BEHAVIOMETRICS FOR MULTIPLE RESIDENTS IN A SMART ENVIRONMENT

By

AARON S. CRANDALL

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

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
Department of Electrical Engineering and Computer Science

MAY 2011

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of AARON S. CRANDALL find it satisfactory and recommend that it be accepted.

Diane J. Cook, Ph.D., Chair

Lawrence B. Holder, Ph.D.

Behrooz Shirazi, Ph.D.

ACKNOWLEDGMENTS

First and foremost I need to recognize the support from my wife, Krystal. Without her solid backing this dissertation would never have been possible. Also, I need to acknowledge my son, Perry. His continual inquisitiveness reminds me on a daily basis of the joys of learning. Thank you both.

My advisor, Diane Cook, has been instrumental in keeping me focused on my goals. Smart homes are fertile ground for new and innovative projects and we only have so much time in our days. Without such a guide this work would never have come to fruition.

Alongside Dr. Cook is a panoply of educators who have poured endless, and sometimes frustrating, hours into my growth. Mr. Chambers for showing me how fun science could be, it was a special moment every time your class began. Thank you for going through the effort to bring us the moon rocks, as very few moments in my life were as memorable as those looking through that microscope. Mrs. Burke-Hengen for never giving up on an 8^{th} grade outcast. Mrs. Burgess for making home room enjoyable. Mr. Noble and Mr. Cotton for guiding a herd of runners through high school and beyond. Mr. Martel for making physics the best course in high school and understanding when we needed a place to hide from the rest of the student body.

Dr. Lu for telling an incredulous, barely passing undergrad that he would do great in grad school. I did not believe you at the time, but you were right, as always. Dr. Osterburg for sharing some of what grad school could be like. Dr. Hauser and Andy for showing me how to be a great professor. One more of note is Mrs. Lincoln for showing me how be a better teacher. Not all lessons are the ones you intended to give, but your students are always learning.

Next, I want to recognize all of the great friends I have had through the years. There are too many to mention, but there are a few notables that need naming. Forehead, chromatic, Brett, asmith, Jim S., Kerry, JD, Cole, Holly, Neeru, Allison and Dave!, you made the years fly by through thick and thin. Jim K., thank you for all the M&Ms. Bacon and Ivan, you are right behind me and I expect great things. La, you are right behind Bacon and I still expect great things. Also all of those in offices like Joy and Cheryl who helped me navigate the ever-looming bureaucracy.

Last, but by no means least, is my family. To my parents, Marilyn and Dan who were always proud of me. My sister, Meredeth, whose accomplishments warrant pride. And my grandparents who were educators, givers, and builders. No one could have better teachers and supporters.

There is no way to express the gratitude I have in my heart for all of you.

Thank you.

BEHAVIOMETRICS FOR MULTIPLE RESIDENTS IN A SMART ENVIRONMENT

Abstract

by Aaron S. Crandall, Ph.D.
Washington State University
May 2011

Chair: Diane J. Cook

Smart homes and ambient intelligence show great promise in the fields of medical monitoring, energy efficiency and ubiquitous computing applications. Their ability to adapt and react to the people relying on them positions these systems to be invaluable tools for our aging populations. This work introduces and explores solutions for issues surrounding real world multiple inhabitant smart home situations. Dealing with multiple residents without requiring wireless tracking devices, while paying heed to privacy concerns, is a difficult proposition at best.

The Center for Advanced Studies in Adaptive Systems research group has developed and tested a number of novel technologies to address the issues of multiple

inhabitants within a smart home context using inexpensive, low profile, privacy sensitive sensors. These smart home implementations, when combined with artificial intelligence tools, are designed to provide localization, tracking, and identification through behaviometric approaches that are useful and deployable in real world situations. They have been evaluated using unscripted living spaces with multiple residents, and their capabilities explored as a means of benefiting other modeling tools, such as detecting the Activities of Daily Living.

Given the complex nature and diverse needs of smart home technologies, the tools presented here are by no means definitive solutions to handling multiple resident smart environment situations. However, they do provide a strong working base for the continued development of smart environments with demonstrable benefits on real-world implementations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Dedication

For the generations that came before me, your efforts were not in vain.

Especially my parents, Marilyn and Dan.

CHAPTER 1. INTRODUCTION

1.1 Background

"Smart homes" represent a rapidly maturing field of study as well as a looming business market. Its concepts are being applied to a wide range of medical, social and ecological issues. The vague definition of "smart home" allows for numerous implementations and variations to exist. At its core, a smart home is any living space that involves sensors, controllers and some kind of computer-driven decision making process. With this loose definition in hand, the research, medical and business communities have been highly creative in leveraging this concept for their various needs.

Smart homes, including the ambient intelligence and ubiquitous computing fields, are an extension of home automation, a phenomenon that has existed as long as there have been engineering-minded people with permanent homes. Any kind of automated tool or machine within a living space can be considered home automation. Devices such as dish washers, clothes dryers, automatic mixers, timed lawn sprinklers, motion-detector controlled lighting and automatic doors are all forms of home automation. What these kinds of devices have in common is that they are directly managed or deterministically controlled by the humans around them. They do not take in information about the current environment to make decisions about what their behavior should be. The addition of a proactive and intelligent decision maker to the aspects of home automation is what produces a smart home.

Ideally, a smart home is subtle in its operation and conforms [Mozer, 1998, Rashidi and Cook, 2009b, Cook et al., 2003] to the residents without detrimentally impacting their lifestyle. The system should take in information about the home environment and attempt to build models about the activities and interests of the residents. Designers of smart homes normally have a particular objective in mind when choosing sensors, controllers and computer algorithms. These objectives have included medical monitoring [Rialle et al., 2002, Maurer et al., 2004, Tolstikov et al., 2008, Liolios et al., 2010], energy efficiency [Chemishkian, 2002, Chetty et al., 2008, Chen et al., 2010] and measurement of social interaction within the space [Wren et al., 2007, Ivanov et al., 2007b, Wigdor et al., 2007]. Every smart home to date has been custom built, though the market for these systems is reaching a state where off the shelf commercial offerings are viable. As the business sector matures, standards of communication, sensing and control will begin to emerge to provide a more stable long-term research framework for future smart home projects.

The area with the greatest long-term feasibility for smart home commercialization is health care, though energy efficiency has a strong future in reducing our home's economic and ecological footprint. For the health care community, the ability to monitor older adults in their home to support "aging in place" [Marek and Rantz, 2000, Mynatt et al., 2004, Cook, 2006] for older adults is of significant interest. A side effect of our medical knowledge and social dynamics is that the average age of the developed world's population is increasing rapidly [Keyfitz and Flieger, 1991, Tuljapurkar et al., 2010]. Within a few decades the largest age groups in the United States will be in the 65+ years old categories [U.S. Census Bureau, 2009]. The current professional nurse graduation rates in the US are declining [OECD, 2009] and our health infrastructure is not growing to meet this expected wave of older adults. Promoting techniques for people to live in their home longer, i.e. aging in place, possess the capability of blunting the negative impacts of this older population. Smart homes are expected to be a tool for dealing with this looming social and economic issue.

To make smart homes capable of supporting these goals, the research community has focused on building technologies for the detection of the Activities of Daily Living (ADLs) [Philipose et al., 2004, Fogarty et al., 2006, Logan et al., 2007, Libal et al., 2009, Singla et al., 2009, 2010], resident tracking [Yiu and Singh, 2007, Crandall and Cook, 2010a] (see Section 2.1 for details), resident identification [Crandall and Cook, 2010b] (see Section 2.2 for details), medical history building [Cook, 2006], social interaction [Chen et al., 2004, Cook et al., 2010], resident mental evaluation [Cook and Schmitter-Edgecombe, 2009] and many others. In the last decade, smart homes have gone from being a small field of researchers to a large and vibrant field of private and public development.

1.2 Problem Statement

This thesis addresses two topics that almost all real-world deployments of smart homes must address in some way: tracking of multiple residents, and determining the residents' identities. With a single resident in the smart home this is a trivial problem, but multiple residents transform it into a serious issue. As soon as a second person (or other entity, such as a pet capable of causing sensor events) enters the smart home space, the multi-resident issue becomes critical. At this juncture, the smart home infrastructure must be designed to either function well in the face of several sources of data, or to differentiate between the sources by some means. If the system ignores the multi-resident problem, unaccounted for residents show up as noise in the data. In most cases, this noise in the data will lower the accuracy of the model building and interfere with operational quality. It will likely cause failure of high quality history building, preference generation, ADL detection and many other computer generated models. Finding a means to address the multiple-resident problem is a current and pressing issue for the smart home field.

1.3 Purpose of this Study

The challenges of tracking multiple residents and identifying them are introduced separately in this work. A set of algorithms is introduced to address both issues, and applied to these problems with the goal of creating tools capable of operating in long-term real-world smart home applications. In choosing and designing these tools, a number of factors were weighed in each case. This blend of theoretical and practical considerations is a common theme in smart home research.

Various smart homes have approached the issue of tracking and attribution of events in a wide variety of ways. One common approach is a wireless tag/device on the resident to localize them [Hightower and Borriello, 2001c]. Alternatively, they use a camera and image processing to find the resident's relative position within the room [Ge and Collins, 2008] or some kind of proximity system where the sensors detect the physical presence of the resident [Helal et al., 2005]. These different means, with their respective benefits and costs, are all discussed in detail in Section 2.1. In this thesis the CASAS Technology Platform is introduced, along with a number of algorithms that count and track multiple residents simultaneously within the smart home space using only low profile, passive, proximity sensors.

For identification, smart homes have often used some form of wireless tag/device to garner a unique label for every resident [Mori et al., 2004], or have applied both active [Jain et al., 2004c] and passive [Jain et al., 2004a] biometrics. Alternatively, a small number of projects have used behavior [Crandall and Cook, 2008c] to identify individuals. Once the system is able to uniquely identify individuals, it can then proceed to build medical and preference histories, and perform more accurate anomaly detection to support features such as detecting medical emergencies. Again, these different approaches have benefits and negatives that are discussed in depth in Section 2.2. This thesis introduces a group of algorithms to identify individuals using their behavior and behaviometrics as seen by the CASAS sensor platform.

1.4 Theoretical Framework

Because smart homes must deal with multiple residents generating events, there will always be a need to delve into strategies for dealing with this issue. After researching the state of the art in sensor platforms, real-world deployments and algorithms, the CASAS technologies were developed with the objective of handling multiple residents and a number of other smart home applications. By exploiting the fact that people occupy different parts of the sensor network and behave in different ways at different times, artificial intelligence techniques should be able to differentiate and identify those individuals.

Few smart home implementations take this approach, though we have demonstrated that humans involved in annotating the CASAS data sets are rapidly able to track and identify the residents of the spaces. If a human is able to intuit the locations and identities of the current residents with only a small amount of experience, then computational algorithms should be able to do it as well. As shown in Chapter 2, these kinds of differentiation have been successful in other fields for tracking and behavior identification, so some form of their application should be effective for similar tasks in the smart home context.

1.5 Research Hypotheses

This thesis has two hypotheses. First, it is possible that given a living space with a number of proximity sensors to report interactions with the residents, an algorithm can be designed to determine the number of residents, localizes them and builds tracklets of events representing an individual's path through the sensor space. Second, the behavior of different people is algorithmically differentiable, leading to the ability to identify residents via behavior alone.

Given such algorithms for tracking and identification, they could be used to reduce the data noise induced by multiple inhabitants. The noise reduction should demonstrably benefit other algorithms doing more complex model building, such as ADL detection and preference history building. Altogether, this becomes a foundation for handling multiple inhabitants with only proximity sensors.

1.6 Importance of the Study

As described in Section 1.2, until the issues of tracking and identification are addressed, smart homes will continue to under-perform in the face of multiple residents. Given that there are many different sensor platforms and desired goals for smart homes, the research community needs to continue building a suite of available tools to address this problem. The approaches introduced in this work use a passive, proximity-based sensor platform to perform tracking and identification. Application of algorithms that require no explicit action of the residents, such as carrying wireless devices, and are more protective of privacy than video-based solutions, fill a key niche in the smart home field.

1.7 Scope of the Study

In this thesis we intend to evaluate both hypotheses put forth in Section 1.5. In pursuit of this goal, a set of smart home testbeds using the CASAS technology platform were installed in a variety of residential contexts, and occupied by diverse residents. Data were annotated for number of residents at any given event, the identities of the residents, and the activities being performed. A number of algorithms for both tracking and identification were defined, implemented and tested for these data sets. The benefit of these algorithms at improving ADL detection was also tested to demonstrate their ability to improve the overall capabilities of deployed smart homes.

1.8 Summary

Smart homes and smart home technologies have a beneficial place in the future of society. Because these technologies can be inexpensively and robustly installed into existing homes, people will be able to derive direct benefits in their daily lives from this research field.

This work introduces a suite of tools designed to give a means of tracking and identifying the residents of a smart home space. There are numerous methods that attempt this, but given the considerations of privacy, cost, and long-term robustness, the CASAS strategy of using passive proximity sensors with appropriate artificial intelligence algorithms has distinct advantages when being installed in private homes. Further, these algorithms and tools have the power to advance the smart home field as a foundation for more refined tools. Their introduction will yield improvements to other applications, such as ADL detection and aging in place services.

CHAPTER 2. RELATED WORK

Target localization, tracking and identification are all significant research problems. They are often a complex mixture of hardware and software solutions. The smart home field has drawn on previous work in a number of other fields when seeking solutions to these open issues. Given the intricate nature of smart home environments and the diversity of the residents, the research to date is still in the early stages of development. This chapter attempts to define the existing ecology of work in both the smart home and related fields to demonstrate the latest in localization, tracking and identification techniques.

Robotics localization is a vibrant and deep research field. Within this body of work, a number of approaches have been applied to smart home needs. These have included wireless triangulation, video processing and proximity sensing. The latest works in tracking research are introduced in Section 2.1.

Unique human identification as applied to smart homes is a younger field than tracking. The types of data available about the entities being identified quickly limits the strategies that can be used to uniquely identify a given resident. Outside of electronic tagging, the algorithms are related to the fields of biometrics, behavior mining and behaviometrics. A survey of human identification strategies published to date is presented in Section 2.2.

2.1 Localization and Tracking

Several strategies have been employed to track individuals in a smart home space. The goal of a smart home tracking system can vary between applications, but it is most often used to associate sensor events with the resident who is most likely causing them. For example, if one of the multiple residents is in the kitchen, then events from the sensors nearby should be attributed to that individual and not someone else in another part of the home. With this capability, the smart home will be better able to build histories for individuals and perform activity or anomaly detection. Choosing sensors and algorithms to successfully track residents is a part of most smart home implementations.

In their survey paper [Hightower and Borriello, 2001c], Hightower & Borriello define three categories of location-sensing techniques. The goal of their work was to categorize all of the approaches to localization, which quickly leads to algorithms for tracking of individuals. Their taxonomy is summarized in Figure 2.1. These different categories are discussed in deeper detail in their earlier works [Hightower and Borriello, 2001a,b] and their topics include physical vs. symbolic location, absolute vs. relative positioning, location of computation, cost considerations and common

- *Triangulation* can be done via *lateration*, which uses multiple distance measurements between known points, or via *angulation*, which measure angle or bearing relative to points with known separation.
- *Scene analysis* examines a view from a particular vantage point.
- *Proximity* measures nearness to a known set of points.

Figure 2.1: Categories of location-sensing systems.

limitations of these kinds of systems.

Each of these categories is well represented by various smart home projects as a means of tracking within a space. During the design phases of the CASAS smart home technologies, examples of each category were considered before finally settling on a proximity style approach. The factors in this decision included privacy, resident acceptance, installation lifespan (power and durability issues) and implementation difficulty. These are all issues that every smart home platform must address at some point.

2.1.1 Triangulation

Triangulation [Linde, 2006] is a widely researched approach for remotely localizing a wireless transmitter. It is used in many applications such as search and rescue [Goodrich et al., 2008], HuffDuff ship tracking [Stephenson, 1999] and satellites [Schmid, 1974]. Aside from handling the variability of wireless transmissions, once the distance from three known points has been determined the mathematics behind triangulation are very simple, making it an attractive solution for smart home localization.

The use of a device and base stations to perform location-sensing is well established in smart homes or device localization. The first indoor badge-based system was the Active Badge sensing system [Want et al., 1992]. It uses infrared transmissions from a identification badge form factor wireless device that receivers within an office space could use to identify the current room where the device is located. This kind of system is limited to line of sight and is sensitive to infrared interference, common under fluorescent lighting. To gain a more accurate location for the resident, different approaches needed to be considered.

The Active Bat [Harter et al., 1999], the Cricket Location-Support System [Priyantha et al., 2000] and the later work by Nishida et al. [Nishida et al., 2004] all use ultrasound time of flight to triangulate a device's location. The Active Bat system works to an accuracy of 9cm, and Nishda's system is accurate to 5cm, while the Cricket is only useful to 1.2m x 1.2m regions. These audio-based solutions also suffered from noise interference. There are a number of common devices that would impact their performance, thus making them less useful to large scale smart home deployments. While audio systems provide a higher fidelity data source than the Active Badge, the field at large has moved primarily to RF-based solutions.

The most common RF triangulation approaches leverage 802.11 devices. These are widely available and use open standards to operate, giving the community more choices and products from which to build. A very early 802.11 platform was the RADAR project [Bahl and Padmanabhan, 2000]. The researchers at Microsoft Research Laboratories™ used RF signals with multiple base stations to triangulate the location of a 802.11 device indoors. It worked for a single floor to within three meters of accuracy, and similar implementations were eventually commercialized. When confronted with multiple building floors their approaches become non-trivial to solve and the carried device is power intensive, leading to a relatively short operational lifespan before recharging.

There have been many advances in using 802.11 for determining location since the RADAR Project was published. The proliferation of wireless devices and the reduction in power consumption has made using 802.11 based systems much more feasible for long smart home deployments.

The Nibble Project [Castro et al., 2001] introduced Bayesian Networks with 802.11b as a means to learn room locations within a building from a set of data points, while LaMarca, et al. [Lamarca et al., 2005] introduce means for a wireless network to learn the location of devices automatically given enough time. In Haeberlen's work [Haeberlen et al., 2004], a method of sampling rooms was used to build a model for the space, and derived a 95% accuracy in a three floor, 1,161 m^2 area. After applying a Markov localization approach with a Gaussian fit sensor model, the system was able to place a device within a 2.7m x 4.9m cell. In Ladd et al. [Ladd et al., 2002b], 802.11b was used to track a device moving in an indoor space. This system was found to have a resolution of less than 1.5m and an accuracy of 83% when given a suitable base station layout. The authors note that these systems use a set of training data sampled from the existing environment, and that the algorithms assume no serious changes in the environment, including people moving about the space. Accommodating drift is a known learning algorithm issue that few of the tracking works address to date.

Since these earlier works were published, the field of 802.11 localization has been very active [Ladd et al., 2004, Letchner et al., 2005, Youssef and Agrawala, 2005, Jacquet et al., 2008, Lassabe et al., 2006, Woodman and Harle, 2008, Lassabe et al., 2009]. Efforts have been made to deal with the continual changes in the wireless strength map [Bolliger et al., 2009], and to accommodate the interference caused by the bodies of the people around the wireless devices [King et al., 2006]. There have been several strategies explored to automatically determine localization and mapping within spaces [Ladd et al., 2002a, Krumm and Platt, 2003, Chai and Yang, 2007, Lorincz and Welsh, 2007, Barry et al., 2009, Park et al., 2010]. Additionally, the Rice Wireless Localization Toolkit [Haeberlen et al., 2004] provides an established platform for determining device locality within a 802.11 network. Finally, new techniques for fusing multiple kinds of wireless protocols to determine location have been proposed [Bolliger, 2008, Aparicio et al., 2008, 2009].

Triangulation using RF is not limited to 802.11. There have been similar works using ZigBee [Blumenthal et al., 2007, Lihan et al., 2008, Alhmiedat and Yang, 2008, Navarro-Alvarez and Siller, 2009], Bluetooth [Bruno and Delmastro, 2003, Cheung et al., 2006, Bargh and de Groote, 2008, Jevring et al., 2008, Diaz et al., 2010], and GSM [Otsason et al., 2005, Chen et al., 2006, Varshavsky et al., 2007, Bolliger, 2008]. These protocols use essentially the same approaches as the 802.11 techniques. They vary in the devices available, range and behavior of the systems. What does not vary is that they use an active device carried on the object being tracked. For smart home applications this means placing a powered device on the person, or people, moving through the smart home. Because the device itself is being tracked, if the person leaves it behind or the power source drains, the home can no longer locate that individual. If the home is relying heavily on the device for localization of residents, it loses a significant portion of its functionality once that happens. Given that people will eventually forget, damage or fail to recharge their device there is in inherit limitation in the use of powered devices for tracking residents within the home. To surmount these issues, the CASAS platform chose not to use powered devices affixed to the residents for localization and tracking.

It is also significant that using localization to properly attribute events to individuals in the smart home requires a relatively high accuracy. Elnahrawy's work [Elnahrawy et al., 2004], proposes that there is an inherent limit to the actual resolution available to all wireless triangulation strategies. Most smart homes are highly unregulated when it comes to wireless interference, so this limitation may make all RF solutions untenable outside of tightly controlled environments.

A popular alternative wireless solution that has been gaining significant support are Radio Frequency Identification Devices (RFID) [Glover and Bhatt, 2006, Finkenzeller, 2003, Want, 2006]. These technologies are designed explicitly to track people and objects [Weinstein, 2005], giving them an advantage over many other RF systems as a localization platform. RFID has existed since the 1970s and has become ever more prevalent as a tracking tool. These systems give a universally unique identifier to every RFID chip and are readily available in off the shelf products.

Notable quantities of research has been invested into using RFID as a triangulation platform, not just a proximity one [Choi et al., 2009, Zhou and Shi, 2009, Akhlaghinia et al., 2009, Chen, 2010]. With ranges of $5\frac{1}{2}$m [Buettner and Wetherall, 2008] for passive RFID systems, most typical home indoor living spaces could deploy these tags without encountering wireless power issues. This gives the ability for a smart home to offer long-term tracking of objects and people without the requirement to change batteries or provide recharging capabilities for the tracking system.

The CASAS project seriously considered deploying an RFID-based tracking system. The price is reasonable, the devices are robust and do not require regular power source maintenance. However, RFID does have a few long-term issues. The first and foremost is issuance of tags for new objects in the space. For example, if residents are using tags in their shoes [Mori et al., 2004, Kodialam et al., 2007, Roberti, 2009] then receive new shoes, new RFID tags must be installed and the tracking system updated for the new tag serial numbers. Similarly, with a RFID-enabled medication reminder system [Agarawala et al., 2004], the pill storage system has to be kept up to date as new medicines are introduced and old ones removed. If the pharmacy or retailer provides an integrated system, it becomes much more feasible to deploy these kinds of RFID-based tools.

RFID has been a contentious topic since its inception. There are numerous privacy issues raised by subjects carrying unique identifiers in public [Garfinkel et al., 2005, Garfinkel and Rosenberg, 2005, Ohkubo et al., 2005, Lee and Kim, 2006, Klasnja et al., 2009]. These concerns must be considered when deploying any RFID-based tracking system, and smart homes are no exception.

2.1.2 Scene Analysis

Scene analysis tracking in smart homes is used to determine a number of aspects of the space. These include, but are not limited to, localizing residents and objects. From that information, more complex models can be built to support ADL detection [Tolstikov et al., 2008], resident identification [BenAbdelkader et al., 2002], dangerous situations [Fleck and Strasser, 2008], social behaviors [Chen et al., 2004] and other tools. Using video cameras for scene analysis is a well established part of smart home research and has some powerful advantages over simpler sensor strategies.

The localization and detection of objects is primarily aimed toward providing the context of the current resident behaviors [Krumm et al., 2000, Tabar et al., 2006, Brdiczka et al., 2007, Tolstikov et al., 2008, Libal et al., 2009], although tracking of residents is also a goal [Brumitt et al., 2000, Drummond et al., 2003, Snidaro et al., 2005, Ivanov et al., 2007a, Köhler et al., 2007, Ge and Collins, 2008, Liu et al., 2009, Li et al., 2009]. By using a scene analysis system to gather data about the current smart home space, object interaction may be derived along with the tracking of individuals. This approach also makes the system much easier to install initially, as the objects in the space do not need to have tags or devices attached as they would for wireless tracking systems.

The trade-offs of video-based systems include privacy implications, cost of devices, handling changing lighting conditions and processing time. There is a body of work that studies how accepting people are of various smart home technologies in their home [Fisk, 2001, Demiris et al., 2004, 2008, Gaul and Ziefle, 2009]. The results of these works reveal that many residents are not willing to have video or audio recording devices installed in their private spaces. The added granularity of information provided by the vision-based solutions requires significantly more computer computation time over a triangulation or proximity solution, and the current technologies have difficulty handling severe lighting changes invariably found in homes. If the smart home is designed have a low energy footprint and/or low cost, the additional infrastructure required to process images from one or more cameras may be prohibitive.

Video-based approaches were considered for the CASAS research testbeds. Having all of the information available about the space would be similar to having a human watching and recording the smart home all day every day. That resolution of data carries almost everything needed to model the space, but it requires an entire set of algorithms to interpret the information before it can be used by model building algorithms. Without going through some kind of information reduction, many artificial intelligence and machine learning algorithms are unable to cope with the large volume of data available. By avoiding a scene analysis approach and working with proximity detection, the CASAS tools were much simpler from the outset.

We also saw benefits when deploying these systems in the real world. During our work, we have placed CASAS-style systems in multiple private homes of volunteers. Every single family was openly pleased that there would be no cameras or audio systems placed in their homes. Including a video camera of any sort would make the smart home infrastructure much less palatable to many people. The researchers with the CASAS group are happy that scene analysis approaches continue to be researched, but privacy and acceptance of monitoring systems will always be a concern to address.

2.1.3 Proximity

The notion of using proximity to track a resident is very common in smart home systems. It may not be explicitly stated, but whenever a project requires that only a single resident may be present and does not use a scene analysis or triangulation approach, it is using proximity. Whenever an event occurs, it implies that the resident is proximal to that sensor. With only a single resident in the space, the latest event caused implies their location and context. Examples of this situation are common [Cook et al., 2003, Tapia et al., 2004, Logan et al., 2007]. By limiting the smart home to a single resident, researchers are able to bypass the complex issues of attributing events to multiple individuals. This allows them to study other aspects of the smart home systems without explicitly addressing these tough issues, but ignore the implications of the single inhabitant constraints on tracking and event attribution.

There are a handful of systems that make use of proximity to track the residents of a smart home space. The most common approach to date is the use of a "smart floor." With a smart floor, pressure or vibration sensors are placed under the floor's surface in the residence. The sensors are read and the resulting map of pressure values may then be used to infer the location of one or more residents.

The earliest smart floor system is the ORL Active Floor [Addlesee et al., 1997]. This system was built to support a number of applications, including tracking, resident identification and security systems. It was only built in a small prototype deployment, but it did demonstrate the promise of using the floor as a means to detect objects and people.

The Georgia Tech Smart Floor [Orr and Abowd, 2000] uses a system very similar to the ORL Active Floor and expands on the algorithms needed to interpret the sensor readings. Since the ORL Active Floor and the Georgia Tech Smart Floor, there has been continued interest in leveraging smart floor technologies [Pirttikangas et al., 2003, Mori et al., 2004, Murakita et al., 2004, Fukumoto and Shinagawa, 2005, Helal et al., 2005, Kaddoura et al., 2005, El Zabadani, 2006, Eyole-Monono et al., 2006, Aipperspach et al., 2006, Yiu and Singh, 2007, Savio and Ludwig, 2007, Wen-Hau et al., 2008, Valtonen et al., 2009]. As the field began to explore this approach, it has had to deal with a number of issues including cost, sensor wear and complex algorithms to determine the location of residents as they move about the space, as well as the relocation of objects in the home. Given their success, smart floor technologies will likely continue be investigated as a means to sense the behaviors and locations of residents within the smart home.

A smart floor approach was considered for the CASAS research platform, but the monetary cost of the implementation was prohibitive. Installing such a system as a retrofit into an existing home is very difficult. More modern materials technology has made it more feasible since this was considered in 2006 [Senanayake et al., 2007]. Additionally, the research at the time did not indicate a strong future in smart floors for tracking of individuals. A number of newer projects have made headway with the strategy since then.

Other proximity techniques include using RFID [Wang and Liu, 2005, Fishkin et al., 2005, Buettner et al., 2009] or motion detectors [Aipperspach et al., 2006, Crandall et al., 2008, Crandall and Cook, 2010a,d]. When using RFID-based tools, residents must either carry a RFID reader or a tag with them. This requirement introduces similar problems to the issues cited in Section 2.1.1. Alternatively, motion detectors and other ambient sensors allow the resident the ability to live without direct interaction with the smart home infrastructure. This choice of proximity sensing is privacy protecting, lower cost, passive and unobtrusive to the residents' daily lives. The combination of these advantages is balanced against a more difficult algorithmic modeling environment. Since the residents are of unknown location and number, a proximity approach with passive sensors creates a number of obstacles that must be overcome before it can be truly useful in real-world smart home deployments. These issues and some algorithms required to apply this kind of system are discussed further in Chapter 4.

Particle Filter Tracking Algorithms

A vibrant area of robotics and object tracking is centered around Particle Filter (PF) algorithms. Using Particle Filters, a number of hypothesis are generated in a probabilistic manner that predict the current world state. Depending upon the placement, type and number of sensors a PF-based solution may be classified as triangulation, scene analysis or proximity. PF approaches have been shown to be robust for a number of tracking situations, including single robot [Dellaert et al., 1999], multiple mobile objects [Schulz et al., 2001, Khan et al., 2003] and multiple mobile robots [Rekleitis, 2003, Rekleitis et al., 2003].

These kinds of algorithms have been successfully employed in the smart home context for tracking of residents [Hightower and Borriello, 2004, Murakita et al., 2004, Yu et al., 2006, Yun and Kim, 2007, Salah et al., 2008] and for forms of activity detection [Woodman and Harle, 2008, Pham et al., 2008, Yu et al., 2009]. The latest CASAS resident tracking algorithm introduced in Section 4.3.3, uses a Particle Filter to approximate the current location of the residents.

2.2 Individual Identification

Identification is the process of uniquely identifying an individual, or grouping of people by behavior. In the smart home context, this is an invaluable part of handling multiple residents. Without the ability to identify or classify the residents, it is very difficult to attribute events to an individual's history and to predict future behaviors. There are a number of approaches and methods available to identify people with various degrees of accuracy and specificity.

Historically, smart home research has taken one of two paths to identifying the resident. Research environments often require that there be only a single resident at a time [Rahal et al., 2007], side-stepping the issue for the purposes of doing research on other problems. This approach succeeds in allowing work on areas of smart homes such as ADL detection and medical profiling. In the end, though, this only delays addressing the issues inherent to multiple resident situations that will eventually impact the capabilities of smart home technologies.

The second approach is to use a carried wireless device that provides a unique identifier for the resident [Matsushita et al., 2000, Venkatesh, 2008]. This suffers from the issues as discussed in Section 2.1, but does function in controlled environments and situations where the need for the smart home features exceeds the overhead of maintaining the carried devices. Ideally, a system that does not require a carried device, but does allow for multiple residents would be developed. Several approaches have been proposed and explored for unique identification [Crandall and Cook, 2008a,b,c, 2010b] and some for grouped behavior classification [Yu, 1999, Heierman III and Cook, 2003].

2.2.1 Biometrics

Most people are familiar with direct biometrics for unique individual identification. These include fingerprints [Galton, 1888, Cole, 2001, Woodward Jr. et al., 2003, Jain et al., 2004c], retinal/iris scans [Sims, 1994, Woodward Jr. et al., 2003, Reid, 2003, Cense et al., 2004, Jain et al., 2004c], and DNA fingerprinting [Woodward Jr. et al., 2003, Burke et al., 1991, Butler, 2009, Jain et al., 2004c]. There are devices available to take measurements of individuals for all of these [Faundez-Zanuy, 2004, Jain et al., 2004c] and, to some degree of accuracy [Phillips et al., 2000], uniquely identify them. In a smart home system with a limited number of people there will be few incorrect identifications using these tools. The primary difficulty with these approaches is that they require that the person being identified take specific actions. For example, the subject might have to place one or more fingers on the fingerprint scanner or look directly into the retinal scanner and remain still until it finishes taking measurements. These kinds of forced behaviors will be burdensome in a private home environment and likely avoided over time, leaving the home without key information needed to properly function. Additionally, if the resident is not mentally capable of performing the actions with any consistency, such as in the case of individuals with dementia or the very young, these tools provide little useful information to a running smart home system.

A more recent field being developed that allows identification of subjects without requiring explicit actions on their part is passive biometrics. These approaches hold more promise for smart home applications due to their position as ambient tools instead of obvious and intrusive ones.

Examples of passive biometrics include facial recognition [Li and Jain, 2005, Wechsler, 2006], body shape (Anthropometry) [Bertillon, 1896], voice recognition [Rose, 2002], footstep shape [Orr and Abowd, 2000, Pirttikangas et al., 2003, Murakita et al., 2004, Helal et al., 2005], personal weight [Jenkins, 2006, 2007, Jenkins and Ellis, 2007], height [Jenkins, 2006, Srinivasan et al., 2010], heart beat pattern [Watanabe et al., 2009], and blood pressure [Begg and Hassan, 2006]. There are also a number of known esoteric biometrics that could be exploited in certain environments [Woodward Jr. et al., 2003] such as: vein patterns, facial thermography, sweat pores, hand grip, fingernail bed, body odor, ear shapes, skin luminescence, and brain wave patterns. Taken individually, these characteristics are often not able to uniquely identify an individual, but when used in conjunction with each other, and possibly augmenting a more direct biometric system [Jain et al., 2004a,b], they can be a strong tool for providing a unique identification.

Some of these approaches, such as facial recognition and body shape, rely on cameras to function. This requirement may or may not be acceptable in the private home space. They also require a number of different sensors installed throughout the home [Kim et al., 2006, Jenkins, 2006, Srinivasan et al., 2010], which may or may not be feasible in the given home implementation.

Similarly, the voice recognition approach requires one or more microphones within the home. Voice printing is a well established field, but the privacy issues with audio recording and the ability to reliably record audio in a noisy home environment limits its utility in smart homes.

These physical measurements have the benefit of being unique to individuals with a high degree of probability. As people grow and change, their physical characteristics change. Any smart home implementation must take this measurement drift into account during long term usage.

Additionally, genetically identical twins will cause serious issues if the system is relying heavily upon physical characteristics. Handling this edge case has been investigated to a limited extent [Jain et al., 2001], but smart home identification technology is still in its infancy so little has been published to date for dealing with identical twins.

2.2.2 Behavioral Biometrics

Another form of passive identification identifies an individual based upon their behavior on the sensor network. This kind of identification approach is not as well established as a biometrics or wireless device tagging, but a growing field dubbed "behaviometrics" is forming around the idea of identification via behavioral biometrics [Wang and Geng, 2010]. Using behavior to classify individuals into likely groups is a skill all humans possess. It is how we determine likely threats or friends at a glance among a crowd. Building computer algorithms to make the same kind of prediction is a natural extension of sensor-based systems.

Old uses of behaviometrics to identify individuals such as handwriting recognition [Srihari et al., 2001] and gait recognition [BenAbdelkader et al., 2002, Collins et al., 2002, Alwan et al., 2003, BenAbdelkader et al., 2006] are now being augmented with more advanced sensors and algorithms. This allows for a number of new methods when attempting to recognize an individual. A new approach provided by the now-common interaction with computers is to identify the user of a computer system via their interaction with the mouse [Shen et al., 2009], keyboard [Nisenson et al., 2003, Moskovitch et al., 2009], software tools [Gamboa and Fred, 2003] or network [Gamboa et al., 2004]. These kinds of classifiers have distinct limits due to their probabilistic nature. They have applications in access control, intrusion detection, security systems and medical data analysis.

There have been few papers within the smart home field to date that use behaviometrics to identify individuals [Rodríguez et al., 2008, Menon et al., 2010]. New tools combined with the numerous data sources of a smart home system allow the opportunity to determine a resident's identity via interaction with the smart home. This is done in a manner similar to identification accomplished through behaviometrics-based approaches applied to interaction with a computer terminal. The difference is that the "terminal" is envisioned as the whole smart home sensor platform instead of only a keyboard and mouse.

The CASAS project uses ubiquitous, passive and simple sensors to enroll individuals in the behaviometric system for future identification [Crandall and Cook, 2008a,b,c, 2009, Cook et al., 2010, Crandall and Cook, 2010b,c]. Given a unique historical profile, a resident can then be re-identified in the future using behavior alone. This work is discussed in more detail in Chapter 5.

CHAPTER 3. CASAS TECHNOLOGY PLATFORM AND TESTBEDS

The sensors, controllers, software, and infrastructure are unique to every smart home. The CASAS project has assembled a collection of hardware and software tools for the implementation of a smart home. Since smart homes are so dependent upon the specifics of their implementation, the research results they produce need to include a quality record of how the data was derived. This record needs to include details of the hardware used, the software infrastructure, the space in which it was installed, any pertinent details about the resident and the conditions under which the data was gathered. This chapter attempts to provide the needed details about the CASAS smart home implementation, in order for future researchers to have a good sense of how and why the algorithms tested may have operated as they did.

This thesis leveraged the CASAS Technology Platform (CTP), a smart home implementation designed to yield a comprehensive smart home system. "Comprehensive" means that the whole space is monitored for the residents' behavior. From the moment a person enters the home, they are interacting with the CTP. This comprehensive approach contrasts with a number of smart home platforms wherein only a distinct subset of the environment is "smart." These smaller projects are valuable in their context, but the CTP is designed to provide a ubiquitous sensing environment throughout the home.

The CTP leverages a variety of sensors, providing the computer monitoring the space with a number of benefits and limitations. These sensors are listed in some detail in Section 3.1, and important details related to their supporting infrastructure are provided. Given that every smart home implementation uses a combination of off the shelf products and custom built parts, this record of how the space is sensed is very important.

The software built for the CTP is an agent-based system. Different sensor types and various software tools are represented by one or more agents that communicate between one another over the CASAS Lightweight Middleware (CLM). The middleware communications infrastructure is covered in Section 3.2.

For long term storage of the data gathered by the sensor platform, a database-backed architecture has been implemented. This database archives the events for later use by researchers and annotators. Given the wide range of sensors and data types, a simple and open style of data representation has been used to store the data. The database and data representation are discussed further in Section 3.3.

3.1 CASAS Sensor Platform

The sensors and infrastructure used in the CTP are designed to be simple, robust, energy efficient, low profile and generally more socially acceptable then smart home technologies discussed in other works [Fisk, 2001, Demiris et al., 2004, 2008, Gaul and Ziefle, 2009]. This is a significant number of goals, but they can be achieved through a careful selection of hardware and algorithms to interpret the available data. Many of the devices utilized are off the shelf commercial products integrated into a variety of data buses to be read by the server. The resulting data events are sent over the middleware for processing and storage. By leveraging existing, well tested devices the robustness, energy efficiency and profile of the system are often improved. Custom sensors may be tailored to fit the needs of the smart home exactly, but the system designer often pays the price for these in terms of engineering time and a less robust final product. Commercial products are also often packaged well, so their visibility profile after being installed is lower and the residents are less likely to notice the system after they become accustomed to it. Participants who have resided in smart home testbeds observe that it only takes a week or two to become comfortable with the new technology in their homes.

The sensors communicate with the server through a handful of different data paths. Most of the sensors are attached to a Dallas 1-wire bus™, which allows for high speed transfer of small data packets along a daisy chain of devices on a common serial bus. To attach the sensors to the 1-wire bus, a custom board was designed and implemented to support a contact switch or stand alone 1-wire chip. This board was dubbed the Lentil Board and is present in the PIR motion sensors, temperature sensors and door sensors of all of the CASAS testbeds. An advantage of the Lentil Board is that it allows for easier connection of a variety of devices without serious modification for most applications. The board is shown in further detail in Section 3.1.1. All of the sensors connected to the server using the 1-wire bus share a single software agent to report their activities.

A handful of the other sensors use RS232 (serial), USB (Universal Serial Bus) and power line signal injection to communicate with the server. These all have their own agents to report events and provide an interface to communicate with them.

3.1.1 Lentil Board

The Lentil Board is an in-house bridge that allows simple sensors to connect to a Dallas 1-Wire Bus. This bus allows up to hundreds of devices on wire lengths of up to hundreds of meters. For a smart home implementation on the scale of most private residences, this is more than sufficient. The wiring specification devised for the Lentil Board also includes a 12v bus to provide power to higher power devices, such as the PIR motion detectors in the current CTP implementations.

The Lentil Board illustrated in Figure 3.1 is capable of serving as a bridge for a PIR motion detector (shown in Figure 3.2), as a contact switch (shown in Figure 3.3) with a Dual Addressable Switch (Dallas part #DS2406) 1-wire chip, or an ambient temperature sensor (shown in Figure 3.8) with a Parasite-Power Digital Thermometer (Dallas part #DS18S20). These sensors make up the bulk of the CTP implementation, with the PIR motion detectors most common. By providing a uniform interface for communicating with the core sensors for the smart home, the system becomes easier to install and maintain. An added advantage is that 1-wire parts are very common on the commercial market, so additional kinds of sensors and devices can be implemented as long as they come in the three pin package the Lentil Board accepts.

All 1-wire devices have a universally unique 64 bit serial number encoded at production time. This number is used by the CTP system as a unique identifier, relieving the smart home installers from having to determine their own serial numbering scheme.

For connection, Lentil Boards provide footings for two RJ11 jacks. This is a standard United States phone jack, which allows for common phone wire and connectors to be used when installing the devices in the home. This common standard makes it easier to find parts and tools to work with the system. The result is faster training for new people installing a system, less expensive wiring, and readily accessible tools,

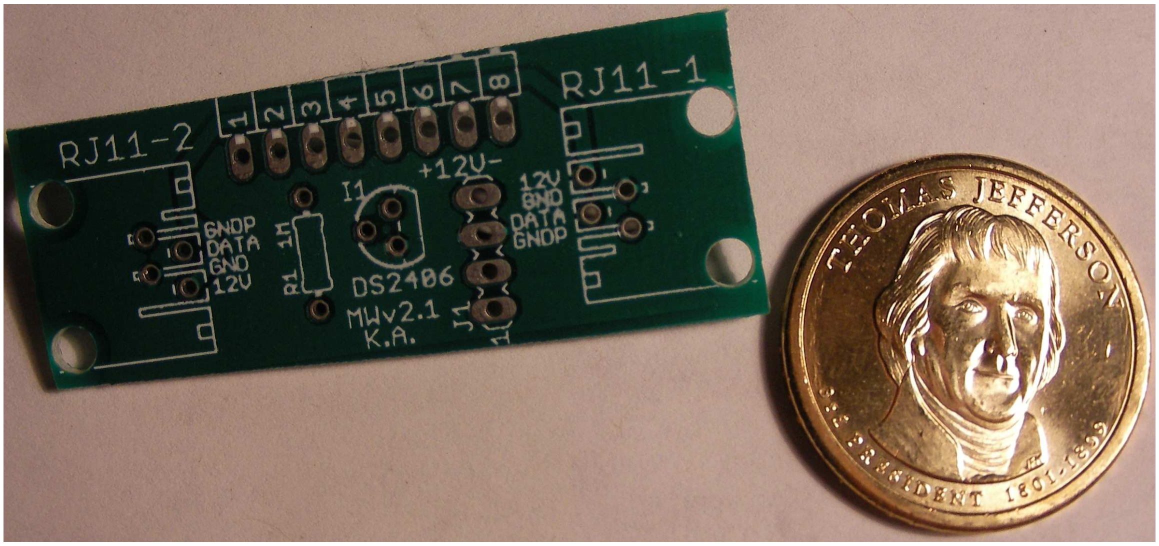


Figure 3.1: The Lentil Board itself. United States Presidential $1 included for scale.

such as crimpers and line testers.

Lentil Board A2D

A second Lentil Board was designed for use with voltage-based sensors, shown in Figure 3.4. This board commonly uses the Quad A/D Converter (part #DS2450) and provides four channels of analog to digital voltage signal conversion. The board has been installed as the stove burner sensor and the water flow sensor in CASAS' *Kyoto* testbed (see Section 3.4.2). Given difficulties of analog to digital signal interpretation and the relative rarity of sensors needing this type of bridge, the Lentil Board A2D sees little use in the more recent CASAS testbeds.

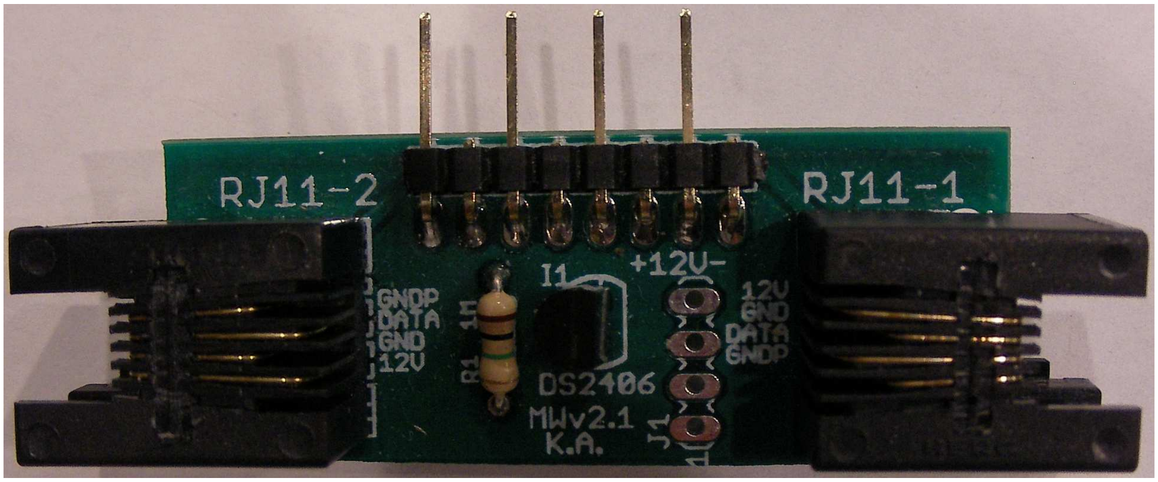


Figure 3.2: Lentil Board configured for integration with the PIR Motion Detector. It mounts directly into the case without modification to the sensor's terminal block.

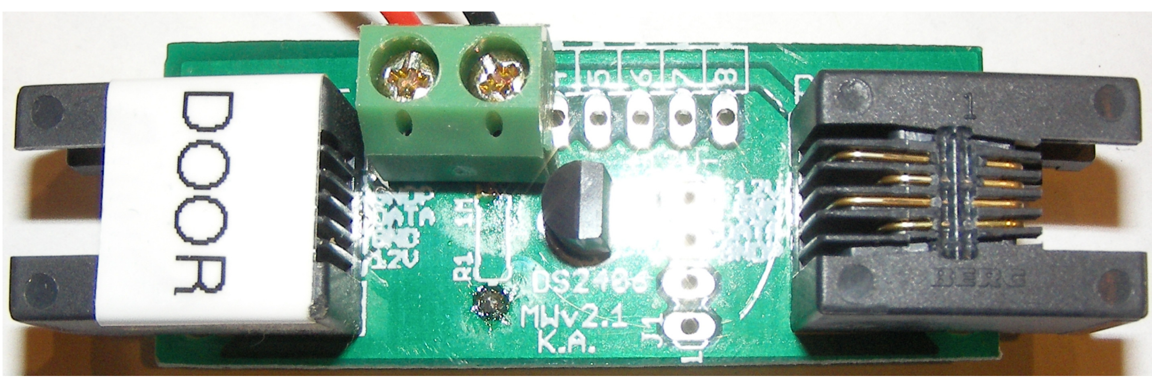


Figure 3.3: The Lentil Board configured for use with contact switches and door closure sensors.

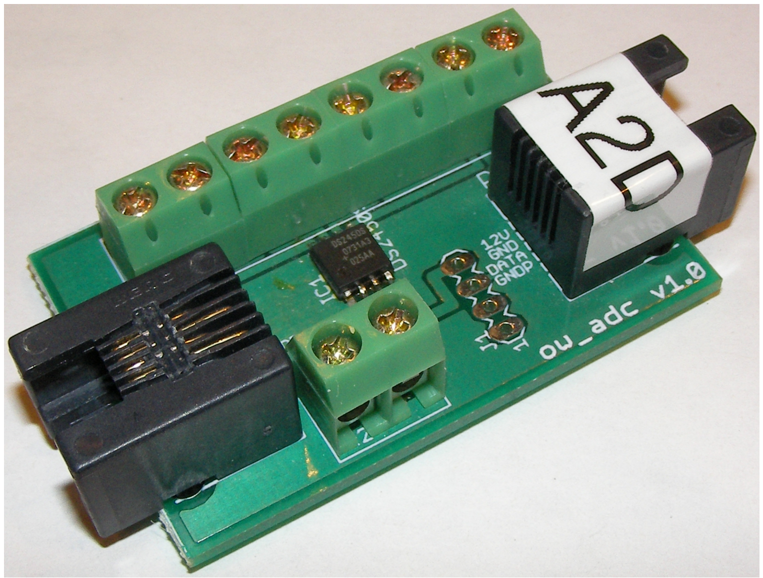
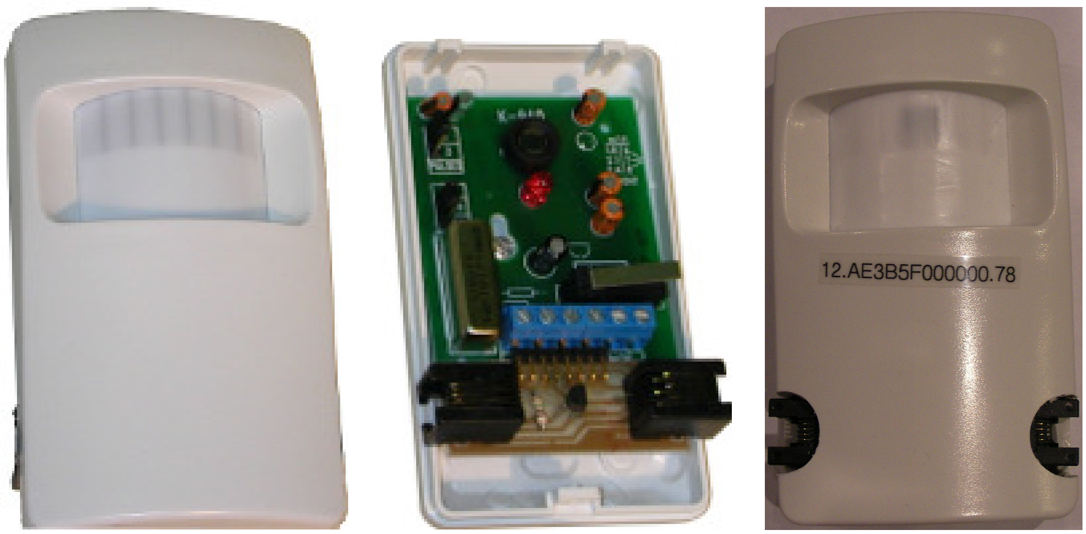


Figure 3.4: The Lentil Board A2D.

3.1.2 PIR Motion Detector

The PIR Motion Detector currently used by the CTP is a Visionic™ model K-940. This device is designed for general purpose home security installations and is somewhat "pet-immune." Normally, these are installed on the wall providing a lateral, human-height field of view. When installed as part of the CTP they are used in one of two modes: as area sensors or downward facing sensors. The detectors also have a Lentil Board, as shown in Figures 3.2 and 3.5(b), installed in them. This provides their power and communication to the rest of the CTP.

The PIR Area sensors are normally positioned so that their field of view encompasses, and is limited to, a single room. The goal of the placement is that the sensor only fires when a resident is within that given space. The result is a sensor that signals events about room occupation, but with no detailed information regarding where or by how many. A stock area sensor is shown in Figure 3.5(a).



(a) A PIR area motion detector.

(b) CTP PIR internals, exposing Lentil Board.

(c) A downward facing PIR.

Figure 3.5: Various versions of the CTP PIR motion sensor platform.

In contrast to the Area sensors, these units are mounted on the ceiling of the residence, with the lens facing downwards. Consequently it only sees the floor directly below it, thereby only sensing a small part of the room and giving any events it generates more locality. With this locality being derived from the events, the system becomes a much stronger proximity tracking system, similar to the ones discussed in

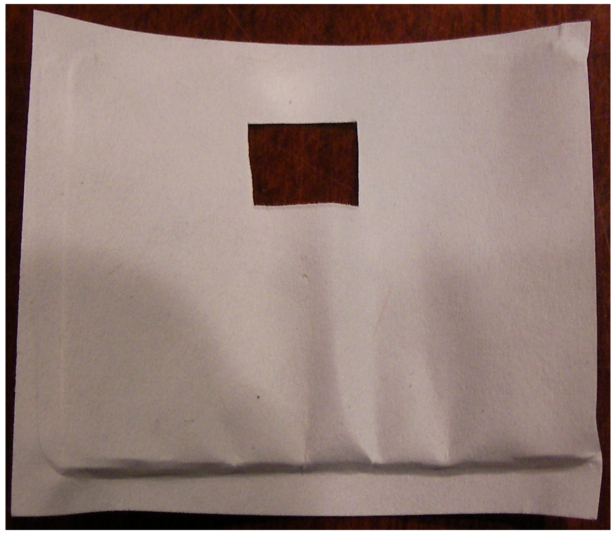


Figure 3.6: An example of the paper used to occlude the view of a CTP motion detector.

Section 2.1.3. By knowing with greater detail where a resident is, more specific context about which objects they are near is available, as well as providing the ability follow their movements more closely. This is why the downward facing motion detectors are the most common sensors in the CTP, as there needs to be enough of them to cover the space adequately.

Downward facing motion detectors have three modifications from the vendor's intended use. The first is the Lentil Board for power and communication, as was done with the Area sensors. The second modification is having its serial number affixed to the front of the sensor's case, as pictured in 3.5(c). This is the 1-wire serial number of the Lentil Board installed in the unit pictured. Having the serial number available makes it easier for debugging and testing of the smart home system.

The last modification is to occlude the view of the sensor to a smaller region. By default, the K-940 sensors have a roughly 120° lateral and a 40° vertical view. This is much too wide for the intended CTP use of this sensor; by reducing the view down to a much smaller aperture, a more focused view of the space can be created. To accomplish this goal, a piece of paper is inserted behind the Fresnel Lens. This paper is fashioned, as shown in Figure 3.6, so that when placed on the ceiling the sensor can only "see" a roughly 1.2m x 1.2m area of the floor. The hole behind the Fresnel Lens is visible as a dark square in the middle of the lens as shown in Figure 3.5(c). After this is done, a field of motion detectors can be mounted, facing downwards, creating a grid of binary presence sensors. The net result is a much finer-grained localization of motion than a single Area sensor could yield.

3.1.3 Insteon Power Line Controls

To control standard United States home appliances and lighting, the CTP has relied upon Insteon™ power line control devices. These devices are similar to the earlier X10 devices commonly used in other smart home systems [Bucceri, 2004, Meyer, 2005, Pacifica, 2005]. While the X10 protocol is well established and devices that leverage it are widely available, the protocol itself is slow. Current smart home applications send significantly more traffic to and from controllers than earlier, simpler X10 installations. With these additional demands, the X10 bus becomes saturated and commands may take many seconds to be enacted. X10 is also very brittle in the face of interference. While Insteon-based controllers are not immune to signal line interference, they are significantly more robust than their older X10 counterparts.

The Insteon devices used are capable of acting as light switches, dimmers and in-line power controllers. For the CTP approach they are used simply as smart switches that report all state changes, such as a user pressing a switch to turn on a light, to the computer listening on the network. Once these switches are installed in a testbed, the middleware is informed of all interactions residents generate with them. Additionally, these provide the ability for intelligent agents to control powered devices throughout the home.

As previously stated, one advantage of using commercial off the shelf products is their lower visibility profile compared to custom built implementations. This is primarily because they are often very similar to the existing controls in most homes. For example, as shown in Figure 3.7(a), the most common Insteon controller resembles a standard US paddle switch. It has additional lights to report the current dimming level, if available, but is otherwise externally similar to switches most people in the US use for controlling lights and devices in the home.

(a) Insteon light switch. (b) An Insteon in line lamp power controller. (c) Insteon controller with more user options.

Figure 3.7: Some of the more common Insteon devices used by the CTP.

The other Insteon devices shown in Figures 3.7(b) and 3.7(c) are used by the CTP as well. Example 3.7(b) is an in-line controller. There is no external switch to turn the power on or off, so all controlling of the device plugged into this in line module must come over the Insteon communications bus. These instructions can come from other Insteon switches, such as the more complex six button switch shown in Figure 3.7(c), or from a computer with a specialized controller that can interact with the Insteon message bus directly. This kind of in line module allows for direct computer control of lights not on existing switches, or other powered devices in the home.

Because it uses an existing consumer product, the CTP has experienced fewer engineering issues in monitoring and controlling powered devices within the smart home system. Insteon devices are Underwriters Laboratories (UL) certified, allowing them to be used in homes with fewer concerns regarding fire and electrocution than with a custom solution. The CASAS research testbeds have experienced some interference with these systems, but the offending appliance can often be located and isolated quickly.

Future updates to the CTP may see wireless power control systems come into operation. ZigBee-based solutions, such as Control4™, are becoming more economical and provide even more communication capabilities then power line solutions like Insteon or X10.

3.1.4 Ambient Temperature Sensor

To measure the ambient temperature in a room, the Lentil Board can be configured with a Dallas DS18S20 chip, as shown in Figure 3.8. This 1-wire temperature sensor reports the surrounding temperature to within $\frac{1}{2}°$ C. These are most often installed on the ceilings of rooms throughout the smart home, a strategy that provides insights into how heating and cooling systems operate, plus the impact of open doors and windows on the environment. These chips have also been used to verify the be-

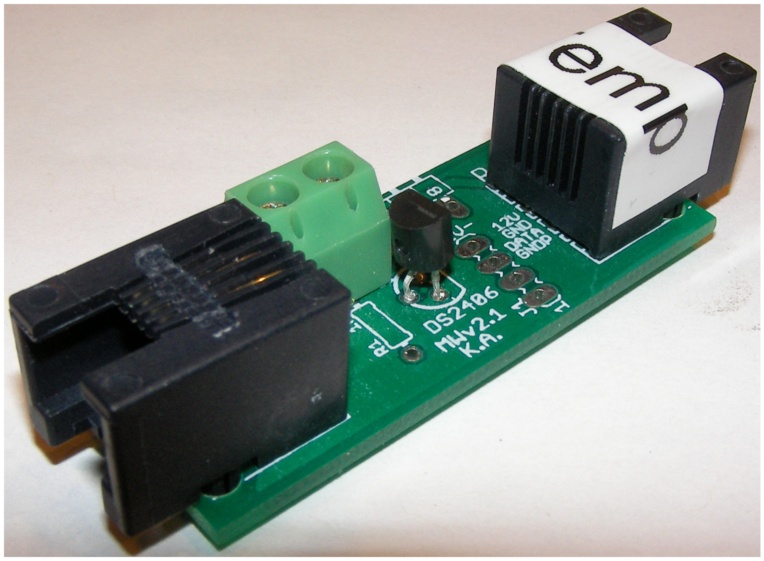


Figure 3.8: A CTP Ambient Temperature Sensor.

havior of other sensors, e.g. the Stove Burner Sensor (Subsection 3.1.7), where the temperature sensor above the stove corroborates the increased power used to run the burner.

3.1.5 *Magnetic Door Sensors*

The CTP uses simple magnet-driven reed switches to detect the opening and closing of doors, such as bedrooms, kitchen cabinets and refrigerators. These are common and inexpensive home security items. They are robust and simple to use when attached to a Lentil Board, as shown in Figure 3.9.

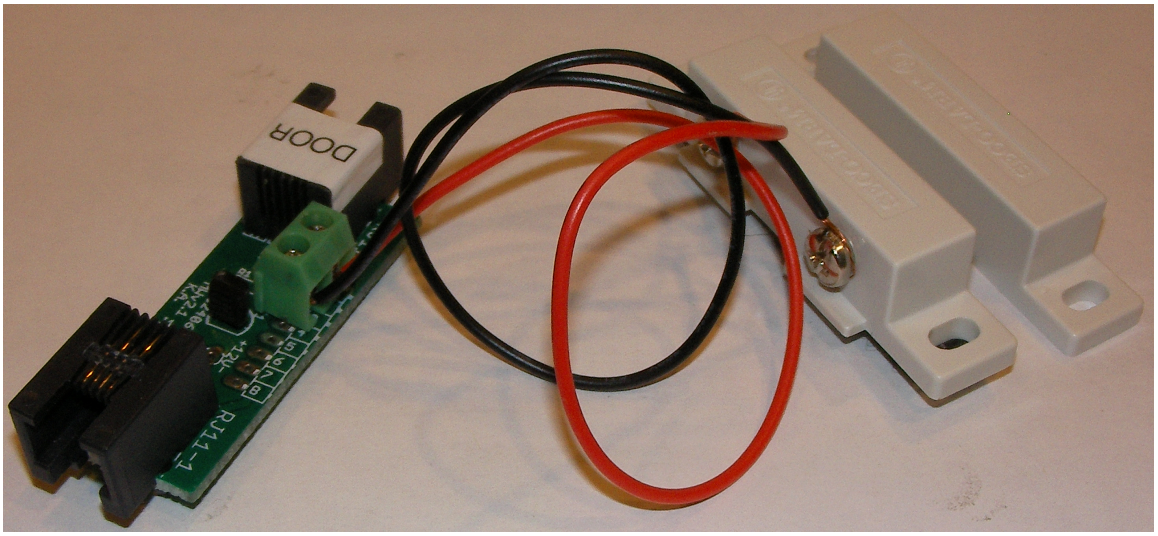


Figure 3.9: A CTP Door Sensor with a reed switch attached.

Whenever the magnet moves away from the reed switch, it closes, changing the state on the 1-wire chip. This change is then reported to the server which sends an "OPEN" event out over the middleware for processing and storage. When the magnet moves back into place, a "CLOSED" event is created. This simple system installed at entrances to rooms and buildings provides a stronger source of evidence for entrances and exits than motion detectors alone.

3.1.6 Item Presence Sensors

A simple contact switch and plate was designed to detect the presence of notable items throughout the home. Items of most interest include medicine dispensers,

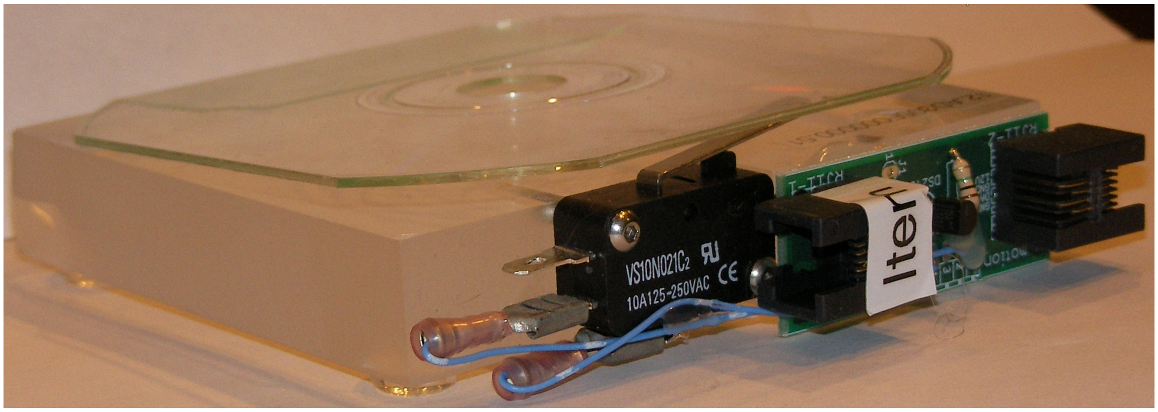


Figure 3.10: A CTP Item Sensor with Lentil Board attached.

cookware and toiletries, though any movable small object may be used with this device.

The item sensor uses a Lentil Board configured to read a contact switch. When the item is placed or removed from the plate, the switch is depressed and an event created. This simple design, shown in Figure 3.10, is primarily used for more controlled testing of specific items within a testbed. These contact plates have been applied to medication, cooking, cleaning and leisure items.

3.1.7 Stove Burner Power Meter

The first system to make use of the Lentil Board A2D platform was an inductive coil used to detect current flow through a stove burner in the *Kyoto* testbed. This

Figure 3.11: An example of the CTP Water Flow Sensor.

device is designed to measure the use of the stove burner. The resulting voltage measurements are interpreted to give both a duration and power setting when the residents are cooking.

3.1.8 Water Flow Sensor

To measure the use of the sink in the *Kyoto* testbed's kitchen, a pair of water flow sensors were installed. These commercial products from Lake Monitors™, pictured in Figure 3.11, were placed on both hot and cold inflow pipes to the sink. As water flows, a voltage is generated and sensed by a Lentil Board A2D to be reported to the server. The ability to detect the use of water in kitchens, bathrooms and other spaces is very valuable to ADL detection in smart home spaces.

3.1.9 OneMeter Power Metering

The OneMeter™ device monitors an inductive coil to determine the current wattage and cumulative kWh passed through a wire. These are normally installed at the main power feed to a breaker panel as a means to monitor all of the sub-circuits' power use. The version used by the CTP communicates with a computer via a serial RS232 connection, which allows the computer to poll for the current power status. An image of the OneMeter head installed in the *Kyoto* testbed, as described in Section 3.4.2, is shown in Figure 3.12.

3.1.10 Sensor Platform Summary

The CTP uses a wide variety of simple, robust sensors to give a diverse flow of information about occurrences within the smart home. These sensors provide easy to store and process events. This simplicity aids in modeling the space with humans and algorithms. Incorporating a range of sensor sources allows for a variety of approaches to sensor fusion and model building to support the gamut of smart home research topics investigated with the CTP.

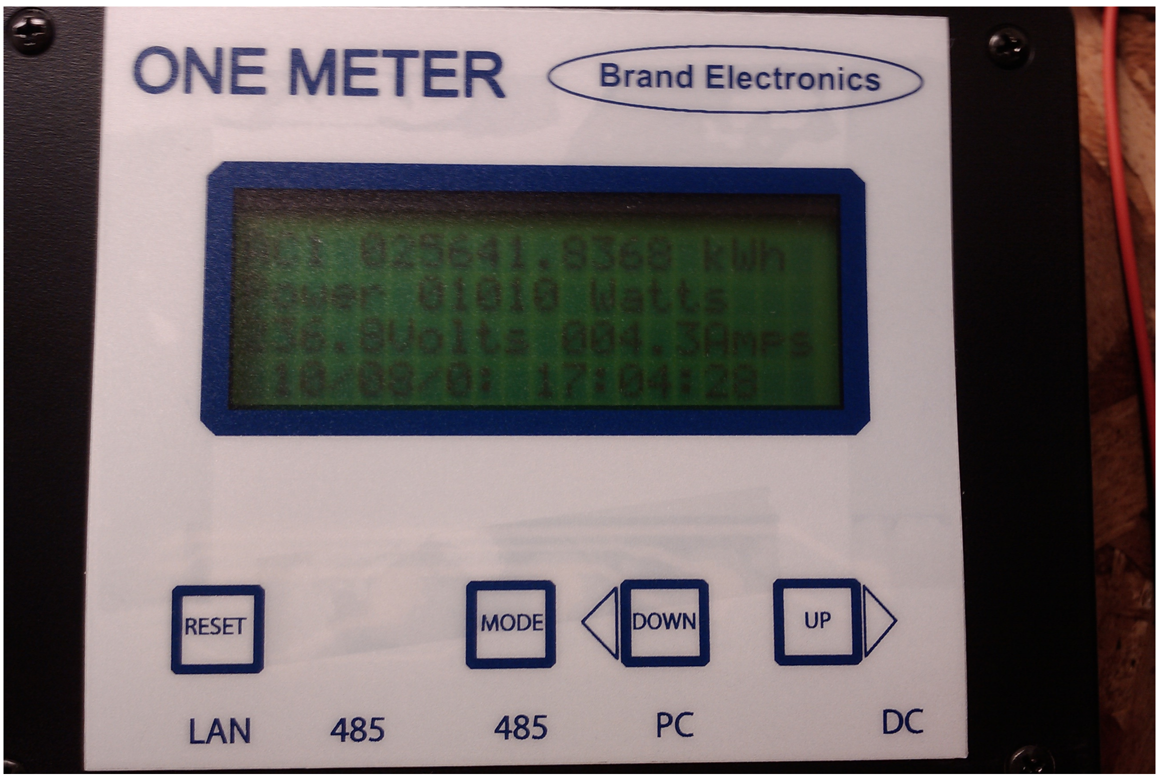


Figure 3.12: The *Kyoto* testbed OneMeter installed and operational.

3.2 CASAS Middleware

As mentioned earlier in Chapter 3, the software infrastructure used by the CTP is an agent-based pattern. Each sensor type, middleware component and model builder is represented by an agent. This creates a network distributed architecture bound together by a well documented message passing language. The benefits include isolation of components and additional modularity when adding and removing parts.

This component of the CTP is documented in depth in Kusznir's Masters Thesis [Kusznir, 2009] under the title "CLM as a Smart Home Middleware." As noted in the Definition of Terms (Appendix A), CLM stands for "CASAS Lightweight Middleware". This section of this thesis touches on the main aspects of the system, but leaves the details to Kusznir's work.

The CTP middleware uses the XMPP protocol [Saint-Andre, 2004a,b] as the messaging and presence layer. This approach means that the CTP can use any full featured XMPP server, such as ejabberd, Wildfire or Jabberd 2, to manage the interconnection of agents and passing of messages. These software packages are implemented and tested as long-term, Internet-facing, high speed, high volume systems, hardened for security. While they do incur greater overhead than a more dedicated implementation using a message passing library, choosing to use these kinds of tools has led to reduced development, maintenance and security costs.

The agents implemented in the CLM system use XML formatted messages to communicate with one another. The schema of these messages is laid out in Kusznir's Thesis, Chapter 2. Essentially, there is a standard language for passing messages to a channel manager agent, called "Manager," which then forwards them onto the various agents subscribed to that channel. There is a handful of out of band messages for specialized control systems, but the preponderance of messages move through the Manager. The individual behavior of the agents implemented for the CLM are detailed in Chapter 5 of Kusznir's work.

3.2.1 Distributed Clocks and Event Timestamps

Time and clocks are an open issue with any distributed message passing system. The CLM is no exception to this. To solve this problem, many smart home projects synchronize the system clocks of the various computing platforms and rely on short experiment durations to mitigate the impact of clock skew. Since the CASAS testbeds are long running and centralized in nature, all network messages are handled by a local network. This made it feasible to use the clock of the main server as the authoritative source of time. As events arrive at the Manager agent, it stamps the current time on the event before passing it along for recording and processing. If the CLM were to be used over a wide area network or a larger scale building, this issue of time and causal

Table 3.1: Data provided by every event, including system and control messages, passed through the CLM for storage.

Field	Notes
date	ISO 8601 format (yyyy-mm-dd)
time	ISO 8601 time format (hh:mm:ss.subsec)
serial	Unique text identifier for sensor reporting
message	Value of sensor event
by	Text identifier of CLM agent reporting the event
category	CLM event category of event {entity, state, control, system}

ordering of events would need to be re-examined.

3.3 CASAS Database and Data Representation

The data gathered from the sensor platforms discussed in Section 3.1 and passed through the CLM introduced in Section 3.2 come primarily in a standard format. The fields in the XML schema provided are shown in Table 3.1.

All events passed by the CTP from the sensor platform are stored in a database. These data are retained for future data mining and history building tools, as well as

Table 3.2: Schema for the data_source table to store information about the active sensors within a CTP deployment.

Column	Type	Modifiers	Notes
dsid	integer	primary key	Auto-generated key field
serial	text		Serial number of sensor
location	text		Assigned location of sensor
type	text		Type of sensor

validation of continued operation of a testbed. While the current implementation is an SQL database, any kind of structured repository would do.

For the work done in this thesis, the database has only two pertinent tables. These tables reflect the format of all events in the standard format laid out for the XML messages as discussed in Section 3.2. The schema of the tables is shown in Tables 3.2 and 3.3.

These simple tables contain all of the various types of sensor information and system status messages derived from the CLM. The design choice to use an SQL system and text fields to contain the message from the sensor provides flexibility for discrete events, but limits the ability for this system to hold large data blobs, such as those from video or audio recorders. The CASAS testbeds do not record video or

Table 3.3: Schema for the event table, which stores every discrete event reported by the sensor platform.

Column	Type	Modifiers	Notes
eventid	integer	primary key	Auto-generated key field
message	text		Message (value) of event from sensor
dsid	integer		Foreign key to data_source dsid field of sensor that reported this given event
stamp	timestamp		Full ISO 8601 timestamp of when event was stamped by the Manager agent
by	text		Name of agent within the CLM that sent the event
category	text		Type of event {entity, state, control or system} [Kusznir, 2009]

audio as smart home data sources, so this limitation is acceptable for the needs of this work.

There are more tables in the complete database schema. These relate to optimization, book keeping and other storage for later processed information. They are not discussed in the interest of brevity, as they have no bearing on the tools introduced here.

3.4 CASAS Testbeds

The CTP has been deployed at six testbeds to date. Each testbed was given a unique code name. The current list of deployments is summarized in Table 3.4.

Each of these facilities has had the CTP infrastructure installed and operated for at least several months. The designs of the different spaces have incrementally improved ADL detection and resident activity prompting to aid older adults with dementia issues. Subsets of the data gathered from these spaces are available from the CASAS shared data set web site [CASAS, 2010].

For this work, data from the *Tokyo* and *Kyoto* testbeds were extracted, annotated and used. To ensure clarity of implementation, a description of these sites is included. The nature of the space, residents and sensors is important for evaluation of the algorithms proposed for both tracking and identification.

Table 3.4: Summary of CTP testbeds deployed to date, in order of deployment.

Site Name	*Brief Description*
Tokyo	University campus lab for testing and office space example. (Still running.)
Kyoto	Two resident university housing apartment and gerontology research facility. (Still running.)
Tulum	University family apartment with full time residents.
Cairo	Two resident (plus cat) older adult private home.
Milan	Single resident (plus dog) older adult private home with prompting systems.
Aruba	Single resident older adult private home with prompting systems.

3.4.1 *Tokyo Testbed Description*

The *Tokyo* testbed is the primary testing facility for the CASAS researchers. It is an office style workplace in one of the EECS buildings at WSU. The space includes a variety of different sub-spaces and is utilized by a number of students. For this work, the data chosen from *Tokyo* are either collected specifically because they derive from known individuals for identification algorithm testing, or are annotated with the number of current occupants for training and evaluation of the tracking algorithms.

Tokyo Testbed Layout

A map of the *Tokyo* testbed is shown in Figure 3.13. The two rooms together comprise a total area of 9.7m x 12.2m. The ceiling is a uniform 3m high, and made of a dropped tile T-bar system as shown in Figure 3.14. Throughout the space, sensors are either attached to the metal T-bar surface between the tiles or affixed to the walls.

The inner room in the lower left is an office, storage space and workbenches for engineering projects. Though it is used for a number of different activities, this space is rarely inhabited by more than two people. A picture of the workbenches is shown in Figure 3.15.

The main space of the *Tokyo* testbed is trisected. The largest segment features tall cubicle walls with desks and work spaces as shown in Figure 3.16. On the opposite side of the center space has a conference table (Figure 3.17) and sitting space

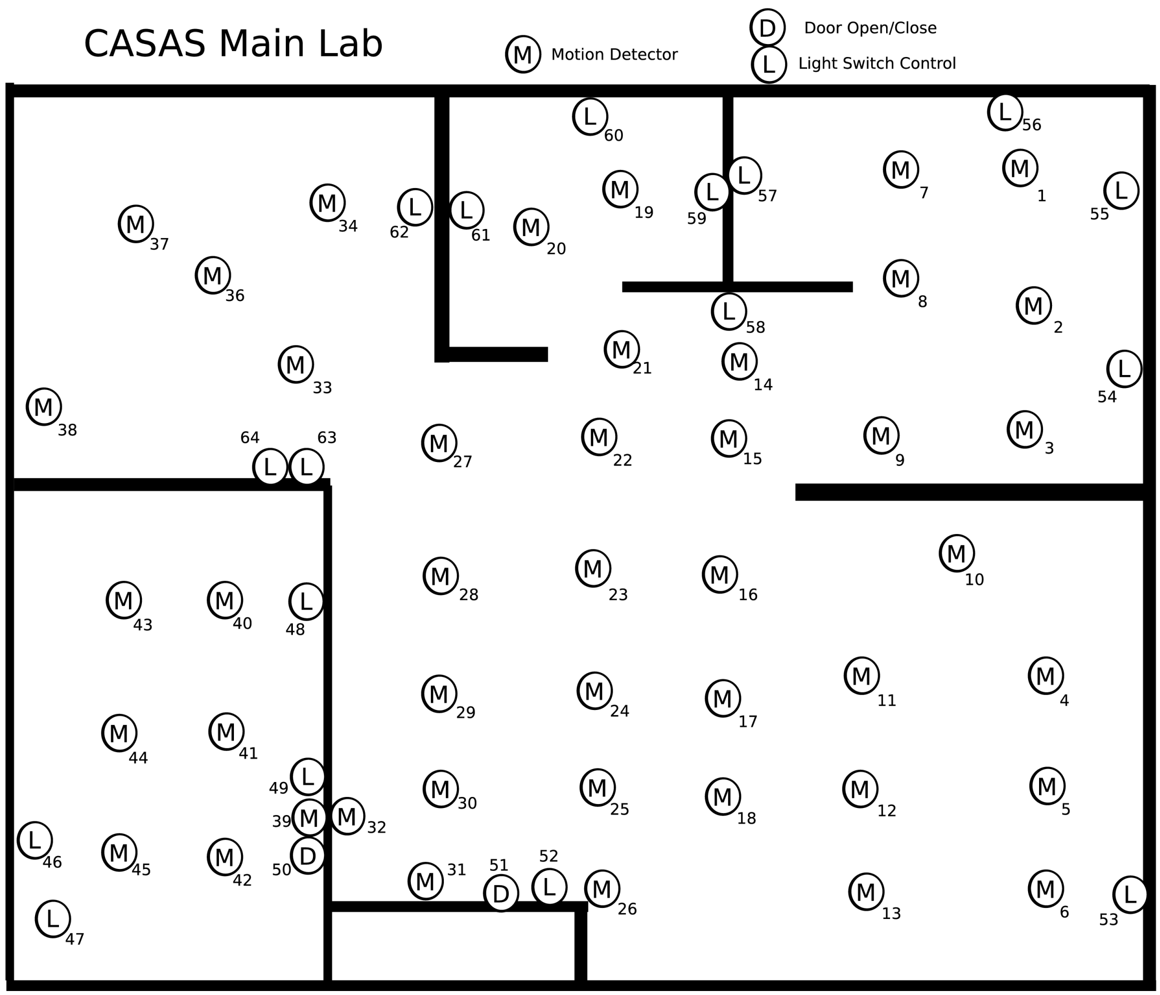


Figure 3.13: Layout of the *Tokyo* testbed, including sensor placement.

Figure 3.14: Example of the *Tokyo* testbed ceiling and sensor installation.

Figure 3.15: The inner room of the *Tokyo* smart home testbed.

Figure 3.16: The *Tokyo* testbed student cubicles.

(Figure 3.18).

To aid in activity tracking and to improve control possibilities, the *Tokyo* testbed is equipped with track lighting instead of the normal fluorescent room lighting. The track lighting is aligned with the shape of the various rooms and cubicles such that every region has an independent lighting system. Each of these tracks has a switch to actuate it, labeled "L" in Figure 3.13. Examples of the track lighting can be seen

Figure 3.17: The *Tokyo* conference table.

in Figure 3.16, showing how the tracks are orientated to the cubicles. The switches to control these lights are all Insteon™ brand devices, discussed in Section 3.1.3.

In addition, each desk has a desk lamp controlled by a local switch. This provides individuals the ability to control their personal space's illumination to their own needs. All of the lights in the space are variable and can be dimmed to suit individual tastes.

Figure 3.18: The *Tokyo* sitting space.

Tokyo Sensors

In addition to the light switches recording data from the room's lighting, as discussed in Section 3.4.1, there are other specialized sensors throughout the space. The most prominent are the standard CTP downward facing motion detectors as shown in detail in Section 3.1.2. These sensors are placed on the ceiling of the *Tokyo* testbed roughly every 1.2m, as seen in Figure 3.14. The objective of this sensor placement is to provide an unoccluded view of the residents with enough resolution to capture their current location and number.

Except in the large open area in the middle of the space, the motion detectors are installed to conform to the shape of the cubicle desk placement and conference table. In Figure 3.13, motion detectors 4–6, 10–13, 19–20 and 33–37 show this non-uniform pattern. Additionally, every door in the space has a detector placed very near the wall, directly centered above the door. This placement limits the view of the sensors so they only generate events when people are passing through the doors and not just standing near them.

After significant experience working with and annotating the resulting data from the CASAS testbeds, it has been often remarked that good sensor placement around doorways is key in understanding the movement of residents throughout the space. Placing motion detectors carefully so that they can only view one side of a doorway gives strong indicators of a resident's current position when around the portal.

Other sensors within the space include ambient temperature sensors from Section 3.1.4 and door open/close sensors from Section 3.1.5. The temperature sensors are placed to detect three stages of air movement within the rooms. One sensor is placed near the air diffuser to measure the incoming air temperature. Another is placed in the middle of the room to gather the ambient temperature, and a third at the outflow vent. With this system in place, a history of the heating and cooling within the room is captured.

Door sensors are installed on each door in the testbed. The sensor on the door to the hallway is shown in Figure 3.13 as number 51, while the inner room sensor is number 50. In the *Tokyo* testbed the inner room door is almost always open, and the door to the outside is propped open when students are present. Consequently, they provide limited information about the current state of residents' coming and going.

Tokyo Residents

The behavior of the *Tokyo* residents is varied and sporadic. They are primarily adults, either graduate students, postdoctoral residents or undergraduate student workers. Some use their desks consistently, while others only intermittently at best. The duration of the gathered data set spans multiple years so some of the students arrive and eventually graduate entirely within the data's time span. In addition to the normal residents, the space hosts events such as lab tours, meetings, and guests. These large influxes of unknown people make for very noisy data periods. Altogether, the

Tokyo testbed residents pose a very complex environment for tracking, identification, and ADL and anomaly detection.

Tokyo Summary

The CASAS *Tokyo* smart home testbed is the longest running CASAS CTP installation. It has evolved in small ways from its initial installation into a consistent and functional smart home implementation. This kind of testbed provides quality data in an office-like environment with a variable number and type of residents. While it is not as interesting for ADL research as the other CASAS sites, it has proven very useful for tracking, identification and preference building projects.

3.4.2 Kyoto Testbed Description

The *Kyoto* testbed is the primary research facility for the CASAS projects. This three bedroom apartment shown in Figure 3.19, as part of the WSU University Housing system, is ordinarily the home of two undergraduate students. *Kyoto* is designed to be a sensor-rich space designed for capturing as many ADLs and behaviors as possible.

Since its initial installation in 2007 this smart home testbed has undergone a series of improvements. These have primarily been software updates, but over time new sensors and interactive technologies have been deployed. These have focused on

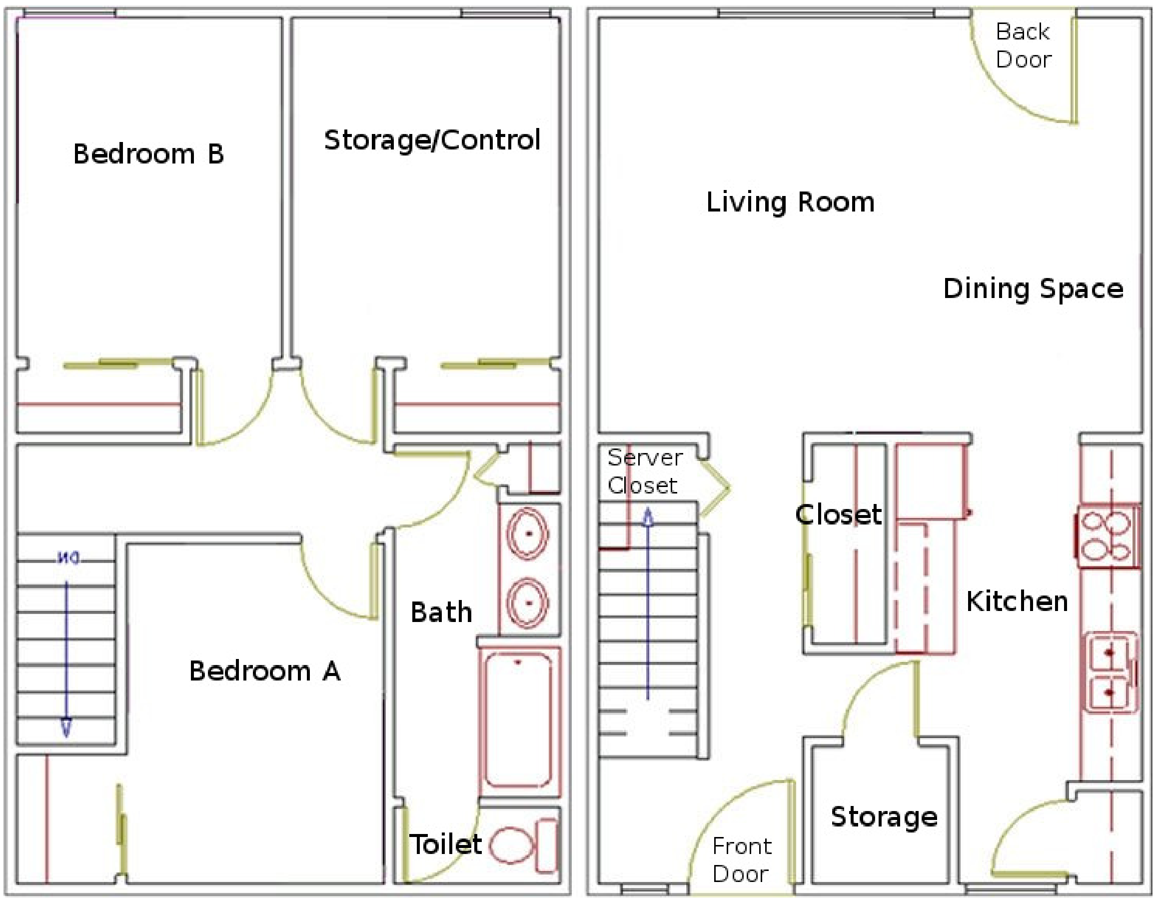


Figure 3.19: Labeled room map of the CASAS *Kyoto* testbed.

supporting the CASAS research objectives, such as early onset dementia evaluation and aging in place tools, although *Kyoto* is also used for studies regarding the associating of activities with energy consumption [Chen et al., 2010, Chen and Cook, 2010]. This testbed has proven highly successful [Singla et al., 2010] at gathering rich and well-documented data sets, some of which are available publicly [CASAS, 2010].

Kyoto Testbed Layout

The *Kyoto* testbed, also known as the "smart apartment" in many CASAS works, is representative of many American living spaces. Each resident has their own room with a bed, desk and closet. There is a shared bathroom, living room and kitchen. This resemblance to many typical homes makes the results from the research done here more applicable than partial smart home implementations or work done with specialized facilities.

The basic *Kyoto* sensor layout, shown in Figure 3.20, follows after the design of the *Tokyo* testbed. The primary sensor type is the downward facing PIR Motion Sensor (Section 3.1.2). These are installed on the 2.4m high ceilings with a field of view that covers roughly a 1.2m x 1.2m section of the floor. Similarly to *Tokyo*, they are installed roughly every 1.2m to provide continuous coverage of the space. This sensor distribution is designed to provide enough resolution for human annotators and algorithms to localize and track the residents.

The rest of the sensors are installed on an "as needed" basis. Many of them

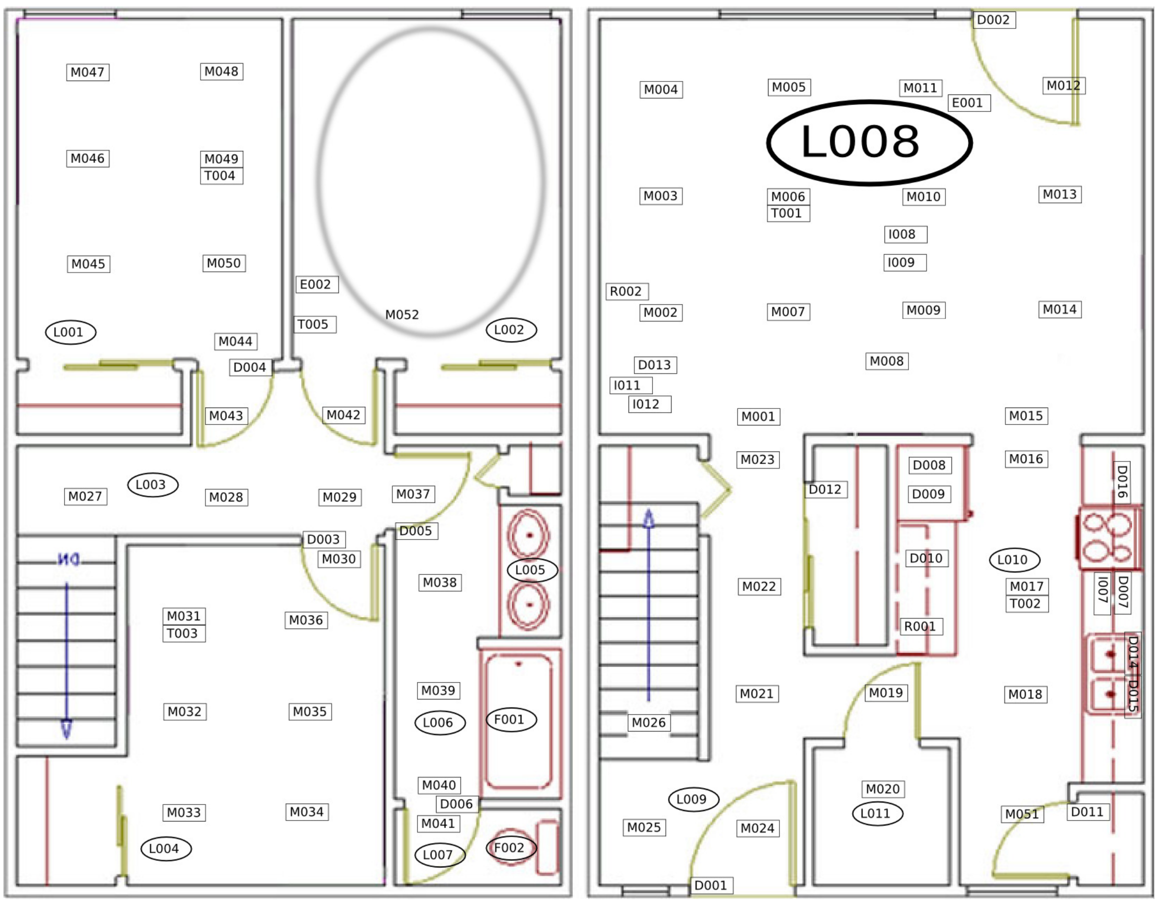


Figure 3.20: Sensor layout for the CASAS *Kyoto* testbed.

have very specific uses to aid the artificial intelligence algorithms in their operation or to give human annotators information about the activities being performed.

Kyoto Sensors

The *Kyoto* motion detectors are installed throughout the apartment, barring the Storage/Control room upstairs. This room is kept nearly sensor-free so that controlled studies can be performed without the controllers introducing more events than need be into the resulting data set. There is a single area motion sensor installed in the room to give annotators a sense of occupancy, but no more.

The light switches throughout the residence have been replaced with Insteon brand switches (Section 3.1.3). These generate events that correspond with light control decisions and preferences of the occupants. They also give the system the ability to automate lighting for power conservation and predicting resident preferences.

Nearly every door in the apartment has a magnetic closure sensor (Section 3.1.5). Additionally, some of the cabinets in the kitchen, the microwave and the refrigerator are monitored in this manner. The kitchen sink has water flow sensors (Section 3.1.8) to monitor water usage. This extra information about door and kitchen behavior is informative regarding which activities are being performed in the room.

The right front burner on the range is metered for power consumption using the Lentil Board A2D sensor from Section 3.1.7. This power monitoring is interpreted by many of the ADL algorithms applied to the data sets to detect cooking, as well as what kinds of cooking are being performed.

There are a number of item presence sensors installed in the kitchen and living room (Section 3.1.6). These contact-switch based plates are primarily used for controlled activity studies. When key items, such as medicine bottles, television remotes and cooking utensils are removed and returned, the system yields additional context for the current activities of the occupant.

In this space the whole home instantaneous wattage is metered. By placing an inductive coil around the main breaker, the OneMeter power meter (Section 3.1.9) monitors and reports the wattage used by the home. Even something as small as opening the refrigerator door creates a noticeable wattage change. In this example the change caused by the incandescent light in the refrigerator space turning on, but many other activities use devices that change the power footprint of the home. Long term power fluctuations on a daily, weekly and annual basis provide indications of the behaviors of the residents and surrounding environment.

Examples of the *Kyoto* sensor installations are shown in Figure 3.21 and Figure 3.22. Because this space does not have a dropped T-bar ceiling like *Tokyo*, the wires become very visible. The later CASAS installations in private homes use wire colored to match the surface and are cut to stretch between the sensors with very little slack. These small details make the resulting sensor installation much more attractive to the residents.

Figure 3.21: Sensors installed in the foyer of the *Kyoto* testbed.

Figure 3.22: Sensors installed on the living room ceiling of the *Kyoto* testbed.

All of the sensors are wired back to a central location. This location is a closet under the stairs as shown in Figure 3.23. To aid in the mounting of the server, power systems and other communications devices a plywood board was affixed to the wall. The various devices and systems are readily attached to this solid mounting base.

Kyoto Residents

The residents monitored in the *Kyoto* testbed fall into two categories: full-time and transient. The full-time residents are students who have volunteered to live in the testbed. They live, work and study there as university students. The facility is designated as undergraduates-only by the WSU Housing Department, so all of the full-time residents have been undergraduates to date.

The full-time residents have changed as the years progressed. Each academic year and for each summer, a new pair of residents is chosen to live in the smart apartment. This turnover of residents provides a wider variety of behaviors that can be monitored and analyzed using a single sensor installation. Both women and men have resided there in turn. For the publicly available data sets and research papers published using the *Kyoto* smart apartment data, the genders of the residents are normally listed.

The transient residents represent a number of groups. An element they all have in common is that they only enter during the daytime, and rarely stay more than four hours in a session. These people might be maintaining the facility, controlling the

Figure 3.23: Central wiring board and server location for the *Kyoto* testbed.

experiments, participating in the experiments, be guests of the residents, or Housing Department maintenance workers. Depending upon the nature of the data set and research, these transient residents are either filtered out or kept as required.

Kyoto Summary

The CASAS *Kyoto* testbed, also known as the "smart apartment," is a long running and highly valuable source of smart home research. It has the benefits of the CTP platform for non-invasive privacy-preserving sensor systems plus long running full-time residents. Using the comprehensive nature of the CTP to monitor numerous activities in the space has supported the ADL, identification and energy research done by the CASAS group to date. The ability to do both controlled aging in place studies while also deriving "real-world" data sets has proved to be invaluable for the smart home community at large.

3.4.3 Other CASAS Sites Overview

The other four CASAS testbeds to date did not contribute data to this work, so they are only summarized here. They all share the same CTP infrastructure, but represent a range of spaces, residents and design objectives. In the next subsections, an overview of each testbed is given in order of their construction.

Tulum Overview

The *Tulum* testbed is another WSU Housing facility, similar to *Kyoto*. It is a two bedroom apartment that houses graduate student families. This facility was designed to be a space comparable to *Kyoto* and supported a number of works on ADL transfer learning [Rashidi and Cook, 2009a, Cook, 2010, Rashidi and Cook, 2010a,b]. It has many of the same features, but a different layout, as shown in Figure 3.24(a). The sensors were limited to area, downward facing motion detectors, temperature and power metering.

Two adult couples are represented in the data sets gathered from this facility. Each pair lived in the smart home for over six months and their activities were annotated. Neither couple had any pets, and all were healthy young adults. After a year and a half of operation, the facility was terminated.

Cairo Overview

Cairo was the first private home installation performed with the CTP. It consisted of a split level home, with two older adults and their pet house cat. The site had only temperature, area, and downward facing motion sensors, placed as shown in Figure 3.24(b).

The objective of *Cairo* was to begin gathering full time real-world data on older adults in their personal environment. The husband of the couple had early onset dementia, while the wife was measured as a healthy older adult. As the study progressed, they were consulted about their activities and impressions of the smart home around them. After roughly eight weeks of data gathering, the installation was shut down and removed.

Milan Overview

Following the successful *Cairo* testbed, another home with an impaired older adult was sought out. The facility shown in Figure 3.24(b) is a two bedroom, single floor condominium. The only residents were an older woman with mild dementia and her pet dog.

With *Cairo*, an interactive console was installed in addition to the door, temperature, area and downward facing motion detectors. This console was built to interpret the current activities in the space and provide reminders for the resident on a handful of activities. After several weeks of operation without the console, it was enabled to prompt the resident to perform activities. She would then be prompted to give feedback about whether she performed the activity, intended to do it later or not at all. The objective of this new system was to begin learning about how smart homes can help people with dementia issues through reminders and interaction [Cook, 2010]. After about four weeks without the prompting and four weeks with the prompting, the testbed was shut down and removed.

Aruba Overview

Aruba is the latest CASAS in-home testbed to date. It is a single floor family residence with a single older adult normally present. The site has door, temperature, area and downward facing motion detectors, as shown in Figure 3.24(d).

The resident has somewhat severe sleep apnea, so the sensor layout is focused on detecting sleep on the couch in the living room and the chairs in the office in addition to ADL detection. There are plans to ultimately prompt for some select activities, and possibly provide a wake up call if the resident sleeps in an inappropriate manner, but these have not been implemented to date. *Aruba* is still in operation, with a planned ending date in early 2011.

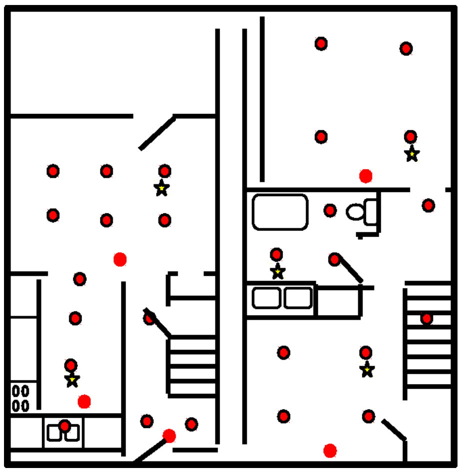
3.5 CASAS Environment Summary

The CASAS Environment is comprised of several elements. The first element is the CASAS Technology Platform, which provides a number of modular tools for sensing, communicating between agents, storing data and controlling smart home spaces. The CTP is designed to conform to the space where it is installed, as different residences and residents require unique suites of sensors and devices. Additionally, the open nature of the middleware for communication lends itself to a very wide variety of sensor types. This leads the CTP to being quickly adaptable if new devices need to be integrated for specialized needs.

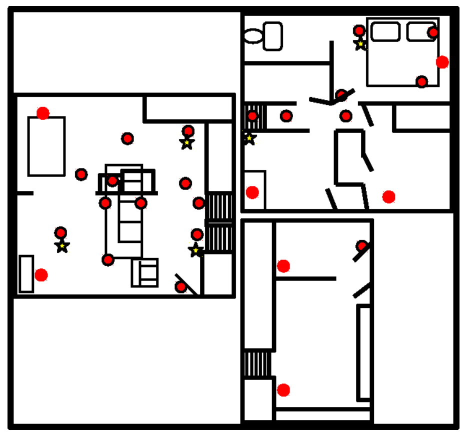
The second element is a variety of testbeds for generating high quality data sets. As the CASAS research group continues to delve into the issues surrounding smart home implementations, it has become clear that creating a diverse set of environments to gather data has proven invaluable. Few smart home groups enjoy such a wide array of data sources for analysis.

The final element is the residents. By working through the university and among members of the community, the CASAS testbeds have derived data from a wide variety of people. Varying in age, gender, ability and behavior, these residents exemplify the need for a smart home to be broadly adaptable.

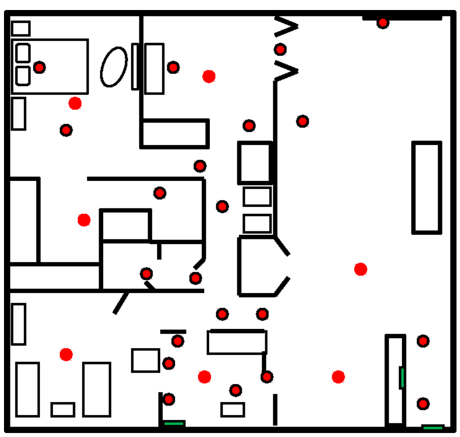
The CASAS Environment is a high quality, long running research platform. Its resulting publications and applications attest to the applicability of this system to ongoing research and ultimate real world commercialization.



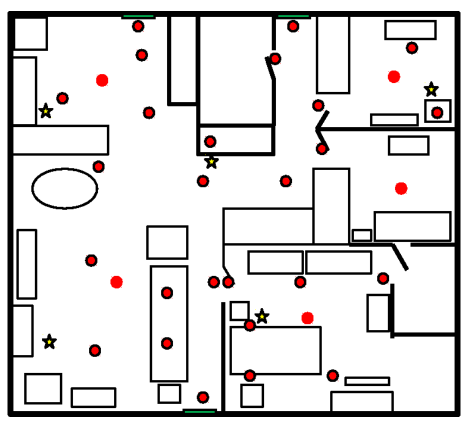
(a) Tulum



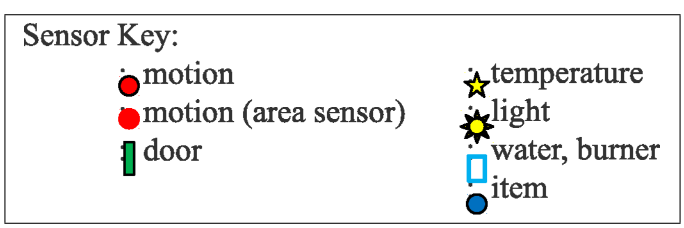
(b) Cairo



(c) Milan



(d) Aruba



(e) Layout Sensors Key

Figure 3.24: Other CASAS testbed sites used for smart home evaluation.

CHAPTER 4. RESIDENT TRACKING APPROACHES

The first hypothesis stated in Section 1.5 noted that there is a need for localization and tracking in the smart home context. The tracking tools introduced here are designed to split the single stream of events produced by the sensor platform into several sub-streams, thus demonstrating support for our hypothesis. They exploit physical locality, timing and historical data to determine the number and location of residents.

Each of the sub-streams created by this attribution of events represents a single source of events, such as an individual human or a pet. In many ways this process is similar in behavior to the algorithms used for identification in Chapter 5. The difference here is that instead of labeling the event with a resident's name, it attributes the stream of events to an anonymous entity. After the events are divided into groups representing individuals, other algorithms can then use the separated streams for other objectives.

4.1 Tracking Introduction

Using the CTP and data from two of the CASAS research testbeds, several algorithms have been designed and implemented to localize and track residents. The algorithms leverage the physical layout of the sensors to build rule-based or probabilistic models to determine the current locations of the residents. Given those locations, each currently received event can be attributed to one of the people in the space, thereby dividing the stream of events among the current inhabitants.

The tools introduced for tracking in this thesis are all forms of proximity localization as introduced in Section 2.1.3. They use the proximal locations of sensors to derive the location of residents. Utilizing this kind of system, residents need not carry a tracking device, and the sensors themselves can be very simple. These are both key goals of the CASAS smart home architecture described in Section 1.6.

When approaching tracking in the smart home context, it was determined that some terminology had to be defined. The researchers use the term "entity" within the models to represent an individual. This is because not every entity in the model represents a person. They are most often humans, but the studies have included smart home installations with cats, dogs and even robots that can trigger sensor events. Using the term entity allows for a wider appreciation of how complex living spaces can be.

The three algorithms introduced in this thesis are similar in many ways, but they do represent several different sources of decision making to allow for a contrast of approaches. The first algorithm is a rule-based tool. It uses a set of simple rules combined with a graph of all possible routes between sensor locations to track individuals. This tool is dubbed "GR/ED," which stands for Graph and Rule based Entity Detector. The initial results for the GR/ED were promising, but the tool began to perform poorly in more complex social situations, as well as in imperfect sensor network environments. The GR/ED is introduced and explored in more depth in Section 4.3.1.

As a means to exploit available historical data to create a better algorithm, a second strategy based on Bayesian Updating was evaluated. Utilizing a corpus of training data annotated with the number of residents, a probabilistic transition matrix is built and applied to update the world model. This tool is dubbed the "BUG/ED," which stands for the Bayesian Updating Graph based Entity Detector. By leveraging a probabilistic model, the system is able to handle more issues within the sensor network and perform somewhat better than the GR/ED in the face of more complex resident behaviors. The BUG/ED is discussed in more detail in Section 4.3.2.

The last algorithm is dubbed the "PF/ED," which stands for the Particle Filter based Entity Detector. This tool draws upon the established field of using Particle Filters to track objects in robotics, as shown in Section 2.1.3. The PF/ED was created to be a Monte Carlo algorithm in contrast to the BUG/ED's Bayesian approach. These two tools were tested on a data set specifically built to test the tracking of residents, and not just the ability to judge occupancy. The PF/ED is examined in more detail in Section 4.3.3.

Each of the algorithms and their method of evaluation is introduced in the next sections. They were tested with data sets from the *Tokyo* and *Kyoto* CASAS testbeds. The *KyotoOccu* and *TokyoOccu* occupancy data sets are used to test the ability of an algorithm to determine the current number of residents in the space. Some applications of smart environment technologies may only require the current occupancy of the space to tailor environmental systems, such as lighting and heating [Fountain et al., 1994].

In the *Tracking* data set drawn from the *Kyoto* testbed, there are always one or two residents and the accuracy for an algorithm is based around the ability for a tool to determine the actual path of a resident. These detailed paths are useful for improving individualized preference building [Rashidi and Cook, 2008], anomaly detection [Jakkula and Cook, 2008, Jakkula et al., 2009] and uses of the smart environment.

As a final demonstration of the benefits of applying a tracking tool to separate the event stream, the BUG/ED tool is used to boost the ADL detection capabilities of a naïve Bayes classifier with a ADL-annotated data set from the *Kyoto* testbed. This test is discussed in Section 4.5.

4.2 Tracking Research Layout

The algorithms built and tested for the tracking of residents draw upon data from two of the CASAS testbeds. These data are from the *Tokyo* and *Kyoto* sites. These two sites provided a large corpus of available data sets from numerous residents coming and going at all hours.

4.2.1 Research Design

This work uses an observational study method tor evaluating the first hypothesis introduced in Section 1.5. The hypothesis states that using data from the CTP while exploiting physical and temporal information contained in the data, individuals may be tracked through the smart home space.

To gather data for testing, several CTP testbeds were installed and operated over a number of months. The data gathered were annotated by humans with the current number of residents in the space, or their identities and complete paths. The case series generated during these data gathering periods provides a suite of data for testing of the algorithms. The various algorithms were then tested against the different data sets to determine their ability to properly count residents in the cases of the *TokyoOccu* and *KyotoOccu* occupancy data sets and to track residents in the case of the *Tracking* data set. The residents were not intervened with while they lived in the smart home spaces and no attempts were made to adjust their behavior over time.

4.2.2 Occupancy Data Sets

To test the algorithms' ability to determine the number of residents, two corpora of data were created. A subset of the stored events for both the *Tokyo* and *Kyoto* testbeds, as described in Sections 3.4.1 and 3.4.2, was taken and annotated by humans as summarized in Table 4.1. The human annotators were taught to observe the events as they were replayed using a visualization tool [Thomas and Crandall, 2011] and to log the current number of residents in the testbed. This value representing the current occupancy could then be used to determine how accurate the tracking tools were at judging occupancy, and in the case of the BUG/ED it was also used to train the transition probability matrix.

The data gathered by the CTP for the occupancy data sets is represented by the following features:

1. Date
2. Time

Table 4.1: Summary of data sets used for validation of occupancy discovery.

Data Set	*Residents*	*Length*	*Num Events*
TokyoOccu	0..9	59 days	209,966
KyotoOccu	0..6	21 days	231,044

3. Sensor location[1]

4. Event message

5. Annotated count of residents

The first four fields are generated automatically by the CASAS middleware at the time of the event's creation. The annotated count field is the number of people in the testbed at the time the event occurred. Sample data collected by the CTP and annotated for the Occupancy data sets is shown in Table 4.2.

The *TokyoOccu* data set represents sensor events that were generated while faculty, students, and staff performed daily working routines in the lab over a course of 59 days. To train the algorithm, the data was manually inspected by a human

[1]Some older CASAS data sets use the sensor serial number instead of a device-independent location value.

annotator and every event was annotated with the current number of residents in the space. In total this made for 209,966 motion sensor events. The number of residents ranged from zero to more than nine.

Once the testbed had more than six to seven people in it, the annotators noted that there was little available information to identify what was happening in the space. This was anecdotal evidence about the limited resolution of the testbed. Adding more sensors should increase this maximum detectable occupancy, though this approach has not been evaluated to date.

The *KyotoOccu* data was was taken from 21 days of the *Kyoto* testbed. This made for 231,044 motion sensor events. Again, the sample data was inspected by a person and annotated with the number of people currently in the space. In this set, the number of residents ranged from zero to five and the annotators noted a marked decrease in their ability to interpret individuals' movements as the occupancy reached about four residents.

4.2.3 Tracking *Data Set*

The data set built to evaluate the ability for an algorithm to track the residents was drawn from the *Kyoto* testbed, as described in Section 3.4.2. The data chosen were times when there was only one or two residents present and every individual

Table 4.2: Subset of data used for occupancy algorithm testing.

Date	*Time*	*Location*	*Message*	*Entity Count*
2007-12-21	14:44:41.0764	L017	ON	2
2007-12-21	14:44:36.8230	L017	OFF	2
2007-12-24	14:44:50.2819	L007	ON	3
2007-12-24	14:44:52.6889	L007	OFF	2

event was attributed to a unique, but anonymous individual. In this way the actual path of a resident may be traced through the space and compared to the output of an algorithm attempting to track entities based only upon the sensor events. This *Tracking* data set was again created by a human annotator inspecting the events as shown in Table 4.3.

The data for this set from the CTP was augmented by a resident identifier as follows:

1. Date
2. Time
3. Sensor location
4. Event message

Table 4.3: Summary of data sets used for validation of tracking algorithms.

Tracking data set	
Length	96 hours
Num Events	20,519
Single Resident Events	8,581
Multiple Resident Events	11,938

5. Resident count

6. Anonymous resident identifier

The first four fields are generated automatically by the CASAS middleware at the time of the event's creation. The annotated fields are the number of people in the testbed at the time the event occurred and the identity of the resident who caused the event. Sample data as collected by the CTP is shown in Table 4.4.

In this sample data, the resident "A" moves from locations M009 to M016 to M015 while resident "B" moves from M023 to M001. The goal of a tracking algorithm is to determine this solely from the first four fields of the data set.

The "OFF" events in the *Tracking* data set are not attributed to residents. This is a subtle difference in the nature of "ON" verses "OFF" events. With an ON, it is known that an entity caused the event by moving in the view of the motion detector.

Table 4.4: Subset of data used for tracking algorithm testing.

Date	*Time*	*Location*	*Message*	*Entity Count*	*ResidentID*
2009-06-08	13:48:49.033446	M009	ON	2	A
2009-06-08	13:48:50.002373	M023	ON	2	B
2009-06-08	13:48:51.042953	M009	OFF	2	
2009-06-08	13:48:52.024474	M001	ON	2	B
2009-06-08	13:48:54.048006	M016	ON	2	A
2009-06-08	13:48:55.015058	M015	ON	2	A
2009-06-08	13:48:55.056504	M008	ON	2	B

Comparatively, an OFF indicates an extended absence of any motion. This means that there is no entity available to cause the OFF. These events are kept because the change in the sensor state is important to understanding what is happening in the space and is used by the GR/ED and PF/ED algorithms in maintaining their internal models.

4.2.4 Assumptions of the Study

The algorithms used to provide tracking and occupancy of residents involve several notable assumptions. The first is that the testbeds used are good exemplars for future smart home implementations. These systems will have a variety of sensor sources, layouts and uses which the CASAS approaches will mirror in some ways, but not others. Given the lack of normalization across smart home systems and testbeds throughout the world, this is a common issue that all research in this field.

The second assumption is that the participants' behavior was not severely impacted by awareness of their residence in a smart home. For the purposes of tracking individuals, this is not likely to be an issue. Residents should be able to act freely and still be tracked by these kinds of tools.

All of these tools assume that there is a large number of sensors throughout the smart home space. Since the tools rely on evidence of multiple residents from events received from physically separated spaces, there must be sufficient sensors available to make that separation possible. If the density, and by extension, the resolution of the sensors drops, their accuracy will drop accordingly.

4.2.5 *Limitations of the Study*

The primary limitation of this study is that the tools provide only an approximation of what is happening in the space. Compared to a system with near-perfect accuracy provided by wireless tracking devices or numerous cameras, these probabilistic models may not be sufficient for some uses.

This study does not evaluate these tools in smart homes with a dearth of sensors. In a room with only a single motion detector, the only evidence the system would provide would be the fact of occupancy when the single sensor fires. Testing the approaches introduced in this work requires a number of sensors viewing different parts of the space to track residents as they move about. It is this division of the space that provides information about where activity is taking place, as well as whether there are multiple residents moving about. An open question in the smart home community deals with required sensor density for various smart home goals. This work on tracking does not address this open question, though its implications are felt when attempting to handle larger numbers of residents within a single smart environment.

4.2.6 Research Design Summary

To evaluate our hypothesis that entities can be counted and tracked in a smart home context with only passive, low-profile sensors, this study melds a number of projects. The first is the installation of the CASAS Technology Platform into a number of testbeds for the purposes of gathering data sets. The second is the annotation of that data with the current occupancy and paths of residents within the smart home space. The last is the implementation and evaluation of algorithms designed to interpret the data to determine the number and location(s) of the residents throughout the space. Taken together, this work builds a privacy-protecting probabilistic smart home system that provides tracking capabilities to support numerous smart home goals in a multi-resident environment.

The next sections introduce the tracking algorithms in detail. The results, evaluation, and comparison of each tool are discussed in Section 4.4.

4.3 Occupancy and Tracking Algorithms

To test the hypothesis that locality and temporal features can be used to count and track individuals in a smart home space, three algorithms were developed. The algorithms are:

1. GR/ED: A graph and rule-based tool
2. BUG/ED: A Bayesian-based tool
3. PF/ED: A sequential Monte Carlo method (Particle Filter)-based tool

These tools each have a set of prerequisites and benefits when used to count or track the current residents in a smart home space. In sum, their needs and implementation details are discussed in sections 4.3.1, 4.3.2 and 4.3.3. The resulting classification accuracies on the three data sets, the algorithms' behaviors and further discussion are available in Section 5.4.

4.3.1 Graph and Rule-based Entity Detector: GR/ED

The GR/ED algorithm was designed to use the order of events to incrementally track individuals in the CASAS testbed. The core idea is that entities will most likely trip sensors as they cross from one place to another, and multiple entities will often have one or more sensors between them as they go about their day.

The "graph" part of the tool represents the physical locations of the sensors within the testbed. The two graphs derived from the CASAS testbeds used in this work are shown in Figures 4.1 and 4.2[2]. These graphs are made up of only the

[2]The edge cutting across the Kyoto graph from M026 to M027 is connecting the sensor

downward facing PIR motion detectors, which are laid out to cover most of the floor space. A graph that represents a given space has vertexes representing the sensors themselves and edges that represent the possible paths between those sensor locations. Since the sensors are placed to fully cover the space, people moving about often generate an obvious and complete chain of events from one place to another.

The rule-based part of GR/ED is a simple set of logical rules for creating, destroying and moving entities within the model. These are all triggered by sensor events or a lack of events over a period of sufficient length.

The first rule is for building a new entity in the model. With this rule, if an "ON" event occurs at a location with no adjacent entities, a new entity is created. This theoretically means that this event was caused by a heretofore unseen entity. It could have either just entered the space, or have been shadowing another one of the residents and only just then been separated enough to have been visible as a separate entity.

The second rule is for destroying entities. An entity is destroyed (removed) from the model under two circumstances. The first circumstance occurs when it has been determined that the entity has left the monitored sensor space. In the case of a CASAS system, this occurs whenever an entity moves onto the sensor most adjacent

---

at the bottom of the stairs with the one at the top leading to the second story of the apartment.

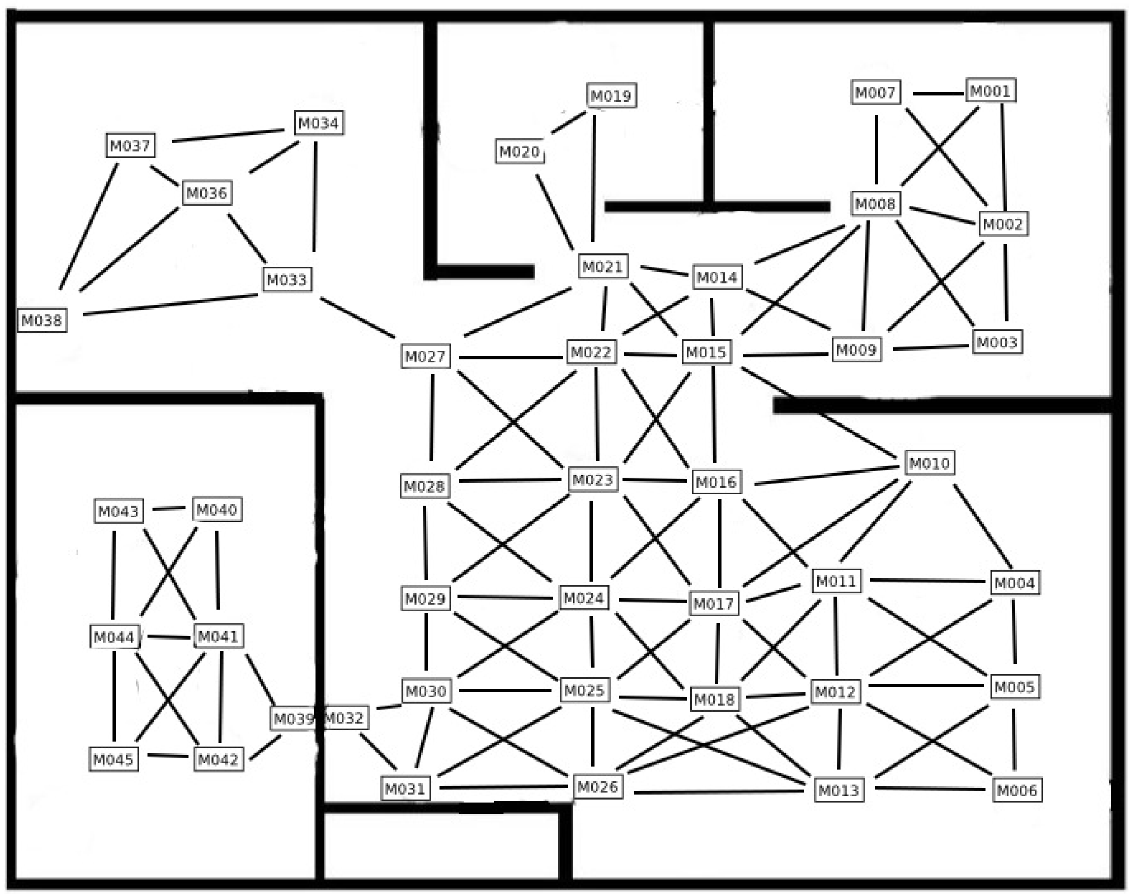


Figure 4.1: Graph of sensor locations for the *Tokyo* testbed.

to an exit. Since there are no sensors available that determine whether someone has moved through the doorways, thereby exiting the space, it has to be assumed that moving next to the door represents an exit.

The second way an entity can be destroyed is when it fails to generate any new events for an extended period of time. This is to address a limitation of the sensors used for the data gathering. Since the PIR sensors do not provide data if an entity does not move then it becomes difficult to determine if it is still in a given location, or if it has moved away without triggering events. This kind of situation can occur either by a flaw in the sensor coverage where a person moves in a path that the graph does not represent, or if two entities move to the same location, followed by moving together across the space. Since the sensors do not provide a magnitude as to the "size" of an entity, it is easy for multiple people to move as a group and leave inaccurate entities in the model that no longer exist in the real world.

To remedy this, a timeout on entities has been imposed. If an entity does not generate events for a period longer than the timeout, then they are assumed missing and removed from the model. After experimentally trying a wide range of values with the *Tokyo* data set, it was determined that a timeout between 300 to 600 seconds is optimal, and 300 seconds was used for this work.

The final rules for the GR/ED tool have to do with movement. The first rule for movement is that when an ON event occurs and an entity is at a neighbor in the

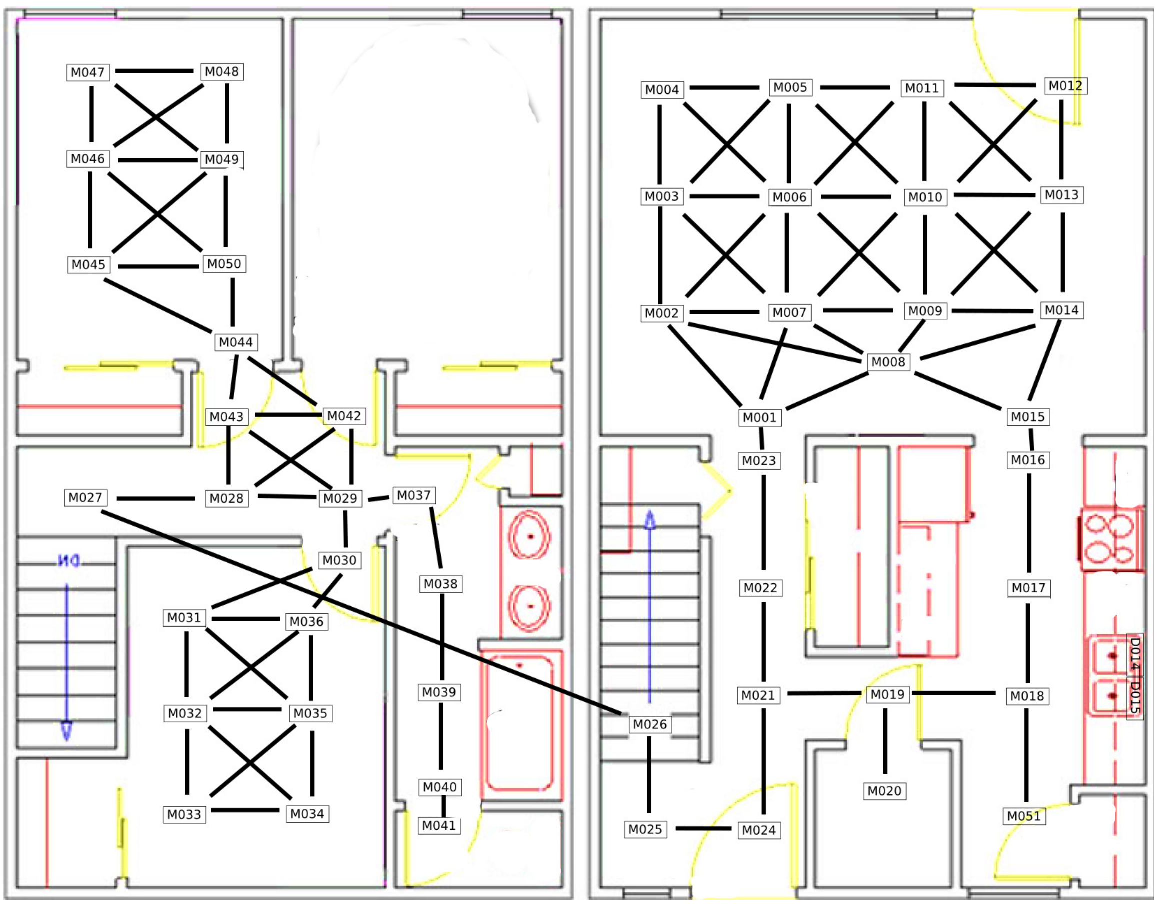


Figure 4.2: Graph of sensor locations for the *Kyoto* testbed.

graph, then that entity moves to the location that generated the event. Only one entity may have that event attributed to them, so if more than one entity is adjacent in the graph to the new event, then the event is attributed the one that moved most recently. This rule that the most recent mover has priority helps the algorithm deal with groups of entities moving in smaller areas.

As the GR/ED operated, it was noted that people could easily fool the GR/ED by walking back and forth. The PIR sensors used are from a commodity home security product line and were not designed for smart home applications. Because home security hardware operates relatively slowly, the sensors stay in the ON state for anywhere from one to five seconds before turning back off once movement stops. Due to this very long time frame of being in the ON state, people can easily walk in the pattern shown in Figure 4.3 and confuse the GR/ED algorithm. With the behavior illustrated, the algorithm would move the person's virtual entity to the node on the left, but the sensor in the middle would stay on long enough that they would then move to the sensor on the right without causing a ON event on the middle sensor. This would leave their old virtual entity on the left, and create a new one from the new event ON event from the right most sensor. At this juncture the model is out of synchronization with the ground truth and the false entity remaining on the left would have to time out before the GR/ED would be correct again. To remedy this failing, the Open List of sensors was proposed.

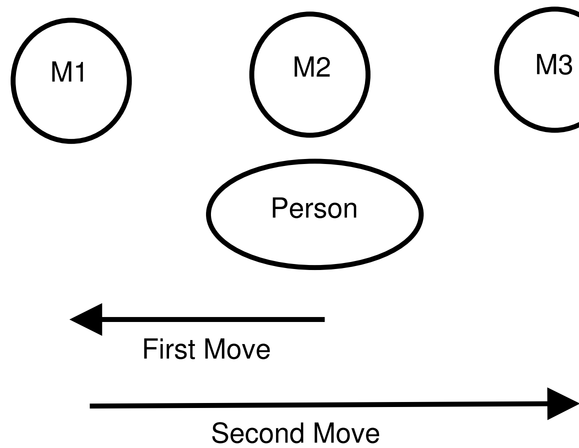


Figure 4.3: Example of a movement pattern that would break the initial GR/ED algorithm.

With the Open List, an entity has a set of locations that define their present location instead of only a single one. For every ON event sent by the sensors, there is always an OFF to match it. When an entity is attributed an ON event, that sensor location is placed in their Open List. Once that sensor finally sends an OFF event, the location is removed from their Open List. Now that this list is available, an entity's location is not merely their current vertex in the graph, but the whole of the Open List. If an ON event occurs that is adjacent to any location in this list, it will be attributed to the entity. This technique remedied most problem instances of people walking back and forth. In the previous example, the entity's Open List would be both the center and left sensors. So when they next trip the right sensor they are still considered "adjacent," due to the middle sensor being in their Open List, and would properly be attributed that new event from the right sensor.

Each entity in the model has a list of locations that it has visited in the past. The ordered list of these locations may be used to build a tracklet for the resident. Alternatively, the current number of entities in the GR/ED model is the estimated occupancy of the space.

Summary

The resulting GR/ED algorithm is efficient and operated in near real time, making it feasible for real-world smart home implementations. As an added advantage, it requires no annotated training data to operate, merely the graph of possible routes between sensor locations. This allows the GR/ED to be deployed and launched once the layout of the sensors is known, without having to wait for any form of annotated training data to be made available.

4.3.2 Bayesian Updating-based Entity Detector: BUG/ED

After reviewing various existing artificial intelligence algorithms used for similar classification problems, it was determined that a Bayesian Updating-based algorithm might be a good choice as a successor to the GR/ED tool. Bayesian Updating is a probabilistic strategy where new evidence is used to probabilistically update the world model. The Bayesian Updating Graph Entity Detector, dubbed the "BUG/ED," is proposed here. This algorithm takes the current model of the smart home space with respect to the current resident locations, and combines it with new evidence in the form of a sensor events, to build the most likely world model for the latest state.

The incremental attribution of events to the entities contained in the BUG/ED's world model represents the tracklets the residents have followed. This behavior is similar in many ways to the GR/ED, but instead of a simple unweighted, undirected graph it uses a transition matrix of probabilities of an entity moving from one location to another. The algorithm relies upon Bayes' Rule as shown in equation 4.1 to determine which entity should be moved given a sensor event. The matrix can also be augmented with other sources of evidence, though the algorithm here was only provided sensor to sensor transition likelihoods.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4.1}$$

The biggest advantage of the BUG/ED over the GR/ED is in the handling of failures in the sensor network coverage. Often a person will bypass a sensor in the graph, causing an immediate failure of the GR/ED tool. This situation causes the GR/ED to create a new entity in the model, and abandon the old one improperly. With the BUG/ED, the transition matrix will often have a likelihood of transition between those two physically distant sensors and will often properly move its entity, even if the person it represents skips sensors occasionally. This ability alone increased the robustness of the algorithm in day to day operation.

Training the BUG/ED probability matrix

With Bayesian Updating, there must be some corpus of information for the algorithm to use in estimating the conditional and joint probabilities. Obtaining or generating that corpus is up to the implementation and domain. The data annotated by humans for the *Tokyo* and *Kyoto* testbeds specifying the number of current residents was used to train the BUG/ED transition matrix. This training process is done before operation of the BUG/ED can begin and resembles the GR/ED algorithm with a very important addition.

Since the annotated data has the true count of residents, the training algorithm can make use of that key data for determining when residents entered and left the space. The training algorithm takes the events from the training data one at a time and incrementally builds a model of the residents' locations and transitions between sensor locations, much like the GR/ED. The key difference is that it uses the resident count from the training data to decide when to create, destroy or move entities.

The training algorithm also makes use of the same graph utilized by the GR/ED tool, but only for counting hop count between sensor locations. This graph has one addition for the BUG/ED algorithm, a virtual sensor location called "OUTSIDE." This OUTSIDE location represents all of the universe not monitored by the smart home sensors. It is directly connected via an edge in the graph to any sensor at an exit to the smart home, such as sensors next to the front and back doors. Entities are also moved from OUTSIDE when they are created, and to OUTSIDE when removed. The graph is used in determining which entity is closest to the OUTSIDE location, or which entity is closest to an event that just occurred.

The training algorithm will either create, destroy or move entities by looking at whether the resident count went up, down or stayed the same between events. If the count goes up, a new entity is created at that location by moving them from the virtual location OUTSIDE to the location of the event. If the count goes down, the entity closest to the exit is immediately moved to the virtual location OUTSIDE. If the count stays the same, the closest entity to the event on the graph is moved there.

Every time an entity is created, destroyed or moved, that transition from one location to another is added to a matrix. The matrix represents the number of times entities transitioned between locations, and is the source of probabilities during the operation of the BUG/ED algorithm on new data. The length of wall clock time an entity resides at a given sensor location is also kept. This set of time lengths is used to determine dynamic timeouts for entities, which will be discussed in greater depth later.

Noise reduction in the BUG/ED probability matrix

Both the annotated data and the training algorithm for the BUG/ED matrix is not perfect. Inspection of the results shows several instances where the state of the model was such that taking the closest entity was inappropriate. This situation would increase the likelihood of transition between two locations improperly, but having a large enough training set mitigates the potential of these errors to impact the performance of the BUG/ED.

In some of the training data the human annotators were also incorrect in their resident count. Since that value is key to the training phase, these inaccurate training files would also impact the overall accuracy of the system.

To overcome these aberrant transitions between sensor locations caused by training flaws, a flooring filter was applied to the transition probability matrix. Any transition likelihood below the filter would be changed to the lowest probability. Setting this flooring value was observed to have a profound effect on the behavior of the system. If too many inappropriate connections were retained, and the floor were set too low, then the BUG/ED would have too little evidence to create new entities as people entered the space. Alternatively, if the floor were set too high then too many entities would end up being created. For each data set, the value to floor was experimentally derived. In future work, a proper outlier detection algorithm for each sensor location should replace this flat number.

An additional noise reduction tool was implemented to remove training data that was too complex for good use, as noted anecdotally by the human annotators in Section 4.2.2. This noise reduction came in the form of a maximum resident occupancy limit on the training data. As the number of residents increased within a space, it because more and more difficult to determine how many were actually present. This limit is determined by the sensor density and resident mobility. It was noted by the annotators that once more than five or six people were in the Tokyo testbed, it became nearly impossible to localize all of the residents. At that juncture, the annotators watched the entrance for people entering and leaving more than individual events anywhere in the space. Since the training algorithm to build the BUG/ED transition matrix is a simple one, a ceiling value on the number of occupants in the space was implemented. If the training data exceeded that number, it was thrown out. Between removing very unlikely connections and not using training data with too many residents, the BUG/ED tool started to perform more accurately in day to day use, and the overall utility of the system improved.

Dynamic Timeouts in the BUG/ED

In the GR/ED tool, a flat timeout for entities was enforced. This was set at 300 seconds, a figure experimentally derived by running the GR/ED tool on the data repeatedly with different timeout values. The overall accuracy at determining the number of residents was compared for each timeout. The best value of 300 seconds was taken for future work with the tool. This flat timeout of 300 seconds is the default used by the BUG/ED as well, though it is supplanted by the dynamic timeout algorithm described below.

It was noted by the residents that the GR/ED would timeout most often when people sat and worked in a location for a period without moving enough to cause sensor events. Because the training algorithm for the BUG/ED is stateful and remembers an entity's location indefinitely until they move, it could be used to find a more appropriate timeout for every sensor location. It was hypothesized that by making a dynamic timeout system that utilizes the training data, the BUG/ED would be improved when handling situations where entities remain still for long periods of time.

As the BUG/ED transition probability matrix is being trained, the length of total time an entity spends on a given sensor is kept. Once the data has all been used for training, these lists of times are used to calculate a customized timeout value for each sensor location. The mean plus three standard deviations of the time lengths in a sensor's list was used for the timeout value at every given location.

Manual inspection of the customized timeouts largely conformed to the expected pattern. Areas such as hallways and kitchens had shorter timeout values, while desks, beds and couches ended up with longer timeouts. This was not always true, but the flaws in the timeout calculations were results of flaws in the simple training rules used to build the transition probability matrix, and not the timeout calculation algorithm.

BUG/ED Bayesian Updating

During operation of the BUG/ED a model of the current entity locations is maintained. This model is modified by motion events with an "ON" message arriving. The only two things that may occur are either an existing entity is moved to the location of the event, or a new entity is created.

The likelihood that an entity e of all existing entities E has moved to the sensor location s_k of the sensor that fired from the entity's old location $e_{s_{k-1}}$ is calculated using Bayes' Rule in equation 4.2.

$$arg\ max_{e \in E}\ P(e|s_k) = \frac{P(s_k|e_{s_{k-1}})\ P(e)}{P(s_k)} \tag{4.2}$$

The value of $P(s_k|e_{s_{k-1}})$ is taken from the probability transition matrix. This is the likelihood that the entity transitions from their current location to where the latest sensor event is located based upon the historical training data. If the transition never occurred in the training data, then it was given a very small minimum value based on the smalled existing value in the transition matrix.

The factor $P(e)$ is considered the same for all entities, as they all have an equal likelihood of moving at any given time. This value could be modified with information about the likely direction, speed or likelihood of movement based on training information and become a serious factor in future versions of the BUG/ED.

The last value in the denominator of $P(s_k)$ is the same for all entities as it is the probability that the given sensor fired. Since this is a constant for all entities being compared, it is only a scaling factor.

Of the existing entities, the one with the highest likelihood (P_{move}) of making the transition to the sensor that fired is chosen to move in the model. This likelihood is compared to a threshold of the probability to create a new entity in the model instead of moving an existing one (P_{create}). If ($P_{move} < P_{create}$), then a new entity is initialized at s_k and the number of active entities in the model increases by one. Otherwise, the most likely entity to move has its tracklet of events increased by adding the most recent event and its location is updated to s_k.

At this juncture the BUG/ED has an updated model from the old model with the new evidence from the latest event. These updates reflect the most likely series of events based on the historical training data.

Summary

The BUG/ED tool is a modification to the Bayesian Updating approach to model building. This algorithm has a means to address the complexities and subtleties of the physical sensor placement that the GR/ED tool cannot accomplish with its simple graph and rule-based solution. Overcoming the vagaries of smart home sensor installations will be an ongoing issue for commercial installations, so approaches like the BUG/ED will continue to be required for managing the uncertainties of the real world.

4.3.3 Particle Filter-based Entity Detector: PF/ED

The PF/ED tool was designed as an alternative approach to the Bayesian Updating strategy of the BUG/ED. With this new algorithm, a sequential Monte Carlo method (Particle Filter) is initialized for each entity in the model. These filters are then updated according to the Action Model representing a person, and scored based on the current sensor evidence.

In a particle filter, a filter is defined as a set of possible hypotheses. Each hypothesis, called a particle or sample, represents a single possible state for the target entity. For example, a common approach to determining the state of a robot would be to represent its coordinates, orientation and velocity.

Particles have their estimated state values changed at every cycle of the algorithm according to an Action Model that represents a distribution the target's likely movement. For example, a history of training data may be used to build a distribution of likely distances and turns for a robot. Alternatively, a recording of movement data about humans may be used to constrain the Action Model to be a good representation of what is likely to happen. At every update of the filter, each variable in the state is modified by drawing a random number based on its distribution. This random selection identifies Particle Filters as a Monte Carlo method. The Action

Model may be as complex or as simple as needed for the application.

Each particle is paired with a weighted value demonstrating how likely it is. This weight is influenced by any evidence available about the ground truth, often coming in the form of sensors. After the particles of the filter are updated using the Action Model, their weights are re-evaluated using new information from the Sensor Model. This Sensor Model is an algorithm determined by which sensor(s) are being used to determine what most likely happened after the target being tracked changed its state. These sensors may be range finders, motion detectors, cameras or anything else available to the algorithm. For example, a particle that places the target closer to the reading from a range finder would be given a higher weight than one that differs greatly. Like the Action Model, the Sensor Model may be as simple or as complex as needed for the given application.

If a particle's weight becomes too low, it will likely be removed from the filter when resampling occurs, as it is doing a poor job of estimating the target's likely state according to the Sensor Model. Resampling is the process of replacing the current set of particles in the filter with a new set. The new set is most commonly chosen using a by particle weight, weighted random selection from the old set of particles. Many of the most likely (heavily weighted) particles will be copied multiple times by this process, which will re-center the filter around the most likely state of the target. Resampling may occur with every run of the filter, or when the Effective Sample Size

(ESS) of the filter falls below a certain threshold.

When the estimated state of the target is to be logged or used by another tool, the current particles in the filter are used to calculate a centroid value. There are a number of ways to determine the current estimated centroid for the target. A common method is to take the weighted mean of all of the particles, though using the particle with the heaviest weight or a subset of the particles are other options. Once the centroid of the filter has been calculated, that value is returned as the most likely current state of the target.

Particle filters are widely used for tracking and state estimation needs in robotics and sensor systems. Their flexible Action and Sensor Models allows them to adapt to complex problems, both with and without a corpus of training data.

PF/ED Core Algorithm

The incremental updating nature of a particle filter is well suited to the CASAS environment, as data comes in the form of discrete events. The algorithm runs to update the model of the space similar to the GR/ED and BUG/ED algorithms by running once after every new sensor event. After every PIR motion detector "ON" event received, the PF/ED algorithm operates as follows:

1. Find physically closest filter to "ON" event through Euclidean distance

    * If this filter's centroid is too "far" from the event, initialize a new filter at

the event's location, leading to more entities represented within the model

2. Update the particles of filter according to the Action Model

3. Determine the sensed state of the environment

4. Update weights of particles based on the Sensor Model

5. Normalize particle weights

6. Resample particles if the Effective Sample Size falls under the threshold

Each of the PF/ED's filters are made of $n = 1000$ particles. Every particle is a double of only a Cartesian pair: $< x, y >$. This algorithm does not estimate an entity's velocity or direction. Future versions will likely try to include these values to improve upon the current simplistic Action Model.

The centroid of the filter, which represents the most likely coordinates ($< centroid_x, centroid_y >$) of the resident, is calculated as a weighted mean of the all n current particles within the filter. This weighted mean is derived by taking the sum of the coordinates $< p_x, p_y >$ of each particle p multiplied with the weight of the particle p_w, according to equations 4.3a and 4.3b. The resulting coordinates represent the entity's location as estimated by the filter. This analog approach operates in contrast to the GR/ED and BUG/ED solutions where entities exist solely at the location of the sensors themselves and never in-between.

$$centroid_x = \sum_{i=1}^{n} p_{i_x} * p_{i_w} \tag{4.3a}$$

$$centroid_y = \sum_{i=1}^{n} p_{i_y} * p_{i_w} \tag{4.3b}$$

The Action Model used for the PF/ED algorithm is a random walk. The particles are moved from their current $< x, y >$ coordinates on both x and y axes by a random distance according to equation 4.4, where k is the event number in the data series, $range = 0.3m$, and $Randnormal(mean, stddev)$ randomly generates a number using a normal distribution with the given mean and standard deviation. This simplistic model could be replaced to improve the capabilities of the PF/ED by taking into account recent activity or historical training data in future versions.

$$x_k = x_{k-1} + (range * Randnormal(0,\ 1)) \tag{4.4a}$$

$$y_k = y_{k-1} + (range * Randnormal(0,\ 1)) \tag{4.4b}$$

For many applications, the process of calculating results from the sensors is complex. This might include processing of video data, audio streams or a large set of range finder measurements. Due to the simple sensors and the uniform size of their viewing range in the CASAS testbeds, the sensor state of the particle filter is merely determining the $< x, y >$ coordinates of the sensor that fired. With those coordinates in hand, the Sensor Model specifies that the entity is within 0.6m of the center of the sensor. All particles that fall within this square are given more weight than those that fall farther away.

The weight of each particle p at event k is updated by determining if its new location is within the viewing range of the sensor s that fired according to equation 4.5. This algorithm means that particles that now fall within the sensor firing retain 90% of their old weight, while all others merely retain 10%, thereby drawing the centroid of the filter towards the area covered by the sensor.

$$w_k = \begin{cases} w_{k-1} * 0.90 & \text{if } p_x = s_x \pm 0.6 \text{ and } p_y = s_y \pm 0.6 \\ w_{k-1} * 0.10 & \text{otherwise} \end{cases} \tag{4.5}$$

After the filter weights $w_1, ..., w_n$ are updated by the Sensor Model, they are then normalized according to equation 4.6. This process prevents the weights from becoming too small and diverging in scale over sequential updates of the filter.

$$w_{i_{new}} = \frac{w_{i_{old}}}{\sum\limits_{i=1}^{n} w_i} \tag{4.6}$$

Lastly, the particles in the filter are resampled if the Effective Sample Size (ESS) falls under the threshold of $threshold = 0.80 * n$. The ESS for the current particle population is calculated using equation 4.7.

$$ESS = \frac{1}{\sum_{i=1}^{n} w_i^2} \tag{4.7}$$

After the update and possible resampling are finished, the PF/ED algorithm waits until another sensor event is presented. The current implementation does not handle residents exiting the space, though it does handle people entering. Future updates will seek solutions to better handle the entrance and exit of residents.

Summary

The PF/ED's Monte Carlo-based algorithm was developed as a contrast to the Bayesian Updating approach of the BUG/ED. Using a tool that only requires the layout of the sensors and no training history would be more suitable for use in smart environments. The process of building a corpus of data to train the BUG/ED is currently a labor intensive one, and makes it difficult to deploy quickly in new installations. The PF/ED should be able to function from the moment the last sensor is installed with no additional configuration. This short installation-to-operation time frame is invaluable for commercial applications.

The current PF/ED is limited by the simplistic Action Model it uses. A more advanced Action Model will have a strong influence over the capabilities of the algorithm. Future updates to this model should include entity motion or temporal information which will improve the overall algorithm in the face of complex resident behaviors or sensor failure.

4.3.4 Occupancy and Tracking Algorithms Summary

Each tool built and evaluated in this work for the localization and tracking of smart home residents has benefits and negatives. Their primary benefit is the ability to operate in a smart home environment without a carried wireless device to track the resident, like those introduced in Section 2.1.1. They also protect privacy by eliminating the need for cameras or other video-based solutions such as the ones discussed in Section 2.1.2. They yield multiple streams of events, each of which represent an entity. These streams are individually less noisy than the complete stream provided directly by the sensor platform.

In exchange for these benefits, these tools only provide a probabilistic model of the space. This limitation needs to be balanced against the issues of carried devices and privacy concerns.

4.4 Tracking Algorithms' Results

The tracking algorithms in Section 4.3 were evaluated with the data sets introduced in Section 4.2.2. The *TokyoOccu*, *KyotoOccu* and *Tracking* data sets used were sometimes highly complex, with many residents and significant movement throughout the space. These tests and results are shown and discussed in this section.

Additionally, the ability for the BUG/ED tracking tool to boost ADL detection was tested with a *Kyoto* activity detection data set. This test was done to demonstrate the efficacy of the tracking tools at reducing the noise in the data stream and provide benefits to other smart home modeling tools. The process and results of this boosting test are shown in Section 4.5.

4.4.1 Testing the GR/ED

The GR/ED tool was tested for its accuracy at counting the current number of residents using both the *TokyoOccu* and *KyotoOccu* data sets. It was evaluated using 10-fold cross validation, divided by days. Once the data sets were run through the tool, the resultant guesses were compared to the human annotated ground truth. The results could then be inspected for the total number of events correct, as well as total length of time correct.

4.4.2 Results for GR/ED vs. Weighted Random on Occupancy

The *TokyoOccu* results were somewhat promising. GR/ED was very accurate with zero and one residents, as was expected, but rapidly fell to a lower accuracy as the number of residents increased. In Figure 4.5, the accuracy by number of events on the *TokyoOccu* data set is shown. Note that as the resident count increased, the accuracy declined. Since the GR/ED tool cannot tell the difference between a single or multiple residents at a given location, while the annotators can, it is often too low in its estimations. Additionally, it can be too high if an entity in the model is a false positive until it times out. Overall, the GR/ED algorithm achieved an overall accuracy of 72.2% with a standard deviation of 25.21% by the counting of events and an accuracy of 88.9% with a standard deviation of 12.8% for the total time represented by the data set. This is significantly better ($p < 0.05$) than a Weighted Random algorithm with a mean of 14.1% and a standard deviation of 0.2, but deeper inspection of the results shows a subtle difference in the operation of the GR/ED tool.

In Figure 4.5, the GR/ED is not significantly better than the Weighted Random algorithm for zero residents. This indicates that the GR/ED is not a good tool for determining when there are no people in the home. It does a significantly better job at one or more residents, but it has trouble detecting the exit of the last entity at the very least.

The *KyotoOccu* data set truly highlighted the flaws in the GR/ED algorithm. This testbed yielded notably more sensor error than the *TokyoOccu* data. Subjects were often able to move past sensors without causing events. With the GR/ED so reliant upon a fixed graph and no residents skipping sensors, the poor sensor coverage in *Kyoto* quickly led to many false entities being created in the model and a marked reduction in accuracy. Overall, the GR/ED had an accuracy of 16% measured by

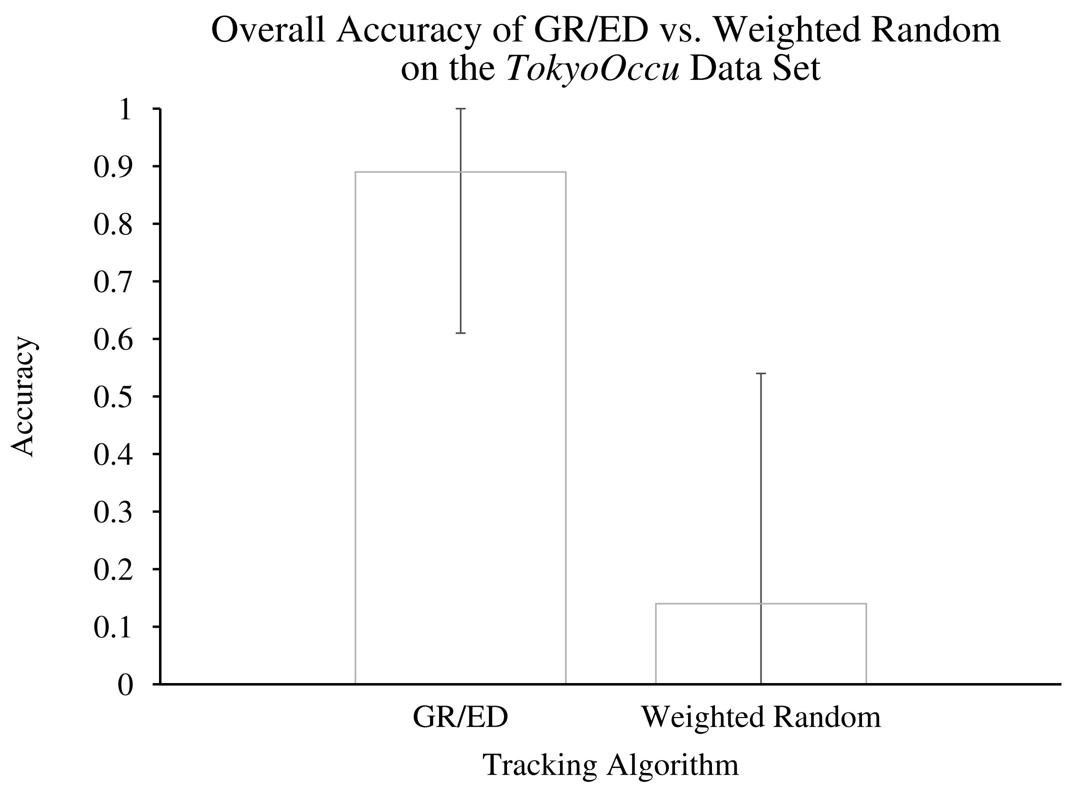


Figure 4.4: Accuracy by time correct for the GR/ED and Weighted Random tools on the *TokyoOccu* data set. The error bars show two standard deviations.

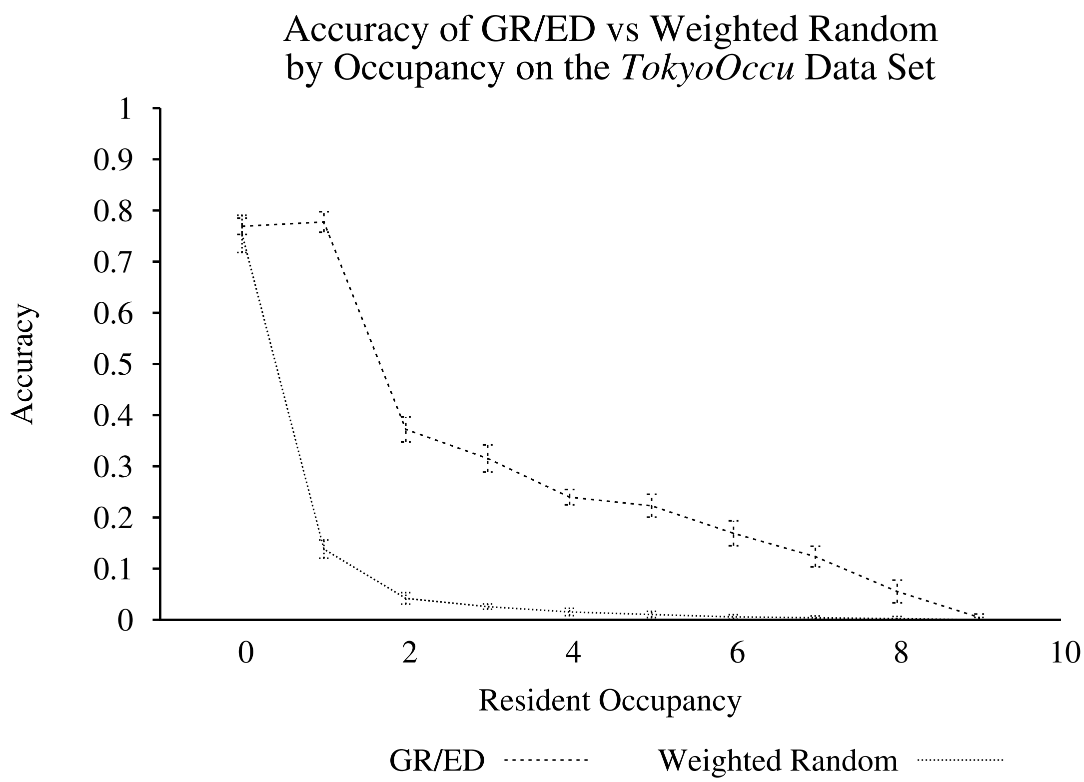


Figure 4.5: Accuracy by occupancy count for the GR/ED and Weighted Random tools on the *TokyoOccu* data set. The error bars show three standard deviations.

number of correctly-labeled events and 45% for total correctly-labeled time on the *KyotoOccu* data set. These low accuracies made it not different from a Weighted Random algorithm which was 14% accurate with a standard deviation of 0.2, so further evaluation of the GR/ED with the *KyotoOccu* data set was abandoned.

The GR/ED tool has the advantage over the BUG/ED of not requiring any training data. It requires only the graph representing the physical sensor layout itself. If the sensor locations can be determined at installation time, or automatically through some means, this tool can be quickly used with a new smart home installation. Depending upon the needs of the other tools within the system, it may be sufficient for the given smart home application.

Because the graph used by the GR/ED is so rigid and its performance so poor in the face of lacking sensor coverage, it was determined that a more probabilistic model might be a better solution. Instead of relying on a human hand-built set of equal connections between locations, perhaps a graph of likely connections derived from the annotated data might serve better. This hypothesis led to exploring a Bayesian Updating algorithm and the creation of the BUG/ED tool.

4.4.3 Testing the BUG/ED vs. GR/ED

The BUG/ED was tested using the same two data sets as the GR/ED tool for determining the occupancy of the smart environment. Because the BUG/ED requires training data, a 3-fold cross validation system was implemented. In this case, 2/3 of the available days were used to train the transition matrix, and the last 1/3 were held aside for testing. The days were randomly selected and the model was reset to no residents with each new day of testing, as one day of testing was not always followed by the proper consecutive day.

The overall accuracy value was calculated by counting the number of events where the BUG/ED was correct in identifying the current number of residents when compared to the human-annotated ground truth. The magnitude of the difference between the true value and the current guess by the tool was also calculated to give a sense of how far off the model was from the ground truth. Some error is to be expected since this is a probabilistic model. Depending upon the final intended use of the tools, an approximation might be sufficient for the smart home system's needs.

4.4.4 Results of the BUG/ED vs. GR/ED for Occupancy

As hoped, the BUG/ED performed better than the GR/ED tool on these data sets. It was noted by researchers watching the BUG/ED operate in real time that

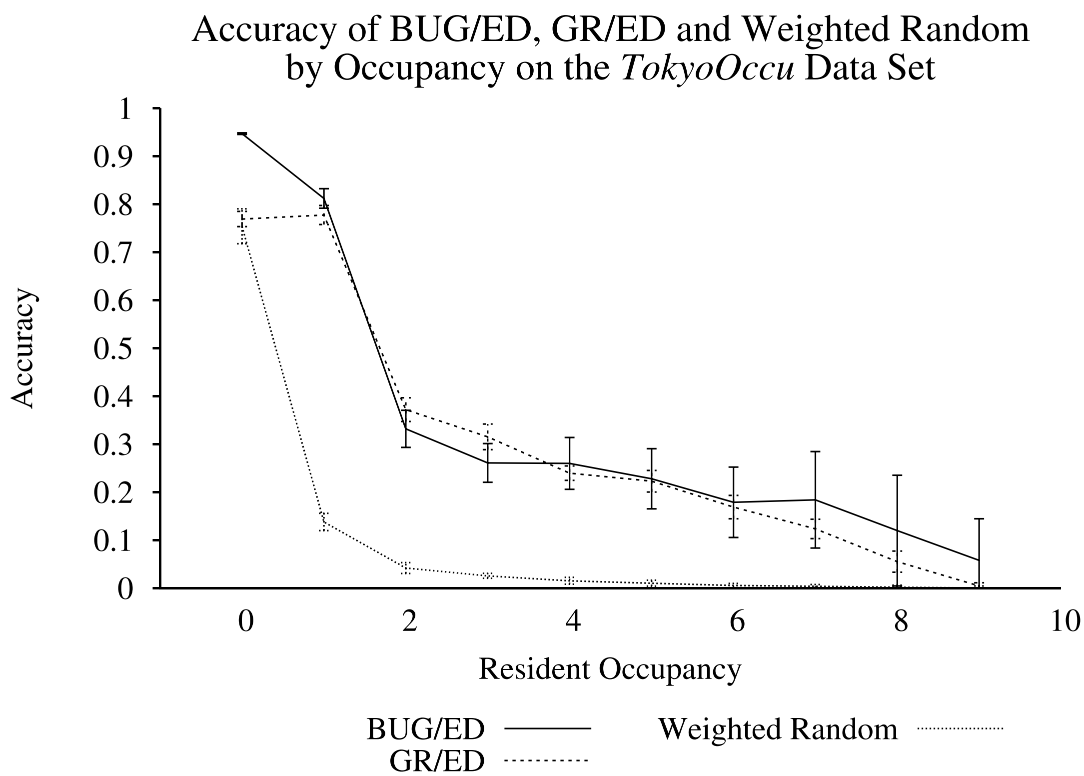


Figure 4.6: Accuracy by occupancy count for the BUG/ED, GR/ED and Weighted Random tools on the *TokyoOccu* data set. The error bars show three standard deviations from the mean.

it felt more "stable." Indeed, the BUG/ED failed less often in the face of skipping sensors and timed out less often when people stayed in one place for an extended period of time. These results were quantified by higher accuracy rates and measurable benefits to the ADL detection tools.

The BUG/ED tool's overall accuracy improved over that of the GR/ED on both occupancy data sets. Overall, the BUG/ED classified 44% of the events correctly, accounting for 85% of the total time on the *TokyoOccu* data set. It improved over the GR/ED tool when more entities occupied the space, though not significantly so. In Figure 4.6, zero and one residents are noticeably higher than the GR/ED results for the same data, but they are no different for more than that.

Where the BUG/ED significantly improved over the GR/ED tool was in detecting the final exits of residents from the space. This is visible in the 94% accuracy for zero residents, compared to the GR/ED's 77%. Both tools operated better than an random guess with more residents, but only the BUG/ED outperformed the weighted random at zero residents.

With a deeper analysis of the occupancy detection behavior of the BUG/ED on the *Tokyo* data set, a variety of details are exposed. Within Figure 4.7, subfigure 4.7(a) shows the accuracy of the BUG/ED within a number of residents. If the goal of the tracking tool is to estimate occupancy, then the BUG/ED will often be within one resident, and almost certainly two or three.

In Figure 4.7(b) the behavior of the results indicate that the BUG/ED overestimates on average when *Tokyo* has up to four residents, then underestimates with more. This over/under behavior is broken down is in Figures 4.7(c) and 4.7(d). Whether the BUG/ED overestimates or underestimates can be tuned by how the transition probability matrix is calculated and what filters are applied to the training data. Since these are probabilistic models of the space there will always be some inaccuracy. Knowing how to tune the tool to suit the needs of the smart home application is important.

The saddle shown in the Under Correct curves from Figure 4.7(d) corroborates the anecdotal evidence from the annotators about the resolution of the sensors in the *Tokyo* space. Once the number of residents exceeded four or five, the annotators reported significantly more trouble tracking individuals and resorted to watching the doorway for entrances and exits. The ability for the BUG/ED properly track every resident without occlusion, and eventually improperly removing entities from the model due to timeouts, occurs at the same occupancy levels. Adding more sensors, providing new types of sensors or changing the algorithm for determining occupancy would notably change the behavior of these kinds of tools.

Where the BUG/ED truly performed markedly better was with the *KyotoOccu* data set. While the GR/ED tool routinely failed as people traversed the space, the BUG/ED correctly track them would much more often. Figure 4.8 shows that in the most common state, an occupancy of two residents, the tool performs perfectly accurately just over 60% of the time. Overall, the BUG/ED classified 59% of the events and 67% of the total time for the *Kyoto* data set correctly. This was significantly ($p < 0.05$) better than the GR/ED tool on this data set.

These improvements in behavior and accuracy attest to the advantage of using a probabilistic model for decision making in this kind of tracking system. There are simply too many uncertainties with sensor placement, resident behavior, and system configuration to expect a purely rule based system to operate well.

4.4.5 *Testing the BUG/ED vs. PF/ED on the* Tracking *Data Set*

Both the BUG/ED and PF/ED tools were tested with the *Tracking* data set. They were given the task of determining the actual paths of the residents, as opposed to the earlier metrics in the *TokyoOccu* and *KyotoOccu* data sets where only the current quantity of residents was known. The BUG/ED tool was trained using a 3-fold cross validation, while the PF/ED used a probabilistic action model to update the particles. Both tools were run 30 times to calculate a significant accuracy mean and variance. The accuracy on this data set was determined by counting the number of resident sensor-to-sensor transitions the tool properly matched.

Table 4.5: Overall accuracy for BUG/ED and PF/ED algorithms on the *Tracking* data set.

Algorithm	Accuracy	STDDEV (σ)
BUG/ED	**92.0%** **	0.5
PF/ED	84.3%	0.2

4.4.6 Results on the Tracking *Data Set*

The results from the *Tracking* data set fall into four evaluations. The first is overall accuracy on all events, including both single and multiple occupancy of the smart home. These are summarized in Table 4.5 and shown in Figure 4.9.

For both single and multiple resident situations the BUG/ED algorithm significantly ($p < 0.01$) outperforms the PF/ED algorithm. This is likely due to the simplistic Action Model adopted by the PF/ED. With an Action Model for a resident that either uses an existing corpus of training data or one that incorporates more temporal information this gap in accuracy would likely close.

The second evaluation is on only the subset of the *Tracking* data set where a single resident was present. These results are summarized in Table 4.6 and shown in Figure 4.10. Again, the BUG/ED outperforms the PF/ED algorithm, though the

Table 4.6: Single resident accuracy for BUG/ED and PF/ED algorithms on the *Tracking* data set.

Algorithm	Accuracy	STDDEV (σ)
BUG/ED	**97.5%** **	0.2
PF/ED	93.8%	0.3

results are much closer then in the overall case.

The third evaluation is for only the multiple resident subset of the *Tracking* data set. These results are summarized in Table 4.7 and shown in Figure 4.11. Here, the BUG/ED strongly outperforms the PF/ED algorithm, though this overall score obscures the behavior of the two tools as the residents come physically closer to one another. This last evaluation of the accuracy of the tools over the separation of the residents shows that the PF/ED algorithm performs significantly better ($p < 0.05$) over the BUG/ED at a distance of zero or one sensor separating the residents. The accuracy across the resident separation is shown in Figure 4.12 and summarized in Table 4.8. Given the overall lower accuracy of the PF/ED tool, this strength at very short separation between residents shows promise if the Action Model and weighting algorithms are improved.

Table 4.7: Multiple resident accuracy for BUG/ED and PF/ED algorithms on the *Tracking* data set.

Algorithm	Accuracy	STDDEV (σ)
BUG/ED	**89.8% ****	0.7
PF/ED	80.1%	0.3

4.5 Tracking Noise Reduction for ADL Boosting

Many of the applications for smart environments that have been explored, e.g. health monitoring, health assistance, context-aware services, and automation, rely upon identifying the activities that residents are performing. Activity recognition is not an untapped area of research and the number of algorithms that have been used to build activity models varies almost as greatly as the types of sensor data that have been employed for this task. Some of the most commonly-used approaches are naïve Bayes classifiers, decision trees, Markov models, and conditional random fields [Maurer et al., 2004, Tapia et al., 2004, Cook and Schmitter-Edgecombe, 2009, Liao et al., 2005].

While activity recognition accuracy has become more reliable in recent years, most existing approaches are applied to situations in which a single resident is in the

Table 4.8: Multiple resident accuracy for BUG/ED and PF/ED algorithms on the *Tracking* data set.

	BUG/ED		PF/ED	
Separation	Accuracy	STDDEV (σ)	Accuracy	STDDEV (σ)
0	72.4%	2.4	**83.1%** *	1.3
1	70.5%	1.1	**76.1%** *	0.7
2	**84.8%** **	0.8	79.3%	0.6
3	**90.8%** **	1.0	79.8%	0.9
4	**93.3%** **	1.1	79.1%	0.8
5	**93.1%** **	2.6	71.0%	0.2
6	**96.9%** **	0.4	82.8%	0.6
7	**96.7%** **	0.4	81.8%	0.8
8	**99.0%** **	0.2	80.2%	1.5
9	**98.0%** **	0.3	85.2%	0.9
10	**97.7%** **	0.4	89.2%	1.0
11	**97.9%** **	0.5	90.4%	0.9
12	97.5%	0.7	95.9%	0.8
13	98.7%	0.4	98.4%	0.4
14	96.7%	1.2	98.3%	0.7

Table 4.9: Attributes of the three tested *Kyoto* ADL data sets.

Set Name	#Months	#Residents	#Activities
Set 1	2	2	12
Set 2	2	2	13
Set 3	5	2	25

space performing activities. Recognition accuracy notably degrades when multiple residents are in the same space. We hypothesize that this accuracy can be improved if the data is separated into multiple streams, one for each resident, or if each event is labeled with the corresponding resident identifier.

To validate this hypothesis, we applied the BUG/ED algorithm to data collected in the *Kyoto* apartment while two residents lived there and performed normal daily routines. The data used for this experiment actually represents different time frames, different residents, and different activities than was used to train the BUG/ED transition probability matrix. Attributes that describe these three data sets are shown in Table 4.9.

To demonstrate that the BUG/ED strategy is useful in further smart home tools, it was used to annotate these three new sets of *Kyoto* data. That data was then used to train and test naïve Bayesian ADL detector. The results with and without the

Table 4.10: Before and after ADL detection accuracies when adding BUG/ED tracking information to *Kyoto* data.

Set Name	Without BUG/ED	With BUG/ED
Set 1	42%	40%
Set 2	63%	88%
Set 3	54%	63%
Overall	56%	67%

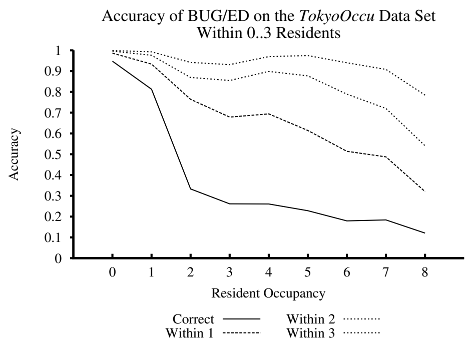
BUG/ED tracking information were compared and summarized in Table 4.10.

These three data sets are annotated for 11 different ADLs in an unscripted environment. There are two residents, though one or even more than two might be present at any given time. The data sets cover nearly a full calendar year in total, and run all day every day. The overall improvement to complex ADL detection was just over 10%.

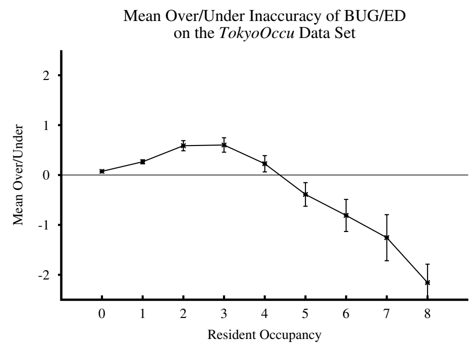
4.6 Localization and Tracking Summary

Any real-world smart home implementations need to address the reality of multi-inhabitant situations. This can be done through tracking devices, scene analysis and/or proximity sensor analysis. There are numerous trade-offs with sensor complexity, installation costs, and privacy issues involved with choosing the best solution for a given implementation.

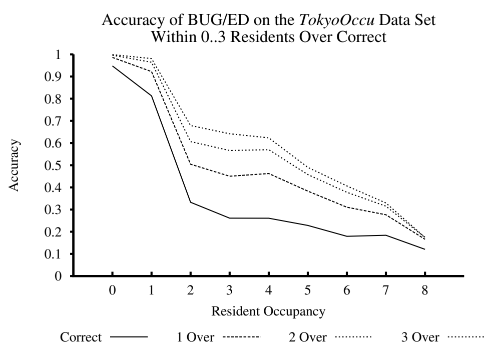
Historically, the smart home community has either pushed aside the multi-inhabitant problem or leveraged wireless tracking and scene analysis, but there exists a need for stronger privacy-sensitive solutions. The algorithms introduced in this work leverage passive, low profile privacy-protecting sensors to provide the benefits of a localization and tracking system. These do have limitations in the number of residents they can handle, and require more installed devices than other solutions. Both of those issues need to be considered when making the best choices for a given smart home application. The GR/ED, BUG/ED and PF/ED tools introduced in this work are significantly better than a purely random guess and run efficiently on commodity hardware, advantages that make them a solid choice for localization and tracking to boost other smart home applications such as ADL detection.



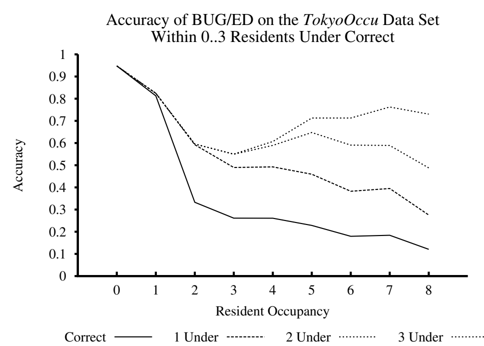
(a) Accuracy within n residents.



(b) Average inaccuracy of occupancy detection. Error bars show two standard deviations.



(c) Accuracy over correct within n residents.



(d) Accuracy under correct within n residents.

Figure 4.7: Behavior of determining occupancy of the BUG/ED on the *Tokyo* data set.

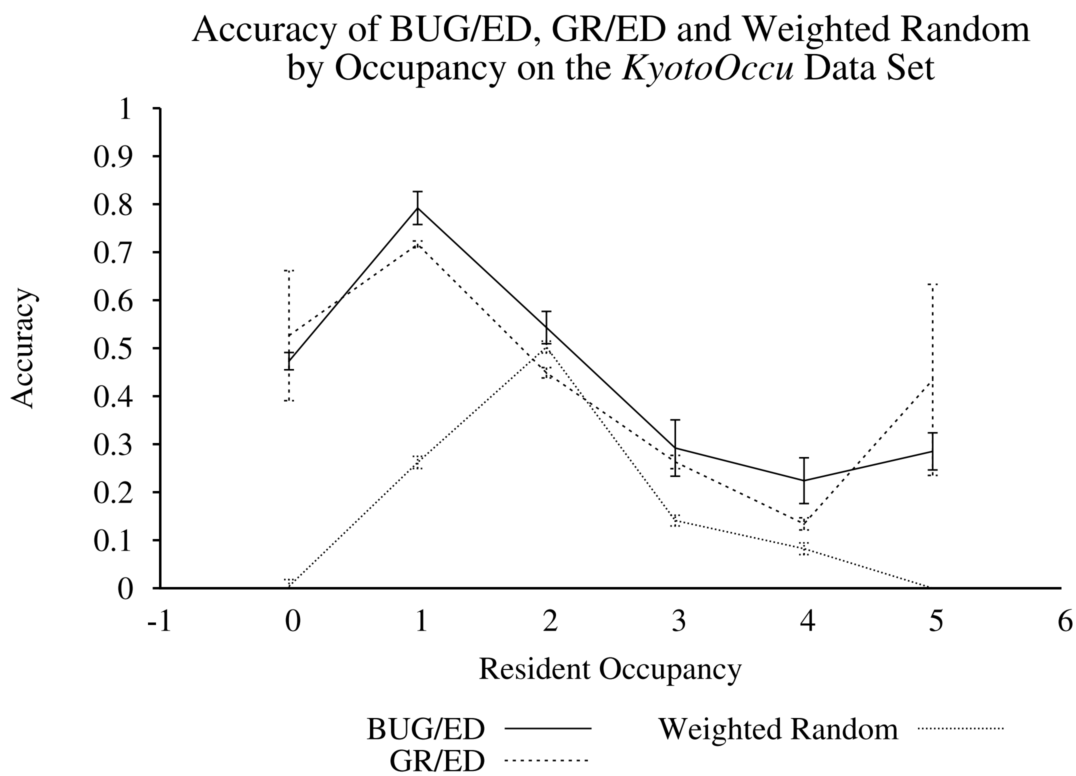


Figure 4.8: Accuracy by occupancy count for the BUG/ED, GR/ED and Weighted Random tools on the *KyotoOccu* data set. The error bars show two standard deviations from the mean.

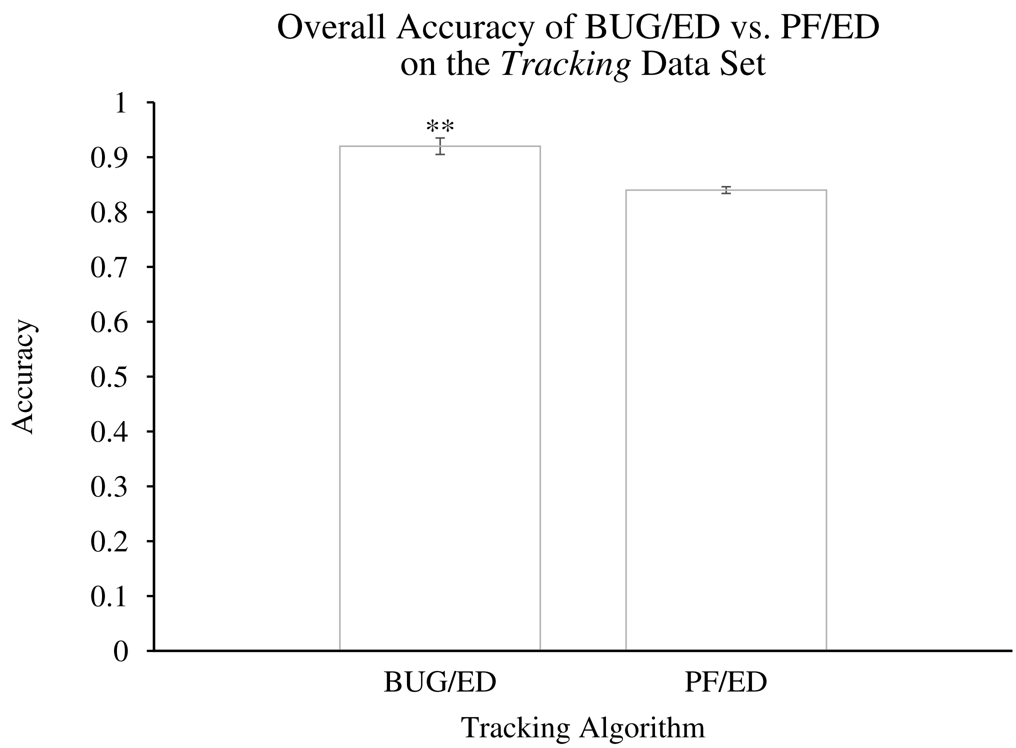


Figure 4.9: Overall accuracy for both BUG/ED and PF/ED algorithms on the *Tracking* data set. The error bars show three standard deviations.

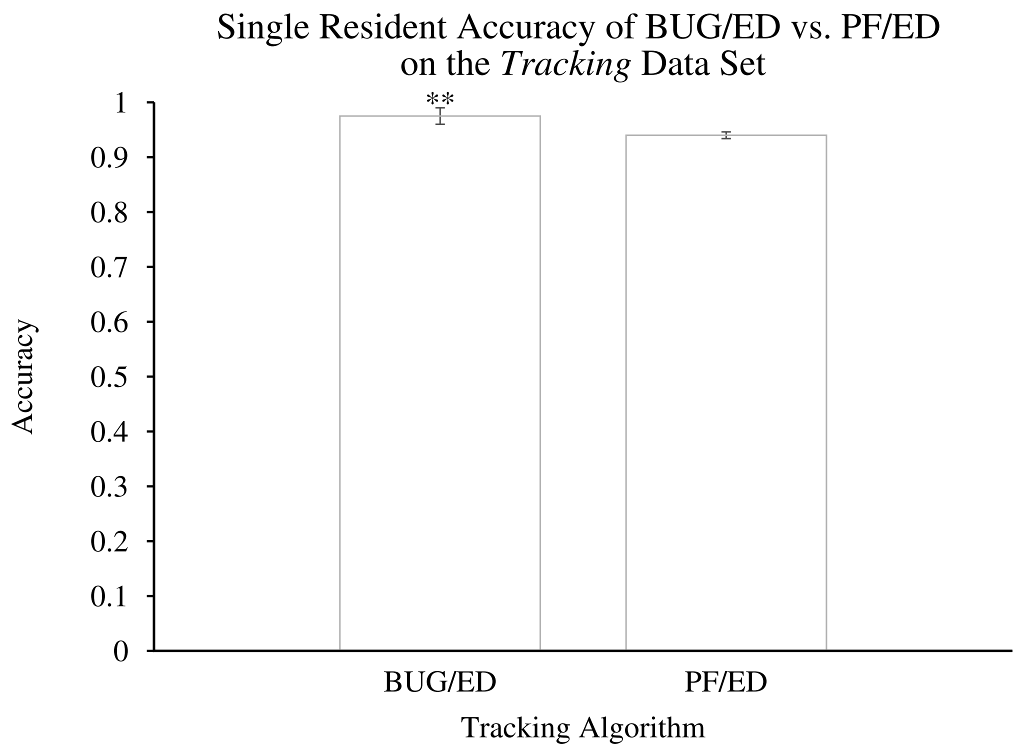


Figure 4.10: Single resident accuracy for both BUG/ED and PF/ED algorithms on the *Tracking* data set. The error bars show three standard deviations.

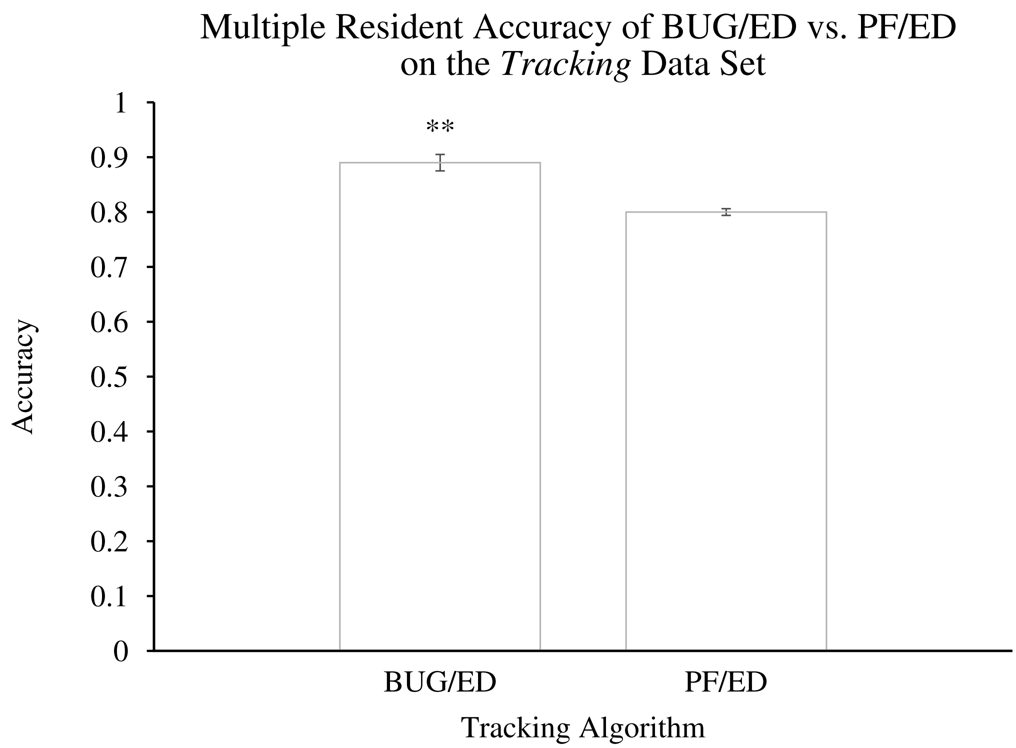


Figure 4.11: Multiple resident accuracy for both BUG/ED and PF/ED algorithms on the *Tracking* data set. The error bars show three standard deviations.

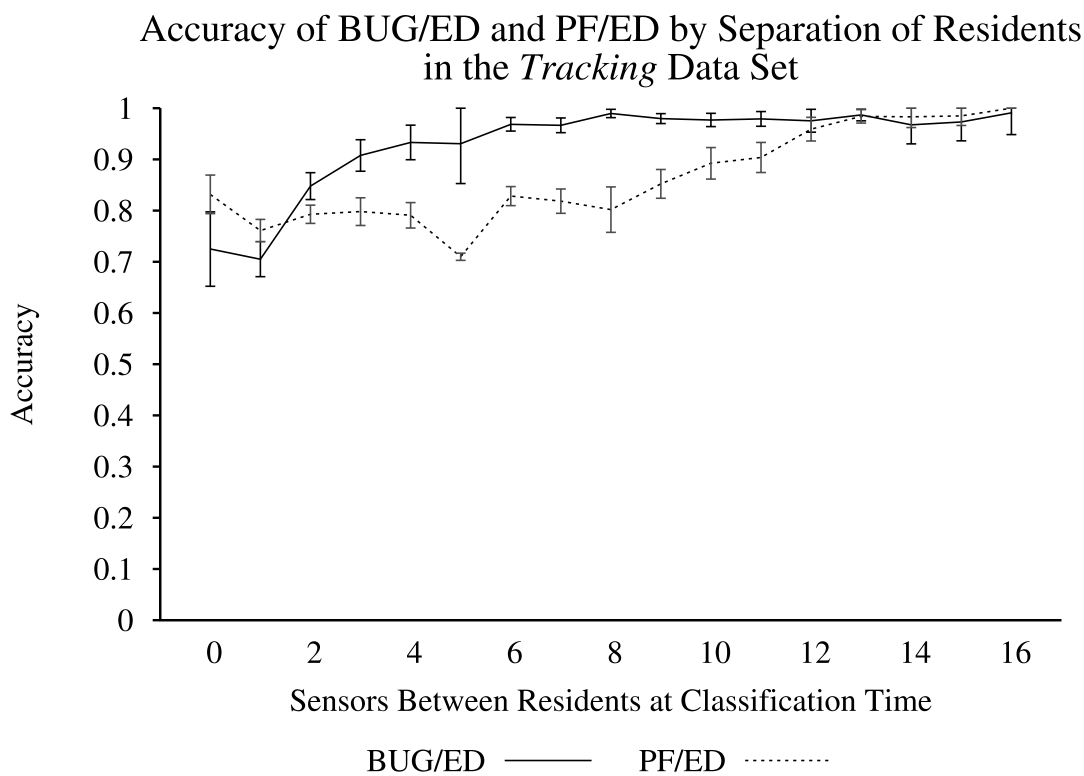


Figure 4.12: Accuracy of the BUG/ED and PF/ED tools over the separation between multiple residents in the *Tracking* data set. The error bars show three standard deviations.

CHAPTER 5. RESIDENT IDENTIFICATION APPROACHES

The second half of this thesis is centered around the utilization of behaviometrics to identify the current residents of a smart home. As shown in the chapter on related works, behaviometrics have been used to identify individuals via direct interaction with computers via keyboards and mice. The hypothesis regarding identification of residents from Section 1.5 applies this concept to the passive interaction residents have with the smart home sensors surrounding them. This chapter provides several algorithms and evidence to support using this kind of approach to identify individuals in a smart home space without the need for wireless tags or cameras.

5.1 Identification Introduction

Using the CTP and two of the CASAS research testbeds, three algorithms have been designed and implemented to identify residents. These algorithms all demonstrate a form of behaviometrics, introduced in Section 2.2.2, and primarily leverage the PIR motion detectors as their data sources. They all have the goal of using a resident's history of sensor events as training data to re-identify those individuals in the future. Additionally, the algorithms possess the ability to incorporate temporal and biometric sensor sources for more accurate identification.

The goal of the three algorithms described in the following sections is to let the data speak for itself. Rather than a hand-built tool for a unique smart home or resident the project's goal was to construct a general suite of approaches to the identification of entities on any kind of sensor network. The algorithms evaluated here were tested within the smart home domain, but have applicability in other environments such as web tracking or security systems. They are all based on well-established classifiers that have been applied with success to other smart home problems. Their varied algorithmic behaviors differentiate between residents in different ways. These algorithm differences are discussed and contrasted in Section 5.4.

Each of the algorithms and how they were evaluated is introduced in the next sections. A variety of data sets from two of CASAS testbeds, *Tokyo* and *Kyoto*, provided a number of different residents and occupancy situations to expose and contrast the behaviors of the algorithms.

5.2 Identification Research Layout

The algorithms built and tested for identification of residents draw upon data from CASAS testbeds, *Tokyo* and *Kyoto*. Results were assembled into a group of data sets, each with different facets that the algorithms can leverage to perform proper classification.

This approach to identifying residents builds upon our assumption that we need not wirelessly tag individuals in order to identify them, nor must we track them through the space. Instead, each resident proves unique in terms of the actions they perform in the space, and these differences will be evident in the resulting data generated by the sensors installed around them. As a result, if we have accumulated enough historical data that associates sensor event information with the resident that triggered the event, we can learn a mapping from sensor event features linked to a resident ID and use this mapping to identify the resident with future sensor events. Once the ID of the resident is determined for a sensor event, we can answer additional questions such as which residents are currently in the space, what is the total number of individuals in the space, and what are the activities that the residents are currently performing.

Since the goal of these algorithms is to properly attribute an event to a known resident, every data set is annotated with the identity of the person who caused each event. This data can then be used to train supervised learning tools to classify future events. While the three classifier algorithms make use of the same data to accomplish the learning goal, they employ very different strategies for identifying a mapping between the four features of a sensor event and the resident ID.

5.2.1 Research Design

This work uses an observational study method to evaluate the hypothesis that behaviometrics can classify individuals based on data provided by the CTP. Several CTP testbeds were installed and operated over a number of months to gather data for testing. During that time the residents' behaviors were recorded and annotated with their identity. The case series generated during these data gathering periods provides a suite of data for testing of the algorithms. The various algorithms were then tested against the different data sets to determine their ability to properly attribute each given event to a resident.

The CASAS team did not intervene with the residents while they lived in the smart home spaces and no attempts were made to adjust their behavior over time. The residents were consulted about their behaviors to ensure an accurate final ground truth during the data annotation period. The sole exception was for the first data set gathered in *Tokyo* that enforced single occupancy for identification purposes.

5.2.2 Identification Data Sets

There were three data sets used to evaluate the identification algorithms. They are summarized in Table 5.1 and come from both the *Tokyo* and *Kyoto* CTP-based testbeds, as described in Sections 3.4.1 and 3.4.2. These data sets had multiple residents reside in the space and were annotated by humans to provide a ground truth for the annotated classification of the residents.

The data gathered by the CTP for the identification data sets is represented by the following features:

1. Date
2. Time
3. Sensor Location[3]
4. Event message
5. Annotated class

The first four fields are generated automatically by the CASAS middleware at the time of the event's creation. The annotated class field is the target feature for our learning problem and contains the resident ID to which the other fields can be mapped. Sample data collected from a CASAS testbed is shown in Table 5.2.

The first data set, labeled *Workplace*, was gathered over the course of three weeks in the *Tokyo* smart workplace environment. During this time there were only

[3]Some older CASAS data sets use the sensor serial number instead of a device-independent location value.

Table 5.1: Summary of data sets used for validation of identification algorithms.

Data Set	*Alias(es)*	*Residents*	*Length*	*Num Events*
Workplace		3	10 days	6,000
B&B	Apartment	2	5 days	20,000
TwoR	Activity Tracking	2	56 days	136,504

three residents working in the space, and they were asked to log their presence by pushing a unique button on a pin pad when they entered and left the space. In order to generate training data for the learning algorithms, this first database was filtered to only use sensor events during the times when only a single resident was in the environment. In this way, it was ensured that each sensor event would be correctly labeled with the corresponding resident ID. Over 6,000 unique sensor events were captured, annotated, and used as data for our evaluation. Table 5.2 shows a portion of the data that was captured during this time.

For the second data set, labeled *B&B*[4], we collected sensor data from the *Kyoto* smart apartment while two residents lived there. This data set assesses the ability of

[4]This data set is referred to as the *Apartment* data set in [Crandall and Cook, 2009], and the *B&B* data set in [Crandall and Cook, 2010b, CASAS, 2010].

Table 5.2: Example of data used for classifier training.

Date	*Time*	*Location*	*Message*	*ID*
2007-12-21	16:41:41.0764	L017	ON	Res1
2007-12-21	16:44:36.8230	L017	OFF	Res1
2007-12-24	08:13:50.2819	L007	ON	Res2
2007-12-24	14:31:30.6889	L007	OFF	Res2

our algorithms to identify residents even when they occupy the space simultaneously, a more challenging situation than the one presented by the *Workplace* data set. Each resident occupied a separate bedroom, but regularly shared the common space downstairs. Unlike the previous data set, we made no constraints on resident activities and did not ask them to log their presence. Instead, our team annotated the sensor data after it was collected and confirmed the annotation with the residents to ensure accuracy of the labels. The result was a corpus of over 20,000 unique sensor events collected over a 5 day period.

The third and final data set, labeled the *TwoR*[5], contains sensor events collected

[5]This data set is referred to as the *Activity Tracking* data set in [Crandall and Cook, 2009], and the *TwoR* data set in [Crandall and Cook, 2010b, CASAS, 2010].

over a period of eight weeks while two residents (different than those in the *B&B* data set) lived in the *Kyoto* smart apartment. As with the *B&B* data set, this was collected to evaluate the mapping of sensor events to specific residents. However, we also used this data set to test ADL detection with other algorithms. To demonstrate the benefits of first determining the resident ID for an event on ADL detection, we performed this activity recognition first without resident identifier information and then second when the data is enhanced by adding the automatically-labeled resident identifier to each sensor event. In this manner, we determined how well residents may be recognized and the degree to which this information aids in other multi-resident tasks such as activity recognition.

Between these three different data sets taken from two different testbeds and with a total of seven unique individuals, the tools proposed in this thesis are well tested smart home algorithms. The different sensor layouts and behaviors of the residents allow the algorithms to exhibit many capabilities when presented with complex data.

5.2.3 Assumptions of the Study

The algorithms used to do identification of residents make three notable assumptions.

1. That the testbeds used are good exemplars for future smart home implementa-

tions. Future systems will be constructed from an unknown variety of sensors, layouts and objectives that the CASAS implementations will mirror in some ways, but not others. Due to the lack of standardization across smart home systems and testbeds across the world, this is a common issue that all research in this field needs to address.

2. The participants' behavior was not significantly impacted by their awareness of residing in a smart home. Even given some deviation in their behavior caused by knowledge of the smart home around them, each individual would still behave in a pattern unique among the group of residents. It is assumed that each is still behaving in a personally uniform, but individually unique manner for identification purposes.

3. The events captured in the three data sets are representative of the people who were monitored. If their behavior were outside the norm during the data gathering window, then these tools would not be as functional on an ongoing basis if deployed in a real-world system.

5.2.4 *Limitations of the Study*

The algorithms introduced for identification of residents have two notable limitations. First is that if two people act very similarly, then the tools will have difficulty differentiating between them, as they only classify behavior, not the actual person causing the events.

Second, if a resident's behavior changes, then these tools will need to accommodate that new pattern of behavior, as they do not incorporate the ability to handle drift in the data sources. For long-term real-world smart home implementations, this shortcoming would need to be addressed.

5.2.5 Research Design Summary

This thesis combines a number of experiments for evaluation of our hypothesis that individuals can be identified through behaviometrics in a smart home context. The first component of this evaluation was the installation of the CASAS Technology Platform into a number of testbeds for the purposes of gathering data sets. The second element was the annotation of that data by tagging individual events with the unique person who caused that event. The last was the implementation and evaluation of machine learning algorithms designed to differentiate between the residents based upon their behavior alone. Taken together, this project constructed and evaluated a novel application of machine learning tools for the identification of smart home residents without an explicit identification device for each individual.

The next sections introduce each algorithm in detail. The results, evaluation, and comparison of each tool are discussed in Section 5.4.

5.3 Identification Algorithms

Three algorithms were developed to test the hypothesis that behaviometrics can be used to identify individuals in a smart home space. They are all based on well established machine learning algorithms and applied here to the smart home domain.

The three algorithms are:

1. NB/ID: A naïve Bayesian-based tool
2. MM/ID: A Markov Model-based tool
3. HMM/ID: A Hidden Markov Model-based tool

Each of these tools has requirements for operation and provides unique benefits when used to identify the current residents in a smart home space. Their needs and implementation details are discussed in Sections 5.3.1, 5.3.2 and 5.3.3. The resulting classification accuracies of the three data sets, the algorithms' behaviors, and further discussion are available in Section 5.4.

5.3.1 Naïve Bayes: NB/ID

The first algorithm built and tested for identification was based around a naïve Bayes classifier. In our study this tool was designated the Naïve Bayes / IDentifier (NB/ID). This classifier leverages Bayes' Rule to use the current event received to guess at the identity of the individual. Naïve Bayes classifiers have been used to good effect in other smart home contexts [Tapia et al., 2004, van Kasteren and Krose, 2007]. The location, message and time features from individual events were exploited to determine the resident's identity.

A naïve Bayes classifier uses the relative frequency of data points, their feature descriptors, and their labels to learn a mapping from a data point description to a classification label. The resident label, r, is calculated as shown in equation 5.1.

$$arg\ max_{r \in R}\ P(r|D) = \frac{P(D|r)P(r)}{P(D)} \tag{5.1}$$

In this calculation, D represents the feature values derived from the event to be classified. The denominator will be the same for all values of r, so we calculate only the numerator values. The numerator is made of $P(r)$, which is estimated by the proportion of cases for which the resident label occurs overall and $P(D|r)$ which is calculated as the probability of the feature value combination for the particular observed resident id, or $\Pi_i\ P(d_i|r)$.

Table 5.3: Naïve Bayes alternative time-based feature formats.

Type #	*Feature Type*	*Example*
1	Plain	M001#ON
2	Hour-of-Day	M001#ON#16
3	Day-of-Week	M001#ON#Friday
4	Part-of-Week	M001#ON#Weekday
5	Part-of-Day	M001#ON#Afternoon

NB/ID Data Features

For a given event, the resident ID is set by the annotation process, but the feature representing that event can be derived in a variety of ways. We could attempt to use only location and message information as input to the learning problem, as shown in Table 5.3 row #1, but this leaves out valuable temporal information about the resident behaviors. The remaining features, date and time, are more difficult to use. Both of these features have a very large number of possible values, so we were required to consider effective methods for abstracting date and time information. The different feature choices that could be considered for these values, as shown in Table 5.3, divide the data in different ways and capture resident behaviors with varying degrees of fidelity.

The "Plain" feature set provides a good baseline to compare with more complex parsings. The more complex parsings, such as Part of Week (e.g. Weekday or Weekend) capture more information about the given behavior, and can furnish the classifier with more information for correct future classifications. Depending on the facets of the data set, different feature types will cause the classifier to perform better or worse.

The different feature choices available (e.g. Plain vs Hour-of-Day, etc.) divide the data up in different ways. Each method captures the behaviors or the residents with varying degrees of accuracy, depending on the feature types chosen and the behavior of the individuals in the data set.

The purely statistical nature of a naïve Bayes classifier has the virtue of being fast for use in prediction engines, but lacks the ability to handle context within the event stream that could be advantageous in discerning subtle differences in behaviors. We test the accuracy of each of these time representations when we evaluate the NB/ID algorithm.

Summary

The statistical calculations of a naïve Bayes classifier offer the benefit of fast learning, but lack an effective approach to reasoning about context in an event stream. In order to capture this context we also consider other approaches to learning resident IDs, as described in the next sections.

5.3.2 Markov Model: MM/ID

In our second approach to resident identification we classify resident behaviors using Markov Models. A Markov Model (MM) is a statistical model of a dynamic system. A MM models the system using a finite set of states, each of which is associated with a multidimensional probability distribution over a set of parameters. The system is assumed to be a Markov process, so the current state depends on a finite history of previous states (in our case, the current state depends only on the previous state). Transitions between states are governed by transition probabilities. For any given state a set of observations can be generated according to the associated probability distribution.

Because our goal is to identify the activity that corresponds to a sequence of observed sensor events, we generate one Markov Model unique to each resident that we are observing. We use the training data to learn the transition probabilities between states for the corresponding activity model and to learn probability distributions for the feature values of each state in the model.

To label a sequence of sensor event observations with the corresponding resident ID, we compute r, the likelihood that the resident represented by the model cause the event, as $argmax_{r \in R}\ P(r|e_{1..t}) = P(e_{1..t}|r)P(r)$. $P(r)$ is estimated as before, while $P(e_{1..t}|r)$ is the result of computing the sum, over all states, S, in model r, of the likelihood of being in each state after processing a sequence of sensor events $e_{1..t}$ that leads up to the current time, t. The likelihood of being in state $s \in S$ is updated after each sensor event (e_j) is processed using the formula found in equation 5.2.

$$P(S_j|e_{1..j}) = P(e_j|S_j) \sum_{S_{j-1}} P(S_j|s_{j-1})P(s_{j-1}|e_{1..j-1}) \tag{5.2}$$

The probability is updated based on the probability of transitioning from any previous state to the current state (the first term of the summation) and the probability of being in the previous state given the sensor event sequence that led up to event e_j.

MM/ID Event Window Size

As with the naïve Bayes classifier, there are decisions to make regarding the presentation of the data that influence the performance of the Markov Model. The primary decision is the event sequence size to provide to the model. As described in Section 5.3.2, a series of events is provided as input to the model in order to output a resident identifier for the most recent event at time t. Because the series size should be the same for each calculation, we do not provide events starting at the beginning of the data collection for each label we generate. Instead, we provide a fixed number of events, or event window size, that occurs immediately prior to and include event t. During the evaluation of the MM/ID tool, this windows size is manipulated to determine which sizes are “good” ones for the given data set.

Summary

A Markov Model-based solution was selected for our second approach because this representation encapsulates additional contextual information. As a result, the context of the sensor event is used when labeling the event with a resident ID. By adding transitions between states in the model, the spatial and temporal relationships between sensor events are captured. Thus, by taking more of both the physical and the temporal information into account, we hypothesize that our algorithms will label events more accurately even when the number of residents increases.

5.3.3 Hidden Markov Model: HMM/ID

In Section 5.3.2, the MM/ID classifier was introduced that used Markov Models with no hidden nodes. As a result, we had to learn a separate model for each resident and run them all in parallel to determine the current resident's identity. To simplify the system by using a single model, we next designed a Hidden Markov Model-based algorithm. With this algorithm, a single model is used to encapsulate all of the residents and the sensor events they trigger. This HMM/ID tool was used to evaluate a Hidden Markov Model's ability to properly attribute events to residents.

Using the hidden Markov model, hidden nodes represent system states that are abstract and cannot be directly observed. In contrast, observable nodes represent

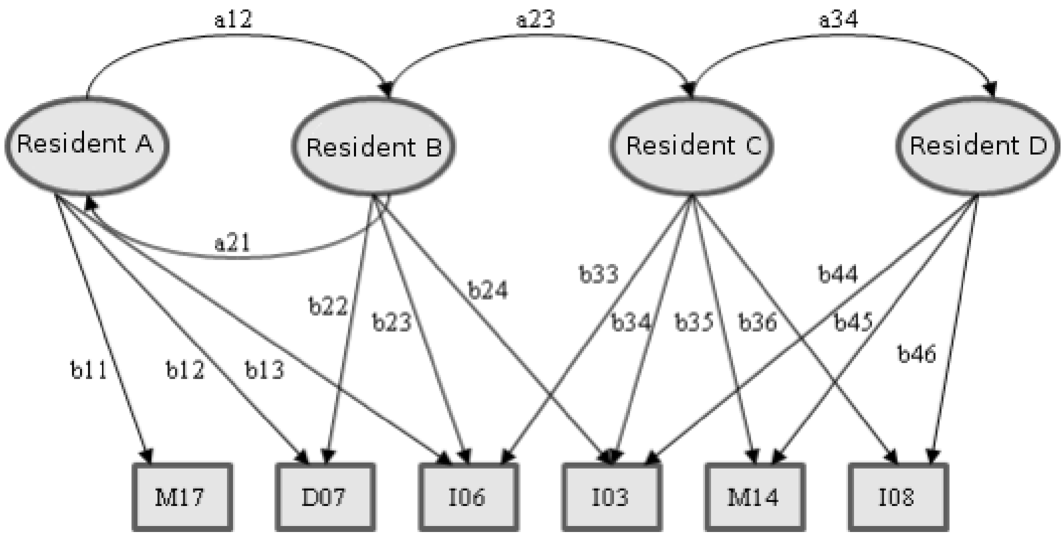


Figure 5.1: HMM architecture of hidden states, transitions and observations.

system states that can be directly observed. Vertical transition probabilities between hidden and observed nodes are learned from training data, as are horizontal transition probabilities between hidden nodes.

In our model, as shown in Figure 5.1, each hidden node represents a single resident. The observable nodes are associated with probability distributions over feature values including the motion sensor ID and the sensor message. We can then use the Viterbi algorithm [Viterbi, 1967] to calculate the most likely sequence of hidden states that corresponds to the observed sensor sequence. This sequence of hidden states provides us with the highest-likelihood resident IDs that correspond to each sensor event in the sequence.

In the HMM/ID structure, the states Y represents the residents from the training corpus. The start probabilities of each state i is kept in π_i while the transition probabilities $a_{i,j}$ are the likelihood of transitioning from resident i to resident j between events. If the testing corpus of sensors events is $x_0, ..., x_T$, then the state sequence $y_0, ..., y_T$ is the most likely attribution of these events to the residents represented by the states. This mapping is given by the recurrence relations in equations 5.3a and 5.3b. The result of $V_{t,k}$ is the probability of the most probable series of resident attributions for the first $t+1$ events in the testing corpus.

$$V_{0,k} = P(x_0|k) \cdot \pi_k \tag{5.3a}$$

$$V_{t,k} = P(x_t|k) \cdot max_{y \in Y} \ (a_{y,k} V_{t-1,y}) \tag{5.3b}$$

As a concrete example of how the HMM/ID algorithm is set up, refer to Appendix B. This small example shows a model built with two residents and three sensors along with a concrete source code example. The trained parameters are taken from the *B&B* data set with the list of shown sensors pared down for brevity. When the *determine_resident*() function is called, the result is the series of entities that most likely caused the series of events given. The output of the HMM/ID is a state sequence $y_0, ..., y_T$ of residents that map to the series of events $x_0, ..., x_T$ provided to be classified.

This algorithm no longer requires the fixed event window like the MM/ID. The events are taken one at a time without modification or manipulation, leaving the capabilities of the system entirely up to the ability of the algorithm and not choices made during pre-processing stages. The trade-off is that the tool often requires more than one event to transition between residents. It relies on some context-dependent amount of evidence for the HMM to transition from one hidden state (resident ID) to another. This sometimes leads to a delay in proper identification during operation, and is a source of error in the results. The behavior of the HMM for both transition lag error and confusion error are both discussed in Section 5.4.

Summary

Hidden Markov Models are robust in the face of noisy data and used for a number of smart home applications. The HMM/ID tool developed for classifying residents is based on a classic HMM approach and eliminates a number of shortcomings to the NB/ID and MM/ID tools developed earlier. This more complex algorithm reacts to the data in such a way that introduces multiple sources of error that are discussed in depth in Section 5.4.

5.3.4 Identification Algorithms Summary

Each of the tools built and evaluated for identification of smart home residents has benefits and negatives. The core algorithms were chosen because of their history of robustness and successful application within the smart home domain, though not for use as identification classifiers. Their different behaviors and capabilities give insights into what kinds of patterns can be exploited to identify individuals based upon behavior alone.

5.4 Identification Algorithms' Results

The identification algorithms introduced in Section 5.3 were evaluated with the data sets introduced in Section 5.2.2. The *B&B* and *TwoR* data sets are more complex in nature and provide a better overall evaluation of any identification tools than the *Workplace* data set. All of these tests and results are shown and discussed in this section.

Additionally, the ability for the identification results to boost ADL detection were tested with the *TwoR* data set. This test was done to demonstrate the ability for identification to provide additional features that may improve other models in the smart home context. The process and results of this boosting test are shown in Section 5.5.

5.4.1 Workplace *Data Set Results*

Results for using behaviometrics to identify residents in the *Workplace* data set have been published in several venues [Crandall and Cook, 2008c,b, Crandall et al., 2008, Crandall and Cook, 2009, 2010c]. It has been used to evaluate the NB/ID, MM/ID and HMM/ID tools. This is the simplest multi-resident data set available from the CASAS project, as it never involves simultaneous multi-resident occupancy of the space. Despite this shortcoming, it can be used to demonstrate that each resident has differentiable behavior and that diverse algorithms leverage these differences through various approaches without the complexity of true multi-inhabitant data.

Workplace with NB/ID

The *Workplace* data set was randomly split into training and testing sets, with 10% of each class set aside for testing. The classifier was trained on the 90% and run against the testing set. Each class was given an accuracy rate and a false positive rate. This process was repeated for each of our feature types for comparison of their various capabilities.

Figure 5.2 shows the classification accuracy of the NB/ID classifier for the three residents present in the *Workplace* data set. In order to keep actual participant names anonymous, we label the three residents John, Abe, and Charlie. In Figure 5.2 we

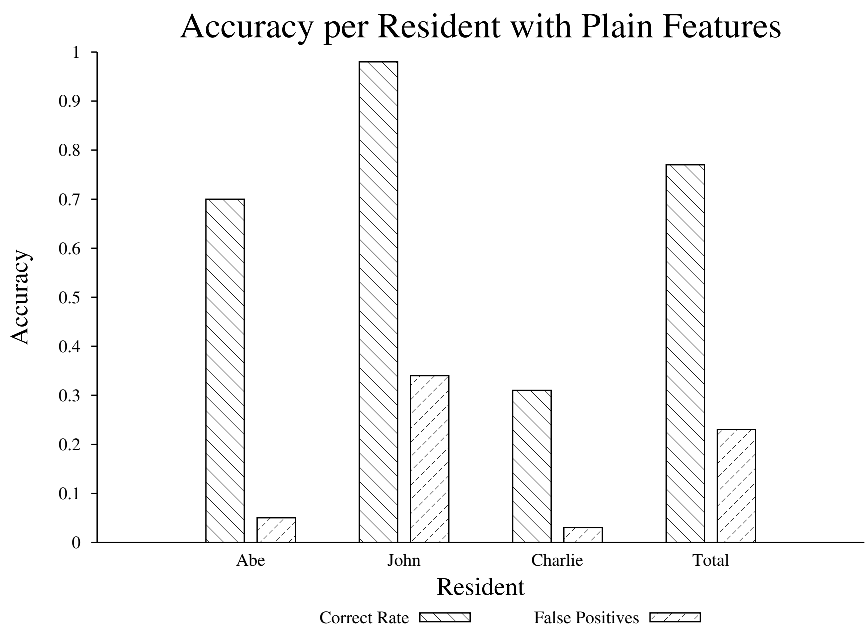


Figure 5.2: NB/ID *Workplace* data set accuracies and false positive rates with Plain features across all residents.

graph not only the classification accuracy for each target value, but also the false positive rate.

Note that the classification accuracy is quite high for the John values, but so is the false positive rate. This behavior is caused by our John participant being responsible for most (roughly 62%) of the sensor events in the training data. As a result, the *a priori* probability that any sensor event should be mapped to John is quite high and the NB/ID algorithm incorrectly attributes Abe and Charlie events to John inappropriately. On the other hand, while Charlie has a much lower correct classification rate, he also has a lower false positive rate. If the intelligent environment algorithms can take confidence values into account, this information about false positives can be leveraged accordingly.

In order to reduce these classification errors, more descriptive temporal features were generated from the data and time information contained in the data set. In particular, these are the various possible time features of each sensor event, as shown in Table 5.3. The classifier may use time of day or day of week information to differentiate between the behaviors of the various individuals. For example, John always arrived early in the day, while Abe was often in the space late into the evening. Discovery of the best features to use may be accomplished by balancing the overall correct rate and false positive rate against one another across all of the residents in the training set.

The choice of feature descriptors to use is quite important and has a dramatic effect on the classification accuracy results. Looking at the accuracy rate as effected by the feature type chosen in Figure 5.3 shows that using Hour-of-Day significantly ($p < 0.05$) increases the identification accuracy over the Plain feature.

An instance of the effects of time-based features on an individual's classification accuracy is shown in Figure 5.4. John has a very high accuracy rate across all feature types due to a high *a priori* weight. The NB/ID tool will often guess John, which leads to attributing too many events to his class. This gives him a very high false positive

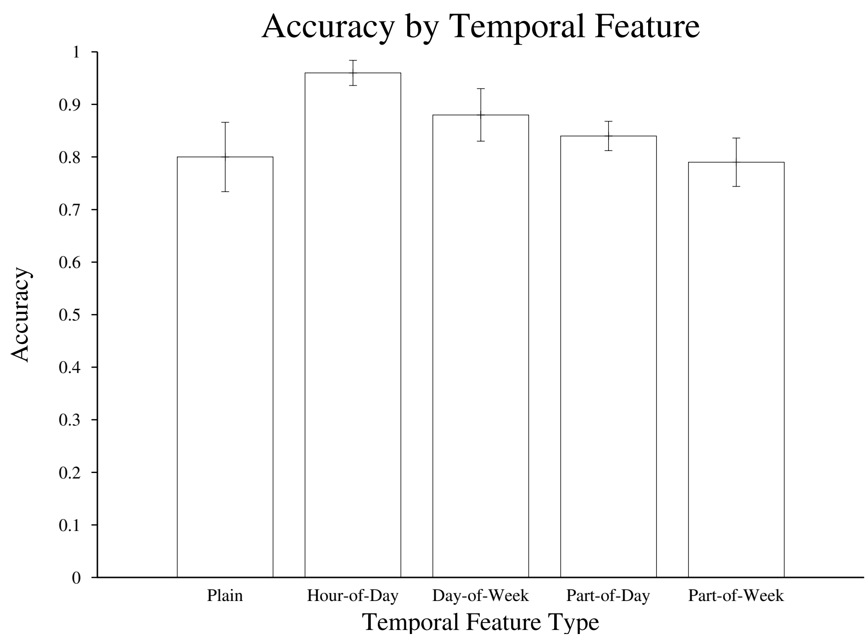


Figure 5.3: NB/ID *Workplace* data set accuracy for different temporal features. The error bars show two standard deviations.

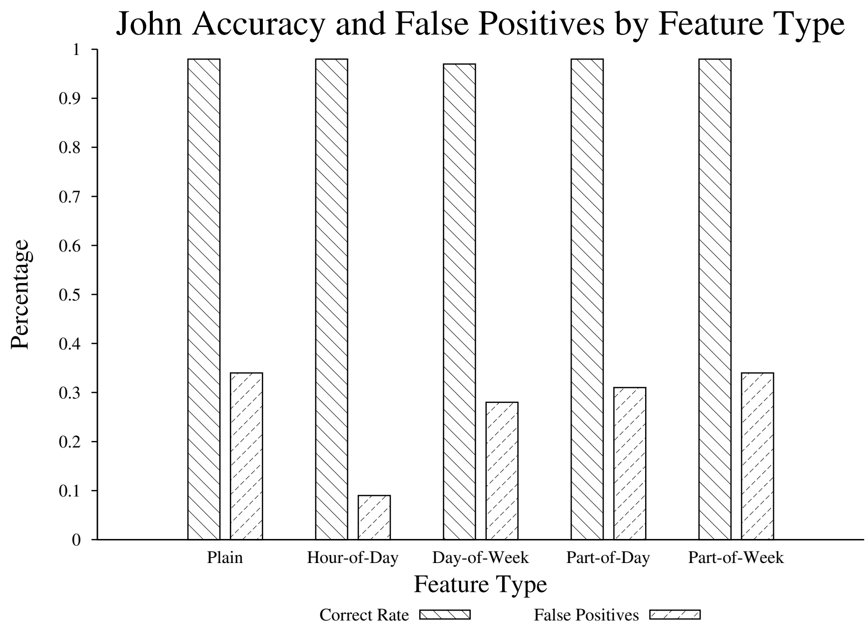


Figure 5.4: John accuracy and false positive rate by temporal feature.

rate with the Plain feature format, as shown in Figure 5.2. By adding the Hour-of-Day feature (Table 5.3, row 2), John's accuracy does not improve significantly, but his false positive rate drops dramatically, as shown in Figure 5.4. This significant drop ($p < 0.05$) from 34% to 9% is a marked improvement. Those events that were erroneously attributed to John are now being properly attributed to Abe and Charlie. Use of the other time-based features results in some improvements to John's classification, but none of the others is as useful as adding the Hour-of-Day feature.

In contrast to John, Charlie's behavior responds differently to the choice of feature type. To demonstrate the improvements in accuracy rate, refer to Figure 5.5.

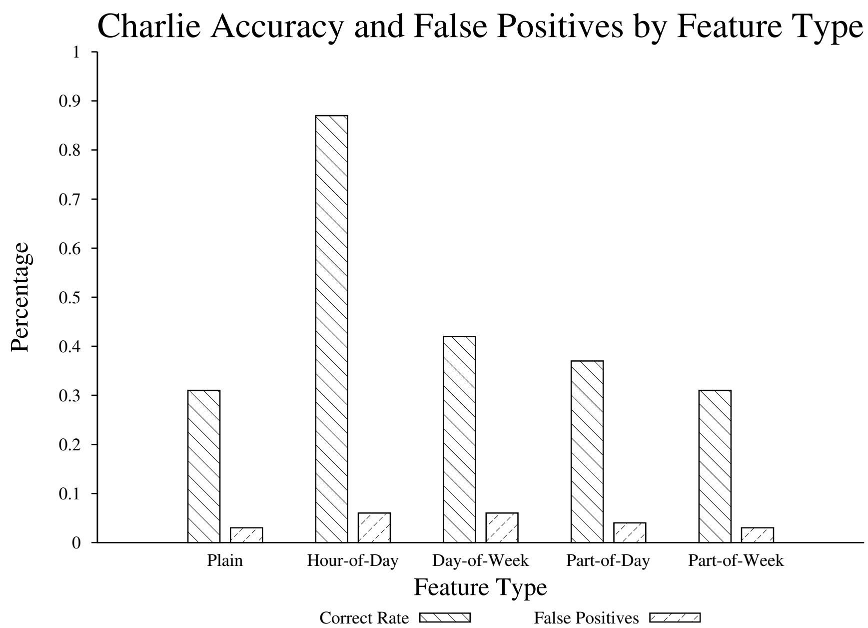


Figure 5.5: Charlie accuracy and false positive rate by temporal feature.

Charlie's initial 31% accuracy with simple features was shown to significantly ($p < 0.05$) jump to 87% by again using the Hour-of-Day feature type.

After further inspection of the data, this improvement is derived from the fact that Charlie's activities do not overlap as much with Abe or John during the day. This improvement comes at a cost, though. In exchange for a dramatic improvement to classification, Charlie's rate of false positives goes up from 3% to 6%, as shown in Figure 5.5. This kind of trade off needs to be taken into account by any system of deciding which features to use for the current classifier.

Time Delta Enhanced NB/ID Classification

Adding more features to our data over and above Hour-of-Day did improve the resident classification accuracy, though the improvements were not a great as anticipated. We hypothesize that one reason for the remaining inaccuracies is the type of sensor events we are classifying. Many motion sensor events occur when individuals are moving through the space to reach a destination, and are not particularly unique to the residents in the space. On the other hand, when a resident is in a single location for a significant amount of time, that location is important to the individual resident. They are likely performing an activity of interest in that location, and as a result the corresponding motion sensor data should be used for resident classification.

To validate our hypothesis, the data set was culled of all extra sensor events where the same sensor generated multiple consecutive readings and only the first event in the series was kept. The multiple readings were likely due to small movements occurring repeatedly within the one small area of the workplace. Replacing the set of readings with one representative motion sensor event allowed the sensor event to represent the entire activity taking place at that location.

With this reduced set of events, the Time Deltas, or time elapsed between the remaining events, were calculated. The chart shown in Figure 5.6 gives a count of how long an individual spent at any one motion sensor location before moving to a new location. The mean time spent on any sensor was 8.78 seconds, with a standard

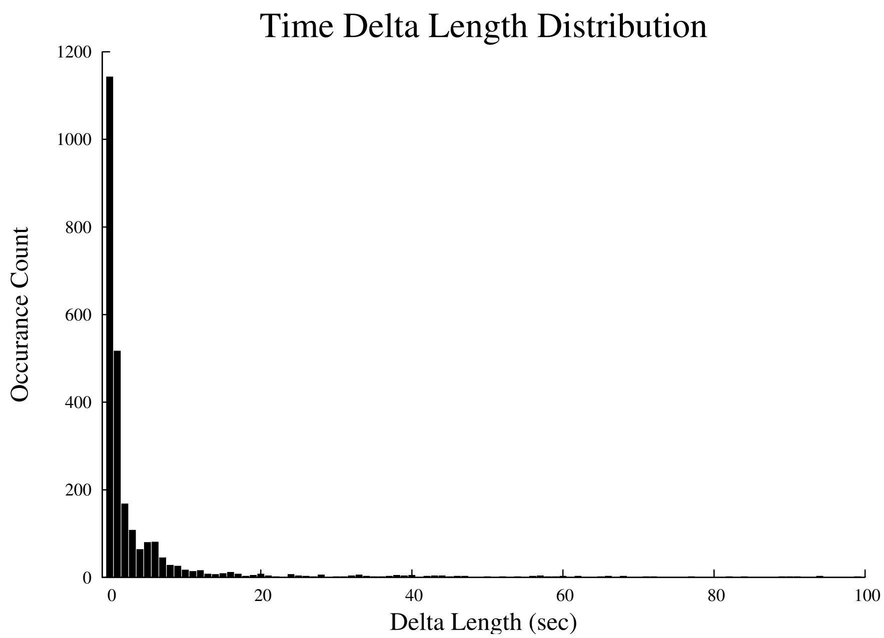


Figure 5.6: *Workplace* Count by Time Deltas.

deviation of 50.8 seconds. With a graph of this shape, the initial hypothesis of being able to garner additional information for training was borne out.

Next we removed from our data set any motion sensor events whose durations fell below the mean time delta value, thereby leaving the longer deltas. With an even more reduced set in hand, the data splitting, training and testing were all done the same way as before with the full data set.

The resulting classifier only used a handful of the available sensors throughout the living space, but the accuracy and false positive rates improved dramatically. This is attributed to the fact that motion sensors in shared spaces or walkways will mostly

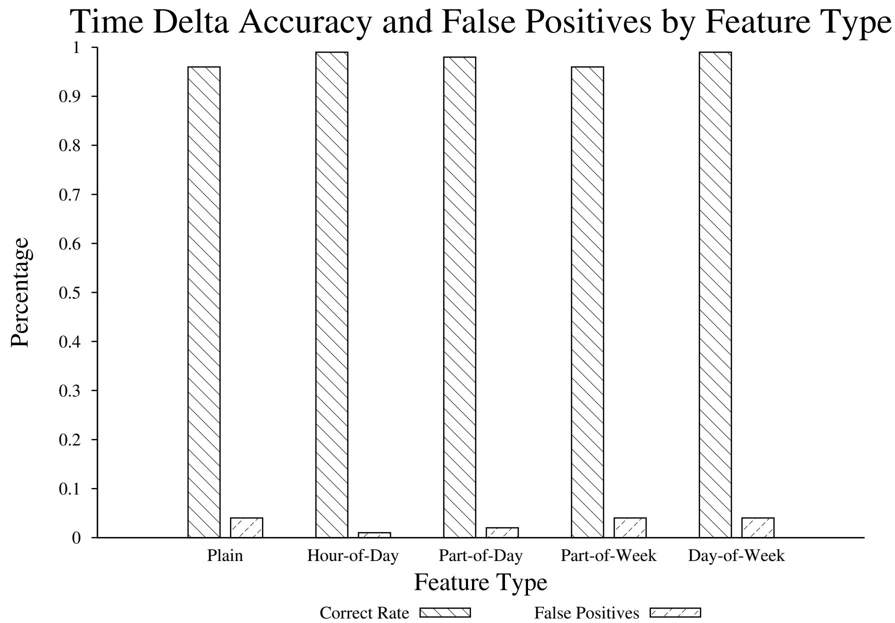


Figure 5.7: *Workplace* overall accuracy with Time Deltas.

have very small time deltas associated with them. Since these sensors are also the ones contributing to the false positive rates in the full set classification, removing these sensor events will improve the overall performance of the classifier. Note that with this filtered-data approach, sensor events with short durations will not be assigned a mapping to a specific resident. However, by combining this tool with one that tracks inhabitants through the space from Chapter 4, only a handful of sensor events need to be classified as long the tools maintain a record of who is moving where.

With the Time Delta filtered data set, the NB/ID had correct classification rates over 98%. Again, there was some difference in performance with different feature choices, as shown in Figure 5.7. Once again, the Hour-of-Day performed the best, as it seems to give the NB/ID classifier information that could be used to differentiate between resident behaviors within this particular set of data.

Workplace **with MM/ID**

The *Workplace* data set was also used to evaluate the MM/ID tool. These evaluations were published in fewer venues [Crandall and Cook, 2008c, 2009] than the NB/ID tool, primarily because the later HMM/ID tool eclipsed the capabilities of MM/ID as the data sets became more complex.

As with the NB/ID classifier, there are decisions to make that influence the performance of the MM/ID classifier. The primary decision is the event sequence size to provide to the algorithm. As described in Section 5.3.2, a series of events is provided as input to the model in order to output a resident identifier for the most recent event at time t. Because the series size should be the same for each calculation, we do not provide events starting at the beginning of the data collection for each label we generate. Instead, we provide a fixed number of events, or event window size, that occur immediately prior to and include event t.

Figure 5.8 shows the classification results from alternative window sizes. As the figure shows, the window size does have an effect on classification accuracy. Because a window size of 25 performs best for the *Workplace* data set we use it for the remainder of our experiments. In general, the algorithm can experimentally derive

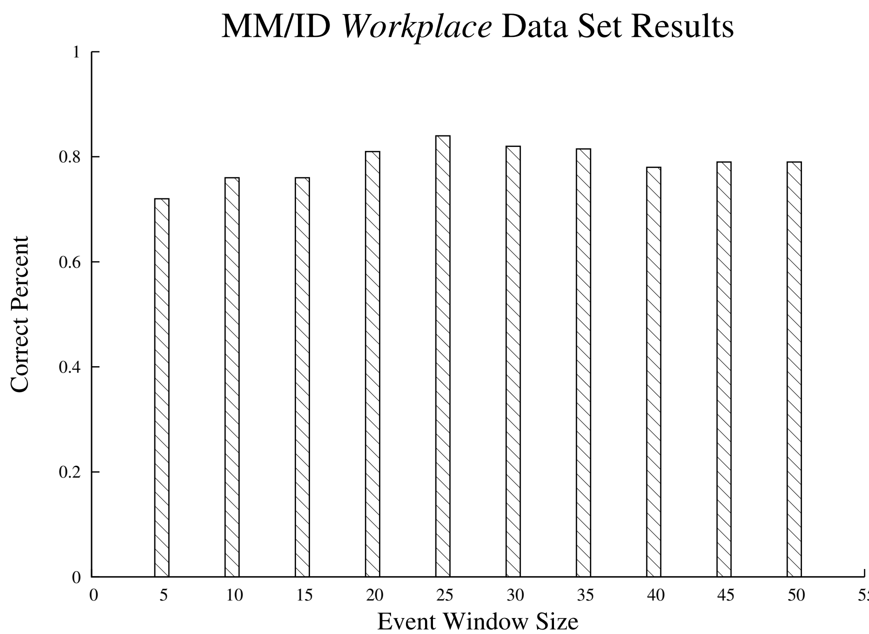


Figure 5.8: MM/ID *Workplace* data set overall accuracy.

the most suitable window size for a given data set.

The most direct comparison between the MM/ID and NB/ID algorithms is to compare accuracies using the "Plain" feature type (see Table 5.3) for the NB/ID classifier and the 25-event window size for the Markov Model. In this case the NB/ID algorithm results in 76% classification accuracy, in comparison with 84% classification accuracy resulting from using the MM/ID. The improved accuracy for the Markov Model is most likely due to the implicit spatial and temporal relationships that are encoded in the states and transition probabilities of the Markovian structure.

On the other hand, the Time Delta filtering steps are not applied to the test of the MM/ID algorithm. All of the events are kept for use with the Markov Model because even the shorter events thrown out by the Time Delta filtering provide valuable contextual information for the model. Removing these events renders the classifier unusable, as there is too little evidence to process from the *Workplace* data set.

***Workplace* with HMM/ID**

As published in [Crandall and Cook, 2009], the HMM/ID algorithm was evaluated with the *Workplace* data set. The resulting classifier performance was 92.4%. This approach outperforms the earlier Markov Model approach. However, while the model represents temporal sequence information, the length of individual events is not captured. This temporal information contains valuable insights into the type of activity or behavior in the data.

To address this issue, we add a time value to the feature list associated with each observable state. The time value corresponds to the amount of time, in milliseconds, that elapsed since the previous sensor event. Because the possible number of time values is inordinately large, we discretized the time values into three equal-size ranges. Testing the HMM/ID with time values on the *Workplace* data, we see an improved classification accuracy of 95.3%. This solution yields the best results of the Markov Model related classifiers presented here and represents an approach that should effectively scale to large numbers of residents.

Table 5.4: Summary of the best identification tools on the *Workplace* data set.

Classifier	*Notes*	*Accuracy*
NB/ID	Plain Feature	80%
	With Hour-of-Day Feature	96%
	Time Delta Filter & Hour-of-Day Feature	98%
MM/ID	Best Time Window	84%
HMM/ID	Simple	92.4%
	Discretized Time Values	95.3%

***Workplace* Results Summary**

The different algorithms introduced here approach the *Workplace* data set differently. They use frequency, length and locality of behavior to correctly attribute events to residents. While the NB/ID tool gets the highest overall accuracy with the Time Delta filtering, as shown in Table 5.4, it does rely on residents having sensors in spaces unique to them in order to operate so well. The HMM/ID tool demonstrates more flexibility by requiring much less data manipulation than the Time Delta approach and using all of the available data to make proper classifications. In the face of more residents and subtle behaviors, the HMM/ID shows a greater capability to handle the data directly from the sensor platform and still operate accurately.

5.4.2 B&B *Data Set Results*

The *B&B* data set involves two simultaneous residents inhabiting the *Kyoto* testbed. It is a relatively short data set at 5 days, but does have the benefit of being occupied nearly the full 120 hours of its duration. The different identification algorithms performed in notably different ways than when provided the *Workplace* data set. These differences are exposed by the interleaved resident tags and the cumulative evidence for an individual's identity effecting the behavior of the NB/ID and HMM/ID models.

B&B **Evaluation**

The NB/ID and HMM/ID classifiers were tested using 30-fold cross validation. Each classifier was trained on 29 out of 30 groups and tested on the remaining one. The results from all thirty permutations were averaged together for an overall accuracy, and their variance calculated for significance values. Additional statistics showing the behavior of the classifiers and the data sets were gathered for insight into the capabilities of the tools.

Rather than repeat all of our experiments performed on the *Workplace* data set, effort was concentrated on comparing the NB/ID classifier and the HMM/ID models for the *B&B* data set using parameter settings as described for the earlier experiments in Section 5.4.1. The results are shown in Figures 5.9 and 5.10, with the

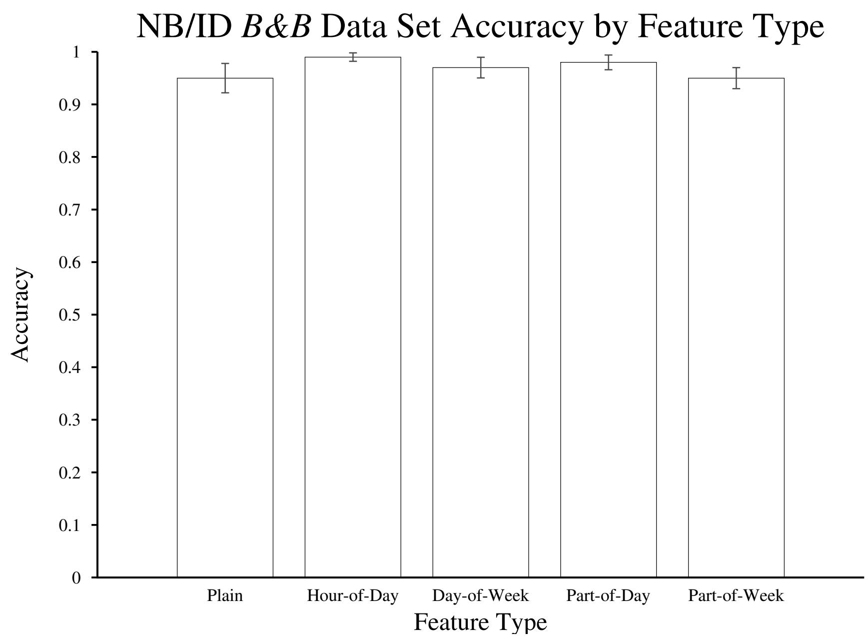


Figure 5.9: NB/ID accuracy on the B&B data set for various temporal features. The error bars show two standard deviations.

Table 5.5: NB/ID *B&B* data set results. The Hour-of-Day feature is significant over the Plain feature.

Feature Type	Accuracy	STDDEV (σ)
Plain	95%	1.4
Hour-of-Day	* **99%**	0.4
Day-of-Week	97%	1.0
Part-of-Day	98%	0.7
Part-of-Week	95%	1.0

Table 5.6: HMM/ID *B&B* data set results.

Feature Type	Accuracy	STDDEV (σ)
No Time Features	91.8%	8.3
Discretized Time Features	91.9%	8.4

source values laid out in Tables 5.5 and 5.6. As can be seen, both the NB/ID and HMM/ID achieve very high classification accuracies on this two-resident, parallel-activity data. The two algorithms tested performed statistically equally on this data set. We hypothesize that having only two classes for the naïve Bayes to choose from benefits it inordinately over the three residents of the *Workplace* data set.

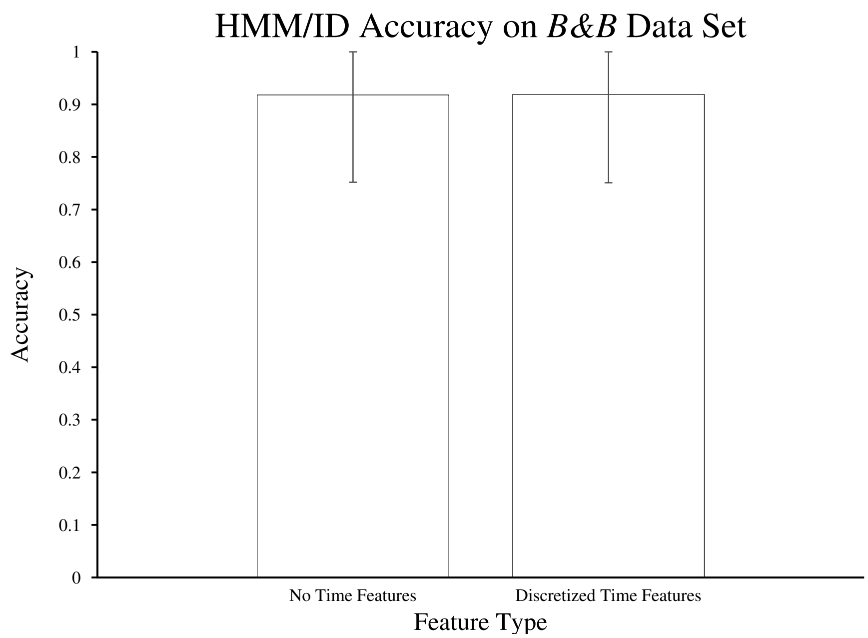


Figure 5.10: HMM/ID accuracy on the *B&B* data set leveraging both time and no time features. The error bars show two standard deviations.

The NB/ID classifier demonstrates a very similar behavior for both the *Workplace* and *B&B* data for the various temporal feature types. We found that once again using the Hour-of-Day gives the best results, and is a significant ($p < 0.05$) improvement over the Plain feature. Surprisingly, the inclusion of the discretized time values in the HMM/ID demonstrates no benefit for the the *B&B* data set as it does for *Workplace*. This provides evidence to support our hypothesis that both temporal and spatial information have different values for different environments and residents. Continued efforts to discover the most valuable combination of features for identifying individuals needs to be pursued.

B&B Summary

The ability of our models to perform well in this unscripted, full-time, multi-resident environment is encouraging. These kinds of classifiers should be able to provide better tools for discerning an individual's activity history, even in complex multi-resident environments.

5.4.3 TwoR *Data Set Results*

The *TwoR* data set provides the largest corpus of data of the three identification data sets. It has the most complex behaviors and social interactions as well. Like the *B&B* data set, the NB/ID and HMM/ID tools were evaluated for accuracy. Additionally, a more in-depth look at the behavior of the HMM/ID is discussed. Given the interleaved and social nature of the residents, the *TwoR* data exposes the various sources of error for the HMM/ID algorithm.

TwoR Evaluation

As with the evaluation of the classifiers with the *B&B* data set in Section 5.4.2, the classifiers were tested using 30-fold cross validation. Additionally, their results were compared to a Weighted Random algorithm as a base case. Each classifier was trained on 29 out of 30 groups and tested on the remaining one. The results from all 30 run permutations were averaged together for an overall accuracy, and their variance calculated for significance values. Additional statistics showing the behavior of the classifiers and the data sets were gathered for insights into the capabilities of the tools.

Both algorithms performed well on the *TwoR* and *B&B* data sets and were significantly ($p < 0.01$) better than a Weighted Random algorithm introduced as a base case for comparison. The overall accuracy of the algorithms are shown in Figure 5.11 and the numerical values are shown in Table 5.7. The HMM/ID performed slightly better than the NB/ID, though not significantly so.

Given the complexity of the data with multiple residents and no given structure to their behavior, the highly accurate results from both algorithms attest to their robustness. The NB/ID accuracy improved notably from the *Workplace* data set

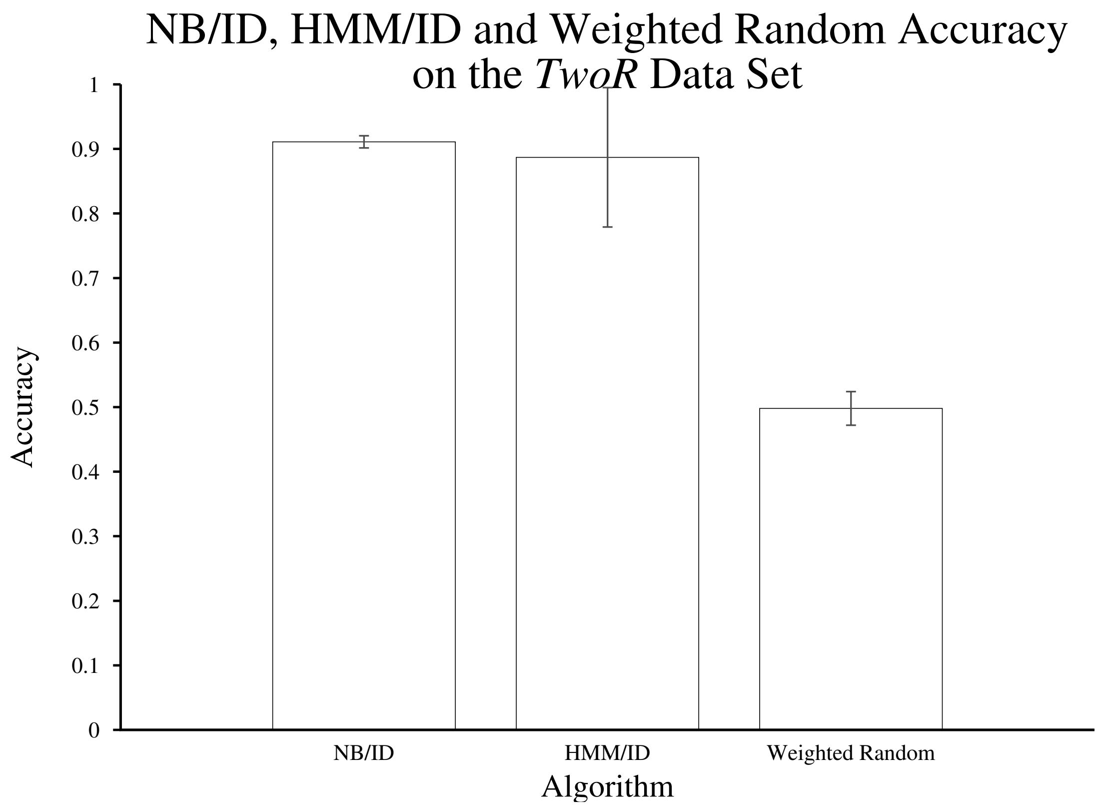


Figure 5.11: *TwoR* data set accuracy for the NB/ID, HMM/ID and Weighted Random algorithms. The error bars show two standard deviations.

Table 5.7: Overall Accuracies for both NB/ID and HMM/ID on the more complex interleaved resident data sets.

Data Set	*B&B*		*TwoR*	
Algorithm	Accuracy	STDDEV (σ)	Accuracy	STDDEV (σ)
NB/ID	95.7%	1.4	91.1%	0.5
HMM/ID	91.8%	8.3	88.7%	5.4
Weighted Random	53.3%	3.6	49.8%	1.3

results. This was likely due to the residents having more personal space and time when compared to the simpler laboratory setting used previously.

Overall, the HMM/ID results are very promising. The initial hypothesis that drawing on additional contextual information across a series of events would allow an algorithm to better differentiate between individuals seems to be supported by the overall accuracy results.

The behavior of the HMM/ID is more complex than the NB/ID when analyzing the actual pattern of classification. As the events arrive, it takes the HMM zero or more additional events to determine to whom the new events belong. For an

Table 5.8: Example HMM/ID transition behavior pattern.

Event Number	Annotated Class	Chosen Class	Result
1	R1	R1	SUCCESS
2	R1	R1	SUCCESS
3	R2	R1	FAIL
4	R2	R2	SUCCESS
5	R2	R2	SUCCESS

example of this behavior, Table 5.8 shows a small snippet of events as classified by the HMM/ID. The left column is the event number, the second represents then annotated resident value for the event, the third the algorithm determined, and the final column being the success or fail results for the given event. This snippet has a transition from R1 to R2 at event #3. The HMM delays until event #4 before it has enough evidence to change states and begins attributing events correctly. This situation, where the events change from one resident to another, has been termed a resident "transition" and is an important feature of HMM/ID algorithm behavior.

By the overall accuracy metric, this example is has a score of 4/5, or 80% accuracy. What is most interesting about this series is that the events arriving at the computer are initially from R1, then change to R2 at some point, but the HMM/ID algorithm takes extra events to properly transition as well. In contrast, the NB/ID algorithm takes every event in isolation, so there is no previous context to consider. With the HMM/ID algorithm, there is now a possibility of a transition window as the evidence that the new events are from a different person accumulates. The concern is that this transition window would significantly impact the effectiveness of the HMM/ID as a tool for identification.

To determine how much this transition error is effecting the HMM/ID, several statistics were gathered from the final tests. The first was the total number of occurrences in the event stream where the annotated resident value switches from one to another. This is an indication of the data complexity. If the number of transitions increases it indicates more simultaneous occupancy of the space, which can be more difficult for the HMM/ID to accurately classify.

The hypothesized inverse relationship between the rate of transitions in the data set and the final accuracy was not borne out by the results, as shown in Figure 5.12. The transition rate line was expected to trend upward, opposite the overall accuracy across the data sets used to test the classifier. Instead it is found to trend with the accuracy, with slopes of -0.038 and -0.046 respectively. On further inspection, it is not merely the number of transitions that effects the overall accuracy, but also

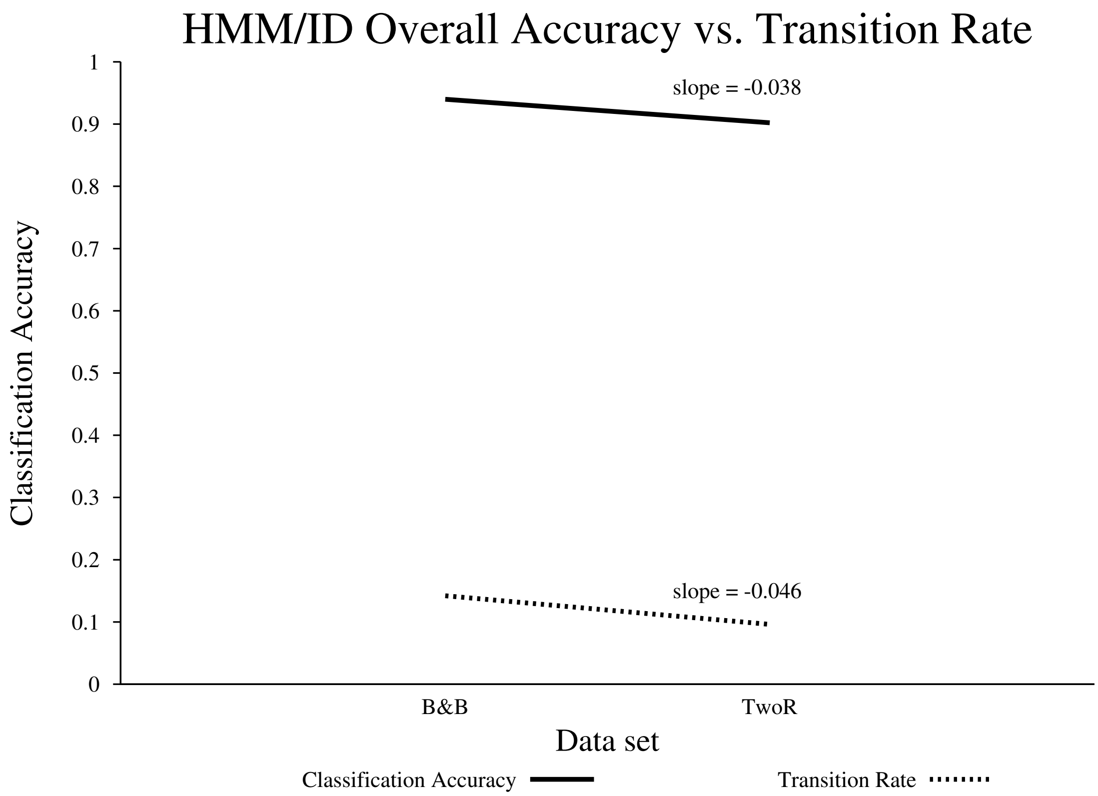


Figure 5.12: HMM/ID's overall accuracy for each data set, with the data sets comparative transition rate. The transition rate was expected to trend opposite to the classification accuracy instead of with it.

the location within the smart home of the residents during those transitions. If the entities' behaviors are physically close to one another, there is less evidence in the emission probabilities that the HMM should change its hidden state, and thereby transition correctly in its classifications.

Essentially, this is a similar algorithmic behavior to that exposed by the NB/ID's Delta Filtering. If the residents in the space are using shared spaces, the emission probabilities are lower and cause the HMM to be less reactive to transitions between residents. In the *B&B* data set, the residents spent notably less time sharing communal spaces than was found in the *TwoR* data set.

As a measure of how much the delay in transition impacts the behavior of the algorithm, some additional analysis about the length of the delay was gathered. The relevant data is the average number of events after a transition before the HMM properly changes to accurately classify the resident. To find this value, the results were processed for the length of the delay in the transition on each data set. Figures 5.13 and 5.14 show the total occurrences of delay lengths (zero or more), grouped by length until proper classification for each data set.

The first column in Figures 5.13 and 5.14 represents the count of transitions in each data set where the HMM changes state properly on the very first event after a transition. In these cases the HMM properly transitions from one resident to another with only the very first event as evidence. The rest of the columns are instances

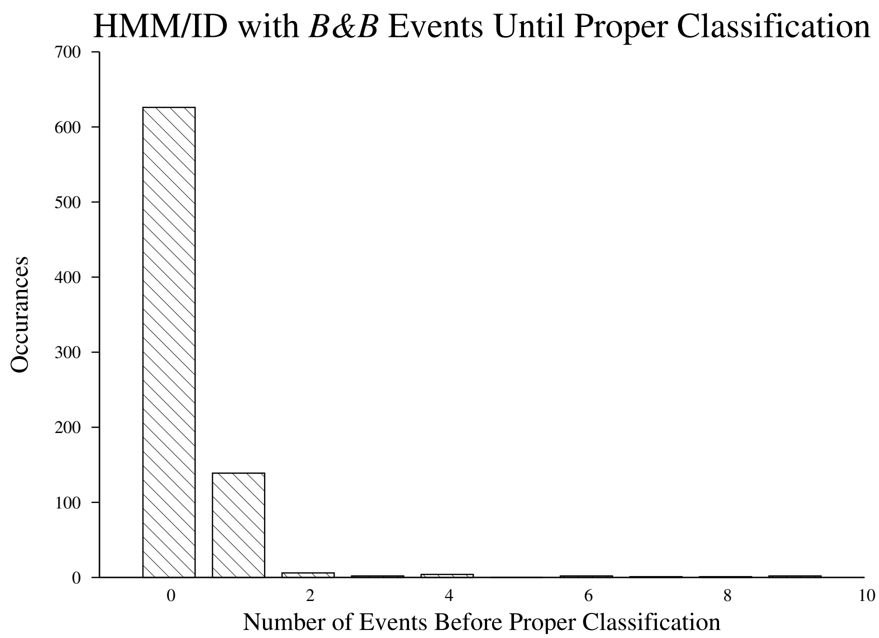


Figure 5.13: HMM delay lengths before proper classification after a transition between residents in the *B&B* data set.

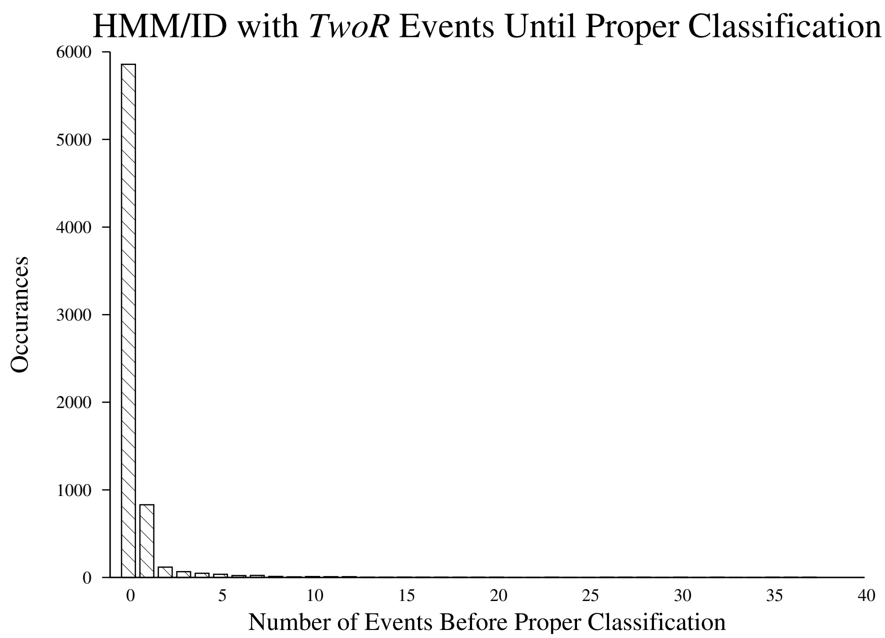


Figure 5.14: HMM delay lengths before proper classification after a transition between residents in the *TwoR* data set.

Table 5.9: HMM average transition delay length for both *B&B* and *TwoR* data sets. An average of zero would represent perfect transition accuracy.

Data Set	*Average delay* (in events)	*Standard Deviation* (σ)
B&B	0.19	0.80
TwoR	0.38	2.17

where there are one or more events improperly classified before the HMM transitions properly. This delay in the HMM after transitions in the data is a notable portion of the HMM's overall error.

Table 5.9 shows the average length of the delay in the HMM transition for each data set. An average of zero would mean that it has no delay whatsoever on the given data set, leading to perfect classification during transitions. The lower average delay for the *B&B* data set is consistent with the overall higher accuracy. This indicates that the HMM was able to use the evidence to accurately transition between residents based upon their behavior in the sensor space. The *TwoR* residents were notably more social than the *B&B* residents, and spent more time near one another in communal spaces during their stay in the testbed. Because they spent more time in close proximity, the resolution of the sensor network had more trouble providing

Table 5.10: HMM/ID non-transition error rates.

Data Set	*Error*
B&B	3.2%
TwoR	6.1%

evidence for the HMM to determine who was whom during the close interactions, causing the overall accuracy to suffer.

The other sign the the *TwoR* residents were more often interacting during the time of this data gathering is the longer lengths of the HMM's transition delay. With the *B&B* data set, there were very few instances where the HMM was not able to properly transition within one or two events. This indicates that the residents were most often physically separated in the testbed space. The very long delay lengths induced by the *TwoR* were observed to be when the two residents were performing activities like cooking or homework together. In those cases, the lack of physical separation meant that the HMM was unable to differentiate between the residents for quite some time.

Another source of error in classification occurs when the HMM outright chooses the incorrect class, but there was no actual transition to another resident. In this case the algorithm is truly confused, and this error type is more akin to the type of error in the NB/ID. The total error rate for this kind of mis-identification is summed up in Table 5.10. The higher rate for the *TwoR* data set indicates that these two individuals had more behavior that was similar to one another than the two people in the *B&B* data set, which again contributes to the lower overall accuracy on the *TwoR* set.

TwoR Results Summary

Containing a much larger selection of behaviors over a longer time than the previous data sets, the *TwoR* data set represents a valuable tool for evaluating behaviometric-based identification algorithms. The residents are closer in behavior to one another than those found in the *B&B* data set and it has the advantage of being an interleaved multi-resident data set, unlike *Workplace*. These additional hurdles provide opportunity for future identification algorithms to improve on those presented here.

5.5 Identification ADL Boosting

As a final demonstration of the usefulness of these identification algorithms, their ability to aid the performance of other types of smart environment tasks needed to be evaluated. Specifically, we apply the NB/ID classifier to the *TwoR* data set

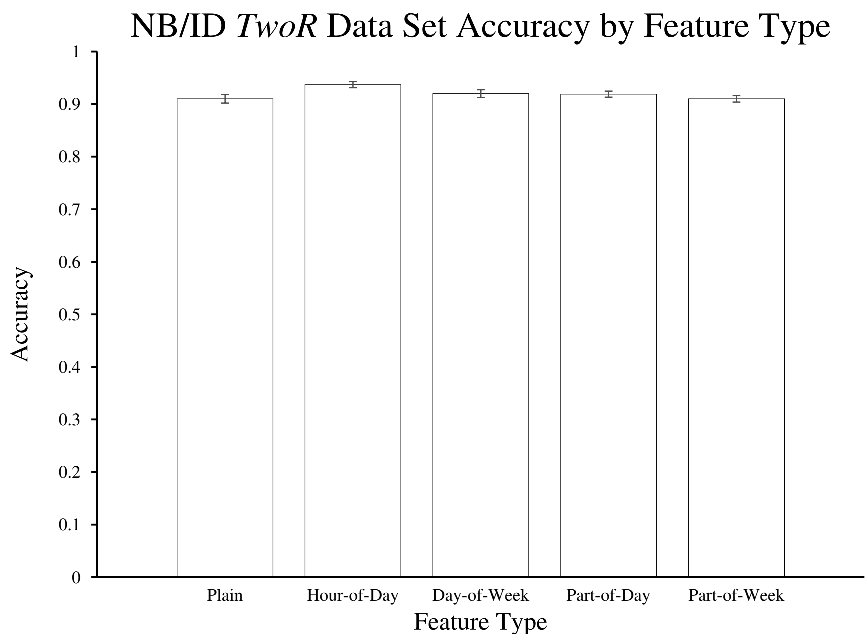


Figure 5.15: NB/ID *TwoR* data set accuracy for different temporal features. The error bars show two standard deviations.

to map sensor events to resident IDs. Given this additional feature, we then use a separate naïve Bayes classifier to identify which of 14 possible activities the residents are currently performing. We evaluate the performance of activity recognition with and without the learned resident identification to determine the extent to which the resident ID actually improves performance of our activity recognition algorithm.

Figure 5.15 summarizes the results of the NB/ID classifier as applied to the sensor events collected in the smart apartment as part of the *TwoR* data collection. The resident identification accuracy is very similar to the accuracy for the *B&B*

data set, peaking at 93.7% accuracy when the Hour-of-Day feature is used, which is significantly ($p < 0.05$) better than just the Plain feature.

Finally, we used a naïve Bayes classifier to perform activity recognition on this data set. We use the classifier to map a sequence of sensor events to one of 14 possible activity labels:

1. Resident1 going from bed to bathroom
2. Resident2 going from bed to bathroom
3. Resident1 preparing/eating breakfast
4. Resident1 preparing/eating breakfast
5. Watching TV (either resident)
6. Cleaning bathtub (either resident)
7. Resident1 working at the computer
8. Resident2 working at the computer
9. Resident1 sleeping
10. Resident2 sleeping
11. Preparing/eating lunch (either resident)

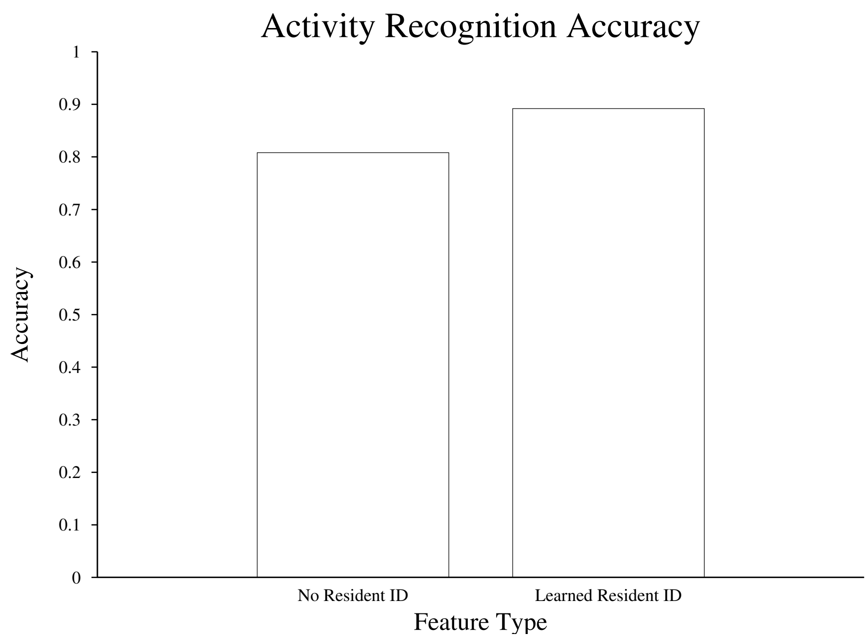


Figure 5.16: Activity recognition accuracy with and without learned resident IDs.

12. Preparing/eating dinner (either resident)

13. Cleaning (either resident)

14. Studying at the dining room table (either resident)

The naïve Bayes classifier initially achieved an accuracy of 80.8% on this data set. This is a good result as compared to other published ADL detection tools, especially given the number of activities that we need to discriminate and the fact that residents are performing activities in an interwoven and parallel fashion.

To determine how activity recognition can benefit from learned resident information, we next enhance the *TwoR* data set by adding an extra field to each sensor event containing the resident ID that is automatically generated by the NB/ID classifier. We test our activity recognition algorithm again on this enhanced data set, and this time achieve an accuracy of 89.2%. The results of these two experiments are graphed in Figure 5.16 and clearly demonstrate that learned resident labels enhance the accuracy of other smart environment tasks such as activity recognition.

5.6 Identification Summary

The three algorithms introduced and explored in this chapter demonstrate the ability of behaviometrics to algorithmically identify smart home residents. They each leverage different aspects of the smart home data and react differently to various quantities and behaviors of residents. They are all demonstrably better than random guesses and provide additional insights into the workings of the smart home system.

The HMM/ID algorithm has managed to reduce the kind of error that the NB/ID was generating from about 6.5% to 3.2% on the *B&B* data and from 10.7% to 6.1% on the *TwoR* data, but introduced an additional delay in identification in instances where the residents are in relative proximity to each other. The reduction in general confusion indicates that our original hypothesis about additional context being valuable to identification holds true. The HMM/ID is able to take into account a series of events to more accurately identify a given individual based on their behavior alone. These improvements in evidence will reduce both the general confusion and shorten the delay in events the HMM/ID algorithm needs to transition between individuals in multi-resident situations.

It was demonstrated by the NB/ID temporal features and the HMM/ID discretized time values, that results on all three data sets where incorporating additional features about the data, such as temporal length or time of occurrence for events, can improve the accuracy of identification. The HMM/ID tool also provides insights into the reasons for error and the behavior of the residents. With the NB/ID it was difficult to tell why it made a good or bad choice at run time. By analyzing the series of classifications by the HMM/ID next to a visualization of the data, the researchers could determine what behaviors were easy or difficult for the HMM/ID to classify. It was then possible to algorithmically detect the reasons for success or failure within the final results. The opaque nature of the NB/ID made similar deep analysis much more troublesome.

The approach of using simple, passive, low resolution sensing environments with the algorithms introduced in this work generated results similar to those using other identification strategies. Controlled facial recognition approaches can see accuracies in the mid to high 90's [Pentland and Choudhury, 2000, Zhao et al., 2003, Li and Jain, 2005], height recognition in doorways may be 95+% [Srinivasan et al., 2010] and footstep and stride recognition has shown results around 87% accurate [Pirttikangas et al., 2003, Rodríguez et al., 2008]. Even RFID-based systems have some error in determining the identity of the RFID tag in real world implementations This can lead to RFID accuracy rates of only 60-70% Fritz et al. [2010], though repeated readings will likely overcome a single erronous transmission. Depending upon the intended use and environment, these different approaches may have more or less utility for a given smart home installation. In the long run, some combination or available strategies will likely become the most successful behaviometric identification methods.

While this study did not have data sets with large numbers of residents, the classifiers are expected to scale in a variety of different ways. The HMM/ID tool should scale against the number of residents better than the NB/ID by exploiting the context available in the data to make fine differentiations between individuals. It also requires less data feature tuning to make high quality classifications in these complex environments, which is beneficial to its ability to be deployed in real-world situations.

Overall, the HMM/ID behavior was easier for the users of the tool to comprehend. This observation was corroborated by the lower general confusion in classification when there were no transitions between residents occurring. From experience and empirically, the HMM/ID was notably more consistent than the NB/ID when a single resident was present and had more comprehensible behavior even in complex situations.

By applying these kinds of tools to the smart home data and generating a resident ID feature, ADL detection is boosted in complex, real world environments. Any modeling tools that improve the ability of smart homes to be functional and usable are important. Using algorithmic approaches to detect identity is a necessity for large scale deployments of smart home technologies that cannot have wireless devices affixed to every resident for identification purposes. The tools introduced and evaluated in this thesis initiate inquiry into these issues for the smart home research community.

SUMMARY AND CONCLUSIONS

This work laid out and evaluated a number of algorithms to address open issues in the smart home domain. The smart home research community possesses a scarcity of approaches for handling multiple residents that do not utilize carried wireless devices or installing cameras. Given that residents often have trouble maintaining devices in the long term, and that there are significant privacy issues with video recording in private homes, there exists a need for innovative solutions to tracking and identification. The CASAS approach of using passive, low-profile and privacy protecting sensors to implement smart home technologies opens the door for addressing the multiple resident problem without wireless or video-based solutions.

This work introduced the GR/ED, BUG/ED, and PF/ED tracking algorithms to detect, localize, and track multiple residents throughout the smart environment. These tools leveraged data from the CASAS Technology Platform for attributing events to discrete individuals as they traversed the space. The resultant added information demonstrably boosts the accuracy of other CASAS tools, such as ADL detection. The tracking tools were significantly better than random guessing and improved a naïve Bayes-based ADL detection tool by a large margin. Unfortunately, the complete system has trouble detecting the current number of residents. This limitation needs to be investigated from both the hardware and software ends of the smart home system to improve the ability for the smart home to determine how many residents are currently in the environment. Overall though, with these tools on hand a smart home system is better equipped to handle multiple residents while still accurately modeling ADLs, energy efficiency and resident behaviors.

In the second part of this work, the NB/ID, MM/ID and HMM/ID algorithms were introduced to uniquely identify the residents in the smart home. These behaviometric tools rely on behavior alone to identify the current residents in the space. This identification information means that the smart home is better equipped to build personalized histories, do individually tailored preference prediction, and improve ADL detection capabilities.

The field of smart homes is growing rapidly. The cost and complexity of the technology is dropping to a price point that makes it feasible to deploy in private homes. With the rapidly aging populations across the world, techniques to inexpensively support aging in place care programs will become imperative. The tools introduced and evaluated here are just one part of the puzzle, but addressing multiple resident issues is a keystone of making these technologies ready for the real world.

Suggestions for Future Research

Dealing with the multiple resident problem is in its infancy. The works introduced here are examples of the various approaches that could be used to overcome this important issue, but they are not the end of the process. In general, Smart homes are so diverse in their construction and organization that there will be many challenges to finding definitive and generalized solutions.

Tracking is a well established field of robotics, but if the robot does not carry a wireless device then many of the established tracking solutions are not applicable. Such is the case with smart homes, so algorithms and sensors need to be developed that track the resident via environmental sensors. The Passive Infra-Red motion detector is a workhorse of the smart home field, but it provides too little information for many applications and has acute limitations for detailed resident localization. This became obvious as the tracking algorithms were unable to reliably determine the number of residents, though they could track with a high degree of accuracy. New techniques for providing more complex sensor readings from PIR sensors need to be explored. With high fidelity data from the sensors available, the algorithms for tracking and identification should be significantly more effective.

There is also a place to expand upon the body of devices that identify residents passively in a smart home. The latest works using height sensors in doorways are a good start, but more work using sensors around door frames as a "checkpoint" to identify a resident, then combining those results with a passive tracking solution will likely be a very good way to uniquely identify residents. Externalizing the process of determining entrances and exits to inform the tracking systems may be a better approach than an all in one algorithm. This combination of passive biometrics, behaviometrics and tracking would likely be more effective than solutions that run in isolation.

Smart homes provide a rich field of opportunities for research. The combination of people, technology and places makes for infinite combinations of situations to be addressed and evaluated.

APPENDIX A

DEFINITION OF TERMS

A Definition of Terms

Acronyms (Table A.1) and terms (Table A.2) used in this work of special note.

Table A.1: Thesis-specific acronym definitions

Acronym	Definition
CASAS	Center for Advanced Studies in Adaptive Systems
CLM	The CASAS Lightweight Middleware. The suite of tools and message definitions used to pass data between agents in the CASAS Environment [Kusznir and Cook, 2010].
CTP	The CASAS Technology Platform. The full set of devices and software used by the CASAS researchers to implement a smart home.
EECS	School of Electrical Engineering and Computer Science at Washington State University
WSU	Washington State University
XMPP	Extensible Messaging and Presence Protocol

Table A.2: Thesis-specific term definitions

Term	Definition
Activities of Daily Living	Activities of Daily Living (ADLs) is a term used in health care to refer to daily self-care activities within an individual's place of residence, in outdoor environments, or both.
ambient intelligence	Ambient intelligence (AmI) refers to electronic environments that are sensitive and responsive to the presence of people.
entity	Any person, place or thing that causes events within the smart home. Most commonly a resident, but could also be a pet, robot or device.
localization	The process of determining the current coordinates of a given object.
medical monitoring	The process of monitoring a resident to derive their current and historical physical or mental state.

Continued on next page

Table A.2 – continued from previous page

Term	Definition
resident	A human who is currently within the smart home space.
smart home	Any living space outfitted with sensors that tries to build models of the activities within the space. It may also have controllers to allow the results of the model building to directly feed back by altering the space in some way. For example, it may turn on a light as a resident enters a room.
testbed	A single smart home installation used to gather data and test the algorithms developed by the researchers.
trace	The series of events attributed to an entity. This is similar to a tracklet, but may be more ambiguous in the face of multiple entities in the space [Crandall and Cook, 2009].

Continued on next page

Table A.2 – continued from previous page

Term	Definition
tracklet	A tracklet is the estimate of a target state or a track that is equivalent to an estimate based upon only a few measurements [Drummond et al., 2003].
ubiquitous computing	Ubiquitous computing (ubicomp) is a post-desktop model of human and computer interaction in which information processing has been thoroughly integrated into everyday objects and activities.

APPENDIX B

HIDDEN MARKOV MODEL VITERBI ALGORITHM

CONCRETE EXAMPLE

B Hidden Markov Model Viterbi Algorithm Concrete Example

```
entities = ('Res0', 'Res1')

events = ('M001', 'M002', 'M003')

start_probability = {'Res0': 0.69, 'Res1': 0.31}

transition_probability = {

        'Res0' : {'Res0': 0.90, 'Res1': 0.10},

        'Res1' : {'Res0': 0.22, 'Res1': 0.77},

   }

emission_probability = {

        'Res0' : {'M001': 0.1, 'M002': 0.4, 'M003': 0.5},

        'Res1' : {'M001': 0.6, 'M002': 0.3, 'M003': 0.1},

   }


def determine_resident():

    return viterbi(events,

                   entities,

                   start_probability,

                   transition_probability,

                   emission_probability)
```

```
def viterbi(evs, entities, start_p, trans_p, emit_p):
    V = [{}]
    path = {}
    #** Initialize base cases (t == 0) **#
    for en in entities:
        V[0][en] = start_p[en] * emit_p[en][evs[0]]
        path[en] = [en]
    #** Run Viterbi for t > 0  **#
    for t in range(1,len(evs)):
        V.append({})
        n_path = {}
        for en in entities:
            (prob, state) =
                max([(V[t-1][en0] * trans_p[en0][en]
                  * emit_p[en][evs[t]], en0) for en0 in entities])
            V[t][en] = prob
            n_path[en] = path[state] + [en]
        path = n_path
    (prob, state) =
        max([(V[len(evs) - 1][en], en) for en in entities])
    return (prob, path[state])
```

BIBLIOGRAPHY

Michael D. Addlesee, Alan Jones, Finnbar Livesey, and Ferdinando Samaria. The ORL active floor [sensor system]. *Personal Communications, IEEE*, 4(5):35–41, October 1997. doi: 10.1109/98.626980.

Anand Agarawala, Saul Greenberg, and Geoffrey Ho. The context-aware pill bottle and medication monitor. Technical Report 2004-752-17, Department of Computer Science, University of Calgary, 2004.

Ryan Aipperspach, Elliot Cohen, and John Canny. Modeling human behavior from simple sensors in the home. In Kenneth Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley, editors, *Pervasive Computing*, volume 3968 of *Lecture Notes in Computer Science*, pages 337–348. Springer Berlin / Heidelberg, 2006. doi: 10.1007/11748625_21.

M. Javad Akhlaghinia, Ahmad Lotfi, Caroline Langensiepen, and Nasser Sherkat. Occupancy monitoring in intelligent environment through integrated wireless localizing agents. In *IEEE Symposium on Intelligent Agents*, IA '09, pages 70–76, April 2009. doi: 10.1109/IA.2009.4927502.

Tareq Ali Alhmiedat and Shuang-Hua Yang. A ZigBee-based mobile tracking system through wireless sensor networks. *International Journal of Advanced Mechatronic Systems*, 1(1):63–70, October 2008. doi: 10.1504/IJAMECHS.2008.020839.

Majd Alwan, Siddharth Dalal, Steve Kell, and Robin Felder. Derivation of basic human gait characteristics from floor vibrations. In *Summer Bioengineering Conference*, Key Biscayne, FL, 2003.

Sofía Aparicio, Javier Pérez, Ana M. Bernardos, and José R. Casar. A fusion method based on Bluetooth and WLAN technologies for indoor location. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 487–491, August 2008. doi: 10.1109/MFI.2008.4648042.

Sofía Aparicio, Javier Pérez, Paula Tarrío, Ana M. Bernardos, and José R. Casar. An indoor location method based on a fusion map using Bluetooth and WLAN technologies. In Juan Corchado, Sara Rodríguez, James Llinas, and José Molina, editors, *International Symposium on Distributed Computing and Artificial Intelligence*, volume 50 of *Advances in Soft Computing*, pages 702–710. Springer Berlin / Heidelberg, 2009. doi: 10.1007/978-3-540-85863-8_83.

Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the Annual Joint Conference* *of the IEEE Computer and Communications Societies*, volume 2, pages 775–784. IEEE CS Press, 2000. doi: 10.1109/INFCOM.2000.832252.

Mortaza S. Bargh and Robert de Groote. Indoor localization based on response rate of Bluetooth inquiries. In *Proceedings of the ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08, pages 49–54, New York, NY, USA, 2008. ACM. doi: 10.1145/1410012.1410024.

Andrew Barry, Benjamin Fisher, and Mark L. Chang. A long-duration study of user-trained 802.11 localization. In *Proceedings of the International Conference on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '09, pages 197–212, Berlin, Heidelberg, 2009. Springer-Verlag. doi: 10.1007/978-3-642-04385-7_14.

Rezaul Begg and Rafiul Hassan. Artificial neural networks in smart homes. In Juan Augusto and Chris Nugent, editors, *Designing Smart Homes*, volume 4008 of *Lecture Notes in Computer Science*, pages 146–164. Springer Berlin / Heidelberg, 2006. doi: 10.1007/11788485_9.

Chiraz BenAbdelkader, Ross Cutler, and Larry Davis. Person identification using automatic height and stride estimation. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 377–380, 2002. doi: 10.1109/ICPR.2002.1047474.

Chiraz BenAbdelkader, Ross Cutler, and Larry Davis. View-invariant estimation of height and stride for gait recognition. In Massimo Tistarelli, Josef Bigun, and Anil Jain, editors, *Biometric Authentication*, volume 2359 of *Lecture Notes in Computer Science*, pages 155–167. Springer Berlin / Heidelberg, 2006. doi: 10.1007/3-540-47917-1_16.

Alphonse Bertillon. *The Bertillon System of Identification*. The Werner Company, Chicago, IL, 1896.

Jan Blumenthal, Ralf Grossmann, Frank Golatowski, and Dirk Timmermann. Weighted centroid localization in ZigBee-based sensor networks. In *IEEE International Symposium on Intelligent Signal Processing*, WISP '07, pages 1–6, October 2007. doi: 10.1109/WISP.2007.4447528.

Philipp Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08, pages 55–60, New York, NY, USA, 2008. ACM. doi: 10.1145/1410012.1410025.

Philipp Bolliger, Kurt Partridge, Maurice Chu, and Marc Langheinrich. Improving location fingerprinting through motion detection and asynchronous interval labeling. In Tanzeem Choudhury, Aaron Quigley, Thomas Strang, and Koji Suginuma, editors, *Location and Context Awareness*, volume 5561 of *Lecture Notes* *in Computer Science*, pages 37–51. Springer Berlin / Heidelberg, 2009. doi: 10.1007/978-3-642-01721-6_3.

Oliver Brdiczka, Patrick Reignier, and James L. Crowley. Detecting individual activities from video in a smart home. In Bruno Apolloni, Robert Howlett, and Lakhmi Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4692 of *Lecture Notes in Computer Science*, pages 363–370, Berlin, Heidelberg, 2007. Springer Berlin / Heidelberg. doi: 10.1007/978-3-540-74819-9_45.

Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, and Steven Shafer. EasyLiving: Technologies for intelligent environments. In Peter Thomas and Hans W. Gellersen, editors, *Handheld and Ubiquitous Computing*, volume 1927 of *Lecture Notes in Computer Science*, pages 97–119. Springer Berlin / Heidelberg, 2000. doi: 10.1007/3-540-39959-3_2.

Raffaele Bruno and Franca Delmastro. Design and analysis of a Bluetooth-based indoor localization system. In *PersonalWireless Communications*, volume 2775 of *Lecture Notes in Computer Science*, pages 711–725. Springer Berlin / Heidelberg, 2003. doi: 10.1007/978-3-540-39867-7_66.

Robert N. Bucceri. *How to Automate Both New & Existing Homes*. Silent Servant, Inc., January 2004. ISBN 978-0-97000-573-1.

Michael Buettner and David Wetherall. An empirical study of UHF RFID performance. In *Proceedings of the ACM International Conference on Mobile Computing and Networking*, MobiCom '08, pages 223–234, New York, NY, USA, 2008. ACM. doi: 10.1145/1409944.1409970.

Michael Buettner, Richa Prasad, Matthai Philipose, and David Wetherall. Recognizing daily activities with RFID-based sensors. In *Proceedings of the International Conference on Ubiquitous Computing*, UbiComp '09, pages 51–60, New York, NY, USA, 2009. ACM. doi: 10.1145/1620545.1620553.

Terry Burke, Gaudenz Dolf, and A. J. Jeffreys. *DNA Fingerprinting: Approaches and Applications*. Birkhauser, July 1991. ISBN 978-0-81762-562-7.

John M. Butler. *Fundamentals of Forensic DNA Typing*. Academic Press, 2009. ISBN 978-0-12-374999-4.

WSU CASAS. CASAS shared smart home datasets repository. `http://casas.wsu.edu/datasets.html`, 2010.

Paul Castro, Patrick Chiu, Ted Kremenek, and Richard R. Muntz. A probabilistic room location service for wireless networked environments. In Gregory Abowd, Barry Brumitt, and Steven Shafer, editors, *Ubiquitous Computing*, volume 2201 of *Lecture Notes in Computer Science*, pages 18–34. Springer Berlin / Heidelberg, 2001. doi: 10.1007/3-540-45427-6_3.

Barry Cense, Nader A. Nassif, Teresa C. Chen, Mark C. Pierce, Seok-Hyun Yun, B. Hyle Park, Brett E. Bouma, Guillermo J. Tearney, and Johannes F. de Boer. Ultrahigh-resolution high-speed retinal imaging using spectral-domain optical coherence tomography. In *Optics Express*, volume 12. Optical Society of America, 2435–2447, 2004. doi: 10.1364/OPEX.12.002435.

Xiaoyong Chai and Qiang Yang. Reducing the calibration effort for probabilistic indoor location estimation. *IEEE Transactions on Mobile Computing*, 6(6):649–662, June 2007. doi: 10.1109/TMC.2007.1025.

Sergey Chemishkian. Building smart services for smart home. In *Proceedings of the IEEE International Workshop on Networked Appliances*, pages 215–224, 2002. doi: 10.1109/IWNA.2001.980856.

Chao Chen. Design of a child localization system on RFID and wireless sensor networks. *Journal of Sensors*, 2010, 2010. doi: 10.1155/2010/450392. Article ID 450392.

Chao Chen and Diane J. Cook. Novelty detection in human behavior through analysis of energy utilization. In *Human Behavior Recognition Technologies*. IGI Global, 2010.

Chao Chen, Barnan Das, and Diane Cook. Energy prediction based on resident's activity. In *Proceedings of the International Workshop on Knowledge Discovery from Sensor Data*, SensorKDD '10, 2010.

Datong Chen, Jie Yang, and Howard D. Wactlar. Towards automatic analysis of social interaction patterns in a nursing home environment from video. In *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '04, pages 283–290, New York, NY, USA, 2004. ACM. doi: 10.1145/1026711.1026757.

Mike Y. Chen, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, Jeffrey Hightower, Jeff Hughes, Anthony Lamarca, Fred Potter, Ian Smith, and Alex Varshavsky. Practical metropolitan-scale positioning for GSM phones. In Paul Dourish and Adrian Friday, editors, *In Proceedings of the International Conference on Ubiquitous Computing*, volume 4206 of *Lecture Notes in Computer Science*, pages 225–242. Springer Berlin / Heidelberg, 2006. doi: 10.1007/11853565_14.

Marshini Chetty, David Tran, and Rebecca E. Grinter. Getting to green: understanding resource consumption in the home. In *Proceedings of the International Conference on Ubiquitous Computing*, UbiComp '08, pages 242–251, New York, NY, USA, 2008. ACM. doi: 10.1145/1409635.1409668.

Kenneth C. Cheung, Stephen S. Intille, and Kent Larson. An inexpensive Bluetooth-based indoor positioning hack. Online, 2006. URL http://architecture.mit.edu/house_n/documents/CheungIntilleLarson2006.pdf. Accessed: 2010.10.22.

Jae Sung Choi, Hyun Lee, Ramez Elmasri, and Daniel W. Engels. Localization systems using passive UHF RFID. In *The International Joint Conference on INC, IMS and IDC*, NCM '09, pages 1727–1732, Los Alamitos, CA, USA, August 2009. IEEE Computer Society. doi: 10.1109/NCM.2009.198.

Simon A. Cole. *Suspect Identities: A History of Fingerprinting and Criminal Identification*. Harvard University Press, 2001. ISBN 978-0-67400-455-9.

Robert T. Collins, Ralph Gross, and Jianbo Shi. Silhouette-based human identification from body shape and gait. In *The Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 366–371, May 2002. doi: 10.1109/AFGR.2002.1004181.

Diane Cook. Health monitoring and assistance to support aging in place. *Journal of Universal Computer Science*, 12(1):15–29, 2006. doi: 10.3217/jucs-012-01-0015.

Diane Cook. Learning setting-generalized activity models for smart spaces. *IEEE Intelligent Systems*, 99:PrePrints, 2010. doi: 10.1109/MIS.2010.112. (to appear).

Diane Cook and Maureen Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of Information in Medicine*, 48(5):480–485, 2009. doi: 10.3414/ME0592.

Diane J. Cook, Michael Youngblood, Edwin O. Heierman III, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja. MavHome: an agent-based smart home. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, PerCom '03, pages 521–524, March 2003. doi: 10.1109/PERCOM.2003.1192783.

Diane J. Cook, Aaron S. Crandall, Geetika Singla, and Brian Thomas. Detection of social interaction in smart spaces. *Cybernetics and Systems: An International Journal*, 41(2):90–104, 2010. doi: 10.1080/01969720903584183.

Aaron S. Crandall and Diane J. Cook. Smart home resident detection and identification using simple sensors. In *Washington State University Academic Showcase*, 2008a.

Aaron S. Crandall and Diane J. Cook. Resident and caregiver: Handling multiple people in a smart care facility. In *AI in Eldercare: New Solutions to Old Problems*, pages 39–47, Menlo Park, California, USA, 2008b. AAAI Press. ISBN 978-1-57735-394-2.

Aaron S. Crandall and Diane J. Cook. Attributing events to individuals in multi-inhabitant environments. In *IET International Conference on Intelligent Environ-*

ments, IE '08, pages 1–8, Amsterdam, The Netherlands, 2008c. IOS Press. ISBN 978-0-86341-894-5.

Aaron S. Crandall and Diane J. Cook. Coping with multiple residents in a smart environment. *Journal of Ambient Intelligence and Smart Environments*, 1(4):323–334, 2009. doi: 10.3233/AIS-2009-0041.

Aaron S. Crandall and Diane J. Cook. Tracking systems for multiple smart home residents. In *Human Behavior Recognition Technologies*. IGI Global, 2010a.

Aaron S. Crandall and Diane J. Cook. Using a Hidden Markov Model for resident identification. In *Proceedings of the International Conference on Intelligent Environments*, IE '10, 2010b.

Aaron S. Crandall and Diane J. Cook. Learning activity models for multiple agents in a smart space. In Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto, editors, *Handbook of Ambient Intelligence and Smart Environments*, pages 751–769. Springer US, 2010c. doi: 10.1007/978-0-387-93808-0_28.

Aaron S. Crandall and Diane J. Cook. Bayesian updating for individual tracking in smart homes. In *Washington State University Academic Showcase*, 2010d.

Aaron S. Crandall, Diane J. Cook, Jim Kusznir, and Brian Thomas. CASAS project:

A comprehensive smart home research testbed. In *Washington State University Academic Showcase*, 2008.

Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte Carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2 of *ICRA '99*, pages 1322–1328, 1999. doi: 10.1109/ROBOT.1999.772544.

George Demiris, Marilyn J. Rantz, Myra A. Aud, Karen D. Marek, Harry W. Tyrer, Marjorie Skubic, and Ali A. Hussam. Older adults' attitudes towards and perceptions of smart home technologies: a pilot study. *Informatics for Health and Social Care*, 29:87–94, 2004. doi: 10.1080/14639230410001684387.

George Demiris, Brian K. Hensel, Marjorie Skubic, and Marilyn Rantz. Senior residents' perceived need of and preferences for smart home sensor technologies. In *International Journal of Technology Assessment in Health Care*, volume 24, pages 120–124. Cambridge University Press, 2008. doi: 10.1017/S0266462307080154.

Javier J. M. Diaz, Rodrigo de A. Maués, Rodrigo B. Soares, Eduardo F. Nakamura, and Carlos M. S. Figueiredo. Bluepass: An indoor Bluetooth-based localization system for mobile applications. In *IEEE Symposium on Computers and Communications*, ISCC '10, pages 778–783, June 2010. doi: 10.1109/ISCC.2010.5546506.

Oliver E. Drummond, W. Dale Blair, George C. Brown, Terry L. Ogle, Yaakov Bar-

Shalom, Robert L. Cooperman, and William H. Barker. Performance assessment and comparison of various tracklet methods for maneuvering targets. In Ivan Kadar, editor, *Signal Processing, Sensor Fusion, and Target Recognition XII*, volume 5096 of *SPIE '03*, pages 514–539, 2003. doi: 10.1117/12.503750.

Hicham El Zabadani. *Self sensing spaces*. Phd dissertation, University of Florida, Gainesville, FL, USA, 2006.

Eiman Elnahrawy, Xiaoyan Li, and Richard P. Martin. The limits of localization using signal strength: a comparative study. In *The Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, SECON '04, pages 406–414, October 2004. doi: 10.1109/SAHCN.2004.1381942.

Mbou Eyole-Monono, Robert Harle, and Andy Hopper. POISE: An inexpensive, low-power location sensor based on electrostatics. In *The Annual International Conference on Mobile and Ubiquitous Systems: Networking Services*, MobiQuitous '06, pages 1–3, July 2006. doi: 10.1109/MOBIQ.2006.340387.

Marcos Faundez-Zanuy. A door-opening system using a low-cost fingerprint scanner and a PC. *Aerospace and Electronic Systems Magazine, IEEE*, 19(8):23–26, August 2004. doi: 10.1109/MAES.2004.1346894.

Klaus Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless*

Smart Cards and Identification. Wiley & Sons LTD, 2^{nd} edition, July 2003. ISBN 978-0-47084-402-1.

Kenneth P. Fishkin, Matthai Philipose, and Adam Rea. Hands-on RFID: wireless wearables for detecting use of objects. In *Proceedings of the IEEE International Symposium on Wearable Computers*, ISWC '05, pages 38–41, October 2005. doi: 10.1109/ISWC.2005.25.

Malcolm J. Fisk. *Implications of Smart home Technologies*, chapter 5, pages 101–124. Inclusive Housing in an Ageing Society: Innovative Approaches. The Policy Press, 2001. ISBN 978-1-86134-263-8.

Sven Fleck and Wolfgang Strasser. Smart camera based monitoring system and its application to assisted living. *Proceedings of the IEEE*, 96(10):1698–1714, October 2008. doi: 10.1109/JPROC.2008.928765.

James Fogarty, Carolyn Au, and Scott E. Hudson. Sensing from the basement: a feasibility study of unobtrusive and low-cost home activity recognition. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 91–100, New York, NY, USA, 2006. ACM. doi: 10.1145/1166253.1166269.

Marc Fountain, Gail Brager, Edward Arens, Fred Bauman, and Charles Benton. Comport control for short-term occupancy. *Energy and Buildings*, 21(1):1–13, 1994. doi: 0.1016/0378-7788(94)90011-6.

Gerald Fritz, V. Beroulle, M.D. Nguyen, O. Aktouf, and I. Parissis. Read-Error-Rate evaluation for RFID system on-line testing. In *Proceedings of the IEEE International Mixed-Signals, Sensors and Systems Test Workshop*, IMS3TW '10, pages 1–6, June 2010. doi: 10.1109/IMS3TW.2010.5503016.

Masaaki Fukumoto and Mitsuru Shinagawa. CarpetLAN: A novel indoor wireless(-like) networking and positioning system. In Michael Beigl, Stephen Intille, Jun Rekimoto, and Hideyuki Tokuda, editors, *UbiComp 2005: Ubiquitous Computing*, volume 3660 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin / Heidelberg, 2005. doi: 10.1007/11551201_1.

Francis Galton. Personal identification and description. *Nature*, 38:173–201, 1888.

Hugo Gamboa and Ana Fred. An identity authentication system based on human computer interaction behaviour. In *In Proceedings of the International Workshop on Pattern Recognition in Information Systems*, PRIS '03, pages 46–55, 2003.

Hugo Gamboa, Ana Fred, and António A. Vieira. Prevention or identification of web intrusion via human computer interaction behaviour - a proposal. In *RTO SCI Symposium on Systems, Concepts and Integration (SCI) Methods and Technologies for Defence Against Terrorism*, October 2004.

Simson Garfinkel and Beth Rosenberg. *RFID: Applications, Security, and Privacy.* Addison-Wesley Professional, 2005. ISBN 978-0-32129-096-0.

Simson L. Garfinkel, Ari Juels, and Ravi Pappu. RFID privacy: an overview of problems and proposed solutions. *Security Privacy, IEEE*, 3(3):34–43, May 2005. doi: 10.1109/MSP.2005.78.

Sylvia Gaul and Martina Ziefle. Smart home technologies: Insights into generation-specific acceptance motives. In Andreas Holzinger and Klaus Miesenberger, editors, *HCI and Usability for e-Inclusion*, volume 5889 of *Lecture Notes in Computer Science*, pages 312–332. Springer Berlin / Heidelberg, 2009. doi: 10.1007/978-3-642-10308-7_22.

Weina Ge and Robert T. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *British Machine Vision Conference*, BMVC '08, 2008.

Bill Glover and Himanshu Bhatt. *RFID Essentials*. Theory in Practice. O'Reilly Media, 1^{st} edition, January 2006. ISBN 978-0-59600-944-1.

Michael A. Goodrich, Bryan S. Morse, Damon Gerhardt, Joseph L. Cooper, Morgan Quigley, Julie A. Adams, and Curtis Humphrey. Supporting wilderness search and rescue using a camera-equipped mini UAV. *Journal of Field Robotics*, 25:89–110, 2008. doi: 10.1002/rob.20226.

Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki. Practical robust localization over large-scale 802.11

wireless networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 70–84, 2004. doi: 10.1145/1023720.1023728.

Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Proceedings of the Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, pages 59–68, New York, NY, USA, August 1999. ACM. doi: 10.1145/313451.313476.

Edwin O. Heierman III and Diane J. Cook. Improving home automation by discovering regularly occurring device usage patterns. In *The IEEE International Conference on Data Mining*, ICDM '03, pages 537–540, November 2003. doi: 10.1109/ICDM.2003.1250971.

Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen. The Gator Tech Smart House: a programmable pervasive space. *Computer*, 38(3):50–60, March 2005. doi: 10.1109/MC.2005.107.

Jeffrey Hightower and Gaetano Borriello. Location sensing techniques. Technical report, University of Washington, Department of Computer Science and Engineering, Seattle, WA, USA, 2001a. UW-CSE-01-07-01.

Jeffrey Hightower and Gaetano Borriello. A survey and taxonomy of location systems for ubiquitous computing. Technical report, University of Washington, Department of Computer Science and Engineering, Seattle, WA, USA, August 2001b. UW-CSE-01-08-03.

Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, August 2001c. doi: 10.1109/2.940014.

Jeffrey Hightower and Gaetano Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In Nigel Davies, Elizabeth Mynatt, and Itiro Siio, editors, *Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*, pages 88–106. Springer Berlin / Heidelberg, 2004. doi: 10.1007/978-3-540-30119-6_6.

Yuri Ivanov, Alexander Sorokin, Christopher Wren, and Ishwinder Kaur. Tracking people in mixed modality systems. In Chang Wen Chen, Dan Schonfeld, and Jiebo Luo, editors, *Visual Communications and Image Processing*, volume 6508 of *VCIP '07*, San Jose, CA, USA, 2007a. SPIE. doi: 10.1117/12.714078.

Yuri Ivanov, Christopher Wren, Alexander Sorokin, and Ishwinder Kaur. Visualizing the history of living spaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1153–1160, November 2007b. doi: 10.1109/TVCG.2007.70621.

Philippe Jacquet, Tonia Papakonstantinou, and Georgios Rodolakis. Localization and path tracking using the IEEE 802.11 infrastructure. Technical report, HIPERCOM (INRIA Rocquencourt), 2008. inria-00271185.

Anil K. Jain, Salil Prabhakar, and Sharath Pankanti. Twin test: On discriminability of fingerprints. In Josef Bigun and Fabrizio Smeraldi, editors, *Audio- and Video-Based Biometric Person Authentication*, volume 2091 of *Lecture Notes in Computer Science*, pages 211–217. Springer Berlin / Heidelberg, 2001. doi: 10.1007/3-540-45344-X_30.

Anil K. Jain, Sarat C. Dass, and Nandakumar Karthik. Can soft biometric traits assist user recognition? In Anil K. Jain and Nalini K. Ratha, editors, *Biometric Technology for Human Identification*, volume 5404, pages 561–572, Orlando, FL, USA, August 2004a. doi: 10.1117/12.542890.

Anil K. Jain, Karthik Nandakumar, Xiaoguang Lu, and Unsang Park. Integrating faces, fingerprints, and soft biometric traits for user recognition. In Davide Maltoni and Anil K. Jain, editors, *ECCV Workshop BioAW'04*, volume 3087 of *Lecture Notes in Computer Science*, pages 259–269. Springer Berlin / Heidelberg, 2004b.

Anil K. Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1): 4–20, January 2004c. doi: 10.1109/TCSVT.2003.818349.

Vikramaditya Jakkula and Diane J. Cook. Anomaly detection using temporal data mining in a smart home environment. *Methods of Information in Medicine*, 47(1): 70–75, 2008.

Vikramaditya R. Jakkula, Aaron S. Crandall, and Diane J. Cook. Enhancing anomaly detection using temporal pattern discovery. In Achilles D. Kameas, Victor Callagan, Hani Hagras, Michael Weber, and Wolfgang Minker, editors, *Advanced Intelligent Environments*, pages 175–194. Springer US, 2009. doi: 10.1007/978-0-387-76485-6_8.

Christopher James Jenkins. *The Weakly Identifying System for Doorway Monitoring*. Dissertation, Duke University, United States – North Carolina, September 2007. Retrieved February 1, 2011, from Dissertations & Theses: A&I.(Publication No. AAT 3255593).

Jam Jenkins. Weakly identifying system for doorway monitoring. In *Duke Fontiers Poster Session*, May 2006.

Jam Jenkins and Carla Ellis. Using ground reaction forces from gait analysis: Body mass as a weak biometric. In Anthony LaMarca, Marc Langheinrich, and Khai Truong, editors, *Pervasive Computing*, volume 4480 of *Lecture Notes in Computer Science*, pages 251–267. Springer Berlin / Heidelberg, 2007. doi: 10.1007/978-3-540-72037-9_15.

Markus Jevring, Robert de Groote, and Cristian Hesselman. Dynamic optimization of Bluetooth networks for indoor localization. In *The International Workshop on Automated and Autonomous Sensor Networks*, CSTST '08, pages 663–668, New York, NY, USA, October 2008. ACM. doi: 10.1145/1456223.1456357.

Youssef Kaddoura, Jeff King, and Sumi Helal. Cost-precision tradeoffs in unencumbered floor-based indoor location tracking. In S. Giroux and H. Pigot, editors, *International Conference On Smart homes and health Telematic*, ICOST '05, pages 75–82, Sherbrooke, Québec, Canada, July 2005. IOS Press.

Nathan Keyfitz and Wilhelm Flieger. *World Population Growth and Aging: Demographic Trends in the Late Twentieth Century*. University Of Chicago Press, 2^{nd} edition, 1991. ISBN 978-0-22643-237-3.

Zia Khan, Tucker Balch, and Frank Dellaert. Efficient particle filter-based tracking of multiple interacting targets using an MRF-based motion model. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 1 of *IROS '03*, pages 254–259, October 2003. doi: 10.1109/IROS.2003.1250637.

Jung Soo Kim, Young Joon Chee, Ju Wan Park, Jin Wook Choi, and Kwang Suk Park. A new approach for non-intrusive monitoring of blood pressure on a toilet seat. In *Physiological Measurement*, volume 27, pages 203–211, February 2006. doi: 10.1088/0967-3334/27/2/010.

Thomas King, Stephan Kopf, Thomas Haenselmann, Christian Lubberger, and Wolfgang Effelsberg. COMPASS: A probabilistic indoor positioning system based on 802.11 and digital compasses. In *Proceedings of the International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, WiNTECH '06, pages 34–40, New York, NY, USA, 2006. ACM. doi: 10.1145/1160987.1160995.

Predrag Klasnja, Sunny Consolvo, Tanzeem Choudhury, Richard Beckwith, and Jeffrey Hightower. Exploring privacy concerns about personal sensing. In *Proceedings of the International Conference on Pervasive Computing*, PerCom '09, pages 176–183, Berlin, Heidelberg, 2009. Springer-Verlag. doi: 10.1007/978-3-642-01516-8_13.

Murali Kodialam, Thyaga Nandagopal, and Wing Cheong Lau. Anonymous tracking using RFID tags. In *The IEEE International Conference on Computer Communications*, INFOCOM '07, pages 1217–1225, May 2007. doi: 10.1109/INFCOM.2007.145.

Moritz Köhler, Shwetak Patel, Jay Summet, Erich Stuntebeck, and Gregory Abowd. TrackSense: Infrastructure free precise indoor positioning using projected patterns. In Anthony LaMarca, Marc Langheinrich, and Khai Truong, editors, *Pervasive Computing*, volume 4480 of *Lecture Notes in Computer Science*, pages 334–350. Springer Berlin / Heidelberg, 2007. doi: 10.1007/978-3-540-72037-9_20.

John Krumm and John Platt. Minimizing calibration effort for an indoor 802.11 device location measurement system. Technical report, Microsoft Research, Redmond, WA, November 2003. MSR-TR-2003-82.

John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-camera multi-person tracking for easy living. In *Proceedings of the IEEE International Workshop on Visual Surveillance*, VS '00, pages 3–10, Washington, DC, USA, July 2000. IEEE Press. ISBN 978-0-76950-698-2.

Jim Kusznir. CLM as a smart home middleware. Master's thesis, Washington State University, 2009.

Jim Kusznir and Diane Cook. Designing lightweight software architectures for smart environments. In *Proceedings of the International Conference on Intelligent Environments*, IE '10, 2010.

Andrew M. Ladd, Kostas E. Bekris, Guillaume Marceau, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki. Using wireless ethernet for localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1 of *IROS '02*, pages 402–408, Lausanne, Switzerland, September 2002a. doi: 10.1109/IRDS.2002.1041423.

Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Lydia E. Kavraki, and Dan S. Wallace. Robotics-based location sensing using wireless ethernet. In *Proceedings of the Annual International Conference on Mobile Computing and Network-*

ing, MobiCom '02, pages 227–238, New York, NY, USA, 2002b. ACM. doi: 10.1145/570645.570674.

Andrew M. Ladd, Kostas E. Bekris, Algis P. Rudys, Dan S. Wallach, and Lydia E. Kavraki. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation*, 20(3):555–559, June 2004. doi: 10.1109/TRA.2004.824948.

Anthony Lamarca, Jeffery Hightower, Ian Smith, and Sunny Consolvo. Self-mapping in 802.11 location systems. In Michael Beigl, Stephen Intille, Jun Rekimoto, and Hideyuki Tokuda, editors, *Proceedings of the International Conference on Ubiquitous Computing*, volume 3660 of *Lecture Notes in Computer Science*, pages 87–104. Springer Berlin / Heidelberg, 2005. doi: 10.1007/11551201_6.

Frederic Lassabe, Philippe Canalda, Pascal Chatonnay, François Spies, and Damien Charlet. Refining WiFi indoor positioning renders pertinent deploying location-based multimedia guide. In *The International Conference on Advanced Information Networking and Applications*, volume 2 of *AINA '06*, pages 126–132, Los Alamitos, CA, USA, April 2006. IEEE Computer Society. doi: 10.1109/AINA.2006.284.

Frederic Lassabe, Philippe Canalda, Pascal Chatonnay, and François Spies. Indoor Wi-Fi positioning: techniques and systems. *Annals of Telecommunications*, 64(9): 651–664, 2009. doi: 10.1007/s12243-009-0122-1.

Hyangjin Lee and Jeeyeon Kim. Privacy threats and issues in mobile RFID. In *Proceedings of the International Conference on Availability, Reliability and Security*, ARES '06, pages 1–5, April 2006. doi: 10.1109/ARES.2006.96.

Julia Letchner, Dieter Fox, and Anthony LaMarca. Large-scale localization from wireless signal strength. In *Proceedings of the National Conference on Artificial Intelligence*, volume 1 of *AAAI '05*, pages 15–20. AAAI Press, 2005. ISBN 1-57735-236-x.

Stan Z. Li and Anil K. Jain, editors. *Handbook of Face Recognition*. Springer, 1^{st} edition, March 2005. ISBN 978-0-38740-595-7.

Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: HybridBoosted multi-target tracker for crowded scene. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 2953–2960, June 2009. doi: 10.1109/CVPR.2009.5206735.

Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition using relational markov networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI '05, pages 773–778, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

Vit Libal, Bhuvana Ramabhadran, Nadia Mana, Fabio Pianesi, Paul Chippendale, Oswald Lanz, and Gerasimos Potamianos. Multimodal classification of activities of daily living inside smart homes. In Sigeru Omatu, Miguel Rocha, José Bravo, Florentino Fernández, Emilio Corchado, Andrés Bustillo, and Juan Corchado, editors, *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, volume 5518 of *Lecture Notes in Computer Science*, pages 687–694. Springer Berlin / Heidelberg, 2009. doi: 10.1007/978-3-642-02481-8_103.

Marc Lihan, Takeshi Tsuchiya, and Keiichi Koyanagi. Orientation-aware indoor localization path loss prediction model for wireless sensor networks. In Makoto Takizawa, Leonard Barolli, and Tomoya Enokido, editors, *Network-Based Information Systems*, volume 5186 of *Lecture Notes in Computer Science*, pages 169–178. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-85693-1_19.

Holger Linde. *On aspects of indoor localization.* PhD thesis, Universität Dortmund, August 2006. URL `http://hdl.handle.net/2003/22854`.

Charalampos Liolios, Charalampos Doukas, George Fourlas, and Ilias Maglogiannis. An overview of body sensor networks in enabling pervasive healthcare and assistive environments. In *Proceedings of the International Conference on Pervasive Technologies Related to Assistive Environments*, number 43 in PETRA '10, pages 1–10, New York, NY, USA, 2010. ACM. doi: 10.1145/1839294.1839346.

Jia Liu, Xiaofeng Tong, Wenlong Li, Tao Wang, Yimin Zhang, and Hongqi Wang. Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, 30(2):103–113, January 2009. doi: 10.1016/j.patrec.2008.02.011.

Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen Intille. A long-term evaluation of sensing modalities for activity recognition. In *Proceedings of the International Conference on Ubiquitous Computing*, UbiComp '07, pages 483–500, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-74852-6.

Konrad Lorincz and Matt Welsh. MoteTrack: a robust, decentralized approach to RF-based location tracking. *Personal and Ubiquitous Computing*, 11:489–503, 2007. doi: 10.1007/s00779-006-0095-2.

Karen Dorman Marek and Marilyn J. Rantz. Aging in place: A new model for long-term care. *Nursing Administration Quarterly*, 24(3):1–11, 2000.

Nobuyuki Matsushita, Shigeru Tajima, Yuji Ayatsuka, and Jun Rekimoto. Wearable key: device for personalizing nearby environment. In *The International Symposium on Wearable Computers*, ISWC '00, pages 119–126, 2000. doi: 10.1109/ISWC.2000.888473.

Uwe Maurer, Asim Smailagic, Daniel P. Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, BSN '04, pages 99–102, Washington, DC, USA, April 2004. IEEE Computer Society. doi: 10.1109/BSN.2006.6.

Vivek Menon, Bharat Jayaraman, and Venu Govindaraju. Biometrics driven smart environments: Abstract framework and evaluation. In Frode Sandnes, Yan Zhang, Chunming Rong, Laurence Yang, and Jianhua Ma, editors, *Ubiquitous Intelligence and Computing*, volume 5061 of *Lecture Notes in Computer Science*, pages 75–89. Springer Berlin / Heidelberg, 2010. doi: 10.1007/978-3-540-69293-5_8.

Gordon Meyer. *Smart Home Hacks: Tips & Tools for Automating Your House.* O'Reilly Media, Inc, 2005. ISBN 978-0-59600-722-5.

Taketoshi Mori, Yoshiko Suemasu, Hiroshi Noguchi, and Tomomasa Sato. Multiple people tracking by integrating distributed floor pressure sensors and RFID system. In *The IEEE International Conference on Systems, Man and Cybernetics*, volume 6 of *SMC '04*, pages 5271–5278, October 2004. doi: 10.1109/ICSMC.2004.1401031.

Robert Moskovitch, Clint Feher, Arik Messerman, Niklas Kirschnick, Tarik Mustafic, Ahmet Camtepe, Bernhard Löhlein, Ulrich Heister, Sebastian Möller, Lior Rokach, and Yuval Elovici. Identity theft, computers and behavioral biometrics. In *Proceedings of the IEEE International Conference on Intelligence and Security Infor-*

matics, ISI '09, pages 155–160, Piscataway, NJ, USA, June 2009. IEEE Press. doi: 10.1109/ISI.2009.5137288.

Michael C. Mozer. The neural network house: An environment that adapts to its inhabitants. In *Proceedings of the AAAI Spring Symposium on Intelligent Environments*, pages 110–114, Menlo Park, CA, 1998. AAAI Press.

Takuya Murakita, Tetsushi Ikeda, and Hiroshi Ishiguro. Human tracking using floor sensors based on the Markov Chain Monte Carlo method. In *Proceedings of the International Conference on Pattern Recognition*, volume 4 of *ICPR '04*, pages 917–920, August 2004. doi: 10.1109/ICPR.2004.1333922.

Elizabeth D. Mynatt, Anne-Sophie Melenhorst, Arthur D. Fisk, and Wendy A. Rogers. Aware technologies for aging in place: understanding user needs and attitudes. *Pervasive Computing*, 3(2):36–41, April 2004. doi: 10.1109/MPRV.2004.1316816.

Ernesto Navarro-Alvarez and Mario Siller. A node localization scheme for ZigBee-based sensor networks. In *The IEEE International Conference on Systems, Man and Cybernetics*, SMC '09, pages 728–733, October 2009. doi: 10.1109/ICSMC.2009.5346711.

Mordechai Nisenson, Ido Yariv, Ran El-Yaniv, and Ron Meir. Towards behaviometric security systems: Learning to identify a typist. In Nada Lavrac, Dragan Gamberger, Ljupco Todorovski, and Hendrik Blockeel, editors, *Knowledge Discovery in Databases*, volume 2838 of *Lecture Notes in Computer Science*, pages 363–374. Springer Berlin / Heidelberg, 2003. doi: 10.1007/978-3-540-39804-2_33.

Yoshifumi Nishida, Toshio Hori, Shin'ichi Murakami, and Hiroshi Mizoguchi. Minimally privacy-violative system for locating human by ultrasonic radar embedded on ceiling. In *The IEEE International Conference on Systems, Man and Cybernetics*, volume 2 of *SMC '04*, pages 1549–1554, October 2004. doi: 10.1109/ICSMC.2004.1399852.

OECD. Health at a glance 2009. *OECD Indicators*, December 2009. doi: 10.1787/health_glance-2009-en.

Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. RFID privacy issues and technical challenges. *Communications of the ACM*, 48(9):66–71, September 2005. doi: 10.1145/1081992.1082022.

Robert J. Orr and Gregory D. Abowd. The smart floor: a mechanism for natural user identification and tracking. In *The ACM Conference on Human Factors in Computing Systems' Extended Abstracts on Human Factors in Computing Systems*, CHI '00, pages 275–276, New York, NY, USA, 2000. ACM. doi: 10.1145/633292.633453.

Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal de Lara. Accurate GSM indoor localization. In Michael Beigl, Stephen Intille, Jun Rekimoto, and Hideyuki Tokuda, editors, *Proceedings of the International Conference on Ubiquitous Computing*, volume 3660 of *Lecture Notes in Computer Science*, pages 141–158. Springer Berlin / Heidelberg, 2005. doi: 10.1007/11551201_9.

Technica Pacifica. *Easy X10 Projects For Creating a Smart Home*. Indy-tech Publishing, April 2005. ISBN 978-0-79061-306-2.

Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. Growing an organic indoor location system. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 271–284, New York, NY, USA, 2010. ACM. doi: 10.1145/1814433.1814461.

Alex Pentland and Tanzeem Choudhury. Face recognition for smart environments. *Computer*, 33(2):50–55, Feburary 2000. doi: 10.1109/2.820039.

Quoc-Cuong Pham, Yoann Dhome, Laetitia Gond, and Patrick Sayd. Video monitoring of vulnerable people in home environment. In Sumi Helal, Simanta Mitra, Johnny Wong, Carl Chang, and Mounir Mokhtari, editors, *Smart Homes and Health Telematics*, volume 5120 of *Lecture Notes in Computer Science*, pages 90–98. Springer Berlin / Heidelberg, 2008. doi: 10.1007/978-3-540-69916-3_11.

Matthai Philipose, Kenneth P. Fishkin, Mike Perkowitz, Donald J. Patterson, Dieter

Fox, Henry Kautz, and Dirk Hähnel. Inferring activities from interactions with objects. *Pervasive Computing, IEEE*, 3(4):50–57, October–December 2004. doi: 10.1109/MPRV.2004.7.

P. Jonathon Phillips, Alvin Martin, C.L. Wilson, and Mark Przybocki. An introduction evaluating biometric systems. *Computer*, 33(2):56–63, February 2000. doi: 10.1109/2.820040.

Susanna Pirttikangas, Jaakko Suutala, Jukka Riekki, and Juha Röning. Footstep identification from pressure signals using Hidden Markov Models. In *Proceedings of the Finnish Signal Processing Symposium*, FINSIG '03, pages 124–128, 2003.

Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proceedings of the Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 32–43, August 2000. doi: 10.1145/345910.345917.

Youcef Rahal, Philippe Mabilleau, and Hélène Pigot. Bayesian filtering and anonymous sensors for localization in a smart home. In *The International Conference on Advanced Information Networking and Applications Workshops*, volume 2 of *AINA '07*, pages 793–797, Los Alamitos, CA, USA, May 2007. IEEE Computer Society. doi: 10.1109/AINAW.2007.108.

Parisa Rashidi and Diane J. Cook. Adapting to resident preferences in smart environments. In *Proceedings of the AAAI Workshop on Advances in Preference Handling*, pages 78–84, 2008.

Parisa Rashidi and Diane J. Cook. Transferring learned activities in smart environments. In *Proceedings of the International Conference on Intelligent Environments*, volume 2 of *IE '09*, pages 185–192, Amsterdam, The Netherlands, 2009a. IOS Press. doi: 10.3233/978-1-60750-034-6-185.

Parisa Rashidi and Diane J. Cook. Keeping the resident in the loop: adapting the smart home to the user. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(5):949–959, September 2009b. doi: 10.1109/TSMCA.2009.2025137.

Parisa Rashidi and Diane J. Cook. Multi home transfer learning for resident activity discovery and recognition. In *Proceedings of the International Workshop on Knowledge Discovery from Sensor Data*, SensorKDD '10, 2010a. (to appear).

Parisa Rashidi and Diane J. Cook. Home to home transfer learning. In *Proceedings of the AAAI Plan, Activity, and Intent Recognition Workshop*, PAIR '10, 2010b. (to appear).

Paul Reid. *Biometrics and Network Security.* Prentice Hall, December 2003. ISBN 978-0-13101-549-4.

Ioannis Rekleitis. *Cooperative Localization and Multi-Robot Exploration*. Ph.d. thesis, McGill University, Montrèal, Québec, Canada, 2003.

Ioannis Rekleitis, Gregory Dudek, and Evangelos Milios. Probabilistic cooperative localization and mapping in practice. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2 of *ICRA '03*, pages 1907–1912, September 2003. doi: 10.1109/ROBOT.2003.1241873.

Vincent Rialle, Florence Duchene, Norbert Noury, Lionel Bajolle, and Jacques Demongeot. Health "smart" home: Information technology for patients at home. *Telemedicine Journal and e-Health*, 8(4):395–409, December 2002. doi: 10.1089/15305620260507530.

Mark Roberti. UHF solution tracks 42,000 runners at the New York City Marathon. Online, November 2009. URL `http://www.rfidjournal.com/blog/entry/5368/`. Accessed: 2010.10.03.

Rubén Vera Rodríguez, Richard P. Lewis, John S. D. Mason, and Nicholas W. D. Evans. Footstep recognition for a smart home environment. *International Journal of Smart Home*, 2(2):95–110, April 2008.

Phil Rose. *Forensic Speaker Identification*. International Forensic Science and Investigation. CRC Press, 1^{st} edition, July 2002. ISBN 978-0-41527-182-0.

Peter Saint-Andre. Extensible messaging and presence protocol (XMPP): Core. RFC 3920 (Proposed Standard), October 2004a. URL `http://www.ietf.org/rfc/rfc3920.txt`.

Peter Saint-Andre. Extensible messaging and presence protocol (XMPP): Instant messaging and presence. RFC 3921 (Proposed Standard), October 2004b. URL `http://www.ietf.org/rfc/rfc3921.txt`.

Albert Salah, Ramon Morros, Jordi Luque, Carlos Segura, Javier Hernando, Onkar Ambekar, Ben Schouten, and Eric Pauwels. Multimodal identification and localization of users in a smart environment. *Journal on Multimodal User Interfaces*, 2 (2):75–91, 2008. doi: 10.1007/s12193-008-0008-y.

Domnic Savio and Thomas Ludwig. Smart Carpet: A footstep tracking interface. In *The International Conference on Advanced Information Networking and Applications Workshops*, volume 2 of *AINAW '07*, pages 754–760, Los Alamitos, CA, USA, May 2007. IEEE Computer Society. doi: 10.1109/AINAW.2007.338.

Hellmut H. Schmid. Worldwide geometric satellite triangulation. *Journal of Geophysical Research*, 79(35):5349–5376, December 1974. doi: 10.1029/JB079i035p05349.

Dick Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proceedings of the IEEE International Conference on Robotics* *and Automation*, volume 2 of *ICRA '01*, pages 1665–1670. IEEE, 2001. doi: 10.1109/ROBOT.2001.932850.

S.M.N.A Senanayake, D. Looi, M. Liew, K. Wong, and J.W Hee. Smart floor design for human gait analysis. In Franz Konstantn Fuss, Aleksandar Subic, and Sadayuki Ujihashi, editors, *The Impact of Technology on Sport II*, pages 187–192. Taylor and Francis, 1^{st} edition, 2007. ISBN 978-0-415-45695-1.

Chao Shen, Zhongmin Cai, Xiaohong Guan, Huilan Sha, and Jingzi Du. Feature analysis of mouse dynamics in identity authentication and monitoring. In *IEEE International Conference on Communications*, ICC '09, pages 1–5, June 2009. doi: 10.1109/ICC.2009.5199032.

Dave Sims. Biometric recognition: our hands, eyes, and faces give us away. *Computer Graphics and Applications, IEEE*, 14(5):14–15, September 1994. doi: 10.1109/38.310718.

Geetika Singla, Diane J. Cook, and Maureen Schmitter-Edgecombe. Tracking activities in complex settings using smart environment technologies. *International Journal of BioSciences, Psychiatry and Technology*, 1(1):25–36, January 2009.

Geetika Singla, Diane J. Cook, and Maureen Schmitter-Edgecombe. Recognizing independent and joint activities among multiple residents in smart environments.

Journal of Ambient Intelligence and Humanized Computing, 1(1):57–63, 2010. doi: 10.1007/s12652-009-0007-1.

Lauro Snidaro, Christian Micheloni, and Cristian Chiavedale. Video security for ambient intelligence. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(1):133–144, January 2005. doi: 10.1109/TSMCA.2004.838478.

Sargur N. Srihari, Sung-Hyuk Cha, Hina Arora, and Sangjik Lee. Individuality of handwriting: a validation study. In *Proceedings of the International Conference on Document Analysis and Recognition*, ICDAR '01, pages 106–109, 2001. doi: 10.1109/ICDAR.2001.953764.

Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. Using height sensors for biometric identification in multi-resident homes. In Patrik Floréen, Antonio Krüger, and Mirjana Spasojevic, editors, *Pervasive Computing*, volume 6030 of *Lecture Notes in Computer Science*, pages 337–354. Springer Berlin / Heidelberg, 2010. doi: 10.1007/978-3-642-12654-3_20.

Neal Stephenson. *Cryptonomicon*. Avon, May 1999. ISBN 978-0-06051-280-4.

Ali Maleki Tabar, Arezou Keshavarz, and Hamid Aghajan. Smart home care network using sensor fusion and distributed vision-based reasoning. In *Proceedings of the ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN

'06, pages 145–154, New York, NY, USA, 2006. ACM. doi: 10.1145/1178782.1178804.

Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 158–175, Berlin / Heidelberg, 2004. Springer-Verlag. doi: 10.1007/978-3-540-24646-6_10.

Brian L. Thomas and Aaron S. Crandall. A demonstration of PyViz, a flexible smart home visualization tool. In *IEEE International Conference on Pervasive Computing and Communications*, PerCom '11, 2011. (to appear).

Andrei Tolstikov, Jit Biswas, Chen-Khong Tham, and Philip Yap. Eating activity primitives detection - a step towards ADL recognition. In *The International Conference on e-health Networking, Applications and Services*, Healthcom '08, pages 35–41, July 2008. doi: 10.1109/HEALTH.2008.4600106.

Shripad Tuljapurkar, Naohiro Ogawa, and Anne H. Gauthier, editors. *Ageing in Advanced Industrial States: Riding the Age Waves*, volume 3 of *International Studies in Population*. Springer, 1^{st} edition, June 2010. ISBN 978-9-04813-552-3.

U.S. Census Bureau. Projections of the population by selected age groups and sex for the United States: 2010 to 2050 constant net international migration series (NP2009-T2-C). Online, December 2009. Table 2-C.

Miika Valtonen, Jaakko Mäentausta, and Jukka Vanhala. TileTrack: Capacitive human tracking using floor tiles. In *The IEEE International Conference on Pervasive Computing and Communications*, PerCom '09, pages 1–10, Los Alamitos, CA, USA, March 2009. IEEE Computer Society. doi: 10.1109/PERCOM.2009.4912749.

Tim van Kasteren and Ben Krose. Bayesian activity recognition in residence for elders. In *The IET International Conference on Intelligent Environments*, IE '07, pages 209–212, September 2007. ISBN 978-0-86341-853-2.

Alex Varshavsky, Eyal de Lara, Jeffrey Hightower, Anthony LaMarca, and Veljo Otsason. GSM indoor localization. *Pervasive and Mobile Computing*, 3(6):698–720, December 2007. doi: 10.1016/j.pmcj.2007.07.004.

Alladi Venkatesh. Digital home technologies and transformation of households. *Information Systems Frontiers*, 10(4):391–395, May 2008. doi: 10.1007/s10796-008-9097-0.

Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967. doi: 10.1109/TIT.1967.1054010.

Fusheng Wang and Peiya Liu. Temporal management of RFID data. In *Proceedings of the International Conference on Very Large Data Bases*, VLDB '05, pages 1128–1139. VLDB Endowment, 2005. ISBN 1-59593-154-6.

Liang Wang and Xin Geng. *Behavioral Biometrics for Human Identification: Intelligent Applications*. IGI Global, 2010. doi: 10.4018/978-1-60566-725-6.

Roy Want. RFID explained: A primer on radio frequency identification technologies. *Synthesis Lectures on Mobile and Pervasive Computing*, 1(1):1–94, 2006. doi: 10.2200/S00040ED1V01Y200602MPC001.

Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992. doi: 10.1145/128756.128759.

Kajiro Watanabe, Yosuke Kurihara, and Hiroshi Tanaka. Ubiquitous health monitoring at home - sensing of human biosignals on flooring, on tatami mat, in the bathtub, and in the lavatory. *Sensors Journal, IEEE*, 9(12):1847–1855, December 2009. doi: 10.1109/JSEN.2009.2030987.

Harry Wechsler. *Reliable Face Recognition Methods: System Design, Implementation and Evaluation*. International Series on Biometrics. Springer, November 2006. ISBN 978-0-387-22372-8.

Ron Weinstein. RFID: A technical overview and its application to the enterprise. *IT Professional*, 7(3):27–33, May 2005. doi: 10.1109/MITP.2005.69.

Liau Wen-Hau, Wu Chao-Lin, and Fu Li-Chen. Inhabitants tracking system in a cluttered home environment via floor load sensors. *IEEE Transactions on Automation Science and Engineering*, 5(1):10–20, January 2008. doi: 10.1109/TASE.2007.911671.

Daniel Wigdor, Yuri Ivanov, and Christopher R. Wren. Soda pop zombies: soft drink consumption and motion. In *Proceedings of the Workshop on Massive Datasets*, MD '07, pages 8–9, New York, NY, USA, 2007. ACM. doi: 10.1145/1352922.1352925.

Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. In *Proceedings of the International Conference on Ubiquitous Computing*, UbiComp '08, pages 114–123, New York, NY, USA, 2008. ACM. doi: 10.1145/1409635.1409651.

John D. Woodward Jr., Nicholas M. Orlans, and Peter T. Higgins. *Biometrics: Identity Assurance in the Information Age.* McGraw-Hill Osborne Media, 1^{st} edition, December 2003. ISBN 978-0-07222-227-2.

Christopher R. Wren, Yuri A. Ivanov, Ishwinder Kaur, Darren Leigh, and Jonathan Westhues. SocialMotion: Measuring the hidden social life of a building. In Jeffrey

Hightower, Bernt Schiele, and Thomas Strang, editors, *Location- and Context-Awareness*, volume 4718 of *Lecture Notes in Computer Science*, pages 85–102. Springer, 2007. doi: 10.1007/978-3-540-75160-1_6.

Candy Yiu and Suresh Singh. Tracking people in indoor environments. In *Proceedings of the International Conference on Smart Homes and Health Telematics*, ICOST'07, pages 44–53, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-73034-7.

Moustafa Youssef and Ashok Agrawala. The Horus WLAN location determination system. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, pages 205–218, New York, NY, USA, 2005. ACM. doi: 10.1145/1067170.1067193.

Chen-Rong Yu, Chao-Lin Wu, Ching-Hu Lu, and Li-Chen Fu. Human localization via multi-cameras and floor sensors in smart home. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 5 of *SMC '06*, pages 3822–3827, October 2006. doi: 10.1109/ICSMC.2006.384726.

Miao Yu, Syed Mohsen Naqvi, and Jonathon Chambers. Fall detection in the elderly by head tracking. In *Proceedings of the IEEE/SP Workshop on Statistical Signal Processing*, SSP '09, pages 357–360, September 2009. doi: 10.1109/SSP.2009.5278566.

Philip S. Yu. Data mining and personalization technologies. In *Proceedings of the*

International Conference on Database Systems for Advanced Applications, DASFAA '99, pages 6–13, Washington, DC, USA, 1999. IEEE Computer Society. doi: 10.1109/DASFAA.1999.765731.

Keunho Yun and Daijin Kim. Robust location tracking using a dual layer particle filter. *Pervasive and Mobile Computing*, 3(3):209–232, 2007. doi: 10.1016/j.pmcj.2006.10.001.

W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December 2003. doi: 10.1145/954339.954342.

Junyi Zhou and Jing Shi. Localisation of stationary objects using passive RFID technology. *International Journal of Computer Integrated Manufacturing*, 22(7):717–726, July 2009. doi: 10.1080/09511920802209074.