Recognition of activities of daily living

by

Yunfei Feng

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Science

Program of Study Committee:
Carl K. Chang, Major Professor
Johnny S. Wong
Peter Martin
Jin Tian
Simanta Mitra

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2018

Copyright © Yunfei Feng, 2018. All rights reserved.
DEDICATION

I would like to dedicate this thesis to my family. I would also like to thank my friends and family for their loving guidance and financial assistance during the writing of this work. I would like to thank my advisor and labmates for their guidance and help.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>x</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>xi</td>
</tr>
<tr>
<td>CHAPTER 1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introducing the ADL Recognition System</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Organization</td>
<td>4</td>
</tr>
<tr>
<td>CHAPTER 2. ADL IDENTIFICATION BASED ON SITUATION-AWARE AUDITION</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Related Work</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Sound Categories</td>
<td>12</td>
</tr>
<tr>
<td>2.3.1 Persistent Background Noise</td>
<td>13</td>
</tr>
<tr>
<td>2.3.2 Non-Impulsive Long Sound</td>
<td>14</td>
</tr>
<tr>
<td>2.3.3 Impulsive Sound</td>
<td>14</td>
</tr>
<tr>
<td>2.4 Situation-Aware Audition Analysis</td>
<td>15</td>
</tr>
<tr>
<td>2.4.1 Audible Action</td>
<td>16</td>
</tr>
<tr>
<td>2.4.2 Audible Event</td>
<td>16</td>
</tr>
<tr>
<td>2.4.3 Audible Situation</td>
<td>17</td>
</tr>
<tr>
<td>2.5 System Overview</td>
<td>19</td>
</tr>
</tbody>
</table>
2.6 Audible Actions / Events Detection and Extraction .................................. 20
  2.6.1 Stages of an Atomic Audible Action / Event .................................. 20
  2.6.2 AE / AA extraction algorithm ...................................................... 24
2.7 HMM-based AE Recognition System ..................................................... 24
  2.7.1 System Design ............................................................................. 25
2.8 Probability-Based AE Recognition Algorithm by AA Fragmentation .......... 26
2.9 Experiments ......................................................................................... 28
  2.9.1 Experiment I: FC_MFCC Extraction .............................................. 28
  2.9.2 Experiment II: GMM-HMM Based AE Recognition ....................... 29
  2.9.3 Results and Discussion of the AE GMM-HMM Recognition .......... 31
  2.9.4 Results and Discussion of the Probability-Based AA Fragmentation on GMM-HMM Recognition .......................................................... 32
2.10 Further Experiment and Discussion ..................................................... 35
  2.10.1 Comparison with a Baseline Classification System ....................... 35
  2.10.2 AA-Based Fragmentation Contributes to Interior Sounds Classification . 37
  2.10.3 AA-based Fragmentation Improves on Another Baseline System .......... 38
2.11 Conclusions ....................................................................................... 39

CHAPTER 3. LILO: ADL LOCALIZATION WITH SMART PHONE AND VISIBLE LIGHT 41
3.1 Introduction ......................................................................................... 41
3.2 Related Work ..................................................................................... 42
3.3 System Overview ............................................................................... 45
  3.3.1 Challenges .................................................................................. 47
3.4 The Radiosity Rendering Model .......................................................... 50
  3.4.1 Discrete Radiosity Overview ........................................................ 51
3.5 Localization Based on Luminance Field Map with Trilateration ............... 53
3.6 Experiments ................................................................. 58
   3.6.1 Experimental Living Environment .......................... 59
   3.6.2 Activities of Daily Living in the Venues ................... 60
   3.6.3 Experiment 1: Localization Based on Orientation, Light Level and Time . . . 61
   3.6.4 Experiment 2.1: Recognizing ADLs Based on Orientation, Light Level and
   Time .................................................................................. 63
   3.6.5 Experiment 2.2: Comparison of Recognition Rates Using Different Attributes 67
   3.6.6 Experiment 3: Recognition in a Given Space by Different Classifiers .......... 69
3.7 Conclusions ................................................................. 71

CHAPTER 4. ONLINE WI-FI CLUSTERING ALGORITHM FOR GLOBAL LOCALIZATION ............................................. 72
   4.1 Problem Statement and Challenges .................................. 72
   4.2 Online Clustering Algorithm ........................................... 74
      4.2.1 Distance Function ..................................................... 74
      4.2.2 Similarity Measures in Large Scale ............................ 75
      4.2.3 RSSI Computation for Directional Fine Scale ................ 75
      4.2.4 RSSI Computation for Distance within A Directional Fine Scale .... 76
   4.3 Algorithm Implementation ............................................. 79

CHAPTER 5. TIME-SERIES BASED SENSOR FUSION ......................... 82
   5.1 Time-Series Sensor Fusion Model .................................. 82
      5.1.1 Time-Series Data Cleaning in the Data Level .............. 84
      5.1.2 Time-Series Single-Source Data Correction in the Information Level .... 85
      5.1.3 Time-Series Multi-Source Data Correlation in the Decision Level ....... 86

CHAPTER 6. SYSTEM DESIGN .................................................. 88
   6.1 The ADL Recognition Services Deployed Over the Cloud ............. 88
   6.2 Agent-Based Information Management Platform (IMP) ............ 89
CHAPTER 7. FULL-SCALE EXPERIMENTS ................................................. 92
  7.1 The Experimental Setup and Results ........................................... 92
  7.2 Full-Scale Experimental Results and Discussions ............................. 94
  7.3 ADL Patten Discovery .................................................................. 95
  7.4 Real-life Services on the Prototype .............................................. 96
CHAPTER 8. CONCLUSION AND FUTURE WORK ................................. 97
  8.1 Overview of Future Work .............................................................. 99
BIBLIOGRAPHY ................................................................................. 101
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Comparison of recognition accuracy to previous work</td>
<td>10</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Categories of extracted audible events</td>
<td>30</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>SDL recognition performance by GMM-HMM based system</td>
<td>31</td>
</tr>
<tr>
<td>Table 2.4</td>
<td>SDL recognition performance by GMM-HMM based recognition and probability-based prediction system</td>
<td>33</td>
</tr>
<tr>
<td>Table 2.5</td>
<td>Recognition accuracy comparison of HMM-based AE recognition system and classification systems provided in Piczak (2015)</td>
<td>36</td>
</tr>
<tr>
<td>Table 2.6</td>
<td>AA-based fragmentation contributes to higher recognition accuracy in the baseline machine classification system provided in Piczak (2015)</td>
<td>37</td>
</tr>
<tr>
<td>Table 2.7</td>
<td>AA-based fragmentation contributes to higher recognition accuracy in the baseline machine classification system provided in Mesaros et al. (2016)</td>
<td>39</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Comparison of recognition accuracy to previous work</td>
<td>46</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Photometry of luminaire</td>
<td>59</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>List of the recorded ADL scenes, and number of recording for each.</td>
<td>64</td>
</tr>
<tr>
<td>Table 3.4</td>
<td>Confusion matrix for 21 scenes classified using Bayesian Network</td>
<td>66</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure 2.1</th>
<th>Waveform of a short audio excerpt</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.2</td>
<td>Hierarchical relationship across three domains</td>
<td>18</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Situation-aware audition system diagram</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>a: &quot;Baseline&quot;, ”onset&quot;, ”attack&quot;, ”first peak”, &quot;climax&quot;, &quot;transient&quot; and ”decay tail&quot; in an audible action $AA$; b: Stages in a bell ringing sound $(AE)$, containing a &quot;sustain&quot; stage.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Thrice $AA$s of a knife hitting on a chopping board in this &quot;cutting&quot; $AE$.</td>
<td>27</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>a: Waveform for a finger snap sound; b: Envelope of first coefficient of MFCC; c: Featured point extraction of the envelope $FC_{MFCC}$; d: Density of featured points; e: Extraction for one non-impulsive long sound, where the audible action is throwing a key on a table. f: Extraction for one non-sense impulsive sound from system noise, which also belongs to impulsive sound.</td>
<td>29</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>The system architecture of LiLo</td>
<td>48</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>The luminance field map of a bedroom derived from the Radiosity method.</td>
<td>49</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Form factor geometry</td>
<td>52</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>The device coordinate system and the world’s coordinate system</td>
<td>53</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>A scenario with a set of luminaires and a mobile terminal.</td>
<td>55</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Floor layout and luminaire placement in one apartment. The numbered positions in green circles are KAAs, and the hollow squares are the luminaire with ID numbers, with different colors meaning the different luminous intensities.</td>
<td>60</td>
</tr>
</tbody>
</table>
Figure 3.7 Real-life data captured. The squares with different colors are the testing points with different illumination. .......................... 62

Figure 3.8 Recognition comparison using different attributes. "g" represents the geomagnetic information; "o" represents the orientation information. .......................... 63

Figure 3.9 Comparison of recognition rates using different combinations of attributes. In the xlabel, "angle" represents the orientation azimuth angle. .......................... 68

Figure 3.10 Recognition comparison in bedroom and living room under different kind of classifiers. "BayesNet" represents using a Bayesian network classifier and "J48" represents using a decision tree classifier. .......................... 70

Figure 4.1 Localizing phones among multiple access points. .......................... 76

Figure 4.2 Distance difference with RSSI computation .......................... 77

Figure 5.1 Time-series sensor fusion cascade model with error correction .......................... 82

Figure 5.3 Error correction framework of time-series based sensor fusion. .......................... 86

Figure 5.2 Accelerometer and orientation data (pitch, roll, azimuth) in various attitudes. 87

Figure 6.1 Architecture of ADL recognition system on information management platform 90

Figure 7.1 Floor layout of one apartment in experiment .......................... 93

Figure 7.2 Confusion matrix of activity classification .......................... 94

Figure 7.3 Visualization of ADL recognition and patterns .......................... 95
ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to my advisor and committee chair, Professor Carl K. Chang, and my committee members, Professor Johnny S. Wong, Professor Peter Martin, Professor Jin Tian, and Dr. Simanta Mitra, for their innovative guidance, tremendous patience, constructive suggestions and enormous support throughout this research and the writing of this thesis. Their insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank all my colleagues in the Smart Home Lab and the Software Engineering Lab for their suggestions and help on my research work. I would like to acknowledge the tremendous support provided by Professor Mu-chun Su of National Central University who helped conduct a full-scale experiment and collect a significant amount of data. Finally, thanks to my families for their encouragement, support and love all the time.
ABSTRACT

Activities of daily living (ADL) are things we normally do in daily living, including any daily activity such as feeding ourselves, bathing, dressing, grooming, work, homemaking, and leisure. The ability or inability to perform ADLs can be used as a very practical measure of human capability in many types of disorder and disability. Oftentimes in a health care facility, with the help of observations by nurses and self-reporting by residents, professional staff manually collect ADL data and enter data into the system.

Technologies in smart homes can provide some solutions to detecting and monitoring a resident’s ADL. Typically multiple sensors can be deployed, such as surveillance cameras in the smart home environment, and contacted sensors affixed to the resident's body. Note that the traditional technologies incur costly and laborious sensor deployment, and cause uncomfortable feeling of contacted sensors with increased inconvenience.

This work presents a novel system facilitated via mobile devices to collect and analyze mobile data pertaining to the human users’ ADL. By employing only one smart phone, this system, named ADL recognition system, significantly reduces set-up costs and saves manpower.

It encapsulates rather sophisticated technologies under the hood, such as an agent-based information management platform integrating both the mobile end and the cloud, observer patterns and a time-series based motion analysis mechanism over sensory data. As a single-point deployment system, ADL recognition system provides further benefits that enable the replay of users’ daily ADL routines, in addition to the timely assessment of their life habits.
CHAPTER 1. INTRODUCTION

The term Activities of Daily Living (ADL) refers to those routine activities a person can finish without external help Investopedia (2016). They are mainly categorized into six groups, namely eating, bathing, dressing, toileting, transferring (like walking) and continence. The extent to which people can persistently perform ADL serves as an important measure to decide their functional status and effectiveness of long-term care. Further more, Instrumental Activity of Daily Living, or IADL, pertains to complicated skills acquired since teenage years that facilitate independent living. Example of IADLs include shopping, driving, and cooking, to name a few.

Normally, staff members at healthcare centers need to collect residents’ ADL data on a 24/7 basis. Current off-the-shelf tracking tools hosted on smart health devices like smart watches, smart wristbands, walking activity trackers, etc., provide less than sufficient functionalities to measure up to their potential in regard to dealing with complex ADLs. Most on-the-market activity trackers only monitor basic fitness data such as calories burned, steps taken, miles run, heart rate achieved, etc.

To tackle these difficulties, researchers have proposed ideas including deployment of a good number of traditional sensors across a smart home environment to serve detection purposes. Examples include installing door contacts Fleury et al. (2009) Sehili et al. (2012), infra-red sensors Fleury et al. (2009), button sensors over lamp switches, faucets and microphones Fleury et al. (2009) Sehili et al. (2012) Chahuara et al. (2012), to detect the uses of such home-based facilities. Other examples take advantage of advanced computer vision techniques to recognize video images. Kinect, integrated audio sensors, high quality motion-tracking cameras as well as infrared projectors all fall into this category. This way, the detection of human activities is made easier to achieve.
Despite all these advances and trials, the set-up overhead and deployment costs of equipment, which is usually unportable, remain high. A noticeable step towards cost saving is Body Sensor Network (BSN) Yang and Yacoub (2006) that resorts to body-contact sensors, e.g., chest nodes, finger nodes, waist nodes Zhu and Sheng (2009), to increase the portability of sensor deployment. However, users will have to be covered with sensors, a feeling usually tantamount to uncomfortably burdensome.

1.1 Introducing the ADL Recognition System

To provide those who would like to participate in the smart living project with better user experiences, at the same time enjoying the major benefits of state-of-the-art ADL monitoring advances, we present an ADL recognition system Feng et al. (2017). It characterizes a single-point smart phone based mobile application at the front end, called ADL Recorder App Feng et al. (2016). The back end support includes sophisticated data fusion and recognition analysis deployed on the cloud. Contrast with BSN, the ADL Recorder App has been designed to significantly free the App users from excessive physical burden.

The standard devices bundled with modern smart phones, in particular the various sensors embedded in the phone, directly benefit our ADL Recorder App. The App launches a background service that commands the embedded sensors to capture the human users’ multi-modal data reflecting their behavioral, such as speaking or singing, and environmental contexts. Once launched, the App works for its entire duration in the background - keeping its promise not to add uncomfortable physical burden to users.

The data flow consumed by the ADL Recorder App starts with the various embedded sensors, including microphone, Wi-Fi scan module, orientation of the heading of phone, light proximity, step detector, accelerometer, gyroscope, magnetometer, timestamp, etc. The ADL Recorder App makes good use of the fact that for most of the time, mobile device embedded sensors are closely carried by human users. The close-to-zero gap between the sensors and the App users enormously
improves the chance of collecting accurate raw data from the surrounding environment the human user intimately perceives.

The instantly collected multi-modal data via the single-point deployed ADL Recorder App then goes through the stages of data pre-processing, data analysis and data fusion at the back-end, i.e., over the cloud, of the ADL Recognition System. To conclude that the user walked up a stairway, for example, the system needs a temporal sequence of that user’s accurate positioning data. Complex problems remain. What if the user turns off her GPS module, accidentally or purposefully? What if the user only moves indoor at her less than spacious home (Q1)? If a user stays in one location but constantly moves her arms, or even changes her head position, can these kinds of tiny motions be detected (Q2)? If a user is typing on a computer keyboard or turning on a microwave oven, can this kinds of motions be detected and made sense of (Q3)? Needless to say, these are all important problems to solve in order for the system to obtain the ADL details of the user.

For question Q1 and Q2, the ADL recognition system offers robust positioning features and complies with the Global System for Module Communication (GSM) standard. In addition to supporting GPS, its enhanced outdoor positioning solution also combines the Wi-Fi fingerprinting technology supported by localization classifier Oguejiofor et al. (2013) Joo-Yub Lee and So (2013). For indoor positioning issues, the ADL Recognition system employs a light-based indoor positioning algorithm to improve the positioning precision down to the room-level and furniture-level. The ADL recognition system, at its back-end, performs data fusion operation on various sensory data from orthogonal modality dimensions. Data fusion algorithms help to decide the user’s activities ranging from running, walking, standing still, sitting, lying, taking an elevator, etc.

Automatic environmental sound recognition Wang et al. (2008) has been integrated into the ADL recognition system’s back-end recognition server. After the front-end ADL Recorder App records the environmental sound as an audio file for each session and transmits it to the recognition server, acoustic features are extracted out of those audio files. Based on those acoustic features, the raw audio files are indexed in the audio feature database and are classified into different activity categories. A state-of-the-art hierarchical situation-aware Chang (2016) Ming et al. (2015), audition
algorithm for ADL identification has also been implemented. The real-world experiments indicate that our system can be used to recognize audible events, audible actions (e.g. cutting vegetables and stirring eggs), remote audible events as well as environmental sound with high accuracy. This answers question Q3.

1.2 Organization

This dissertation is organized as follows: Chapter 2 introduces the situation-aware audition theory for fine-grained ADL and experiments. Chapter 3 presents an approach of ADL localization with smart phone by visible light. Chapter 4 describes the online WI-FI clustering algorithm for global localization. Chapter 5 demonstrates the cascade model for time-series based sensor fusion with error correction feature. Chapter 6 illustrates the system design and introduces an Agent-based Information management platform that hosts the ADL recognition server. Chapter 7 reports the overall experiments for ADL recognition and ADL pattern discovery. Chapter 8 concludes our research contributions and proposes future work.
CHAPTER 2. ADL IDENTIFICATION BASED ON SITUATION-AWARE AUDITION

2.1 Introduction

As we continue developing a large array of computing services as part of the Internet of Things (IoT), it is imperative to first acquire a deeper understanding of human activities of daily living (ADLs) Opara (2012) for those receiving such services. For example, sensor data enables an IoT system to monitor elderly rehabilitation Bisio et al. (2017). Human ADLs are intimately embedded in the physical space and our very nature makes us socially engaged. This paper focuses on automatic identification of ADLs pertaining to IoT as an important basis to improve the quality of experience (QoE) and promptness when providing computer-based services in various forms.

ADLs are things people normally do during their day-to-day routines, including eating, bathing, dressing, grooming, working, homemaking, leisure, etc. The ability or inability to perform ADLs can be a very practical measure of a person's capabilities when suffering from certain types of medical disorders Lazarou et al. (2016). Oftentimes in a health care facility, with the help of observations by nurses and self-reporting by residents, professional staff collects and manually enters ADL data into a documentation system.

Some smart healthcare technologies have already been applied to detect ADLs. For instance, flexible touch sensors, embedded in a mattress, detect the time when a bed is in use. Other examples include: door contacts, infrared sensors, button sensors that detect the use of utilities, and surveillance camera that captures the ADLs after video processing, microphones set into the ceiling that record in real-time Sehili et al. (2012). There are drawbacks to these conventional solutions Feng et al. (2016). Many devices are both expensive and labor intensive as it requires installation in every room where sensing is required. Also, these methods can infringe on privacy. Smartphones can act as an intermediary between body sensors and the Web Kamilaris and Pitsillides (2016).
We developed an ADL Recorder App Feng et al. (2016) Feng et al. (2017) to recognize ADLs with sensor fusion from multiple different sensor sources in one smartphone. It supports early detection of abnormal situations, remote monitoring, and promotion of safety and well-being. This App and the back-end recognition system is among the first mobile applications to recognize ADLs using only a single standard smartphone. Whereby, we use the microphone on a smartphone as one kind of sensor to record surrounding sounds. Oftentimes ADLs create sounds alongside, and sounds are reflections of the elements in the environment as well. The sound that an object makes can be mapped back to the action the human took. For example, a fan turns on when someone flips its switch, and the rotating fan blades transmit sound into the environment. The smartphone is able to record the audible sounds that are audible to human ears. This study aims to recognize ADLs through the sounds of daily living (SDL), where ADLs are from the perspective of human and SDLs are from the perspective of physical objects. We assume that SDL represents a portion of the scene of ADL. In this case, our App interacts with human’s behaviors and transfers data over a network to the ADL Recognition System Feng et al. (2017), which then sends the ADL history information back to reviewers, thus completing the human-to-human and human-to-computer interaction in the IoT setting.

Some household event sounds can be weak, such as the sound of tearing aluminum foil, flipping a newspaper, etc. In order to pinpoint such ADL events, it is very necessary to filter out blank periods and trivial sound segments in order to reduce the computational load. To achieve this, we present a novel hierarchical situation-based architecture, combining several audible event detection and identification techniques, specifically for household sounds classification.

In order to improve the recognition accuracy, we first detect and extract audible events out of the situations from this record session, and train those audible events as acoustic models afterwards rather than training signals in one session as a whole. We consider one session of compound sound as an audible situation (see Sec. 2.4.3) in our ADL study. The benefit is the ability to recognize the pure audio clips at the same level, so that the key parts stand out in the audio clips. Subsequently, our recognition rate gains a better performance. A novel fragmentation technique is introduced
in our system. The fragmentation technique pinpoints when acoustic events are happening and extracts those clips for better recognition. This fragmentation technique can serve as a general processing step plugged into other algorithms, which is not dependent on the types of acoustic features and classifiers.

The acoustic event detection and classification (AED/C) system for a variety of acoustic events in meeting room environments was developed in the Computers in the Human Interaction Loop (CHIL) project CHIL (2007). The TALP Center Segura et al. (2008) developed the Acoustic Source Localization (ASL) system to obtain positions. CLEAR 2006 Stiefelhagen et al. (2007) and CLEAR 2007 Stiefelhagen et al. (2008) are two international evaluation campaigns to evaluate AED tasks. Our study presents a general audible event detection and identification technique specifically for detecting featured household sounds, such as chopping, turning on faucet and placing a pan on a counter.

ADLs correspond to the fulfillment of human goals. Goal decomposition into sub-goals enables a sequence of actions to map to different hierarchical situation-awareness levels. Since our work helps to dig the situation-aware information from the SDL, recognizing SDL in hierarchical situation-tiers propels the inference of actual ADL goals.

\subsection{2.2 Related Work}

Mel-scale frequency cepstral coefficient (MFCC), as a kind of acoustic features, is commonly used for speech/speaker recognition Lu (2001), because it considers human perception sensitivity with respect to frequencies. For example, Muda et al. (2010) used MFCC feature extraction and Dynamic Time Warping (DTW) techniques for feature matching. On the other hand, this study focuses on the non-speech SDL recognition and classification. Note that, compared to speech (human voice), SDL signals have broader frequency band and less temporal characteristics. Typically, researches gain better performance by leaving out the first coefficient of MFCC (FC\_MFCC). It is because the first coefficient relates to the magnitude of power Steidl et al. (2009) , usually much larger than other coefficient, causing distortion in computation. Our research happens to benefit from using
FC-MFCC to extract and segment valuable parts from a long audio file. Therefore, MFCC feature are used for three steps of event detection, audio segmentation and feature extraction altogether.

Currently, there are a number of audio recognition systems using Hidden Markov Models (HMMs) to deal with the temporal variability, because HMM has advantages to render acoustic characteristics. Gaussian Mixture Model (GMM) Rabiner (1989) takes audio samples as scattered dots, measured from acoustic perspective, including much non-speech information. However, GMM totally lacks sequence information that can be used to produce semantic information on acoustic feature vector Reynolds and Rose (1995). It is, therefore, applicable for non-speech sound recognition problems. A simple Gaussian model is unlikely to be adequate since the feature (MFCC) data are unlikely to follow a simple normal distribution; more feasible is a mixture of Gaussians as commonly used within HMM. Consequently, we may fit each state of HMM into a short window of frames with coefficients representing the acoustic input, and train data into Gaussian mixture models. Thus in this work, GMM is used for reflecting static patterns, and HMM is used for training sequential patterns in the audio clips.

The objectives of the previous works aim diversely, and multiple kinds of combinations of acoustic features and classifiers are experimentally used. Some studies focus on the discrimination between variations of music, speech and silence Scheirer and Slaney (1997) Srinivasan et al. (1999). While Chu et al. (2009) works on the recognition of different environmental sound. Here, some performances of related environmental sound recognition are summarized in Table 2.1. Differences of audio tested in the previous work exist in the aspects of duration of the testing audio excerpts, occasions (e.g. traffic, bathroom, cafe, hallway), sound classes, and so on. Due to not testing in one common test bench, the recognition rates listed in the table are not comparable absolutely. Moreover, the classification outcomes in different levels. Eronen et al. (2006) investigates both event-level (e.g. car, bus) and occasion-level (e.g. public places and home) contexts. El-Maleh et al. (1999) classifies five environmental noise classes (babble, car, bus, factory, and street) using line spectral features and a Gaussian classifier, on a frame-by-frame basis using short segments (e.g. 20 ms) of the signal. The noise analyzed in El-Maleh et al. (1999) sometimes represents background
sound (traffic), sometimes event sound (bus), and sometimes simultaneous conversation. All the kinds of noise are fed into classified as a whole, which yields the overall error rates ranging from 20-35%.

There are still some general issues overlooked by the previous work. Most of the previous researches (Temko and Nadeu (2009) Peltonen et al. (2002)) designed experiments using isolated acoustic events in silent environment. However, contexts in the real world is more complex than these ideal ones. In practice, multiple acoustic events interweave, the acoustic features are more complicated than those recorded independently. All the data in our experiments are drawn from real-life environments.

To the best of our knowledge, the audio clips for training in most previous works are a compound sound mixing environmental sound and behavioral sound. For example, the background noise of a particular environment composed of many sound events we explained in Chu et al. (2009), which does not consider each constituent sound event individually, but as many interweaving properties of each environment. We need a finger-grained technique in order to detect ADLs in real-life settings.

Computer-based sound detection generally lacks selective attention ability of human ears, which results in a large error rate under the circumstance with mixture of overlapping audio events. The raw audio can capture general properties of all sounds including background, music, speech, and those caused by audible actions. Instead of separately estimating a GMM for each audio segment, Campbell et al. (2006) estimates a GMM for each audio segment by adapting the parameters of a universal background model (UBM). Thus, UBM becomes a trained GMM to represent all types of audio. However, the whole session contains many audio types incurring much interference, resulting in inaccuracy in the model training period.

The audio session here corresponds to an audible situation (see Sec. 2.4.3) in our ADL study. In order to improve the recognition accuracy, we first detect and extract audible events related to the situations from one session, and train them as acoustic models afterwards rather than training signals in one session as a whole. The benefit is to ensure recognize the pure audio clips at the same level, so that the key parts stand out from the testing audio clips. Accordingly, the recognition rate
<table>
<thead>
<tr>
<th>Reference</th>
<th>Features</th>
<th>Classifier</th>
<th>Recognition rate</th>
<th>Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chu et al. (2009)</td>
<td>MP with MFCC</td>
<td>k-NN; GMM</td>
<td>system = 83%; listeners = 82%</td>
<td>14 audio environments</td>
</tr>
<tr>
<td>Ebenezer et al. (2004)</td>
<td>Modified MP</td>
<td>Modified MPD</td>
<td>83%</td>
<td>12 unclear classes</td>
</tr>
<tr>
<td>Umapathy et al. (2005)</td>
<td>Time-frequency</td>
<td>LDA</td>
<td>97.60%</td>
<td>six music groups (rock, classical, country, jazz, folk and pop)</td>
</tr>
<tr>
<td>Peltonen et al. (2002)</td>
<td>MFCC</td>
<td>K-NN; GMM</td>
<td>68.4%</td>
<td>classify 17 scenes out of 26</td>
</tr>
<tr>
<td>Eronen et al. (2006)</td>
<td>MFCC</td>
<td>GMM; 1-NN; HMM</td>
<td>system for 27 contexts = 58%;</td>
<td>27 classes (e.g. street, car, restaurant, office, kitchen, hall, etc.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12 MFCC + GMM = 63%; listener = 69%;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>system for six high-level classes = 82%; listeners for high-level contexts = 88%</td>
<td></td>
</tr>
<tr>
<td>Srinivasan et al. (1999)</td>
<td>Energy; ZCR; Spectral energy in 4</td>
<td>Threshold</td>
<td>80%</td>
<td>silence, speech, and music in video</td>
</tr>
<tr>
<td></td>
<td>bands; Harmonic frequencies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zhang and Kuo (1999)</td>
<td>Timbre, rhythm</td>
<td>Segmentation; HMM</td>
<td>80%</td>
<td>10 classes, including applause, birds’ cry, dog bark, explosion, foot step, laugh, rain, river flow, thunder, and wind storm</td>
</tr>
<tr>
<td>El-Maleh et al. (1999)</td>
<td>Line spectral</td>
<td>QGC; DTC</td>
<td>86.4%, 88.1%</td>
<td>5 classes, including babble, car, bus, factory, and street</td>
</tr>
<tr>
<td>Gaunard et al. (1998)</td>
<td>LPCC</td>
<td>Discrete HMMs</td>
<td>system = 90%+; listeners = 91.8%</td>
<td>5 classes, including car, truck, mopeds, aircraft, and train</td>
</tr>
<tr>
<td>Lu et al. (2002)</td>
<td>Linear spectral</td>
<td>KNN; Quasi-GMM</td>
<td>96%+</td>
<td>speech, music, environment sound, and silence</td>
</tr>
<tr>
<td>Li et al. (2001)</td>
<td>MFCC and LPC</td>
<td>Energy</td>
<td>90%</td>
<td>noise, speech, music, silence</td>
</tr>
<tr>
<td>Study</td>
<td>Features and Techniques</td>
<td>Accuracy</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Cano et al. (2004)</td>
<td>Spectro-temporal descriptors; Bark-band energy; MFCC</td>
<td>1-NN 85%</td>
<td>6-class percussive instrument, trumpet, sax, piano, guitar, brass, trombone, violin, harpsichord</td>
<td></td>
</tr>
<tr>
<td>Herrera et al. (2002)</td>
<td>CFS, reliefF, MFCC and variances, energy bands, temporal descriptors, spectral descriptors</td>
<td>CDA; 1-NN; C4.5; PART</td>
<td>super-category level = 99%; basic level = 97%; sub-category level = 90% membranes versus plates, and some basic classes plus some sub-classes</td>
<td></td>
</tr>
<tr>
<td>Tzanetakis and Cook (2002)</td>
<td>Timbral Texture Features (FFT; MFCC) Rhythmic Content Features; Pitch Content Features</td>
<td>GMM</td>
<td>10 musical genres, including classical, country, disco, hiphop, jazz, rock, blues, reggae, pop, metal.</td>
<td></td>
</tr>
<tr>
<td>Zhang and Kuo (2001)</td>
<td>Energy, ZCR, fundamental Frequency, spectral peak tracks</td>
<td>Threshold 90.7%</td>
<td>speech, nonspeech and music</td>
<td></td>
</tr>
<tr>
<td>Temko and Nadeu (2009)</td>
<td>FFBE;ZCR; STE;4SBE;SF;</td>
<td>SVM 81%</td>
<td>15 types (e.g. door knock, steps, laugh, etc.)</td>
<td></td>
</tr>
<tr>
<td>our work</td>
<td>FFT</td>
<td>GMM+HMM</td>
<td>9 kinds of SDLs from audible events (e.g. bell, cover, drawer, etc.) or audible action (e.g. aluminumFoil, cutting, spoonScratchPlate, etc.)</td>
<td></td>
</tr>
</tbody>
</table>

could gain a better performance. Furthermore, a novel fragmentation technique is introduced in our system. Fragmentation technique pinpoints when acoustic events are happening and extracts the clips for better recognition.

We also defined audio classification resolution (ACR) quantifying the extent of detail that the classification system resolves from the audio excerpts. Higher resolution means more audio detail. For instance, provided that cooking a pizza is one of the basic-level actions, then the set of \{washing dishes, chopping, etc.\} is the super-category with a lower resolution, representing a kitchen environment where the actions are happening. And cooking’s sub-category higher resolution is for recognizing more basic elements of higher-level details, such as placing a pan onto an oven, switching on a heater, tapping water, etc. However, the dataset selected by a number of previous works Chu et al. (2009) El-Maleh et al. (1999) provides audio clips in different ACRs. The specific ACR addressed in this work is to achieve the high-ACR sound recognition with high accuracy, for example, distinguishing sound of aluminum foil from a quiet environment.

The rest of this paper is organized as follows. Section 2.3 illustrates three major sound categories in terms of duration of audible parts. Section 2.4 describes the fundamental relationship between sound waveform domain and situation-aware sound domain. Section 2.5 describes the system in a general way. Section 2.6 focuses on the progressing of an atomic audible action / event signal in depth, and proposes the extraction algorithm for valuable segments. Section 2.7 describes an audible event recognition system based on two sets of statistical models (GMMs and HMMs). A new probability-based audible event recognition algorithm is proposed in section 2.8. We present the experiment setup, employing several Android-based phones, and discuss the results in section 2.9. Finally, section 2.11 summarizes the present study and concludes with future work.

2.3 Sound Categories

The terminologies for acoustic event classification are quite confused, such as ”environmental sound Chu et al. (2009)”, ”background noise El-Maleh et al. (1999)”, ”environmental noise Gaunard et al. (1998)”, etc. While some terms used diversely represent the same problem, some works Chu
et al. (2009) consider event sound as an environmental sound. In this paper, we need to first clearly define the categories in order to convey the essential concepts introduced in this paper.

Our living environment is full of various types of sound every day. At home, we experience sound from movement, conversation, television and radio, usage of household appliances, etc. On the road, we are exposed to noises, such as construction, traffic and automobile horn.

In this research, we broadly categorize the natural sound into three types in terms of time duration and amplitude. They are persistent background noise (PBN), non-impulsive long sound (NILS), and impulsive sound (IS).

2.3.1 Persistent Background Noise

Persistent background noise (PBN) is a kind of perpetual background sound. Such continuous sound always lasts over an extended period of long-term time, such as noise generated in a copy / printing center, sound from heater fan, air conditioner, mist vaporizer, vacuum cleaner and kitchen hood fan.

Definition 2.3.1 In one time period of SDL signal $S$, the PBN parts are set \( \text{PBN} = \bigcup_i \{ S_i \mid D_i == \text{length}(S_i) \text{ and } \text{ave}(A_i) \leq \text{upperbound}_A \} \), that is to say, the PBN is always long-lasting in the environment.

where $D_i$ is the duration of signal $S_i$, $A_i$ is the amplitude of signal $S_i$, upperbound$_A$ is the upper bound of amplitude, as such

$$20 \log \frac{\text{upperbound}_A}{\text{max(\text{amplitude}(S))}} dB \leq -12 dB \tag{2.1}$$

We can use these features to predict the place where actions take place, so set \( \text{PBN of interest} \) = \( \bigcup_i \{ S_i \mid D_i \geq \text{Threshold}_D \text{ and } \text{ave}(A_i) \leq \text{upperbound}_A \} \), where Threshold$_D$ is the threshold of time duration, generally Threshold$_D \geq 12$ seconds. In addition, we use a fixed time period for sound sampling. It is possible that several events may occur within one-time period.
2.3.2 Non-Impulsive Long Sound

The second type of sounds refers to the non-impulsive long sounds (NILS), such as an explosion, bell ringing, and is short yet audible. Such non-impulsive sounds are not always hidden behind the background PBN, and are longer than the impulsive sound. In general, NILS comes from either human behavior context or environmental context Chang et al. (2009). Mostly, NILS occurs along with some events, thus we can figure out audible actions or events involving humans, environments and their interactions. For example, running water sounds indicate that someone is doing washing, and sounds from moving utensils are clues of people’s actions.

**Definition 2.3.2** In one time period of SDL signal $S$, the NILS parts are set(NILS) = $\bigcup_i \{ S_i \mid \text{Threshold}_D \leq D_i \leq \text{length}(S_i) \text{ and } \text{ave}(A_i) \geq \text{upperbound}_A \}$, generally Threshold$_D$ takes $[2,12)$ seconds.

In this research, we use these features to conjecture people’s activities of daily living (ADL).

2.3.3 Impulsive Sound

Impulsive sound (IS) refers to a category of unexpected, almost instantaneous (thus impulse-like) sharp sounds (like clicks and pops). Such an intense ”impulse” sound usually occurs once in a very short time interval, atomically lasting less than 1 second. And some of those are punctuating, such as system noise (electromagnetic interference). Frequency energy of some impulsive sounds are over a large portion of the acoustic spectrum, such as a hammer blow or hand clap.

**Definition 2.3.3** In one time period of SDL signal $S$, the IS parts are set(IS) = $\bigcup_i \{ S_i \mid 0 \leq D_i \leq \text{Threshold}_D \text{ and } \text{ave}(A_i) \geq \text{upperbound}_A \}$, where, normally, the Threshold$_D \leq 1$ second. Mostly, IS happens within a NILS session.

When someone exerts a force on an object, we assume that the reaction of the object immediately emits audible signal and lasts for a period of time in some cases. And the lasting part gives the new state of the object. Thus, NILS often represents the new state of an object. For instance,
water starts running after someone turns on the faucet, and the NILS of running water represents the state of faucet turned on. The NILS ends when the state of faucet reverts to off.

In the real world, such three major sound categories interleave with each other, because events happen together as the time progresses. Fig. 2.1 is a waveform of a sound excerpt. The whole audio file is recorded in a kitchen, which contains many featured sounds as the subject was cooking a meal. The person cooked as usual, conducting many actions meanwhile, such as placing a baking pan onto oven, taking a big bowl from drawer, etc. We extracted a four-second sound clip out of the entire cooking audio file and computed the wave diagram. In Fig. 2.1, the blue part is background noise, which belongs to PBN. The green part, from stirring eggs, is one kind of NILS. The yellow part of the sound file is bell ringing, lasting around one second, which also belongs to NILS. Finally, the red part of sound is from jingling a spoon and a dish, lasting around ten milliseconds, which belongs to IS.

![Figure 2.1 Waveform of a short audio excerpt](image)

### 2.4 Situation-Aware Audition Analysis

Levesque et al. (1998) defines situation as "histories" which are finite sequences of primitive actions. Similarly, a set of contexts in a system over a period of time is considered as a situation Yau et al. (2008), which can be atomic, logical composite or temporal. A situation here is a sequence of actions under a certain environment. When a user is taking his actions in sequence,
the sound happening simultaneously reflects the changing situations. As such, one action may introduce new auditory information into the earlier auditory environment. By situation-awareness of auditory information, the sequence of actions can be revealed. In this section, we leverage the sound happening along situations to decompose one session of auditory information into several audible events and audible actions, discovering the situation information based on auditory characteristics, and helping perceive a user’s demands. Here an audible action is considered as an atomic situation, an audible event is considered as a logical composite or temporal situation, and an audible situation is the composite situation under a certain auditory environment.

2.4.1 Audible Action

**Definition 2.4.1** An audible action (AA) is an atomic action carried out by people, and is mostly accompanied by perceptible sound in short time intervals.

Our project focuses on key nontrivial AA, such as unpredictable slamming door, stepping heel, snap fingers, etc. Usually, an IS corresponds to one AA, where one AA just has one crest of the wave. While in reality, there can be a certain single AA that contains more than one crest of the wave, such as a jar cover keeps fluctuating after being placed on a table. Therefore, each crest is considered as one micro-IAA.

2.4.2 Audible Event

**Definition 2.4.2** An audible event (AE) is the representation of the sound effect due to human presence, or due to sound of objects and nature.

An audible event may contain one audible action by one person, or a number of audible actions from more than one person. This work focuses on the single agent domain. Acoustic Event Detection (AED) Zhuang et al. (2010) aims to identify both timestamps and types of events in an audio stream.

\[
AE = (\cup_i AE_i) \bigcup (\cup_j \overline{AE_j})
\] (2.2)
An audible event (AE) comprises a sequence of AAs along the time domain, related to human behavioral context. A specific example AE is the frying eggs that contains multiple times of pan touching with a cooking shovel, and each of which is an AA. In this scene, ISs happen in a composite way. Each touching IS is an audible action and the session of ISs burst is an audible event. The first type of AE is the continuous sequence of AAs, which reside in one session of AE.

\[ AE_i = sequence\{AA_i^j, AA_i^{j+1}, AA_i^{j+2}, \ldots, AA_i^{j+e}\} \] (2.3)

Another type of audible event, environmental context related, is the audible state of an object. For example, running water is the audible state of a faucet, and ringing is that of a bell.

\[ \overline{AE_i} = \text{audibleState}(object) \] (2.4)

Note that \( \text{audibleState} \) is a mapping between the state variable (object) to a set of state values such as \{busy, idle\} for the object is a faucet, or \{ringing, silent\} for the object is a phone.

### 2.4.3 Audible Situation

In terms of ADL recognition, it often relates human subject’s actions to the daily goals to be accomplished, such as personal hygiene, feeding, entertainment, etc. Moreover, goals can be decomposed into several sub-goals.

In general, much of the SDL information is composed of several simultaneous audible events that lead to a complex audio environment, and increasing the difficulty in audio retrieval and analysis. An audible situation is even more complex as it is normally mixed by all of three different kinds of sounds. Thus, we give the following definition.

**Definition 2.4.3** If there is at least an audible event happening in a named situation (i.e., situation pertinent to the application), then we say this is an audible situation (AS).

\[ AS = \{A, E\} = \cup_i AS_i = \cup_i \{\cup_j AE_i^j, E_i\}_{\Delta t} \] (2.5)

where each AE completes a sub-goal for the time being. \( A \) is a set of the users’ actions to achieve a goal, and \( E_i \) is a set of context values with respect to a subset of the environmental context variables during a period of time \( \Delta t \) when the sequence of AEs is completed.
Now that goals can be decomposed into several levels of sub-goals, \( AS \) can be also decomposed into several levels of \( AE_i \) and \( E_i \) appropriately. Suppose in a named audible situation \( (AS_i) \) of making salad in the kitchen, it contains many high-ACR \( AE \) sessions, such as taking tomatoes from a plastic bag, cutting carrots, stirring salad and so on. And the environmental context \( E \) is in the kitchen. Therefore, we can predict the auditory scenes, in reference to a location with different acoustic characteristics such as a printing center, on a bus or quiet hallway, from an audible situation.

On the scale of ACR, \( AA \) has a higher resolution than \( AE \), which has a higher resolution than \( AS \).

Our belief is to utilize sound information as much as possible. Environmental contexts help predict the place and environment that subjects reside in. In any audible situation, environmental contexts can be retrieved by background sound or \( AE \), and behavioral contexts can be captured and recorded as the sound snippets of \( AE \).

![Figure 2.2 Hierarchical relationship across three domains](image_url)

The hierarchical analysis is carried out from low-level SDL layer through situation-awareness process, to high-level ADL situation layer, as shown in Fig. 2.2.

**Definition 2.4.4** Let \( P \) be the projection function from the sound waveform (SDL) domain to the situation-aware process domain.
Definition 2.4.5 Let \( B \) be the background function from the sound waveform (SDL) domain to the environmental context of an ADL situation.

Definition 2.4.6 Let \( R \) be the recognize function from the situation-aware sound domain to the ADL situation domain.

Definition 2.4.7 Let \( I \) be the goal inference function existing in the situation-aware process domain and ADL situation domain.

\[
\begin{align*}
P(PBN + NILS + IS) &= AS \\
P(NILS) &= AE \\
P(a \text{ sequence of } ISs) &= AE \\
P(IS) &= AA
\end{align*}
\]

\( B(PBN) = E = \text{environmental context} \) \hspace{1cm} (2.6)

\[
\begin{align*}
R(AE) &= E = \text{environmental context} \\
R(AE) &= \text{a sequence of actions} \hspace{1cm} (2.8) \\
R(AA) &= \text{an atomic action}
\end{align*}
\]

\[
\begin{align*}
I(a \text{ sequence of } AAs) &= \text{goal}(AE) \\
I(a \text{ sequence of } AEs, E) &= \text{goal}(AS)
\end{align*}
\]

2.5 System Overview

In Chu et al. (2009), classification is conducted across 14 different audio environments, such as restaurant, street, thundering, waves, etc. Specifically, from our perspective, the environments are not in the same ACR. ACRs in different levels correspond to different sound categories defined previously. Thundering belongs to NILS; waves belong to PBN; restaurant is an AS which composes the noisy, quiet, with-music PBN. It also composes some NILS in AE level, such as moving chairs, slamming a door. It even composes much sound of spoon/cup jingle, which is an IS in AA level.
We distinguish between environmental sound and behaviors sound clearly, concentrating on the behaviors sound branch. The major task of this project is to extract NILS and IS out of a long audio sample by inventing the function \( P \) (Eq.2.6.(2)(3)(4)), applying \( R \) (Eq.2.8) to recognize the actions under a certain situation, and eventually \( I \) (Eq.2.9.(1)) to infer the sub-goals or goals of ADLs.

In this work, we have first accomplished the task of audible events detection and extraction from a long SDL audio file, which is audible situation \( (AS) \). The acoustic feature used is first coefficient in MFCC, which is newly proposed. Subsequently, in order to improve the recognition accuracy of audible events, a technique of fragmenting \( AE \) files into several audible action \( (AA) \) snippets is applied. Pre-emphasizing the high-frequency region of \( AA \) samples are processing to enhance the acoustic characteristics. The combination of GMM and HMM classification computes on each audible action snippet and yields a potential type outcome. At last, the synthesis \( AE \) classification is determined by the \( AA \) type with the largest probability. The schematic diagram of the proposed methodology has been shown in Fig. 2.3.

2.6 Audible Actions / Events Detection and Extraction

The extraction process is operated on the first coefficients of MFCC \( (fc.mfcc) \) array. The key points of \( fc.mfcc \) is converted into corresponding positions in a real wave file, according to which audible action / event clips are extracted. Every element in this array corresponds to a frame from the audio clip.

2.6.1 Stages of an Atomic Audible Action / Event

Basically, an isolated \( AA \) is composed of six stages: baseline, onset, attack, peak, transient and decay tail, as the waveform of a simple case shown in Fig. 2.4(a).

1. The baseline is the base sound prior to and posterior to the \( AA \) session, whereas \( v(baseline) \) is the value of base sound, whose acoustic features are used to establish environmental context.
2. The *onset* of the AA is the earliest moment when the AA occurs, which coincides with the starting point of the *transient*.

The height of a frame is its amplitude difference beyond the *baseline*.

\[
h(f_i) = v(f_i) - v(\text{baseline})
\]  
(2.10)

where \(i\) is the index of a frame, \(h(f_i)\) denotes the referenced amplitude level of the \(i\)-th frame, and \(v(f_i)\) denotes the amplitude of \(i\)-th frame.

The jump height of a frame is the rising amplitude difference compared to the previous frame.

\[
\Delta h(f_i) = h(f_i) - h(f_{i-1}), i = 1, 2, \ldots, \text{framesize}.
\]  
(2.11)
Figure 2.4  a: "Baseline", "onset", "attack", "first peak", "climax", "transient" and "decay tail" in an audible action AA; b: Stages in a bell ringing sound (AE), containing a "sustain" stage.

\[ F(i) = \Delta h(f_i) - \text{jumpHeightThreshold}. \quad i = 1, 2, \ldots, \text{framesize}. \]

where \text{jumpHeightThreshold} is a threshold for onset detection. The larger \text{jumpHeightThreshold} is, the fewer number of points of interest are obtained. It iterates through each frame and computes \( F(i) \), and the index of the first frame makes \( F(i) \geq 0 \), where the onset is detected. So,

\[ (\Delta h(f_{\text{onset}}) - 1) < \text{jumpHeightThreshold} \quad \text{and} \quad (\Delta h(f_{\text{onset}}) \geq \text{jumpHeightThreshold}) \]

3. The \textit{climax} is the maximum amplitude within the whole AA session.

\[ i(\text{climax}) = \arg \max_{f \in S}(f_{\text{c}_m \text{f}_c(f)}) \quad (2.12) \]

where \( i(\text{climax}) \) is the index of the climax in the \textit{first coefficient in MFCC} vector within the whole AA / AE; \( S \) is the set of MFCC frames over the whole AE session.

\[ h(\text{climax}) = v(\text{climax}) - v(\text{baseline}) \quad (2.13) \]

A large climax reflects when a significant pitch occurs, so that human ears can easily detect and recognize those. On the other hand, from the audition computation perspective, those larger amplitude regions with larger power have more distinct characteristics.
4. The peak is the highest point over a certain region.

\[ i(\text{peak}) = \arg \max_{f \in \Delta S}(f_{\text{c.mfcc}}(f)) \] (2.14)

where \( \Delta S \) is the \( f_{\text{c.mfcc}} \) of a small limited excerpt in the whole AE session, \( \Delta S \subseteq S \).

5. For an AA or AE signal, the attack of the AA is the process during which the amplitude envelope increases from the baseline to the first peak.

\[
\begin{align*}
    i(\text{first peak}) &= \min \{i(\text{climax})\} \\
    \{ i(\text{attack starting}) &= i(\text{onset}) \\
    i(\text{attack ending}) &= i(\text{first peak}) \}
\end{align*}
\] (2.15)

\[
\begin{align*}
    i(\text{attack starting}) &= i(\text{onset}) \\
    i(\text{attack ending}) &= i(\text{first peak})
\end{align*}
\] (2.16)

6. The period of transient can be reliably detected after onset. The process of attack is embedded within the transient, whose interval begins at onset, ends until the amplitude decreases to an attenuating ratio.

\[ 20 \log \frac{h(\text{transient ending})}{h(\text{climax})} = G_{dB} \] (2.17)

where \( G_{dB} \) is the amplitude ratio or gain in dB, usually set as -6 dB.

\[ i(\text{transient starting}) = i(\text{onset}) \] (2.18)

From Eq. 2.17, the index of transient ending is

\[ i(\text{transient ending}) = \arg i(h(\text{peak}) \cdot 10^{G_{dB}/20}) \] (2.19)

The transient ending comes after the peak.

\[ i(\text{transient ending}) > i(\text{peak}) \] (2.20)

Signal in the transient stage plays the most significant role in retrieving representative acoustic features.

7. An isolated AE may contain sustain within the transient interval. The sustain exists when within a time duration \( T_{sus} \), the amplitude levels are always maintained upon a certain level,
displayed in Fig. 2.4(b). The $T_{sus}/t$ gives the number of MFCC frames, where $t$ is the frame blocking size.

$$
\sum_{j}^{j+T_{sus}-1} h(j) \geq \frac{T_{sus}}{t} \times \text{jumpHeightThreshold}
$$

(2.21)

8. The decay tail is a slow decaying period after the transient of the sound session. And the lengths of decay tails vary. Some signal even has very short decay tail, almost reducing sharply.

$$
\mathcal{L}(offset) = [v(\text{climax}) - \sum_{f=it}^{it+T_{sl}} \frac{v(f)}{T_{sl}}] - (1 - \epsilon) \times h(\text{peak})
$$

(2.22)

where $i_t$ is $i(\text{transient_ending}) + 1 + offset$, offset is used for deriving the index when $\mathcal{L}$ value begins to be positive.

$$
i(\text{decay_tail_ending}) = i_t + T_{sl}
$$

(2.23)

where $T_{sl}$ is the sliding window size, and we compute the average amplitude in this window. Note that in Eq. 2.22, $\epsilon$, ranging from $(0,-G_{dB})$, adjusts the dropping height from the peak to the decay_tail.

### 2.6.2 AE / AA extraction algorithm

At first, a FC MFCC vector is computed from audio waveform of a long audio file. To locate the onset and end of decay of one audible action / event, $\text{jumpHeightThreshold}$ is set to adjust detection sensitivity for Eq. 2.12. Eq. 2.23 is applied to determine the ending of an AA or AE.

Another fast AE extraction method is locating onset and setting a fixed length of AE session as a parameter $\text{stepLength}$, which is used to confine the parts from onset to decaying tail within one session. Note that fewer AE segments are extracted with a larger $\text{stepLength}$.

### 2.7 HMM-based AE Recognition System

In this research, sound recognition is to single out specific AE features, such as the sound of running water and chopping vegetables. In the AE domain, each type of AE sounds varies in
terms of rhythm, beats, repetition and object texture, etc. The basic assumption is that individual
perception on sound differs less, especially under a certain known circumstance. NILS share many
features with speech, especially for isolated words, in terms of time duration, short-term variations
in acoustic intensity and frequency, etc. We assume that a combination of techniques (GMM and
HMM) screening each AE sound at different granularity levels can help identify NILS and IS.

2.7.1 System Design

There are two major stages within NILS / IS recognition: a training stage and a testing stage.
Training involves building an acoustic model for each AE category. A good acoustic model should
be derived from audio characteristics that will enable the system to distinguish between the different
audible events.

In the testing stage, we use acoustic models to recognize isolated audio segments using a class-
ification algorithm. For isolated audio recognition, each audio segment is represented by the
parameters of its GMM. To estimate the parameters of a GMM for a set of frequency feature vec-
tors extracted from training audio, we use an iterative Expectation-Maximization (EM) algorithm
to obtain a Maximum Likelihood (ML) estimate. Given some test audio samples, we again extract
the frequency feature vectors from each frame of the detected subsample. The frequency feature
vectors host six most dominant frequencies in each audio. The objective is to find the audio model
with the maximum a posteriori probability for the set of test feature vectors, which reduces to
maximizing a log-likelihood value.

Frame length is an important parameter. If it is too small, it becomes hard to pick out mean-
ingful features and result in more computation overhead afterwards. On the other hand, if it is
too large, temporal information is lost between two frames. Another important parameter is the
number of different states in each model. The goal is that each state or some sequential states
should represent a phoneme in the audio. Here we borrow the ”phoneme” concept from the lin-
guistics field, and that is one of a small set of sounds that are distinguishable by people. Visually,
the subsample in Fig. 2.6(a) has only one phoneme, and that in Fig. 2.6(e) has three phonemes.
The assumption is that phonemes in each state follow Gaussian distribution, while in practice the acoustic feature vectors associated with a state may be non-Gaussian. In this case an M-component Gaussian mixture model is an appropriate way, the output probability density function at state \( s_j \) is Eq. 2.24.

\[
p(x|s_j) = \sum_{m=1}^{M} c_{jm} N(x; \mu^{jm}, \sigma^{jm})
\]  

(2.24)
the two parameters are the mean \( \mu \) and the standard deviation \( \sigma \). Given enough components, this family of functions can model any distribution.

HMM models are trained with labeled training data, and the classification is performed by passing the features to each model and then selecting the best match.

### 2.8 Probability-Based AE Recognition Algorithm by AA Fragmentation

Instead of by examining a large area of surface texture, human are able to distinguish glass and wood by throwing a glimpse over a small area, even in a coin size, of surface texture. Similarly, it is also true in the audio domain. So, if the audio acoustics are relatively uniform, not varying too much, they can be classified by a short audio excerpt.

**Fragmentation:** By knowing the onsets and ending points of decay tails using the method introduced in Sec. 2.6.1, a session of \( AE \) is fragmented into several \( AA \)s. The fragmentation allows the computer to process the audible event with a finer granularity. Likewise, people always pay more attention to sound that is difficult to discern. In the SDL domain, mostly an atomic \( AA \) takes 40 \( \sim \) 400 ms, which is hard for the human auditory system to discern as well, whereas a computer auditory system is capable of this.

\[
length(AE_i) \geq \sum_{k=0}^{N-1} length(AA_i[k])
\]

(2.25)
where \( AE_i \) denotes the \( i \)-th \( AE \) session; \( AA_i[k] \) is one \( AA \) fragmented from the \( AE_i \); \( N \) is the number of \( AA \)s fragmented from the \( AE \). Except for \( AA \)s, each \( AE \) session contain non-contributing parts of baseline, because they are from environmental context.
**AA classification based on GMM-HMM algorithm**: Each atomic AA label is classified by FFT-GMM-HMM algorithm as in the audible event recognition module described in Sec. 2.7.

**AA weighting rule**: Long period AA clips with high amplitude have larger weight, because long durations let it have rich acoustic features, and high amplitudes let it have larger range of amplitude variation and more maximal values. Thus, AA in the yellow region of Fig. 2.5 has a smaller weight than that of the other two.

![Figure 2.5](image)

**Figure 2.5** Thrice AAs of a knife hitting on a chopping board in this "cutting" AE.

\[
\text{energy}(AA_i[k]) = \max(\text{fc}_mfcc(AA_i[k]))
\]  

(2.26)

The energy function of an AA session is the maximum value (peak) of its first MFCC vector. Here, we use maximum value to simplify the computation of energy.

\[
i(\text{peak}) = \arg \max_{f \in S}(\text{fc}_mfcc(AA_i[k][f])). \quad f = 0, 1, \ldots, \text{framesize} - 1.
\]  

(2.27)

\text{fc}_mfcc() has a length of \text{framesize}, which is the total number of frames of the AA session. \(S\) is the set of the atomic AA MFCC frames.

\[
\Delta \Delta(\text{fc}_mfcc(AA_i[k][i(\text{peak})])) = 0
\]  

(2.28)
The peak index is where the second derivative of $f_{c \text{ mfcc}}(\cdot)$ equals to zero.

$$\omega_k = \frac{\text{energy}(AA_{i,j}[k])}{\sum_{k=0}^{N-1} \text{energy}(AA_{i,j}[k])}$$  \hspace{1cm} (2.29)

where $AA_{i,j}[k]$ denotes the $k$-th AA predicted as type $j$ in the AE of session $i$, and the weight value $\omega$ according to the amplitude level is obtained.

**Probability-based AE prediction:** We compute the probability of potential AAs in one AE based on each AA’s accumulative portion in the AE.

$$p_{i,j}[k] = \frac{\text{length}(AA_{i,j}[k])}{\text{length}(AE_i)} , \text{ and } \sum_{k=0}^{N-1} p_{i,j}[k] = 1$$  \hspace{1cm} (2.30)

where $p_{i,j}[k]$ denotes the probability of the $k$-th AA predicted as type $j$ in the AE of session $i$;

$$P_{i,j} = \sum_{k=0}^{N-1} \omega_k \cdot p_{i,j}[k]$$  \hspace{1cm} (2.31)

where $N$ is the total number of AAs fragmented from the AE session; $P_{i,j}$ is the probability of $i$-th AE session predicted as SDL type of $j$.

$$\text{label}(AE_i) = \text{label}(\text{max}(P_{i,j})), j = 0, 1, \ldots, M - 1$$  \hspace{1cm} (2.32)

At last, it labels the AE session with the AA tag of the maximal probability.

### 2.9 Experiments

#### 2.9.1 Experiment I: FC_MFCC Extraction

The audio files, more than 5 minutes of length each, sampled at 44100 Hz, quantized with 32-bit float, and in 705 kbps bit rate, are recorded by the ADL recorder APP Feng et al. (2016). During the FC_MFCC extraction, the FFT size is 512, successive window length is 25 milliseconds, step between successive windows is 10 milliseconds, and sample rate is 16000 Hz.

By using FC_MFCC extraction algorithm, one of extraction outcomes is shown in Fig. 2.6, including both NILS and IS. Computers have better “hearing sensitivity” than that of humans in the sound detection stage. The higher sensitivity expresses in all of three basic dimension: time,
frequency and amplitude. For example, computers can detect a sound, the time duration of which is much shorter than the human perception resolution, or the frequency of which is beyond human hearing range, or even when the amplitude is lower than human hearing threshold and resolution. It is hard for people to hear and perceive the tiny transient sound as the subsample in Fig. 2.6(f) since it requires high detection resolution. That is to say, once indeed occurs an AA, our system is able to screen it out.

The results of FC_MFCC extraction include AE excerpts from a whole AS audio file, and AA fragments from an AE session. Classification is set after extraction according to common perceptual knowledge on SDLs shown in Table 2.2. Those SDL excerpts will be identified in the following steps.

2.9.2 Experiment II: GMM-HMM Based AE Recognition

Six most dominant frequencies of an AE sample are stored in a Gaussian mixture matrix, which contains six states. The clustering of the Gaussians is however unsupervised and will depend on the initial values used for the Baum-Welch algorithm, a well-known EM algorithm Bishop (2006). For
Table 2.2 Categories of extracted audible events

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aluminumFoil</td>
<td>using aluminum foil;</td>
</tr>
<tr>
<td>bell</td>
<td>a bell rings, belonging to $AE$;</td>
</tr>
<tr>
<td>cutting</td>
<td>a knife cutting carrots, sound with a relatively lower volume;</td>
</tr>
<tr>
<td>coverLarge</td>
<td>placing a large can bottle cover on a kitchen table;</td>
</tr>
<tr>
<td>coverSmall</td>
<td>placing a smaller can bottle cover on a kitchen table; This type of sound is</td>
</tr>
<tr>
<td></td>
<td>much different from the previous one. These sound files are short and quick</td>
</tr>
<tr>
<td></td>
<td>over, while those previous ones are long-lasting and gradually soundless in</td>
</tr>
<tr>
<td></td>
<td>the end;</td>
</tr>
<tr>
<td>drawer</td>
<td>opening a drawer;</td>
</tr>
<tr>
<td>forkBowl</td>
<td>placing a fork in a bowl;</td>
</tr>
<tr>
<td>placingKnife</td>
<td>placing a knife on a marble-top kitchen island;</td>
</tr>
<tr>
<td>placingPan</td>
<td>placing a pan on an oven;</td>
</tr>
<tr>
<td>spoonScratchPlate</td>
<td>a spoon scratching a plate;</td>
</tr>
<tr>
<td>stirEggs</td>
<td>a fork stirring eggs in a bowl. Occurrences of stirring (beating) give rise to</td>
</tr>
<tr>
<td></td>
<td>sudden change of amplitude in the signal, and occurrences period is much</td>
</tr>
<tr>
<td></td>
<td>unlike that of Type(coverLarge). We can imagine that both beat strength</td>
</tr>
<tr>
<td></td>
<td>and period changing in amplitude forms a prominent signature for this type</td>
</tr>
<tr>
<td></td>
<td>of $AA$.</td>
</tr>
</tbody>
</table>

this project, totally random guesses (that obey the statistical properties) for transition probability matrix and its probability inside were used as initial values. The diagonal covariance matrix for the training data was used for all states. The training examples for each supervised audible events are in the audio samples already classified, and Baum-Welch is run for up to 15 iterations with 75% confidence interval. the training goal of EM (forward-backward) algorithm is to efficiently estimate the parameters of a HMM model from an observation sequence.

The performance of the system was measured by five-fold cross-validation on the recorded data set.
Table 2.3 SDL recognition performance by GMM-HMM based system

<table>
<thead>
<tr>
<th>No.</th>
<th>Types #</th>
<th>Total#</th>
<th>Mcr%</th>
<th>Most errors</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>36</td>
<td>0.0</td>
<td></td>
<td>types: {bell, stirEggs}.</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>76</td>
<td>0.0</td>
<td></td>
<td>types: {cutting, coverLarge, placingPan, stirEggs}.</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>51</td>
<td>19.61</td>
<td>bell ⇒ spoonScratchPlate 4/12, stirEggs ⇒ spoonScratchPlate 5/24</td>
<td>types: {bell, coverLarge, coverSmall, placingPan, spoonScratchPlate, stirEggs}.</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>81</td>
<td>4.94</td>
<td>spoonScratchPlate ⇒ stirEggs 5/12, spoonScratchPlate ⇐ stirEggs 1/23</td>
<td>types: {bell, coverLarge, coverSmall, placingPan, spoonScratchPlate, stirEggs}.</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>100</td>
<td>25</td>
<td>cutting ⇒ spoonScratchPlate 18/19, cutting ⇐ stirEggs 1/23</td>
<td>types: {bell, coverLarge, coverSmall, cutting, placingKnife, spoonScratchPlate, stirEggs}.</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>100</td>
<td>22.22</td>
<td>spoonScratchPlate ⇒ cutting 8/10, cutting ⇒ bell 14/19</td>
<td>types: {bell, coverLarge, cutting, drawer, placingPan, spoonScratchPlate, stirEggs}. mcr(bell) = 0.0.</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>91</td>
<td>17.58</td>
<td>bell ⇒ stirEggs 3/12, bell ⇒ placing 2/12, bell ⇒ cutting 2/12, cutting ⇐ stirEggs 1/19, spoonScratchPlate ⇒ cutting 8/10</td>
<td>types: {bell, coverLarge, coverSmall, cutting, placingPan, spoonScratchPlate, stirEggs}. mcr(cutting) drops sharply, but mcr(bell) rises largely.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>130</td>
<td>27.69</td>
<td>aluminumFoil ⇒ drawer 17/54, aluminumFoil⇒spoonScratchPlate 9/54</td>
<td>types: {aluminumFoil, bell, coverLarge, coverSmall, drawer, forkBowl, spoonScratchPlate, stirEggs}.</td>
</tr>
</tbody>
</table>

[Note] 1: A ⇒ B x/y: x out of y SDL samples of type A are mis-classified as that of type B.

2.9.3 Results and Discussion of the AE GMM-HMM Recognition

1. Exp. 1 and 2 show GMM-HMM can gain highly satisfactory performance for a small number of types. Note that the amplitude of "cutting" is much lower than other SDLs, making detection and recognition of the type "cutting" difficult.

2. Mis-classifications in Exp. 3 happen among types of {bell, spoonScratchPlate, stirEggs}, because there always exists a metal object hitting another object during these AE types.

3. Exp. 4 imports two new types of SDL samples ({coverSmall, placingPan}). Compared to Exp. 3, mis-classification rate mcr(stirEggs ⇒ spoonScratchPlate) declines, where mutual mis-classifications occur.

4. Furthermore, Exp. 5 imports SDL of type "cutting", which almost totally loses the recognition accuracy.
5. Exp. 6 runs again with the same data in Exp. 5. The type "cutting" begins to be misclassified from "stirEggs" to "bell", while no mis-classification happens in type "bell".

6. In Exp. 7, additional audio samples of another type "bell" are added to train the model of "bell" as a whole, leading to a larger mcr(bell). Interestingly, mis-classifications of cutting ⇒ spoonScratchPlate and spoonScratchPlate ⇒ stirEggs disappear. However, mis-classifications of spoonScratchPlate ⇒ cutting appears.

7. Exp. 8 imports type "aluminumFoil", mis-classification rate of which becomes more significant. Samples in types of \{bell, spoonScratchPlate, stirEggs\} are always mutually misclassified. The SDL of type "aluminumFoil" has a relatively lower amplitude, resulting in high mis-classification rate.

2.9.4 Results and Discussion of the Probability-Based AA Fragmentation on GMM-HMM Recognition

1. Exp. 9 first groups 3 super categories of sample files into different sub-categories. From previous experiments, we know that the 3 super categories of \{aluminumFoil, bell, stirEggs\} are more likely to be mis-classified. CrossTypeMcr is a new metric to evaluate the recognition effect, which only counts the number of mis-classification between different super categories, in a super-category level. An observation is that a small crossTypeMcr is gained, therefore, fine categorization can help achieve a better recognition performance.

2. Exp. 10 uses the same dataset with Exp. 10, but with six states per Multi-Gaussian model instead of three. Performance gets slightly better with the increasing number of states in each Gaussian mixture model. No other SDL types are mis-classified as type "aluminumFoil" anymore. Note that in this experiment, mis-classifications only happen within the same type, so the overall crossTypeMcr reaches 2.2%.
Table 2.4 SDL recognition performance by GMM-HMM based recognition and probability-based prediction system

<table>
<thead>
<tr>
<th>No.</th>
<th>Types #</th>
<th>Total#</th>
<th>Mcr%</th>
<th>Most errors (after probability prediction)</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8 (3)¹</td>
<td>90</td>
<td>22.22</td>
<td>aluminumFoil₁ ⇒ aluminumFoil₂ 5/16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bell₁ ⇒ bell₂ 2/6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ stirEggs₂ 7/22</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types: {aluminumFoil₁, aluminumFoil₂, aluminumFoil₃, bell₁, bell₂, bell₃, stirEggs₁, stirEggs₂}.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Three states / Gaussian.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>8 (3)</td>
<td>90</td>
<td>22.22</td>
<td>aluminumFoil₁ ⇒ aluminumFoil₂ 5/16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bell₁ ⇒ bell₂ 2/6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ stirEggs₂ 7/22</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types: {aluminumFoil₁, aluminumFoil₂, aluminumFoil₃, bell₁, bell₂, bell₃, stirEggs₁, stirEggs₂}.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Six states / Gaussian.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8 (3)²</td>
<td>417</td>
<td>27.34</td>
<td>aluminumFoil₁ ⇒ bell₁ 7/73</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>aluminumFoil₃ ⇒ bell₁ 6/73</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ stirEggs₂ 22/187</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types: {aluminumFoil₁, aluminumFoil₂, aluminumFoil₃, bell₁, bell₂, bell₃, stirEggs₁, stirEggs₂}.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Six states / Gaussian. Probability-based prediction.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4 (3)</td>
<td>365</td>
<td>53.85</td>
<td>cutting ⇒ spoonScratchPlate 18/61</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ stirEggs₂ 119/248</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ stirEggs₂ 10/30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(cutting ⇒ spoonScratchPlate 3/24)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types: {cutting(61)³, spoonScratchPlate(26), stirEggs₁(248), stirEggs₂(30)}.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Six states / Gaussian. Probability-based prediction.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>4 (3)</td>
<td>54</td>
<td>14.81</td>
<td>cutting ⇒ spoonScratchPlate 3/24</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>cutting ⇒ stirEggs₁ 1/24</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>spoonScratchPlate ⇒ stirEggs₂ 1/5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ spoonScratchPlate 1/22</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ stirEggs₂ 1/22</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types: {cutting, spoonScratchPlate, stirEggs₁, stirEggs₂}.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Six states / Gaussian.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pre-emphasis process.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4 (3)</td>
<td>447</td>
<td>36.02</td>
<td>spoonScratchPlate ⇒ stirEggs₂ 6/26</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ stirEggs₂ 100/248</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₂ ⇒ stirEggs₁ 1/30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(cutting ⇒ spoonScratchPlate 1/25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>cutting ⇒ stirEggs₁ 1/25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types: {cutting(143), spoonScratchPlate(26), stirEggs₁(248), stirEggs₂(30)}.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Six states / Gaussian. Probability-based prediction.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pre-emphasis process.</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>14 (9)</td>
<td>155</td>
<td>36.77</td>
<td>aluminumFoil₁ ⇒ bell₁ 16/28</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>aluminumFoil₂ ⇒ stirEggs₁ 5/28</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>cutting ⇒ spoonScratchPlate 2/5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types: {aluminumFoil₁, aluminumFoil₂, aluminumFoil₃, bell₁, bell₂, bell₃, coverLarge, coverSmall, cutting, drawer, forkBowl, spoonScratchPlate, stirEggs₁, stirEggs₂}.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pre-emphasis process.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>14 (9)</td>
<td>862</td>
<td>41.07</td>
<td>drawer ⇒ aluminumFoil₂ 15/30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SpoonScratchPlate ⇒ stirEggs₂ 6/26</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stirEggs₁ ⇒ forkBowl 1/22</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>types: {aluminumFoil₁, aluminumFoil₂, aluminumFoil₃, bell₁, bell₂, bell₃, coverLarge, coverSmall, cutting, drawer, forkBowl, spoonScratchPlate, stirEggs₁, stirEggs₂}.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pre-emphasis process. Probability-based prediction.</td>
<td></td>
</tr>
</tbody>
</table>

[Note] A ⇒ B x/y: x out of y SDL samples of type A are mis-classified as that of type B; 1: Categorizing 8 types out of 3 super categories; 2: Fragmenting 417 clips out of 90 audio samples; 3: Fragmenting 61 clips out.
3. Exp. 11 fragments $AE$ audio samples into $AA$ clips that are recognized by probability-based prediction described in Sec. 2.8. Here the final probability-based mcr reaches satisfactory zero percent.

4. Exp. 12 analyzes those types with high possibility of mis-classification, such as "cutting", "spoonScratchPlate", "stirEggs". After probability-based prediction, the overall mcr reaches 5.56% compared to 18.4% of crossTypeMcr.

5. Because acoustic features are present mostly in the high-frequency region, rather than the low-frequency region where most energy resides in. By emphasizing the high-frequency part, the better representative feature becomes more outstanding. The samples in Exp. 13 are first pre-emphasized the high-frequency part through a filter, then recognized by GMM-HMM system afterwards. The filter is as Eq. 2.33,

$$y[n] = x[n] - \alpha \times x[n - 1]$$  \hspace{1cm} (2.33)

where $x[n]$ is the signal and the value of $\alpha$ is usually 0.95. And the filter’s Z-transform process as Eq. 2.34,

$$H[z] = 1 - \alpha \times z^{-1}$$  \hspace{1cm} (2.34)

Compared to the result in Exp. 12, both mcr and crossTypeMcr decline largely.

Before pre-emphasizing, amplitudes of "cutting" audio samples are very low, even hard for HAS to detect and recognize; after pre-emphasizing, it allows signal of high-frequency to render more acoustic feature, thus the recognition accuracy among types of \{forkBowl, Spoon-ScratchPlate, stirEggs\} gain a better performance.

6. Samples of type "cutting" in Exp. 14 are fragmented into finer granularity, and recognize in probability-based system as well, reducing mcr to 3.7% than that in Exp. 12.

7. In Exp. 15, audio samples are pre-emphasized, and fed into GMM-HMM recognition system. As the number of SDL types increases, the accuracy of type "cutting" declines. Here, mcr(cutting) reduces from 94.7% in Exp. 5 to 48%.
8. Exp. 16 fragments 862 clips out of 155 audio samples from 9 type super categories. After pre-emphasis and probability-based GMM-HMM recognition, only 2/155 samples are misclassified. Compared to Exp. 15, the recognition accuracy improves significantly. Hence, the fragmentation with finer granularity improve the recognition accuracy.

Noteworthily, the AA fragments should not be too short, because it loses much acoustic feature belonging to any type, and therefore, in that case, it causes much inaccuracy in recognition.

2.10 Further Experiment and Discussion

This study first applies the proposed hierarchical situation audition approach using featured household sounds, such as chopping, turning on the faucet and placing a pan on a counter, where our algorithm gains a high accuracy. The recordings of sound clips we used are from two datasets. One is recorded from multiple personal mobile phones with the ADL Recorder App Feng et al. (2017), the other is obtained from Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database of Life Technology Center (2010), which recorded subjects performing the tasks involved in cooking and food preparation. In order to illustrate the classification performance in a more general manner, here we present a full-scale experiment based on two recent standard open datasets, ESC Piczak (2015) and TUT Mesaros et al. (2016), commonly accepted by this research community.

2.10.1 Comparison with a Baseline Classification System

In this experiment, a publicly available dataset, ESC-10 Piczak (2015), was selected for evaluation of the AA-based recognition algorithm.

ESC-10 is a less complex standardized subset of 10 classes (dog bark, rain, sea waves, baby cry, clock tick, person sneeze, helicopter, chainsaw, rooster, fire crackling), which has 40 clips per class.

Feature vectors were fed as input to three types of classifiers: k-Nearest Neighbors (k-NN), random forest ensemble and Support Vector Machine (SVM) with a linear kernel. As shown in Table 2.5, the results on second to fourth columns are derived from testing baseline machine classification provided by Piczak (2015) on an Ubuntu 64-bit platform. The fifth column shows the results from
Table 2.5  Recognition accuracy comparison of HMM-based AE recognition system and classification systems provided in Piczak (2015)

<table>
<thead>
<tr>
<th>Esc-10</th>
<th>k-NN</th>
<th>Random forest</th>
<th>SVM</th>
<th>w/o AA</th>
<th>AA-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Babycry</td>
<td>85.0%</td>
<td>85.0%</td>
<td>80.0%</td>
<td>60%</td>
<td>80.0%</td>
</tr>
<tr>
<td>Chainsaw</td>
<td>40.0%</td>
<td>52.5%</td>
<td>40.0%</td>
<td>45%</td>
<td>50%</td>
</tr>
<tr>
<td>Clocktick</td>
<td>27.5%</td>
<td>47.5%</td>
<td>50.0%</td>
<td>45%</td>
<td>62.5%</td>
</tr>
<tr>
<td>Dogbark</td>
<td>67.5%</td>
<td>77.5%</td>
<td>82.5%</td>
<td>5%</td>
<td>70%</td>
</tr>
<tr>
<td>Firecrackling</td>
<td>67.5%</td>
<td>87.5%</td>
<td>70.0%</td>
<td>25%</td>
<td>85%</td>
</tr>
<tr>
<td>Helicopter</td>
<td>62.5%</td>
<td>70.0%</td>
<td>62.5%</td>
<td>65%</td>
<td>67.5%</td>
</tr>
<tr>
<td>Personsneeze</td>
<td>92.5%</td>
<td>82.5%</td>
<td>75.0%</td>
<td>82.5%</td>
<td>87.5%</td>
</tr>
<tr>
<td>Rain</td>
<td>60.0%</td>
<td>65.0%</td>
<td>62.5%</td>
<td>62.5%</td>
<td>67.5%</td>
</tr>
<tr>
<td>Rooster</td>
<td>75.0%</td>
<td>82.5%</td>
<td>80.0%</td>
<td>62.5%</td>
<td>85%</td>
</tr>
<tr>
<td>Seawaves</td>
<td>90.0%</td>
<td>77.5%</td>
<td>72.5%</td>
<td>60%</td>
<td>72.5%</td>
</tr>
<tr>
<td>Average</td>
<td>66.7%</td>
<td>72.7%</td>
<td>67.5%</td>
<td>51.3%</td>
<td>70.75%</td>
</tr>
</tbody>
</table>

Running GMM-HMM recognition algorithm without AA-based fragmentation, and the results on the sixth column are derived from running our AA-based recognition algorithm from MATLAB on an Ubuntu 64-bit platform. An iterative Expectation-Maximization (EM) algorithm to obtain a Maximum Likelihood (ML) estimate is applied to train the GMM model.

The ESC-10 dataset had an average classification accuracy ranging from 66.7% for the k-NN classifier to 72.7% for the random forest ensemble, with SVM in the middle (67.5%) Piczak (2015). We secured a better accuracy of 70.75% by using our AA-based recognition algorithm.

In terms of acoustic features, the baseline system utilizes the first 12 MFCC coefficients and zero-crossing rates were summarized for each clip with their mean and standard deviation across frames. Therefore, from acoustic feature extraction step to classification step, the system needs to deal with 15 features. However, the AA-based recognition system only extracts 5 dominant frequencies from standard FFT as the acoustic features, which largely reduces memory consumption.

Besides, AA-based algorithm fragments audio clips in the preprocessing step, the and the total size of the audio clips is largely reduced from 168 MBytes to 85.2 MBytes. Furthermore, the total sound length before AA fragmentation is 2,003 seconds, and the total sound length after AA
Table 2.6 AA-based fragmentation contributes to higher recognition accuracy in the baseline machine classification system provided in Piczak (2015)

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Type</th>
<th>Fold1</th>
<th>Fold2</th>
<th>Fold3</th>
<th>Fold4</th>
<th>Fold5</th>
<th>Overall accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>kNN</td>
<td>Baseline</td>
<td>60.0%</td>
<td>66.2%</td>
<td>61.2%</td>
<td>52.5%</td>
<td>57.5%</td>
<td>59.5%</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>52.8%</td>
<td>50.0%</td>
<td>59.1%</td>
<td>70.0%</td>
<td>70.0%</td>
<td>60.1%</td>
</tr>
<tr>
<td>RF</td>
<td>Baseline</td>
<td>60.0%</td>
<td>66.2%</td>
<td>67.5%</td>
<td>68.7%</td>
<td>60.0%</td>
<td>64.5%</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>58.3%</td>
<td>60.9%</td>
<td>55.5%</td>
<td>70.0%</td>
<td>80.0%</td>
<td>64.9%</td>
</tr>
<tr>
<td>SVM</td>
<td>Baseline</td>
<td>65.0%</td>
<td>61.2%</td>
<td>65.0%</td>
<td>56.2%</td>
<td>55.0%</td>
<td>60.5%</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td>58.3%</td>
<td>60.0%</td>
<td>53.6%</td>
<td>80.0%</td>
<td>100.0%</td>
<td>70.4%</td>
</tr>
</tbody>
</table>

[Note] RF: random forest.

fragmentation is decreased to 1,011 seconds. Similarly, it saves nearly half of the memory in the feature extraction step.

2.10.2 AA-Based Fragmentation Contributes to Interior Sounds Classification

This experiment continues to use the baseline classification system, while the dataset contains only 10 classes interior domestic sounds selected from ESC-50 Piczak (2015). The types include clock alarm, can opening, door knocking, clock ticking, glass breaking, door wood creaking, mouse clicking, keyboard typing, washing machine running, and vacuum cleaner operating, 40 clips per class.

The aim of this experiment is to investigate how AA-based fragmentation technique contributes to the accuracy in the classification system when selecting the same acoustic features. Learning is performed on the datasets with a 5-fold cross-validation regime.

In Table 2.6, the results on rows 1, 3, and 5 are the accuracy retrieved from the baseline classification system by using k-Nearest Neighbors, random forest ensemble and Support Vector Machine classifiers. The results on rows 2, 4, and 6 are the accuracy when applying AA-based fragmentation before learning and classifying. We can see that AA-based fragmentation slightly increases the accuracy for both k-NN and random forest classifiers, and it accounts for an improvement of 9.9% for the SVM classifier.
2.10.3 AA-based Fragmentation Improves on Another Baseline System

In this experiment, it investigates how well AA-based fragmentation technique contributes to the acoustic scene classification accuracy for the baseline system provided in Mesaros et al. (2016). The audio dataset contains 15 types, such as beach, bus, cafe/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, park, residential area, train, and tram. Each type has 78 audio segments. Thus, it has 1,170 audio clips totally.

The baseline system utilizes a classical MFCC and GMM based classifier. The first 20 coefficients of MFCCs are calculated for all audio, including the first coefficient. Hamming window with 50% overlap is covered on 40 ms frames, and 40 mel bands totally. Delta and acceleration coefficients were also calculated using a window length of 9 frames, resulting in a frame-based feature vector of dimension 60. A GMM class model with 32 components is trained using expectation maximization algorithm for each acoustic scene Mesaros et al. (2016). The testing stage uses maximum likelihood decision among all acoustic scene class models. The classification results using the cross-validation setup are presented on the first row of Table 2.7, overall performance is 72.5%, being measured as the number of correctly classified segments among the total number of test segments.

Here, we have two-round experiments. In the first round, it generates 55,634 AA fragments out of the 1,170 audio clips. In the second round, it generates 55,353 AA fragments in total. The classification accuracies in the AA-level and AE-level are shown from second to fifth rows in Table 2.7. By comparison, the AA-based technique improves the baseline accuracy from 72.5% to 77.0% for the first round and 77.6% for the second round. In addition, the accuracies achieved from folds have less deviation than those from the baseline system without AA-based fragmentation, that is to say, AA-based technique gains a high consistency of performance because it extracts the most valuable fragments out of the whole audio files.

We must admit that the high recognition rate derived in our experiments is only in kitchen environments, which is in a super-category level. Our current approach may encounter a certain difficulty if we aim for functioning in a sub-category level when the super-category environment is unknown. For example, the recognition for the fine-granularity SDL will be very challenging when
the environment could either be in bedroom, kitchen, or bathroom. However, such a limitation can be eased if the super-category environment is predicted at first by PBN background estimation or other technologies, such as Wi-Fi localization.

2.11 Conclusions

We have presented an audible event (AE) / audible action (AA) onset detection scheme based on the first coefficient of MFCC feature. The progress of typical AE / AA waveform is exploited in depth, which makes it possible to distinguish sounds with variable lengths, such as persistent background noise (PBN), non-impulsive long sound (NILS) and impulsive sound (IS). We relate the signals in sound waveform to the elements in situation-aware environments. As explained, our aim is to identify the SDLs by recognizing the $AE$ in a certain audible situation (AS).

Some researchers work on identifying sounds in smart homes, but those sounds belong to single isolated NILS, most of which last 1~5 seconds, such as phone ringing, tapping water, which are considered as $AE$. While in this study, not only the single $AE$, but also $AE$ with multiple micro-AAs are analyzed and recognized. For the long AS session signal, we utilize the low-frequency part to detect and extract $AE / AE$; for each $AE / AE$, we emphasize high-frequency to recognize and predict. We presented a GMM-HMM-based $AE$ recognition system with probability-based AA classification in Sec. 2.8, which is suitable for predicting $AE$ by recognizing short IS fragments. The experiment results show that the probability-based AA classification largely improves the recognition accuracy for $AE$. The recognition rate of fine audio classification resolution (ACR)

<table>
<thead>
<tr>
<th>Type</th>
<th>Fold1</th>
<th>Fold2</th>
<th>Fold3</th>
<th>Fold4</th>
<th>Overall accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based line</td>
<td>66.5%</td>
<td>68.9%</td>
<td>72.3%</td>
<td>82.2%</td>
<td>72.5%</td>
</tr>
<tr>
<td>round 1 AA</td>
<td>77.0%</td>
<td>77.3%</td>
<td>76.4%</td>
<td>76.9%</td>
<td>76.9%</td>
</tr>
<tr>
<td>round 1 AE</td>
<td>76.7%</td>
<td>77.0%</td>
<td>77.0%</td>
<td>77.2%</td>
<td>77.0%</td>
</tr>
<tr>
<td>round 2 AA</td>
<td>75.8%</td>
<td>76.3%</td>
<td>76.3%</td>
<td>76.3%</td>
<td>76.2%</td>
</tr>
<tr>
<td>round 2 AE</td>
<td>78.5%</td>
<td>78.0%</td>
<td>76.8%</td>
<td>77.1%</td>
<td>77.6%</td>
</tr>
</tbody>
</table>
improves as promising as above 90% in certain cases. To our best knowledge, this work gains the best performance for the finest ACR problems to date.

In the future, we will perform deeper analysis on acoustic characteristics of some SDL types, by exploring a new and stable algorithm $\mathcal{R}$ to recognize $PBN$ and infer $E$. In addition, new technologies will be applied to compress and store the acoustic templates and models.

The significance of the presented work lies with the premise that humans living in the contemporary IoT social environment engaging various daily activities and oftentimes involve multiple agents. Automatic detection and identification of human activities can support researchers and developers to envision and develop novel services to improve the quality of experience and promptness for those who receive and consume such services. It is hopeful that we may help stimulate more interests for those like-minded researchers to produce more fruitful results in this critical emerging topic in IoT computing.
CHAPTER 3. LILO: ADL LOCALIZATION WITH SMART PHONE AND VISIBLE LIGHT

3.1 Introduction

Indoor localization is a research area that is gaining increasing attention recently. A wide range of location-based services (LBS) are attractive, such as navigating users inside shopping malls, pushing precise advertisements to the users, personalized recommendation, and proximity notification, etc.

Activities of daily living (ADLs) are routine activities that people tend to do every day, including six basic types: eating, bathing, dressing, toileting, transferring (walking) and continence. A person’s ability to perform ADLs is an important index for the nursing-home care or in-home care evaluation and service. Since the ADLs are highly correlated with the indoor locations, this work focuses on indoor localization for the ADL recognition. To provide LBS in the ADL area, high location accuracy is an essence.

Accurate indoor positioning is difficult to achieve by global positioning system (GPS) because devices usually are not able to connect to highly attenuated GPS signals at all. Therefore, the radio frequency (RF)-based technique is one of the possible alternatives. Many RF-based studies have attempted to provide precise location information during the past 20 years, including wireless local area network, radio-frequency identification, cellular, ultrasound, Bluetooth, and so forth Hightower and Borriello (2001); Kavehrad and Weiss (2015). There are some other localization systems relied signals such as magnetism Chung et al. (2011), lower-frequency FM broadcast radio signals for robust indoor fingerprinting Chen et al. (2012). These methods deliver positioning accuracies from tens of centimeters to several meters. However, this amount of accuracy is not sufficient for the applications described above, as most ADL recognitions expect a furniture-level differentiation within a small room. Apart from the relatively poor accuracy of indoor positioning
achievable by RF-based techniques, they are also subject to the electro-magnetic (EM) interference. Thus, the inaccurate estimation does not satisfy the room-level positioning for ADL recognition. GPS assisted by cellular and Wi-Fi methods Zandbergen (2009) can provide accurate and robust location information in most outdoor environments. Conversely, indoor positioning systems (IPS) only have low accuracy. Wi-Fi Received Signal Strength Indicator (RSSI) varies significantly over time and are susceptible to human presence, multipath, and fading, resulting in erratic locations. Moreover, one challenge in Wi-Fi and other RF-based IPS is high-cost pervasive infrastructures, needing setup related infrastructure Youssef and Agrawala (2005); Bahl and Padmanabhan (2000). Energy is an important consideration for mobile devices; nevertheless both obtaining a GPS signal and scanning for Wi-Fi would consume a significant amount of energy.

Our project LiLo, employing only a single-point smartphone, is trying to keep user devices as simple as possible with less battery consumption and higher accuracy.

For reasons mentioned above, the visible light communications (VLC)-based technology is gaining more attraction Komine and Nakagawa (2004). Furthermore, visible light-based approaches have shown some promise for indoor visible-light positioning (VLP) Xu et al. (2015); Yoshino et al. (2008); Sertthin et al. (2009); Liu et al. (2008).

The remainder of this chapter is structured as follows: Section 3.2 summarizes the related work on analysis methods of indoor ambient illumination. Section 3.3 presents the overview of the proposed system, followed by the introduction of the optical channel model and the Radiosity rendering method in Section 3.4. Details of trilateration-based algorithm is given in Section 3.5. In Section 3.6, we explain the design of the experiment and report the results of a series of experiments. We conclude the chapter in Section 3.7.

### 3.2 Related Work

There are two major hardware technologies supporting the algorithms to calculate the receiver coordinates. One technology form is a photo diode (PD) employed to detect received signal strength (RSS) information. As the distance varies according to the power attenuation, the receiver is
coordinated by lateration algorithms Yang et al. (2013). Another technology form is by image sensor detecting angle of arrival (AOA) information for the angulation algorithm to calculate the receiver location Tanaka and Haruyama (2009); Moon and Choi (2014).

Instead of relying on channel measurements, imaging techniques can be used to measure geometric relations between luminaries for localization Yoshino et al. (2008). However, imaging techniques alone often derive unsatisfactory inaccurate orientation, and consumes battery power fast.

A few recent simulation works are exploring in the visible-light positioning (VLP) area. In Rahman et al. (2011); Yoshino et al. (2008), image sensors are used to locate based on the lighting ray projection model. In Panta and Armstrong (2012), frequency division multiplexing was used for the peak-to-peak value of signals from different interferences stations, and time-difference-of-arrival (TDOA) was inferred based on phase difference of arrival. Proof of concept experiments are using two white light-emitting diodes (LEDs) as the transmitters. IDyLL Xu et al. (2015), an indoor localization system using existing inertial and light sensors on smartphones, is under the existing light infrastructure as well. IDyLL does not use absolute intensity readings. Instead, peak detection is employed to help infer the trajectory when the user is under a luminary. IDyLL builds upon existing work on pedestrian dead reckoning (PDR) (e.g. Gustafsson (2010); Davidson et al. (2010)), and the device in IDyLL requires to be facing up. Displacement is estimated through many factors, including step counts, stride length estimation and velocity estimation, heading orientation, which impose much possibility of inaccuracy. In addition, the illumination peak detection algorithm is largely confined to light arrangements in building hallways. The inertial measurement unit (IMU) Li et al. (2012) relies on inertial sensors to track a user by continuously estimating displacement from a known location. In Luxapose Kuo et al. (2014), it requires a high density of overhead LED luminaries to be placed with known positions and identification beacons. A camera-equipped smartphone decodes the LED identifiers and determines the phone’s absolute location and orientation in the local coordinate system with an angle-of-arrival localization algorithm.

Based on VLC, localization with a single LED or multiple LED by a trilateration method is discussed in Li et al. (2014). Epsilon Li et al. (2014) employs a light sensor on a smartphone
to retrieve the LED beacon information, and measures the received signal strengths (RSSs) from
multiple bulbs and computes the distances to each bulb through an optical channel model. After
decoding the beacon identifications, location is estimated. Epsilon deals with the light sources at the
similar height, while in reality, they may be deployed at any height. Certain optical channels where
beacons are transmitted need to be free from interferences from ambient light such as sunlight and
fluorescent light. In practice, such an ideal environment is rare. In real usage, the receiver (hence
the light sensor) may be in arbitrary orientation, which is considered a complicated problem by Li
et al. (2014). However, LiLo addresses this common situation and utilizes it as a helpful feature for
ADL recognition. LiLo leverages the equipped orientation sensors (e.g., inertial measurement unit
IMU on the phone) to measure the device’s attitude. With the light value, a tuple (value, attitude)
is recorded into the system. Then this tuple will be recognized as a location and ADL type. In
addition, an accurate optical channel model applicable to localization, working with multiple light
sources, also is proposed in Li et al. (2014).

Trilateration localization method in Zhang and Kavehrad (2012) computes the distances be-
tween a receiver and multiple light sources by varying the transmitting power. Similarly, the LEDs
will transmit ID code modulated. Zhang and Kavehrad (2012) assumes that a receiver is located
on the floor, LED bulbs are on the ceiling, as well as both the receiver axis and the transmitter
axes are perpendicular to the ceiling. Rajagopal et al. (2014) offers landmarks with approximate
room-level semantic localization depending on modulated LED. A VLC system using fluorescent
lamps and photodiodes sensor has been proposed in Liu et al. (2008), which can also estimate the
2D location. A single photo diode (PD) was used to estimate the vertical and horizontal angles
between the PD and the fluorescent lamps.

Furthermore, in Lee and Kavehrad (2012), proximity positioning concept is used to take a grid
of transmitters as reference points with known coordinates, however by nature, the accuracy is not
better than the resolution of the source grid.

Previous works require knowledge of the receiver orientation to solve for a position Jung et al.
(2013); Zhang and Kavehrad (2012). The use of a receiver with a 6-axis IMU to provide this
orientation information has been applied in Sertthin et al. (2009). The authors Sertthin et al. (2009) proposed a switching estimated receiver position scheme where the Estimated Receiver Positions (ERP) is switched depending on the receiver’s tilt angle. Obviously, the tilt angle with the 6-axis sensor limits the estimated error distance. Therefore, the accuracy of angulations data from 6-axis sensor is critical for the accuracy of the estimated positions.

Technically, phone’s attitude information is one of the crucial features for ADL recognition applications Feng et al. (2016). Nowadays, most models of smartphones already feature ambient light sensors (photodiodes) and 6-axis sensor (geomagnetic sensor and gravity acceleration sensor). The ambient light (illuminance) sensor is visible on the face of the device. Thus, LiLo takes full advantage of Android smartphone to retrieve the light level, with attitude, to derive the indoor localization. Note that LiLo does not require additional infrastructure support or device modification beyond standard smartphone. Light level is derived from the PD and attitude is derived from 6-axis sensor, all from the smartphone.

Most of the VLC-based techniques use LEDs as the light source, since they can be modulated more easily and hence, luminary ID with location data can be transmitted. In contrast, LiLo supports the various conventional light sources, including incandescent, fluorescent and LED luminaries, etc., to extract useful location information. From our knowledge, it is the first system that exploits conventional indoor luminaries for fine-grained indoor localization to recognize ADLs and demonstrate their usefulness experimentally.

Performance comparison of LiLo with the results reported in prior work is shown in Table 3.2.

3.3 System Overview

Key active areas (KAAs) are where the resident usually resides indoor, such as a PC desk in a reading room, an island table in a kitchen, a sofa area in a living room, a closet in a bathroom, etc. Each KAA is associated with one ADL usually. The locations of KAAs are highly determined by the floor plan and furniture layout, imaging that people can only wash nearby a faucet. In this work, The KAAs are known beforehand according to the subject’s routing history.
Table 3.1 Comparison of recognition accuracy to previous work

<table>
<thead>
<tr>
<th>Reference</th>
<th>Modalities</th>
<th>Face-up</th>
<th>Method</th>
<th>Average Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epsilon Li et al. (2014)</td>
<td>Modulated LED, Beacon-based, RSS, custom light sensor, smartphone, user’s gestures</td>
<td>No</td>
<td>MB, DC</td>
<td>0.4m</td>
</tr>
<tr>
<td>IDyLL Xu et al. (2015)</td>
<td>Accelerometer, Gyro, Compass, IMU, PDR, others conventional luminaries PD on smartphones office buildings, WD</td>
<td>Yes</td>
<td>Illumination peak detection</td>
<td>0.38 ∼ 0.5m</td>
</tr>
<tr>
<td>Luxapose Kuo et al. (2014)</td>
<td>LED ID with DC, camera, smartphone</td>
<td>Yes</td>
<td>AOA</td>
<td>0.1m</td>
</tr>
<tr>
<td>Lightitude Hu et al. (2015)</td>
<td>RSS, IMU, PDR, smart phone</td>
<td>No</td>
<td>MB, weights on RSS set</td>
<td>1.93m</td>
</tr>
<tr>
<td>LiLo</td>
<td>RSS, Accelerometer, Gyro, orientation, time in a day</td>
<td>No</td>
<td>FP, AoA</td>
<td>0.4m</td>
</tr>
<tr>
<td>Sertthin et al. (2009)</td>
<td>LED, PD</td>
<td>2D</td>
<td></td>
<td>1~2m</td>
</tr>
<tr>
<td>Liu et al. (2008)</td>
<td>Fluorescent lamp, PD</td>
<td>2D</td>
<td></td>
<td>0.1~0.3m</td>
</tr>
<tr>
<td>Yoshino et al. (2008)</td>
<td>Image sensor</td>
<td>3D</td>
<td></td>
<td>1.5m</td>
</tr>
</tbody>
</table>

[Note] DC: device configuration, Received Signal Strength (RSS), FP: Fingerprinting-based, MB: modeling-based, inertial measurement units (IMU), pedestrian dead reckoning (PDR), PD: photodiode sensors, ML: maximum likelihood, WD: war-driving, model based: based on ray projection model, AOA: angle-of-arrival

Normally, the subject either stays at a certain KAA for a relatively long time, or moves to another KAA during a relatively short time. For example, to most residents living at home, writing on a desk or cooking usually takes longer than walking to move from the desk to the kitchen. The objective of LiLo is to localize the subject’s current KAA and recognize the corresponding ADL.

Mostly, a PD is able to give one reading representing the incoming light value, as the phone is still. The reading changes either when the lighting environment changes or the phone moves. In our work, step detector and accelerometer in a smartphone are employed to distinguish if the phone is still, moving at current KAA, or transiting to another position. Moreover, when the phone is tending to moving, the relative peak of light with the phone’s attitude is helpful for further localization.
3.3.1 Challenges

In practice, there are some challenges need to be addressed, both to previous works and LiLo:

**From the smartphone’s perspective:** It is less possible to collect all the attitudes directly facing all the luminaires when the subject uses his phone naturally, so triangulation (using angle of arrival information) is not always feasible.

Zhang et al. (2014) uses linear least square estimation by knowing the distances from several reference points (transmitters’ horizontal coordinates), while in reality, these distances are hard to obtain if no photometric information is given. Furthermore, these distances computed by theoretical optical channel model do not meet the typical room-level accuracy.

**From the luminaires’ and occasions’ perspectives:** First, for some interior lighting design, LED light array is widely installed on the ceiling. High density of array and a large number of luminaires make it too hard to sense the orientation of the one who delivers the peak light. Second, the luminaire on the ceiling is too high to be considered as a spot light, while their light are scattering to the room as complete ambient light. Third, the attitude from the embedded 6-axis sensor is not always accurate, which yields a large deviation when facing a magnetic inference. Fourth, the optical transmission channel and the luminaires’ photometry are not as ideal as that in theory.

**From the user’s perspective:** In experimental research, in order to obtain the light information for each KAA, pre-collection is reasonable to conduct. While in real life usage, pre-collection is not always allowed by the users. Besides, the usages for the luminaire varies among different inhabitants in terms of his routine habit and time periods in a day. Thus, how to localize in the initial step is a big headache problem.

LiLo provides a solution to tackle the challenges described above by generating a illuminance field map. The system diagram is described in Fig.3.1.

1). **Initialization of illuminance field map:** According to information of house structure, furniture layout, and luminaire information, the indoor light field map is generated by the Radiosity algorithm described in Sec. 3.4.1. The luminance field map of a bedroom is shown in Fig. 3.2,
computed by the Radiosity algorithm. All the luminaires in this scene are ON. The height of the plate computed is 70 centimeters, which is as high as the KAAs on a desk. From this map, the computed luminance at KAAs t10, t11, t12 are 5-10 units of measurement, 30-75 units and 10-30 units, respectively. The computed luminance values at KAAs are recorded and sorted with respect to different facing azimuth angles. For example, four luminance lists of different azimuth angles are generated. Each list has the KAA IDs of the sorted luminance. The facing azimuth angle is largely determined by the furniture layout. As shown in layout Fig. 3.6, the south is the most likely orientation at p9, and the east is at p1. The most frequent heading orientation of the phone at each KAA can be computed by statistics. And the phone and the subject always head toward a similar direction. If the assumption does not satisfy all the time, the localization in the initial stage can be computed by the algorithm in Sec. 3.5.

2). Ambient light, geomagnetic field and orientation sensor data collection: The client (smartphone) generates and bundles a series of ambient Light, Orientation, Time and Step information (LOTS), then it sends the information packages to a cloudlet server, which holds the indoor illuminance field map. These coming LOTS packages are two-fold. On one hand, it serves as the
candidate to derive a KAA candidate label after comparing in a KAA luminance list. The candidate label is validated by the AOA algorithm. This label record is fed into the historical database to update the labels in the same azimuth group. This system trains a light-level model for every KAA in this floor, as shown in Fig. 3.7. Another serves as a new coming record to be classified by the trained model.

3). **Calibration to generate an advanced illuminance field map:** The LOTS packages at each KAA are archived into a database no matter which direction the smartphone is facing. Now that the illuminance field map only takes effect when the smartphone approximately faces up, so the selected recording in the coming package to compare in a luminance list only chooses the facing-up ones. With the real LOTS data of the phone, the illuminance field map, generated at the cold start, is upgraded to an advanced modality with six dimensions, including (x,y) position in the map, (azimuth, pitch, roll) gyroscope information on the (x,y) position, and (t) time. And
the multi-dimension illuminance KAA lists are stored as the templates in the location estimation stage.

4). Location estimation: In practice, the client sends the LOTS packages to the reference server. During the initial inference phase, the location is determined using the illuminance field map with orientation knowledge. After the cold-start phase, the server compares the measured light value and geomagnetic field signals with the records in the advanced illuminance field map database. Eventually, the location results are computed from a machine learning classifier.

3.4 The Radiosity Rendering Model

In this section, we will introduce the optical channel model and a method of rendering, Radiosity, based on an detailed analysis of light reflections off diffuse surfaces. For better illustration, first we introduces several types of light defined by computer graphics researchers:

- Ambient light’s color scatters to all the objects in the scene globally.

- Directional light shines from a specific direction, as if it is infinitely far away, so the rays are considered as all parallel. The sun is a typical light source.

- Hemisphere light source positions directly above the scene.

- Ceiling light is more like a hemisphere light source.

- Point light is at a specific position in the scene, and light shines in all directions.

- Spot light is a point light that can cast a shadow in one direction within a falloff cone.

The original Radiosity system was developed by Goral et al. (1984). This module calculates the light exchange between luminaires and any other surfaces (direct lighting) and the light exchange between illuminated surfaces (indirect lighting). Not only direct lighting emitted by a certain luminary can be calculated, but also lighting from the sky (daylight) or direct sunlight can also be calculated with the calculation kernel. Based on the energy conservation principle, the premise
is that any light which is projected onto a surface and is not absorbed will be reemitted by this surface.

3.4.1 Discrete Radiosity Overview

Surfaces are assumed to be perfectly Lambertian (diffuse), which reflects incident light in all directions with equal intensity. With the radiosity method an equation is made for each surface. The scene is divided into a set of small areas, or patches. The radiosity, \( B_i \), of patch \( i \) is the total rate of energy leaving a surface, and the radiosity over a patch is constant.

This equation defines the light emitted which is a product of light absorbed from other surfaces and, if present, from its own luminance. Altogether this provides a set of equations whose solution represents the brightness of each individual surface. Thus, the reflected light which is perceived is a combination of multiple light sources Goral et al. (1984). We separate the scene into \( n \) patches, over which the radiosity is constant

\[
B_i = E_i + \rho_i \sum_{j=1}^{N} F_{ij} B_j, \quad i = 1 \ldots n
\]

where \( B_i \) is the light leaving patch \( i \), \( E_i \) is the light emitted from patch \( i \), \( \rho \) is the material reflectivity, the reflectivity of surface \( i \), which will absorb a certain percentage of light energy which strikes the surface. \( F_{ij} \) is the form factor, fraction of light energy leaving patch \( j \) that arrives at patch \( i \), which is determined by both geometry (size, orientation, and position of the two patches) and visibility, such as any existing occlusions in between.

\( n \) simultaneous equations with \( n \) unknown \( B_i \) values can be written in matrix form:

\[
\begin{bmatrix}
1 - \rho_1 F_{11} & -\rho_1 F_{12} & \ldots & -\rho_1 F_{1n} \\
-\rho_2 F_{21} & 1 - \rho_2 F_{22} & \ldots & -\rho_2 F_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
-\rho_n F_{n1} & \rho_n F_{n2} & \ldots & 1 - \rho_n F_{nn}
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_n
\end{bmatrix}
= 
\begin{bmatrix}
E_1 \\
E_2 \\
\vdots \\
E_n
\end{bmatrix}
\]

(3.2)

The "full matrix" radiosity solution calculates the form factors between each pair of surfaces in the environment, then forms a series of simultaneous linear equations.
A single radiosity value $B_i$ is for each patch in the environment so that a view-independent solution is computed. The radiosity of a single patch $i$ is updated for each iteration by gathering radiosities from all other patches:

$$
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_i \\
\vdots \\
B_n \\
\end{bmatrix}_{t+1} = 
\begin{bmatrix}
E_1 \\
E_2 \\
\vdots \\
E_i \\
\vdots \\
E_n \\
\end{bmatrix} + 
\begin{bmatrix}
\rho_1 F_{11} & \rho_1 F_{12} & \cdots & \rho_1 F_{1i} & \cdots & \rho_1 F_{1n} \\
\rho_2 F_{21} & \rho_2 F_{22} & \cdots & \rho_2 F_{2i} & \cdots & \rho_2 F_{2n} \\
\vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
\rho_i F_{i1} & \rho_i F_{i2} & \cdots & \rho_i F_{ii} & \cdots & \rho_i F_{in} \\
\vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
\rho_n F_{n1} & \rho_n F_{n2} & \cdots & \rho_n F_{nn} & \cdots & \rho_n F_{nn} \\
\end{bmatrix} 
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_i \\
\vdots \\
B_n \\
\end{bmatrix}_t
$$

where $t$ is the $t$-th iteration. This method is fundamentally a Gauss-Seidel relaxation.

The geometric terms in the form-factor derivation are illustrated in Fig. 3.3. For non-occluded environments, the form-factor between finite surfaces (patches) is defined as the area average and is thus:

$$
F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_{ij} dA_j dA_i
$$

the "form factor" between surfaces $i$ and $j$, which accounts for the physical relationship between the two surfaces. Where $A_i$ is the area of surface $i$, $r$ is the vector from patch $i$ to $j$, $r^2$ is the square
of distance $r$. $\theta_i$ is the angle between normal $i$ and vector $r$. $V_{ij}$ is a boolean visibility function between patch $i$ and $j$, taken either 0 if point on $i$ is occluded with respect to point on $j$, or 1 if unoccluded. As the reciprocity law says:

$$A_i F_{ij} = A_j F_{ji}$$  \hspace{1cm} (3.5)

The "radiosity equation" describes the amount of energy which can be emitted from a surface, as the sum of the energy inherent in the surface (a light source, for example) and the energy which strikes the surface, being emitted from some other surfaces.

### 3.5 Localization Based on Luminance Field Map with Trilateration

The Android operating system is set up to calculate a rotation matrix $R$ which is defined by

$$R = \begin{bmatrix} E_x & E_y & E_z \\ N_x & N_y & N_z \\ G_x & G_y & G_z \end{bmatrix}$$

where $x$, $y$ and $z$ are axes relative to the smartphone, see Fig. 3.4, and where

- $E = (E_x, E_y, E_z) = a$ unit vector which points East
- $N = (N_x, N_y, N_z) = a$ unit vector which points North
- $G = (G_x, G_y, G_z) = a$ unit vector which points away from the centre of the earth (gravity vector).  \hspace{1cm} (3.6)
The Euler angles $\phi$, $\theta$ and $\psi$ in Android operating systems are defined as:

- **azimuth** $-\phi$ : rotation about $G$, $z$ – axis
- **pitch** $-\theta$ : rotation about $E$, $x$ – axis
- **roll** $-\psi$ : rotation about $N$, $y$ – axis

Thus, the $3 \times 3$ rotation matrix $R$ is expressed in terms of Euler angles,

$$
R = \begin{bmatrix}
\cos \phi \cos \psi & \sin \phi \sin \psi \sin \theta & \sin \phi \cos \theta \\
-\sin \phi \cos \psi & \cos \phi \sin \psi + \sin \phi \cos \psi \sin \theta & \cos \phi \cos \theta \\
-\sin \psi \cos \theta & -\sin \theta & \cos \psi \cos \theta
\end{bmatrix}
$$

The unit vectors in the direction of the $x$, $y$, and $z$ axes of a three dimensional Cartesian coordinate system are

$$
\hat{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \hat{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \hat{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
$$

A direction vector can be transformed between rotated reference frame from the device coordinate system to the world’s coordinate system with the rotation matrix $R$, which helps to rotate vectors:

$$
\vec{V}_G = R\vec{V}_D
$$

where, $V_D$ is a vector $V$ measured in the frame of reference of the device; $V_G$ is a vector $V$ measured in the frame of reference of the global world.

The unit vector points away from the face of the smartphone is $\hat{k}$, and its transformation in the world frame of reference is

$$
\vec{k}_G = R\hat{k} = \begin{bmatrix}
\cos \phi \sin \psi + \sin \phi \cos \psi \sin \theta \\
-\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta \\
\cos \psi \cos \theta
\end{bmatrix}
$$
As shown in Fig. 3.5, a set of bulbs and a mobile terminal are in the room. In our project, the bulbs could be anywhere other than just situated on the ceiling. A light sensor inside the mobile terminal extracts the location information to perform positioning. The optical channels considered here are all line-of-sight (LOS) links.

Mostly, the distances from reference points (transmitters’ horizontal coordinates) are unknown. Therefore, this system relies on the angle of arrival rather than the computed distances.

Three luminaire and a smartphone are exhibited in Fig. 3.5. The known coordinates of the luminaire in the global world’s coordinate system are \((x_1, y_1, z_1)\), \((x_2, y_2, z_2)\), and \((x_3, y_3, z_3)\). We want to estimate the coordinate of the smartphone \((x_p, y_p, z_p)\). The vectors

\[
\begin{align*}
\overrightarrow{OL_1} - \overrightarrow{OP} & = \overrightarrow{PL_1} = \lambda_1 k_{G_{R_1}} \\
\overrightarrow{OL_2} - \overrightarrow{OP} & = \overrightarrow{PL_2} = \lambda_2 k_{G_{R_2}} \\
\overrightarrow{OL_3} - \overrightarrow{OP} & = \overrightarrow{PL_3} = \lambda_3 k_{G_{R_3}}
\end{align*}
\]

(3.12)
$|\overrightarrow{PL_i}|$ is the distance from reference point $i$.

\[
\begin{align*}
(x_1, y_1, z_1) - (x_p, y_p, z_p) &= \lambda_1(\Delta x_{r1}, \Delta y_{r1}, \Delta z_{r1}) \\
(x_2, y_2, z_2) - (x_p, y_p, z_p) &= \lambda_2(\Delta x_{r2}, \Delta y_{r2}, \Delta z_{r2}) \\
(x_3, y_3, z_3) - (x_p, y_p, z_p) &= \lambda_3(\Delta x_{r3}, \Delta y_{r3}, \Delta z_{r3})
\end{align*}
\] (3.13)

Eq. (3.13) is a restatement of Eq. (3.12), with the subtractions expanded in terms of the elements of the vectors.

\[
\begin{align*}
z_1 - \lambda_1\Delta z_{r1} &= z_p \\
z_2 - \lambda_2\Delta z_{r2} &= z_p \\
z_3 - \lambda_3\Delta z_{r3} &= z_p \\
z_1 - \lambda_1\Delta z_{r1} &= z_2 - \lambda_2\Delta z_{r2} = z_3 - \lambda_3\Delta z_{r3}
\end{align*}
\] (3.14)

where $\lambda$ is the scaling factor

\[
\begin{align*}
|(z_1 - \lambda_1\Delta z_{r1}) - (z_2 - \lambda_2\Delta z_{r2})| &\leq \text{lengthPhone}/2 \\
|(z_1 - \lambda_1\Delta z_{r1}) - (z_3 - \lambda_3\Delta z_{r3})| &\leq \text{lengthPhone}/2 \\
|(x_1 - \lambda_1\Delta x_{r1}) - (x_3 - \lambda_3\Delta x_{r3})| &\leq \text{lengthSpace}/2 \\
|(y_1 - \lambda_1\Delta y_{r1}) - (y_3 - \lambda_3\Delta y_{r3})| &\leq \text{lengthPhone}/2 \\
0 &\leq x_p \leq \text{lengthSpace} \\
0 &\leq y_p \leq \text{widthSpace} \\
0 &\leq z_p \leq \text{heightSpace}
\end{align*}
\] (3.15)

The solution of this Eqs. (3.14) gives points of circles intersection, providing a zone of indoor localization.

Alg. 1 shows the algorithm for localization computation of the smartphone. Alg. 2 shows the algorithm for location validation of the smartphone. At least two luminaires detected in the space are necessary to estimate the precise localization. Afterwards, the estimated position can be validated when more than two luminaires are detected.
Algorithm 1: Phone localization calculation

**Input:** Quantity $n$ and coordinates $(x_i, y_i, z_i)$ of $L_i$. **Known:** The length of smartphone $lengthPhone$.

**Output:** Coordinate of $P$ - $(x_p, y_p, z_p)$

1. Find a list of directions $\{\text{azimuth} \phi, \text{pitch} \theta, \text{roll} \psi\}_i$ angles when the smartphone receive the $n$ peak light levels;
2. Compute the $z$-axis increment vectors $\overrightarrow{k_{GRi}}$ from $\{\phi, \theta, \psi\}_i$ in the frame of reference of the world;
3. Select the x-coordinate $\{x_1, x_2\}$ and z-coordinate $\{z_1, z_2\}$ values of the $L_i$ with the two new least $z_i$ for prediction, and any of others for validation;
4. $\text{temp}_z \leftarrow heightSpace$;
5. $x1\_array \leftarrow \Phi$;
6. $x2\_array \leftarrow \Phi$;
7. $i \leftarrow 0$;
8. while $\text{temp}_z \geq 0$ do
   9.     $\text{temp}_z \leftarrow z_1 - i \Delta z_{r1}$;
   10.    $x1\_array(j).add(x1 - i \Delta x_{r1})$;
   11.    $i \leftarrow i + 1$;
   12.    $\text{temp}_z \leftarrow heightSpace$;
   13.    $j \leftarrow 0$;
   14.    $\text{minDifferent} \leftarrow lengthPhone$;
   15.    $\text{temp}_j \leftarrow 0$;
16. while $\text{temp}_z \geq 0$ do
   17.     $\text{temp}_z \leftarrow z_2 - j \Delta z_{r2}$;
   18.     $x2\_array(j).add(x2 - j \Delta x_{r2})$;
   19.     for each $x_i$ in $x1\_array$ do
      20.         if $|x2\_array(j) - x_i| \leq lengthPhone/2$ then
         21.             if $|x2\_array(j) - x_i| < \text{minDifferent}$ then
         22.                 $\text{minDifferent} \leftarrow |x2\_array(j) - x_i|$;
         23.                 $\text{temp}_j \leftarrow j$;
      24.         $j \leftarrow j + 1$;
   25.     $x_p \leftarrow x2\_array(\text{temp}_j)$;
   26.     $y_p \leftarrow y2 - \text{temp}_j \Delta y_{r2}$;
   27.     $z_p \leftarrow z2 - \text{temp}_j \Delta z_{r2}$;
   28.     if validation($x_p, y_p, z_p$) then
      29.         return coordinate($x_p, y_p, z_p$);
   else
      30.         goto step 3;
Algorithm 2: Phone localization validation

1. **validation** \((x_p, y_p, z_p)\)

**Input:** Coordinate \((x_p, y_p, z_p)\) of the estimated \(L_p\). **Known**-The coordinate \((x_v, y_v, z_v)\) of \(L_v\) and the direction \(\{\phi, \theta, \psi\}_v\) from the validation set, the length of smartphone \(\text{lengthPhone}\), the length, width, height of the space.

**Output:** Coordinate of \(P - (x_p, y_p, z_p)\) is valid or not.

2. Compute the \(x, y, z\)-axis increment vectors \(\overrightarrow{k_{GRv}}\) from \(\{\phi, \theta, \psi\}_v\) in the frame of reference of the world;

3. \(x\text{Diff} \leftarrow \text{lengthSpace} ;\)

4. \(z\text{Diff} \leftarrow \text{heightSpace} ;\)

5. **if** \(0 \leq x_p \leq \text{lengthSpace} \) and \(0 \leq y_p \leq \text{lengthSpace} \) and \(0 \leq z_p \leq \text{lengthSpace} \) **then**

6. \(x\text{Diff} \leftarrow |(x_v - x_p)/\Delta x_{rv} - (y_v - y_p)/\Delta y_{rv}| ;\)

7. \(z\text{Diff} \leftarrow |(x_v - x_p)/\Delta x_{rv} - (z_v - z_p)/\Delta z_{rv}| ;\)

8. **if** \(x\text{Diff} < \epsilon \) and \(z\text{Diff} < \epsilon \) **then**

9. **return** True;

10. **return** False;

When just one luminary is detected by the smartphone in the space, it is still possible to localize the phone. The changing light value will be detected when the smartphone moves. First, when it gets the peak light value, the attitude of the phone is retrieved by the 6-axis sensor. Along the direction towards to the luminaire we can draw a line from the device to the luminaire. Different luminaire will yield different light volume, which in turn will validate the location result. In the localization period, when new recordings with the approximately same direction are detected, comparing with the historical records of the light values and the attitude, this system relocates those clusters under the same direction.

### 3.6 Experiments

We evaluate the performance of LiLo in three sections. Section 1: Experiment 1 is designed to validate how well LiLo can locate users under different luminaire arrangements across different KAAs in an apartment.
Table 3.2 Photometry of luminaire

<table>
<thead>
<tr>
<th>ID.</th>
<th>Category</th>
<th>Location</th>
<th>Height (m)</th>
<th>Power (W)* Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>l1</td>
<td>ceilingLight</td>
<td>entrance hall</td>
<td>2.4</td>
<td>40 * 1</td>
</tr>
<tr>
<td>l2</td>
<td>fluorescent</td>
<td>above the faucet</td>
<td>1.65</td>
<td>13 * 2</td>
</tr>
<tr>
<td>l3</td>
<td>fluorescent</td>
<td>kitchen ceiling</td>
<td>2.4</td>
<td>54 * 2</td>
</tr>
<tr>
<td>l4</td>
<td>ceilingLight</td>
<td>living room</td>
<td>2.4</td>
<td>13 * 3</td>
</tr>
<tr>
<td>l5</td>
<td>undercloset</td>
<td>above the stoves</td>
<td>1.23</td>
<td>13 * 2</td>
</tr>
<tr>
<td>l6</td>
<td>bathLight</td>
<td>above the vanity table</td>
<td>2.1</td>
<td>50 * 2</td>
</tr>
<tr>
<td>l7</td>
<td>ceilingLight</td>
<td>hallway</td>
<td>2.4</td>
<td>40 * 1</td>
</tr>
<tr>
<td>l8</td>
<td>tableLamp</td>
<td>on the desk</td>
<td>1.0</td>
<td>40 * 1</td>
</tr>
<tr>
<td>l9</td>
<td>ceilingLight</td>
<td>bedroom</td>
<td>2.4</td>
<td>40 * 1</td>
</tr>
<tr>
<td>l0</td>
<td>floorLamp</td>
<td>on the desk</td>
<td>1.3</td>
<td>40 * 1</td>
</tr>
</tbody>
</table>

[Note] All the luminaire are spot light source, because the lamp shade channels the direction of the emitting light. l2 and l3 are fluorescent, others are incandescent.

Section 2: Experiment 2.1 and 2.2 mine the historical data in real-world scenarios, from the ADL Recorder App project Feng et al. (2016). This experiment aims to answer if the proposed LiLo system contributes to location accuracy and ADL recognition.

Section 3: Experiment 3 tests the performances of different machine learning classifiers for fine recognition in a defined space.

3.6.1 Experimental Living Environment

Here we give one apartment to illustrate the experimental process. This work took place in an actual apartment of size 800 ft\(^2\) as the living environment, the layout of which is shown in Fig. 3.6, containing a bedroom (position p1), a bathroom (position p3), a living room (position p8) with a combined kitchen (position p5). The locations of luminaire are marked with hollow squares in the floor plan map. The photometry of luminaire is shown in Table 3.4. In this luminance environment, the beam angle of luminary l8 is ±30 degree; and that of luminary l0 is ±180 degree, so luminary l0 is a point light; all other luminaire have beam angles of nearly ±90 degree.

The smartphone used is Google Nexus 5 with the Android 6.0.1 Marshmallow operating system. This model features sensors for geomagnetic information, orientation information, and light level.
3.6.2 Activities of Daily Living in the Venues

Here, we defined typical activities of daily living that frequently happen in our daily life. Mostly, the locations of ADLs are largely determined by the layout of furniture and appliances. For example, one can only do washing by a faucet. The circles are ADL capturing points, and the arrow around circle denotes the most frequent facing orientation in each.

* Working on PC - The major pieces of furniture in the bedroom are a long combined desk, and a bed. The facing orientation of ADL is much subject to the furniture layout. In this case, the subject often works on computer and reads some material at position p1 facing east.

* The subject usually falls asleep, takes a nap, or reads on his smartphone any time in the bed. The illuminance environment at position p2 varies in time. It has a relatively lower light level in the daylight, while it has a relatively higher light level in the night when the luminary 8 turns on.
* Hygiene activities (2 types) - Once the inhabitant is inside the bathroom, he could perform normal hygiene activities at position p4 facing either west or east or wash at position p3 facing east.

* Cooking - At position p5, the subject cooks, chops and prepares food facing south, which is toward the direction that the stoves locate.

* Washing dishes - At position p6, the subject washes dishes, vegetable, fruits, etc., with the heading orientation of west.

* Eating - The sofa at position p9 is most frequently used when the resident has meals for best convenience, including breakfast, lunch, dinner and mid-night snack. The smartphone is usually put on the sofa table or on the sofa, t3 and t2, respectively, as shown in Fig. 3.7.

* Dressing up - At the entrance hall (position p7), the subject usually selects clothes by the wardrobe, and puts on / removes shoes by the shoe closet.

3.6.3 Experiment 1: Localization Based on Orientation, Light Level and Time

Data were pre-collected from the Light Meter App running on Nexus 5 smart phone. The Light Meter, developed by our group, detects ambient light level values, 3 attributes from a geomagnetic field sensor, and 3 attributes from a orientation sensor. It encapsulates the data and sends this package to the cloudlet server. At each KAA, the smartphone is rotated arbitrarily to detect data from multi-direction, so a number of records with different phone’ attitudes are collected.

The light levels at multiple KAAs are tested, and different illuminations are displayed in various colors as shown in Fig. 3.7.

For each KAA, the illumination is determined by the different ON/OFF status combinations of surrounding luminaire. So the possible light situations at each KAA need to be recorded. For instance, at KAA p6, the most contributive light source are the lamp above the faucet (l2), the kitchen ceiling lamp (l3), and the lamp under the microwave oven (l5) as illustrated in Fig. 3.6. The data in p6 are from eight combinations of luminaire (l2, l3, and l5) status, where each luminary is either ON or OFF.
The total number of instances collected in this experiment is 24631. Six comparison sessions are analyzed and the model is trained with a Bayes Network classifier in the WEKA (2018). Sessions (a) and (b) are analyzed with data from both the geomagnetic field sensor and the orientation sensor. Sessions (c) and (d) are analyzed with data only from the orientation sensor. Sessions (e) and (f) are analyzed with data only from the geomagnetic sensor. Sessions (b), (d), and (f) are analyzed with data only when the smartphone faces up approximately, with both roll and pitch angles are less than 15 degrees.

Testing data are evaluated on training data. Two metrics are considered here. $Mcr$ represents mis-classification rate and $crossTypeMcr$ represents the $Mcr$ happening among different types. The types here mean different combinations of light status ($ON$ or $OFF$). For example, the mis-classification between different light situations at KAA is not regarded as an error, because the KAA result is the same. Thus, $crossTypeMcr$ draws more attention to the KAA-level recognition.
63

Figure 3.8  Recognition comparison using different attributes. "g" represents the geomagnetic information; "o" represents the orientation information.

And Mcr focuses on not only the KAA recognition, but also the status of the surrounding luminaire. The comparison result is shown in Fig. 3.8.

In general, recognition result of the facing-up smartphone is better than multi-direction, mainly because the ambient light sensor retrieves more precise illuminance information when the phone faces up. And the combination usage of both geomagnetic field sensor and orientation sensor outperforms than using the orientation sensor alone, which also works better than using geomagnetic field sensor alone. The best performance is obtained using data from both geomagnetic field sensor and orientation sensor when the smart phone is facing up.

3.6.4 Experiment 2.1: Recognizing ADLs Based on Orientation, Light Level and Time

3.6.4.1 Experimental Setup and Data Collection

The training set includes all the sensing data records (21 different scenes, 4388 samples), and test set uses the same training set.

The classification performance was evaluated using leave-one-out cross-validation, where a classifier is trained with all instances except the one that is left out for classification. In this way,
Table 3.3 List of the recorded ADL scenes, and number of recording for each.

<table>
<thead>
<tr>
<th>Main context</th>
<th>Scene</th>
<th>No. of recordings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bathroom (910)</td>
<td>Bathroombowl</td>
<td>405</td>
</tr>
<tr>
<td></td>
<td>Bathroomfaucet</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td>Bathroomflushing</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>Bathroompee</td>
<td>208</td>
</tr>
<tr>
<td>Kitchen (765)</td>
<td>Chop</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Cooking</td>
<td>506</td>
</tr>
<tr>
<td></td>
<td>Washingdishes</td>
<td>227</td>
</tr>
<tr>
<td>DiningRoom (1293)</td>
<td>Breakfast</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Dinner</td>
<td>491</td>
</tr>
<tr>
<td></td>
<td>LivingRoom</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Lunch</td>
<td>501</td>
</tr>
<tr>
<td></td>
<td>Midnightsnack</td>
<td>275</td>
</tr>
<tr>
<td>Bedroom (1230)</td>
<td>Dinnerbedroom</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Gettingup</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Nap</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>Sleep</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>WorkingonPCathome</td>
<td>1010</td>
</tr>
<tr>
<td>Public places (159)</td>
<td>Amespubliclibrary</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>PCoffice</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Gilman1353</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>RossHall</td>
<td>6</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>4388</strong></td>
</tr>
</tbody>
</table>

the training data is maximally utilized, even though the system has never experienced that particular recording before. The overall recognition rates were calculated as the sample mean of the recognition rates of the individual scenes.

Real-time recordings from a variety of different ADL scenes were generated from the ADL Recorder App Feng et al. (2016). The aim of the App project is to recognize ADLs only via a single-point Android-based smartphone, which captures the ADL types with multiple sensing data, including light level, azimuth angle, time, Wi-Fi RSSI values and so on. A Nexus 5 smartphone with the Android system (Marshmallow 6.0.1) is used as the experimental device. The assumption for this experiment is that the smartphone is used normally and placed by the subject at every KAA.
Attributes used in the ADL recognition stage include time, orientation and light level. The time attribute here plays a more critical role than timestamps, as natural factors also impact the lighting environment. For example, the light volume through windows are different from daylight to night. Furthermore, the luminaire usage is changing from time to time. One could turn on a desk lamp for reading at night, and afterwards turn on the floor lamp for wandering in the room.

In Table 3.3, the different scenes and number of recordings from each scene are listed. The recordings are categorized into five general classes according to locations of the scenes (bathroom, kitchen, dining room, bedroom, and some public places). All the first four scenes happen in the apartment as Fig. 3.6, and the public places includes four buildings far away located in the city of this experiment - Ames, Iowa. The apartment in the experiment is located in the northeastern area of the city. "Amespubliclibrary" represents the Ames public library located in the center of downtown. "Pcoffee" represents that the subject is working in the office where is in the Atanasoff Hall, which is in the center of the campus. "Gilman1353" represents the classroom 1353 in the Gilman Hall on the campus. The "RossHall" represents a classroom in the basement of the Ross Hall in the eastern area of the campus.

After preprocessing the data, and deriving the location information from the combination of Wi-Fi RSSI, an ARFF (Attribute-Relation File Format) file is produced and fed into WEKA. We trained and tested the dataset with Bayesian network classifiers.

### 3.6.4.2 Recognition Accuracy and Discussions

The confusion matrix for 21 classified scenes using the Bayesian network classifiers is presented in Table 3.4. The rectangular boxes enclose the more general contexts as presented in Table 3.3. The overall recognition rate was 89.93% for analysis 4388 instances on attributes of \{ light, orientation angle, hour, location \}. The boxes enclose more general classes with high-level location. In the meantime, the recognition rate of general contexts level is 99.84%.

- The type of ”Bathroomflushing” are the most common type for mis-classifications, because ”Bathroomflushing” and ”Bathroomfaucet” usually take place near position p3 at the same time. How-
ever, after the audio process stage, mis-classifications are largely modified, because the SDL of "Bathroomflushing" type is distinct from that of "running water from faucet", "Pee", and "Bow-
el".

- The types of "Bathroomfaucet" and "Bathroompee" are nearly 30% mis-classified in between. Basically because the two ADLs positions are close to each other, near position p3 in Fig.3.6, the light level and orientation are roughly similar, even the two ADLs occur simultaneously.

- The ADLs of "chopping", "cooking", and "washing dishes" usually interweave, and the actual spots are close to each other (in the neighborhood of positions p5 and p6), so that the illuminance atmosphere, generated from the luminaire \( \{l_2, l_3, l_5\} \), are more or less the same. Similarly, audio processing is helpful for the mis-classifications improvement.

- All "eating" ADLs happen in the living room, and a slight mis-classifications exist between "dinner" and "midnight snack", as the shared the position p9 and occurred subsequently.

- "Lunch" is sometimes mis-classified as "dinner", partially because of the mis-labels. The subject sometimes reported the second meal in a day as "lunch", no matter when the meal was, even after 5 pm.
- The "getting up", "nap" and "sleep" are mis-classified as "Working on PC at home" due to the same orientations and venues shared position (p1 and p2). Besides, the subject usually "works on PC at home" any time in a day.

- Interestingly, the attributes of time and light level are helpful for the recognitions of "dinner in bedroom", although of which the orientations and venues (position p1) are roughly the same with those of "working on PC at home". Specifically, "Dinner in bedroom" takes place at t12 in Fig. 3.7, and "working on PC at home" happens at t11. The distance between the two venues is just less than half a meter.

  The recognition accuracy of scenes in a location level ranged from 99.7% (bathroom) to 100% (public places). Basically, the illuminance atmosphere and time in each place are distinct. Note that the satisfactory performance is gained without usage of GPS information, while the localization for the "public places" is still of high accuracy.

  If the attributes fed into the Bayesian network classifier is \{light level, orientation, and hour\}, leaving out the location estimation based on Wi-Fi RSSI, the overall correct classification rate is 76.16%. And the classification rates of the five location-level general categories are 79.78%, 70.07%, 93.27%, 83.51%, 96.23%. The most major mis-classification is "kitchen"-related to "dining room"-related. And the majority of ADL-level mis-classification is "cooking" to "dinner" or "lunch". The possible reason is that the functional places are connecting with each other and the luminaire l3 and l4 illuminate the entire place.

### 3.6.5 Experiment 2.2: Comparison of Recognition Rates Using Different Attributes

The dataset of this experiment is the same as that in Exp. 3.6.4, which has 4388 samples from 21 different scenes. In the experiments (a) - (g), we trained and tested the dataset with Bayesian network classifiers; and in the experiment (h), we select J48 decision tree algorithm as the classifier. The pre-recognized location is derived from the Wi-Fi localization algorithm via SVM method Feng et al. (2016). The performance comparison is shown in Fig. 3.9.
Figure 3.9 Comparison of recognition rates using different combinations of attributes. In the xlabel, "angle" represents the orientation azimuth angle.

Suppose via any Wi-Fi based localization algorithm, we can get nearly high-accuracy locations, the results show that the "light" feature positively contributes to improve performance of localization. By adding the attribute of "light", the ADL-level recognition rate raise from 87.03% (b) to 89.93% (c).

However, retrieving the accurate locations by Wi-Fi localization is not always plausible. There are two major causes: first, Wi-Fi RSSI value usually shifts transiently, sometimes Wi-Fi connection is even lost; second, mostly, in a house Wi-Fi RSSI status of each room has less distinguishing features, because the near distances from each room and the wall blockage of the signal. From the comparison of experiment (d) and (e), by adding the attribute "light", the ADL-level recognition rate grows from 69.07% to 76.32% distinctly. And the location-level recognition rate obviously increases from 77.80% to 83.89% as well. Thus, "light" attribute substantially contributes to the performance.
The basic combination of \{light, azimuth angle, hour\} (a) is a solution to recognize both ADL types and locations without the Wi-Fi based localization algorithm process. The performance difference of this basic combination from that of importing attribute of inaccurate "pre-recognized locations" (e) is negligible. Yet without the Wi-Fi RSSI-based process, it saves the battery consumption of scanning Wi-Fi RSSI, the storage of the combination of RSSIs, and the computation overhead of the pre-recognition process.

From the comparison of experiments, including nearly correct pre-recognized locations (b and c) or inaccurate locations (d and e), it indicates that the attribute "light” contributes to performance improvement under both cases. With the attribute of 3-axis "acceleration” data, we can see the performance result of experiment (g), grows furthermore. Here the accuracy of 84.23% is consistent with that in the Exp. 3.6.3 Fig. 3.8 session (e).

In the experiment (h), the J48 decision tree algorithm is selected to classify the ADL types and locations, the performance grows to an acceptable 93.92% and 96.74%, respectively.

3.6.6 Experiment 3: Recognition in a Given Space by Different Classifiers

There are two sessions in this experiment, one for a living room and kitchen, the area of which is 7.6m \times 3.4m; the other one is for a bedroom, the area of which is 3.5m \times 3.5m; From the floor plan in Fig. 3.6, only by a Wi-Fi based localization algorithm, it is hard to differentiate the area between kitchen and living room areas, due to less salient signal features. Hence, the objective of this experiment is to validate the recognition rate of the precise areas in a given room.

From the floor plan in Fig. 3.7, in the living room, the ADLs usually happen in two KAAs. The "eating"-series ADLs, including ”breakfast”, ”lunch”, ”dinner” and ”mid-night snack” take place on the tea table in KAAs of t2 or t3 region; the ADLs of ”chopping”, ”cooking”, and ”washing dishes” usually happen in the kitchen neighboring KAA t5.

In the bedroom, the ADLs usually happen in three KAAs. The ADLs of ”getting up”, ”napping”, and ”sleeping” take place in the bed near KAA t10; The ADLs of ”working on PC at home”
usually happens on the desk near KAA t11; Sometimes, the subject has "dinner in the bedroom" in the KAA t12 region. These KAAs above are considered as the location-level measurement.

Figure 3.10 Recognition comparison in bedroom and living room under different kind of classifiers. "BayesNet" represents using a Bayesian network classifier and "J48" represents using a decision tree classifier.

We trained the data with attributes of {light level, azimuth angle, 3-axes accelerometer, and time} via WEKA, the Bayesian network and the J48 tree are selected as the classifiers. The comparison is shown in Fig. 3.10. The metric of "ADL-level" recognition is considered as correct if and only if the specific ADL, such as "getting up", "napping", etc., is recognized correctly. Likewise, the metric of "location-level" recognition is considered as correct if and only if the specific locations of ADLs (e.g. KAA t3, t5, t10, etc.) are recognized correctly.

From the comparison result, in such a small bedroom, the recognition rate of both the ADL-level and the location-level gains above 94%. Note that the distance between t10 and t11 is as small as less than 40 centimeters, and the mis-classification rate in between is less than 1.9% under J48 tree classifying. The distance between t11 and t12 is less than 60 centimeters, and the ADL-level mis-classification rate in between is less than 4% under J48 tree classifying.
The reason of higher recognition accuracy rate in a small room, compared with in multiple rooms, is partially due to fewer ADL categories. Besides, the fewer luminaries with the similar luminance characteristics is one factor resulting in classification errors. For example, the light coming through the window in the living room or through the window in the bedroom in the afternoon are of less difference. The properties of light (especially luminous intensity) of the ceiling light in the living room, l4 in Fig. 3.6, is similar to that of ceiling light l9 in the bedroom, because they have the similar heights, luminance intensities, and colour temperatures.

3.7 Conclusions

We first present and develop LiLo, a light-based localization system, to estimate smartphone users’ ADLs via machine learning algorithms. This service employs ambient light level, with orientation information obtained all by a single-point smartphone, to estimate the indoor positioning with high accuracy. This application does not impose much extra burden and battery consumption to the phone, because the ambient light sensor and orientation sensor are running in the background all the time. In addition, we tackle the technical challenge of localization in the cold start without certain reference records, and solve the problem of less likelihood to collect data in the off-line phase. The surrounding environment influences is a big factor causing location error, such as wall reflection. Nevertheless, LiLo take full advantage of those characteristics to generate a luminance field map. On the server end, in order to achieve a more realistic indoor luminance field and luminance on KAAs, we do consider light propagation between surfaces in the scene. Moreover, compared with the previous works, LiLo is able to deal with the spots in storage closets. We validate the performance of LiLo in both an experimental environment and a real-world environment. Our results show that the overlooked light from conventional off-the-shelf luminaire add much potential feature into the environment and LiLo achieves decimeter indoor location accuracy for regular smartphones. The efficient management of illuminance field map at different time is one of our future tasks.
CHAPTER 4. ONLINE WI-FI CLUSTERING ALGORITHM FOR GLOBAL LOCALIZATION

4.1 Problem Statement and Challenges

GPS features are not always working for some users, in order to save energy or under indoor environments, so we will develop an alternative approach to differentiate multiple major locations from analysis on the environmental context, such as Wi-Fi Received Signal Strength Indicator (RSSI) and Global System for Mobile Communications (GSM). With the new positions being visited, the number of total Wi-Fi access points increases as new ADL data arrives. That means the dataset grows both in quantity of records and number of features. Thus, a new clustering algorithm is needed to solve this kind of problems. Moreover, this clustering algorithm should be adaptive to distinguish different location-resolutions, such as city, campus, building, apartment, and room levels.

Wi-Fi access point (AP) scanning is a common feature on smart phones. Normally, multiple APs are visible in the surrounding, and more APs can be detected in some public area, such as libraries, airports and shopping malls. Therefore, the total number of APs in the vicinity increases and tends to a large number gradually. In one session of our real-world experiments, one’s phone kept scanning APs during its 2-month life, more than 5000 APs were scanned in total. The increasing large number of total APs is one of the new challenges for this work. The target of this work is for a global localization, which means, it can provide localization in various levels, such as nation, city, building levels. This large feature number results in a heavy burden for some conventional clustering algorithms, e.g. K-Means, to cluster all features in all examples.

In some previously visited area, where a phone is connected to an AP used to be connected, it is possible that not other AP information are provided at that moment. Thus, there can be basically four different situations that a phone yields its Wi-Fi environment.
1. without any AP information;

2. many APs without any connection;

3. only one connected AP without other APs information;

4. both connected AP and many surrounding APs. Moreover, even though for those situations with both connected AP and surrounding APs, the number of detected APs is not always same for each time. In practice, the number and locations of AP in a given place are totally unknown beforehand. Consequently, it causes diversity in AP features. Applying a dimensionality reduction like PCA with K components on those variable length of feature vectors is not feasible for such case, because each RSSI from any AP has separate contribution.

In a building with many bearing-walls, where light-of-line condition can not be indefinitely guaranteed, it is less likely to gain high accuracy of distance between the mobile device to any AP. Thus, trilateration Oguejiofor et al. (2013) OnkarPathak et al. (2014) algorithm is not suitable in the real world.

Another major challenge is that, from the nature of wireless radio communication, Wi-Fi signal from any AP is not going to be continuously reliable. It is easy to observe that Wi-Fi signal quality keeps fluctuating all the time. As a result, this undermines the localization precision. Then again, it is likely to predict as a different location sometimes at the same area.

Because multiple WLANs can coexist in one airspace, a Service Set Identifier (SSID), simply the 1-32 byte alphanumeric name, represents a unique name of the WLAN network. A basic service set identifier (BSSID), as a identifier of a access point, as multiple access points can exist within each WLAN. By convention, an access point’s MAC address is used as a BSSID. Received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal. The RSSI value is represented in a negative form (e.g. -91 dB), the closer the value is to 0, the stronger the received signal has been. In order to perform Wi-Fi based localization, the ADL recorder App collects APs information in the surrounding, including the connected AP, and many other APs. Each AP’s information consists of RSSI and BSSID, indicating the signal strength coming from a
given AP, whose MAC address is the BSSID. Thus, an AP feature vector can be represented as the following format: \([B_c : R_c, B_1 : R_1, B_2 : R_2, \ldots, B_i : R_i]\), where \(B_c\) is the BSSID of the connected AP, and \(R_c\) is its RSSI. \(R_i\) is the RSSI received from the AP whose BSSID is \(B_i\).

4.2 Online Clustering Algorithm

We develop an online clustering algorithm fitting for the global Wi-Fi localization which both the feature number and the record number are increasing. Then, we use the distance function and training data to map each instance into a feature vector with a variable length.

4.2.1 Distance Function

Based on the characteristics of RSSI, some rules are observed.

- Rule I: If two AP feature vectors share the same connected AP, the two instances have high likelihood to be in the same / close area.

- Rule II: If two AP feature vectors have their own connected APs (or say, one does not connect to any AP), but they share some AP BSSIDs, the two instances still have high likelihood to be in the close area. The RSSIs associated with the shared BSSIDs are factors of the distance metric.

- Rule III: If two AP feature vectors have their own connected APs (or say, one does not connect to any AP), but they do not share any AP BSSID, the two instances should not be in the close area.

- Rule IV: The more BSSIDs the two AP feature vectors share, the closer the two locations resides. In the meantime, as a negative value, the higher is the RSSI number, the stronger is the signal. Thereof, RSSI of each AP should also be considered as a quantization of signal strength.

- Rule V: While the connected AP does not mean that the its signal is stronger than other APs, it can only address that the client has the AP’s permission.
4.2.2 Similarity Measures in Large Scale

Here we define a distance function computing the similarity between two instances of variable number of AP features. The Jaccard coefficient Jaccard (1912) measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

\[ J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (4.1) \]

where A, B are two AP BSSID vectors extracted from AP features vectors. Jaccard coefficient is employed to measure the similarity based on Rule II and III.

4.2.3 RSSI Computation for Directional Fine Scale

Jaccard coefficient of AP BSSID vectors can only determine if the two locations are in the same large zone rather than the fine zone. Here RSSI from each AP can help to separate a large zone into several small directional zones. As an example in Fig. 4.1, all of these eight APs are scanned by the four mobile phones in the same zone, so the Jaccard coefficients between each two AP feature vectors always reach \( J(A, B) = 1 \). To simplify the situation, only line-of-sight propagation is taken account.

An RF propagation model (e.g., the Log-Distance Path Loss (LDPL) model) can be used to predict RSS at various locations in the indoor environment Chintalapudi et al. (2010).

\[ p_{ij} = P_i - 10\gamma_i \log d_{ij} + R \quad (4.2) \]

Eqn. 4.2 indicates that the j-th mobile user receives a signal strength of \( p_{ij} \) (measured in dBm) when it locates at a distance \( d_{ij} \) (measured in meters) from the i-th AP. \( P_i \) denotes the RSS from the i-th AP at a distance of one meter (referred to as transmit power henceforth). The path loss exponent, \( \gamma_i \), captures the rate of fall of RSS in the vicinity of the i-th AP. \( R \) is a random variable, capturing the variations in the RSS due to multi-path effects, asymmetries in the physical environment (e.g., obstructions) and other imperfections in the model itself.
In turn, the distances $d_{ij}$ between mobile devices and APs can only be inferred from RSS values with \textit{a priori} knowledge of $P_i$ and $\gamma_i$.

$$d_{ij} = 10^\frac{P_i - R_{ij}}{10\gamma_i} \tag{4.3}$$

Figure 4.1  Localizing phones among multiple access points.

In an ideal environment, the signal strength is negatively correlated with the distance. In the case as in Fig. 4.1, the RSSI value of each AP received by a mobile phone should be totally different. The RSSI values of the farther APs are less than those of the near APs. So from mobile 1 ($m_1$)’s perspective, the AP vector with RSSI values sorted from greatest to least is probably $[ap_1, ap_5, ap_4, ap_2, ap_6, ap_3, ap_7, ap_8]$; and the AP vector from mobile 3 ($m_3$) is probably $[ap_3, ap_2, ap_7, ap_6, ap_4, ap_8, ap_5, ap_1]$. Each mobile in this zone has its own AP vector of sorted RSSI values, which hence makes it possible to separate this large zone into several small zones directionally.

\subsection*{4.2.4 RSSI Computation for Distance within A Directional Fine Scale}

This section will compute the summation of distances from a mobile phone to all visible APs in order to distinguish the finer zones, corresponding to the Rule IV. Suppose $ap_1$ and $ap_5$ do not exist in Fig. 4.1, as what we aforementioned, the AP vectors what $ap_1$ and $ap_2$ collect are in the
same form \([ap_1, ap_2, ap_6, ap_3, ap_7, ap_8]\), because the relative positions to those APs are similar. In order to differentiate the position where \(m_1\) and \(m_2\) reside, summation of distances are employed as follows:

\[
S_i = \sum_j d_{ij}^2
\]  

(4.4)

where \(i\) is the id of mobile phone, and \(j\) is the id of AP. Eqn. 4.4 computes the distance summation of mobile phone \(m_i\) to all visible APs \([ap_1, ap_2, \cdots, ap_j, \cdots]\). Thus, as the example above, \(S_1\) should be larger than \(S_2\), which means \(m_1\) is further from those visible APs than \(m_2\).

Figure 4.2  Distance difference with RSSI computation
Fig. 4.2 illustrates the principle of the RSSI computation to differentiate fine zones. In this scenario, two mobile phones exist in a Wi-Fi environment with three APs. If drawing a straight line connecting the two mobile phones, we can use this reference line as the basis for computation. 

$h_j$ is the perpendicular line from $AP_j$ to the reference line; $d_{ij}$ is the distance from the mobile phone $m_i$ to $AP_j$; perpendicular line $h_j$, hypotenuse $d_{ij}$ and part of the reference line form a triangle. We can find the hypotenuse represents the distance from a given mobile phone to an AP.

According to the Pythagorean Theorem, in a right triangle, the sum of the squares of the lengths of the triangle’s legs is the same as the square of the length of the triangle’s hypotenuse. So

$$
\begin{align*}
S_1 &= d_{11}^2 + d_{12}^2 + d_{13}^2 \\
S_2 &= d_{21}^2 + d_{22}^2 + d_{23}^2
\end{align*}
$$

(4.5)

$$
S_1 - S_2 = (d_{11}^2 - d_{21}^2) + (d_{12}^2 - d_{22}^2) + (d_{13}^2 - d_{23}^2)
$$

(4.6)

$$
S_1 - S_2 = (b_{11} - b_{21}) + (b_{12} + b_{22})(b_{12} - b_{22}) + (b_{13} + b_{23})(b_{13} - b_{23})
$$

(4.7)

where $N$ is the total number of APs. and $d_x$ is the distance between the two mobile phones.
Furthermore, Eqn. 4.8 achieves a general mode, which suggests that the two mobile phones’ RSSI difference is determined by physical distance in between and 2-fold average total distances from the reference mobile phone to those APs. Since APs scanned are always around the mobile phones, the vector sum of reference mobile phone to those APs tends to be zero. So the two mobile phones’ RSSI difference is directly proportional to the square of distances between the two mobile phones. Note that if \( d_x \) is far larger than \( \frac{2}{N} \sum_{j=1}^{N} |b_{2j}| \), the two mobile phones can not scan the same APs set.

### 4.3 Algorithm Implementation

**Nearest neighbor(s) in signal space (NNSS)**

Bahl and Padmanabhan (2000) is a metric to compare multiple locations and pick the one that best matches the observed signal strength. The idea is to compute the distance (in signal space) between the observed set of SS measurements \((ss_1, ss_2, ss_3)\) and the recorded SS \((ss'_1, ss'_2, ss'_3)\) at a fixed set of locations, and then pick the location that minimizes the distance. The Euclidean distance measure is used, i.e.,

\[
\sqrt{(ss_1 - ss'_1)^2 + (ss_2 - ss'_2)^2 + (ss_3 - ss'_3)^2}
\]

However, the signal strength is logarithmically proportional to the distances. The APs with stronger RSSI should be more contributive than those with weaker RSSI, because the difference of stronger RSSI has more representatives than the weaker ones, as the signal index value is not linear. In order to represents this characteristics, a weight coefficient \( \mu \) is introduced.

\[
p_{ik} - p_{jk} = 10\gamma_k \log d_{jk} - 10\gamma_k \log d_{ik} = 10\gamma_k \log \frac{d_{jk}}{d_{ik}}
\]

\[
\frac{d_{ik}}{d_{jk}} = 10^{-\frac{1}{\gamma_k} (p_{ik} - p_{jk})}
\]

\[
\mu(P_{ik}, P_{jk}) = 10^{-\frac{1}{\gamma_k} (p_{ik} - p_{jk})} \ast |P_{ik} + P_{jk}|
\]
By using Manhattan distance metrics, the sum of the absolute differences for each base station weighted by the signal strength level is computed.

\[
\text{strength\_distance}(s_{ik}, s_{jk}) = \mu(s_{ik}, s_{jk}) \cdot |s_{ik} - s_{jk}|
\]

(4.12)

Algorithm 1   Similarity Calculation: determining if two mobile phones are in the same zone

**Input:** \(v_i = [B_i c: R_i c, B_i 1: R_i 1, B_i 2: R_i 2, B_i 3: R_i 3, \ldots], v_j = [B_j c: R_j c, B_j 1: R_j 1, B_j 2: R_j 2, B_j 3: R_j 3, \ldots]\), and similarity threshold \(\theta_r\).

**Output:** True or False

11 if \(B_i c \neq \Phi \land B_j c \neq \Phi \land B_{ic} \neq B_{jc}\) then
12 return False
13 compute Jaccard coefficient \(J(v_i, v_j)\)
    if \(J(v_i, v_j) \leq 0.5\) then
14 return False
15 else
16 \(v'_i = \text{sorting } v_i \cap v_j \text{ in } v_i\)
17 \(v'_j = \text{sorting } v_i \cap v_j \text{ in } v_j\)
18 \(N = \text{size}(v_i \cap v_j)\)
    \(\text{Rssi\_accu\_diff} = \sum_{k=0}^{N-1} \text{strength\_distance}(s_{ik}, s_{jk})\)
    if \(\text{Rssi\_accu\_diff}/N \geq \text{Threshold}\) then
19 return False
20 else
21 return True

Figure 1 shows the matching algorithm to determine if two mobile phones are in the same zone based on the AP RSSI feature vectors. In the matching algorithm, some pre-conditions are checked in the beginning to reduce the execution time. Lines 1 \sim 2, based on the Rule III; Lines 3 \sim 5, based on the Rule IV.

In the clustering step, we match every coming AP RSSI feature vector with every vector in RSSI seed database, categorizing this coming vector into a group once it matches to any seed vectors, so
the coming vector is labeled as the matched zone ID. Otherwise, the new coming vector is added into the seed database, and labeled as a new zone ID.
CHAPTER 5. TIME-SERIES BASED SENSOR FUSION

Mobile devices are becoming increasingly sophisticated and the latest generation of smart cell phones now incorporates many diverse and powerful sensors. The sensors used in this work include GPS sensors, audio sensors (i.e., microphones), proximity light sensors, atmospheric pressure sensors (i.e., barometer), direction sensors (i.e., magnetic compasses), and acceleration sensors (i.e., accelerometers), Wi-Fi module, cellular signal module, etc.

Time-series based sensor fusion technique is adopted to analyze fast growing data, mainly serves as a feedback mechanism of error correction.

5.1 Time-Series Sensor Fusion Model

Figure 5.1  Time-series sensor fusion cascade model with error correction
Harris et al. (1998) Elmenreich (2007) proposed a waterfall model of hierarchical architecture for data fusion community. We enhance this model with time-series features.

This model describes the flow of data operates from the data level to the decision-making level. The sensor system is continuously updated with feedback information arriving from the decision-making module. The feedback element advises the multi-sensor system on re-calibration, re-configuration and data gathering aspects. We believe that time-series feedback can be applied not only for the outer loop from the decision-making module to the sensor level, but also within each module in every level, as shown in Fig. 5.1:

- At the physical level, multiple sensors keep capturing the behavioral context and environmental context. Physical sensors usually are in "On" state, the transmission links of data into storage and bottleneck data control are considered.

- When obtaining data from the physical level, the data level checks for physical transmission errors and packages sensing data into "frames". Time-series filter are responsible for reducing or removing noise, and smoothing data in the pre-processes. As mentioned in Sec. 5.1.1, smoothing heading direction for case:96 in Fig. 5.2 is one example applied here. The data level also formats the data into packets delivered up to the information level.

- In the information level, data are properly extracted into formatted features. The extraction algorithm for each module varies. Wi-Fi based localization module is built on utilising the Wi-Fi RSSI feature; acoustic feature of environmental sound is extracted and recognized from audio files; activity recognition focuses on the data collected from the accelerometer, gyroscope sensors, etc. Based on the features along the time going, time-series pattern recognition and discovery are conducted.

Time-series correction feedback can handle error correction in the information level. As an example, due to the unstable Wi-Fi signal, one-round Wi-Fi localization algorithm can not gain accurate locations. Adding the correction feedback can improve the accuracy. In this case, some basic rules are readily applied. It is impossible for a user to relocate without any
motion. Likewise, it is impossible that a phone receives totally same Wi-Fi RSSI from every AP after it moves directly forwards. Such kinds of impossible rules are useful for corrections.

- In the decision level, through the context assessment, where context awareness is imported, which plays a big role in combining the environmental context and behavioral context. This level is capable of adapting multiple types of features together for high-level decision making furthermore. Time-series correction feedback in this level is responsible for rule-based error correction based on context as well. In practical recognition, it is less likely that one's ADL pattern is "cooking, sleeping, then cooking" within 30 minutes. Therefore, the ADL label of "sleeping" in the middle should be purged.

- The output of presentation level might not be as much as that from the decision making. Presentation is for end-user to review, the content thus should be compact and clear. The output of decision making is usually numerously generated from algorithm, and such result is more beneficial for the scientists and analysts. Hence the compact form results need to yield for the end consumers in the presentation level.

5.1.1 Time-Series Data Cleaning in the Data Level

One of the aims is to discover user’s motion according to the phone’s attitude. With the help of motion/transition analysis, ADL patterns can be discovered. In each motion package, the time-series based sensor fusion algorithm needs to be applied on the sensory data from accelerometer, magnetometer, rotation, orientation, etc. Therefore, ADL in each position before and after each transition can be recognized on the fly. Furthermore, motion analysis can help recognize ADL through phone’s various attitudes.

Most earlier work in accelerometer-based activity recognition performed their experiments by placing multiple sensors on several parts of the subjects’ body Parkka et al. (2006) Bao and Intille (2004) Mannini and Sabatini (2010). Hip, wrist, arm, ankle are usually selected to attach sensors. Nowadays Android-based cell phones mostly contain tri-axial accelerometers. The acceleration in three spatial dimensions with the magnetometer sensor and Earth’s gravity can detect the
orientation of the device. An accelerometer-based activity recognition system Kwapisz et al. (2011) identifies physical activity a user is performing, such as walking, jogging, climbing stairs, sitting, and standing. This section is to determine the phone’s attitude and the walking direction by combining multiple types of sensors in addition to accelerometers.

Some practical real-world situations are worthy of taking account. Even holding in one’s palm, it does not really reflect motion direction. It is possible to detect a stride, while distinguishing whether the phone is heading directly same with the motion direction or totally opposite is another challenge. Sometime, walking backwards also happens.

What’s more, detected heading angle is always fluctuating when a smartphone is in the pocket heading downwards to ground. Obviously seen in Fig. 5.2, the detected raw data are very different from the correct walking direction in case:96. Thereafter, a time-series filter is needed. And time-series recognition over several intervals with feedback are necessary.

Some basic anchor points are:

- It is most likely to gain a correcting heading direction when a phone is in the horizontal viewing attitude (case:0 in Fig. 5.2).

- Determine the orientation in pocket by Kalman filter with current detected attitudes and how the phone is turning for viewing.

5.1.2 Time-Series Single-Source Data Correction in the Information Level

Error in the information level results from two source: improper recognition approach and defective data passing from data level. Different pedometer Apps have different method to count steps and calculate Calorie, yielding almost different values. All of those values can not be correct at the same time, thus causing errors.

Error in the data level can also affect the recognition result in the information level. Wi-Fi signal strength from a certain AP varies from time to time. Apparently, the strength of the magnetic field may vary a lot around some electric appliance, such as sound stereo receiver, refrigerator, colour
TV, etc. Also magnetic filed of door access control systems can interfere the orientation. These noise and interfere immediately affects the recognition results in the information level.

### 5.1.3 Time-Series Multi-Source Data Correlation in the Decision Level

![Error correction framework of time-series based sensor fusion.](image)

Figure 5.3 Error correction framework of time-series based sensor fusion.

Time-series analysis can serve as a feedback for calibration of the prior ADL results. In step with new ADL types, more new knowledge can be learned and stored in the system. Then we will develop a mechanism that utilizes the new knowledge to modify the unreasonable results recognized earlier in the decision level.
Figure 5.2 Accelerometer and orientation data (pitch, roll, azimuth) in various attitudes.
CHAPTER 6. SYSTEM DESIGN

6.1 The ADL Recognition Services Deployed Over the Cloud

While the front-end of the ADL recognition system, i.e., ADL Recorder App, has already been up and running for some time Feng et al. (2016), the application keeps evolving itself to embrace new types of sensor modules, or even new types of smart devices like smart watches, rings or smart wristband. More concretely, at the advent of the age of Internet of Things (IoT), the system design, implementation and maintenance face the following challenges:

- make the process of integrating new sensor type modules into the system as-is easy and less error-prone. At the micro level, this unfolds into issues regarding data storage format, the adjustment of the existing tables in the back-end databases,

- make the process of removing the dated, out-of-market sensor type modules from the system as-is easy and error-prone. Google Glass is such an out-of-market example.

- make the process of adding more analysis algorithms targeting new sensor data easy and error-prone, and

- lower the computational complexity for the pre-processing of a wide variety of data; simplify the process of audio data processing, acoustic feature extraction and result labelling, as well as Wi-Fi based positioning machine learning.

To measure up with these challenges, the system level design revolves around an agent-based information management platform (IMP) integrating Service-Oriented Architecture (SOA) over the cloud. It features an observers-pattern based ADL recognition server on the cloud to provide customized services to its subscribing parties and clients.
6.2 Agent-Based Information Management Platform (IMP)

Through the mechanism of publishers and subscribers, the observers-pattern features an event-based asynchronous communication paradigm. It decouples the service providers from the service consumers. Combined with a service-oriented architectural (SOA) design where service reuse is facilitated, they bring needed flexibility to our agent-based IMP to meet the above challenges. Under the observers pattern, our agent-based IMP presents a unified set of programming interfaces, e.g., protocols and APIs, to hook up the third party publisher and subscriber agents.

An example subscriber agent is the ADL Reviewer (see Fig. 6.1), which as its name suggests, offers the ADL record reviewing services. The ADL Reviewer empowers both PC-based web apps and mobile version web apps. An authorized family member, who takes the role of an observer, can use the ADL Reviewer to check out the historical ADL records of a particular observee. Pie chart-based statistical results are provided according to the observer’s queries, which carry constraints of temporal and location natures. For instance, the observer can review the monthly report with inferred patterns of the observee’s sleep activities. The observer may also choose animation view of the observee’s ADL data, a functionality named ”activities tour” offered by ADL Reviewer that revive the observee’s activities of daily living over a selected period of time.

The following list consists of the clients who consume the services provided by the agent-based Information Management platform (IMP):

- **Client application developer.** Client application developers focus on innovation, design, and optimization for the ADL Recorder App. The registry in IMP provides URLs for the client application programmers as interfaces to post raw data to the cloud infrastructure server. Thus, after embedding the uploading module (publish agent), ADL Recorder App readily publishes data packages to the IMP. In order to save data traffic, ADL Recorder APP offers an option for only uploading data under Wi-Fi environment.

- **Web developer.** A web developer implements content presentation on ADL Reviewer APP to serve the observers. The media includes both Apps on smart devices and website. The
Figure 6.1  Architecture of ADL recognition system on information management platform
IMP infrastructure generates the URLs for subscribers as interfaces, and each ADL Reviewer receives the high-level messages via subscribe agent. Thus the IMP turns a “black box” to web application developers, letting them implement independently.

- **Data analyst.** The data analyst is responsible for analyzing data with domain knowledge Ming et al. (2015) Bernstein (2011), and retrieving valuable information. The high-level outcomes are stored into a knowledge database, ready for broadcasting to the subscriber agents. The algorithms, applied by the data analyst, vary greatly from module to module, so module design in IMP is required here. For example, the developer for the localization module does not have to learn anything from the audio processing module.

- **System administrator.** The responsibility of the system administrator is to configure, maintain and administer the server storage and routine automation, such as backup, upgrade, maintaining security policies, troubleshoot, monitoring performance, etc.

- **Living domain expert.** Within IMP, living domain experts need to set some rules according to their living domain knowledge. Those rules are fundamental for the data analysts’ algorithm. After acquiring domain knowledge from the expert, a set of rules in the form of IF-THEN-ELSE are applied with machine learning algorithms.

The expert system Jackson (1998) includes the pre-rule and the post-rule. The pre-rule is, for example, to pre-process, reduce noise in the data, and add weightage onto the feature data before feeding into the machine learning algorithm. The post-rule is, for example, to diagnose and modify the obvious errors from the machine learning output. For instance, the preliminary result from the machine learning is that somebody gets hurt after falling, while shortly afterward a sensor message indicates this person is jumping and bicycling in one minute. Then the post-rule should ignore the mis-classification to improve accuracy. Post-rule can also serve as a translator from the classified labels to the understandable meaningful expression for the end users. The objective of this mechanism is to improve accuracy and expedite processing with less human labor.
CHAPTER 7. FULL-SCALE EXPERIMENTS

7.1 The Experimental Setup and Results

In a preliminary study, we tested our ADL recognition system in four apartments. Here we only give one such apartment of size 900 ft\(^2\) to illustrate, and the layout of which is shown in Fig. 7.1. We used a bedroom (Position 1), a bathroom (Position 3), and a living room (Position 8) with a combined kitchen (Position 5) for this work. A main Wi-Fi access point is on the top of the TV set in the living room (Position 8). The smart phone used is Nexus 5 made by Google and has the latest Android 6.0.1 Marshmallow operating system. The circles are ADL capturing points, and the arrow around circle denotes the most frequent facing orientation in each from observation.

We then collected much more data in a series of full-scale experiments, from real-life situations. When every ADL is conducted, the subject is required to select an ADL label on screen, which is quite similar to the think-aloud process. A total of 3511 standard ADL records (including indoor and outdoor) were stored, and only indoor records were selected to pass on to the recognition stage.

By conducting the experiment, we captured several frequent ADLs as follows:

**Working on desktop PC in the bedroom** - The subject often worked on the computer and did some reading in the bedroom. Here, the ADL Recorder App captured the sound of the keyboard typing and detected that the subject faced approximately east.

**Hygiene activities** - Once the inhabitant is inside the bathroom, he performed normal hygiene activities, brushing his teeth and did some washing. When the subject did the washing, he faced East. In the bathroom, the App is likely to record the sound of running water and flushing toilet.

**Cooking** - The subject cooked and prepared food facing South. Here the ADL recorder App recorded the sound of boiling water, boiler, jingling sound of cooking utensils, chopping and so on and so forth.
Washing dishes - The ADL recorder captured the running water sound and detected that the heading orientation was West, when the subject washes dishes, vegetable, fruits, and etc.

Eating - If just one subject was in the apartment, the sofa was the most frequently used furniture for dinning purposes. The ADL recorder captured the sound of kitchen utensils and detected that the heading orientation was South. We distinguished the breakfast, lunch, dinner and mid-night snack activities during their related period of time.

Wandering walk - the subject often wandered in the living room. The heading orientation and the time period varied. When the subject was on the move (walking or running), the value of "stepdetecor" was set into 1.0, which meant that steps were detected. Combined with the location and orientation information, we knew the whereabouts while the subject was walking and his heading orientation. The location information were obtained from Wi-Fi fingerprinting pre-stored.
in database while he was indoor. By the increased value of sensor STEPCOUNTER and the interval between two stepping records, the pace was calculated.

### 7.2 Full-Scale Experimental Results and Discussions

Here, we give one experiment for the discussion. ADL includes those activities conducted either in the kitchen, living room or bathroom, The types of activities corresponds to those described in section 7.1. bathroom belongs to "Hygiene activities"; breakfast, lunch, dinner, midnightsnack belongs to "eating"; washinginbathroom belongs to "Hygiene activities"; and cooking, chopping, washingdishes belongs to "cooking".

A total of 972 valuable instances were fed into J48 pruned tree to train a pattern. The number of correctly classified instances was 888, and the accuracy rate hit 91.36% by 10-fold cross-validation. The confusion matrix is shown in Fig. 7.2.

![Confusion matrix of activity classification](image)

The sound of running water from faucet always goes with flushing toilet, so that the sound is not pure to "using bathroom". However, the two ADLs have different heading orientations, and the predictions of "using bathroom" and "washing in bathroom" gain high accuracy. "Cooking" sometimes mis-classified as "Washing dishes" can be accepted, because the sound of moving utensils, running water, collision of dishes happens together. The group of "eating" ADLs is in high accuracy, except that a small portion was mis-interpreted between "midnight snacks" and "dinner". That is mainly because the characteristics of "midnight snacks" and "dinner" are similar, and time periods are close. The prediction mistakes of "washing dishes" spread to all "eating" groups and "cooking",
that is because all of those categories may have a similar sound, such as moving utensils, collision of dishes, etc.

ADL recognition performances in the other three apartments are 96.15%, 98.92% and 99.17%, respectively.

7.3 ADL Pattern Discovery

A visualization of ADL pattern for one resident is shown in Fig. 7.3. The size of colored circles represents the duration spent at each position, and the colored lines represent the light illuminance level in the surrounding. Bright color means higher value intensity and dark means lower. A pattern here narrates that starting from desk at the second 0, the subject walks to open the door at second 6. After walking through a dark hallway, he flips the light buttons at second 13 and 15. And then he picks something (newspaper) on the table at second 20 and walks to the couch at
96

second 45. In a preliminary study, we tested our ADL recognition system in four apartments, all of these correct recognition rates gained up to 90% or above.

The highly satisfactory performance makes it convincing to release our system as a real product, currently under consideration.

7.4 Real-life Services on the Prototype

Our real-life experiments were conducted in several places, including Iowa, Illinois, Wisconsin, Florida, California in USA, Tokyo in Japan, Beijing, Shanghai, Shenyang, Wuhan, in China, Taipei and Taoyuan in Taiwan district. The ADL Recorder App serves various types of customers. Health care centers for older adults use the ADL Recorder App to track residents’ routine ADL, and data analysts on life health discover the population’s health status and predict trends. Since each resident has his own routine databank, nurses and doctors will master the resident’s physical status and anomaly from ADL reports. Moreover, older adults and kids can be remotely observed, building real-time care links. Family and friends are able to form a social network to keep track of each other’s real-time ADL, increasing the intimacy and friendship. The ADL Recorder App can help researchers collect more life data and movement data to build a valuable big dataset.

Not only ADL Recorder App provides the recognition for regular ADL, new algorithm for more customized ADL types are invented for some particular users. For example, tracking when a 70-years-old woman walks her dog every day, and whether a housekeeper completes every duty.

This system exploits in depth the mobile data along with the usage of smart phone as normal life routines, compared to traditional smart home deployment and wearable sensors, the ADL recognition system increases zero burden for users.
CHAPTER 8. CONCLUSION AND FUTURE WORK

This thesis proposes an ADL recognition system by analyzing mobile data collected from a single smart phone. Our contributions include:

1. An ADL Recorder App is developed to capture mobile data from multiple embedded sensors, including microphone, Wi-Fi scan module, heading orientation, light proximity, step detector, accelerometer, gyroscope, magnetometer, timestamp, etc. The aim of the ADL Recorder App is to collect the phone user’s behavioral context and environmental context where the user resides. This study makes it possible to collect ADL data just by one standard smart phone, which can be accessible to many common users.

2. Key technologies in the ADL recognition system cover audio processing, Wi-Fi indoor positioning, proximity sensing localization, and time-series sensor data fusion. We invent different algorithms for different module, such as situation-aware computer audition analysis, Wi-Fi based indoor localization, visible light based localization, multi-source sensor data fusion. Each module achieves high recognition accuracy. At the end, outcomes from different modules are merged together to derive final ADL identification results.

3. We propose an agent-based Information Management Platform (IMP) to host the ADL recognition server, which is in the SOA-based publish/subscribe operational mode. IMP largely decouples web-based applications, as a large application can be decomposed into several modules. Different publisher clients can post information to IMP independently, and subscriber clients will benefit from a well-defined interface to access the high-level integrated information. The ultimate goal of IMP is to deliver agility to support any distributed information sharing application.
4. The *ADL recognition system* not only paves a road for recording, detection, recognition, and electronic documentation of user’s ADL, but also yields some health-related reports through statistical analysis. Experiments in real-life situations prove the feasibility of our *ADL recognition system*.

5. Various types of machine learning classifiers and data pipelines are applied in each processing module. Bayesian network, hidden Markov model, Gaussian mixture model, random forest, k-nearest neighbors are used in situation-aware audition module. J48 decision tree and Bayesian network are applied in LiLo visible light based localization. Online clustering algorithm is invented for global wide localization. J48 pruned tree is utilized for final ADL recognition.

6. Specialization in everything from front-end to back-end is involved in this full-stack development. It requires Java language for Android App development, Swift language for IOS App development, Java-based server on the Amazon Web Services (AWS) to receive data package, load into data warehouse, transform as the input to MySql database. Representational state transfer (RESTful) web services provide interoperability between AWS and publish/subscribe clients. Feature extraction and processing of audio, Wi-Fi localization, and other raw data are conducted by Python scientific language. C# language with .NET MVC 5 are selected to publish ADL results for *ADL Reviewer App*, which is used by subscribing customers. A bunch of JavaScript framework is employed for the representation website, such as AngularJS, JQuery, Telerik Kendo UI, AmCharts, Bootstrap, etc.

7. Modular programming facilitates the construction of this large system. Each module, such as audio processing, indoor localization, has its own independent functionality with interface to the whole system. During the development, each module experiences interchangeable upgrade gradually.

There are, however, still several limitations in this thesis: The optimization problems on the client side consists of minimizing battery consumption, reducing data package transmission. Higher recognition accuracy on both each processing module and the overall final ADL results are always
needed. Currently the ADL Recorder App captures sensor data in every 3 minutes. Although it meets the requirement in this study, it can not fit for highly real-time needs, for instance, fall detection. Although the interal duration is programmable, high frequency consumes electricity power largely. Thus, we face a tradeoff between performance and efficiency.

8.1 Overview of Future Work

In order to build a sound theoretical foundation for a practical product, the ADL recognition system should achieve both a high-level academic standard and a ultimate satisfaction from real users, so improving the recognition accuracy should be a highly critical task. As smart phones are usually with users all the time, the ADL Recorder App works as a quiet background service to record both indoor and outdoor ADL in various environments. After analysis and statistics on the mobile data, the system could recover user’s every ADL details and assessment of life habits.

As more users continuously provide data, the knowledge database for the unfamiliar environments should grow alongside. We will consult with more living domain experts to enhance the dataset. In addition, related API for new researchers and programmer to retrieve the dataset will be readily provided.

Now that the ADL Recorder App has evolved into a stable version in our work, data are accumulated day by day from real-life users, data analysis becomes a driver in our future work. Building a store of big data and support for analytics in order to further research are in process.
Publications


of Life Technology Center, Q. (2010). Grand challenge data collection.


Technologies for Perception of Humans, volume 4122 of Lecture Notes in Computer Science, pages 1–44. Springer Berlin Heidelberg.


