid-term defences, and doctoral pre-defence.

I would also like to thank all my colleagues, lab-mates, and friends, specially Dr. Joseph Korpela, Dr. Teerawat Kumrai, Dr. Xia Qingxin, Dr. Ryoma Otsuka, Mr. Naoya Yoshimura, Mr. Ji Yuchen, Mr. Jaime Morales, Mr. hi Li, Mr. Cao Guanyu, Mr. Jiao Shengzhe, Mr. Zhou Heng, Mr. Ryosuke Taniguchi, Mr. Yuuki Nishino, Ms. HiKaru Nomura, Mr. Kohei Hirata, Mr. Wang Yuqiao, Mr. Kentaro Shiga, Mr. Kei Tanigaki, Mr. Rikuto Tsubouchi, Mr. Takuma Yamashita, Mr. Kazuyoshi Aoyama, Mr. Ryuichiro Okuda, Mr. Yukihiro Kadono, Mr. Ryo SHirai, Mr. Keisuke Tsukamoto, Mr. Yuma Douse and Mr. Ramos Julio Alejandro who willingly and enthusiastically participated in experiments and made my stay in Hara laboratory a wonderful experience. I am specially thankful to my friend and previous post-doctoral researcher at Hara laboratory, Dr. Renée Schulz, for always helping to keep my morale high during troubled times. Additionally, I would like to thank the non-academic staff of the laboratory, Mrs. Chikako Kawabe, Mrs. Kana Yasuda, Mrs. Makiko Higashinaka, Mrs. Takako Ishikawa, and Mrs. Makiko Otsuka for helping me with my experiments and problems related to the student life at Osaka University.

I am ever grateful to the Government of Japan for facilitating me to study in Japan and funding my education until I finished my Master's studies. I am also thankful to Sumitomo Chemical Co., Ltd. for providing me a scholarship for my Doctoral studies. I would also like to extend my gratitude to Humanware Innovation Program, Osaka University for providing me financial support and funding my research activities for five years.

My sincere gratitude goes to my parents and my brother who always encouraged me on my journey towards the PhD. My farther was my inspiration to work towards greatness and my mother's kindness always made me a better person. You laid a solid foundation for me to obtain my PhD. My brother always made me laugh during troubled times and helped me to relieve my mind. Upon the completion of this thesis, I would also like to thank all my teachers both in Sri Lanka and Japan who shared their knowledge with me without hesitation. You made me the person I am today. Last but not least,

this special token of appreciation goes to my loving wife, Yashodmi Kaluarachchi who sacrificed a lot to be by my side during this journey, and for always being my strength. She always took care of me and was supportive towards my decisions. Thank you for providing me warmth when I desperately needed.

REFERENCE

[1] H. Abdelnasser, R. Mohamed, A. Elgohary, M. F. Alzantot, H. Wang, S. Sen, R. R. Choudhury, and M. Youssef. Semanticslam: Using environment landmarks for unsupervised indoor localization. *IEEE Transactions on Mobile Computing*, 15(7):1770–1782, 2015.

[2] T. Amesaka, H. Watanabe, M. Sugimoto, and B. Shizuki. Gesture recognition method using acoustic sensing on usual garment. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–27, 2022.

[3] N. Aoshima. Computer-generated pulse signal applied for sound measurement. *The Journal of the Acoustical Society of America*, 69(5):1484–1488, 1981.

[4] S. Asano, Y. Wakuda, N. Koshizuka, and K. Sakamura. A robust pedestrian dead-reckoning positioning based on pedestrian behavior and sensor validity. In *IEEE/ION Position Location and Navigation Symposium*, pages 328–333, 2012.

[5] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, pages 261–272, 2009.

[6] D. Banerjee, S. K. Agarwal, and P. Sharma. Improving floor localization accuracy in 3d spaces using barometer. In *Proceedings of the International Symposium on Wearable Computers*, pages 171–178, 2015.

[7] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of the International Conference on Pervasive Computing*, pages 1–17, 2004.

[8] X. Bao, B. Liu, B. Tang, B. Hu, D. Kong, and H. Jin. PinPlace: associate semantic meanings with indoor locations without active fingerprinting. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 921–925, 2015.

[9] C. Beckmann, S. Consolvo, and A. LaMarca. Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. In *Propceedings of 6th International Conference on Ubiquitous Computing*, pages 107–124, 2004.

[10] C. Bi, G. Xing, T. Hao, J. Huh, W. Peng, and M. Ma. Familylog: A mobile system for monitoring family mealtime activities. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications*, pages 21–30. IEEE, 2017.

[11] J. T. Biehl, A. J. Lee, G. Filby, and M. Cooper. You're where? prove it! towards trusted indoor location estimation of mobile devices. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 909–919, 2015.

[12] I. Bisio, C. Garibotto, F. Lavagetto, and A. Sciarrone. Outdoor places of interest recognition using wifi fingerprints. *IEEE Transactions on Vehicular Technology*, 68(5):5076–5086, 2019.

[13] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[14] A. Caldini, M. Fanfani, and C. Colombo. Smartphone-based obstacle detection for the visually impaired. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 480–488. Springer, 2015.

[15] B. Campbell and P. Dutta. An energy-harvesting sensor architecture and toolkit for building monitoring and event detection. In *Proceedings of the 1st ACM*

Conference on Embedded Systems for Energy-Efficient Buildings, pages 100–109, 2014.

[16] J. Chen, A. H. Kam, J. Zhang, N. Liu, and L. Shue. Bathroom activity monitoring based on sound. In *International Conference on Pervasive Computing*, pages 47–61, 2005.

[17] N. Cherif, Y. Ouakrim, A. Benazza-Benyahia, and N. Mezghani. Physical activity classification using a smart textile. In *Proceedings of the 2018 IEEE Life Sciences Conference*, pages 175–178. Ieee, 2018.

[18] S.-B. Cho. Exploiting machine learning techniques for location recognition and prediction with smartphone logs. *Neurocomputing*, 176:98–106, 2016.

[19] J. Choi and R. Gutierrez-Osuna. Using heart rate monitors to detect mental stress. In *Proceedings of the 6th International Workshop on Wearable and Implantable Body Sensor Networks*, pages 219–223. IEEE, 2009.

[20] S. Chung and I. Rhee. vtrack: virtual trackpad interface using mm-level sound source localization for mobile interaction. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 41–44, 2016.

[21] B. Clarkson, A. Pentland, and K. Mase. Recognizing user context via wearable sensors. In *Proceedings of the Symposium on Wearable Computers*, pages 69–75, 2000.

[22] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see Bob?: human localization using mobile phones. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking*, pages 149–160, 2010.

[23] R. Cortés, X. Bonnaire, O. Marin, and P. Sens. Sport trackers and big data: Studying user traces to identify opportunities and challenges. 2014.

[24] M. Cowling. *Non-speech environmental sound recognition system for autonomous surveillance*. PhD thesis, Griffith University, 2004.

[25] T. Dissanayake, T. Maekawa, D. Amagata, and T. Hara. Detecting door events using a smartphone via active sound sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):1–26, 2018.

[26] T. Dissanayake, T. Maekawa, T. Hara, T. Miyanishi, and M. Kawanabe. Indolabel: Predicting indoor location class by discovering location-specific sensor data motifs. *IEEE Sensors Journal*, 22(6):5372–5385, 2021.

[27] T. Dissanayake, T. Maekawa, and H. Takahiro. Preliminary investigation of employing smartphone active sound sensing to predict distance and relative velocity of the roadside obstacles to aid distracted pedestrians. *Research Report Ubiquitous Computing Systems (UBI)*, 2022-UBI-76(25):1–8, Nov 2022.

[28] T. Dissanayake, T. Maekawa, and H. Takahiro. Preliminary investigation of obstacle recognition via smartphone active sound sensing for pedestrian safety. *Research Report Ubiquitous Computing Systems (UBI)*, 2021-UBI-71(21):1–8, Aug 2022.

[29] M. Elhamshary, M. Youssef, A. Uchiyama, H. Yamaguchi, and T. Higashino. Transitlabel: A crowd-sensing system for automatic labeling of transit stations semantics. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 193–206. ACM, 2016.

[30] M. Fan, A. T. Adams, and K. N. Truong. Public restroom detection on mobile phone via active probing. In *International Symposium on Wearable Computers (ISWC 2014)*, pages 27–34, 2014.

[31] A. Farina. Simultaneous measurement of impulse response and distortion with a swept-sine technique. In *Audio Engineering Society Convention 108*. Audio Engineering Society, 2000.

[32] A. Farina. Advancements in impulse response measurements by sine sweeps. In *Audio Engineering Society Convention 122*. Audio Engineering Society, 2007.

[33] B. Fu, D. Vaithyalingam, A. Kuijper, F. Kirchbuchner, and A. Braun. Exercise monitoring on consumer smart phones using ultrasonic sensing. In *Proceedings*

of the 4th International Workshop on Sensor-based Activity Recognition and Interaction (iWOAR 17), 2017.

[34] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[35] C. Gini. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.]*. Tipogr. di P. Cuppini, 1912.

[36] B. Gozick, K. P. Subbu, R. Dantu, and T. Maeshiro. Magnetic maps for indoor navigation. *IEEE Transactions on Instrumentation and Measurement*, 60(12):3883–3891, 2011.

[37] S. Gupta, D. Morris, S. Patel, and D. Tan. Soundwave: using the doppler effect to sense gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1911–1914. ACM, 2012.

[38] M. Hardegger, G. Tröster, and D. Roggen. Improved ActionSLAM for long-term indoor tracking with wearable motion sensors. In *International Symposium on Wearable Computers (ISWC2013)*, pages 1–8, 2013.

[39] S. Jain, D. Hallac, R. Sosic, and J. Leskovec. Casc: Context-aware segmentation and clustering for motif discovery in noisy time series data. *arXiv preprint arXiv:1809.01819*, 2018.

[40] M. E. S. M. Jasman, A. N. Nasaruddin, and B. T. Tee. Analysis of indoor environment condition towards thermal comfort level in an air-conditioning office building. In *IOP Conference Series: Earth and Environmental Science*, volume 373, page 012015. IOP Publishing, 2019.

[41] W. Jin, S. Murali, Y. Cho, H. Zhu, T. Li, R. T. Panik, A. Rimu, S. Deb, K. Watkins, X. Yuan, et al. Cycleguard: A smartphone-based assistive tool for cyclist safety using acoustic ranging. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–30, 2021.

[42] W. Jin, M. Xiao, H. Zhu, S. Deb, C. Kan, and M. Li. Acoussist: An acoustic assisting tool for people with visual impairments to cross uncontrolled streets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4):1–30, 2020.

[43] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[44] H. Kang, G. Lee, and J. Han. Obstacle detection and alert system for smartphone ar users. In *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11, 2019.

[45] S. Kawanaka, Y. Kashimoto, A. Firouzian, Y. Arakawa, P. Pulli, and K. Yasumoto. Approaching vehicle detection method with acoustic analysis using smartphone for elderly bicycle driver. In *Proceedings of the 10th International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, pages 1–6. IEEE, 2017.

[46] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[47] P. A. Kodeswaran, R. Kokku, S. Sen, and M. Srivatsa. Idea: A system for efficient failure management in smart IoT environments. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 2016)*, pages 43–56, 2016.

[48] J. Korpela, R. Miyaji, T. Maekawa, K. Nozaki, and H. Tamagawa. Evaluating tooth brushing performance with smartphone sound data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 109–120, 2015.

[49] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[50] K. Kunze and P. Lukowicz. Symbolic object localization through active sampling of acceleration and sound signatures. pages 163–180, 2007.

[51] M. Kurz, R. Gstoettner, and E. Sonnleitner. Smart rings vs. smartwatches: Utilizing motion sensors for gesture recognition. *Applied Sciences*, 11(5):2015, 2021.

[52] H. Kuttruff. *Room acoustics*. CRC Press, 2009.

[53] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al. Place lab: Device positioning using radio beacons in the wild. In *International Conference on Pervasive Computing*, pages 116–133, 2005.

[54] G. Laput, X. Chen, and C. Harrison. Sweepsense: Ad hoc configuration sensing using reflected swept-frequency ultrasonics. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 332–335, 2016.

[55] F. Li, H. Chen, X. Song, Q. Zhang, Y. Li, and Y. Wang. Condiosense: High-quality context-aware service for audio sensing system via active sonar. *Personal and Ubiquitous Computing*, 21(1):17–29, 2017.

[56] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser. Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9. IEEE, 2016.

[57] S. Li, X. Fan, Y. Zhang, W. Trappe, J. Lindqvist, and R. E. Howard. Auto++ detecting cars using embedded microphones in real-time. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–20, 2017.

[58] M.-I. B. Lin and Y.-P. Huang. The impact of walking while using a smartphone on pedestrians' awareness of roadside events. *Accident Analysis & Prevention*, 101:87–96, 2017.

[59] M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh. Valmod: A suite for easy and exact detection of variable length motifs in data series. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1757–1760. ACM, 2018.

[60] C. Liu, S. Jiang, S. Zhao, and Z. Guo. Infrastructure-free indoor pedestrian tracking with smartphone acoustic-based enhancement. *Sensors*, 19(11):2458, 2019.

[61] P. Lukowicz, G. Pirkl, D. Bannach, F. Wagner, A. Calatroni, K. Förster, T. Holleczek, M. Rossi, D. Roggen, G. Tröster, et al. Recording a complex, multi modal activity data set for context recognition. In *Proceedings of the 23th International Conference on Architecture of Computing Systems*, pages 1–6. VDE, 2010.

[62] L. Ma, B. Milner, and D. Smith. Acoustic environment classification. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(2):1–22, 2006.

[63] T. Maekawa, Y. Kishino, Y. Yanagisawa, and Y. Sakurai. Mimic sensors: Battery-shaped sensor node for detecting electrical events of handheld devices. In *Proceedings of the International Conference on Pervasive Computing*, pages 20–38, 2012.

[64] T. Maekawa, Y. Kishino, Y. Yanagisawa, and Y. Sakurai. Wristsense: wrist-worn sensor device with camera for daily activity recognition. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 510–512, 2012.

[65] T. Maekawa, D. Nakai, K. Ohara, and Y. Namioka. Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1088–1099, 2016.

[66] T. Maekawa and Y. Sakumichi. Easy to install indoor positioning system that parasitizes home lighting. In *European Conference on Ambient Intelligence*, pages 124–129. Springer, 2017.

[67] T. Maekawa, N. Yamashita, and Y. Sakurai. How well can a user's location privacy preferences be determined without using GPS location data? *IEEE Transactions on Emerging Topics in Computing*, 2014.

[68] T. Maekawa, Y. Yanagisawa, Y. Kishino, K. Ishiguro, K. Kamei, Y. Sakurai, and T. Okadome. Object-based activity recognition with heterogeneous sensors on

wrist. In *Proceedings of the International Conference on Pervasive Computing*, pages 246–264, 2010.

[69] T. Maekawa, Y. Yanagisawa, Y. Sakurai, Y. Kishino, K. Kamei, and T. Okadome. Web searching for daily living. In *SIGIR 2009*, pages 27–34, 2009.

[70] T. Maekawa, Y. Yanagisawa, Y. Sakurai, Y. Kishino, K. Kamei, and T. Okadome. Context-aware web search in ubiquitous sensor environment. *ACM Transactions on Internet Technology (ACM TOIT)*, 11(3):12:1–12:23, 2012.

[71] M. A. Mahler, Q. Li, and A. Li. SecureHouse: A home security system based on smartphone sensors. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, pages 11–20. IEEE, 2017.

[72] W. Mao, J. He, and L. Qiu. Cat: high-precision acoustic motion tracking. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 69–81, 2016.

[73] W. Mao, W. Sun, M. Wang, and L. Qiu. Deeprange: Acoustic ranging via deep learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4):1–23, 2020.

[74] R. M. McIntyre and R. K. Blashfield. A nearest-centroid technique for evaluating the minimum-variance clustering procedure. *Multivariate Behavioral Research*, 15(2):225–238, 1980.

[75] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1515–1525, 2016.

[76] J. L. Nasar and D. Troyer. Pedestrian injuries due to mobile phone use in public places. *Accident Analysis & Prevention*, 57:91–95, 2013.

[77] K. Ohara and T. Maekawa. Easy-to-install methods for indoor context recognition using wi-fi signals. In *Proceedings of the International Conference on Distributed, Ambient, and Pervasive Interactions*, pages 112–124. Springer, 2018.

[78] K. Ohara, T. Maekawa, and Y. Matsushita. Detecting state changes of indoor everyday objects using Wi-Fi channel state information. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 1(3):88, 2017.

[79] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of the 12th International Conference on Pattern Recognition*, volume 1, pages 582–585. IEEE, 1994.

[80] M. Ono, B. Shizuki, and J. Tanaka. Touch & activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 31–40, 2013.

[81] S. N. Patel, M. S. Reynolds, and G. D. Abowd. Detecting human movement by differential air pressure sensing in HVAC system ductwork: An exploration in infrastructure mediated sensing. In *Proceedings of the International Conference on Pervasive Computing*, pages 1–18, 2008.

[82] S. T. Payne. Transparent texting, Mar. 27 2014. US Patent App. 13/627,959.

[83] K. Pearson. Principal components analysis. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 6(2):559, 1901.

[84] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, pages 1–14, 2007.

[85] E. Peng, P. Peursum, L. Li, and S. Venkatesh. A smartphone-based obstacle sensor for the visually impaired. In *Proceedings of the International Conference on Ubiquitous Intelligence and Computing*, pages 590–604. Springer, 2010.

[86] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hähnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, 2004.

[87] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 32–43, 2000.

[88] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[89] L. E. Raileanu and K. Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[90] M. Rossi, J. Seiter, O. Amft, S. Buchmeier, and G. Tröster. RoomSense: an indoor positioning system for smartphones using active sound probing. In *Proceedings of the 4th Augmented Human International Conference*, pages 89–95, 2013.

[91] T. D. Rossing, F. R. Moore, and P. A. Wheeler. *The science of sound*, volume 3. Addison Wesley San Francisco, 2002.

[92] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan. Audiogest: enabling fine-grained hand gesture detection by decoding echo signal. In *Proceedings of the ACM international joint conference on pervasive and ubiquitous computing*, pages 474–485, 2016.

[93] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[94] D. A. Russell, J. P. Titlow, and Y.-J. Bemmen. Acoustic monopoles, dipoles, and quadrupoles: An experiment revisited. *American Journal of Physics*, 67(8):660–664, 1999.

[95] R. Saffoury, P. Blank, J. Sessner, B. H. Groh, C. F. Martindale, E. Dorschky, J. Franke, and B. M. Eskofier. Blind path obstacle detector using smartphone camera and line laser emitter. In *Proceedings of the 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, pages 1–7. IEEE, 2016.

[96] M. R. Schroeder. Integrated-impulse method measuring sound decay without using impulses. *The Journal of the Acoustical Society of America*, 66(2):497–500, 1979.

[97] S. Shen, H. Wang, and R. Roy Choudhury. I am a smartwatch and i can track my user's arm. In *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*, pages 85–96, 2016.

[98] S. Shi, S. Sigg, and Y. Ji. Passive detection of situations from ambient FM-radio signals. In *the 2012 ACM Conference on Ubiquitous Computing (UbiComp 2012)*, pages 1049–1053, 2012.

[99] T. Shiraishi, N. Komuro, H. Ueda, H. Kasai, and T. Tsuboi. Indoor location estimation technique using uhf band rfid. In *Proceedings of the 2008 International Conference on Information Networking*, pages 1–5. IEEE, 2008.

[100] T. Sztyler and H. Stuckenschmidt. Online personalization of cross-subjects based activity recognition models on wearable devices. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 180–189. IEEE, 2017.

[101] M. Tachikawa, T. Maekawa, and Y. Matsushita. Predicting location semantics combining active and passive sensing with environment-independent classifier. In *UbiComp 2016*, pages 220–231, 2016.

[102] M. Takagi, K. Fujimoto, Y. Kawahara, and T. Asami. Detecting hybrid and electric vehicles using a smartphone. In *the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 267–275, 2014.

[103] R. Tang, G. Duan, L. Xie, Y. Bu, M. Zhao, Z. Lin, and Q. Lin. Static obstacle detection based on acoustic signals. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2. IEEE, 2022.

[104] Y. Tange, T. Konishi, and H. Katayama. Development of vertical obstacle detection system for visually impaired individuals. In *Proceedings of the 7th ACIS*

International Conference on Applied Computing and Information Technology, pages 1–6, 2019.

[105] Y. Tange, S. Takeno, and J. Hori. Development of the obstacle detection system combining orientation sensor of smartphone and distance sensor. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6696–6699. IEEE, 2015.

[106] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Proceedings of the International Conference on Pervasive Computing*, pages 158–175, 2004.

[107] R. Tapu, B. Mocanu, A. Bursuc, and T. Zaharia. A smartphone-based obstacle detection and classification system for assisting visually impaired people. In *the IEEE International Conference on Computer Vision Workshops*, pages 444–451, 2013.

[108] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, pages 155–168, 2011.

[109] B. Thiel, K. Kloch, and P. Lukowicz. Sound-based proximity detection with mobile phones. In *the Third International Workshop on Sensing Applications on Mobile Phones*, pages 1–4, 2012.

[110] Y.-C. Tung and K. G. Shin. EchoTag: accurate infrastructure-free indoor location tagging with smartphones. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 525–536, 2015.

[111] Y.-C. Tung and K. G. Shin. Use of phone sensors to enhance distracted pedestrians' safety. *IEEE Transactions on Mobile Computing*, 17(6):1469–1482, 2017.

[112] T. UMETANI and Y. TAMURA. Change detection of state of indoor environment based on singular spectrum transformation of strength of wireless lan signals. In

Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications, volume 2015, pages 65–69. The ISCIE Symposium on Stochastic Systems Theory and Its Applications, 2015.

[113] A. Vahdatpour, N. Amini, and M. Sarrafzadeh. Toward unsupervised activity discovery using multi-dimensional motif detection in time series. volume 9, pages 1261–1266, 01 2009.

[114] G. Valenza, M. Nardelli, A. Lanata, C. Gentili, G. Bertschy, R. Paradiso, and E. P. Scilingo. Wearable monitoring for mood recognition in bipolar disorder based on history-dependent long-term heart rate variability analysis. *IEEE Journal of Biomedical and Health Informatics*, 18(5):1625–1635, 2013.

[115] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th International Conference on Ubiquitous Computing (Ubicomp 2008)*, pages 1–9, 2008.

[116] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, pages 197–210, 2012.

[117] T. Wang, G. Cardone, A. Corradi, L. Torresani, and A. T. Campbell. Walksafe: a pedestrian safety app for mobile phone users who walk and talk while crossing roads. In *the Twelfth Workshop on Mobile Computing Systems & Applications*, pages 1–6, 2012.

[118] W. Wang, A. X. Liu, and K. Sun. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 82–94, 2016.

[119] Y. Wang, X. Yang, Y. Zhao, Y. Liu, and L. Cuthbert. Bluetooth positioning using RSSI and triangulation methods. In *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC 2013)*, pages 837–842, 2013.

[120] Z. Wang, S. Tan, L. Zhang, and J. Yang. Obstaclewatch: Acoustic-based obstacle collision detection for pedestrian using smartphone. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):1–22, 2018.

[121] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[122] H. Watanabe, T. Terada, and M. Tsukamoto. Gesture recognition method utilizing ultrasonic active acoustic sensing. *Journal of Information Processing*, 25:331–340, 2017.

[123] L. R. Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4), 2003.

[124] J. Wen, J. Cao, and X. Liu. We help you watch your steps: Unobtrusive alertness system for pedestrian mobile phone users. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, pages 105–113. IEEE, 2015.

[125] M. Wu, P. H. Pathak, and P. Mohapatra. Monitoring building door events using barometer sensor in smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2015)*, pages 319–323, 2015.

[126] Y. Wu, F. Li, Y. Xie, S. Yang, and Y. Wang. Hdspeed: Hybrid detection of vehicle speed via acoustic sensing on smartphones. *IEEE Transactions on Mobile Computing*, 2020.

[127] Q. Xia, J. Korpela, Y. Namioka, and T. Maekawa. Robust unsupervised factory activity recognition with body-worn accelerometer using temporal structure of multiple sensor data motifs. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):97, 2020.

[128] Q. Xia, A. Wada, J. Korpela, T. Maekawa, and Y. Namioka. Unsupervised factory activity recognition with wearable sensors using process instruction information.

Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 3(2):60, 2019.

[129] Y. Xie, F. Li, Y. Wu, S. Yang, and Y. Wang. Hearsmoking: Smoking detection in driving environment via acoustic sensing on smartphones. *IEEE Transactions on Mobile Computing*, 2021.

[130] Q. Xu, Y. Chen, B. Wang, and K. R. Liu. Trieds: Wireless events detection through the wall. *IEEE Internet of Things Journal*, 4(3):723–735, 2017.

[131] Q. Xu, Y. Han, B. Wang, M. Wu, and K. R. Liu. Real-time indoor event monitoring using csi time series. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6393–6397. IEEE, 2018.

[132] Q. Xu, Y. Han, B. Wang, M. Wu, and K. R. Liu. Indoor events monitoring using channel state information time series. *IEEE Internet of Things Journal*, 6(3):4977–4990, 2019.

[133] W. Yang and S. Krishnan. Combining temporal features by local binary pattern for acoustic scene classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6):1315–1321, 2017.

[134] Y. Yuan, L. Wu, and X. Zhang. Gini-impurity index analysis. *IEEE Transactions on Information Forensics and Security*, 16:3154–3169, 2021.

[135] H. Zhang, W. Du, P. Zhou, M. Li, and P. Mohapatra. Dopenc: Acoustic-based encounter profiling using smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 294–307, 2016.

[136] S. Zhang, R. H. Venkatnarayan, and M. Shahzad. A wifi-based home security system. In *Proceedings of the IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 129–137. IEEE, 2020.

[137] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. In *Proceedings of the*

10th International Conference on Mobile Systems, Applications, and Services, pages 1–14, 2012.

[138] P. Zinemanas, P. Cancela, and M. Rocamora. Mavd: A dataset for sound event detection in urban environments. *Detection and Classification of Acoustic Scenes and Events, DCASE 2019, New York, NY, USA, 25–26 oct, page 263–267*, 2019.