TYPE-1 AND TYPE-2 FUZZY SYSTEMS FOR DETECTING VISITORS IN AN UNCERTAIN ENVIRONMENT

A Thesis Presented to the

Faculty of the Graduate School of the

University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

KEVIN W. REED

Dr. Marjorie Skubic, Thesis Supervisor

July 2009

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

TYPE-1 AND TYPE-2 FUZZY SYSTEMS FOR DETECTING VISITORS IN AN UNCERTAIN ENVIRONMENT

presented by Kevin W. Reed,

a candidate for the degree of Master of Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Marjorie Skubic, Ph.D.

Dr. James Keller, Ph.D

Dr. Gregory Alexander, Ph.D

DEDICATION

I am dedicating this work to all those people who have given so much of themselves to help me get to where I am today. My beautiful wife Jenn and our newborn son Jarrett are the two most important people in my life. It is impossible to express how much I love you two.

My parents, Bill and Peggy Reed, are the origins of my being. They have shown me work ethic, love of family, and faith in God that has been the basis of my success over the years. They also provided an older sister, Meredith, who looked after me during the K-12 years. She is the best sister I could ask for. My younger brother, Chris, has helped me learn about unconditional love and patience.

I have also been blessed with wonderful grandparents. Ray and Gertie Reed used to let me visit for a week during the summers. Grandpa has passed on now, but his legacy lives on in my memory. Grandma Reed taught me a lot about cooking and caring for others. Grandma and Grandpa Pence taught me a lot about being passionate about what I love. I still enjoy the big family reunions we have every couple of years.

Without getting into too great of detail, a few specific individuals I would like to mention: Dwight Haun – who first taught me to love math and science, Rick Barnhouse – a trusted friend and mentor, and Don Tosh – who continues to be a close friend and confidant inside the math and academic communities.

ACKNOWLEDGEMENTS

I first need to recognize Dr. Marjorie Skubic, my academic and thesis advisor. Dr. Skubic has provided research guidance through my time as a Master's student. I really had no idea what I was getting myself into, but thankfully I had an expert to point me in the right direction.

I would also like to acknowledge Dr. Jim Keller and the other professors who have taught me so much about engineering during my time at Mizzou. My professors have been gracious with their time and understanding.

A few colleagues of mine have been instrumental in my transition from work life back to school life: Erik Stone, Sam Blisard, and Ian Kable. I would also like to thank my friends in Columbia, Springfield, and abroad for their support and encouragement.

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ABBREVIATIONS AND TERMS

PIR	Passive Infrared
Firing, Hit, or Triggered	The motion sensor has sent data
Visitor	Anyone other than the resident
SISO	Single Input, Single Output
MISO	Multiple Input, Single Output
Tigerplace	An assisted living facility associated with Missouri University
NSSP	National Strategy for Suicide Prevention
N-ary	If N = 2, binary; N = 3, tertiary; etc.
Fuzzification	Converting crisp input to a fuzzy value
Rule Base	The set of IF-THENs for inference
Defuzzification	Converting fuzzy values to crisp output
Resident	A senior that is part of the study
FOU	Footprint of Uncertainty
IT2	Interval Type-2
GT2	Generalized Type-2
Baffle	A physical blind to limit the maximum view angle of a motion sensor
T1M	Type-1 MISO
T1S	Type-1 SISO
T2M	Type-2 MISO

LIST OF ABBREVIATIONS AND TERMS

T2S	Type-2 SISO
Agent	A sub classifier that produces part of the final confidence value
Function	The function that approximates the other classifiers

TYPE-1 AND TYPE-2 FUZZY SYSTEMS FOR DETECTING

VISITORS IN AN UNCERTAIN ENVIRONMENT

Kevin W. Reed

Dr. Marjorie Skubic, Thesis Supervisor

ABSTRACT

In this work, I have developed an algorithm to detect the presence of visitors in a noninvasive manner. This algorithm is designed as part of an in home monitoring system. The data from the algorithm will be used as a way to monitor the social health of the resident. It will also be used to help isolate the times when the resident is in the apartment alone, so that parameters like activity levels can be calculated.

Type-1 and Type-2 fuzzy systems are compared for classification performance. A series of lab tests provided the information necessary to model the motion sensors. Results from the motion sensor tests are used as guides for the Footprint of Uncertainty (FOU) used in the Type-2 systems. It is shown that the FOU values do not significantly impact the classification accuracy of the Type-2 systems. Classification accuracy of the ground truth data collected in a test apartment reached 88%. Additionally, the Type-2 MISO 2 Agent and Type-2 SISO systems best identify the known visitor times in the resident apartments.

Ongoing human subject monitoring data is evaluated empirically. The results from an organized set of tests in a test apartment are presented.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Visitor Detection Problem Statement

The goal of this project was to use sparse motion sensor data to detect the presence of visitors in a senior's home. The first step in this process is to model the motion sensors that my research group has deployed in a test group's apartments. Lessons learned from the motion sensor model led to the creation of features used to detect occupants other than the senior. By tracking motion density, approximated velocity, and event duration, I am able to detect visitors with a reasonable confidence. The last task is to reduce or remove high frequency noise from the results.

Two important considerations for the appropriate sensor type are cost and privacy. PIR sensors fit both considerations. The sensors can be purchased off the shelf for around 20 dollars apiece. Using environmental sensors allows us to collect data in a natural and unobtrusive way (i.e. without the senior's intervention). The type of data collected and its use limits any privacy issues. The data are then used to track a variety of metrics that tell us about the senior's well being, as well as possible health events such as depression, falls, and other ailments.

My part in this project has taught me a great deal of respect for the time and commitment it takes to perform research. The work that went into this project has been rewarding, and I think the results will benefit the seniors that are able to take advantage of the smart home system being produced by the Eldertech group.

1.2 Overview

The Visitor Detection Algorithm is a piece of a larger project to benefit seniors. It is called the Eldertech project, and has been covered by a series of grants from NSF, NIH, and the U.S. Administration on Aging. The work behind this thesis is through an NSF ITR grant. The Eldertech research is aimed at producing a marketable noninvasive monitoring system that can be deployed in homes, independent living facilities, and assisted living facilities. The completion of this system will allow seniors to remain independent longer and reduce care costs.

The Eldertech project is concerned with all aspects of a senior's well being. A few of the areas being researched include:

1. Self Care (Hygiene, Meals, etc.)
2. Physical Ailments (Falls, Limps, Breathing Troubles, etc.)
3. Mental Ailments (Alzheimer's, Depression, etc.)
4. Social Interaction (Visitor Detection, etc.)

The focus of this paper is the creation of an algorithm to detect visitors. This algorithm creates a history of visitor confidences in the apartment. An additional algorithm produces a visual display for validation purposes. Its appearance is based on a representation developed by Wang and Skubic [1].

The current methods for determining the social health of seniors are inadequate. Most health care providers must depend on questionnaires. The questionnaires are subject to a lot of variance. Nurses in the Eldertech group have described two primary concerns. The first is when the senior feels that if they answer the questions truthfully, they will be forced to lose some of their independence. The second case is that the senior does not want to feel as though they will be inconveniencing anyone. Both cases prevent honest answers to the questions on the survey.

Another issue that has been brought up with the current testing method is that the questions are only taking a snapshot out of the senior's day or week. This problem also presents itself for other doctor office type tests. A senior may have normal blood pressure at 10 am every Wednesday morning when they are tested, but exhibit high blood pressure and hypertension in the afternoon.

Creating an environmental sensor network enables us to gather objective information that can be used to determine the health of the senior. By looking at this objective information, it is possible to form a better model of the senior's health. Consistent data is easier to analyze for small changes in behavior, and social health.

Data for visitor detection is gathered using passive motion sensors, typically found in basic home security systems. In measuring several features of the PIR sensors, the capabilities of the sensor can be derived. This information is used to help form an uncertainty model of the sensors. Modeling of the sensors is covered in detail in section 3.3.

Based on the fact that the environment and sensors we are working with are wrought with uncertainty, it is a natural choice to use fuzzy set theory to create a confidence for the presence of visitors. Eight systems are tested in this paper. The first classifier is a multiple input, single output function. It is a fast classifier that approximates the fuzzy systems. The second and third classifiers are Type-1 fuzzy inference systems, whose background is covered in section 3.2.1 and whose implementation is covered in 4.6. One is a single input, single output (SISO) Type-1, and the other is a multiple input, single output (MISO) Type-1 system. The remaining systems are of the Type-2 variety. A SISO Type-2 No FOU version (similar to SISO Type-1) is followed by two MISO Type-2 No FOUs. The first MISO Type-2 No FOU averages two agents, while the second performs all calculations inside the inference system. Similarly, two MISO Type-2 systems with FOU complete the list.

Type-2 fuzzy system proponents claim that Type-2 systems produce better models of uncertainty, which would be ideal for this problem. The background for Type-2 fuzzy systems is covered in section 3.2.2. Implementation information can be found in 4.7. As a comparison point, I have also created a simple function that tries to imitate the behavior of the fuzzy classifiers. If the results are similar to the fuzzy systems, the simpler design will make it an appealing alternative computationally. The function creation is described in section 4.8.

Some related Type-1 and Type-2 research can be found in Chapter 2 Section 1. Section 2 compares this work to other engineering approaches to monitoring senior health. Chapter 3 contains a background of the techniques that were used in this paper, which include: social health of seniors, Type-1 fuzzy systems, Type-2 fuzzy systems, and motion sensors. Chapter 4 covers the visitor detection algorithm, beginning with data extraction. Chapter 4 continues with the construction of each of the features and their membership functions. Chapter 4 concludes with the rule base and a description of the output data. Chapter 5 contains a description of tests performed in the test apartment, with results following in Chapter 6. Conclusions and some comments on future work may be found in Chapter 7.

Chapter 2

RELATED WORKS

2.1 Type-1 and Type-2 Fuzzy Systems

Modeling uncertainty in terms of classification membership is the realm of fuzzy systems. The term "fuzzy," as it applies to set theory, first appeared in Lofti Zadeh's 1965 paper [2]. Ten years later, Zadeh wrote a trio of papers that included fuzzy sets with fuzzy membership functions [3], [4], and [5]. Zadeh named the new system "Type-2." So the primary conceptual difference between Type-1 and Type-2 fuzzy systems is how the membership functions are defined.

Type-1 fuzzy systems have been used in industry for control applications for several years now. Type-1 systems are also becoming increasingly popular in other applications as well. The step to Type-2 fuzzy systems is taking longer, generally due to the additional computational demands from the algorithm in a real time environment.

Type-2 fuzzy system proponents argue that Type-1 fuzzy systems are too crisp. This comes from the observation that the edges of the membership functions are generally uncertain, yet are crisply defined [6]. They argue that an expert opinion typically cannot be adequately defined using Type-1 membership.

Mendel and John coauthored a paper to encourage the fuzzy community to try the more complex fuzzy system, and discussed the computational issues [7]. The Type-2 system benefits the user by providing a better model for the uncertainty involved in their problem. The downside is the complexity, both in terms of understanding the algorithm and computation time. Mendel and John have also produced a couple of papers that do a good job simplifying the concepts required to implement Type-2 systems [7], [8]. Research continues in the Type-2 community to reduce the computational complexity [9], [10].

Hani Hagras has compared Type-1 and Type-2 fuzzy systems in several applications and concluded that the Type-2 systems are the better choice [11], [12]. Hagras found that in an uncertain environment, Type-2 has better generalization and can tolerate input and output noise in changing environments. Hagras also points out that when performance of a Type-1 set degrades due to outside variables it must be re-tuned; which is a time consuming process. He found that Type-2 sets are more robust and generally will not need to be re-tuned or re-tuned less often. Hagras has also written an overview of his findings that provides a nice summary of his work with autonomous robots [13]. A particularly interesting application of Type-2 systems is modeling unanticipated network states that transmit video data [14].

Wang and Acar compare and contrast Type-1 and Type-2 fuzzy systems in their proof driven paper [15]. They show that Type-2 systems are better able to handle uncertainty regarding the rule base and memberships. Wang and Acar create and prove two theorems that help the user convert a Type-1 system and its uncertainties into a Type-2 system.

Paetz's paper discusses the fact that the calculations performed in fuzzy logic are still based on a finite subset of the real numbers [16]. Instead, inputs and outputs are quantized based on the variable length defined by the computer. Since we are performing calculations in an exact space (limited significant digits), it makes sense to build fuzzy memberships to deal with the additional uncertainty from the data loss.

This paper focuses primarily on classifying past data, because the visitor detection algorithm is not currently a real time application. I compare a function, two Type-1 systems, SISO and MISO, as well as five Type-2 systems, a SISO and four MISOs (two with FOU, two without; two with 2 agents, two with 1 agent).

2.2 Monitoring Seniors

The smart home concept has been around for a long time. For the senior, a smart home system gives them an opportunity to age in place without the fear that any trouble they have will go unnoticed. There are several systems that are in production, and the technologies garner a lot of attention from the research community as well.

Gaddam summarizes the current state of wireless sensor network technologies, and how these can transform our homes in his recent paper [17]. He classifies the technology into four branches: adaptive technology, assistive technology, inclusive designs, and medical technologies. Gaddam also lays out a checklist of considerations and concerns for the design of the wireless sensor network. Like Gaddam, our research group is particularly concerned by any intrusiveness of the sensors. We feel that a natural environment where the elder is able to ignore or forget the network gives us a better look at how they actually live. Gaddam lists other concerns like: interference from other sensors (see Section 3.3.5), data acquisition, data management, data security, reliability, and energy sources.

Glascock and Kutzik have developed a monitoring system that checks for specific tasks performed by the senior [18]. For instance, their system determines if and when the senior prepared their lunch or took their medicines. The interface is web based, so a care giver can check on them at any time. The system is self limiting, though. It will only summarize a predesigned list of basic activities.

A much more invasive system has been developed by Wu and others [19]. Their system still uses some passive environmental sensors such as smoke detectors, but also requires that the senior wear an accelerometer. The data collected from the accelerometer is used to infer health problems and nervousness among other things. An interesting caveat of this product is that the hardware connected to the accelerometer communicates with a distributed data logging system located in all of the apartments.

Hudson and Cohen discuss a rural version of telecare [20]. Their focus is on seniors in rural areas that are not able to actively seek medical care. The proposed solution begins by inserting health monitoring equipment into the home. The health care provider calls the patient at home, and is able to access data from the monitoring equipment. A video conferencing system simulates a visit with the doctor without the need for a commute to the office. Alerts are generated for family members and rescue teams that may need to take action to secure the safety of the senior.

Le and others track activity levels in an interesting non-invasive way [21]. Rather than attaching devices to the senior, they set up a fuzzy system that assigns mobile and immobile states to them. The subject is assigned the mobile state until there is no recorded activity for 10 seconds. Similarly, they remain in the immobile state until there are five sensor hits that are consecutively less than 10 seconds apart. From the mobile and immobile assignments, Le infers the senior's ability to perform tasks contextual to the senior's location.

An advanced health monitoring solution has been developed by Intel, and is discussed in a paper by Dishman [22]. The Intel system detects signs of disease and treatment compliance (Are they taking their medicine?). But where Dishman's research is really different from other research is in instructional assistance. The system monitors the progress of an activity such as making tea. During cognitive decline, a senior may have difficulty remembering how to perform simple tasks. When the system detects that the senior is performing steps out of order, or is taking a long time to finish a step, an audio/video monitor asks if the senior needs help. Upon confirmation, the monitor can play audio and video directions for the senior to follow. This technology will be particularly beneficial for those suffering from Alzheimer's disease.

The Eldertech group has also collaborated to construct a smart home monitoring system. Volunteer participants at Tigerplace have environmental sensors installed in their apartments. A network of motion sensors provides data for visitor detection, as well as a host of other uses [1], [23]. Also, bed sensors collect restlessness, breathing, and pulse rates that have been correlated to health events [23], [24].

Other technologies are also being developed for use in the apartments. Video data shows a lot of promise, but the cameras are perceived as intrusive, and cannot be placed in all rooms of the apartment. However, there is no real threat to privacy, because the video is not stored or viewed. The data collected from the cameras are processed into silhouettes from multiple angles. The silhouettes are then reconstructed into a voxel image, which can be interpreted as a rough 3D representation of the senior [25]. Calculations on the 3D reconstruction are used for several measurements and metrics.

The features and data collection for visitor detection in this paper are most similar to those presented in Martin's paper [26]. Martin uses a similar network structure to the one at Tigerplace, but with many more nodes (motion sensors). His algorithm is based on five features. The first is activity level change, which is similar to the motion density used in my algorithm. Another feature is the number of room changes recorded. The layout of the Tigerplace apartments does not facilitate this metric very well. There are overlapping regions that artificially inflate this value. The last three features Martin uses are delay time, hallway time, and door time. We do not have hallway motion sensors in Tigerplace, and the motion sensors we are using have a resolution of about seven seconds, making the door time feature unreliable. The delay feature assists the algorithm by inferring quick changes in activity associated with visitors entering and leaving compared to general activity changes by the senior.

As stated before, the visitor detection design is most closely related to Martin's paper. The benefit of my algorithm is that it better handles sparse data, and displays a single confidence over regions where a visitor could not have entered or left the residence. In other words, the confidence does not change until there is an event that potentially signals a change in state between visitors and no visitors. I have also developed a way to infer the approximate walking speed of the senior, which is very useful for detecting visitors. In larger homes, or homes with long hallways, walking speed could be an excellent way to predict the identity of a visitor.

Chapter 3

BACKGROUND

The visitor detection algorithm is the culmination of research from three main areas. I begin Chapter 3 with the motivation for visitor detection as a measure of social health. Next, I present a brief background of fuzzy logic, including Type-1 and Type-2 systems. Lastly, I cover passive infrared technology and a series of tests on the motion sensors installed at Tigerplace that I conducted last summer. All three parts combine together to form the basis for visitor detection.

3.1 Senior Social Interaction as it Relates to Overall Health

Friedrich Nietzsche once said, “To forget one’s purpose is the commonest form of stupidity.” Since I am striving to avoid stupidity, I will begin the background section with the purpose of the visitor detection algorithm.

I am interested in monitoring visitors for a couple of reasons. One reason is that social interaction is a key component for a high quality of life. This is especially true of seniors, where a support structure helps deal with problems from mundane to tragic [27]. Conversely, diminishing social interaction puts the senior in a state of isolation that can lead to a host of other problems.

Tracking visitation times are also important for parameterizing other algorithms that measure the health of the resident. For instance, the ability to differentiate times when visitors are not present is essential for producing an accurate model of the resident's activity levels. The motion densities tend to increase with the presence of visitors, so it is desirable to discard these times from the data.

A seniors' quality of life has been classified into three areas: physical health, mental health, and social health. Each one is somewhat dependant on the others. A socially isolated senior will tend to be less active, and exhibit signs of depression. Lower activity levels can lead to a host of physical ailments including blood clots and obesity. The NSSP, National Strategy for Suicide Prevention, has found that suicide rates are highest in the 65 and older population [28]. Isolated seniors without a consistent group of people to interact with often have the signs of depression go unnoticed.

Seniors are the most likely group to be experiencing losses from their support structures. As they get older, friends and relatives move away or pass on. The loss of friends and loved ones adds to the stress from diminished independence and a feeling of lessened purpose. Seniors need a strong support group to cope with the problems associated with age.

A study of online communities for seniors has shown a range of social groups. Some of the social groups simply tell jokes and give well wishes, while others are committed to community building and self-disclosure [27]. These communities can not only provide a place for support, but also foster a sense of self-worth from being able to help others. The online communities are not, however, a replacement for traditional social support structures.

Visitors are a good indicator that a senior has a strong traditional social network. Having guests into the home suggests that the two parties are friends, and the social interaction is bidirectional. Using the number of visitors, and the lengths of their stays, the social well being of a senior can be better understood. Additionally, detecting the resident's normal activity level sans visitors gives us an idea if there are any activity reductions that might otherwise go unnoticed. It is our goal to use the visitor information as a tool to identify ways to improve the resident's quality of life.

3.2 What is Fuzzy Logic?

Uncertainty can take many forms. Most people think of uncertainty as a probability, which is a way of expressing the belief that an event will occur. However, the form of uncertainty pertaining to visitor detection is of the form, "does this belong?" Jim Bezdek clarifies the difference in a series of editorials published in 1994 [29].

I feel it is important here to specify that fuzzy logic does not imply imprecise logic. Fuzzy set theory and fuzzy logic are well defined and robust. Fuzzy is only fuzzy in that where N-ary logic maps $F : x \to \{1,..., N\}$, fuzzy logic maps $F : x \to [0,1]$. In other works, N-ary logic supports N truth values, while fuzzy logic supports an infinite number of truth values (up to whatever can be represented).

Binary logic has been around since Aristotle. The consequent of a binary statement has two possible truth values: true or false. This is what it means to be crisp. Binary logic, also known as Boolean logic, has a complete algebraic system that was defined by George Boole in the 20th century. There is a certain comfort in a black and white world. An object is either in the set or not in the set.

However, what if it is unknown whether an object belongs in the set? One common extension in literature is Lukasiewicz's three valued logic. Three valued logic is a generalization of binary logic. The set of values still contains true and false, but adds a third value "unknown." In the three valued algebra, all binary values behave the same as before.

3.2.1 Type-1 Systems

A simple extension of multivalued logic leads us to infinite valued logic. Fuzzy set theory is introduced in Zadeh's 1965 paper, "Fuzzy Sets" [2]. The main benefit of using a Type-1 system for visitor detection is that there are no all purpose features available that allow us to say with total confidence whether a visitor was in the apartment. Fuzzy logic is able to distinguish likely scenarios where visitors may have been in the apartment and assign a value between 0 and 1. For instance, there is no clear distinction between the number of motion sensor hits when visitors are in the apartment versus when the resident is simply being active. A membership function is created to represent the relative confidence that a visitor was present based on the motion density.

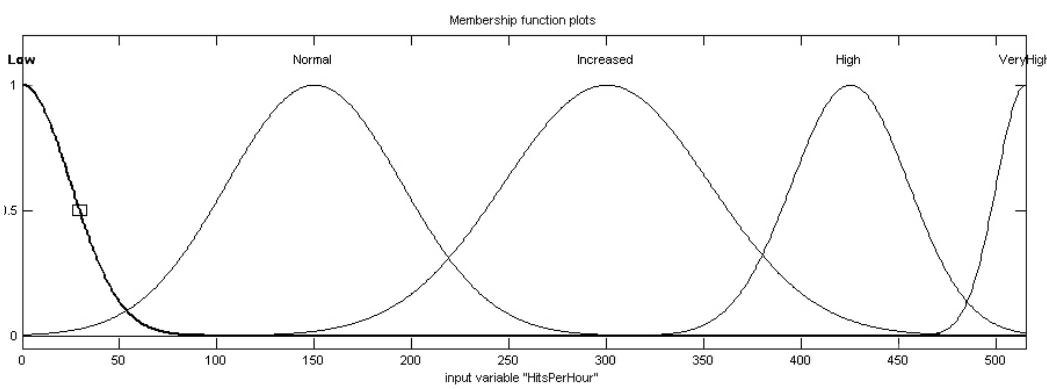


Fig. 3.1: Type-1 fuzzy membership function for motion density

Fuzzification is the task of converting world knowledge into a fuzzy membership. Figure 3.1 is an example of this. As discussed in Section 3.3, the motion sensors fire about every seven seconds. Sensor logs have shown that during peak activity times, such as getting ready in the morning, seniors will produce between 100 and 200 hits per hour.

When the sensor firing rates drop below 50, then the resident is sedentary or out of the apartment. Conversely, when sensor firings are much greater than 200, the likelihood that a visitor is in the apartment is increased.

After creating membership functions for each feature to be considered, a rule base must be created. The rule base performs operations on the membership functions. Input membership functions constitute the antecedent, or "If" part of a rule. In a Mamdani system, the outputs are also made of membership functions. The output members are the consequents, or the "Then" part of a rule. For instance: If motion is high and speed is high, then visitor is likely.

The standard logic operations are extended to fuzzy logic. There are many implementations, but all of them must produce equivalent results when used in traditional binary logic. This means that true and true can't equal anything aside from true, and similarly with other combinations and operators. The new operators are still associative, commutative, and distributive. Some of the common examples are defined in table 3.1. These operators are used to construct the rules.

Table 3.1: Common fuzzy logic operators

Operator	Standard	Product	Yager	Sugeno
a AND b	Min(a,b)	a*b	$1\text{-Min}(1,((1\text{-a})^{p}+(1\text{-b})^{p})^{\frac{1}{p}})$	Max(0,a-b)
a OR b	Max(a,b)	a+b–a*b	$1\text{-Min}(1,(a^{p}+b^{p})^{\frac{1}{p}})$	Min(1,a+b-1)
NOT a	1-a		$(1\text{-a}^{p})^{\frac{1}{p}}$	

The completed rule base takes the input and applies the fuzzy operators. There are two ways to represent the results. Shown in figure 3.2, Sugeno type fuzzy inference creates an output and firing strength. To defuzzify the result, the average of the output is multiplied by the firing strength. Alternatively, Mamdani type fuzzy inference may be used. The only difference between Mamdani and Sugeno inference is the way defuzzification is performed. In Mamdani inference, the firing strength is determined by an output membership function. The process takes the output values from the system

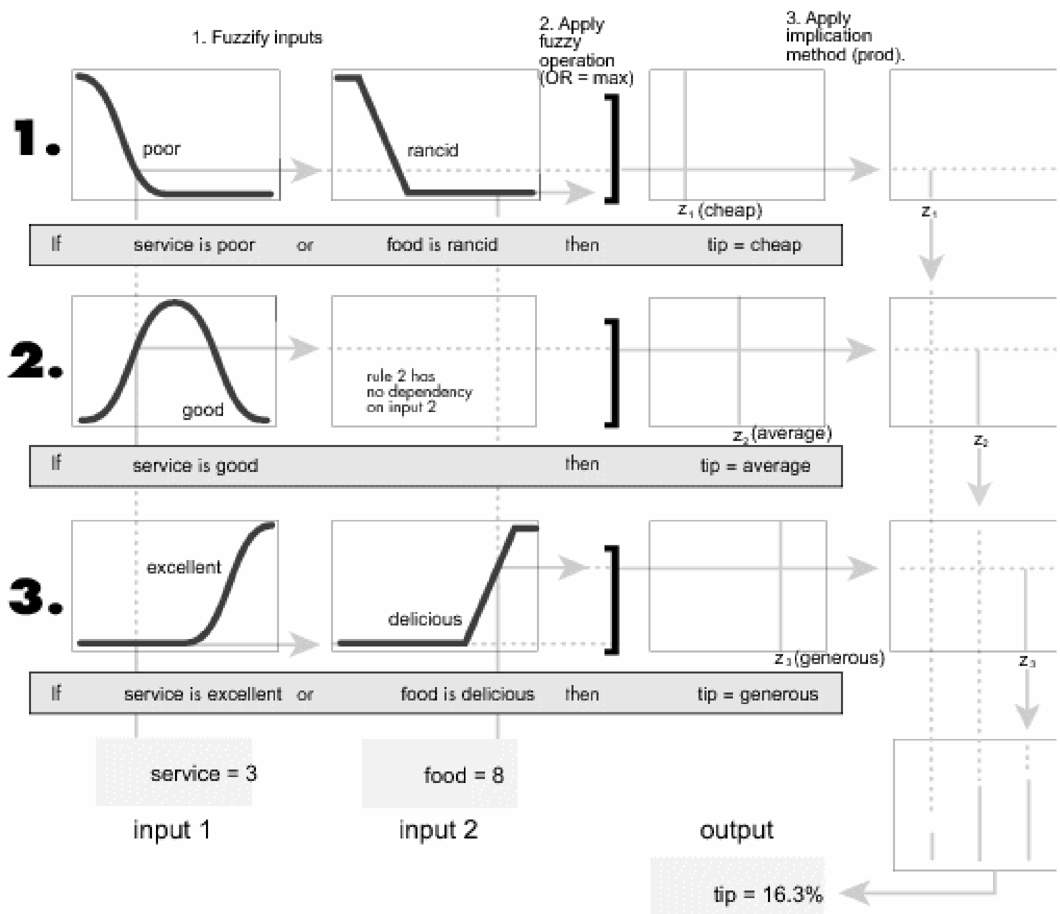


Fig. 3.2: Sugeno type rule application & defuzzification (image from Matlab)

and levels the output members at the point of contact. Figure 3.3 shows an example of Mamdani inference. The resulting functions are recombined, using a maximum operator. The Mamdani inference system solution is the centroid of the area under the curve.

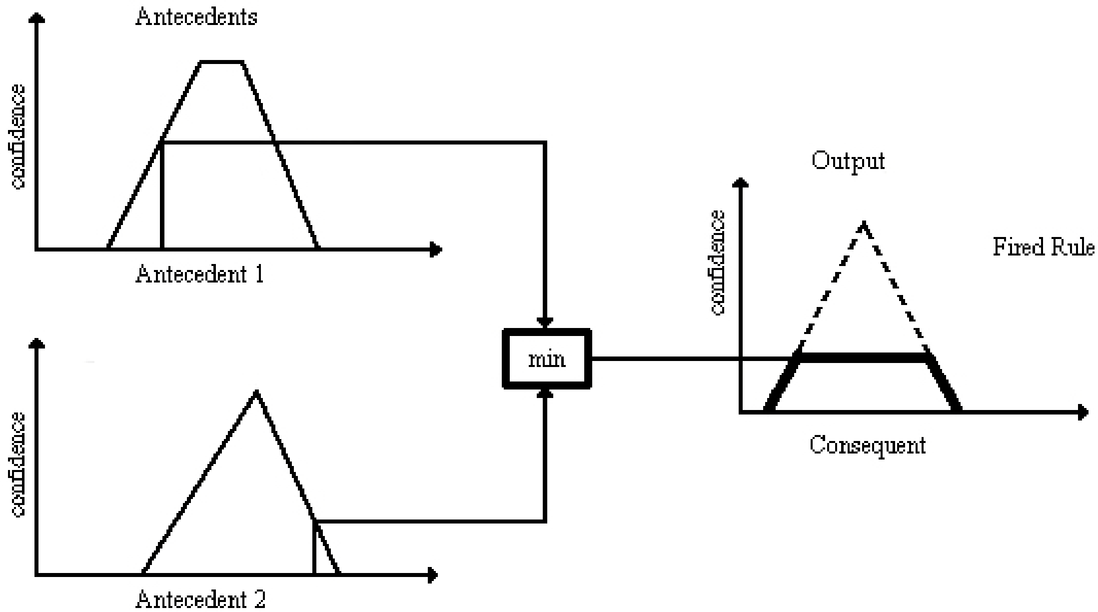


Fig. 3.3: An example of Type-1 Mamdani inference

3.2.2 Type-2 Systems

Ten years after Zadeh's paper on fuzzy sets [2], he produced a series of three papers that further extended fuzzy logic to Type-2 systems [3], [4], [5]. The conceptual difference between Type-1 and Type-2 systems hinges on the creation of the membership functions. One source of classification uncertainty is where the edges of the membership functions ought to be. The term for this area is the "Footprint of Uncertainty," or FOU. If the domain of interest is not well understood, it is difficult to model the data. By fuzzifying the edges of the membership functions, the FOU can be modeled.

Type-2 fuzzy logic appears as two main cases: interval Type-2 and generalized Type-2. The simpler of the two is interval Type-2 (IT2). IT2 allows the creation of membership functions with an interval edge (as opposed to a crisp value). The interval range can be set to model the uncertainty the designer has about the proper location of the left and right endpoints or the mean. Defuzzification of an IT2 is a little more complicated than a Type-1 system when finding the centroid. The Karnik-Mendel (KM) or Enhanced Karnik-Mendel (EKM) algorithm is used during type reduction to find switch points [10]. A pair of switch points is needed to compute two separate values, c_l and c_r. The value c_l represents the minimum centroid for all embedded Type-1 sets from the left switch point L, while c_r represents the maximum centroid for all embedded Type-1 sets with the right switch point R. IT2 will approach a Type-1 set as the interval approaches 0.

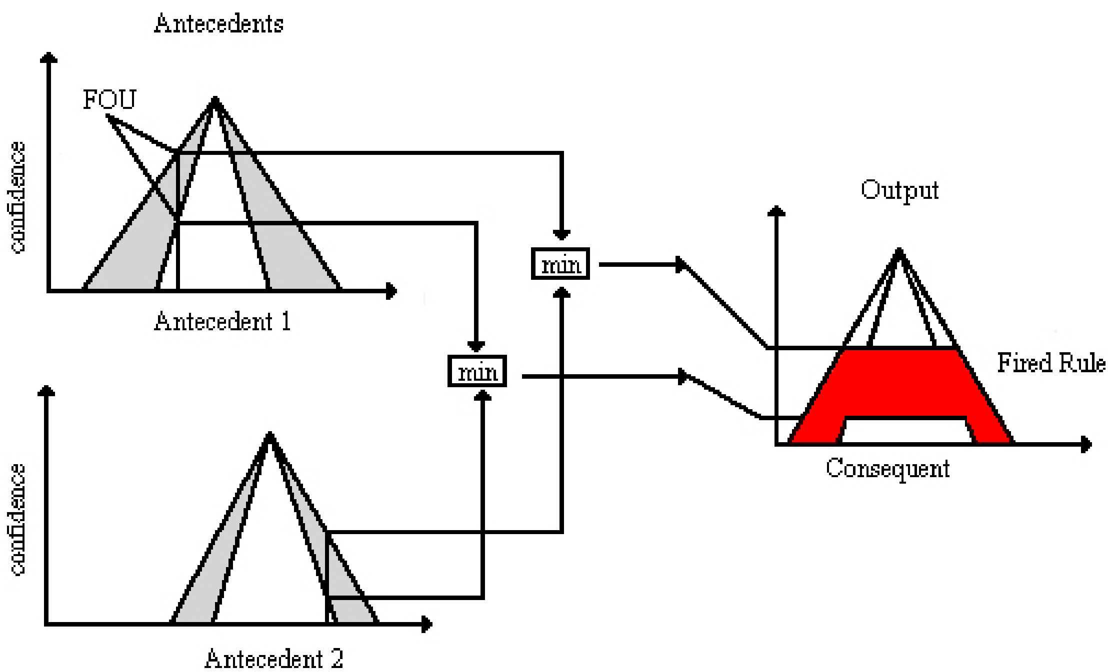


Fig. 3.4: An example of Type-2 Mamdani inference

Generalized Type-2 sets (GT2) also fuzzify the boundaries of the memberships. GT2 makes the assumption that we have a reasonable grasp of the data when drawing the memberships. Therefore instead of an interval edge, GT2 implements a nonlinear edge such as a Gaussian. The confidence is higher near the mean (the equivalent Type-1 edge), and decreases as distance from the mean increases. As the standard deviation of the Gaussian approaches zero, the GT2 approaches a Type-1 system. The biggest problem with GT2 is the computational complexity associated with the fuzzy operators. There is ongoing work to reduce the complexity of this problem [6].

3.3 Non-invasive Monitoring Using Passive Infrared Sensors

This section is a summary of the PIR sensors used and the experiments conducted to characterize the sensor output. An in depth look at motion sensor technology as well as the full study of the Hawkeye II sensor can be found on the SharePoint site under weekly reports. It is a Word document under the file name "MotionSensorSpecsAndProcs.doc".

In an effort to reduce costs of our sensor network, we chose to use off the shelf Hawkeye II PIR motion sensors that operate using the wireless X10 protocol. The X10 protocol is a common protocol for wireless PIR sensors, and it is open source, which make the technology very affordable. PIR motion sensors are not only affordable and wireless; they are designed to be noninvasive. We have found several uses for the data collected by these simple, easy to use sensors. For those interested, they are an Active Home product model number RMS18.

3.3.1 Motion Sensor History and Technology Overview

The use of infrared technology for monitoring activity is not a new concept. In the late 1960's, security installations began looking for alternatives to microwave and ultrasonic motion detection [29] and found the PIR sensors to be promising. Passive infrared was not an immediate success, often plagued by false alarms and construction problems.

In 1979, a major breakthrough in commercial technology occurred when dual pyroelectric sensors were used in the motion detector. By this time, new hardware materials had been created that were less sensitive to noise and sensory spikes. Adding the second pyroelectric sensor made it possible for the motion detector to ignore ambient infrared changes like wind.

The 1980's brought another technological breakthrough when Fresnel lenses replaced a series of mirrors (see figure 3.5). The mirrors, and later lenses, produced a signal change when motion was detected on the pyroelectric material. The Fresnel lenses are easier to produce, and can easily be replaced if damaged. They suffer fewer environmental problems such as dust and humidity.

The last major innovation of the 1980's affected all electronics; improved components. Purer materials produce less noise and are less prone to defects. Continued improvements in transistor and processing components have led to smaller implementations. The motion sensors we use are less than four inches square, pictured in figure 3.6.

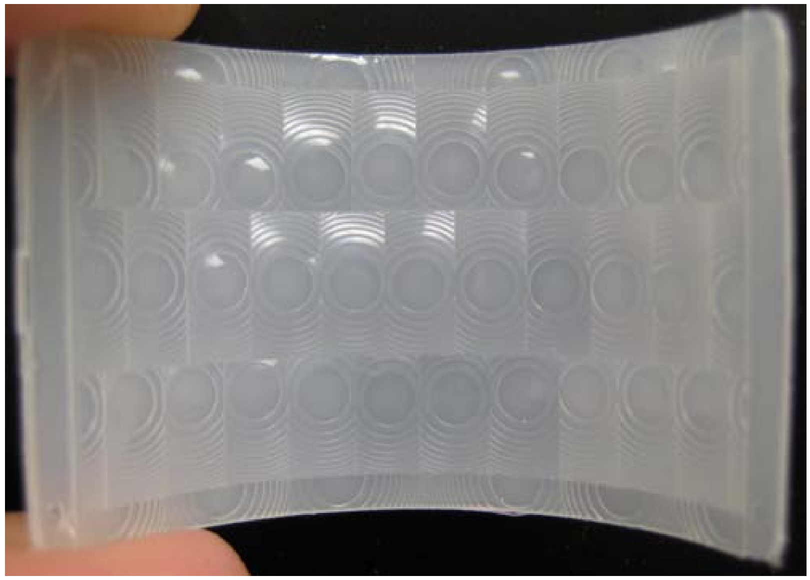


Fig. 3.5: A Fresnel lens from Active Home's Hawkeye II

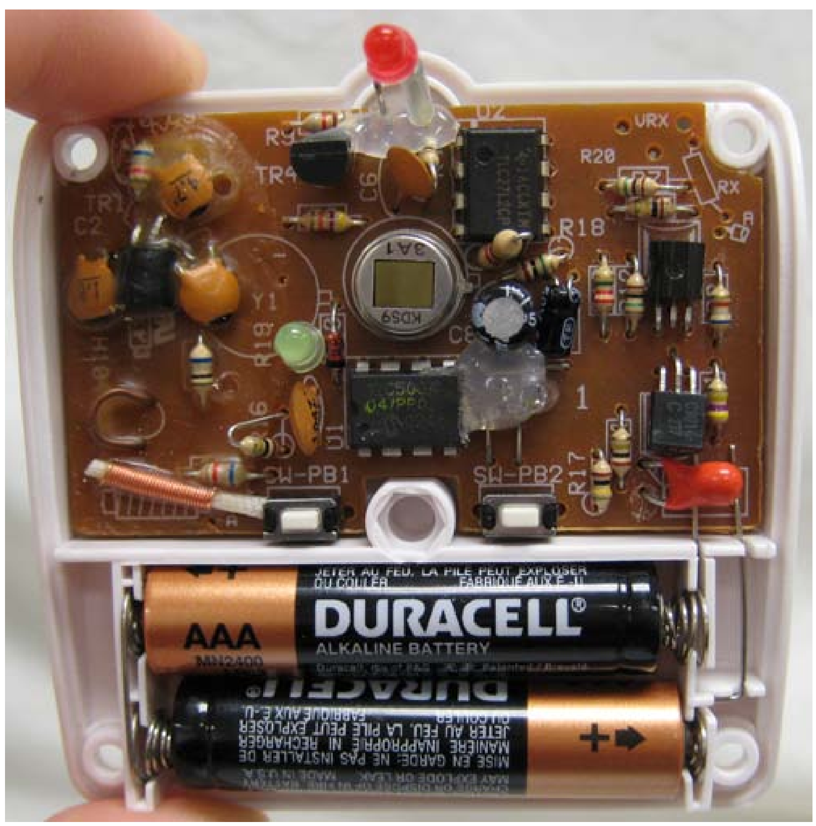


Fig. 3.6: The Hawkeye II is a nonintrusive size for use as a motion detector.

3.3.2 How PIR Motion Sensors Work

PIR sensors detect what is known as black body radiation. All objects emit infrared light, which is not visible to the human eye. The PIR sensor absorbs and is heated by the black body radiation and temporarily produces electric potential.

The ability to detect motion comes from the focusing power of a Fresnel lens. The Fresnel lens gathers the infrared light and focuses it across the sensor. Motion causes the focus to shift, which alters the electric output from the sensor. Two or Four receptors are typically used to reduce false alarm rates from environmental temperature changes.

A feedback loop is used to control false alarms from slow environmental changes, like daytime to nighttime temperatures. The resulting voltage is sent to an amplifier, and then on to a processing unit. The processor determines if the magnitude of the signal has surpassed a specified threshold. The processor must also determine if enough time has passed to transmit again.

The Hawkeye II sensor transmits using the X10 wireless protocol. This protocol does not implement hand-shaking, and so attempts to send 5 pulses of identical data per packet. More information on packet loss is discussed in section 3.3.5.

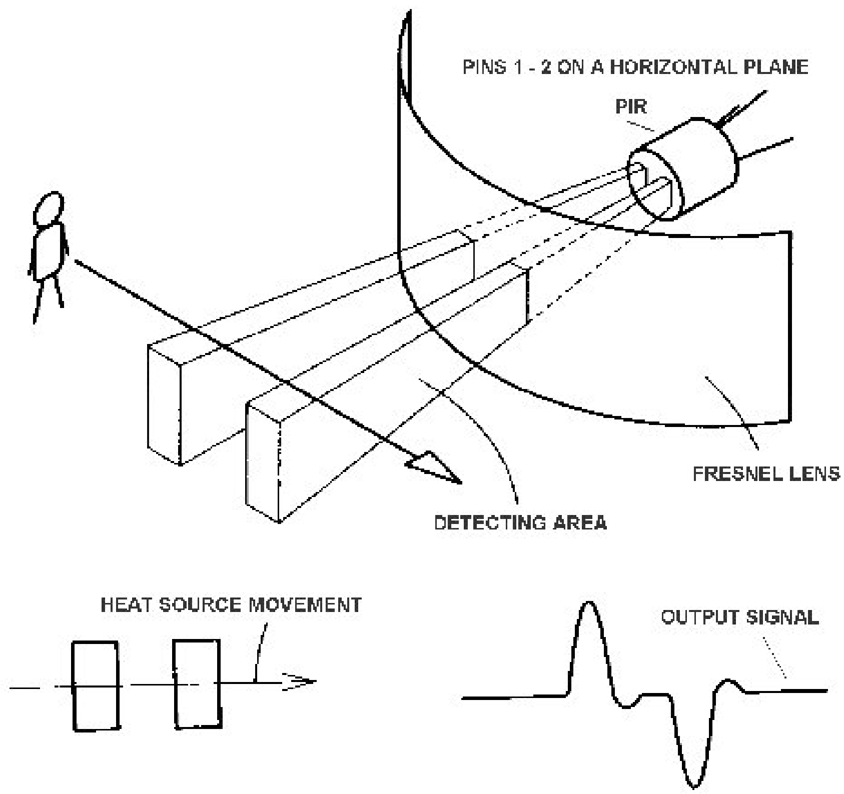


Fig. 3.7: Movement across the view of the motion sensor causes output differences (Image from www.glolab.com)

3.3.3 Test: View Angles

Even with a good understanding of how a PIR motion detector works, there are still many variables. The view angle of the lens is largely based on the Fresnel lens design and focusing distance. This series of tests are designed to help define a fuzzy model of the sensors.

The best way to deal with this uncertainty is to perform lab tests. I began with 10 sensors and tested their maximum viewing angle using a short range setup with a cardboard heat shield, shown in figure 3.8. I began the tests by mounting a protractor underneath the center of the PIR sensor, which was being held in a jig. I then measured the view angle by blocking most of the sensor’s field of view with the heat shield. I could then move a heat source (heating blanket) from behind the heat shield and test for firings. If the sensor did fire, I would then move the heat shield to cover more of the view area. Otherwise, I would reduce the coverage. Tables and graphs of the motion sensor results may be found in Appendix A. Apartment layouts and the sensor distances are in Appendix B.

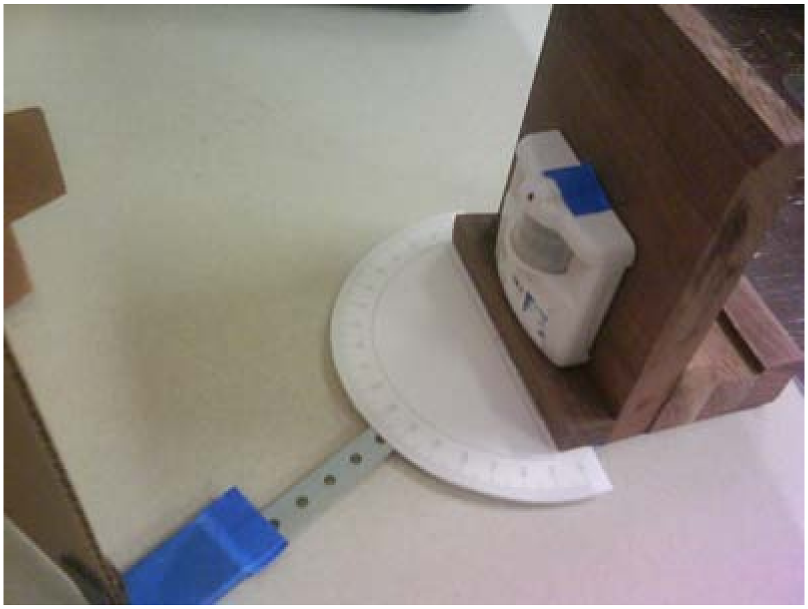


Fig. 3.8: The view angle jig

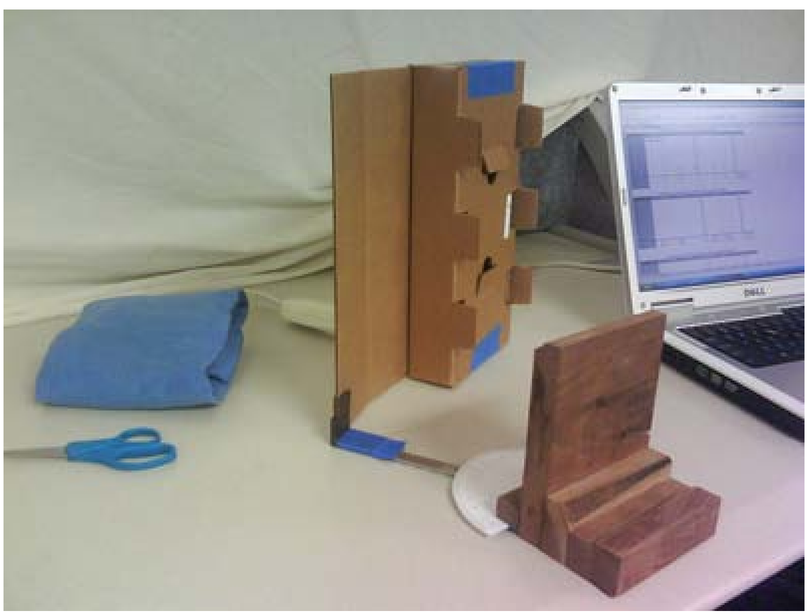


Fig. 3.9: The view angle jig, heat shield, and heat source

I conducted tests that measured four view angles: Left, Right, Up, and Down. These tests were conducted while controlling the heat source temperature and the baffle. The first test acted as a control group. There was no baffle, and the heat source was set to the typical temperature of human skin, 93° F. The second round measured an elevated temperature reading, 97° F.

We found that a baffle system would be desirable so that we could achieve more focused view angles. The baffled system was tested in two parts. The first was a used as a materials search. Five materials were tested; each material was placed such that it occluded the sensor. The materials tested include scotch tape, copy paper, card stock (an index card), wax paper, and aluminum foil. The foil tests were conducted with both the shiny side inward, and the dull side inward. I found that copy paper was the most effective at limiting the view angle. Table 3.2 below shows the results of this rudimentary test.

Table 3.2: Multiple materials were tested to determine those best suited to create a baffle. A refrigerator thermometer was used to detect the temperature of the heating pad and the room. The temperature in the lab was 76º F during the test.

	Body heat (93)	low heat (92-94)	med heat (96-98)	high heat (106-108)
control	hit	hit	hit	hit
tape	hit	hit	hit	hit
copy paper	no hit	no hit	no hit	no hit
index card	no hit	no hit	no hit	no hit
wax paper	hit	hit	hit	hit
Aluminum foil, shiny in	hit	no hit	hit	no hit
Aluminum foil, dull in	hit	no hit	hit	no hit

The motion sensors proved to be more consistent than I had previously believed. The standard deviation for both low (93° F) and medium (97° F) heat sources were at or near 2°. The average view angle for low heat was 147° by 52°. Medium heat had an average view angle of 152° by 56°. That is a 3% average increase in horizontal view by 7% average increase in vertical view.

3.3.4 Firing Rates

The oscilloscope images used to determine the length, quantity, and timing of the sensor firings can be found in Appendix A. Measurements can be taken on the images to determine timing sequences.

The first test is designed to find the length of a pulse, which contains all of the data necessary to interpret the command associated with the signal. The X10 protocol transmits 2 bytes (16 bits) by default. However this number can expand based on the system. The Hawkeye II sensors use an extended protocol that transmits 3 bytes. The transmission pulse (1 of 5 pulses in a packet) contains the command codes used to identify the sensor. The length of a single pulse is 70 milliseconds.

Sensor#	Length of single pulse	
B1	691 px	0.07s
B2	696 px	0.07s
B3	681 px	0.07s
B4	682 px	0.07s
B5	680 px	0.07s
B6	677 px	0.07s
B7	682 px	0.07s
B8	684 px	0.07s
B9	669 px	0.07s
B10	684 px	0.07s
Avg	683 px	0.07s
Std	7 px	0.07s

Table 3.3: The length of a single pulse is 70 ms, with a mere 0.69 ms standard deviation. One hundred pixels is equivalent to 0.01 seconds.

Table 3.4: The average transmission length is 0.51 seconds, with a 4.8 ms standard deviation. A transmission contains 5 pulses, with short pauses in between. One hundred pixels is equivalent to .1 seconds.

Sensor#	Length of five pulses	
B1	512 px	0.51s
B2	506 px	0.51s
B3	506 px	0.51s
B4	505 px	0.51s
B5	505 px	0.51s
B6	497 px	0.50s
B7	510 px	0.51s
B8	507 px	0.51s
B9	496 px	0.50s
B10	507 px	0.51s
Avg	505 px	0.51s
Std	5 px	0.51s

The second test determines the number of pulses in a transmission and measures the time to transmit (aka fire). Each transmission contains 5 pulses, with a pause between each pulse. Each transmission averages 505 ms.

The final test measures the sensor's cycle time. The cycle time is the time it takes for the motion sensor to transmit and pause before transmitting again. This may also be referred to as a sensor's refire rate. These times have a greater variance than the previous two tests, and so I have taken the long and short times of 5 trials. These two are averaged together, and then averaged over all 10 sensors. The average firing cycle is 7.06 seconds. The longest refire was 7.65 seconds and the shortest is 6.64 seconds.

Table 3.5: The average short, average long, sensor average, and overall average firing cycle times. One second is represented by 100 pixels.

Sensor#	Short		Long		Avg	
B1	664 px	6.64s	765 px	7.65s	715 px	7.15s
B2	692 px	6.92s	747 px	7.47s	720 px	7.20s
B3	691 px	6.91s	737 px	7.37s	714 px	7.14s
B4	673 px	6.73s	731 px	7.31s	702 px	7.02s
B5	688 px	6.88s	718 px	7.18s	703 px	7.03s
B6	695 px	6.95s	695 px	6.95s	695 px	6.95s
B7	671 px	6.71s	716 px	7.16s	694 px	6.94s
B8	655 px	6.55s	716 px	7.16s	686 px	6.86s
B9	684 px	6.84s	741 px	7.41s	713 px	7.13s
B10	700 px	7.00s	746 px	7.46s	723 px	7.23s
Avg	681 px	6.81s	731 px	7.31s		
Std	13 px	0.13s	18 px	0.18s		

Totals	Pixels	Seconds
Avg	706 px	7.06s
Std	30 px	0.30s

100 px = 1 seconds

3.3.5 Transmission Collision Potential

Determining the potential for transmission collision is used as part of the footprint of uncertainty.

To achieve a successful data transmission, only one of the five pulses has to avoid a collision. These pulses require 0.07 seconds, with an average firing cycle taking 7.06 seconds. Total transmission time, including all 5 pulses, is 0.51 seconds. The transmission range is between 50 and 100 feet. In Tigerplace, that range would include the sensor networks from about four apartments.

A quick empirical study showed that transmission collisions were not only possible, but probable. Predicting the number of collisions can be approached by building the collision rate, or subtracting the probability of no collision from 1. I have approached the problem as a best worst case scenario. For the sake of simplicity, I am only considering cases where only one sensor is being fired per apartment. There are many assumptions that could affect the probability of a collision:

- The sensors are independent of each other;
- The sensor firings are distributed over time;
- The sensor firings started at random;
- All of the apartments have active motion;
- Only one sensor per apartment is being active;
- The sensors have a constant and consistent refire rate of 7.06 seconds.

The first way to find the probability for collisions is to find the probability that no other sensors will fire during the first 0.07 seconds of the first. The probability of two sensors firing at the same time is $\frac{0.07}{7.06} = 0.0099$. These numbers will the basis for many of the following equations. 1-0.0099 = 0.99, so there would be a 99% chance that in a two sensor scenario there will not be a collision. The probability that no other sensors will fire in the 4 apartment scenario is around $0.99^3 = 0.97$. On the low end, there is a 3% chance that there will be a collision. This only considers complete collisions, so cases where more sensors fire within the 0.51 seconds will add to the collision rate.

The second way to find the probability for collisions is to build the collision rate. This requires a case structure, or tree, to determine the final probability. A sensor firing does not simply consume the 70 milliseconds; it will transmit during the entire 0.51 seconds. The following equation begins the probability calculation. It describes a situation where none of the other sensors will fire in the next half second.

$\left(\frac{7.06 - 0.51}{7.06}\right)^3 = 0.80$. So the probability of at least one collision within the 0.51 seconds is 20%. This is where a tree becomes necessary. From that 20%, the probability of a single collision during 0.51 seconds can be found using the formula $3*\left(\frac{0.51}{7.06}\right)*\left(\frac{7.06-0.51}{7.06}\right)^2 = 0.185$. The probability of two collisions is $3*\left(\frac{0.51}{7.06}\right)^2*\left(\frac{7.06-0.51}{7.06}\right) = 0.014$. The probability of three collisions is $3*\left(\frac{0.51}{7.06}\right)^3 = 0.001$.

It is now necessary to break down the cases. For the single case, initially 18.5% collision, so long as one sensor starts 70 milliseconds later then there will not be a collision. $\left(\frac{0.51-0.07}{0.51}\right) = 0.86$. 86% of 0.185 is 0.16, so 16% of the sensor firings will make it through with a 2.5% loss. In the three sensor collision case, there is a guaranteed loss. Three sensors fire within 0.51 seconds, so the middle sensor's data will collide with the end of the first sensor and the beginning of the last sensor. This is true for all of the following sensors as well. Totaling the losses, a final collision calculation is as follows: 0.025 + 0.014 + 0.001 = 0.04, or 4%. A tree representation is represented in Fig 3.10.

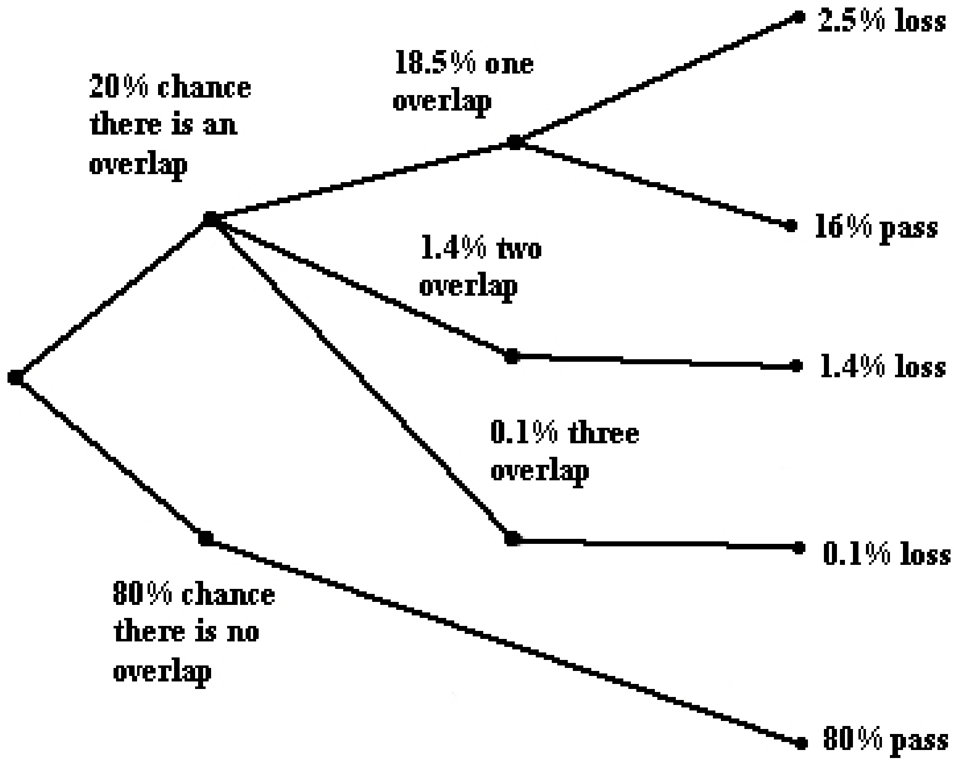


Fig. 3.10: Calculating the collision probability in a 4 sensor case

Again, these numbers are approximate. They require many assumptions, and are based around the conditions at Tigerplace. Apartments that have visitors, or have sensors overlapping more than three other apartments, will have a higher collision rate. The first technique is good for estimating the results, since the second method takes a little more computation.

3.3.6 Test: Approximating Velocity

The process of approximating velocity using motion sensors returns mixed results. The biggest source of errors comes from the data logger, which only records in whole seconds. Using a constant distance, as the velocity increases the accuracy decreases. Conversely, when either distance increases or velocity decreases the accuracy improves.

This is a good point to mention a separate sensor study. Other students have been working with a gate monitoring sensor. This sensor can detect a multitude of attributes including walking speed. Data collected by this group found that residents average between 65 cm/sec and 75 cm/sec. That translates to 2.13 ft/sec and 2.46 ft/sec respectively. A 92 inch distance, equivalent to 234 cm, begins to deteriorate in accuracy around the average resident velocity 65 to 75 cm/sec.

I performed 10 trials that compared velocity calculated using a stopwatch to velocity calculated using timestamps from the database. I simulated a hallway using strips of painter's tape. Using sensors that exhibited similar view angles, the motion sensors were spread equidistantly along the ceiling. I calculated the distance between the cones on the floor. The distance between the sensor's viewable areas was about 3 feet 10 inches. The results of these trials are shown in table 3.6. Figure 3.10 is a picture of the setup.

Table 3.6: Approximating velocity with a stopwatch compared to motion sensors B3, B4, and B7. 1 ft/sec = 30.48 cm/sec

Trial#	Watch Time	ft/sec	B3	B4	B7	B3 to B4	B4 to B7	B3 to B7	ft/sec 1	ft/sec 2	ft/sec 3
1	4.53s	3.53	0s	2s	3s	2s	1s	3s	1.92	3.83	2.56
2	5.91s	2.71	0s	2s	3s	2s	1s	3s	1.92	3.83	2.56
3	8.84s	1.81	0s	3s	4s	3s	1s	4s	1.28	3.83	1.92
4	9.81s	1.63	0s	3s	5s	3s	2s	5s	1.28	1.92	1.53
5	13.81s	1.16	0s	3s	7s	3s	4s	7s	1.28	0.96	1.10
6	16.93s	0.95	0s	4s	7s	4s	3s	7s	0.96	1.28	1.10
7	20.41s	0.78	0s	9s	11s	9s	2s	11s	0.43	1.92	0.70
8	23.41s	0.68	0s	4s	15s	4s	11s	15s	0.96	0.35	0.51
9	42.88s	0.37	0s	6s	20s	6s	14s	20s	0.64	0.27	0.38
10	51.09s	0.31	0s	9s	25s	9s	16s	25s	0.43	0.24	0.31

Fig. 3.11: The velocity approximation setup

Chapter 4

ALGORITHMS FOR DETECTING VISITORS

4.1 Data Extraction and Preprocessing

Motion sensor data is collected by a data logger and then transmitted in chunks to the storage server at Tigerplace. The data is then backed up on a university server every night. I am then able to download the dump file and load it into MySQL. I have written a query in Microsoft Access that accesses MySQL and processes the data log.

The database only contains three tables of interest for this algorithm. The first table contains location information. The second table maps the sensor location and type data to the log, which is the third table.

The user's id and the sensor's location information are drawn directly from the database, but six other fields are produced from the log table's time stamp. Microsoft Access has some great built in functions for handling date types. I begin by extracting the Year, Month, Day, Hour, and Minute information using built in functions. The last field converts the logger's time stamp to UNIX time.

Processing in UNIX time makes sense with the current system. UNIX time is defined to be the number of seconds since January 1, 1970, not including leap seconds. A simple conversion from Microsoft's DateDiff function to UNIX time is as follows:

1. Read a valid date from a database table (or write in a valid date)
2. Call DateDiff with three parameters
    a. 's' stands for seconds

b. #1/1/1970# is the date January 1, 1970. The hash marks are delimiters so the program knows that there is a date value inside.

c. [Date] is a reference call to the date from the table. This may be replaced with a constant date following the syntax described above.

3. DateDiff will calculate the difference in time from January 1, 1970 to the date pulled from the database, and return a value in seconds.

The function will look like this:

UNIX Time: DateDiff (‘s’,#1/1/1970#,[Date])

The information is then exported to a comma separated file (CSV), which is easier for Matlab to read. Matlab reads each field into a vector. The fields are combined into a cell matrix and sorted. The data is now ready to be processed into events.

4.2 Segmenting Time

Originally, I had created 15 minute blocks of time that would each be measured for visitor likelihood. This model looked similar to Wang’s work on motion density maps [1]. I quickly found that this was not a great way to approach the problem. The time stamp in the data log is specified to the second. That means the algorithm was quantizing 900 seconds of diverse action to one event. The 15 minute block implementation had also broken up any data that could help determine if a visitor remained in the apartment after the block.

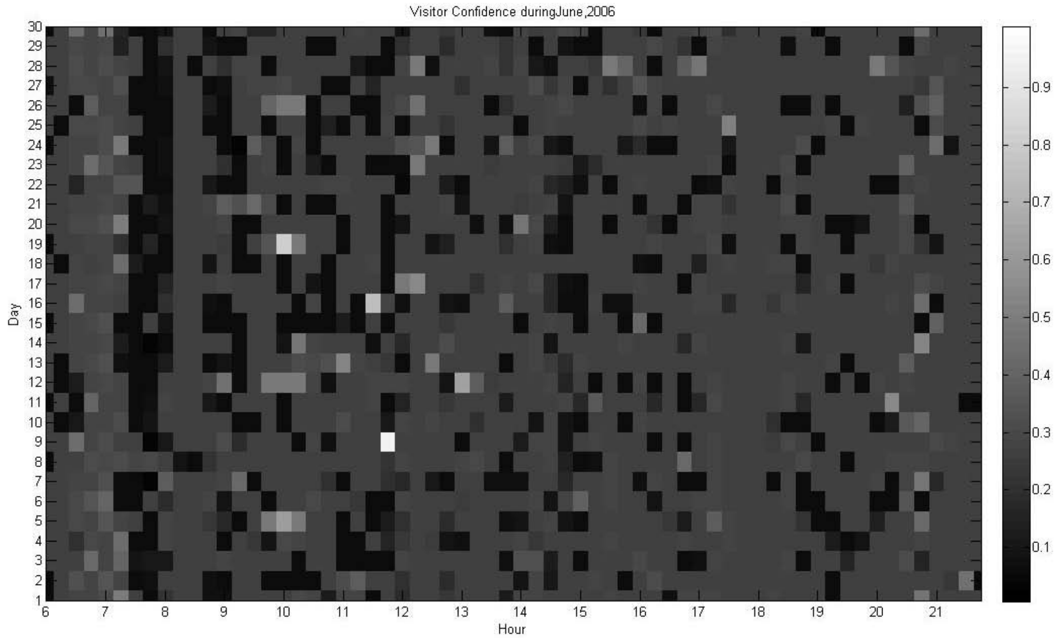


Fig. 4.1: Visitor confidence for every 15 minutes

4.2.1 Defining Events

If the visitor did not exit by the door, did the visitor actually leave? It sounds a little like the philosophical riddle "Tree in a forest." Chances are that if the visitor is going to leave, they will not exit through a window. Based on this assumption, a second version of the algorithm was designed to create events. I am using the term "event" as a period of time bounded by door sensors. The door sensors act as potential switch points, where a visitor may enter or leave the apartment.

Each event contains seven items:

1) startDoor: This is the UNIX timestamp of the door sensor firing that begins the event.
2) endDoor: This is the UNIX timestamp of the door sensor firing that ends the event.

3) duration: This is the number of seconds contained in the event.

   duration = endDoor – startDoor

4) occupantType: Each sensor location is assigned a privacy value from a finite set in the range [-1 1]. The occupantType value is an average of the event's privacy values.

5) hitRatePerHour: $\frac{hits\_in\_event}{duration\_of\_event} * 3600$. The calculation is the number of hits in the event, divided by the duration of the event in seconds, and then multiplied by 3600 seconds per hour. The maximum value is set to be 515 hits per hour (3600seconds/7seconds per firing cycle).

6) maxVelocity: Maximum velocity is calculated by first referencing a table of distances between the current sensor, and the last two sensors. The time differences between the current sensor and the last two sensors are then used to calculate the velocity. $\frac{distance(cm)}{seconds}$. Next, take the maximum of the two calculated velocities. The maximum velocity is not allowed to be greater than 200cm/sec.

7) confidence: The confidence is calculated using one of the eight classifiers: Function, Type-1 SISO, Type-1 MISO, Type-2 SISO No FOU, Type-2 MISO No FOU 2 Agents, Type-2 MISO No FOU 1 Agent, Type-2 MISO FOU 2 Agents, Type-2 MISO FOU 1 Agent.

All door sensor firings are potential switch points. However, when testing this version of the algorithm, I found that the motion sensor for the front door can view a large area. Some of the motion in the kitchen and dining area could be detected by the door sensor. In an effort to combat this, I introduced a smoothing algorithm.

4.2.2 Reducing Algorithmic Noise Near Event Boundaries

This algorithm was applied after the initial time segmentation and event creation. In a second (and again until convergence) look at the events, fragmented periods of time were looked at as likely sources of noise. Two parameters control the noise reduction.

The first parameter is the minimum event length. Minimum event length operates in two ways. Either two short events are combined to form a longer event, or a short event is combined with a long event. Both reduce, and eventually eliminate, events that contain fewer seconds than the minimum event length.

The second parameter is the confidence tolerance. The best way to describe the motivation for this parameter is to give an example. Suppose it is the middle of the night, and the senior gets out of bed for a drink of water from the kitchen. As the senior passes the front door to get to the kitchen, the door sensor is triggered. Now there is an event marker in the middle of the night.

If the door sensor were to be fired twice, the minimum event length parameter will likely combine the short event with one of the two longer events (bed time to drink, drink to bed time). However, that still leaves an event break in the middle of the night. In the algorithm, two adjacent events are compared. If the difference in confidence is less than the tolerance, they will be combined.

4.2.3 Combining the Contents of Multiple Events

Each event contains seven pieces of information:

1. Event Start Time in Unix time
2. Event End Time in Unix time
3. Duration of the Event in seconds
4. Occupant Type
5. Motion Density per hour
6. Maximum Velocity recorded
7. Confidence a visitor has been detected

Start and end times are pretty self explanatory. They are used primarily in the graphing algorithm, discussed in Section 4.9.2. When events are combined, the new event takes its start and end times from the appropriate parent events. The duration of the event, used as a feature for the confidence measure, is the sum of the time from both events.

The occupant type, clarified in Section 4.5, is combined as a weighted average.

$\frac{O1 * hits1 + O2 * hits2}{hits1 + hits2}$, where

1. O1 is the occupant type from the first event
2. O2 is the occupant type from the second event
3. hits1 is the number of hits from the first event
4. hits2 is the number of hits from the second event

The first step in finding the motion hits per hour of the new event is to find the number of hits associated with each of the parent events. To denormalize this field, the parent event's hits per hour (hph) are multiplied by the duration of the event and divided by 3600.

Number of event hits = $\frac{hph * duration}{3600}$

After calculating this value for both parent events, the hits can be added together and divided by the new event duration.

New hph = $\frac{hits1 + hits2}{total\_duration}$

While the maximum velocity is simply the max of the two parents, the new confidence is unlike the rest of the calculations. Now that all of the features have been combined, the confidence can be recalculated.

4.3 Motion Detection

Motion detectors are the only sensors required for this algorithm to operate. While this supports the project goals for an inexpensive noninvasive monitoring network, it also means that some assumptions must be made. The feature space has been built upon these assumptions:

1. All data are being logged
2. The sensors are operating correctly
3. The sensors have not been occluded
4. The sensors are in reasonable locations

4.3.1 Relating Motion Density to Visitors

Motion density shows the activity level for a given period of time. As more people enter the apartment, the density level is expected to rise. While motion density is a good indication of visitors, it does not tell the whole story. The motion density can also be used to detect some of the resident's routines, like the process of getting ready in the morning..

The morning and bedtime routines are both associated with a sleeping period overnight. Since there is very little motion activity during sleep periods, the overall motion density is very low. This reduces the false alarm rate that would otherwise occur when the resident is most active. Figure 4.2 is a typical motion density map. It is easy to see when the morning routine takes place from 6:00 AM to 8:00 AM.

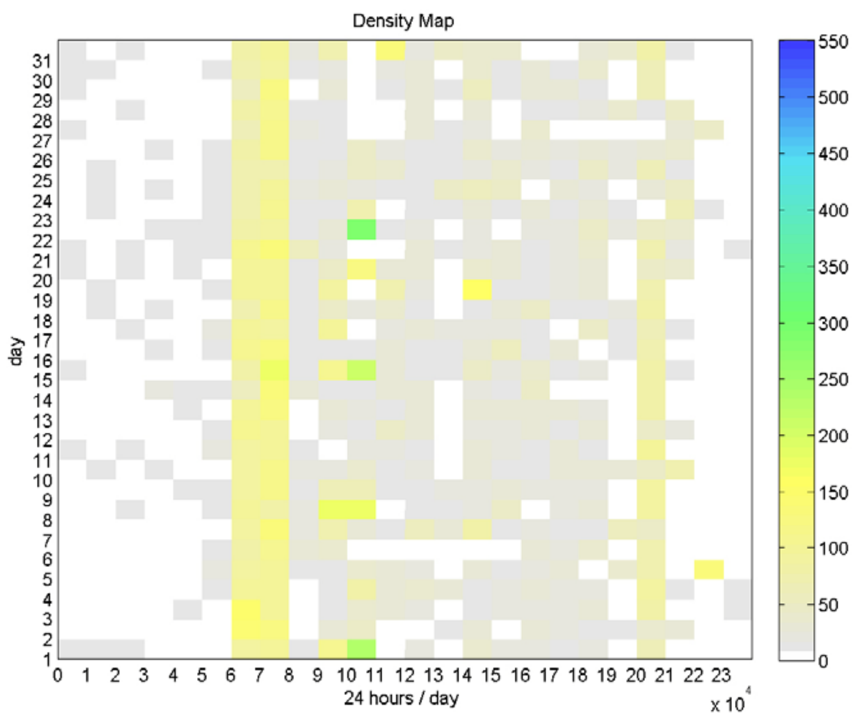


Fig. 4.2: A sample density map, produced by Wang's algorithm [1]

4.3.2 The Importance of Motion Sensor Location

There are two important points to consider when planning the locations for the motion sensors. The first is how well the sensors will be able to cover the space. That means reducing the area not covered, as well as reducing the areas in which sensor overlap occurs.

Adequate and independent coverage is not a trivial problem. For instance, as mentioned before, the door sensor overlaps some of the same area that the kitchen sensor covers. Baffles can assist in correcting this problem. Conversely, there is a limit on the number of motion sensors that can be used to cover the space. Transmission collisions can quickly become a problem, and the X10 sensors can only address up to 256 sensors inside the data logger's receiving range. So a reasonable placement solution is essential to any algorithm that uses motion.

The second important consideration relates to the velocity feature. As discussed in Section 3.3.6, the velocity measure improves with distance between sensors. The Tigerplace apartment layouts are relatively small, and well connected. There are no hallways that can be exploited for velocity calculations, so motion sensors must be set up in such a way that the coverage areas are separated by a reasonable distance. A sample layout is shown in Figure 4.3.

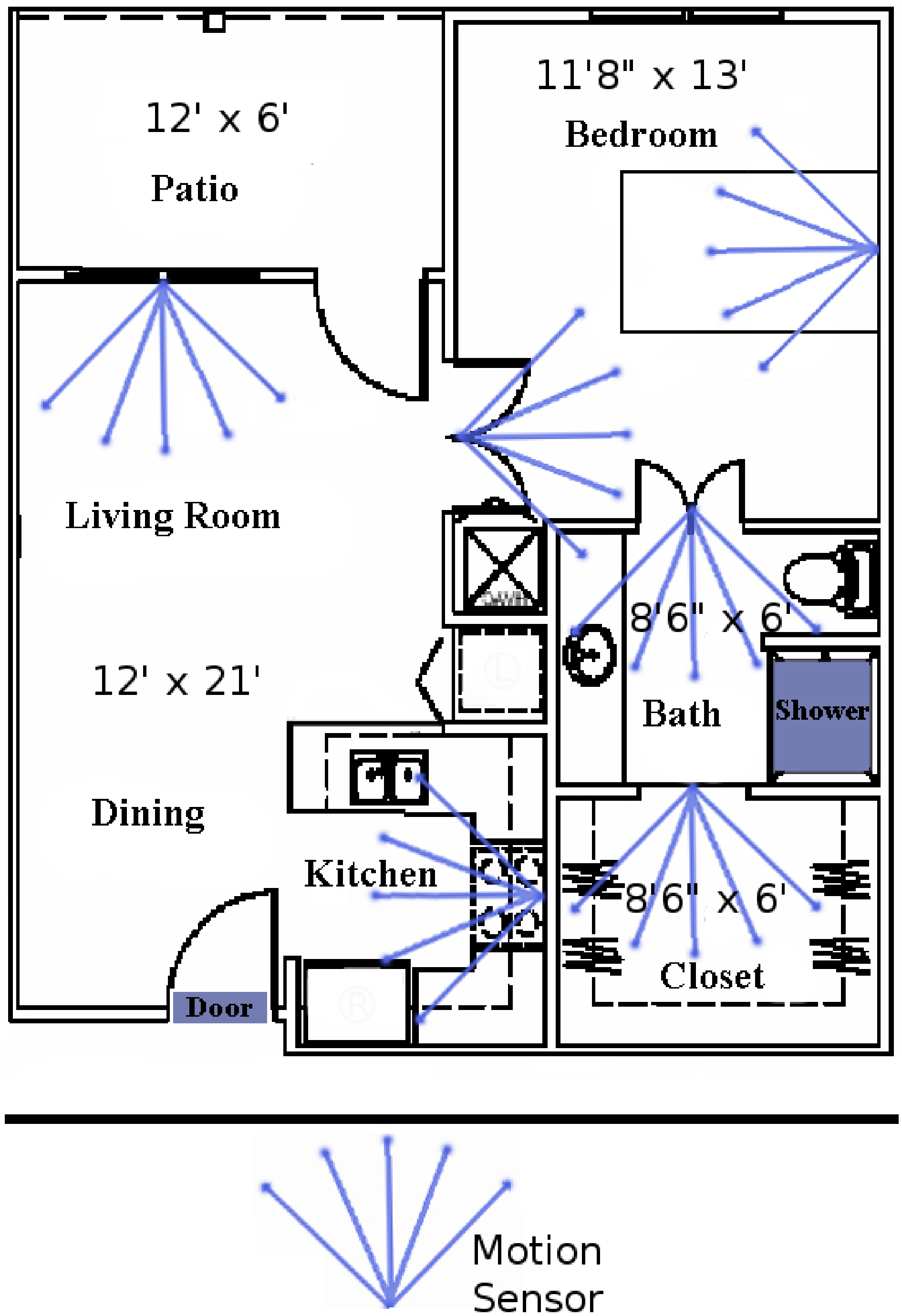


Fig. 4.3: Example motion sensor layout. Sensor coverage is in blue.

4.4 Approximating Velocity

The velocity feature handles two potential situations with respect to visitor detection: velocity, and multiple room occupation. The first step is to gather measurements from the apartment. Combinations that have near adjacent, adjacent, or overlapping motion sensors receive a distance of zero. By setting the distances to zero, the algorithm can filter out sensor sequences that have a high error potential. An example of a distance table can be found in table 4.1, which correlates to the floor plan in figure 4.3.

When there is only one person moving in the apartment, then the motion sensors should fire in order. Since there isn't any information that can be gathered by moving between adjacent sensors, the algorithm compares a sensor firing to both the previous firing and the one before that.

There are two reasons for checking consecutive motion sensor firing locations. The first is that there is a possibility a sensor malfunctions or has a transmission collision, discussed in Section 3.3.5. If the occupant moves to a nonadjacent room and back, and the data from the intermediate room is lost, then the time and distance of the nonadjacent room would be ignored. There is also the possibility that there are two occupants in the apartment. As I found in Section 3.3.4, the sensors fire every seven seconds on average. Two occupants may be in nonadjacent rooms, but their sensors may not be firing on the same second. Then the velocity becomes a measure of the likelihood of the two events relating to the same occupant. Larger distances indicate a higher likelihood that the sensors are detecting two entities. The maximum allowable velocity is 175 cm per second, unless sensors in two nonadjacent areas fire at the same time. If two nonadjacent sensors fire in the same second, a velocity value of 200 is assigned.

Table 4.1: Distances for the floor plan in figure 4.3

	Living Room	Kitchen	Bathroom	Bedroom	Front Door	Shower	Closet
Living Room	0 cm	0 cm	366 cm	183 cm	0 cm	488 cm	732 cm
Kitchen	0 cm	0 cm	732 cm	549 cm	0 cm	853 cm	914 cm
Bathroom	366 cm	732 cm	0 cm	0 cm	792 cm	0 cm	0 cm
Bedroom	183 cm	549 cm	0 cm	0 cm	610 cm	183 cm	366 cm
Front Door	0 cm	0 cm	792 cm	610 cm	0 cm	914 cm	975 cm
Shower	488 cm	853 cm	0 cm	183 cm	914 cm	0 cm	0 cm
Closet	732 cm	914 cm	0 cm	366 cm	975 cm	0 cm	0 cm

4.5 Inferring Additional Information

There is a small bit of information of interest that is not covered by visitor confidence. What types of activities are being performed in the apartment? I created a multivalued set that can be applied to the motion sensor hit from each location. The motion sensor locations are assigned a privacy value from a discrete set in the range [-1 1]. The percentage of time in various public or private areas of the home can help classify the occupant. For instance, the cleaning person spends a fairly equal amount of time in each room. Table 4.2 shows the values I used, which were determined empirically based on the floor plans at Tigerplace.

Table 4.2: Privacy values for potential rooms in a Tigerplace apartment

Room	Privacy Value
Living Room	1
Kitchen	1
Bathroom	-0.7
Bedroom	-1
Office	0
Laundry	-1
Front Door	1
Shower	-1
Bed	-0.7
Medicine Cabinet	-0.5
Dining Room	1
Den	-0.1
Closet	-0.5
Bathroom2	-0.7
Bedroom2	-1
Closet2	-0.5
Patio	0
Bed2	-0.7

This concept could also be extended to the percentage of time spent in individual rooms for each event.

4.6 Type-1 and Type-2 Membership Functions (Features)

There are two ways to look at a fuzzy classification problem. The first is to use a SISO, or single input single output, system that calculates the confidence for each feature individually. Each feature then receives a weight associated with that feature's ability to detect visitors. The final solution is the sum of the weighted feature confidences.

The second way to look at the problem is to have multiple membership inputs that combine to form the confidence. This solution only allows one output membership function, but the ability to introduce meaningful rules is a plus.

In order to objectively compare Type-1 and Type-2 implementations, the same membership functions were used. The only difference is that the Type-2 input and output memberships were fuzzified. I also tested a MISO style implementation using a function.

4.6.1 Duration of Event

The first membership function measures the duration of the event. This function is a type of measure that weights the relative value of motion density to the maximum velocity. A longer duration encourages a higher confidence in the averaged hit rate per hour.

An example of this would be when the resident is up in the middle of the night getting a drink of water. The door sensor would fire, starting an event, and then the resident would be detected by the kitchen and living room sensors. On the way back to bed, the resident triggers the door sensor again. Within a 30 second time frame, there have been 4 sensor hits. This would cause the motion density to indicate that there was a time of 480 hits per hour, a much higher value than the actual activity level merited. A lot of this problem will be eliminated with the minimum event length parameter, but using a threshold value does not completely solve the problem.

The duration membership function contains five Gaussian members. They approximate the following regions:

1. Very Short: from 0 to about 14 seconds. These are the most unreliable times for the motion density. Expect 2 or fewer hits.
2. Sparse: about 14 to about 100 seconds. Events in this region are still unlikely to contain visitors. Expect 14 or fewer hits.

3. Reasonable: about 100 seconds to 7 or 8 minutes. By 5 minutes the motion density is a reasonable estimation. However, there still are not going to be a lot of visitors that pop in for such a short time. Expect 70 or fewer hits.
4. Good: from about 7 to 8 minutes to around 15 minutes. This area constitutes short visits from friends and nurses. By this point, the hit rate average is going to be pretty reliable. It would be difficult to produce a large number of hits for 15 minutes from a resident's standpoint.
5. Excellent: Greater than about 15 minutes. This is going to be the most accurate average of the activity levels inside the event. General activity rates should be able to distinguish visitors from the resident.

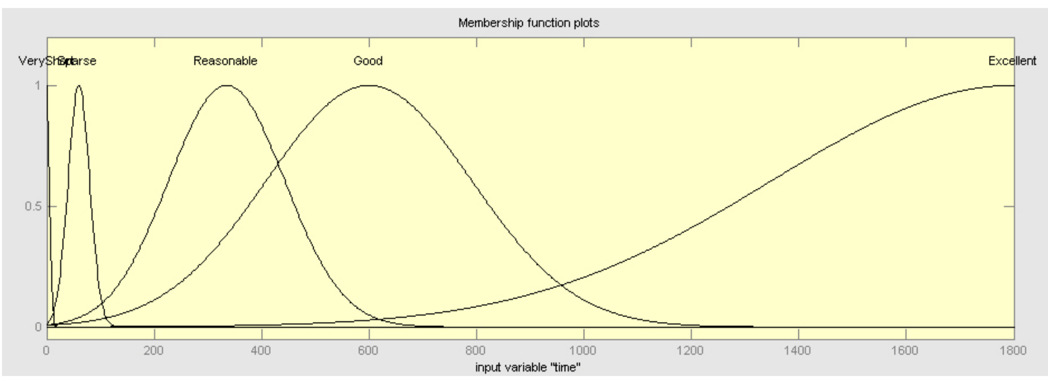


Fig. 4.4: The Type-1 duration membership, based on time in seconds

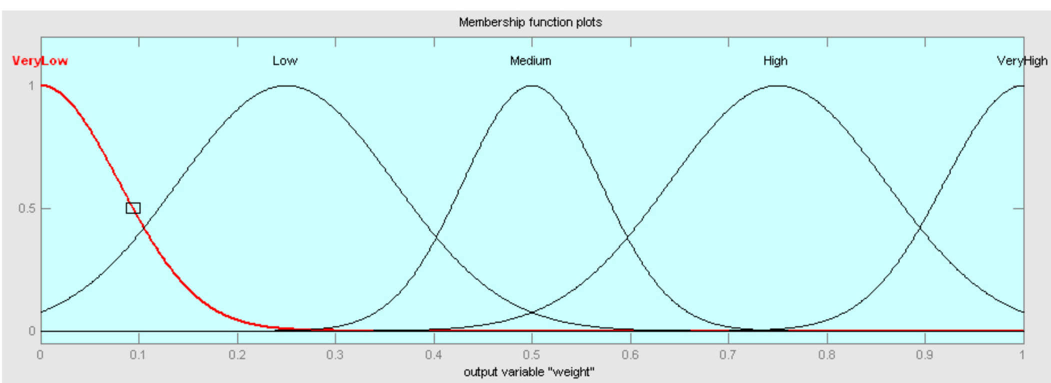


Fig. 4.5: The Type-1 duration output membership for the SISO system

4.6.2 Motion Density

As described in Section 4.3, a higher motion density represents more activity. Based on the calculation that a sensor fires every 7 seconds, the maximum number of times that a sensor could fire in one hour is 515. By studying the motion density plots produced by Wang [1], it can be seen that a resident typically will not produce more than 150 hits in one hour. Densities above 150 hits per hour are out of the ordinary, and so indicate an increased possibility that there are visitors present in the apartment.

The motion membership function contains five Gaussian members. They approximate the following regions:

1. Low: from 0 to about 50 hits per hour. Sedentary activities produce fewer than 50 hits.
2. Normal: from about 50 to about 225 hits. These correlate to the active times of a resident. There are generally not a lot of room changes, and the resident typically stays in the apartment for longer periods of time.
3. Increased: from about 225 hits to about 375 hits. This is an unusually large number of sensor hits. This corresponds with the kind of activity levels observed in the motion density plots when the cleaning person is in the apartment. Some of the density increase may be attributed to room changes (no seven second cycle time).
4. High: from about 375 hits to about 475 hits. This is a large number of hits, and is approaching the maximum number of sensor hits that could be produced by a single sensor. This is indicative of several room changes and likely a second occupant.

5. Very High: more than about 475 hits. This represents a very high activity level. There are likely multiple occupants in separate rooms.

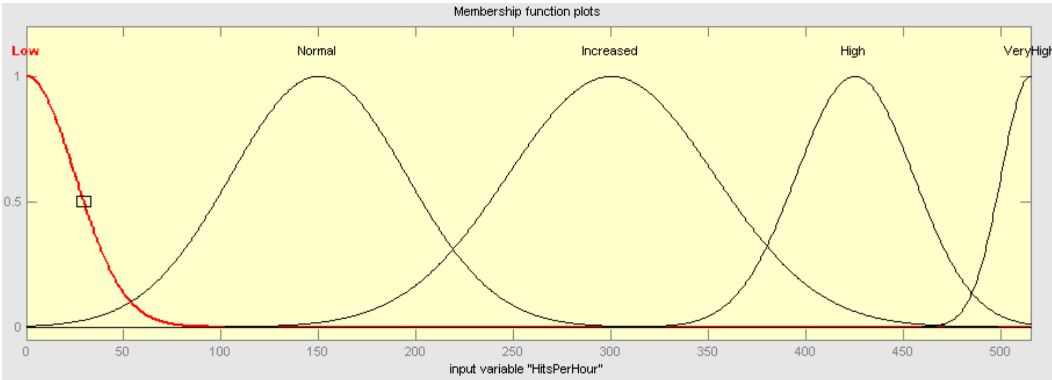


Fig. 4.6: The Type-1 motion membership, based on time in seconds

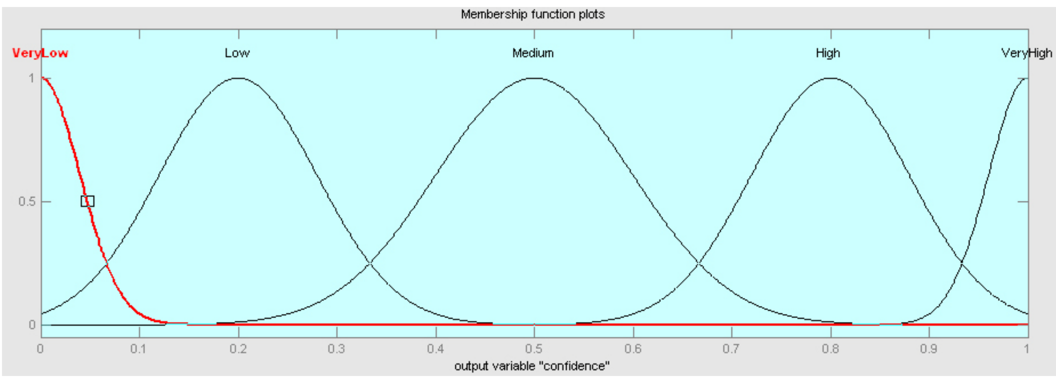


Fig. 4.7: The Type-1 motion output membership for the SISO system

4.6.3 Approximate Velocity and Multiple Occupants

Section 3.4 describes two purposes for the velocity measure. Readings on the gate monitoring sensor showed that residents average between 65 and 75 cm per second. Those numbers are used to calculate the mean of the normal member of the velocity function. Slower velocities are expected, since there are very few places in the apartment that allow for a purposeful forward walking motion.

The velocity membership function contains five Gaussian members. They approximate the following regions:

1. Slow: 0 to about 50 cm per second. 50 cm per second is a little over one mile per hour. This represents typical unhurried activities.
2. Normal: about 50 to about 90 cm per second. The normal region describes a velocity between 1 and 2 miles per hour. This is a normal speed for the active times of the resident's morning routine.
3. Increased: about 90 to about 130 cm per second. This is an above average level of speed for the resident. It is approximately 2 to 3 miles per hour.
4. Fast: about 130 to 175 cm per second. Approximately 3 to 4 miles per hour.
5. Super Fast: about 175 to 200 cm per second. Approximately 4 to 4.5 miles per hour. If no time has elapsed and two sensors fire in nonadjacent rooms, a value of 200 cm per second is assigned. Otherwise, the limit is 175 cm per second.

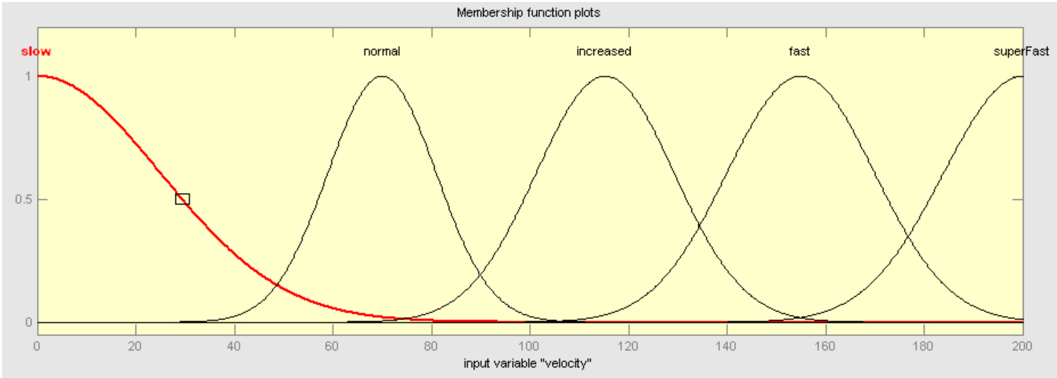


Fig. 4.8: The Type-1 velocity membership, based on time in seconds

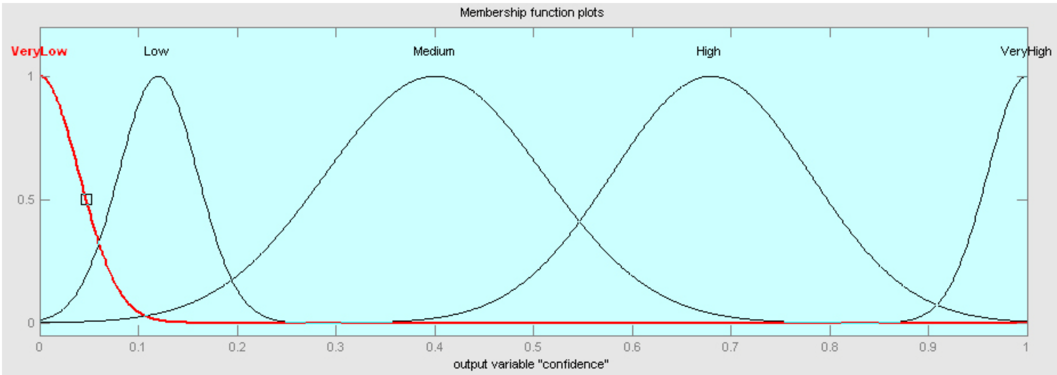


Fig. 4.9: The Type-1 velocity output membership for the SISO system

4.7 The Rule Base

The SISO output membership functions are displayed above in figures 4.5, 4.7, and 4.9. They relate to Duration, Motion, and Velocity respectively and appear in blue. Their rules are all very simple, and are shown in Table 4.3.

Table 4.3: 15 rules for SISO type visitor confidence

Velocity	Motion (hph)	Duration	Consequent (output) Confidence/Weight
Slow			Very Low
Normal			Low
Increased			Medium
Fast			High
Super Fast			Very High
	Low		Very Low
	Normal		Low
	Increased		Medium
	High		High
	Very High		Very High
		Very Short	Very Low
		Sparse	Low
		Reasonable	Medium
		Good	High
		Excellent	Very High

To complete the confidence measure of the SISO systems, the output from the duration value is applied to the motion and velocity outputs.

$$confidence = weightOut * motionOut + (1 - weightOut) * velocityOut$$

In the three input case, the fuzzy systems handles the duration internally as part of the rule base. There are 30 rules. Five of the rules build a base confidence from the velocity measure. The other 25 represent values derived by applying the five duration steps to the five motion densities. While the velocity and motion features are directly related to the visitor confidence, the duration feature acts as a weight. The duration feature determines the confidence of the system in the motion density feature (hph). The rules can be found in table 4.4.

Table 4.4: 30 rules for MISO visitor confidence

Velocity	Motion (hph)	Duration	Consequent (output) Confidence/Weight
Slow			Very Low
Normal			Low
Increased			Medium
Fast			High
Super Fast			Very High
	Low	Very Short	Very Low
	Low	Sparse	Very Low
	Low	Reasonable	Very Low
	Low	Good	Very Low
	Low	Excellent	Low
	Normal	Very Short	Very Low
	Normal	Sparse	Very Low
	Normal	Reasonable	Low
	Normal	Good	Low
	Normal	Excellent	Low
	Increased	Very Short	Low
	Increased	Sparse	Low
	Increased	Reasonable	Medium
	Increased	Good	Medium
	Increased	Excellent	High
	High	Very Short	Low
	High	Sparse	Low
	High	Reasonable	Medium
	High	Good	High
	High	Excellent	Very High
	Very High	Very Short	Low
	Very High	Sparse	Medium
	Very High	Reasonable	High
	Very High	Good	Very High
	Very High	Excellent	Very High

4.8 Defining the Footprint of Uncertainty

The Footprint of Uncertainty (FOU) is the parameter that fuzzifies the endpoints of Type-2 membership functions. The characterization tests on the motion sensors (sections 3.3.x) are used to model the FOU. I was not able to find an example of this in the literature, and so is a novel way to approach the calculation of the FOU. There are 4 membership functions that need to be converted to Type-2 membership functions.

The first is the HitsPerHour feature (hph). I began by calculating the largest and smallest number of maximum possible hits per hour. The largest maximum hits per hour disregards the collision potential (section 3.3.5), and uses the average minimum firing cycle rate (section 3.3.4).

$$\left\lceil \frac{60\min * 60\sec}{6.81\sec/\,cycle} \right\rceil = 529hph\ .$$

To calculate the smallest maximum hits per hour uses the collision potential as well as the average maximum firing cycle rate. First, calculate the rate without collisions.

$$\frac{60\min * 60\sec}{7.31\sec/\,cycle} = 492.48hph\ .$$

Next, apply the collision potential.

$$\lfloor 492.48 - 492.476 * .04 \rfloor = 472hph\ .$$

To finish the FOU calculation, find the difference.

$529hph - 472hph = 56hph$ is the FOU associated with the motion density.

The second FOU to be calculated is the velocity. The velocity FOU is generalized on the empirical data associated with the velocity tests in Table 3.6, section 3.3.6. In addition to the average error, the view angle differences vary by a few degrees. The view angle standard deviation represents an uncertainty in the distance between the sensors viewable areas. The FOU is calculated by using the final uncertainty, 15%, multiplied by the maximum velocity.

$0.15 * 200cm/\sec = 30cm/\sec$ is the FOU associated with the velocity feature.

Table 4.5: Average velocity error over 350 cm distance

Trial#	Stopwatch ft/sec	Sensor ft/sec	Error
1	3.53	2.56	27%
2	2.71	2.56	7%
3	1.81	1.92	6%
4	1.63	1.53	6%
5	1.16	1.10	5%
6	0.95	1.10	16%
7	0.78	0.70	10%
8	0.68	0.51	25%
9	0.37	0.38	3%
10	0.31	0.31	0%
Avg			10.5%

The FOU for the duration is a simple selection. I use a 28 second FOU, based on the 7 second cycle time. I chose this number because the minimum event length parameter makes the input membership value at least a “Good Length” at 10 minutes. Twenty-eight seconds is around a 5% uncertainty (4.7%).

The last FOU is for the output membership functions. There is no great way to calculate this FOU, but the final range of values will be in the range [0 1]. I selected a 5% uncertainty, 0.05.

4.9 Implementation

This algorithm is run in Matlab 7.6.0. For the Type-1 system, I modified the fuzzy toolbox to use a higher resolution when performing calculations on the memberships. This can be accomplished by searching the text in the fuzzy toolbox for the number 171, which is the hard coded resolution value, and updating it to 1000 (the resolution you want).

The Type-2 systems are implemented using a set of Matlab files available on Mendel's website [30]. The same means (peaks of the memberships) and standard deviations are used for the Type-1 and Type-2 systems. The FOU's are input as parameters, with a value chosen above and below to make sure that there is not any obvious gradient information that produces better solutions. This is especially significant for the duration and output membership functions, which are not strongly modeled in the lab tests.

One difference is made between the Type-1 MISO and Type-2 MISO systems, in that Mendel's Matlab files do not allow the user to input a rule base that does not use all of the rules. Consequently, one solution is to approximate by averaging the velocity rule outputs with the motion density and duration rule outputs (two agents). Since neither the velocities nor the density/duration pairs are dependant on each other, this appears to be a reasonable solution.

On account of the defuzzification being a little different, I have run the Type-2 MISO 2 Agents system with and without an FOU. By setting the FOU to zero for each of the membership functions, it is effectively a Type-1 system. I refer to this implementation as the Type-2 MISO No FOU 2 Agents system because the interval

Type-2 software was used to produce the results. The corresponding system with FOU is called the Type-2 MISO FOU 2 Agents system.

A more traditional approach to this problem is to replace unused memberships with flat functions. In this case, the Gaussian membership functions have a standard deviation of infinity. The flat function is reduced out of the equation when minimized with a non-flat membership function. This approach is also broken into FOU and No FOU cases: Type-2 MISO FOU, and Type-2 MISO No FOU.

The remaining fuzzy classifiers are Type-1 SISO, Type-2 SISO (No FOU), and the functions. The Type-1 SISO system is a product of the fuzzy toolbox, and the Type-2 SISO system is implemented in Mendel’s Matlab files.

4.10 Function Approximation

As a sixth option, I made a simple function that approximated the concepts employed in the SISO fuzzy systems. Multiple inputs are combined together to form a final confidence. Each of the functions is a normalized quadratic.

$$\text{Motion} = \frac{Min(hitsPerHour,515)^2}{515^2}$$

$$\text{Velocity} = \frac{Min(velocity,200)^2}{200^2}$$

$$\text{Weight} = \frac{Min(duration,1800)^2}{1800}$$

$$\text{Confidence} = Weight * Motion + (1 - Weight) * Velocity$$

4.11 Understanding the Results

It is not enough to simply produce a confidence value, without knowing how it performs. There are a couple of ways I looked at the data. The first is by implementing a ROC curve. The second is a display output, similar to the motion density maps. The display can be visually inspected for known visitor times, and as a product, is used to show visitor interaction.

4.11.1 ROC Curves

ROC stands for Receiver (or Relative) Operating Characteristic. While the name is somewhat cryptic, it represents the percentage of true positives to false positives. The information can be used two ways. By plotting the true positives to false positives, a measurement of the area under the curve can be taken. If the true positive rate is known, or if an acceptable false positive rate can be established, two implementations can be compared equally. Otherwise, a larger area represents a better classifier.

4.11.2 Displaying the Data in a Meaningful Way

I took inspiration from the motion density maps when designing the data view. The graph is laid out in a grid. The rows contain the data from each day, while the columns contain the times of day.

The visitor confidence information I calculate is quantized to the second. Not wanting to throw out the additional information, I display the confidences in seconds rather than hours. Another difference with my implementation is that the confidence display blocks are higher or lower, dependant on if the visitor type was public or private (see Section 4.5). A line runs across the center of each day to make the visitor type easier to read. Fig. 4.10 shows an example of the display data.

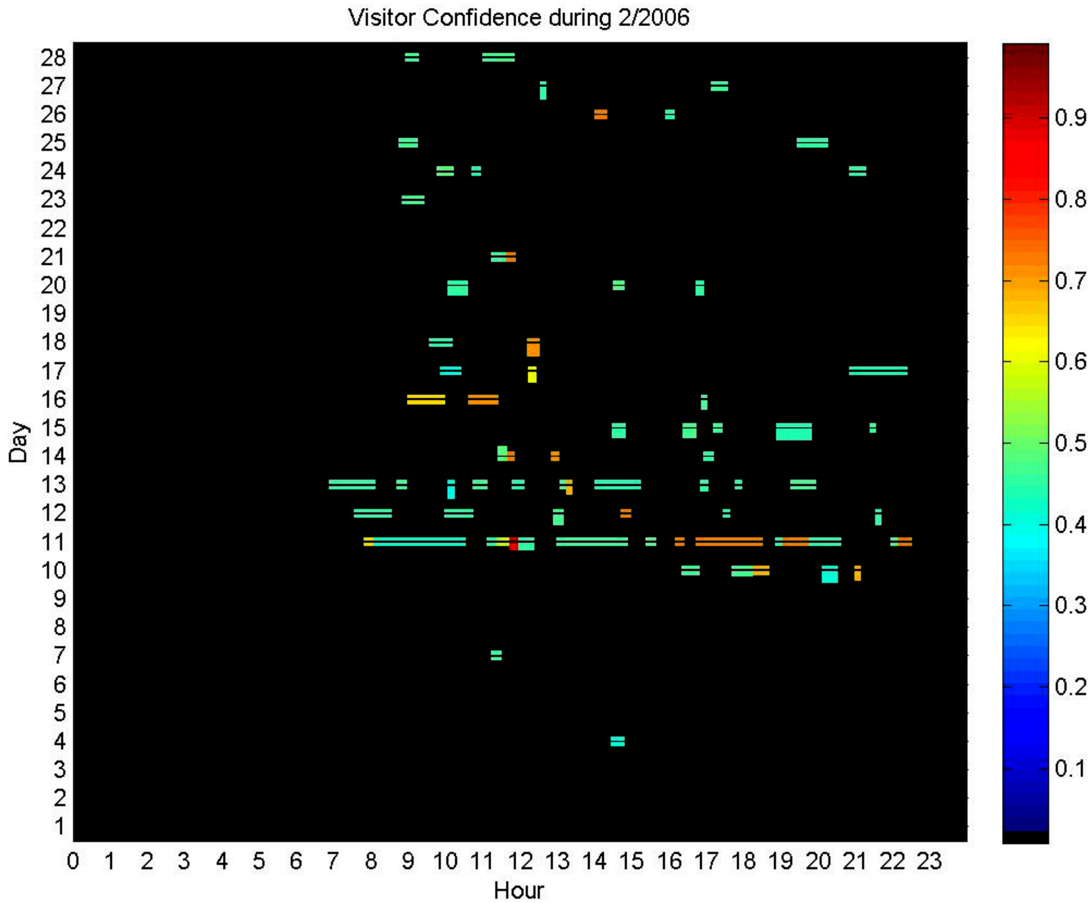


Fig. 4.10: An example of the visitor confidence display output

Chapter 5

THE TEST APARTMENT

5.1 Purpose of the Test Apartment

To measure the effectiveness of a system, it must be tested in a controlled environment. To this end, a test apartment was set aside at Tigerplace for students to work on their algorithms.

Nine motion sensors were installed along with a data logger. The sensors were placed in each area of the home, in a typical configuration shown in Fig. 5.1.

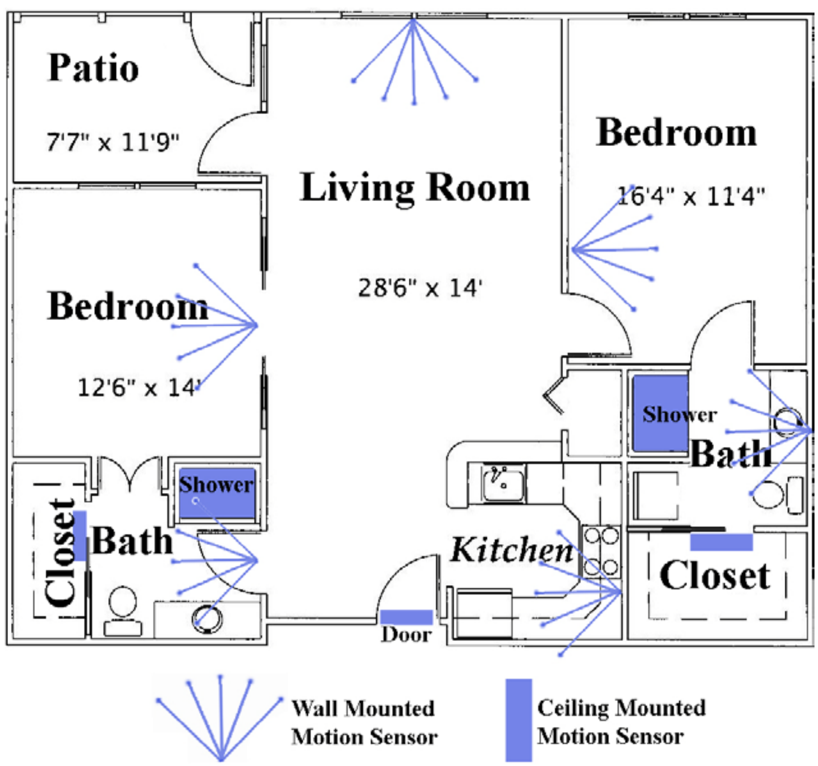


Fig. 5.1: The test apartment layout

5.2 Organization of the Tests Performed

A series of tests were performed in the test apartment at Tigerplace. The test apartment was a new model, containing two bedrooms. In the case of a visitor and a resident, Ian Kable was recruited to act as the resident. All tests began with the visitor, me, out of the apartment.

1. Series 1: An active visitor comes to the apartment without the resident. The visitor opens the door.
    a. Housekeeper: 15 minutes of high activity, while working in all rooms of the apartment.
    b. Tech Person: 15 minutes of working on the data logger and the sensor network. Generally in the public areas of the home.
2. Series 2: An active visitor comes to the apartment, which is occupied by the resident. The visitor opens the door.
    a. Housekeeper: 15 minutes of high activity, while working in all rooms of the apartment. The resident converses with the visitor and reads the paper.
    b. Paid Helper: 15 minutes of medium activity in the public areas of the home. The resident converses with the visitor and is generally inactive.
    c. Tech Person: 15 minutes of working on the data logger and the sensor network. Generally in the public areas of the home.
    d. Other Resident: 15 minutes of conversation. The residents move some about the public areas of the apartment. The residents stay in the same room together.

e. Family: 15 minutes of conversation interspersed with other activities, such as going into the bathroom alone.

3. Series 3: An active visitor comes to the apartment, which is occupied by the resident. The resident opens the door. All other factors from the test are the same as those from Series 2.
4. Series 4: A sedentary visitor comes to the apartment, which is occupied by the resident. The visitor opens the door.
   a. Other resident: A slow entry into the apartment, followed by 15 minutes of conversation in the living room. The other resident then exits at a slow pace.
   b. Family: A quick entry into the apartment, followed by 15 minutes of conversation in the living room. The family member moves quickly (relatively) when leaving the apartment.
5. Series 5: A sedentary visitor comes to the apartment, which is occupied by the resident. The resident opens the door. All other factors from the test are the same as Series 4.
6. Series 6: Neither a visitor nor the resident is in the apartment.
7. Series 7: The resident is alone, but active.
   a. The resident cleans up around the kitchen area.
   b. The resident cleans up around the living room.
   c. The resident performs their morning routine in the bathroom.
   d. The resident performs their morning routine in the closet.
   e. The resident performs their morning routine in the living room.

f. The resident prepares to leave the apartment.

g. The resident exercises in the room or dances. This particular test is from a story told by a member of the research group. The resident had asked him if we could see that she danced for exercise every morning.

h. The resident prepares for bed.

i. The resident wakes up in the night to use the restroom.

8. Series 8: The resident is alone and sedentary.

   a. The resident is reading the paper.

   b. The resident is watching television.

   c. The resident is taking a nap in the chair.

   d. The resident is using the computer.

Each series of tests was designed to answer a specific question, or to try to classify border cases. Series 1 contains a single occupant other than the resident. I consider this visitor ground truth because the algorithm is designed to detect activity levels and velocities that do not match the resident. Series 2 tests the presence of the resident with an active visitor, which is the ideal visitor ground truth. Series 3 is the same as Series 2, except that the resident opens the door. Series 4 and 5 are borderline cases where both the resident and their visitor are sedentary for visitor ground truth. Series 6 is simply to verify that the algorithm has no visitor confidence during times when there is no one in the apartment. Series 7 is a borderline case where the resident is particularly active, and is a no visitor ground truth. Series 8 is no visitor ground truth for times when the resident is sedentary.

Chapter 6

TEST APARTMENT RESULTS AND KNOWN USER DATA

6.1 Test Apartment Results and Analysis

Each test series from section 5.2 is performed once to generate the ground truth data. While at first they appear to represent a small sample size, the algorithm operates on a second by second basis. There are 22,775 seconds in the ground truth data. The visitor ground truth has 14,060 seconds (61.7%), and the no visitor ground truth has the remaining 8,715 seconds (38.3%).

The first set of tests compares the minimum event lengths. The function implementation is used in this test, because it has a shorter computation time. The area underneath the ROC curve is measured without noise reduction (second by second), as well as for 1 thru 20 minute minimum event lengths, shown in figure 6.1. A second test measures the maximum accuracy for the same sets, shown in figure 6.2. These tests help qualify the various minimum event lengths, so that an informed decision can be made on which minimum length value to use.

Since the first tests did not provide any conclusive data, three minimum event length samples are chosen for the test apartment data: 5 minutes, 10 minutes, and 15 minutes. The relative performances are shown for ROC curve area in figures 6.3, 6.6, and 6.9. The corresponding maximum accuracy data can be found in figures 6.4, 6.7, and 6.10. The ROC curves themselves are in figures 6.5, 6.8, and 6.11. The point that corresponds with the maximum accuracy (the confidence level) is marked by a red dot. The test apartment tests were all around 15 minutes long, which biases the 15 minute minimum event length results. In the residents' apartment data, the 10 minute minimum event length is found to be preferential, because visitor activity shorter than the minimum event length is at risk for going undetected. The 15 minute event length seems to be too long, based on empirical evidence.

The ROC curves appear blocky for a couple of reasons. The major reason for the blocky appearance is that the minimum event length often creates a constant confidence across the individual tests (bounded by red and blue boxes in figures 6.12 thru 6.27). The other reason for the blocky ROC curves is the limited number of trials. Repeating each of the test series would provide smoother ROC curves for the data.

For reference, I have also included the confusion matrices for each of the classifiers and minimum event lengths. The confusion matrices are based on the confidence level associated with the best classification accuracy. Each set of confusion matrices are on the page following the ROC curves.

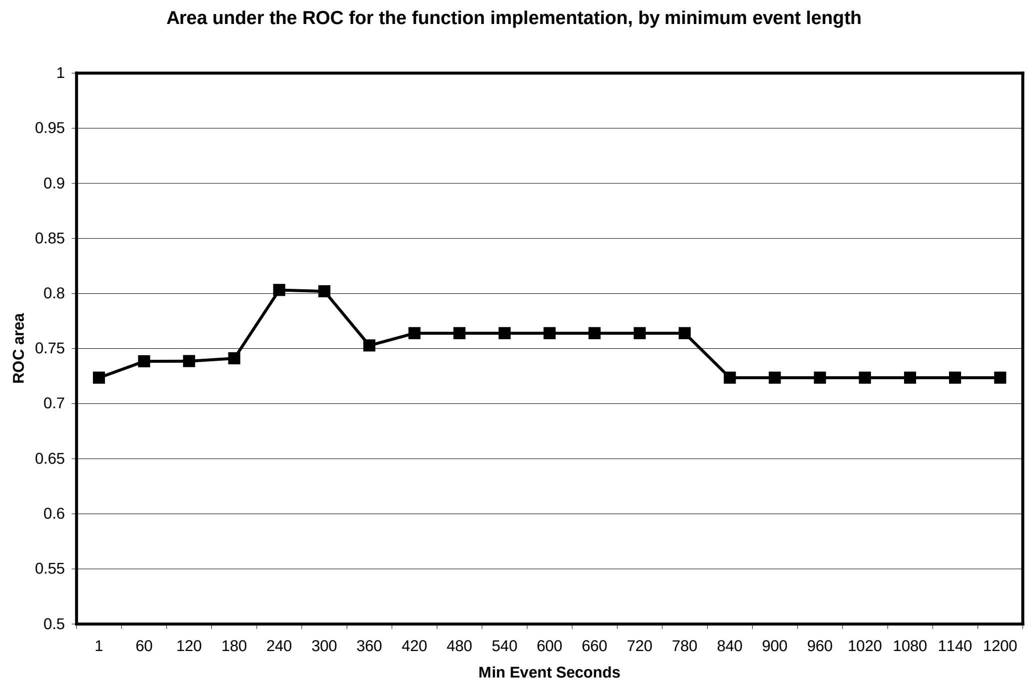


Fig. 6.1: ROC curve area by minimum event length for the function classifier

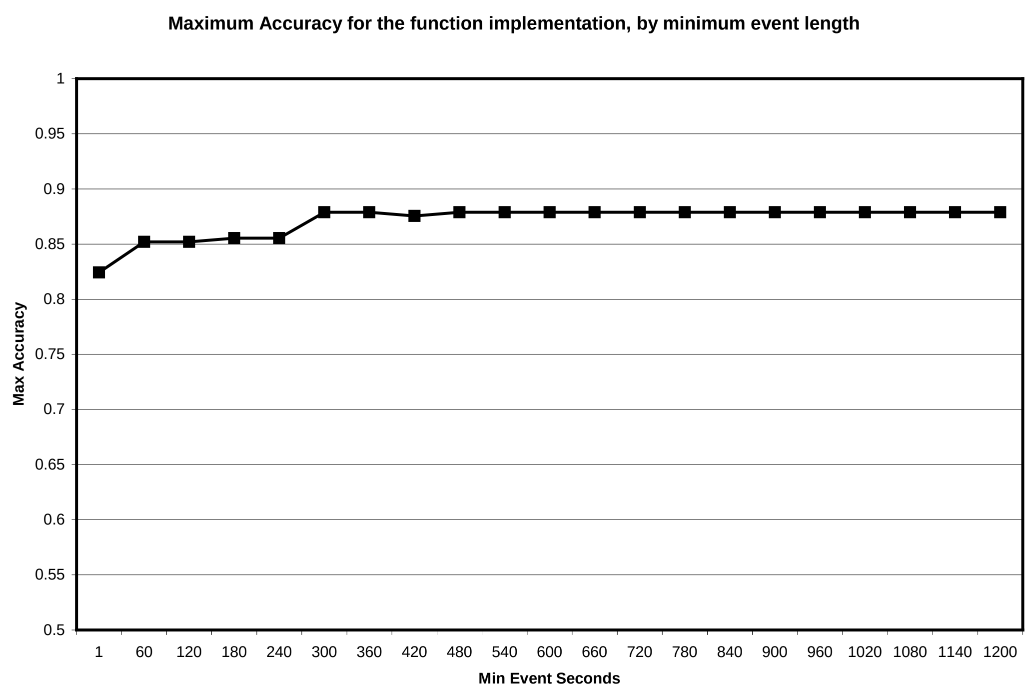


Fig. 6.2: Accuracy by minimum event length for the function classifier

Classification Rates by ROC area at 5 minutes

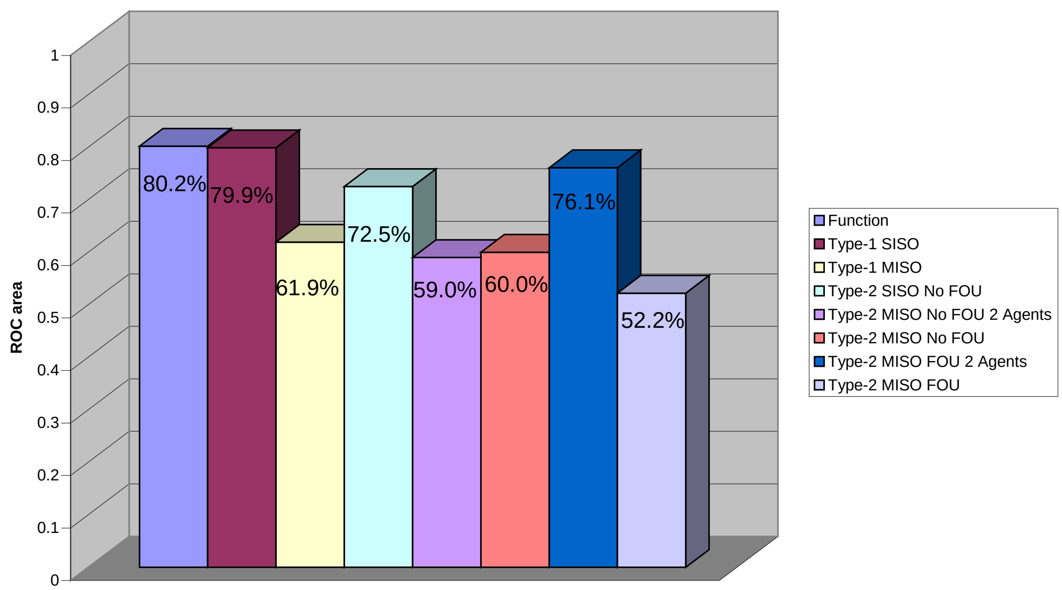


Fig. 6.3: ROC areas of the 8 classifiers for 5 minute minimum event length

Classification Rates by Accuracy at 5 minutes

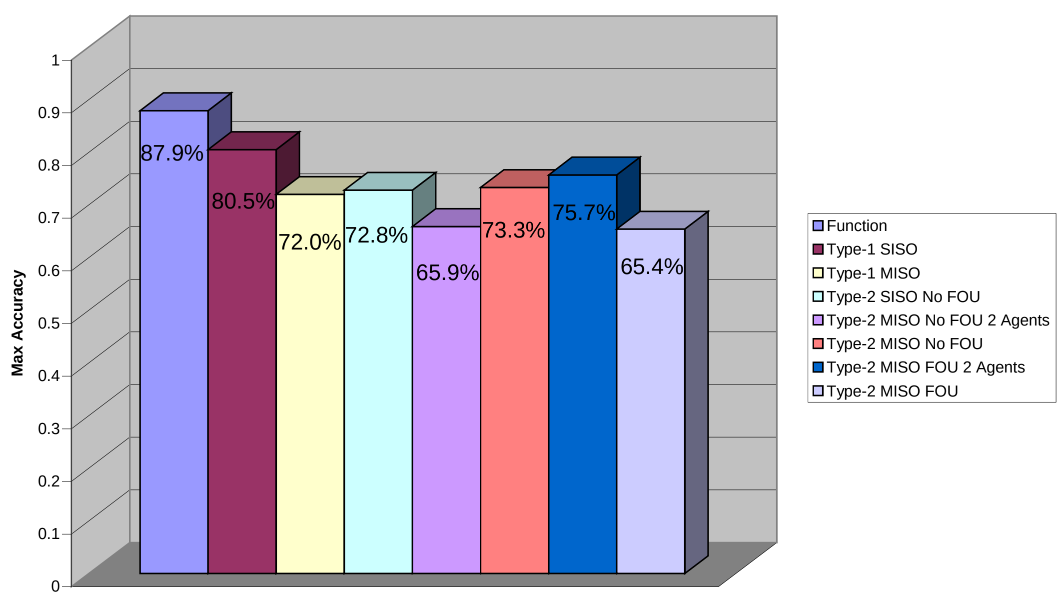


Fig. 6.4: Maximum accuracy of the 8 classifiers for 5 minute minimum event length

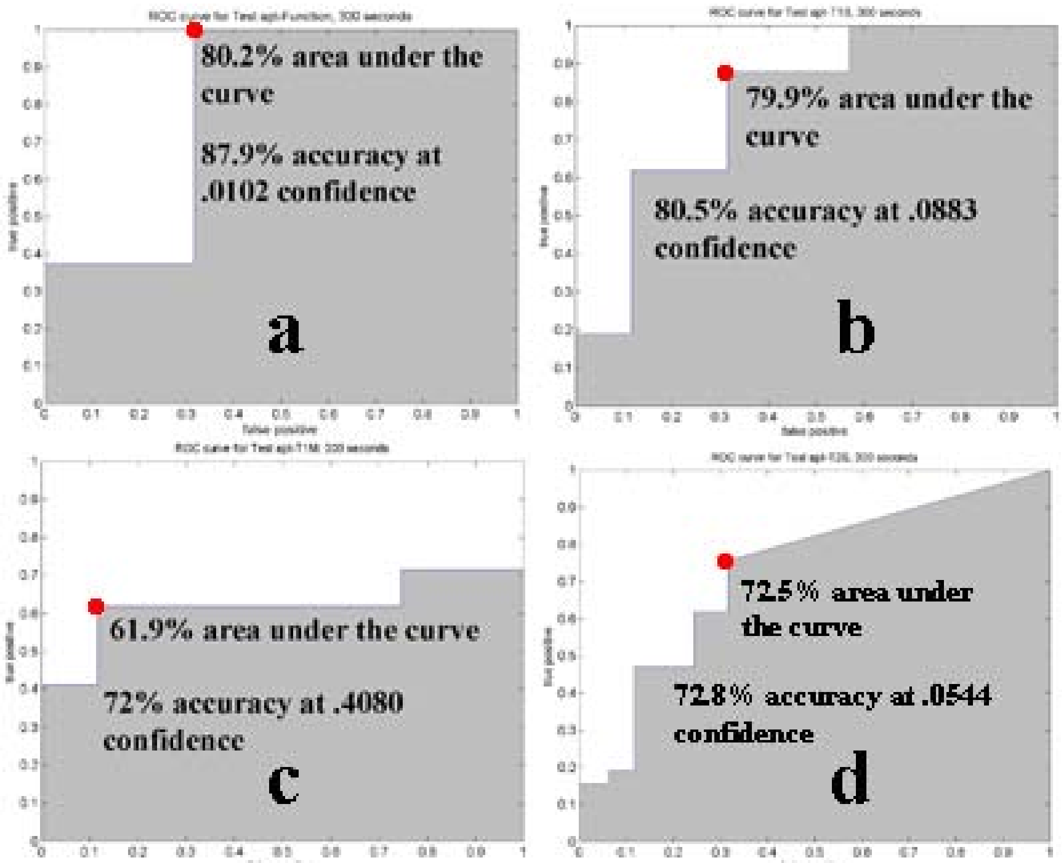


Fig. 6.5: ROC curves for the 5 minute minimum event length implementations.
The red dots mark the places where the confidence maximizes accuracy.
a) Function, b) Type-1 SISO, c) Type-1 MISO, d) Type-2 SISO No FOU,
e) Type-2 MISO No FOU 2 Agents, f) Type-2 MISO No FOU 1 Agent,
g) Type-2 MISO FOU 2 Agents, h) Type-2 MISO FOU 1 Agent

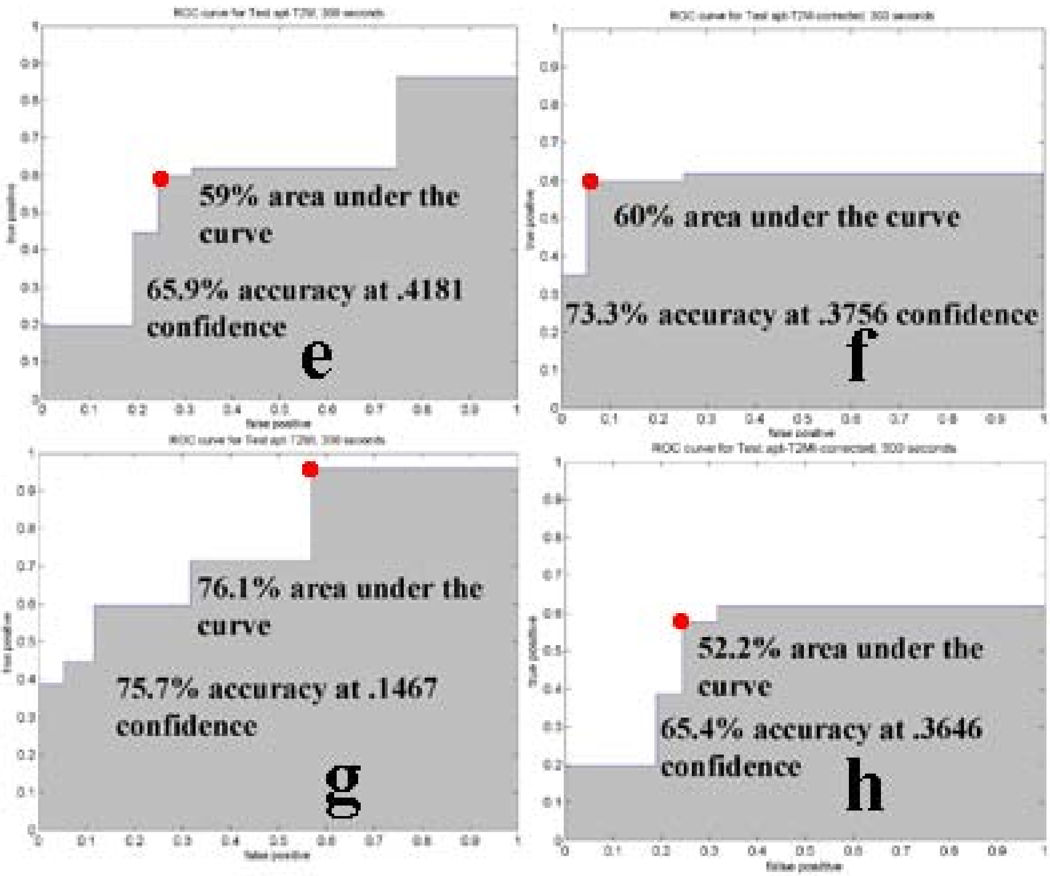


Fig. 6.5: ROC curves for the 5 minute minimum event length implementations.
The red dots mark the places where the confidence maximizes accuracy.
a) Function, b) Type-1 SISO, c) Type-1 MISO, d) Type-2 SISO No FOU,
e) Type-2 MISO No FOU 2 Agents, f) Type-2 MISO No FOU 1 Agent,
g) Type-2 MISO FOU 2 Agents, h) Type-2 MISO FOU 1 Agent

Table 6.1: Confusion matrices for the 5 minute minimum event length classifiers
T1S = Type-1 SISO, T1M = Type-1 MISO, T2S = Type-2 SISO No FOU,
T2M 2A = Type-2 MISO No FOU 2 Agent, T2M 1A = Type-2 MISO No FOU 1 Agent
T2MF 2A = Type-2 MISO FOU 2 Agent, T2MF 1A = Type-2 MISO FOU 1 Agent

Function	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T1S	Visitor	No Visitor
TRUE	88.0%	31.7%
FALSE	12.0%	68.3%

T1M	Visitor	No Visitor
TRUE	61.9%	11.7%
FALSE	38.1%	88.3%

T2S	Visitor	No Visitor
TRUE	75.6%	31.7%
FALSE	24.4%	68.3%

T2M 2A	Visitor	No Visitor
TRUE	60.0%	24.5%
FALSE	40.0%	75.5%

T2M 1A	Visitor	No Visitor
TRUE	60.0%	5.6%
FALSE	40.0%	94.4%

T2MF 2A	Visitor	No Visitor
TRUE	96.0%	57.0%
FALSE	4.0%	43.0%

T2MF 1A	Visitor	No Visitor
TRUE	57.7%	24.5%
FALSE	42.3%	75.5%

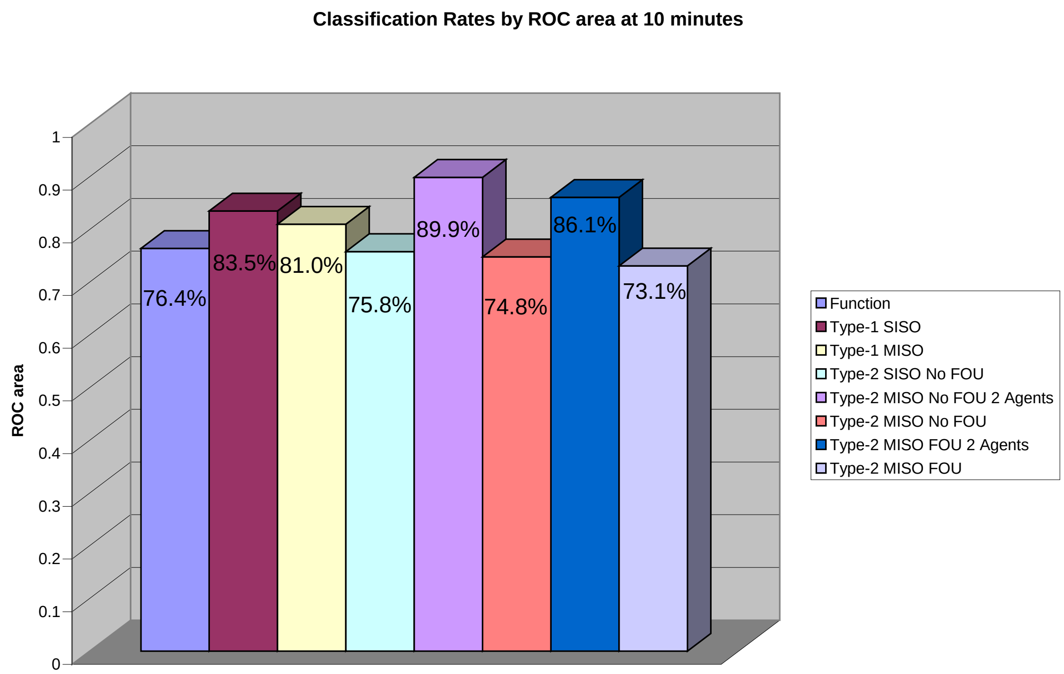


Fig. 6.6: ROC areas of the 8 classifiers for 10 minute minimum event length

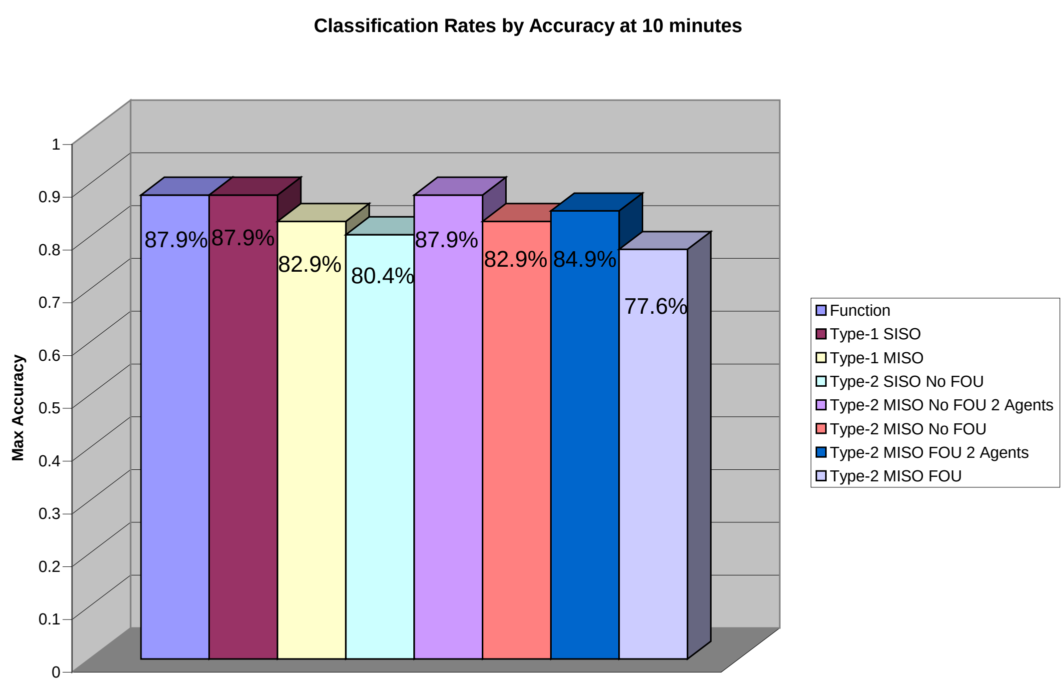


Fig. 6.7: Maximum accuracy of the 8 classifiers for 10 minute minimum event length

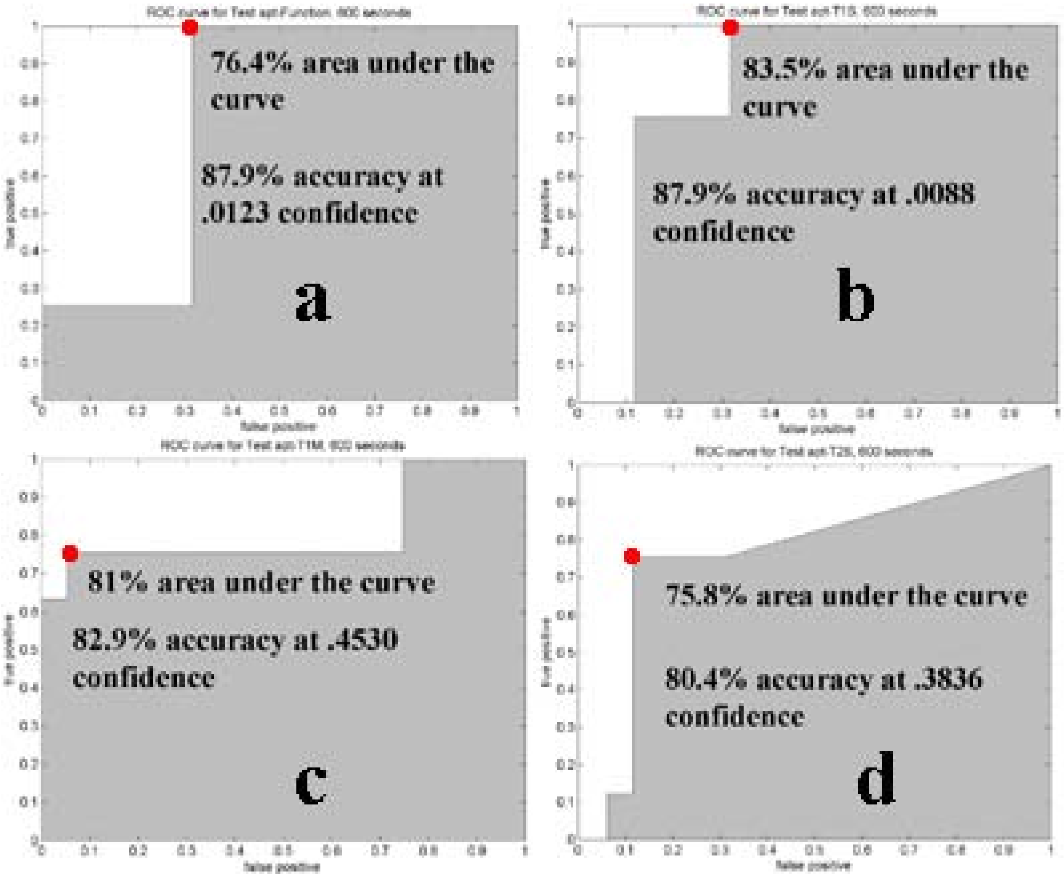


Fig. 6.8: ROC curves for the 10 minute minimum event length implementations.
The red dots mark the places where the confidence maximizes accuracy.
a) Function, b) Type-1 SISO, c) Type-1 MISO, d) Type-2 SISO No FOU,
e) Type-2 MISO No FOU 2 Agents, f) Type-2 MISO No FOU 1 Agent,
g) Type-2 MISO FOU 2 Agents, h) Type-2 MISO FOU 1 Agent

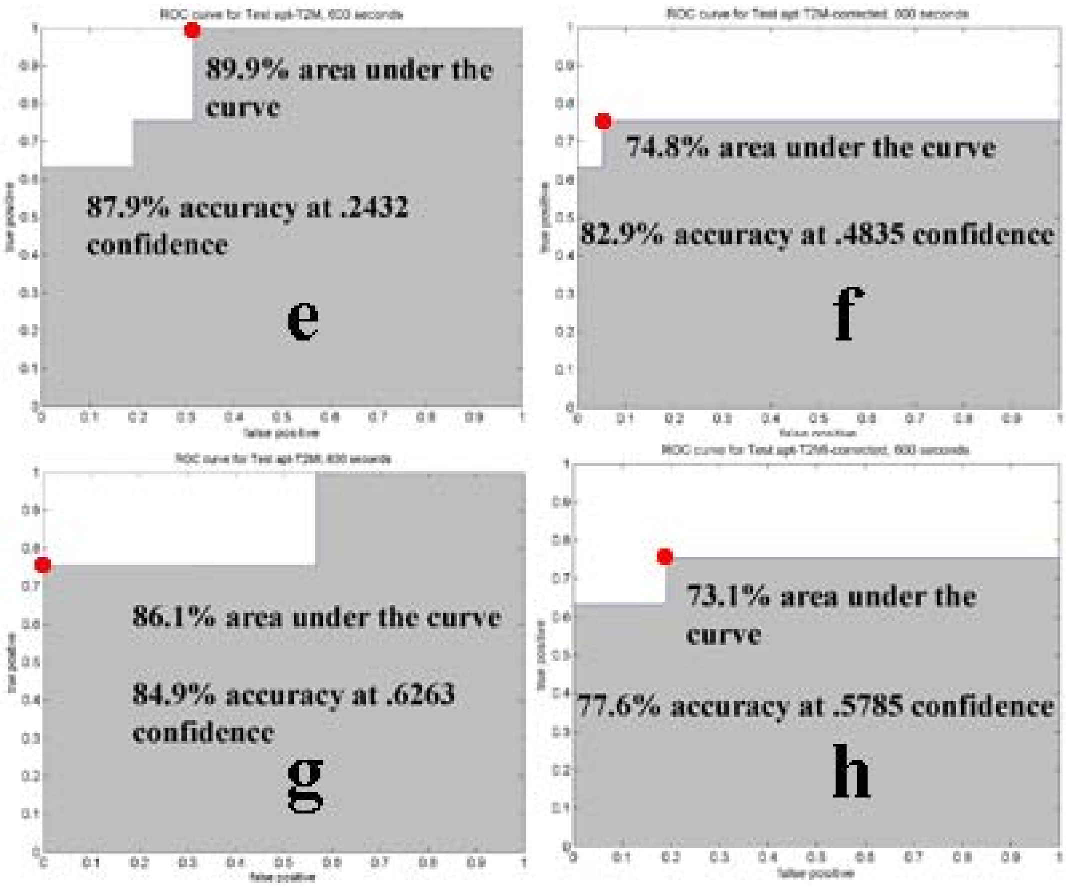


Fig. 6.8: ROC curves for the 10 minute minimum event length implementations.
The red dots mark the places where the confidence maximizes accuracy.
a) Function, b) Type-1 SISO, c) Type-1 MISO, d) Type-2 SISO No FOU,
e) Type-2 MISO No FOU 2 Agents, f) Type-2 MISO No FOU 1 Agent,
g) Type-2 MISO FOU 2 Agents, h) Type-2 MISO FOU 1 Agent

Table 6.2: Confusion matrices for the 10 minute minimum event length classifiers
T1S = Type-1 SISO, T1M = Type-1 MISO, T2S = Type-2 SISO No FOU,
T2M 2A = Type-2 MISO No FOU 2 Agent, T2M 1A = Type-2 MISO No FOU 1 Agent
T2MF 2A = Type-2 MISO FOU 2 Agent, T2MF 1A = Type-2 MISO FOU 1 Agent

Function	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T1S	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T1M	Visitor	No Visitor
TRUE	75.6%	5.4%
FALSE	24.4%	94.6%

T2S	Visitor	No Visitor
TRUE	75.6%	11.7%
FALSE	24.4%	88.3%

T2M 2A	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T2M 1A	Visitor	No Visitor
TRUE	75.6%	5.4%
FALSE	24.4%	94.6%

T2MF 2A	Visitor	No Visitor
TRUE	75.5%	0.0%
FALSE	24.5%	100.0%

T2MF 1A	Visitor	No Visitor
TRUE	75.6%	19.1%
FALSE	24.4%	80.9%

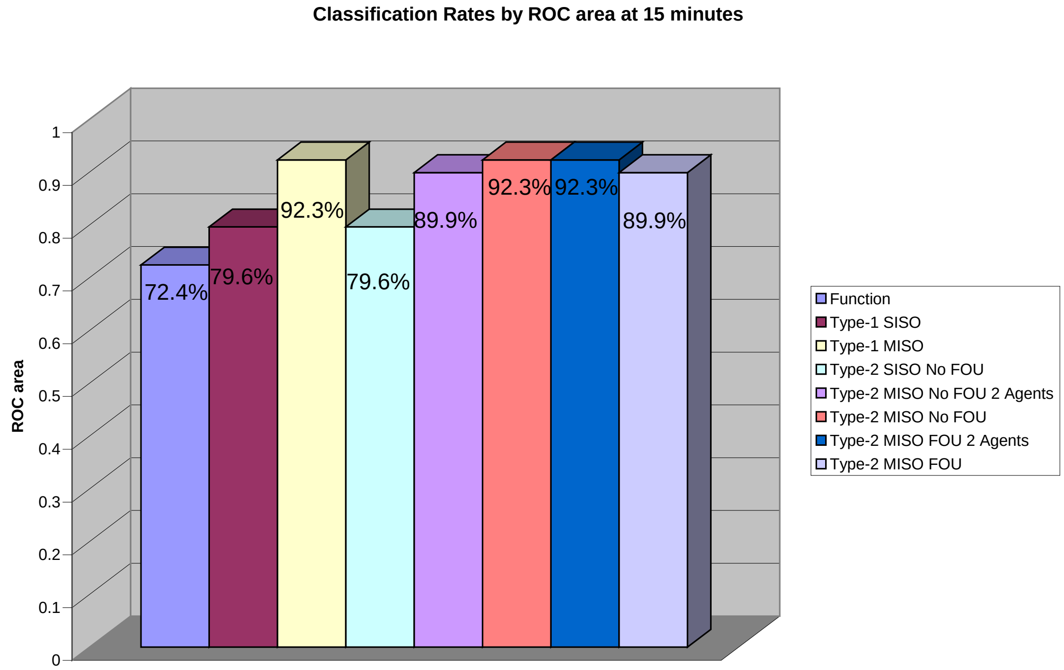


Fig. 6.9: ROC areas of the 8 classifiers for 15 minute minimum length

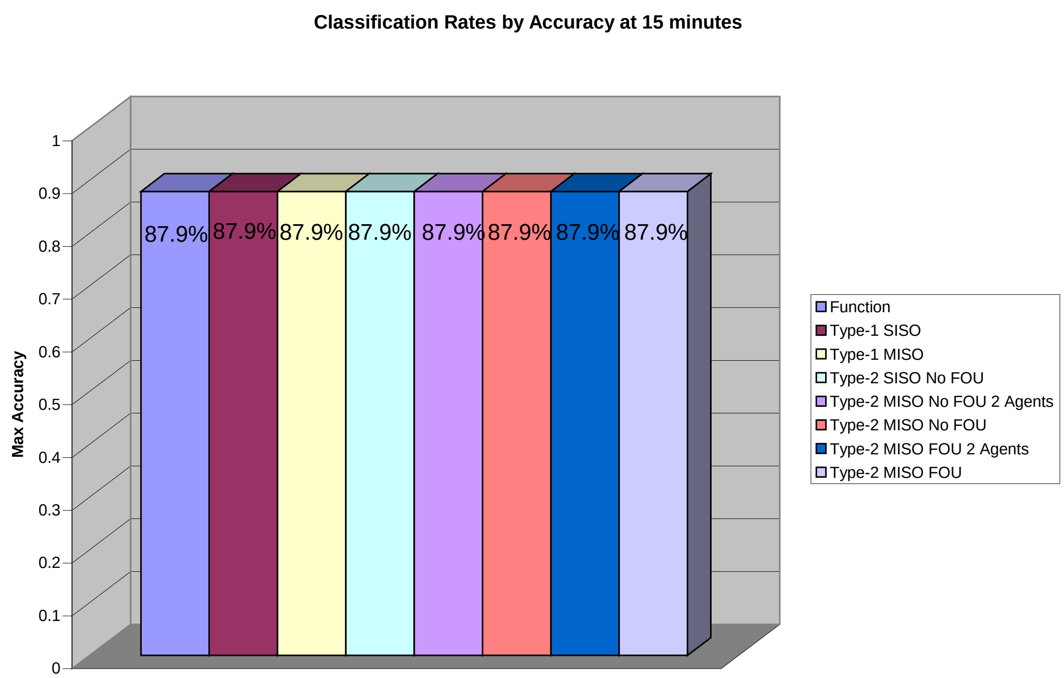


Fig. 6.10: Maximum accuracy of the 8 classifiers for 15 minute minimum length

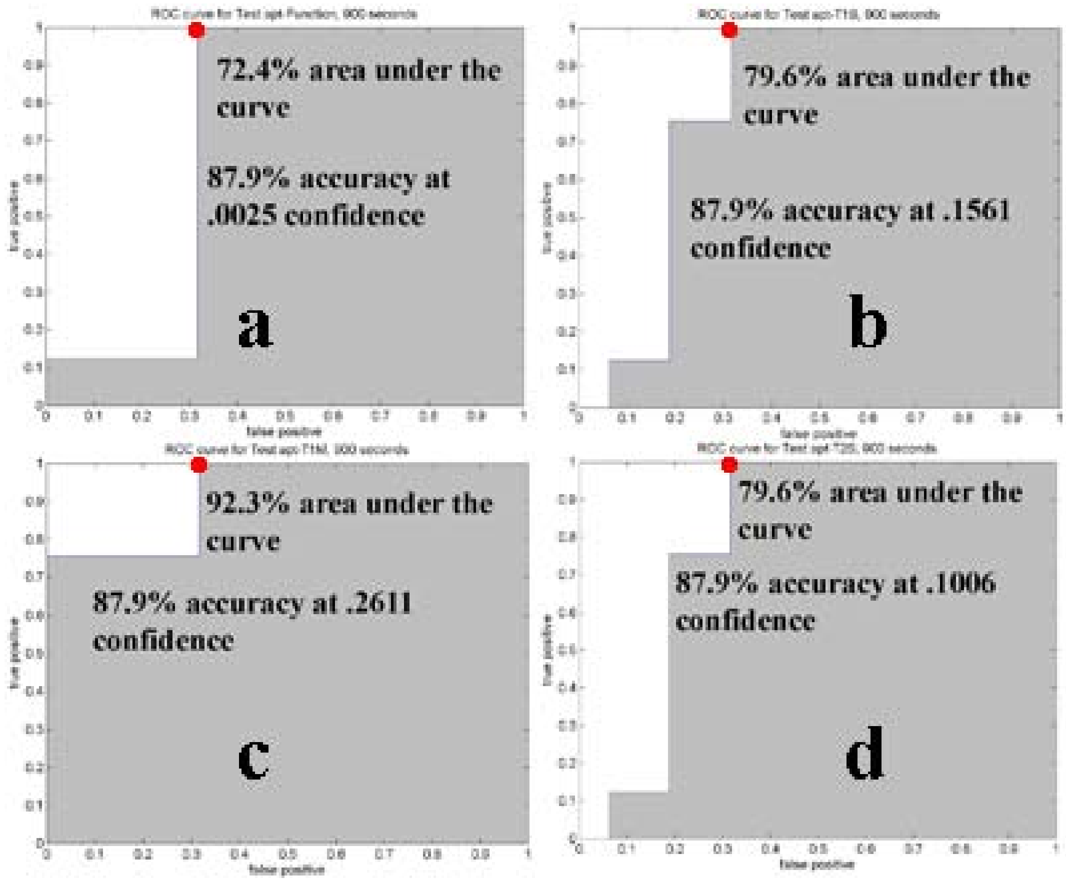


Fig. 6.11: ROC curves for the 15 minute minimum event length implementations.
The red dots mark the places where the confidence maximizes accuracy.
a) Function, b) Type-1 SISO, c) Type-1 MISO, d) Type-2 SISO No FOU,
e) Type-2 MISO No FOU 2 Agents, f) Type-2 MISO No FOU 1 Agent,
g) Type-2 MISO FOU 2 Agents, h) Type-2 MISO FOU 1 Agent

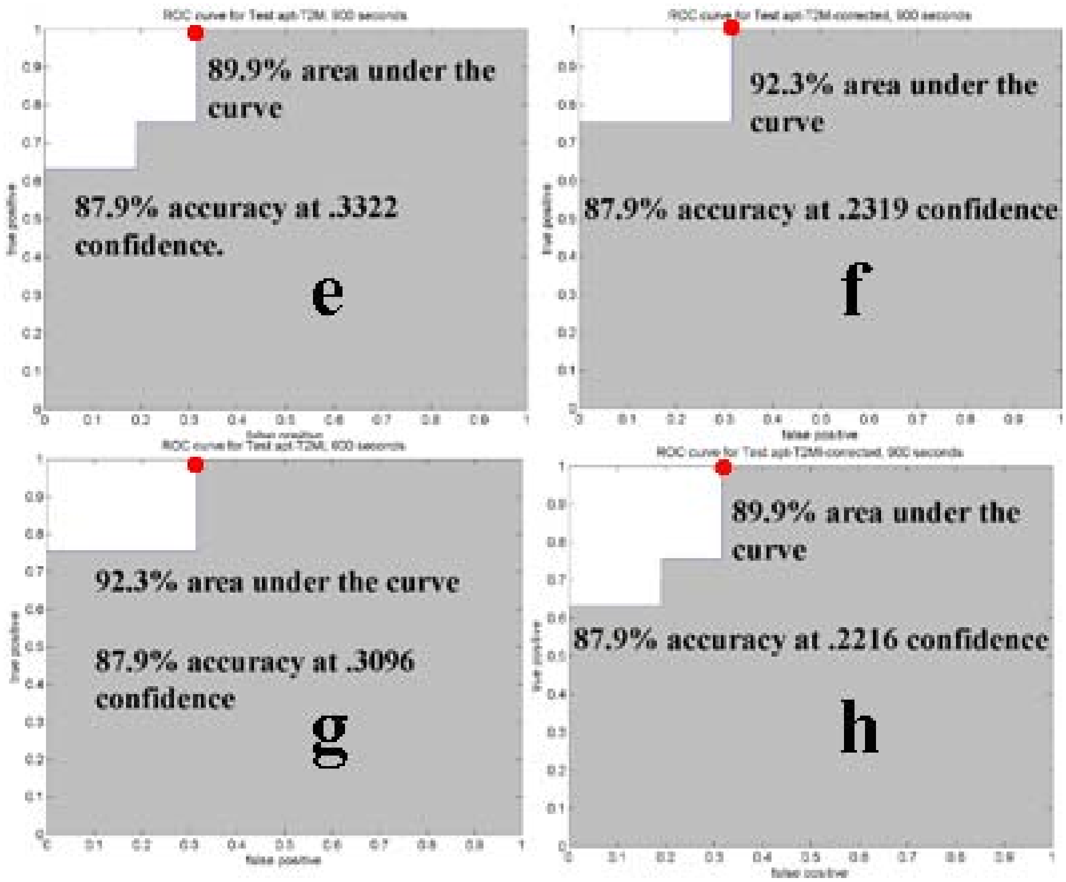


Fig. 6.11: ROC curves for the 15 minute minimum event length implementations.
The red dots mark the places where the confidence maximizes accuracy.
a) Function, b) Type-1 SISO, c) Type-1 MISO, d) Type-2 SISO No FOU,
e) Type-2 MISO No FOU 2 Agents, f) Type-2 MISO No FOU 1 Agent,
g) Type-2 MISO FOU 2 Agents, h) Type-2 MISO FOU 1 Agent

Table 6.3: Confusion matrices for the 15 minute minimum event length classifiers
T1S = Type-1 SISO, T1M = Type-1 MISO, T2S = Type-2 SISO No FOU,
T2M 2A = Type-2 MISO No FOU 2 Agent, T2M 1A = Type-2 MISO No FOU 1 Agent
T2MF 2A = Type-2 MISO FOU 2 Agent, T2MF 1A = Type-2 MISO FOU 1 Agent

Function	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T1S	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T1M	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T2S	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T2M 2A	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T2M 1A	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T2MF 2A	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

T2MF 1A	Visitor	No Visitor
TRUE	100.0%	31.7%
FALSE	0.0%	68.3%

The following figures are used for visual inspection of the algorithm results (figures 6.12 to 6.27). They are based on the 10 minute minimum event length. Ground truth data is bounded in red and blue boxes for easier viewing. The red boxes surround test times that contain visitor ground truth. Blue boxes surround times that contain no visitor ground truth, meaning times when the algorithm should return a low confidence in visitors. I use no(n) visitor ground truth in the figures to clarify the intended purpose.

Times that are not bounded by red or blue boxes are not from this series of tests. Other students were using the apartment during those times, and they should be disregarded for the purposes of verification.

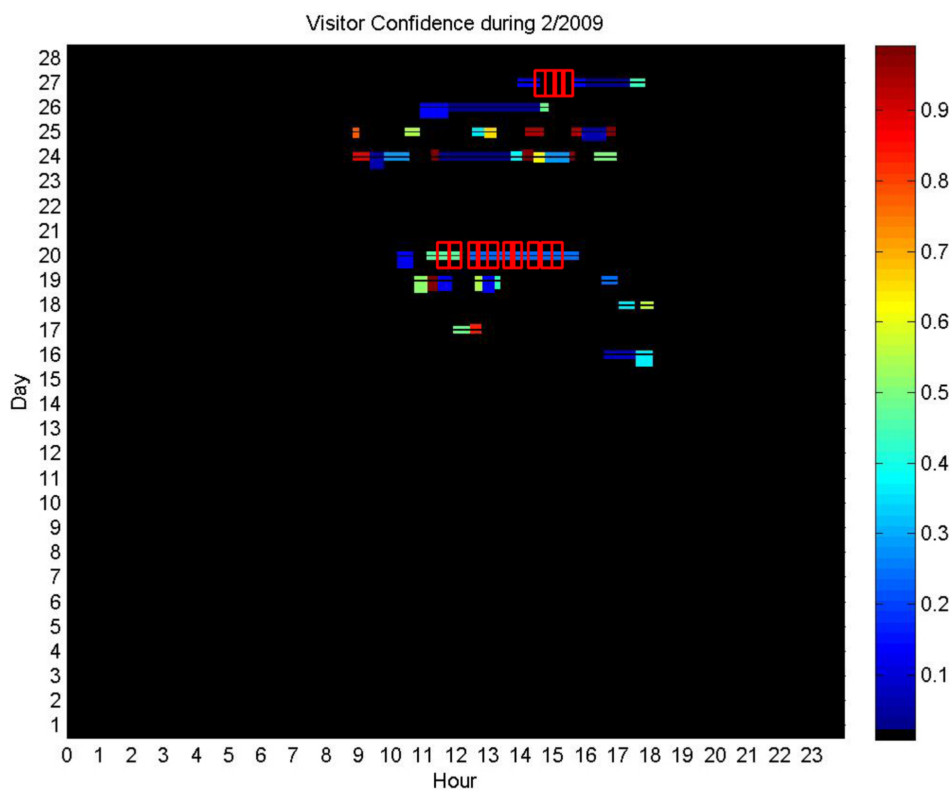


Fig. 6.12: Display from function implementation on the test apartment data. The red bounds indicate visitor ground truth.

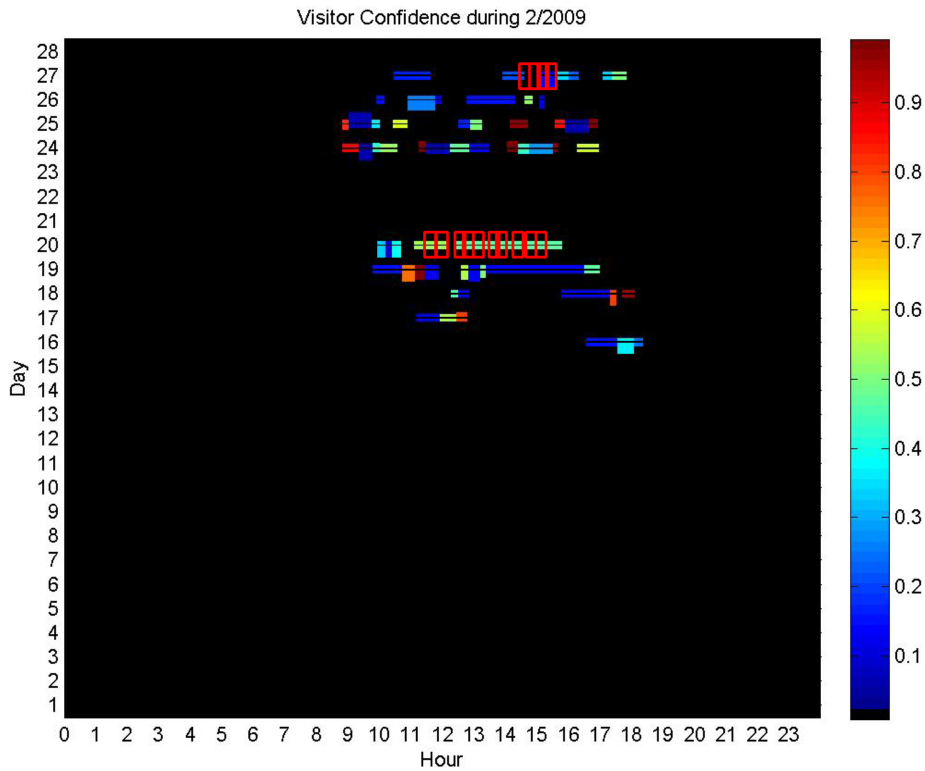


Fig. 6.13: Display from Type-1 SISO implementation on the test apartment data. The red bounds indicate visitor ground truth.

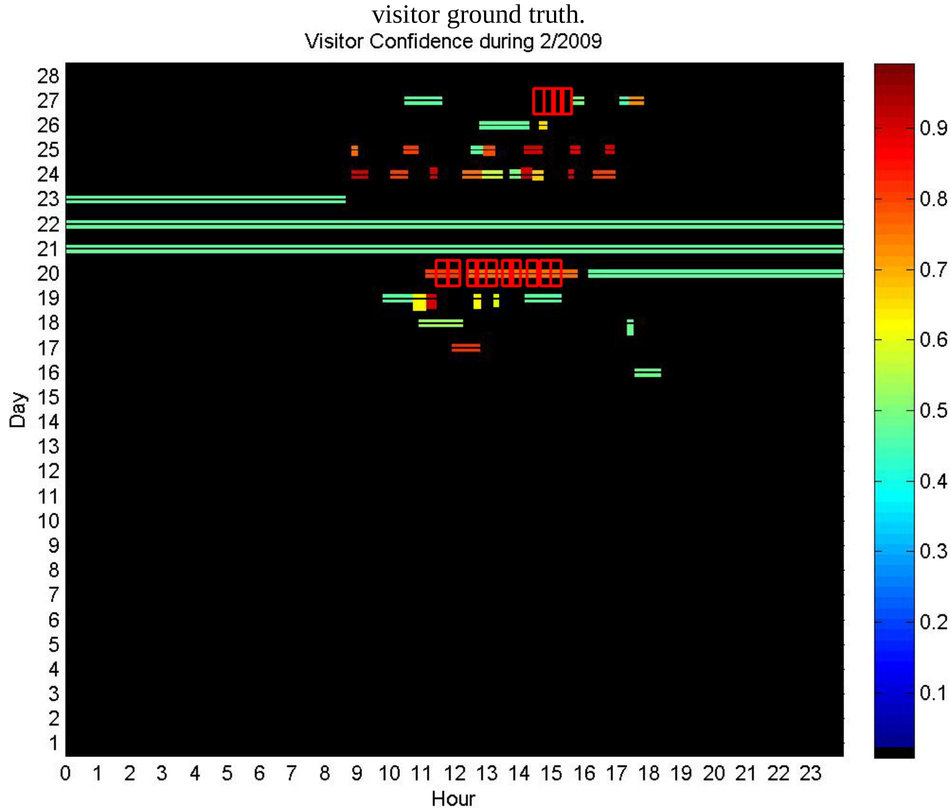


Fig. 6.14: Display from Type-1 MISO implementation on the test apartment data. The red bounds indicate visitor ground truth.

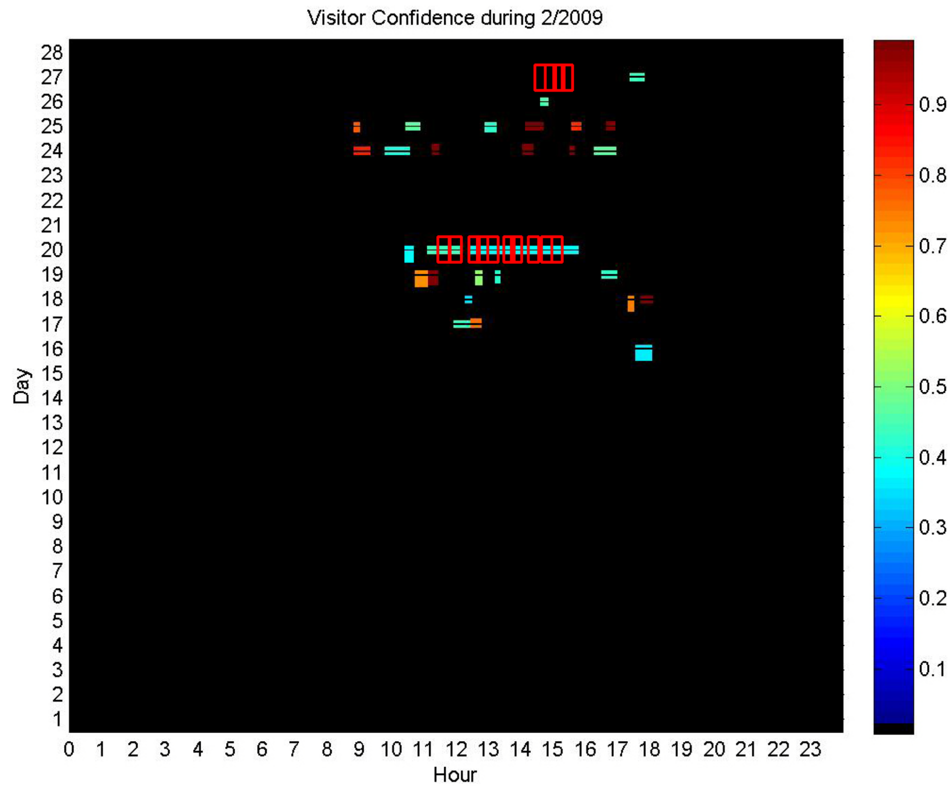


Fig. 6.15: Display from Type-2 SISO No FOU implementation on the test apartment data. Red bounds indicate visitor ground truth.

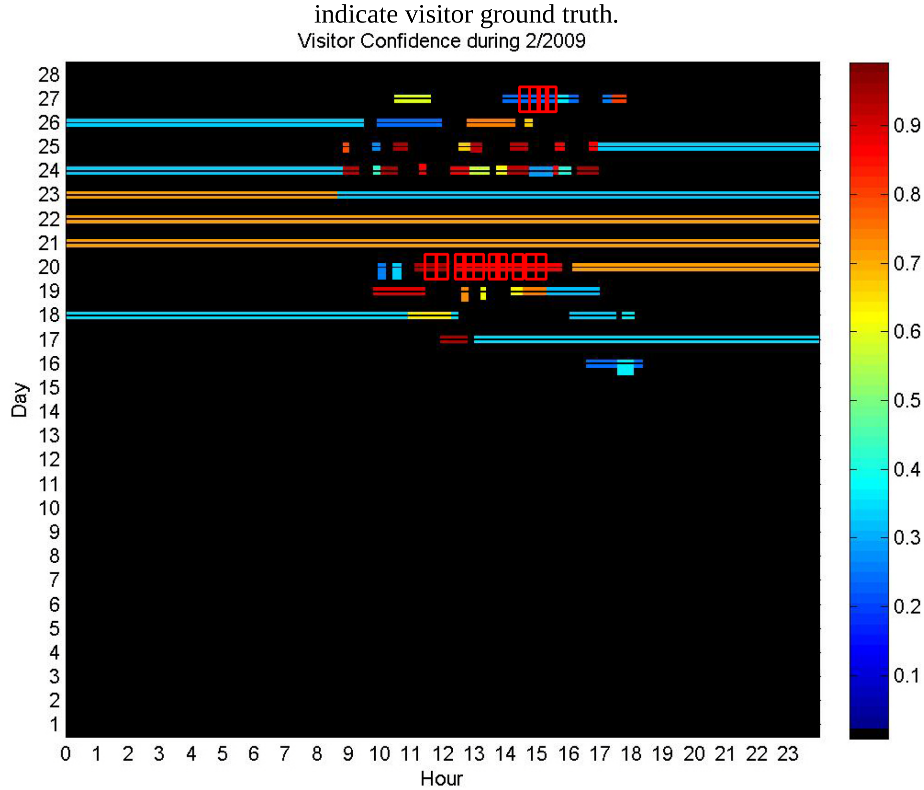


Fig. 6.16: Display from Type-2 MISO No FOU 2 Agents implementation on the test apartment data. Red bounds indicate visitor ground truth.

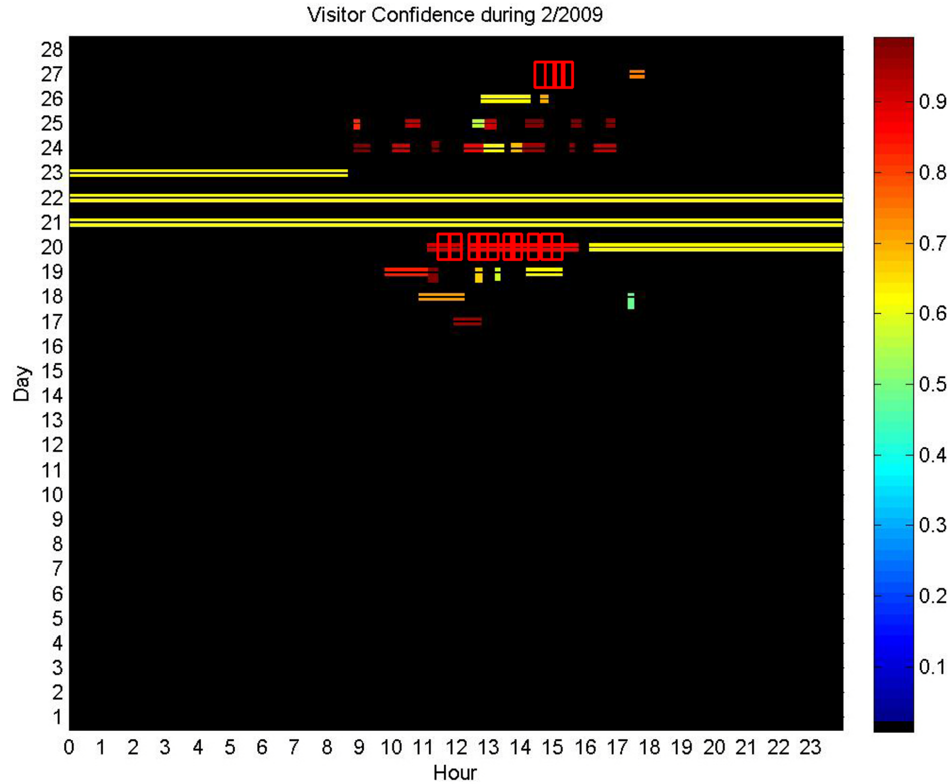


Fig. 6.17: Display from Type-2 MISO No FOU 1 Agent implementation on the test apartment data. The red bounds indicate visitor ground truth.

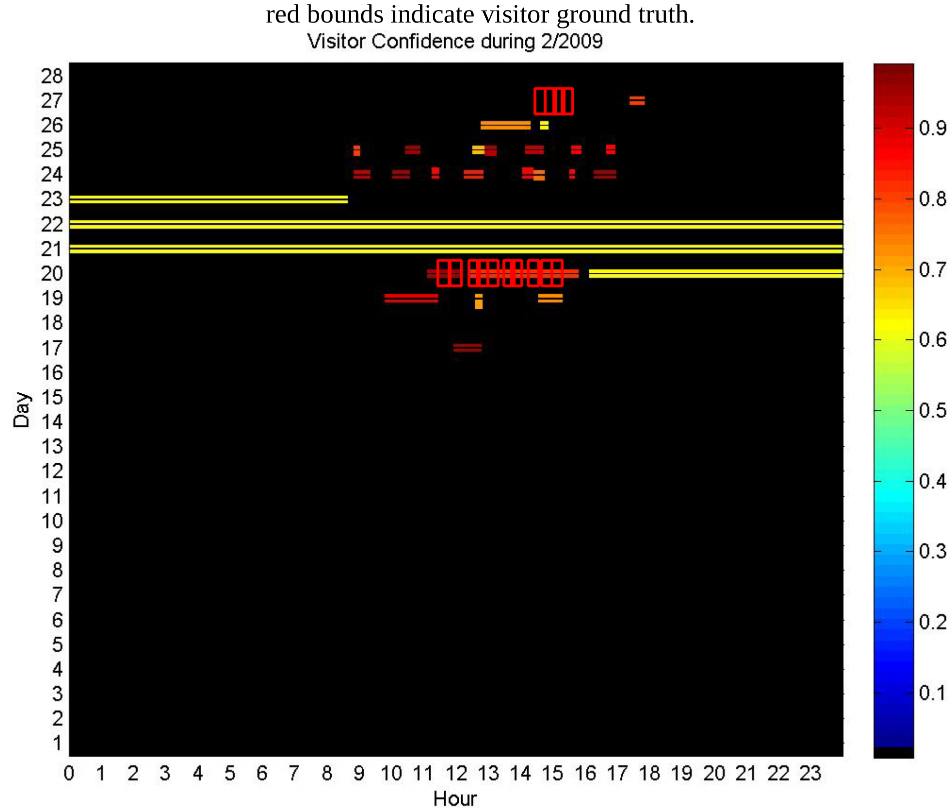


Fig. 6.18: Display from Type-2 MISO FOU 2 Agents implementation on the test apartment data. The red bounds indicate visitor ground truth.

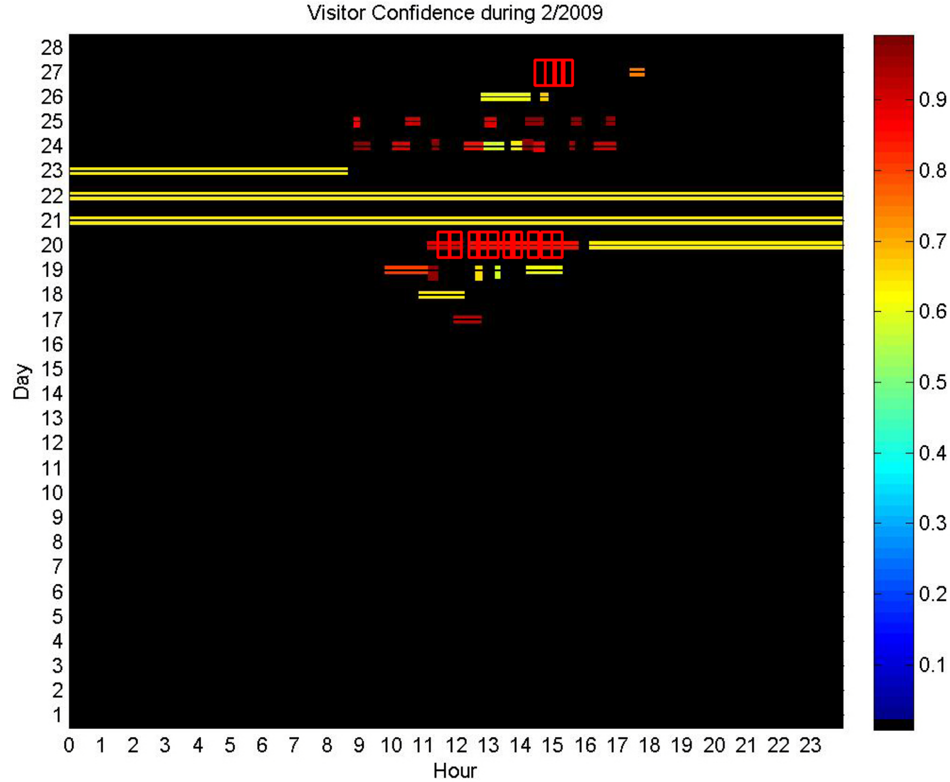


Fig. 6.19: Display from Type-2 MISO FOU 1 Agent implementation on the test apartment data. The red bounds indicate visitor ground truth.

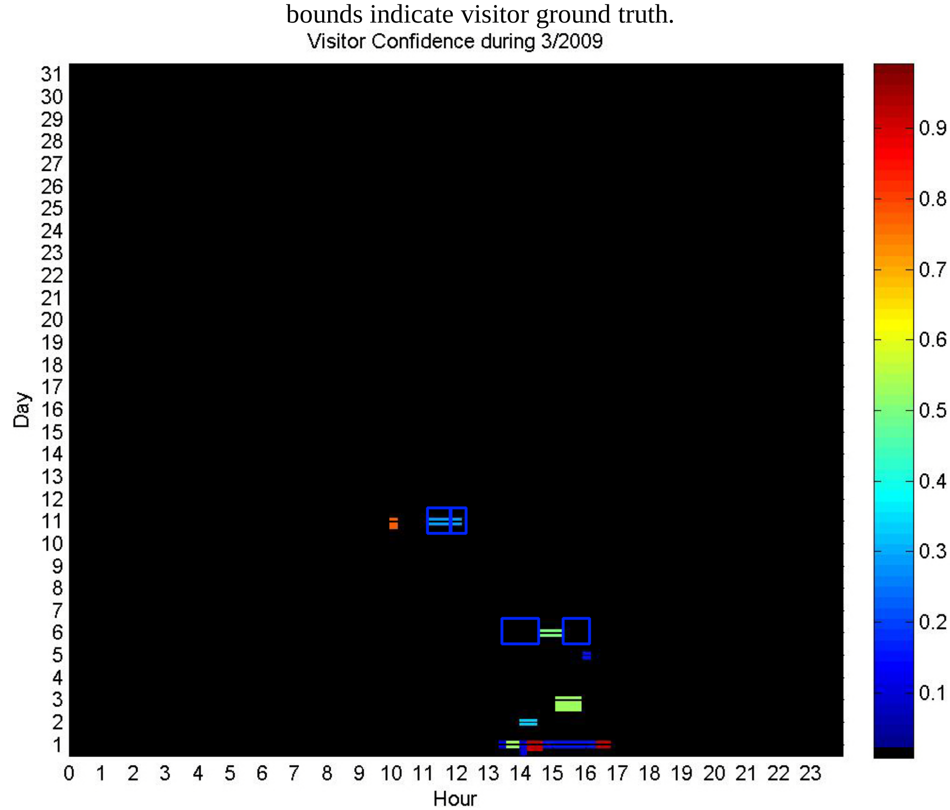


Fig. 6.20: Display from function implementation on the test apartment data. The blue bounds indicate no(n) visitor ground truth.

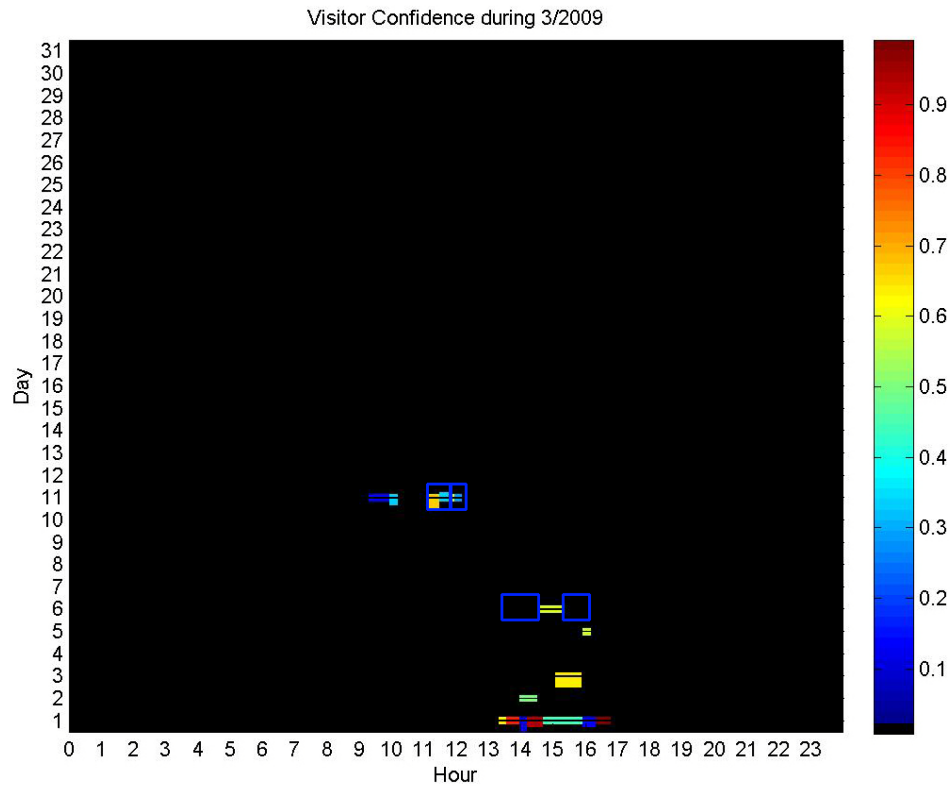


Fig. 6.21: Display from Type-1 SISO implementation on the test apartment data. The blue bounds indicate no(n) visitor ground truth.

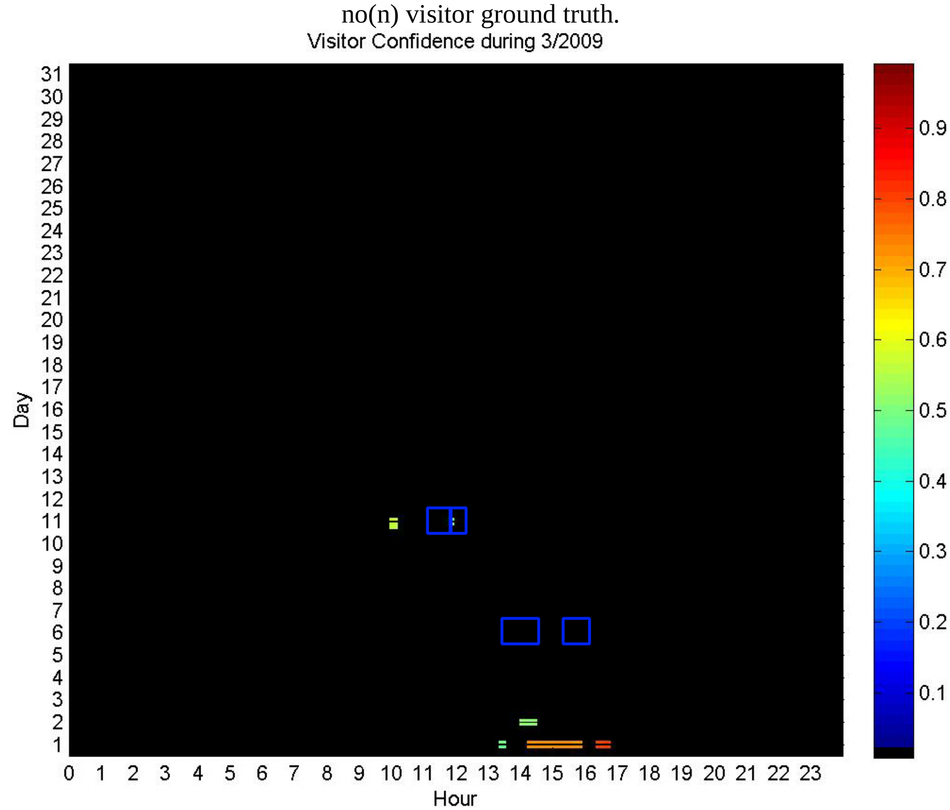


Fig. 6.22: Display from Type-1 MISO implementation on the test apartment data. The blue bounds indicate no(n) visitor ground truth.

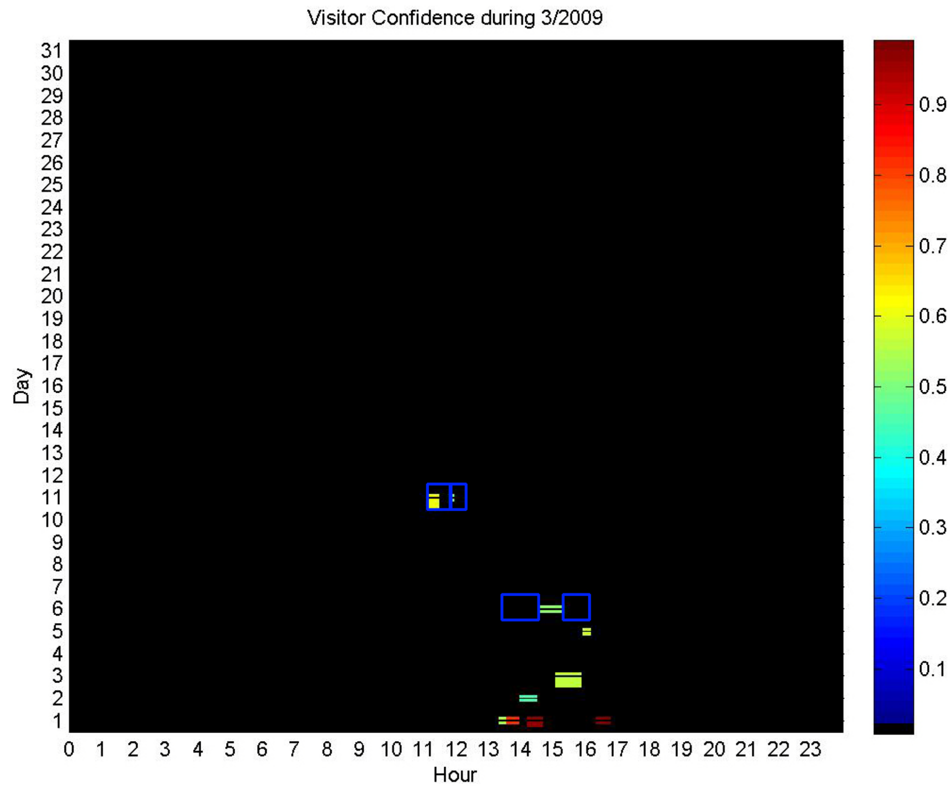


Fig. 6.23: Display from Type-2 SISO implementation on the test apartment data. The blue bounds indicate no(n) visitor ground truth.

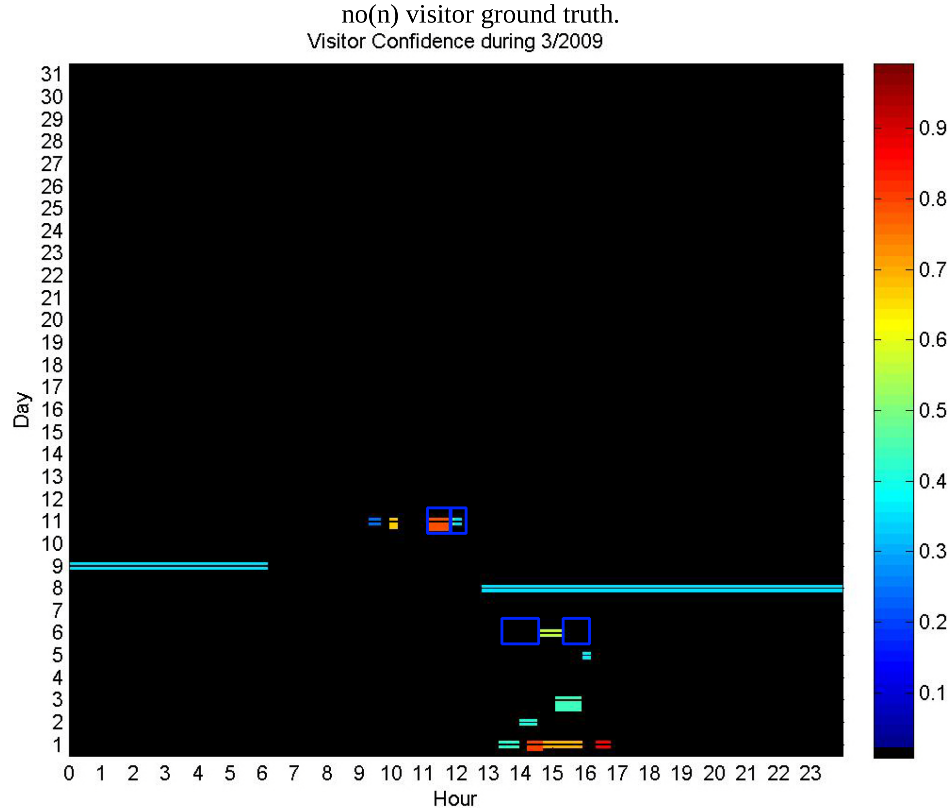


Fig. 6.24: Display from Type-2 MISO No FOU 2 Agents implementation on the test apartment data. The blue bounds indicate no(n) visitor ground truth.

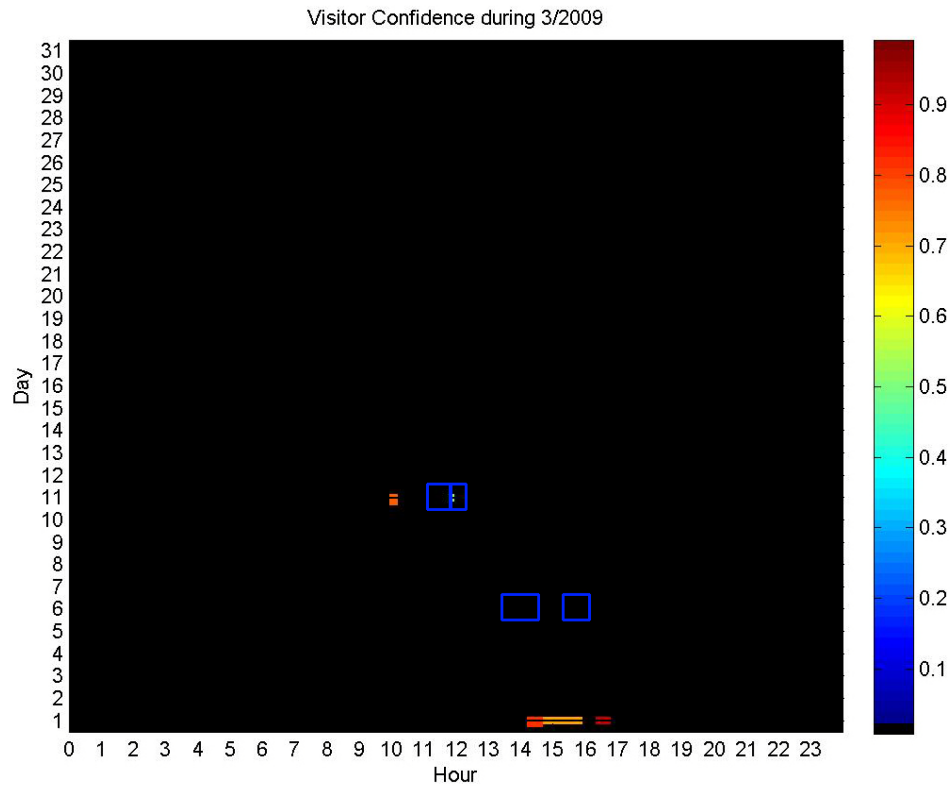


Fig. 6.25: Display from Type-2 MISO No FOU 1 Agent implementation on the test apartment data. The blue bounds indicate no(n) visitor ground truth.

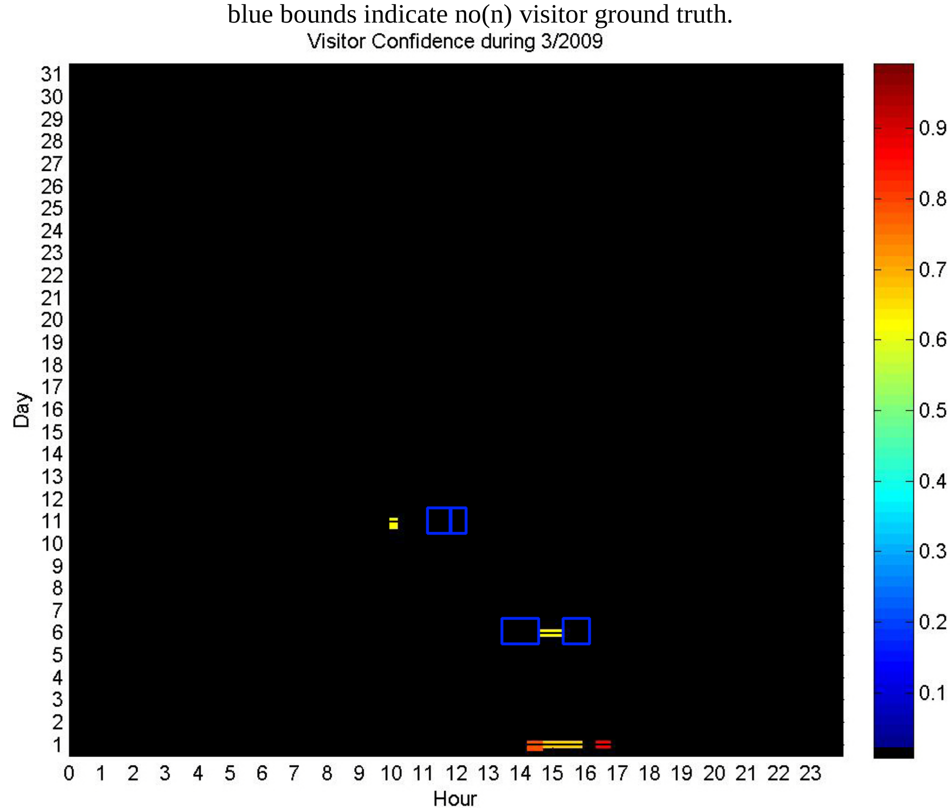


Fig. 6.26: Display from Type-2 MISO FOU 2 Agents implementation on the test apartment data. The blue bounds indicate no(n) visitor ground truth.

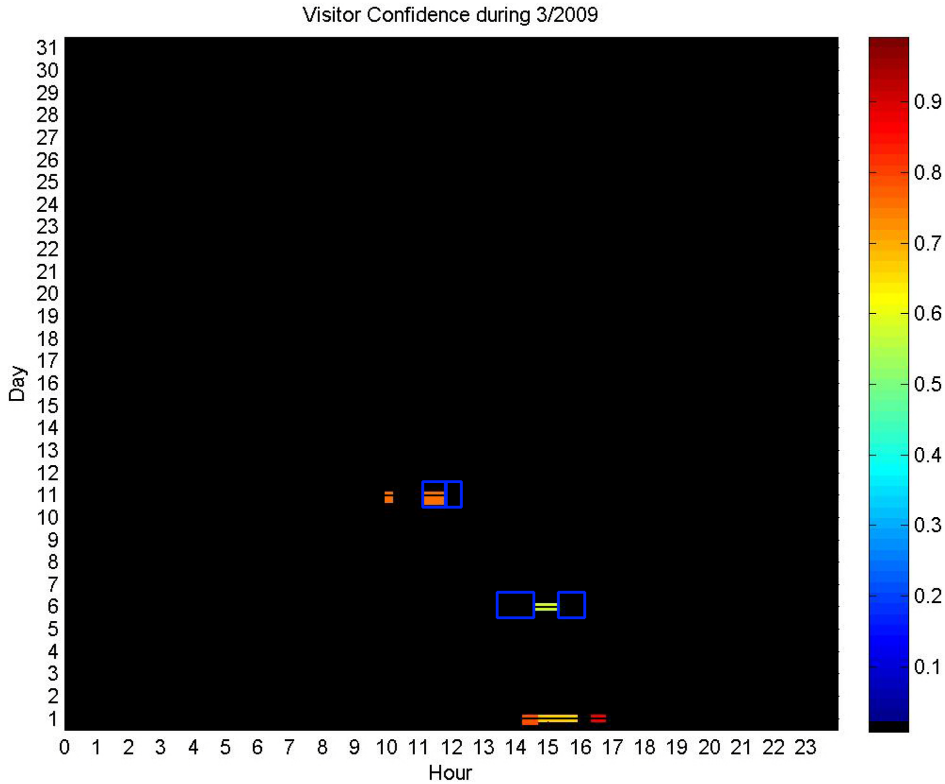


Fig. 6.27: Display from Type-2 MISO FOU 1 Agent implementation on the test apartment data. The blue bounds indicate no(n) visitor ground truth.

6.2 Tigerplace User Results and Analysis

While the apartment tests returned good results, the confidence levels used are concerning (.0088 confidence level is good?). This carries over to the resident data, which will naturally require a different set of confidences if the same membership functions are used. For the resident tests, I used the same set of confidences for all three tests (Users 3004, 3007, and 3010). However, each implementation of the classifier uses a different confidence threshold (Table 6.4).

Table 6.4: Confidence thresholds used when graphing the resident apartment displays

Classifier	Confidence Threshold
Function	0.40
Type-1 SISO (T1S)	0.50
Type-1 MISO (T1M)	0.75
Type-2 SISO No FOU (T2S)	0.40
Type-2 MISO No FOU 2 Agents (T2M 2A)	0.40
Type-2 MISO No FOU 1 Agent (T2M 1A)	0.75
Type-2 MISO FOU 2 Agents (T2MF 2A)	0.75
Type-2 MISO FOU 1 Agent (T2MF 1A)	0.75

Since there is no real ground truth data for the resident apartments, there are no ROC curves or confusion matrices. However, the cleaning schedule times are bounded by red boxes so that a sense of performance can be ascertained. The motion density map is included first (figures 6.28, 6.37, and 6.46) as a comparison. Visitor confidence graphs follow their respective motion density maps. User 3004 is shown in figures 6.29 thru 6.36. User 3007 is shown in figures 6.38 thru 6.45. User 3010 is shown in figures 6.47 thru 6.54.

6.2.1 User 3004

User 3004 led a highly regimented life. Based on the motion density map, there exist clear times when this resident would wake, eat meals, and go to bed. This resident gets up every morning at 7:00 AM, and leaves for breakfast at 8:00 AM. Dinner is at 6:00 PM, and bedtime is between 10:00 and 11:00 PM. The activity levels are high in the mornings, and otherwise are pretty moderate.

The function classifier found several short intervals that it classified as visitor times. From event inspection, these are mostly from times where the resident is in the kitchen. The door sensor area overlaps some of the kitchen, and the motion density becomes inflated. The Type-1 SISO system found several times when visitors are likely. The MISO system was able to greatly reduce the noise from the motion densities, but may have not included all of the visitor times. The Type-2 2 Agent systems behaved almost identically to each other (and the Type-2 SISO), where the FOU system granted an increase in the confidence of the visitor times detected. The single agent cases made the visitation stays appear much longer. The Type-2 MISO No FOU 2 Agent system, while returning lower confidences, returns values that best match the confidences I would have assigned the times in question. However, without additional ground truth data the Type-2 MISO FOU 2 Agent system may have performed better (it definitely was more sure of the solution).

Figures 6.29 thru 6.36 display the visitor confidences for the month of January 2006. The housekeeping times are bounded by a red rectangle. The displays are based on the 10 minute minimum event length, and the confidences from Table 6.4. For comparison, I also included the motion density plot in figure 6.28.

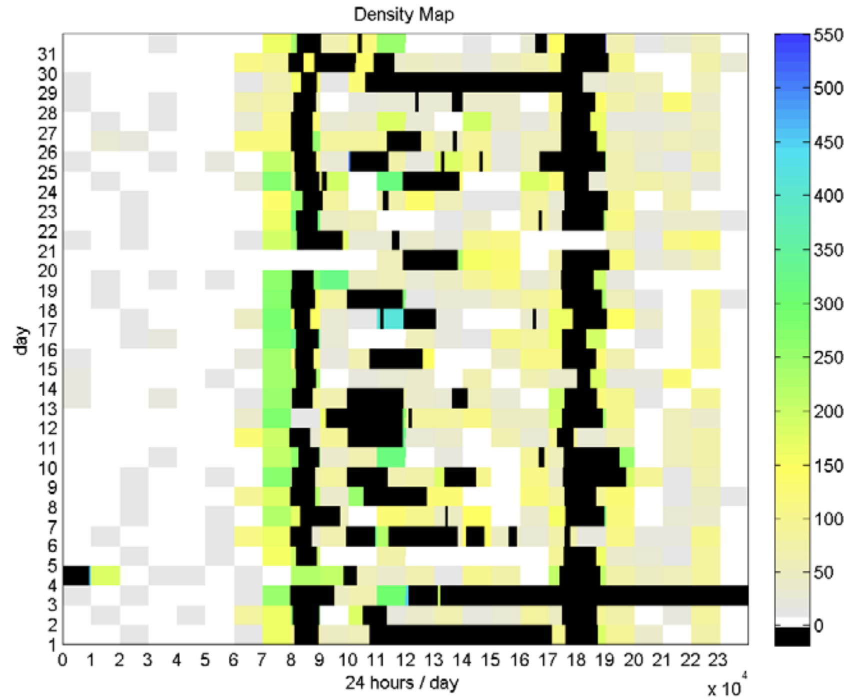


Fig. 6.28: Motion density plot for User 3004

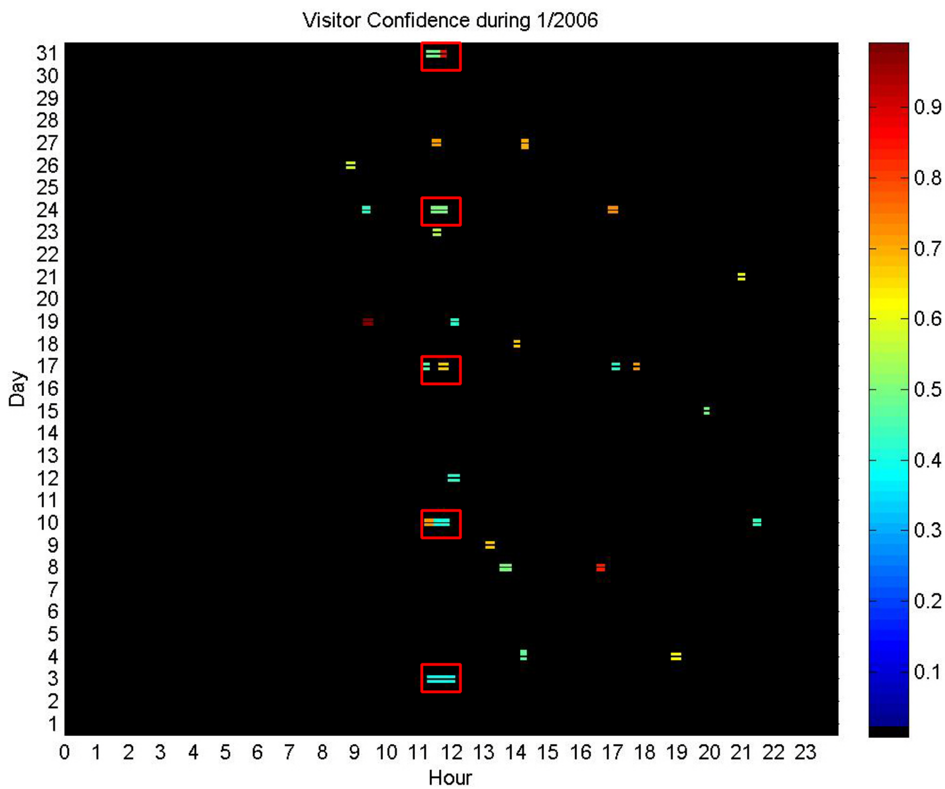


Fig. 6.29: User 3004 function implementation display

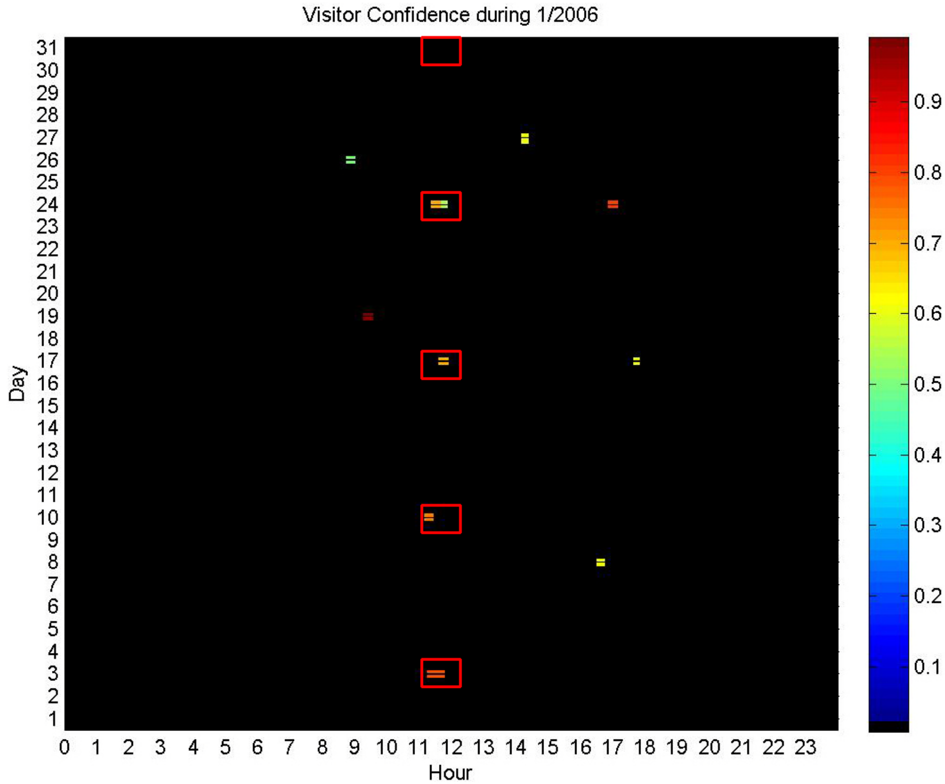


Fig. 6.30: User 3004 Type-1 SISO implementation display

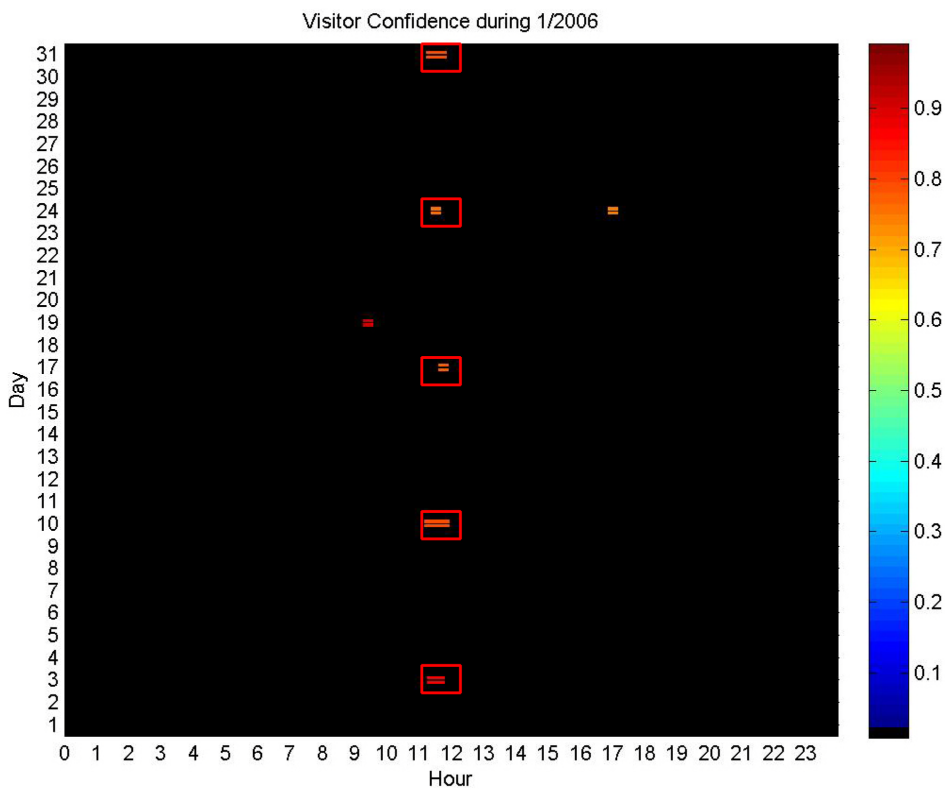


Fig. 6.31: User 3004 Type-1 MISO implementation display

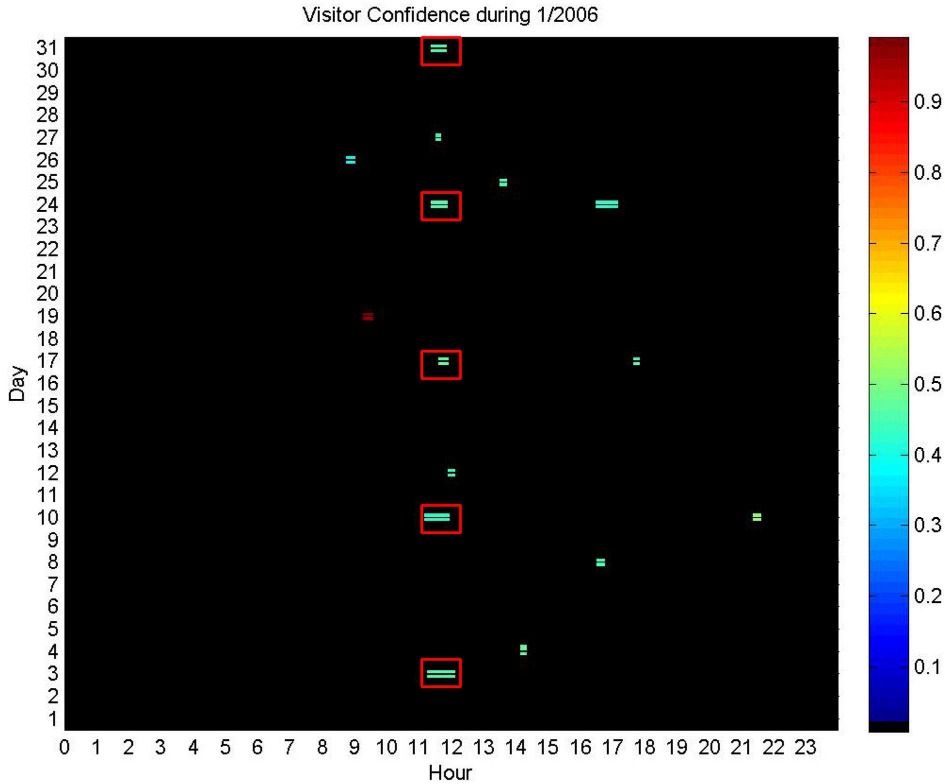


Fig. 6.32: User 3004 Type-2 SISO No FOU implementation display

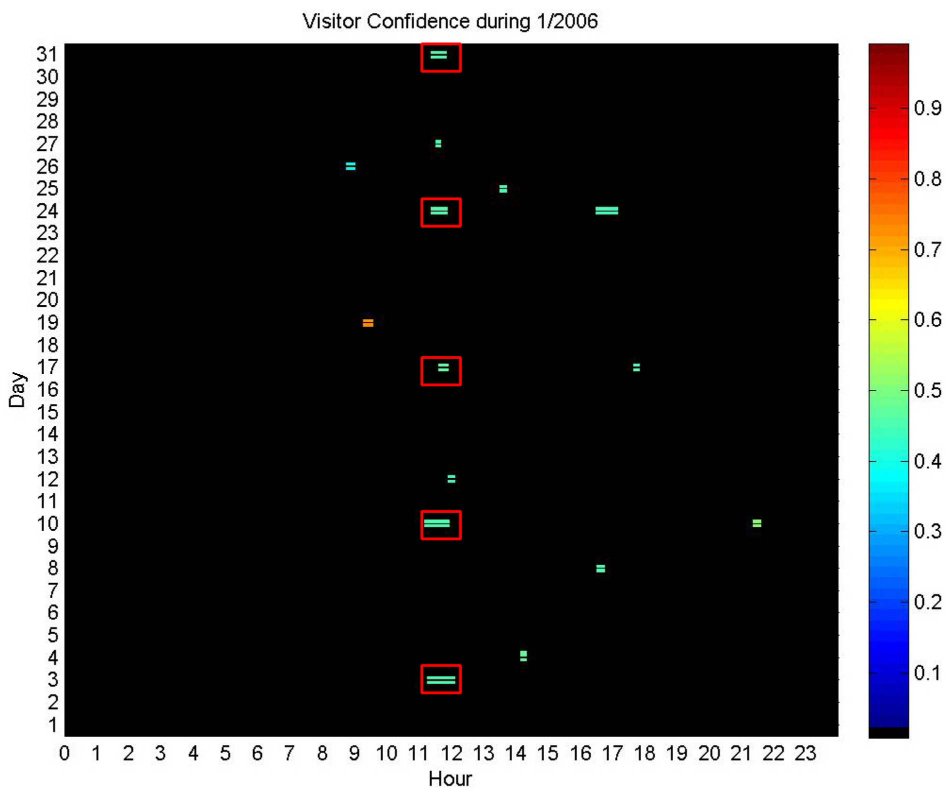


Fig. 6.33: User 3004 Type-2 MISO No FOU 2 Agents implementation display

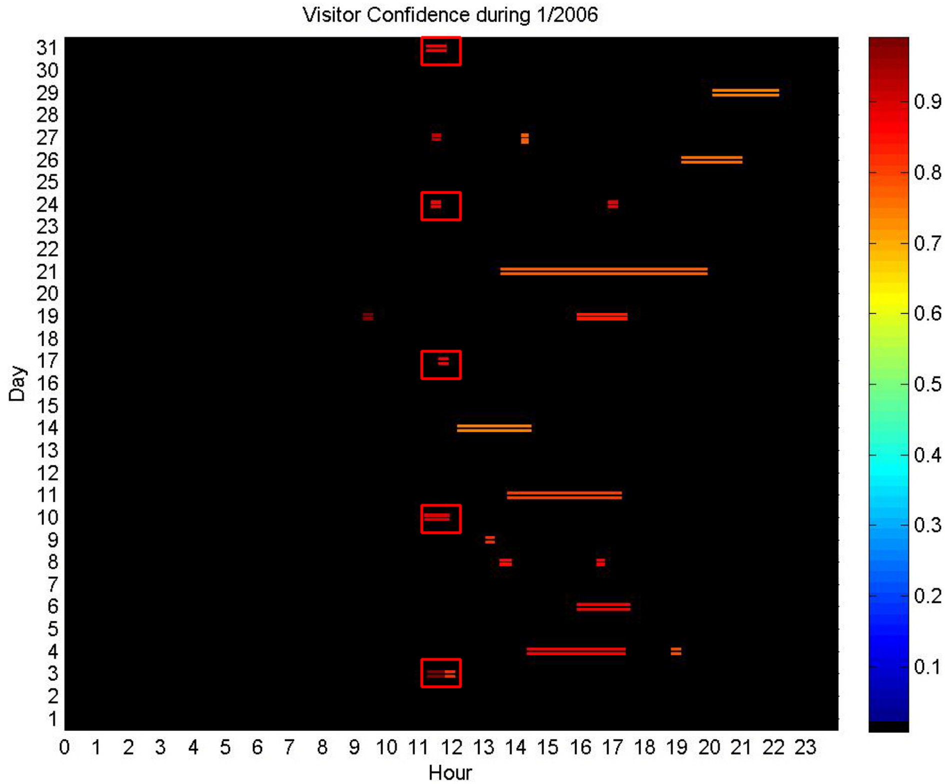


Fig. 6.34: User 3004 Type-2 MISO No FOU 1 Agent implementation display

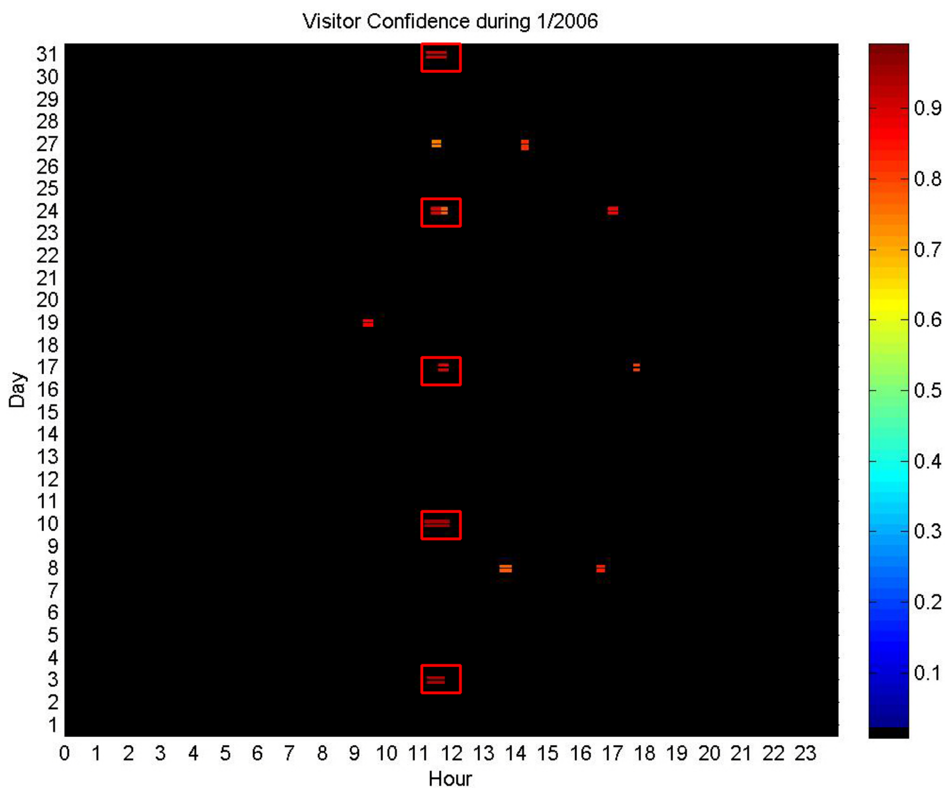


Fig. 6.35: User 3004 Type-2 MISO FOU 2 Agents implementation display

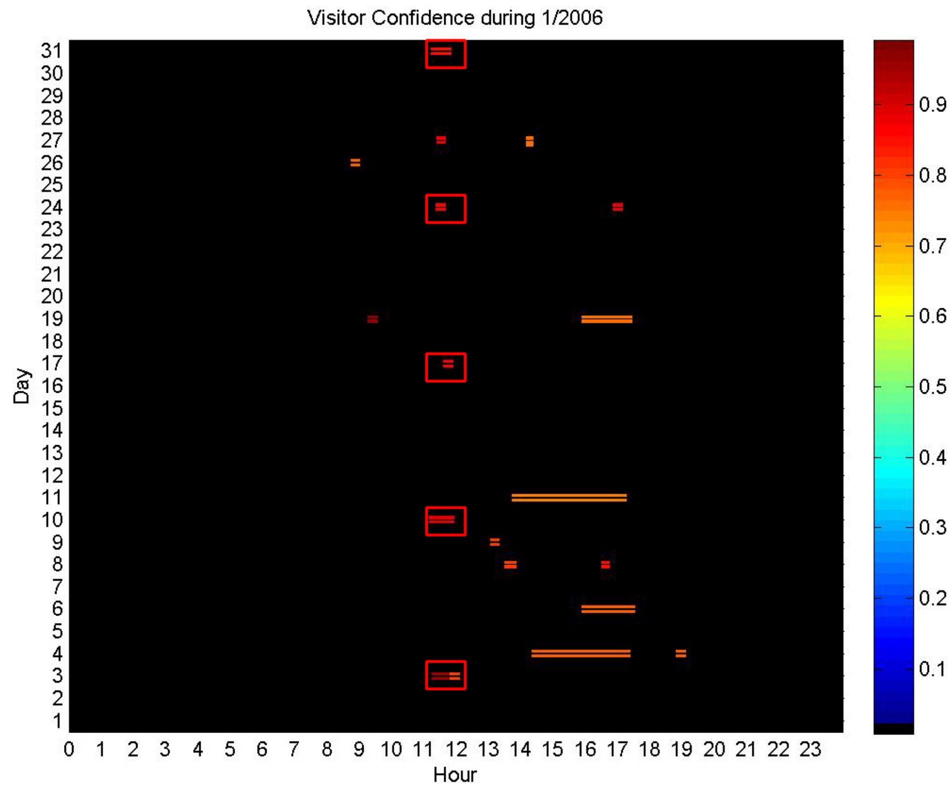


Fig. 6.36: User 3004 Type-2 MISO FOU 1 Agent implementation display

6.2.2 User 3007

User 3007 only had one regular period of the day, which was leaving the apartment for dinner. The motion density map displays a very restless life, and mostly sedentary activities. It is difficult to tell if there were many visitors from the map, and the lack of time out of the apartment leads me to believe this resident was socially isolated.

The unpredictable, sedentary activity levels make this a tough visitor detection case. During this display month, September 2007, it appears as though the resident had an aid to help get ready in the mornings (confirmed). The function classifier collected broken visitor information during the morning visitor detections, but all four housekeeping visits were correctly classified. The Type-1 SISO system caught all four of the housekeeping visit, but like the function classifier broke the morning into separate visitor cases. The Type-1 MISO system again seems to have thrown out possible visitor times, including two housekeeping visits. The Type-1 MISO system was, however, able to combine the detected information into logical blocks. The Type-2 SISO No FOU system performed similarly to the Type-1 MISO and function classifiers. The Type-2 MISO No FOU 2 Agent system did not display great confidence in any of the anticipated times. It also dropped a housekeeping visit, and likely several other visitors (based on the other algorithm results). The Type-2 MISO FOU 2 Agent system either cleaned up a little bit of extra noise, or dropped some valid detections compared to the other systems (it is unclear). The Type-2 MISO FOU 2 Agent system also dropped the last housekeeping visit, but, had a high confidence in the detections it made. The single agent Type-2 systems again had a high confidence in the times it detected visitors, but classified elongated periods of time as high visitor confidence.

Figures 6.38 thru 6.45 display the visitor confidences for the month of September 2007. The housekeeping times are bounded by a red rectangle. The displays are based on the 10 minute minimum event length, and the confidences from Table 6.4. For comparison, I also included the motion density plot in figure 6.37.

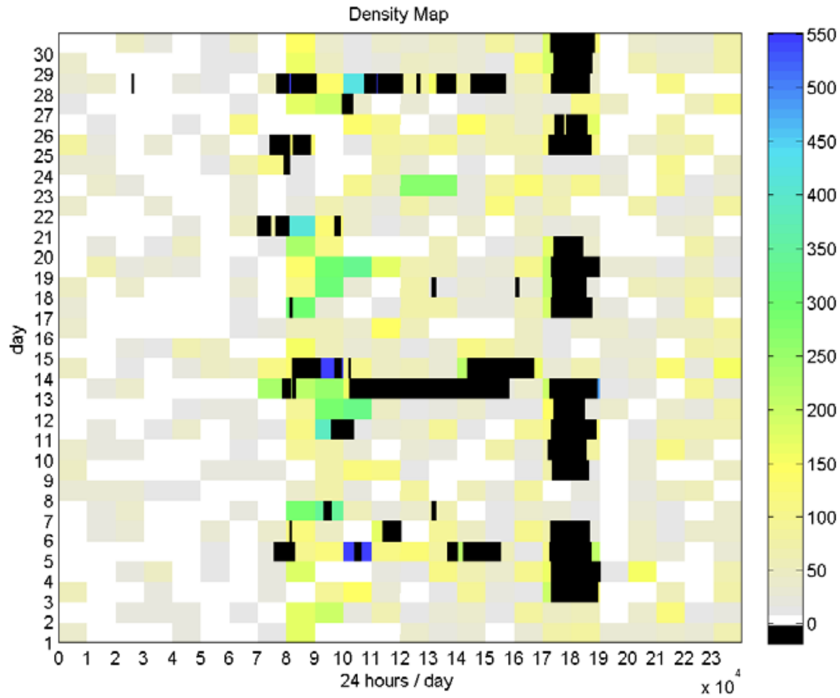


Fig. 6.37: Motion density plot for User 3007

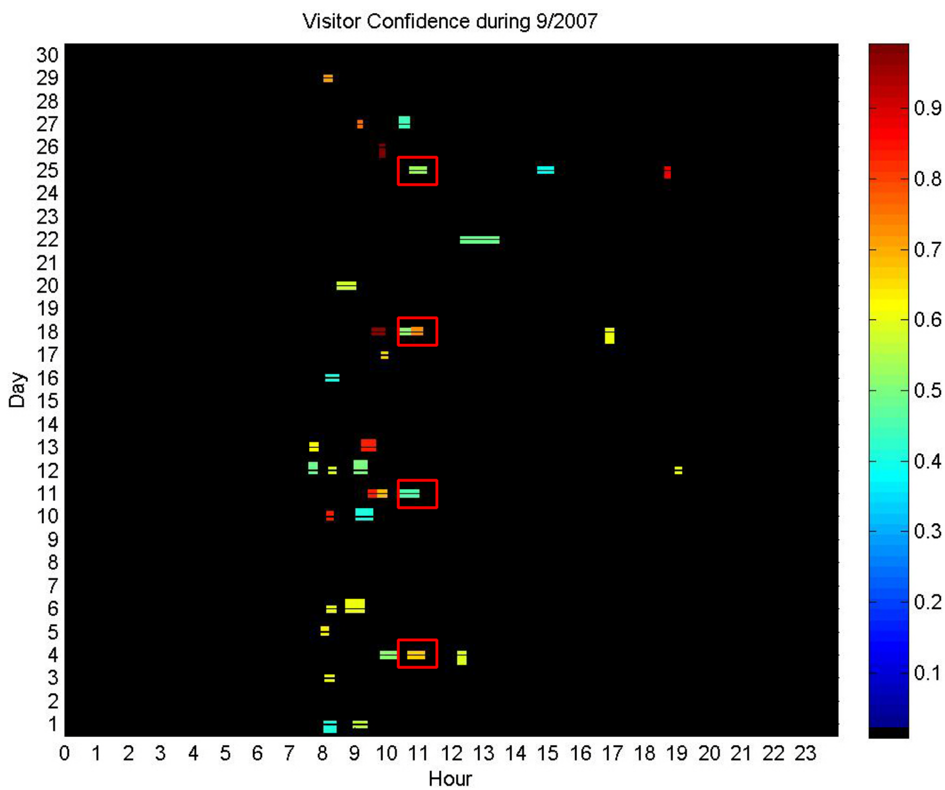


Fig. 6.38: User 3007 function implementation display

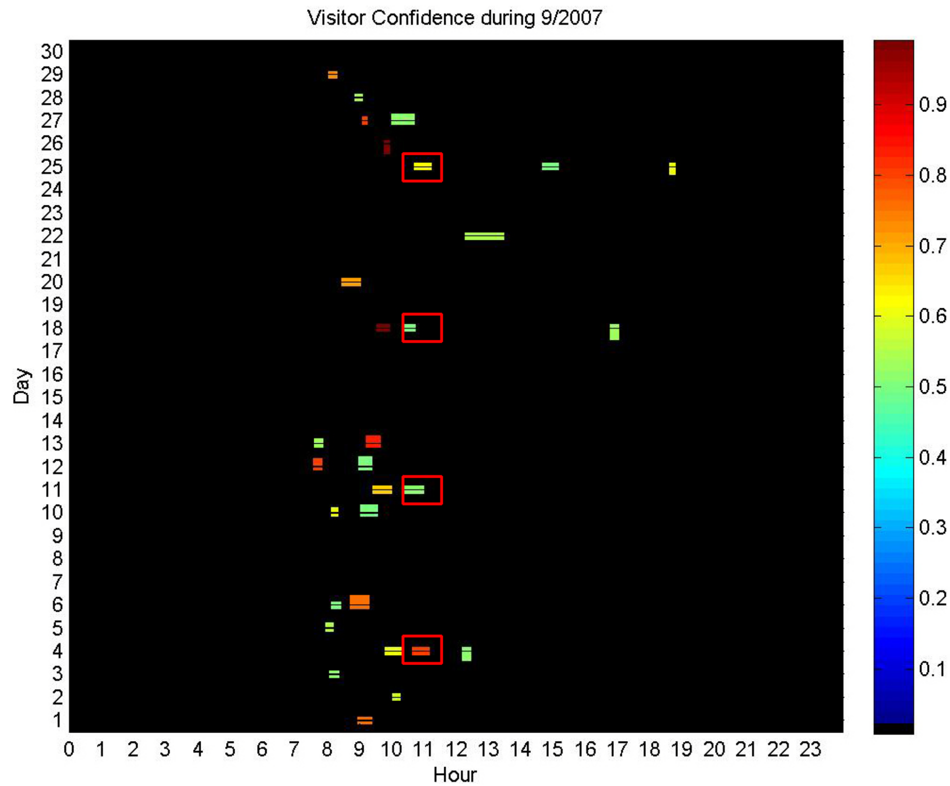


Fig. 6.39: User 3007 Type-1 SISO implementation display

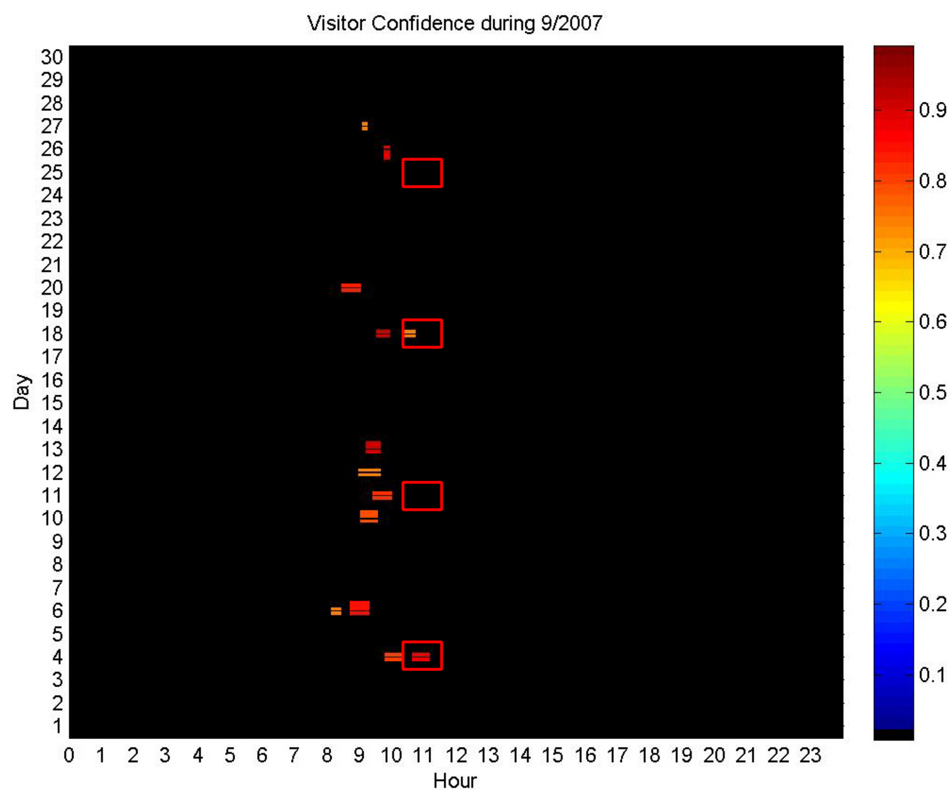


Fig. 6.40: User 3007 Type-1 MISO implementation display

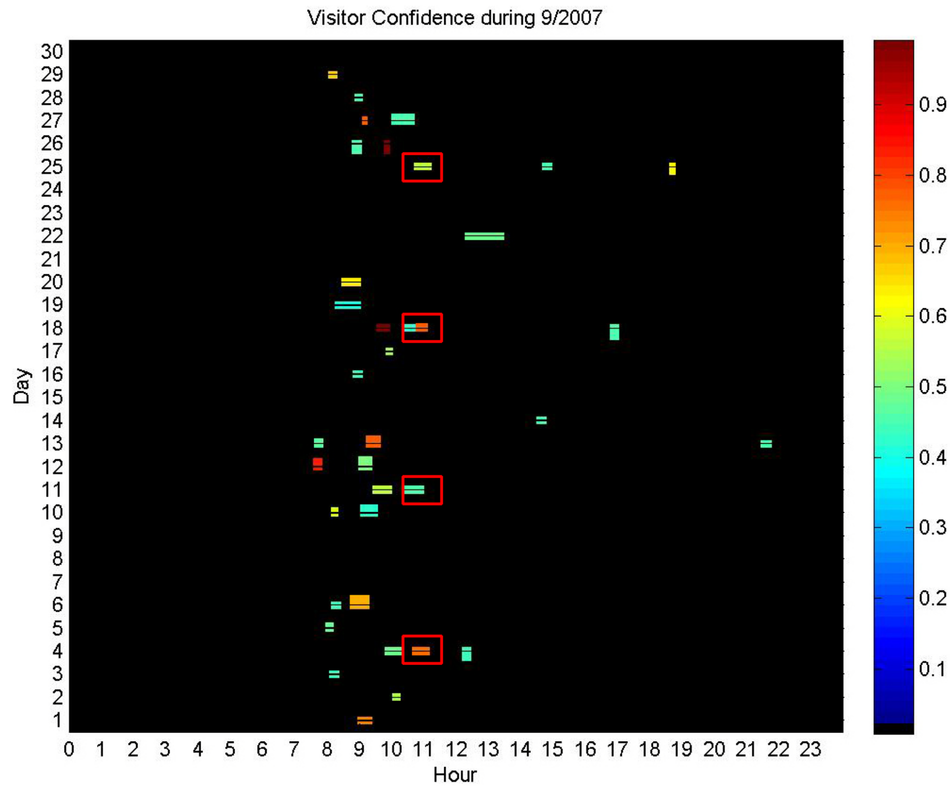


Fig. 6.41: User 3007 Type-2 SISO No FOU implementation display

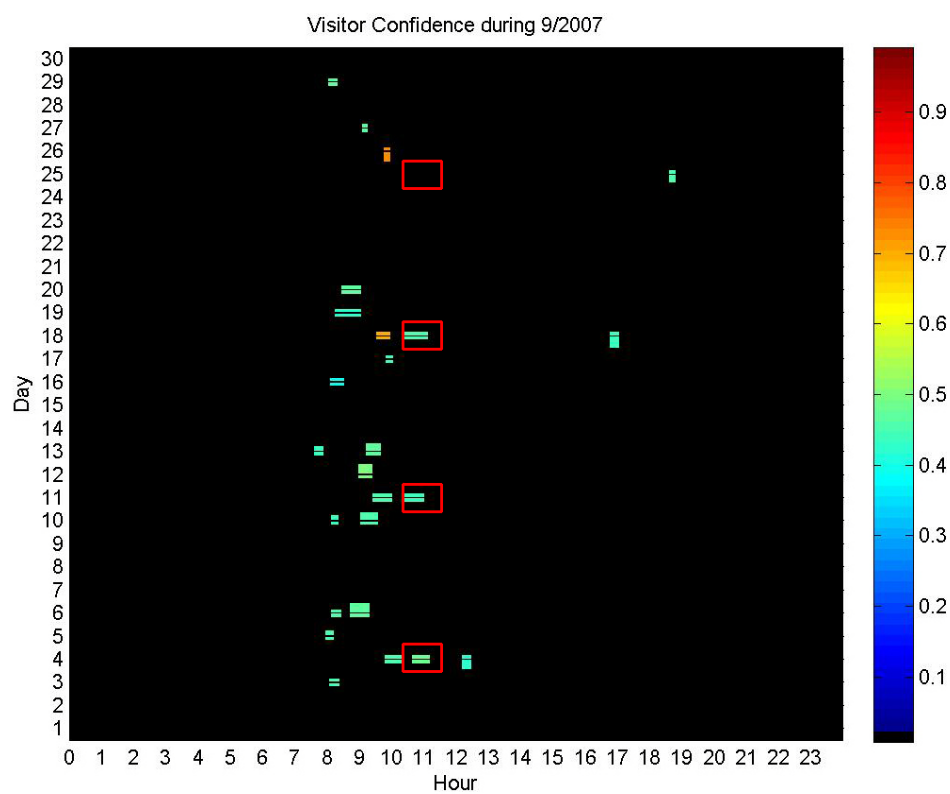


Fig. 6.42: User 3007 Type-2 MISO No FOU 2 Agents implementation display

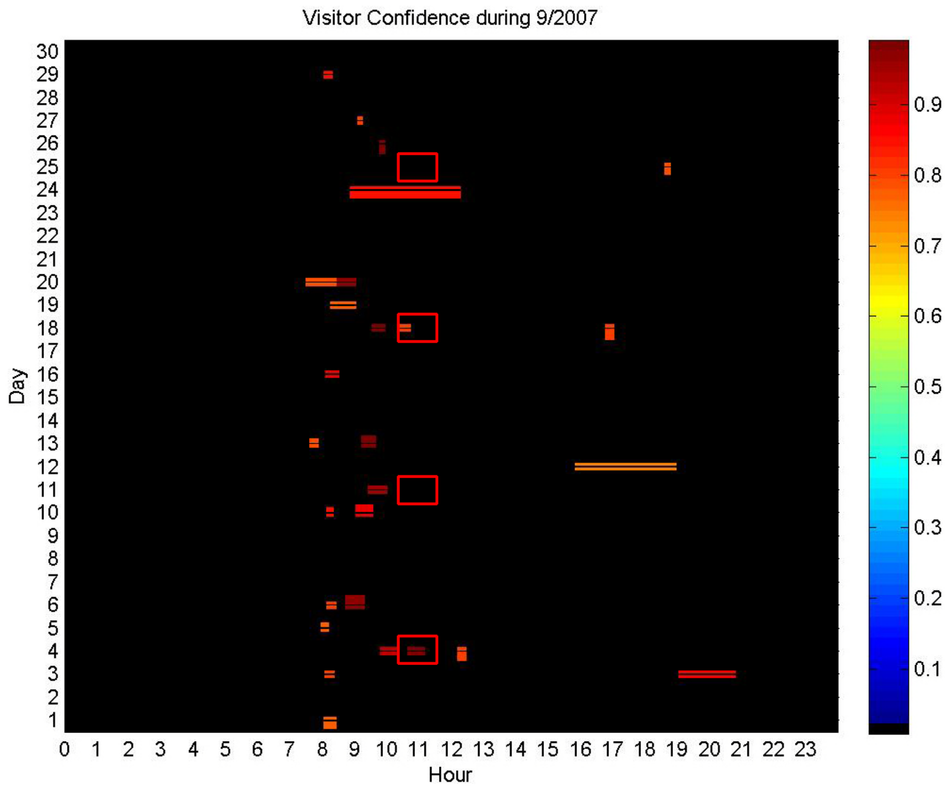


Fig. 6.43: User 3007 Type-2 MISO No FOU 1 Agent implementation display

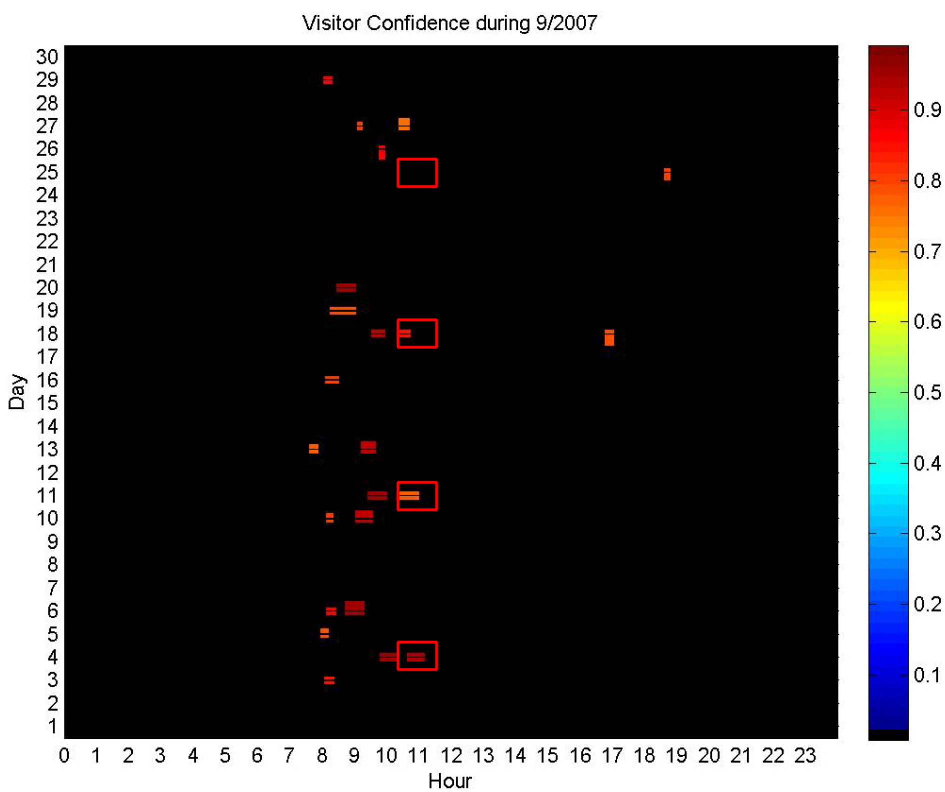


Fig. 6.44: User 3007 Type-2 MISO FOU 2 Agents implementation display

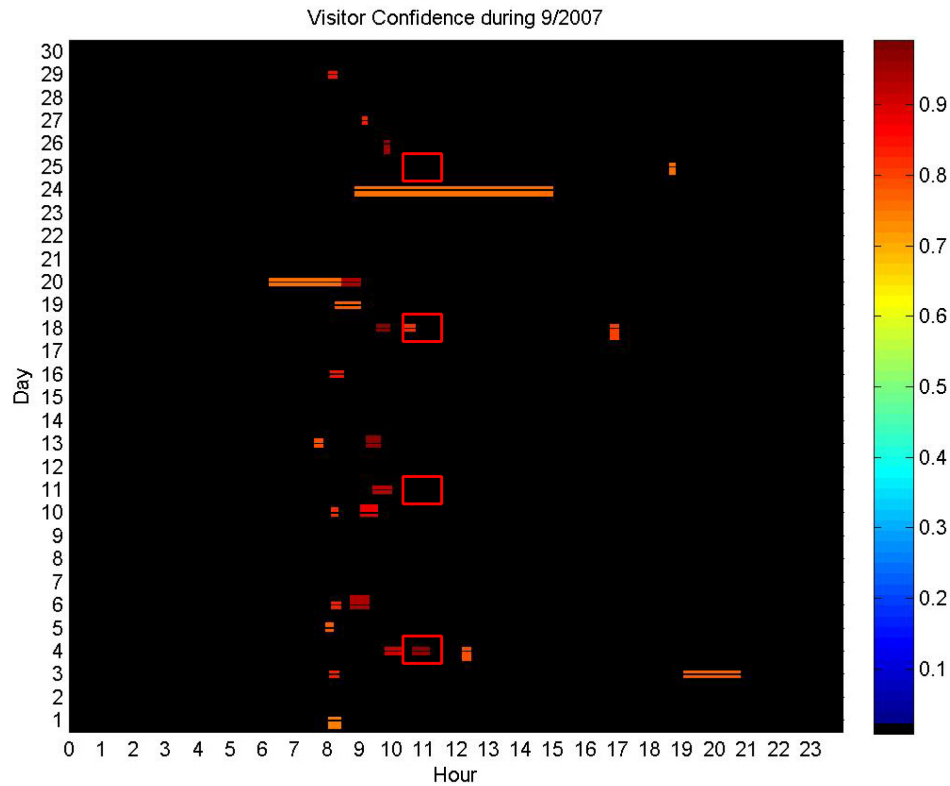


Fig. 6.45: User 3007 Type-2 MISO FOU 1 Agent implementation display

6.2.3 User 3010

User 3010 regulated their day only slightly more than User 3007. The biggest challenge with this user is the high activity levels. There very easily could have been visitors in the apartment much of the time, but it is certain that the Type-2 MISO No FOU 2 Agent system suppressed the most noise while detecting all of the housekeeping visits. The data from the Type-2 MISO No FOU 2 Agent system also displays some regularity. Typically, there are visitor detections within a day of the cleaning staff, in the early afternoon.

While it is clear that this resident has a lot more activity in their room than User 3007, the addition activity makes it difficult to distinguish visitors. The function classifier is a good example of this, because it includes a significant number of potential visitor times. While the function classifier has confidence in a lot of smaller regions, the

Type-1 MISO No FOU 2 Agent system has a high confidence in very few visitor times. However, the Type-1 MISO No FOU 2 Agent system does not detect the last housekeeping visit. The Type-1 and Type-2 SISO systems produced the noisiest displays, which appear to correlate to the morning routine of the resident. It is possible that this resident had help in the morning as well. The Type-2 MISO FOU 2 Agent classifier detected a few more cases where visitors were likely in the apartment compared to the Type-1 MISO system, but still missed the last housekeeping visit. As before, the single agent implementations classified long stretches of time as visitor times, but otherwise were similar to their 2 Agent counterparts. My favored implementation for this user is the Type-2 MISO No FOU 2 Agent system. It is able to clean up the visitor confidence throughout the month, while still giving a reasonable confidence to known visitor regions.

Figures 6.47 thru 6.54 show one month taken from the display data. The housekeeping times are bounded by a red rectangle. These are based on the 10 minute minimum event length. For comparison, I also included the motion density plot in figure 6.46.

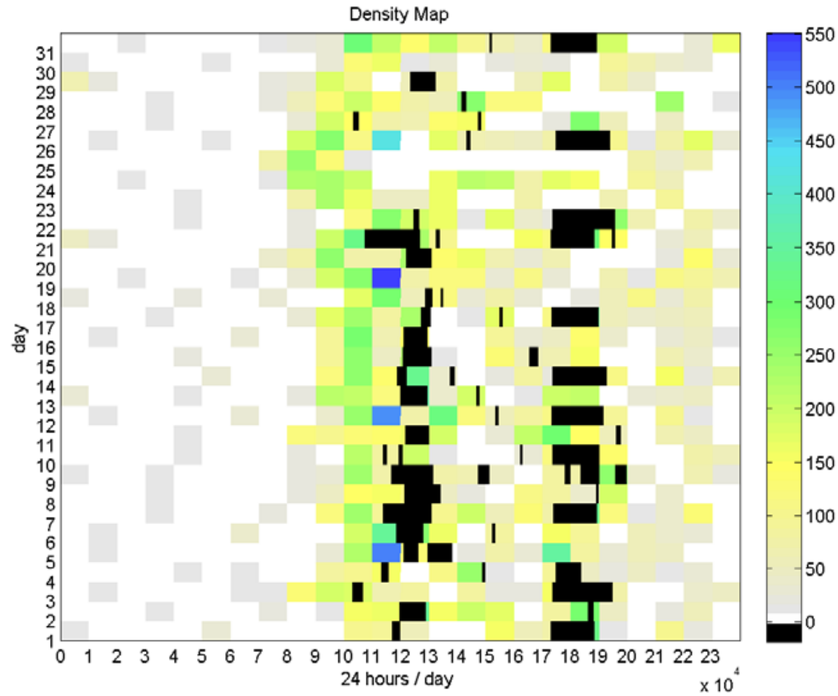


Fig. 6.46: Motion density plot for User 3010

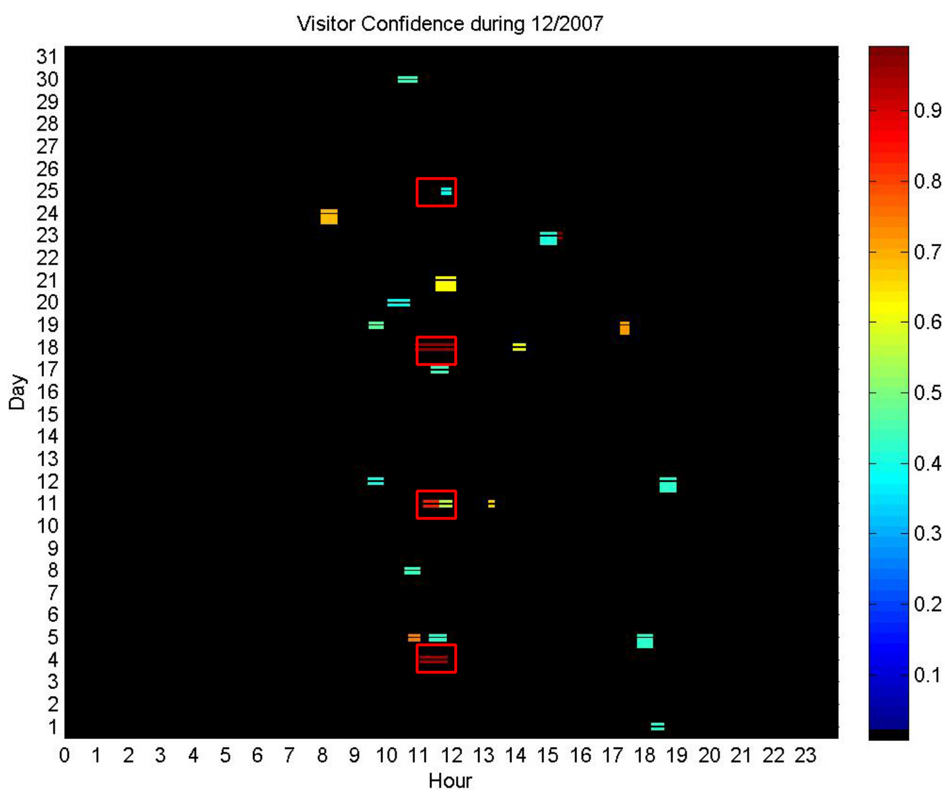


Fig. 6.47: User 3010 function implementation display

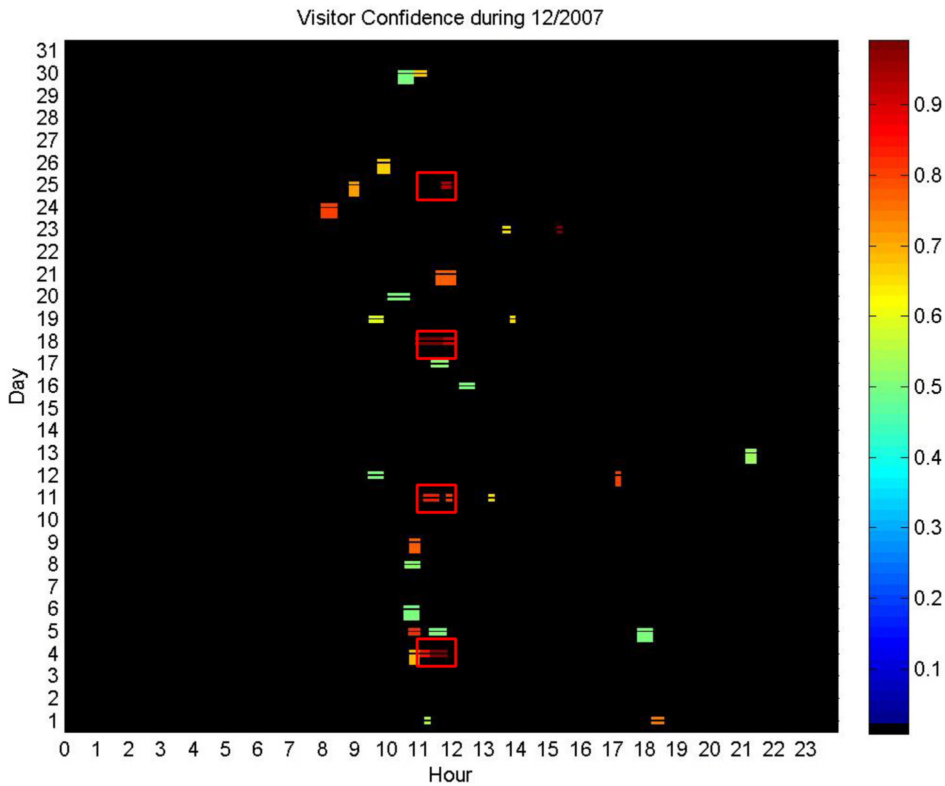


Fig. 6.48: User 3010 Type-1 SISO implementation display

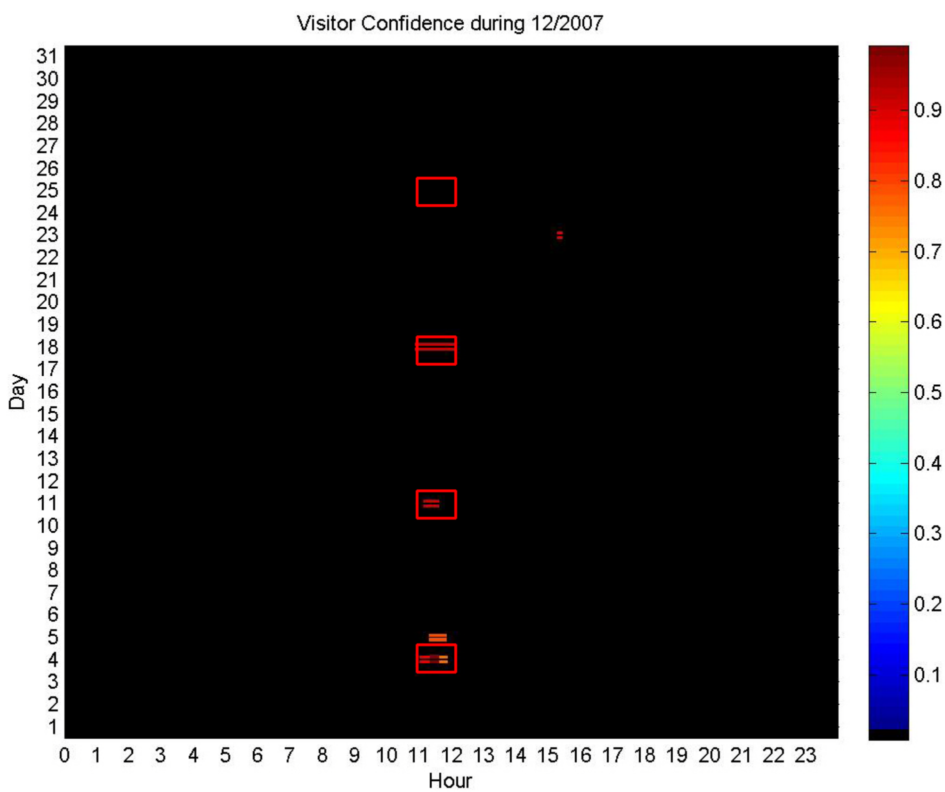


Fig. 6.49: User 3010 Type-1 MISO implementation display

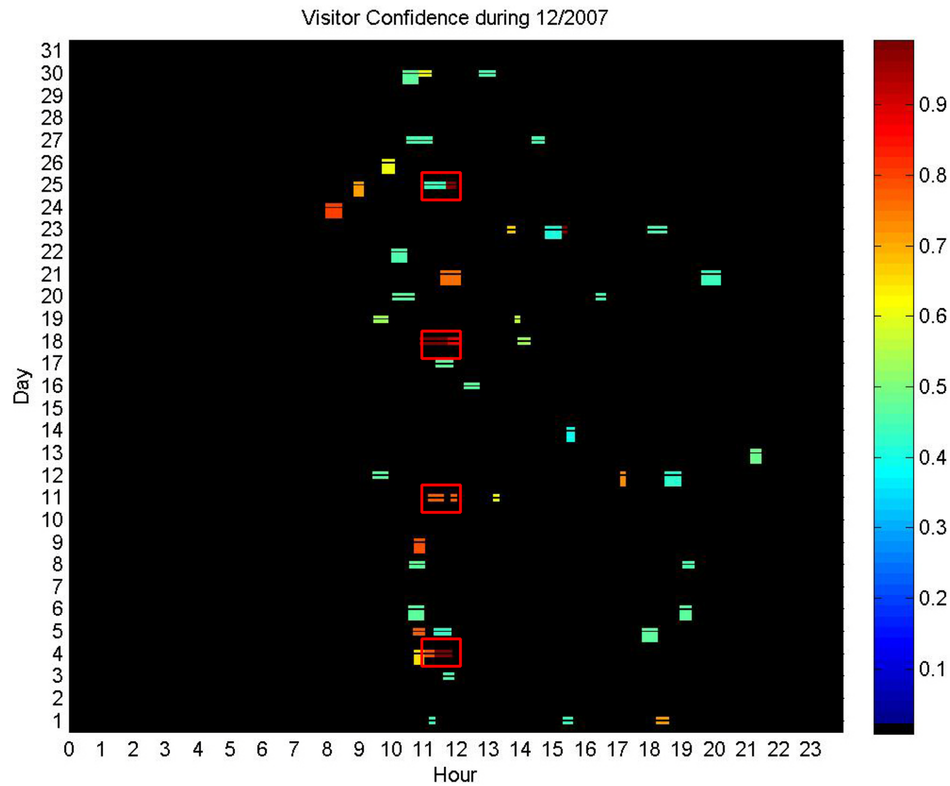


Fig. 6.50: User 3010 Type-2 SISO No FOU implementation display

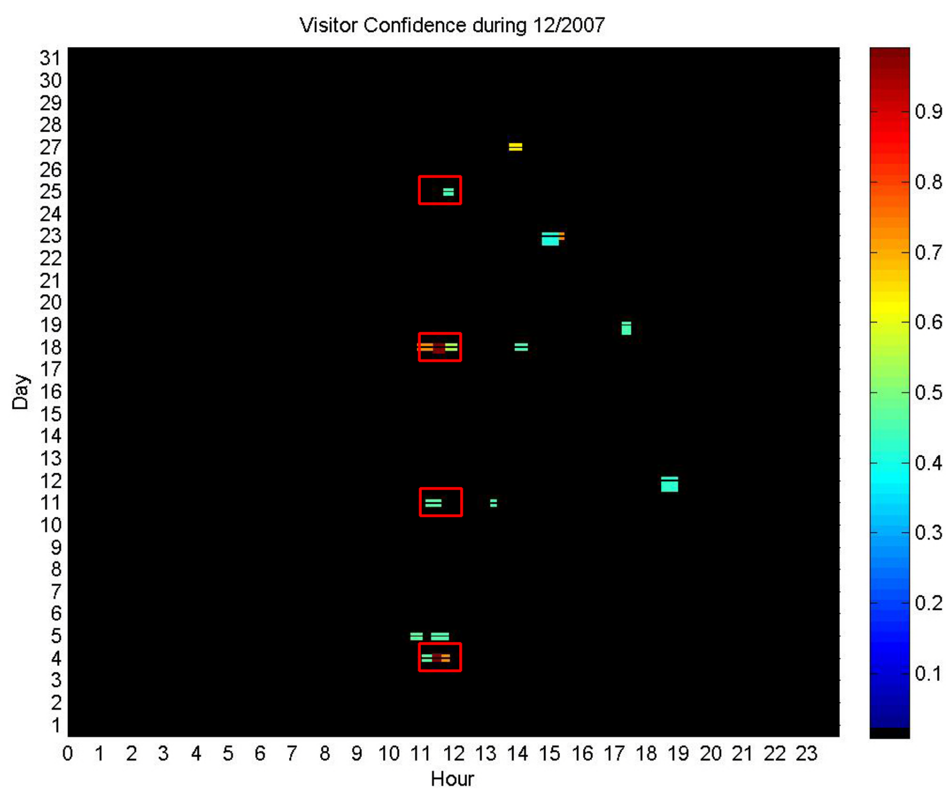


Fig. 6.51: User 3010 Type-2 MISO No FOU 2 Agents implementation display

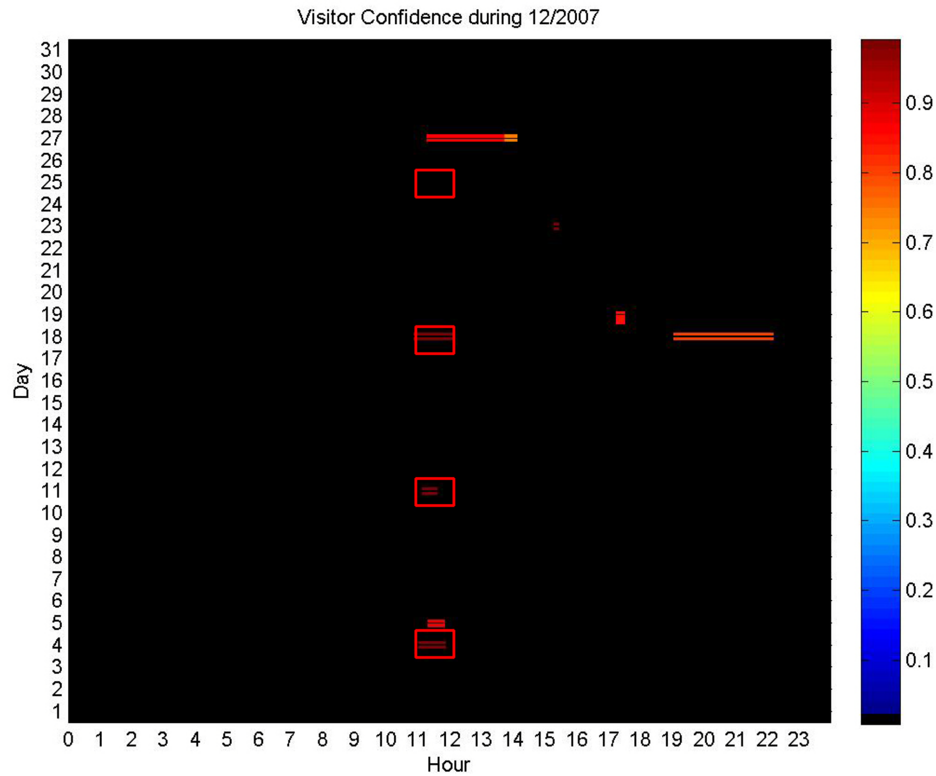


Fig. 6.52: User 3010 Type-2 MISO No FOU 1 Agent implementation display

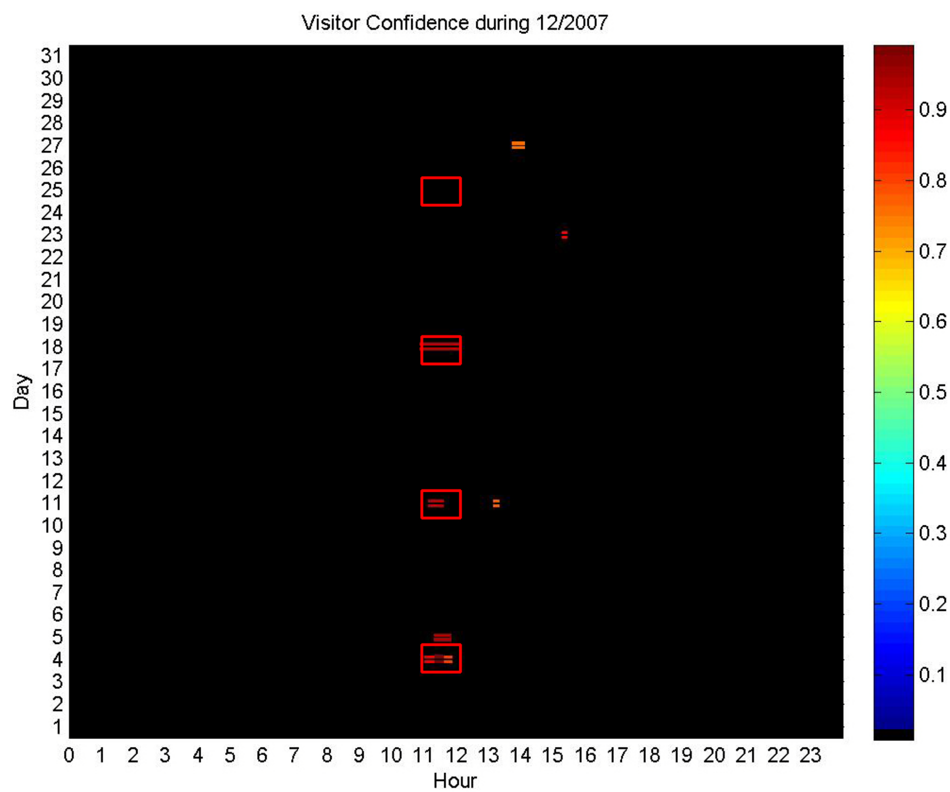


Fig. 6.53: User 3010 Type-2 MISO FOU 2 Agents implementation display

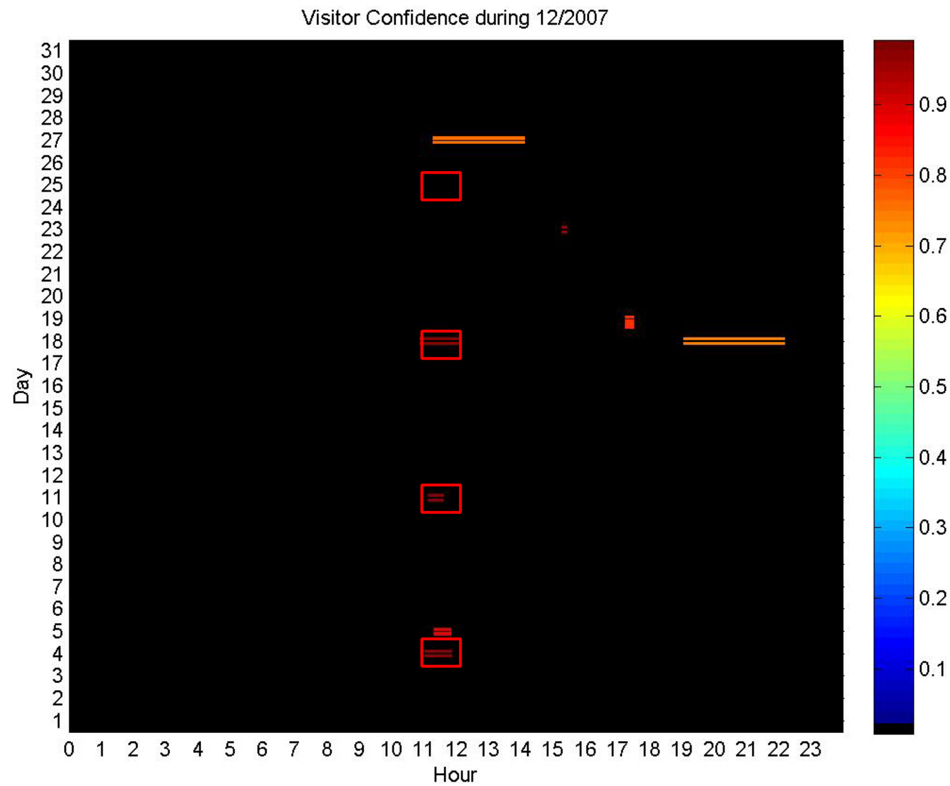


Fig. 6.54: User 3010 Type-2 MISO FOU 1 Agent implementation display

6.3 Discussion of the Results

The goal of this research was to build an algorithm that could accurately identify times when visitors were in the apartment. Two methods were explored for the evaluation of the effectiveness of the algorithm. The more objective method is the ROC curve. However, there is very little reliable ground truth data from the resident apartments. The only recorded visitor information is from the cleaning staff.

There are two cases in which this algorithm is most likely to fail, assuming the sensor layout and measurements are implemented correctly. The first case is when two sedentary individuals enter the apartment and stay in the same room. There is no significant increase in the motion density, and no velocity data. This failure is more of a problem for the social health measurement concern than the activity level (the activity displayed is normal for the resident alone). The second potential area of failure is if either the sensors malfunction (fire without motion) and peak the maximum velocity, or if the resident is frequently changing rooms (running up the motion density). The empirical results display robustness for this problem.

Using a 5 minute minimum event length turned out to be a too short for the resident data. The noise reduction was greatly improved by increasing to a 10 minute minimum event length. Though the test apartment data worked well with a 15 minute minimum event length, many of the visitor times were being lost or combined over an unrealistic amount of time (Ex. 8:00 PM to 7:00 AM). Ten minutes turned out to be a good parameter choice. Noisy information near the doorway is reduced, but documented visits still show up well.

I found that the minimum event length has a far reaching impact on the preferential classification system. When using a 5 minute minimum event length, the systems produced erratic results. But when the minimum event length is increased to 10 minutes, the fuzzy systems significantly improve (particularly the MISO systems). The improved results approach the function classifier at 87.9% classification accuracy. A test of 15 minute minimum event lengths improves all of the classifiers to 87.9%, but as was previously stated, this number is biased to the length of the test series.

It is clear from all of the resident data that the confidence threshold will be much different from individual to individual. The test apartment data was able to achieve a high degree of accuracy at unreasonable confidence levels. It is also interesting that the Type-2 MISO systems performed better without attempting to create an FOU. The 2 Agent implementation was only slightly different from the Type-1 MISO system, based on a limitation of Mendel's software [30], which is explained in Section 4.9. A final pair of single Agent implementations did not improve the results. The Type-2 MISO FOU systems greatly increased the confidence levels, but likely reduced the number of detected visitor times.

Chapter 7

CONCLUSIONS AND FURTHER STUDY

7.1 Conclusions

This project turned out to be much more difficult than I had originally anticipated. The most difficult part is defining helpful features from the data. Motion sensors are obviously going to be good at detecting motion, but the data has a very low dimensionality. It is a good challenge to create a high level detection scheme from low level data.

The fuzzy Type-2 system has a lot of upsides. The ability to fuzzify the memberships improves the algorithm in two areas. The first is that I do not have to be an expert on each resident's motion data profile to create a reliable classifier. The second improvement is that the classifier generalizes pretty well over a variety of activity profiles.

The major problem with Type-2 systems is that they take an inordinate amount of time to produce data. Compared to the Type-1 system, the Type-2 system works almost 7.5 times as long. As the number of events rises, the difference quickly becomes overwhelming. For instance, the fuzzy Type-2 MISO system took almost 3 hours to run the combine events algorithm. It is important to keep in mind that User 3004 has recorded data for two and a half years, and in practice the algorithm will likely only be processing data for one month at a time.

The Type-2 MISO FOU 2 Agents and No FOU 2 Agents systems produce results too similar to make a decision based on resident data. The data from the test apartment suggests that the No FOU system performs slightly better. I would consider this too close to call, but personally favor the No FOU system. It may just be a matter of trusting results that have such a high confidence where I am uncertain at best. There is also the case where the Type-2 MISO FOU system did not detect the housekeeping for User 3010, albeit a short visit.

While it is surprising that the function performs so well on the test data, it ended up recording a lot of artifacts, causing the resulting display to be noisy. This is a good method for quick tests, possibly real time tracking. I think with more work, a better function could be derived to model the interaction of the features.

The fuzzy Type-1 system is a fair compromise between the function and the Type-2 systems. While not as fast as the function approximation, it is better at removing a lot of the noise. I am basing this opinion primarily on the empirical data collected for the users, but the Type-1 system performed pretty well on the test apartment data as well.

There are mixed results when comparing SISO to MISO fuzzy systems. The SISO system boosts the Type-1 classification accuracy by 5%, but causes a drop of 7.5% with the Type-2 system. There is a definite advantage to the SISO system in run time. By simplifying the fuzzy inference (removing complex rules), the run time for the Type-2 system is cut in half. In spite of the processing complexity problem, there are two reasons I recommend the MISO system over the SISO system:

1. MISO system empirically performs much better in the Type-2 case on resident data.

2. By choosing classification performance over processing performance when selecting to go with a Type-2 system, the additional processing time is less of a problem.

I find it fascinating that the 2 Agent implementations appear to have performed better in the residential displays than their 1 Agent counterparts. Much of this has to do with the Minimum Event Length Algorithm, which combines similar confidences. Regardless, I would have figured the traditional approach to the Type-2 implementation would return, by far, the best results. I did not find that in this work.

Overlapping sensor areas that are not documented will artificially increase the velocity. It is impossible to get reasonable measurements of the distance traveled between motion sensors otherwise. If it is known that the sensors overlap, the distance in the algorithm can be set to zero. Once the distance is zero, no velocity information will be calculated between the two points.

Depending on the specific requirements for the algorithm, another potential problem occurs when multiple occupants stay in the same room or in adjacent areas. Without the benefit of velocity data, the motion sensors would need to be very active to increase the confidence in the visitor.

The last area of concern is dependant on how the user wants to define the social levels of the resident. I would classify the resident as engaged in social activity while outside of the apartment. It is difficult to specifically identify times when there are multiple occupants in the apartment. Additionally, there is no guarantee that one of the multiple occupants is the resident.

I feel this is algorithm meets the need, and performs admirably on the data available. There are many steps that can be taken to improve the environment that would positively impact the effectiveness of this algorithm.

7.2 Future Study

It may be possible to take better advantage of the matrix functionality in Matlab to reduce processing costs. It would be particularly valuable when calculating the Type-2 system. There are a lot of intensive calculations, such as the EKM (Enhanced Karnik-Mendel) algorithm, that take a considerable amount of the processing time per pass. Currently the problem with pursuing this tactic is that events are being combined on the fly, some of which is based on confidence.

While this algorithm is designed in part to help determine the activity level of the resident alone, the results from the activity level algorithm could be used to further refine the memberships and confidences of this algorithm. The cyclic nature of the process would allow both algorithms to learn, so long as the cycle does not enter an unstable state.

Another means for training the system is to use a neural network or genetic algorithm to create an expert system from ground truth data. Training a neural network is black boxing the algorithm, but a system that can be trained on the individual is ideal. The system may initially learn from training data produced by through the fuzzy visitor detection algorithm.

The final task will be integrating the information produced by this algorithm into the system being created by the Eldertech team. Most of the project that is associated with the motion sensors is written in Matlab, so the integration should be smooth after the timestamps are associated.

BIBLIOGRAPHY

[1] Wang, S. and Skubic, M., "Density map visualization from motion sensors for monitoring activity level", Proc. of the 2008 IET Conf. on Intelligent Environments, July 21-22, 2008 pp. 1 – 8.

[2] Zadeh, L., "Fuzzy sets", Information and Control 8, 1965 pp. 338 – 353.

[3] Zadeh, L., "The concept of a linguistic variable and its application to approximate reasoning – I", Information Sciences, Vol. 8, No. 3, July 1975 pp. 199-249.

[4] Zadeh, L., "The concept of a linguistic variable and its application to approximate reasoning – II", Information Sciences, Vol. 8, No. 4, October 1975 pp. 301 – 357.

[5] Zadeh, L., "The concept of a linguistic variable and its application to approximate reasoning – III", Information Sciences, Vol. 9, No. 1, December 1975 pp. 43 – 80.

[6] John, R. and Coupland, S., "Extensions to Type-1 fuzzy: Type-2 fuzzy logic and uncertainty", In Gary Y. Yen and David B. Fogel, editor, Computational Intelligence: Principles and Practice, 2006 pp. 89 – 102.

[7] Mendel, J., John, R., and Liu, F., "Interval Type-2 fuzzy logic systems made simple", IEEE Transactions on Fuzzy Systems, December 2006 pp. 808 – 821.

[8] Mendel, J., "Type-2 fuzzy sets and systems: An overview", Computational Intelligence Magazine, IEEE, Vol. 2, Issue 2, Part Supplement, May 2007 pp. 20 – 29.

[9] Ye, X., Fu, L., and Zhang, Y., "Type-2 fuzzy logic system and level set", Semantics, Knowledge and Grid, Third International Conference on, October 29-31, 2007 pp. 80 – 85.

[10] Wu, D. and Mendel, J., "Enhanced Karnik-Mendel algorithms for Interval Type-2 fuzzy sets and systems", North American Fuzzy Information Processing Society, 2007, June 24-27, 2007 pp.184 – 189.

[11] Hagras, H., "A hierarchical Type-2 fuzzy logic control architecture for autonomous mobile robots", IEEE Transactions on Fuzzy Systems, Vol. 12, Issue 4, August 2004 pp. 524 – 539.

[12] Hagras, H., "A Type-2 fuzzy logic controller for autonomous mobile robots", Proc. of the 2004 IEEE Conf. on Fuzzy Systems, Vol. 2, July 25-29, 2004, pp. 965 – 970.

[13] Hagras, H., "Type-2 FLCs: A new generation of fuzzy controllers", Computational Intelligence Magazine, IEEE, Vol. 2, Issue 1, February 2007 pp. 30 – 43.

[14] Jammeh, E., Fleury, M., Wagner, C., Hagras, H., and Ghanbari, M., "Interval Type-2 fuzzy logic congestion control of video streaming", Proc. of the 2008 IET Conf. on Intelligent Environments, July 21-22, 2008 pp. 1 – 8.

[15] Wang, D. and Acar, L., "An analysis of Type-1 and Type-2 fuzzy logic systems", Proc. of the 1999 IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics, September 15-17, 1999 pp. 353 – 358.

[16] Paetz, J., "On the role of numerical preciseness for generalization, classification, Type-1, and Type-2 fuzziness", Proc. of the 2007 IEEE Symposium on Foundations of Computational Intelligence, April, 1-5, 2007 pp. 208 – 213.

[17] Gaddam, A., Mukhopadhyay, S., Sen Gupta, G., and Guesgen, H., "Wireless sensors networks based monitoring: Review, challenges and implementation issues", Proc. of the 2008 ICST Conf. on Sensing Technology, November 30, 2008 - December 3, 2008 pp. 533 – 538.

[18] Glascock, A. and Kutzik, D., "An evidentiary study of the uses of automated behavioral monitoring", Proc. of the 2007 International Conf. on Advanced Information Networking and Applications Workshops, Vol. 2, May 21-23, 2007 pp. 858 – 862.

[19] Wu, J., Bao, X., Cheng, D., and Sriram, R., "Sensor augmented aging in place", Proc. of the 2004 IEEE International Workshop on Biomedical Circuits and Systems, December 1-3, 2004 pp. 13 – 16.

[20] Hudson, D. and Cohen, M., "Intelligent agent model for remote support of rural healthcare for the elderly", Proc. of the 2006 IEEE International Conf. on Engineering in Medicine and Biology Society, August 30, 2006 - September 3, 2006 pp. 6332 – 6335.

[21] Le, X., Di Mascolo, M., Gouin, A., and Noury, N., "Health Smart Home - Towards an assistant tool for automatic assessment of the dependence of elders", Proc. of the 2007 International Conf. on Engineering in Medicine and Biology Society, August 22-26, 2007 pp. 3806 – 3809.

[22] Dishman, E., "Inventing wellness systems for aging in place", Computer, Vol. 37, Issue 5, May 2004 pp. 34 – 41.

[23] Popescu, M., Florea, E., Skubic, M., and Rantz, M., "Prediction of elevated pulse pressure in elderly using in-home monitoring sensors: A pilot study", Proc. of the 2007 IET International Conf. on Intelligent Environments, July 21-22, 2008 pp. 1 – 4.

[24] Tyrer, H., Alwan, M., Demiris, G., He, Z., Keller, J., Skubic, M., and Rantz, M., "Technology for successful aging", Proc. of the 2006 IEEE Conf. on Engineering in Medicine and Biology Society, August 30, 2006-September 3, 2006 pp. 3290 – 3293.

[25] Anderson, D., Luke, R., Keller, J., Skubic, M., Rantz, M.J., and Aud, M., "Modeling human activity from Voxel person using fuzzy logic", IEEE Transactions on Fuzzy Systems, Vol. 17, Issue 1, February 2009 pp. 39 – 49.

[26] Martin, T., Majeed, B., Lee, B., and Clarke, N., "Fuzzy ambient intelligence for next generation telecare", Proc. of the 2006 IEEE Conf. on Fuzzy Systems, July 16-21, 2006 pp. 894 – 901.

[27] Dara-Abrams, B., "Toward a model for collaborative gerontechnology: Connecting elders and their caregivers", Proc. of the 2008 International Conf. on Creating, Connecting and Collaborating through Computing, January 14-16, 2008 pp. 109 – 114.

[28] The National Strategy for Suicide Prevention website, http://mentalhealth.samhsa.gov/suicideprevention/elderly.asp, May 1, 2009.

[29] Bezdek, J., "The thirsty traveler visits Gamont: a rejoinder to Comments on fuzzy sets-what are they and why?" IEEE Transactions on Fuzzy Systems, Vol. 2, Issue 1, February 1994 pp. 43 – 45.

[30] Karnik, N., Liang, Q., Liu, F., Wu, D., and Mendel, J., http://sipi.usc.edu/~mendel/software/, May 1, 2009.

APPENDIX

Appendix A

TABLES AND GRAPHS OF MOTION SENSOR TEST RESULTS

Motion sensing with infrared sensor through various materials
Lab temperature: 76º F

	Body heat (93)	low heat (92-94)	med heat (96-98)	high heat (106-108)
control	hit	hit	hit	hit
tape	hit	hit	hit	hit
copy paper	no hit	no hit	no hit	no hit
index card	no hit	no hit	no hit	no hit
wax paper	hit	hit	hit	hit
Al shiny in	hit	no hit	hit	no hit
Al dull in	hit	no hit	hit	no hit

I used a refrigerator thermometer to detect the temperature of the heating pad and room

Standard configuration, no baffle, low heat

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	18°	162°	**144°**	65°	117°	**52°**
B2	15°	162°	**147°**	64°	117°	**53°**
B3	18°	162°	**144°**	67°	119°	**52°**
B4	13°	160°	**147°**	67°	115°	**48°**
B5	14°	161°	**147°**	65°	119°	**54°**
B6	14°	160°	**146°**	62°	118°	**56°**
B7	13°	159°	**146°**	65°	117°	**52°**
B8	14°	160°	**146°**	68°	120°	**52°**
B9	14°	167°	**153°**	67°	115°	**48°**
B10	11°	164°	**153°**	67°	121°	**54°**
Avg	14°	162°	147°	66°	118°	52°
StDev	2°	2°	3°	2°	2°	2°

Standard configuration, no baffle, med heat

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	15°	166°	**151°**	66°	117°	**51°**
B2	15°	167°	**152°**	63°	118°	**55°**
B3	12°	163°	**151°**	68°	122°	**54°**
B4	13°	167°	**154°**	65°	121°	**56°**
B5	14°	162°	**148°**	65°	118°	**53°**
B6						
B7	12°	163°	**151°**	61°	117°	**56°**
B8	13°	164°	**151°**	62°	121°	**59°**
B9	14°	167°	**153°**	62°	121°	**59°**
B10	11°	165°	**154°**	62°	121°	**59°**
Avg	13°	165°	152°	64°	120°	56°
StDev	1°	2°	2°	2°	2°	3°

Standard configuration - with Fresnel lenses, no baffle, heat difference

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	3°	4°	**7°**	-1°	0°	**-1°**
B2	0°	5°	**5°**	1°	1°	**2°**
B3	6°	1°	**7°**	-1°	3°	**2°**
B4	0°	7°	**7°**	2°	6°	**8°**
B5	0°	1°	**1°**	0°	-1°	**-1°**
B6						
B7	1°	4°	**5°**	4°	0°	**4°**
B8	1°	4°	**5°**	6°	1°	**7°**
B9	0°	0°	**0°**	5°	6°	**11°**
B10	0°	1°	**1°**	5°	0°	**5°**
Avg	1°	3°	4°	2°	2°	4°
StDev	2°	2°	3°	3°	2°	4°

First Alternate, 14mm copy paper baffle, low heat

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	63°	117°	**54°**	67°	112°	**45°**
B2	66°	115°	**49°**	65°	113°	**48°**
B3	66°	120°	**54°**	66°	111°	**45°**
B4	65°	113°	**48°**	66°	114°	**48°**
B5	66°	114°	**48°**	68°	105°	**37°**
B6						
B7	64°	117°	**53°**	68°	106°	**38°**
B8	60°	120°	**60°**	68°	115°	**47°**
B9	66°	114°	**48°**	65°	107°	**42°**
B10	65°	114°	**49°**	67°	110°	**43°**
Avg	65°	116°	51°	67°	110°	44°
StDev	2°	2°	4°	1°	3°	4°

First Alternate, 14mm copy paper baffle, med heat

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	60°	120°	**60°**	66°	115°	**49°**
B2	63°	121°	**58°**	66°	114°	**48°**
B3	64°	121°	**57°**	65°	113°	**48°**
B4	60°	118°	**58°**	64°	115°	**51°**
B5	63°	117°	**54°**	66°	113°	**47°**
B6						
B7	66°	119°	**53°**	66°	112°	**46°**
B8	59°	121°	**62°**	68°	119°	**51°**
B9	63°	119°	**56°**	65°	112°	**47°**
B10	62°	117°	**55°**	64°	113°	**49°**
Avg	62°	119°	57°	66°	114°	48°
StDev	2°	2°	3°	1°	2°	2°

First Alternate, 14mm copy paper baffle, heat difference

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	3°	3°	**6°**	1°	3°	**4°**
B2	3°	6°	**9°**	-1°	1°	**0°**
B3	2°	1°	**3°**	1°	2°	**3°**
B4	5°	5°	**10°**	2°	1°	**3°**
B5	3°	3°	**6°**	2°	8°	**10°**
B6						**0°**
B7	-2°	2°	**0°**	2°	6°	**8°**
B8	1°	1°	**2°**	0°	4°	**4°**
B9	3°	5°	**8°**	0°	5°	**5°**
B10	3°	3°	**6°**	3°	3°	**6°**
Avg	2°	3°	6°	1°	4°	4°
StDev	2°	2°	3°	1°	2°	3°

Second Alternate, 17mm copy paper baffle, low heat

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	65°	111°	**46°**	76°	103°	**27°**
B2	65°	112°	**47°**	77°	110°	**33°**
B3	67°	110°	**43°**	80°	100°	**20°**
B4	66°	112°	**46°**	77°	103°	**26°**
B5	70°	112°	**42°**	83°	99°	**16°**
B6						
B7	69°	111°	**42°**	84°	98°	**14°**
B8	68°	115°	**47°**	78°	101°	**23°**
B9	69°	111°	**42°**	78°	98°	**20°**
B10	68°	112°	**44°**	80°	102°	**22°**
Avg	67°	112°	44°	79°	102°	22°
StDev	2°	1°	2°	3°	3°	5°

Second Alternate, 17mm copy paper baffle, med heat

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	60°	114°	**54°**	68°	108°	**40°**
B2	61°	113°	**52°**	76°	111°	**35°**
B3	60°	112°	**52°**	70°	105°	**35°**
B4	64°	113°	**49°**	71°	108°	**37°**
B5	64°	113°	**49°**	72°	109°	**37°**
B6						
B7	66°	113°	**47°**	74°	105°	**31°**
B8	62°	120°	**58°**	69°	107°	**38°**
B9	62°	113°	**51°**	72°	103°	**31°**
B10	66°	112°	**46°**	77°	106°	**29°**
Avg	63°	114°	51°	72°	107°	35°
StDev	2°	2°	3°	3°	2°	3°

Second Alternate, 17mm copy paper baffle, heat difference

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	5°	3°	**8°**	8°	5°	**13°**
B2	4°	1°	**5°**	1°	1°	**2°**
B3	7°	2°	**9°**	10°	5°	**15°**
B4	2°	1°	**3°**	6°	5°	**11°**
B5	6°	1°	**7°**	11°	10°	**21°**
B6						**0°**
B7	3°	2°	**5°**	10°	7°	**17°**
B8	6°	5°	**11°**	9°	6°	**15°**
B9	7°	2°	**9°**	6°	5°	**11°**
B10	2°	0°	**2°**	3°	4°	**7°**
Avg	5°	2°	7°	7°	5°	11°
StDev	2°	1°	3°	3°	2°	6°

Fifth Alternate, 14mm copy paper 3mm pinhole baffle, low heat

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	78°	102°	**24°**	87°	91°	**4°**
B2	77°	99°	**22°**	86°	92°	**6°**
B3	82°	107°	**25°**	85°	91°	**6°**
B4	79°	98°	**19°**	87°	92°	**5°**
B5	85°	100°	**15°**	86°	91°	**5°**
B6						
B7	85°	98°	**13°**	90°	91°	**1°**
B8	84°	106°	**22°**	90°	97°	**7°**
B9	85°	98°	**13°**	89°	91°	**2°**
B10	85°	106°	**21°**	91°	93°	**2°**
Avg	82°	102°	19°	88°	92°	4°
StDev	3°	4°	4°	2°	2°	2°

Fifth Alternate, 14mm copy paper 3mm pinhole baffle, med heat

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	71°	109°	**38°**	84°	94°	**10°**
B2	74°	103°	**29°**	85°	92°	**7°**
B3	82°	107°	**25°**	84°	91°	**7°**
B4	72°	106°	**34°**	86°	93°	**7°**
B5	78°	103°	**25°**	85°	91°	**6°**
B6						
B7	84°	109°	**25°**	87°	92°	**5°**
B8	76°	112°	**36°**	82°	92°	**10°**
B9	83°	103°	**20°**	86°	92°	**6°**
B10	83°	107°	**24°**	85°	92°	**7°**
Avg	78°	107°	28°	85°	92°	7°
StDev	5°	3°	6°	1°	1°	2°

Fifth Alternate, 14mm copy paper 3mm pinhole baffle, heat difference

Sensor#	left angle	right angle	horizontal	up angle	down angle	vertical
B1	7°	7°	**14°**	3°	3°	**0°**
B2	3°	4°	**7°**	1°	0°	**-1°**
B3	0°	0°	**0°**	1°	0°	**-1°**
B4	7°	8°	**15°**	1°	1°	**0°**
B5	7°	3°	**10°**	1°	0°	**-1°**
B6						**0°**
B7	1°	11°	**12°**	3°	1°	**-2°**
B8	8°	6°	**14°**	8°	-5°	**-13°**
B9	2°	5°	**7°**	3°	1°	**-2°**
B10	2°	1°	**3°**	6°	-1°	**-7°**
Avg	4°	5°	9°	3°	0°	-3°
StDev	3°	3°	5°	2°	2°	4°

Per pulse

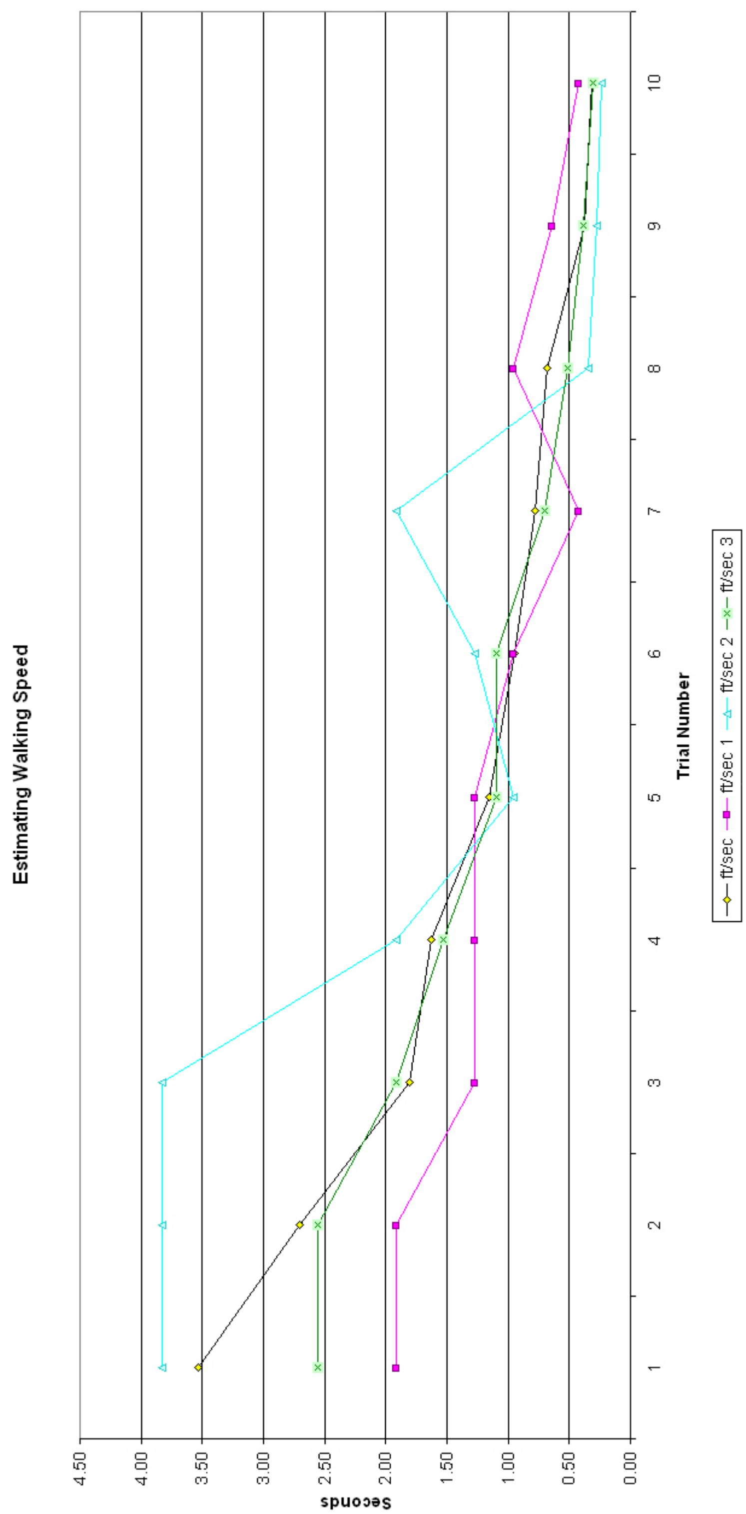
Sensor#	Length of single pulse	
B1	691 px	0.07s
B2	696 px	0.07s
B3	681 px	0.07s
B4	682 px	0.07s
B5	680 px	0.07s
B6	677 px	0.07s
B7	682 px	0.07s
B8	684 px	0.07s
B9	669 px	0.07s
B10	684 px	0.07s
Avg	683 px	0.07s
Std	7 px	0.00s

Per transmission (5 consecutive pulses)

Sensor#	Length of five pulses	
B1	512 px	0.51s
B2	506 px	0.51s
B3	506 px	0.51s
B4	505 px	0.51s
B5	505 px	0.51s
B6	497 px	0.50s
B7	510 px	0.51s
B8	507 px	0.51s
B9	496 px	0.50s
B10	507 px	0.51s
Avg	505 px	0.51s
Std	5 px	0.00s

Time between sensor firings in seconds (100px = 1 sec)

Sensor#	Short		Long		Avg	
B1	664 px	6.64s	765 px	7.65s	715 px	7.15s
B2	692 px	6.92s	747 px	7.47s	720 px	7.20s
B3	691 px	6.91s	737 px	7.37s	714 px	7.14s
B4	673 px	6.73s	731 px	7.31s	702 px	7.02s
B5	688 px	6.88s	718 px	7.18s	703 px	7.03s
B6	695 px	6.95s	695 px	6.95s	695 px	6.95s
B7	671 px	6.71s	716 px	7.16s	694 px	6.94s
B8	655 px	6.55s	716 px	7.16s	686 px	6.86s
B9	684 px	6.84s	741 px	7.41s	713 px	7.13s
B10	700 px	7.00s	746 px	7.46s	723 px	7.23s
Avg	681 px	6.81s	731 px	7.31s		
Std	13 px	0.13s	18 px	0.18s		



Estimating Walking Speed
Seconds
4.50
4.00
3.50
3.00
2.50
2.00
1.50
1.00
0.50
0.00
1
2
3
4
5
6
7
8
9
10
Trial Number
ft/sec
ft/sec 1
ft/sec 2
ft/sec 3

Velocity with Stopwatch

Trial#	Time	ft/sec	B3	B4	B7	B3 to B4	B4 to B7	B3 to B7	ft/sec 1	ft/sec 2	ft/sec 3
1	4.53s	3.53	0.00s	2.00s	3.00s	2.00s	1.00s	3.00s	1.92	3.83	2.56
2	5.91s	2.71	0.00s	2.00s	3.00s	2.00s	1.00s	3.00s	1.92	3.83	2.56
3	8.84s	1.81	0.00s	3.00s	4.00s	3.00s	1.00s	4.00s	1.28	3.83	1.92
4	9.81s	1.63	0.00s	3.00s	5.00s	3.00s	2.00s	5.00s	1.28	1.92	1.53
5	13.81s	1.16	0.00s	3.00s	7.00s	3.00s	4.00s	7.00s	1.28	0.96	1.10
6	16.93s	0.95	0.00s	4.00s	7.00s	4.00s	3.00s	7.00s	0.96	1.28	1.10
7	20.41s	0.78	0.00s	9.00s	11.00s	9.00s	2.00s	11.00s	0.43	1.92	0.70
8	23.41s	0.68	0.00s	4.00s	15.00s	4.00s	11.00s	15.00s	0.96	0.35	0.51
9	42.88s	0.37	0.00s	6.00s	20.00s	6.00s	14.00s	20.00s	0.64	0.27	0.38
10	51.09s	0.31	0.00s	9.00s	25.00s	9.00s	16.00s	25.00s	0.43	0.24	0.31

Note: collisions were a problem with this experiement. We had activity in the downstairs lab tripping interfering signals. The trials affected were discarded.

Appendix B

USER APARTMENT LAYOUTS

Table of User 3004 location distances

	LivingRoon	Kitchen	Bathroom	Bedroom	Office	Laundry	FrontDoor	Shower	Den	Closet
LivingRoon	0	0	216	0	0	74	0	325	0	446
Kitchen	0	0	483	292	0	74	0	425	0	713
Bathroom	216	483	0	0	709	330	738	0	709	0
Bedroom	0	292	0	0	484	254	514	151	484	230
Office	0	0	709	484	0	414	303	842	0	939
Laundry	74	74	330	254	414	0	368	463	414	560
FrontDoor	0	0	738	514	302	368	0	871	303	968
Shower	325	425	0	151	842	463	871	0	842	0
Den	0	0	709	484	0	414	303	842	0	939
Closet	446	713	0	230	939	560	968	0	939	0

UserID: 3004

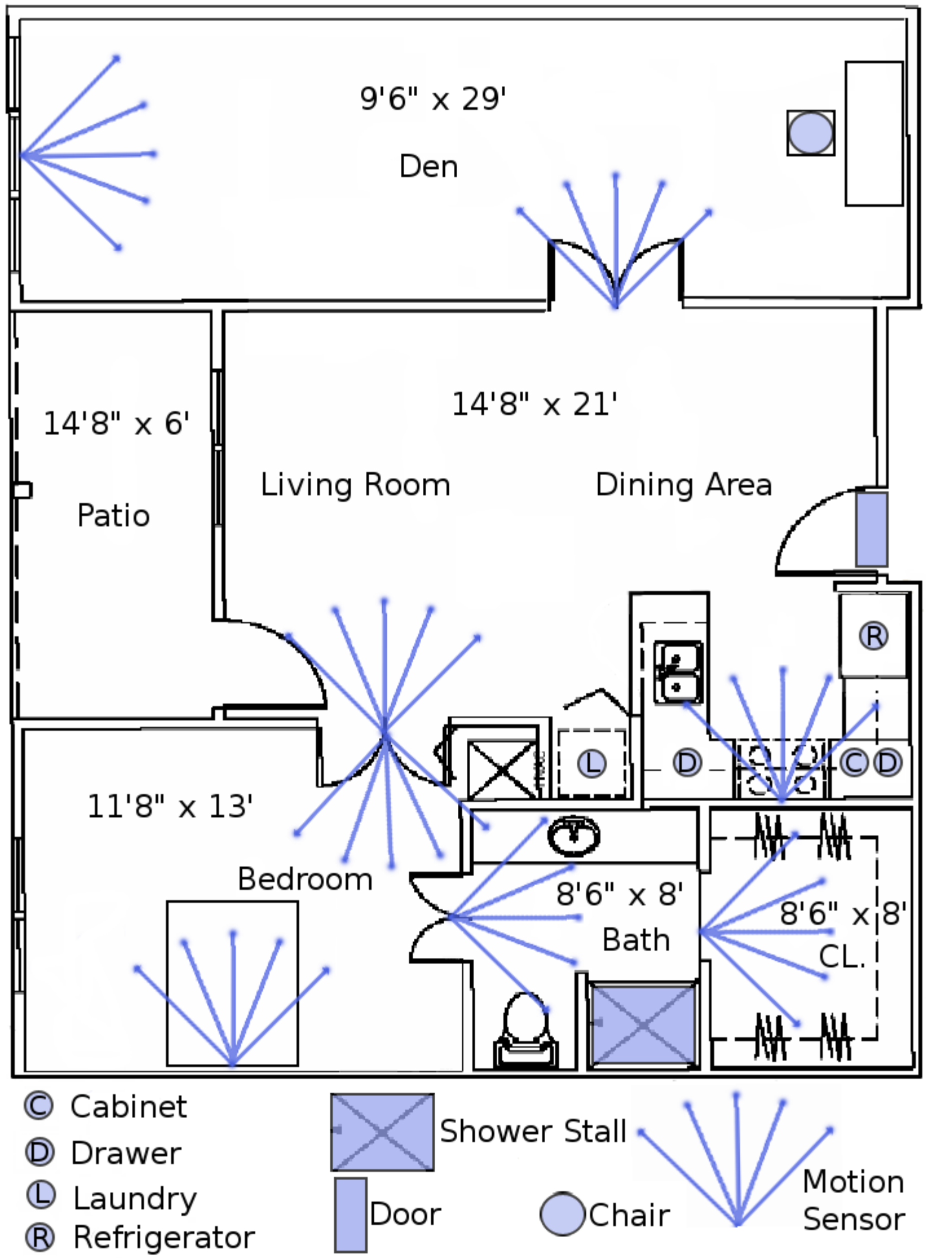

9'6" x 29'
Den
14'8" x 6'
Patio
14'8" x 21'
Living Room
Dining Area
11'8" x 13'
Bedroom
8'6" x 8'
Bath
8'6" x 8'
CL.
C Cabinet
D Drawer
L Laundry
R Refrigerator
Shower Stall
Door
Chair
Motion Sensor

Table of User 3007 location distances

	LivingRoon	Kitchen	Bathroom	Bedroom	Laundry	FrontDoor	Shower	MedicineC	Closet	Patio
LivingRoon	0	0	368	135	0	0	498	0	563	0
Kitchen	0	0	338	167	0	0	468	0	533	0
Bathroom	368	338	0	0	344	556	0	0	0	208
Bedroom	135	167	0	0	176	363	130	0	179	0
Laundry	0	0	344	176	0	270	474	0	523	0
FrontDoor	0	0	556	363	270	0	686	0	735	285
Shower	498	468	0	130	474	686	0	0	0	362
MedicineC	0	0	0	0	0	0	0	0	0	0
Closet	563	533	0	179	523	735	0	0	0	387
Patio	0	0	208	0	0	285	362	0	387	0

UserID: 3007

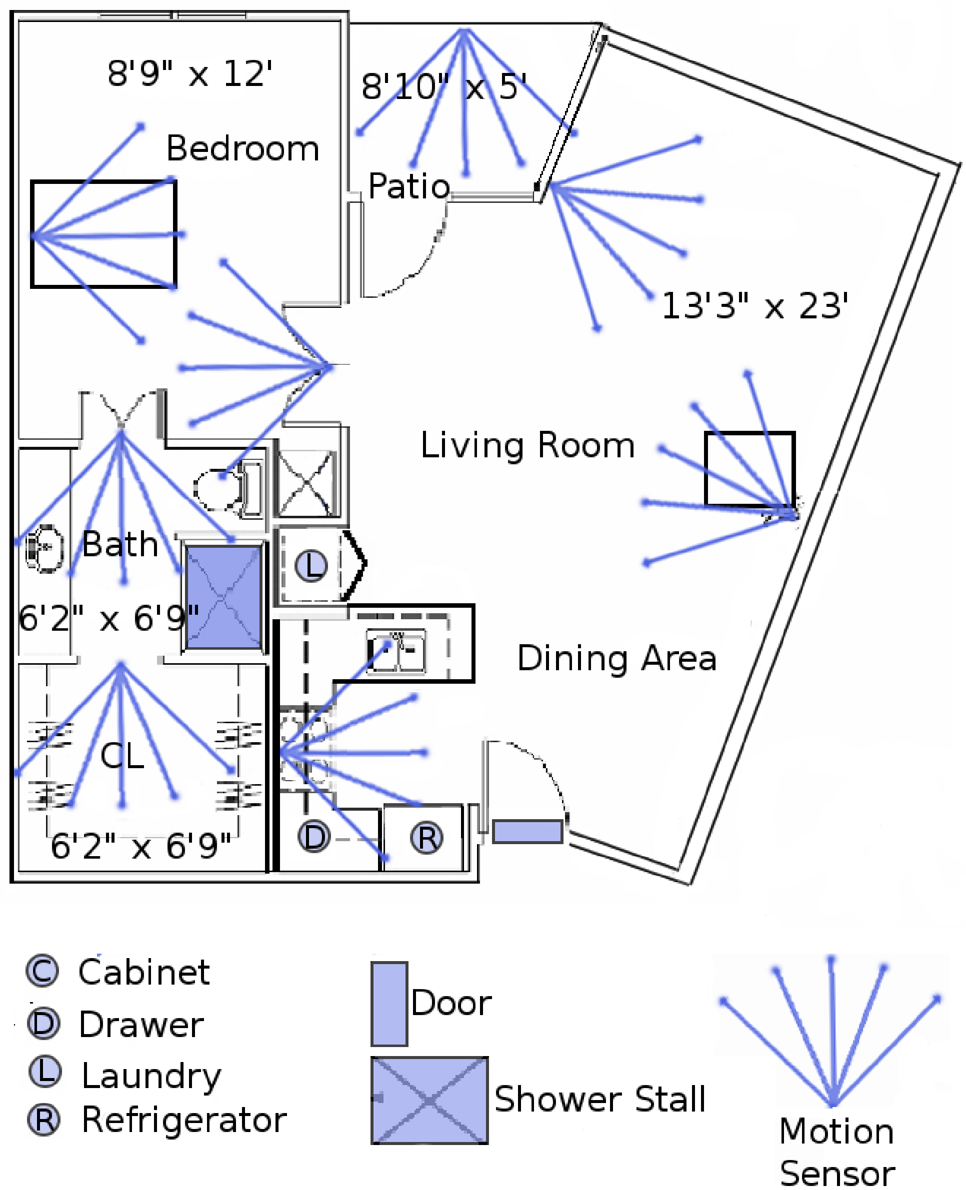

8'9" x 12'
Bedroom
8'10" x 5'
Patio
13'3" x 23'
Living Room
Bath
6'2" x 6'9"
L
Dining Area
CL
6'2" x 6'9"
D
R
C Cabinet
D Drawer
L Laundry
R Refrigerator
Door
Shower Stall
Motion Sensor

Table of User 3010 location distances

	LivingRoom	Kitchen	Bathroom	Bedroom	FrontDoor	Shower	Closet
LivingRoom	0	0	183	0	0	533	549
Kitchen	0	0	493	310	0	660	676
Bathroom	183	493	0	0	682	0	0
Bedroom	0	310	0	0	499	167	183
FrontDoor	0	0	682	499	0	849	865
Shower	533	660	0	167	849	0	0
Closet	549	676	0	183	865	0	0

UserID: 3010

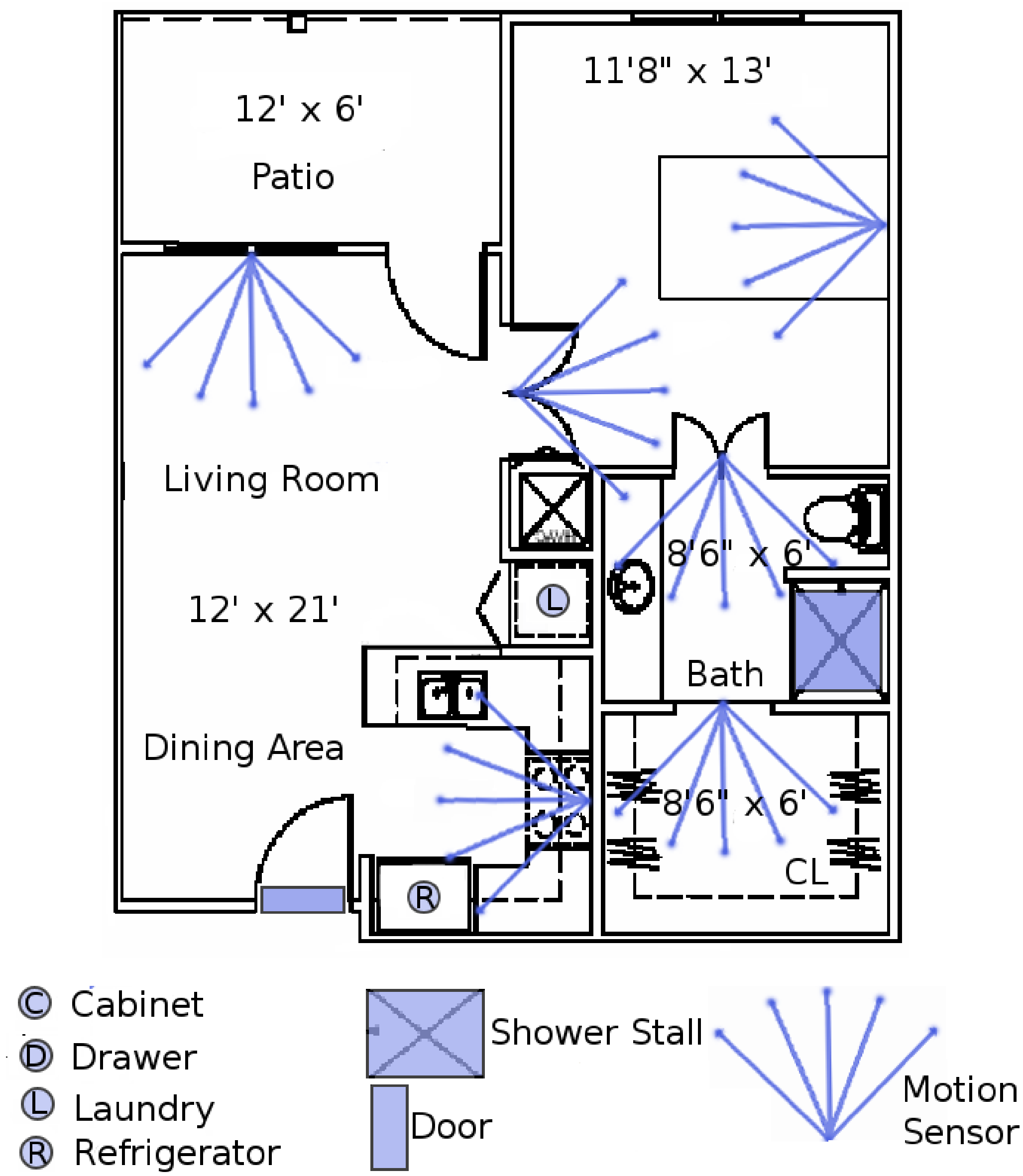
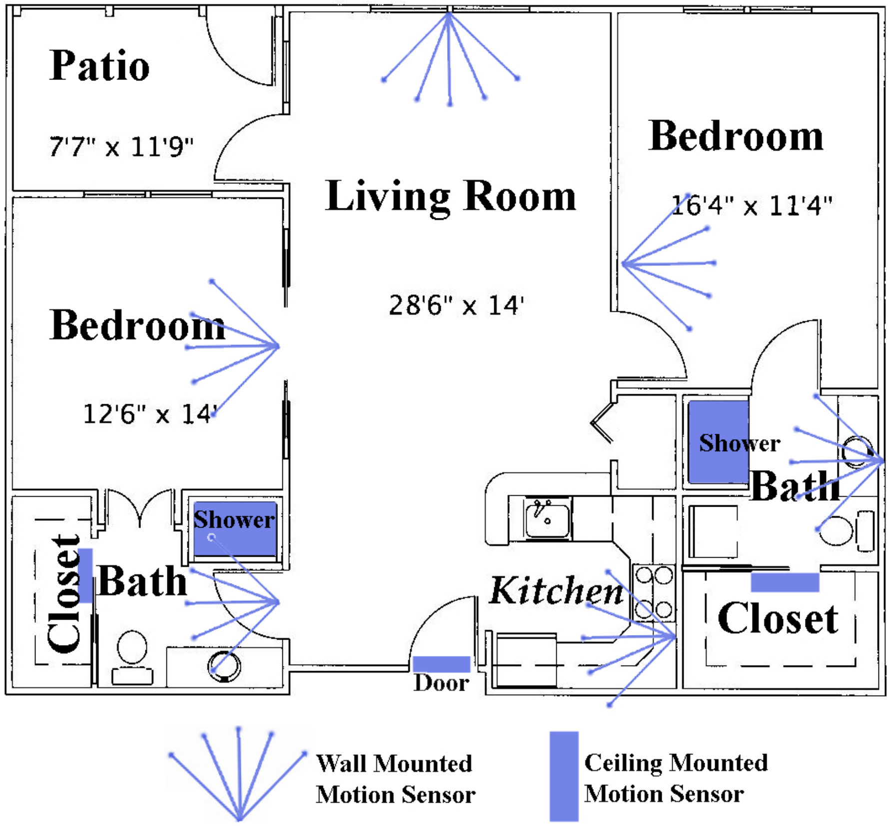

12' x 6'
Patio
11'8" x 13'
Living Room
12' x 21'
Dining Area
8'6" x 6'
Bath
8'6" x 6'
CL
L
R
Ⓒ Cabinet
Ⓓ Drawer
Ⓛ Laundry
Ⓡ Refrigerator
Shower Stall
Door
Motion Sensor

Table of User 3022 location distances (Test Apartment)

	LivingRoom	Kitchen	Bathroom	Bedroom	FrontDoor	Closet	Bathroom2	Bedroom2	Closet2
LivingRoom	0	0	0	0	0	274	0	0	437
Kitchen	0	0	0	0	0	274	162	152	523
Bathroom	0	0	0	0	163	0	686	671	1041
Bedroom	0	0	0	0	168	0	564	549	919
FrontDoor	0	0	163	168	0	396	564	549	919
Closet	274	274	0	0	396	0	879	864	1234
Bathroom2	0	162	686	564	564	879	0	0	0
Bedroom2	0	152	671	549	549	864	0	0	183
Closet2	437	523	1041	919	919	1234	0	183	0

User 3022 (Test Apartment)



Appendix C

MEMBERSHIP FUNCTIONS

Feature	Member	Mean	Std Dev
Velocity	Slow	0	25
Velocity	Normal	70	11
Velocity	Increased	115	14
Velocity	Fast	155	15
Velocity	Super Fast	200	16
Velocity Out	Very Low	0	0.04
Velocity Out	Low	0.12	0.04
Velocity Out	Medium	0.4	0.11
Velocity Out	High	0.683	0.1
Velocity Out	Very High	1	0.04
Motion	Low	0	25
Motion	Normal	150	45
Motion	Increased	300	53
Motion	High	425	30
Motion	Very High	515	15
Motion Out	Very Low	0	0.04
Motion Out	Low	0.2	0.08
Motion Out	Medium	0.5	0.1
Motion Out	High	0.8	0.08
Motion Out	Very High	1	0.04

Feature	Member	Mean	Std Dev
Duration	Very Short	0	5
Duration	Sparse	60	20
Duration	Reasonable	335	108
Duration	Good	600	191
Duration	Excellent	1800	450
Duration Out	Very Low	0	0.08
Duration Out	Low	0.25	0.11
Duration Out	Medium	0.5	0.07
Duration Out	High	0.75	0.11
Duration Out	Very High	1	0.08
MISO Output	Very Low	0	0.1062
MISO Output	Low	0.25	0.1062
MISO Output	Medium	0.5	0.1062
MISO Output	High	0.75	0.1062
MISO Output	Very High	1	0.1062