Simultaneous Tracking & Activity Recognition (STAR) Using Many Anonymous, Binary Sensors

Thesis Proposal

Daniel Wilson

The Robotics Institute
Carnegie Mellon University

June 1, 2004

Abstract

Automatic health monitoring helps enable independent living for the elderly by providing specific information to caregivers. This goal, called *aging in place*, is increasingly important as an unprecedented portion of the population enters old age. I introduce the simultaneous tracking and activity recognition (STAR) problem, whose solution provides this key information. I propose using data from many minimally invasive sensors commonly found in home security systems to provide simultaneous room-level tracking and recognition of many of the activities of daily living (ADLs). ADLs have been chosen by physicians to gauge the severity of cognitive and physical ailments. I describe a Rao-Blackwellised particle filter for room level tracking, rudimentary activity recognition, and data association, as well as a Monte Carlo EM approach for online parameter learning. I demonstrate results from experiments in an instrumented home and on simulated data. Proposed extensions improve the approach and add more complex activity recognition. We discuss how to integrate a growing vocabulary of activities into the tracker.

Contents

1 Introduction

People aged 65 and older are the fastest growing segment of the US population [6]. Over 20% of people 85 and older have a limited capacity for independent living and require continuous monitoring and daily care [24]. Additionally, about 1 in 5 Americans have some kind of disability, and 1 in 10 have a severe disability [47]. There is a growing need to develop technology to support *aging in place*, in which elders live independently and safely in their own homes without being institutionalized. Automatic health monitoring is the process of using sensors to automatically infer information about occupants. This information is primarily comprised of location and activities of daily living (ADLs). ADLs are a set of activities chosen by medical professionals to represent the status and abilities of a patient. Studies have shown that pervasive monitoring of the elderly and those with disabilities can improve the accuracy of pharmacologic interventions, track illness progression, and lower caregiver stress levels [21]. Additionally, [72] has shown that movement patterns alone are an important indicator of cognitive function, depression, and social involvement among people with Alzheimer's disease.

In this proposal we introduce the simultaneous tracking and activity recognition (STAR) problem. A solution to this problem provides the information vital for automatic health monitoring. Solving STAR includes identifying people, tracking people as they move, and knowing what activities people are engaged in. More challenging goals include recognizing when people deviate from regular patterns of behavior and predicting future behavior. Much current work, including our own, has focused on implementing machine vision and auditory systems to do these tasks [5].

This research uses a particle filter approach to exploit many *simple* sensors. Particle filters offer a sample-based approximation of probability densities that are too difficult to solve in closed form. Solving the STAR problem in a home environment with several occupants and several hundred sensors is such a problem. Particle filters are desirable because they can approximate a large range of probability distributions, unlike Kalman filters which are limited to Gaussian distributions. Using resampling focuses resources on the most promising hypotheses and the number of samples can be dynamically adjusted according to available computational resources.

We choose to use information from existing sensor infrastructures, particularly those employed by security systems, such as motion detectors, contact switches, breakbeam sensors, and pressure mats. We call a sensor anonymous and binary because it can not directly identify people and at any given time it supplies a value of one or zero. This severely limited amount of information comes with minimal privacy, monetary, and computational cost, and could be used on a large scale in homes and businesses. We show that such sensors can solve the STAR problem, telling us which rooms are occupied, counting the occupants in a room, identifying the occupants, tracking occupant movements, and recognizing occupant activities. We propose extending this work to recognize more complex activities, which will in turn improve tracking accuracy.

This thesis proposal is organized as follows: In section 2 we introduce a future scenario describing the broad goal and impact of this research. In section 3, we review existing instrumented facilities and discuss the state of the art in location estimation and activity recognition. In section 4 we introduce our approach, including our rationale for choosing simple sensors and the details of our learner. Section 4 also contains experimental results both from a real instrumented environment and from simulations. We discuss proposed work in section 5, including improvements to our tracker, advanced activity recognition, and a new method for labeling examples of activity. In section 6 and 7 we cover expected contributions and a tentative schedule of thesis activities, respectively.

2 Scenario

A man has an elderly mother living alone one hour away. Last week she knocked the phone off the hook and was unavailable for an entire day. The man walks into a hardware store and emerges with a large brown box. It contains several dozen nondescript, quarter-sized sensors that stick to any surface. Following directions, the man attaches sensors to doors, drawers, and chairs. He pulls out a CD-ROM and installs software on a personal computer and plugs a device into a USB port. The software instructs him to perform a quick walk-through of the house, touching every sensor. Later that week the man logs onto the Internet, types a password, and checks to see that his mother has eaten lunch. One week later he checks that she has been cooking and eating meals. One month later he checks whether her activity levels are steady. The system reports that activity levels are abnormally low today. He calls and finds that his mother seems to be coming down with the flu.

3 Related Work

Over the last several years much effort has been put into developing and employing a variety of sensors to solve key problems in the ubiquitous computing domain, including camera networks for people tracking [76, 13, 66], as well as cameras and microphones for activity recognition [17, 50]. Wearable sensors have been used for health monitoring [43], the facilitation of group interactions [32], and memory augmentation [64]. In this section we discuss these efforts in terms of automatic health monitoring, people tracking, and activity recognition. We pay special attention to sensor choice.

3.1 Automatic Health Monitoring

People tracking and activity recognition experiments typically occur in a laboratory setting in a corporate or educational building [38, 13, 16]. Recently, there has been an increase in the number of stand-alone instrumented home environments. The Aware Home project at Georgia Tech has built a house instrumented with ubiquitous computing technology for a variety of experiments [2]. The house has been fitted with a great variety of sensors with the goal of helping elderly adults live independently by providing memory augmentation, accident detection, and behavioral trend tracking. Researchers at MIT working on the House_*n* project have purchased a house and instrumented it with their own version of generic, simple sensors [37]. Currently, they deploy sensors for weeks at a time, collect sensor data as well as occupant labeled activity data, and then retrieve sensors for off-line activity recognition. Initial results show that for multiple instrumented houses clustered activity episodes correspond to data labeled by occupants. Researchers at the University of Florida have also instrumented a house with ultrasound localization and displays with the goal of providing timely and relevant information to residents [30]. Finally, the Neural Network House sensed appliance use and environmental changes to train neural networks to control levels of energy conservation and comfort [51]. These laboratories have explored an exciting variety of sensors to solve a variety of highly interrelated problems, mostly subsets of localization and activity recognition. Usually, these instrumented homes do not host long-term residents. Other groups, including our own, have instrumented actual health care facilities for a variety of experiments [5, 9, 49]. The instrumented home in this proposal is unique in that we use cheap, off-the-shelf sensors for simultaneous tracking and activity recognition over a long period of time. These instrumented facilities are valuable testbeds for a variety of algorithms and sensor configurations. Our own instrumented environment is intentionally constrained by the needs of elderly inhabitants. Rather than provide specific services, we intend to observe, recognize, and predict behavior.

There has been some research into using binary sensors for automatic health monitoring. For several years a group of researchers at the Tokyo Medical and Dental University have been instrumenting homes with sensors such as motion detectors and contact switches to collect data for months at a time [53, 55, 54]. Although learning algorithms have not been applied, the raw data generated during these experiments was made available to physicians who were able to pick out patterns of activity by hand. Researchers at the Medical Automation Research Center (MARC) at the University of Virginia have used an array of motion detectors and contact switches to attempt to detect activities of daily living (ADLs) [9]. They cluster sensor readings into rough groups based on room, duration, and time of day and demonstrate that many of the clusters do correspond to ADLs. Researchers have solidly identified the potential of simple sensors for automatic health monitoring.

3.2 People Tracking

People tracking is a fundamental problem in ubiquitous computing and has been approached via a variety of sensors, including cameras, laser range finders, wireless networks, RFID (Radio frequency identification) badges, and infrared or ultrasound badges [1, 3, 10, 17, 40, 51, 28, 66]. See [31] for a survey of location estimation techniques. Cost of sensors and sensor acceptance are pivotal issues, especially in the home. Many people are uncomfortable living with cameras and microphones. Laser scanning devices are anonymous, but costly and have limited range. We find that people are often unwilling, forget, change clothes too often, or are not sufficiently clothed when at home to wear a badge, beacon, set of markers, or RF tag. Elderly individuals are often very sensitive to small changes in environment [14], and a target population of institutionalized Alzheimer's patients frequently strip themselves of clothing, including any wearable sensors [15]. A distributed network of many low cost sensors has several advantages over co-located sensors on a single platform (e.g., wearable sensors or mobile robots). The total coverage may be much larger and redundancy may exist between overlapping sensors. Also, sensor networks are more robust against failure or loss of individual components. In this research it was valuable to be able to quickly replace malfunctioning sensors, although sensor network robustness issues were not explored. We have chosen to explore a set of sensors that are already present in many homes as part of security systems. These sensors are cheap, computationally inexpensive, and do not have to be continuously worn or carried. We aim for room level tracking, as our sensors do not provide the higher spatial resolution of other types of tracking systems.

Combining anonymous sensors and sensors that provide identification information for people or object tracking is an open problem. In the multi-target tracking community it is commonly known as the *data association* problem. The goal is to associate a set of current measurements with a set of existing ”tracks” or object trajectories. In AI literature the problem of *object identification* is essentially the same, to determine if a newly observed object is the same as a previously observed object. A technique introduced by [33] uses pairwise sensor-based *appearance probabilities* to match images of cars between two traffic cameras. Researchers at Berkeley noted that this technique could not scale as more sensors were added. They used a Markov chain Monte Carlo approach to make the problem tractable for accurately tracking a single car between many cameras with minimal noise [57]. The vast number of possible *assignments* and noisy real-world data has spurred a variety of probabilistic approaches. Bayesian techniques, particularly particle filters, have been introduced as effective solutions [7, 41, 35]. In a recent experiment a particle filter implementation used laser range finders and infrared badges to track six people simultaneously in an office environment for 10 minutes [27]. The range finders provided anonymous, high granularity coordinates while the badge system identified occupants. We also use a particle filter approach to solve the data association problem, however, we use ID sensors only at entrances and exits and rely upon individual motion and activity models to resolve ambiguity within the environment. Data collected over the long-term provides an ever-improving model of the unique patterns of each occupant. We explore the ability of these models to identify occupants in lieu of additional ID-sensors.

3.3 Activity Recognition

An impressive amount of research falls under the umbrella of *activity recognition*. In particular, researchers have used cameras to detect a variety of activities, including sign language recognition [67], human gait recognition [48], sitting, standing and walking behaviors via wearable cameras [43], and recognizing American football [36] and basketball [39] plays from video. A variety of other sensors have been applied as well, including GPS readings to infer walking, driving and bus riding behaviors [59], laser range finders to learn motion paths in a home [10], audio to recognize conversational interactions over cell phones [75], and combinations of audio and video to recognize behavior in an office environment [56], group meeting interactions [69], and interactions between individuals [17]. Recently, researchers at Intel Research have used radio frequency identification tags to recognize several ADLs [26]. We are unaware of any research that has attempted to use machine learning techniques to automatically recognize multiple ADLs using the sort of binary sensors common to home security systems, with or without simultaneous people tracking to improve recognition results.

A growing variety of Bayesian techniques have been used for activity recognition [56, 58]. For a survey of Bayesian techniques applied to activity recognition see [27]. In healthcare literature Dynamic Bayes nets have been used extensively for *execution monitoring*, a more intensive form of activity recognition in which the goal is to determine whether a person is following a plan appropriately [4, 68, 19]. Execution monitoring calls for recognition of specific parts of an activity as well as possible paths of progression through the plan. These approaches vary depending on the characteristics of the activity to be recognized. We plan to compare several modified versions of these Bayesian activity recognition techniques. Our novel contribution arises from the interplay of tracking and activity recognition in an integrated system that uses only information from many anonymous, binary sensors.

4 Current & Completed Work

4.1 Anonymous, Binary Sensors

Automatic health monitoring is predominantly composed of **location** and **activity** information. Below is a list of exactly what we wish to automatically recognize.

4.1.1 Sensing Goals for this Thesis

- **Presence.** Determine how many and which people are in the environment.
- **Individual Identification.** Determine the identity of each person.
- **Room-Level Tracking.** Determine the location of each person.
- **Locomotion.** Recognize whether an occupant is moving or sitting still (e.g., walking, wheelchair use, etc.,).
- **Activities of Daily Living.** Recognize eating, bathing, dressing, and toileting.
- **Extended Activities of Daily Living.** Recognize washing dishes, preparing meals, grooming in front of sink, watching tv, getting the mail, and sleeping.

4.1.2 Promising Future Sensing Goals

- **Activity Patterns.** Detect and form models of recurrent activities. These models can be used to recognize when behavior deviates from the usual routines.
- **Predicted Behaviors.** Use advanced, occupant-specific models of activity and location to predict future behavior, and possibly future interactions between groups of occupants. Depending on the granularity, these predictions could be for minutes, hours, or days ahead. Depending on capabilities, predicted behavior could be encouraged or inhibited.
- **Extrapolated Behaviors.** Use estimates of location and activity as features in even higher-level learners, trained to recognize ideals such as overall physical fitness, depression, or social involvement.

4.1.3 Sensor Constraints & Issues

Automatic health monitoring necessarily occurs in a home environment. Instrumentation of a home raises two critical issues, (1) choice of sensors, and (2) sensor placement. Ideally, the sensors we choose should offer solutions to the following issues :

- **Invisible or Familiar.** Sensors and monitoring systems should be invisible. Alternately, devices should fit into familiar forms with familiar interfaces. They should remain unobtrusive and require no change in routine. Most people are not interested in dealing with new devices without a large perceived benefit.
- **Private.** Sensor data alone should not provide private information, especially identity. Alternately, sensitive information should never be stored unless privacy can be guaranteed. It is equally important that sensors not be *perceived* as invasive (as cameras and microphones frequently are).
- **Economical.** Sensors should be inexpensive and available off-the-shelf.
- **Computation.** Processing sensor data should require minimal computational resources, ideally requiring nothing more than a contemporary desktop computer.
- **Installation.** Sensors should be easy to install. Wireless sensors can be mounted to a surface, while wired sensors may require running cable to a central location. We envision an off-the-shelf system installed and configured by a consumer.
- **Maintenance.** Sensors should be easy to replace and maintain. Sensors will be neglected and should be robust to damage. Sensors which self-report their status are ideal. Alternately, simple algorithms could determine sensor status.
- **Power.** Sensors should require no external power or run as long as possible on batteries. As a last resort the device may need to be plugged in or powered by low voltage wiring.

Sensors that are *anonymous* and *binary* satisfy many of these properties. Anonymous sensors satisfy privacy constraints because they do not directly identify the person being sensed. Binary sensors, which simply report a value of zero or one at each time step, satisfy computational constraints. Many anonymous, binary sensors exist in home security systems. This helps with perceived privacy issues because they have become a familiar sight in public buildings as well as in private homes. These sensors are valuable to the home security industry because they are economical, easy to install, require minimal maintenance and supervision, and are robust to damage. We choose them for the same reasons, and because they *already exist* in many of our target environments. We typically use a denser installation of sensors than in a home security system, however.

4.1.4 Sensor Properties

We introduce several sensor properties in order to choose a subset that simplifies our problem as much as possible.

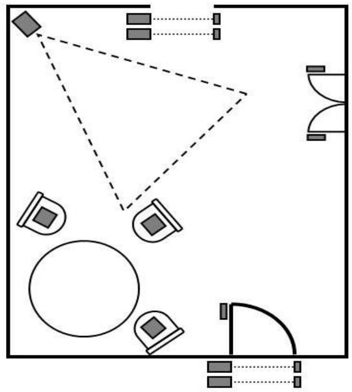

Figure 1: Overview of typically instrumented room. Grey squares represent contact switches, motion detectors, pressure mats, and break beam sensors.

- **Coverage.** There is a *coverage area* associated with each sensor. This area can convey important information about occupant location. For instance, a motion detector has a coverage of around the size of an average room. Break beam sensors cover a linear area. Contact switches and pressure mats must be directly manipulated through interaction with a physical object.
- **Trigger.** Different sensors are *triggered* in different ways. We use this property to determine whether an occupant is moving or not, by drawing a line between sensors that are triggered *actively* and those triggered *passively*. We define ”active” as a person who is moving.
- **Timing.** Different sensors require different types of contact in order to trigger. Again, the type of contact can connote the location or activity of the occupant. Motion detectors are triggered *occasionally* (every few seconds) as occupants move nearby. Contact switches *change state* as they are opened and closed. Break-beam sensors and pressure mats trigger *continuously* while the beam is interrupted or sufficient weight is applied.

4.1.5 Sensor Choice and Placement

There are many sensors to choose from. In fact, any sensor can be anonymous and binary with the proper thresholds. We chose four commonly available anonymous, binary sensors : motion detectors, contact switches, break-beam sensors, and pressure mats. These four sensors have different properties which combine to reveal a surprising amount of information.

In our work, motion detectors are placed near the ceiling in order to maximize room coverage. Thus, a motion detector is most likely to trigger due to a person walking around a room, or other gross movements. Negative information indicates that when a motion detector does not fire with an occupant in the room that the occupant is probably not moving. Contact switches are placed on doors and drawers of all types, including cabinets and refrigerators. Contact switches placed in these areas will be manipulated only by active occupants. Pressure mats are placed under couches, chairs and rugs. Break-beam sensors are triggered by occupants walking through the beam. With two beams we can infer direction. See Figure 1. for an overview of a typically instrumented room. Although anonymous sensors can be used to maintain identity and location of occupants, it simplifies the problem to have a small number of identity sensors. The natural time and place to obtain this information is when an occupant enters or leaves the environment. We instrumented a familiar interface by using radio frequency identification (RFID) tags in place of house keys. An RFID reader is placed in the doorway. When a ”key” or RFID tag is waved near the reader, the door is automatically unlocked for a few seconds. Afterwards the door locks itself. Currently, we use the following sensors:

- **Motion detectors.** We use wireless X10 Hawkeye motion detectors. They provide a binary indication of heat and movement (e.g., human presence) in an area. After each reading these sensors pause for eight seconds before becoming active again. X10 is a communications protocol that allows devices to communicate via existing electrical wiring. Upon sensing motion a radio signal is sent to a receiver, which transmits a unique signal over the powerline. This signal is collected by a CM11A device attached to a computer. The detectors are wireless, pet-resistant, require heat and movement to trigger, and run on battery power for over one year.

- **Contact switches.** These inexpensive magnetic contact switches indicate a closed or open status.

- **Pressure mats.** These sensors detect presence on chairs and couches. The pressure mats are made of two metal screens separated by a piece of foam with holes. The weight necessary for contact depends on the size and number of holes cut into the foam layer.

- **Break-beam sensors.** We use these sensors in groups of two to determine when an occupant passes through a doorway and in what direction. They work by generating a beam across a space and monitoring when it is reflected back. While the beam is interrupted the sensor changes state.

- **Radio Frequency Identification (RFID).** We use low frequency RFID to identify occupants entering and leaving the environment. The system sends a modulated RF signal to an antenna, which amplifies the signal, creating a small field near the front door. When the credit card sized transponder or ’tag’ is in the field, an integrated circuit detects the signal and uses its energy to send a unique identification signal. This signal is decoded and sent to a computer via an RS-232 interface. The entire process takes less than 100ms and multiple tags can be read simultaneously. Each occupant is given a unique tag; upon recognition the tag will automatically unlock the door, as well as identify the occupant entering or leaving the environment. This interface faces challenges when occupants forget badges or when guests visit.

All sensors interface with an Intel Pentium IV 1.8 GHz desktop computer with 512MB ram. An expanded parallel port monitors contact switches and pressure mats, a serial interface attached to a CM11A device monitors motion detector activity, and a serial interface connects to the RFID reader.

4.1.6 Future Instrumentation Possibilities

As we continue to instrument environments and experiment we will inevitably find useful new sensors and configurations. For example, we are currently testing the placement of float sensors in toilet tanks to determine when flushes occur, moisture detectors in sink drains to detect water flow, and AC current detectors for appliance usage. These new sensors fit easily into our current framework because they are anonymous and binary. Rich sensors such as cameras can be easily integrated by extracting anonymous, binary features or by discretizing information into a finite number of ”buckets”. However, integrating continuous valued features will require significant modification to algorithms optimized for a discrete state space.

In the future, specialized sensors may solve certain activity recognition problems outright. For instance, a ”mailbox sensor” might indicate when an occupant fetches the mail. These sort of black box sensors could fit well into our system by directly providing information to the location tracker and the rest of the activity recognizers. The more activities that can be recognized instantly by special sensors, the better the results for both location estimation and more traditional activity recognition algorithms.

4.1.7 A Note on Privacy

This research does not directly address privacy issues except inasmuch as we choose to use sensors that are inherently private, simply because they are anonymous and binary. Aside from the RFID antenna (which requires a tag), none of the individual sensors in this research can be used to identify a person. The locations and activities that are collected by applying machine learning algorithms to the entire collection of sensor information is obviously of a private nature. The dissemination of and access privileges to this information (whether to family, physicians, or to the general public) will depend on the services provided using this system, and are outside of the scope of this document.

4.2 Simultaneous Tracking & Activity Recognition (STAR)

Our solution to the simultaneous tracking and activity recognition problem aims to provide an account of the room and activity of every occupant in the environment. Occupants are identified by an ID sensor at the front door, but afterwards identity, location, and activity must be inferred from a continuously updated set of binary, anonymous observations. In our preliminary work we attempt to provide room-level tracking and recognition of locomotion. We do not limit the number of occupants, sensors, or rooms although some simplifying assumptions are made.

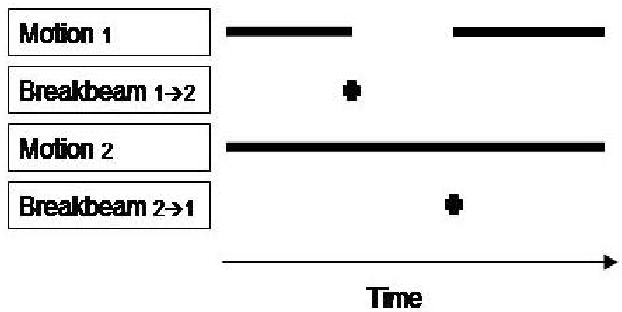

Figure 2: Sensor evidence of the movement of two occupants through two rooms.

There are two main problems when solving STAR for multiple people, (1) what is the state of each person and, (2) which person is which? In the first problem, observations are used to update the state of each occupant. In the second problem, identity of the occupants is estimated and anonymous observations are assigned to the occupants most likely to have generated them. Uncertainty occurs when several occupants trigger the same set of anonymous sensors. The tracker does not know which occupant triggered which sensor (i.e., which data to associate with which occupant). In Figure 2 the occupants begin in separate rooms. Eventually, the occupant from room 1 enters room 2. Without further information the identity of each occupant is potentially confounded after someone exits the shared room.

There are several ways to simplify the problem. First we could **increase the number of ID sensors.** This simple approach solves the problem by using sensors that identify occupants outright. Unfortunately, ID sensors are expensive, have significant infrastructure requirements, and/or must be worn or carried by the occupant (see section 3.1.2). Second, we could **increase the sensor granularity.** The more sensors there are, the smaller the probability that multiple occupants will share the same anonymous sensors. For example, Thrun et. al., tracked people with high granularity laser range finders and sensor collision did not occur unless people occluded each other very closely [70]. This problem has extra significance for room level tracking with low granularity sensors. We use clever sensor choice and placement to maximize granularity (see section 3.1.4). For example, noting which contact switches are out of reach of pressure mats potentially separates two occupants when one is seated and the other opens a drawer. Third, we could **learn individual movement and activity patterns.** Over time, statistical models can represent particular habits of select individuals. Individual motion models can help the tracker recover from ambiguity as occupants follow their regular habits (e.g., sitting in favorite chairs or sleeping in their own beds). This could resolve the ambiguity in Figure 2. If one occupant rarely enters room 1 then the other occupant is more likely to have left the shared room. Recognizing activities provides a novel method of recovery. Preliminary experiments model occupant loco-

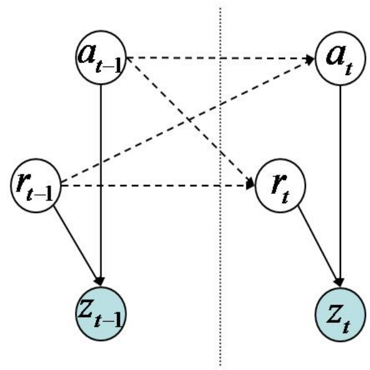

Figure 3: A dynamic Bayes net describing tracking and activity recognition. Arcs indicate causal influences, with dotted arcs representing causality through time. Circles represent variables. Shaded variables are directly observable, the rest are hidden.

motion. In this case, moving occupants can be separated from still occupants, even if they are in the same room. For example, if one occupant is not moving then the other occupant is more likely to be exiting room 2.

4.2.1 Bayes Filter Approach

First, we address the question of how to update occupant state given sensor measurements. Bayes' filters offer a well-known way to estimate the state of a dynamic system from noisy sensor data in real world domains [23]. The *state* represents occupant location and activity, while sensors provide information about the state. A probability distribution, called the *belief*, describes the probability that the occupant is in each state $p(X_t = x_t)$. A Bayes filter updates the belief at each time step, conditioned on the data. Modeling systems over time is made tractable by the Markov assumption that the current state depends only on the previous state.

We estimate the state $x_t = \{x_{1t}, x_{2t}, ..., x_{Mt}\}$ of M occupants at time t using the sensor measurements collected so far, $z_{1:t}$. At each time step we receive the status of many binary sensors. The measurement $z_t = \{e_{1t}, e_{2t}, ..., e_{Et}\}$ is a string of E binary digits representing which sensors have triggered during time step t. We choose a non-metric (i.e., discrete) state representation due to low sensor granularity and limited computational resources. Rooms provide a natural and intuitive discretization of possible occupant locations. Moving or not moving are the initial categories of possible activity. The update equation is analogous to the forward portion of the forward-backward algorithm used in hidden Markov models (HMMs). See [61] for a detailed description of how HMMs work.

$$p(X_t = x_t|z_{1:t}) \propto p(z_t|X_t = x_t) \sum_{x' \in X} p(X_t = x_t|X_{t-1} = x')p(X_{t-1} = x'|z_{1:t-1}). \tag{1}$$

The *sensor model* $p(z_t|X_t = x_t)$ represents the likelihood of measurement z_t occurring from state x_t. The *motion model* $p(X_t = x_t|X_{t-1} = x')$ predicts the likelihood of transition from the state x' to the current state x_t. How these models are learned is discussed in section 4.4.

The graphical model in Figure 3 represents the dependencies we are about to describe. The state space $x \in X$ for occupant m is the range of possible locations and activities, $x_{mt} = \{r_{mt}, a_{mt}\}$, where $r \in R$ denotes which **r**oom the occupant is in, and $a \in \{\text{moving,not moving}\}$ denotes occupant **a**ctivity. The raw sensor values are the only given information; the rest must be inferred. Each observation is composed of a collection of *events* and appear $z_t = \{e_{1t}, e_{2t}, ..., e_{Et}\}$. Event generation is straightforward. When a motion detector triggers a movement event is generated. Upon a state change a contact switch evokes a manipulation event. While a pressure mat is depressed a sit event is generated. When a pair of break beam sensors are triggered, depending upon the order, an enter event is generated for the appropriate room.

Tracking multiple people causes the state to have quite large dimensionality, making model learning intractable. Currently, a simplifying independence assumption between m occupants means that the update equation is factored as:

$$p(X_t = x_t|X_{t-1} = x') = \prod_{m \in M} p(X_{mt} = x_{mt}|X_{m,t-1} = x'_m). \tag{2}$$

In section 4.3 we propose partially relaxing this assumption through the use of two models, one for occupants that are alone and another for multiple occupants. This abstraction avoids the exponential blow up resulting from joint models of combinations of specific individuals. A similar approach has been applied successfully to tracking multiple interacting ants in [42].

Equation 1 describes the Bayes filter update using all observations up to the current time step $z_{1:t}$. Higher accuracy is usually obtained off-line by using past and future information at each time step. This is commonly known as *smoothing*. Smoothing provides higher accuracy for off-line purposes, such as a daily summary of movement activity [74]. The update equation is analogous to the *backward* step of the forward-backward algorithm commonly used in HMMs.

$$p(X_t = x_t|z_{t+1:T}) \propto \sum_{x \in X} p(X_{t+1} = x|z_{t+2:T})p(X_{t+1} = x|X_t = x_t)p(z_{t+1}|X_{t+1} = x). \tag{3}$$

4.2.2 Classical Data Association Methods

The above approach works well for tracking a single occupant in a noisy domain (the Bayes filter is named for its ability to *filter* spurious noise). This approach fails to track multiple occupants because other occupants do not behave like noise processes, and the tracker is confused by constantly conflicting sensor measurements. We need some manner of determining which occupant generated what observation. This is the data association problem, and in our domain it can become severe. For t seconds and m occupants each association has $m!^t$ possibilities. It is reasonable to imagine several hundred cheap sensors monitoring a half dozen occupants, resulting in too many data assignments to enumerate.

There are several classical data association methods, for a survey see the paper [63] or the book [8] Probably the simplest approach is called the *nearest neighbor standard filter* (NNSF). In NNSF only the closest observations to any given state are used to perform the measurement update step. This method has a hard time recovering lost targets because unlikely observations are ignored. When multiple observations are close enough, they are all equally likely. A more accurate method is called the *probability data association filter* (PDAF). PDAF uses the probability of an observation from a target versus from *clutter* to assign weighted measurements. For multiple targets multiple independent PDAs are used. This approach fails for the same reason independent Bayes filters do – occupants do not behave like noise. This problem is dealt with by the *joint probability data association filter* (JPDAF), which finds the joint probability of all possible assignments for the current time step. JPDAF then updates the state by a sum over all the association hypotheses weighted by the probabilities from the likelihood. These methods treat each time step independently. A more general method, called *multi hypothesis tracking* (MHT), calculates every possible association hypothesis over time as well. PDAF, JPDAF, and MHT approaches require exhaustive enumeration of every possible association, which can quickly become intractable. To cope, it is common to define a *gate* around each target and only consider sensor readings that occur within the gate. MHT can reduce computational time by creating a tree of possible hypothesis and then pruning unlikely associations. For the STAR problem, these approaches are only feasible for room level tracking with few sensors and few occupants. They become intractable with many sensors and people. Recently, particle filters have been applied successfully to the data association problem [11]. In the next section we implement a particle filter approach to the data association problem.

4.2.3 Particle Filter Implementation

At each time step we wish to find the best assignment of sensors to occupants and to use this assignment to update the state of each occupant. Assignments between sensor measurements and occupants are not given. Therefore, we must now estimate the posterior distribution over both occupant state and sensor assignments.

We let θ_t represent a sensor assignment matrix such that $\theta_t(i, j)$ is 1 if event e_{it} belongs to occupant j and 0 otherwise. See Figure 4 for the updated graphical model. We accommodate our expanded posterior efficiently by using a Rao-Blackwellised particle filter (RBPF) [23]. We expand the posterior of Equation 1 to incorporate data association. By the chain rule of probability,

$$p(X_{1:t}, \theta_{1:t}|z_{1:t}) = p(X_{1:t}|\theta_{1:t}, z_{1:t})p(\theta_{1:t}|z_{1:t}). \tag{4}$$

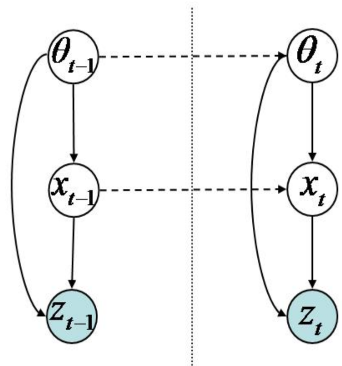

Figure 4: A dynamic Bayes net describing tracking and activity recognition (combined into the state x) as well as data associations θ.

The key idea is to update the **state** $p(X_t = x|\theta_{1:t}, z_{1:t})$ analytically using the Bayes filter update already described, and to use a particle filter to generate a sample-based approximation of **assignments** $p(\theta_{1:t}|z_{1:t})$. This streamlines our approach by sampling only from the intractable number of possible sensor assignments and solving exactly for our (relatively) small number of possible state configurations.

The desired posterior from Equation 4 is represented by a set of N weighted particles. Each particle j maintains the current state of all occupants via a bank of M Bayes filters, as well as the sensor assignments and the weight of the particle.

$$s_t^j = \{x_t^{(j)}, \theta_{1:t}^{(j)}, w_t^{(j)}\}. \tag{5}$$

Note that for filtering purposes we can store only the latest association $\theta_t^{(j)}$. In our case, x_t^j is a distribution over all possible states of all occupants. The θ_t^j are updated via particle filtering, and the x_t^j are updated exactly using the Bayes filter update. The marginal distribution of the assignment (from Equation 4) is therefore approximated via a collection of N weighted particles,

$$p(\theta_{1:t}|z_{1:t}) \approx \sum_{j=1}^{N} w_t^{(j)} \delta(\theta_{1:t}^{(j)}, \theta_{1:t}). \tag{6}$$

where $w_t^{(j)}$ is the *importance weight* of particle j, and $\delta(x, y) = 1$ if $x = y$ and 0 otherwise.

Given the sample-based representation of assignments from Equation 6, the marginal of the state node is,

$$
\begin{aligned}
p(X_t|z_{1:t}) &= \sum_{\theta_{1:t}} p(X_t|\theta_{1:t}, z_{1:t})p(\theta_{1:t}|z_{1:t}) &(7)\\
&\approx \sum_{\theta_{1:t}} p(X_t|\theta_{1:t}, z_{1:t}) \sum_{j=1}^{N} w_t^{(j)} \delta(\theta_{1:t}^{(j)}, \theta_{1:t}) &(8)\\
&= \sum_{j=1}^{N} w_t^{(j)} p(X_t|\theta_{1:t}^{(j)}, z_{1:t}). &(9)
\end{aligned}
$$

Given a sampled data association $\theta_{1:t}^{(j)}$ and an observation z_t, it is straightforward to update the belief $p(X_t = x|z_{1:t}, \theta_{1:t})$ exactly according to a slightly modified version of the Bayes filter from Equation 1. First, we show the *predictive* distribution, where information up to time step $t-1$ is used to predict the next state for particle j.

$$
p(X_t = x|z_{1:t-1}, \theta_{1:t-1}^{(j)}) = \sum_{x'} p(X_t = x|X_{t-1} = x')p(X_{t-1} = x'|z_{1:t-1}, \theta_{1:t-1}^{(j)}). \tag{10}
$$

We derive the full update equation given information up to time t according to Bayes rule.

$$
\begin{aligned}
p(X_t = x|z_{1:t}, \theta_{1:t}^{(j)}) &= \frac{p(z_t|X_t = x, \theta_t^{(j)})p(X_t = x|z_{1:t-1}, \theta_{1:t-1}^{(j)})}{\sum_x p(z_t|X_t = x, \theta_t^{(j)})p(X_t = x|z_{1:t-1}, \theta_{1:t}^{(j)})} &(11)\\
&\propto p(z_t|X_t = x, \theta_t^{(j)})p(X_t = x|z_{1:t-1}, \theta_{1:t-1}^{(j)}). &(12)
\end{aligned}
$$

Given these definitions we now discuss the overall RBPF approach.

The following sampling scheme, called *sequential importance sampling with re-sampling* is repeated N times at each time step to generate a full sample set S_t (composed of N samples $s_t^{(j)}$ where $j = 1...N$) [23]. In each iteration, a sample is chosen from the previous sample set, a new data association is proposed and used to update the state, and the sample is re-weighted based upon how likely it is. After N iterations, the importance weights are normalized. This procedure provides a sample-based approximation of the Bayes filter update.

Initialization. Occupant location and identity are gathered by RFID upon entry and exit. During these time steps id sensor measurements are assigned automatically. In other words, the assignment matrix θ_t is partially constructed before sampling begins. If this is the first time step we use this information to initialize our first sample set S_0.

Re-sampling. Using the sample set from the previous time step S_{t-1}, draw with replacement a random sample $s_{t-1}^{(j)}$ according to the discrete distribution of the importance weights $w_{t-1}^{(j)}$.

Sampling. Sample a possible sensor assignment matrix $\theta_t^{(j)}$. There are several ways to propose this assignment matrix, ranging from assigning measurements uniformly to assigning them based on occupant location and activity. See the next section for details.

Exact Update. Use the association $\theta_t^{(j)}$ to update the state of each occupant in sample j via Equation 12.

Importance Sampling. Weight the new sample $s_t^{(j)}$ proportional to the likelihood of the resulting posteriors of the state of each occupant. This is equal to the denominator of Equation 11,

$$w_t^{(j)} = \eta \sum_x p(z_t | X_t = x, \theta_t^{(j)}) p(X_t = x | z_{1:t-1}, \theta_{1:t}^{(j)}), \tag{13}$$

where η is a normalizing constant so that the weights sum to one.

4.3 Improvements

4.3.1 Data Association

During the sampling step a possible data association is proposed for the new sample. Choosing an impossible association will cause that particle to have a zero weight and wastes computational time. A more efficient particle filter will propose data associations in areas of high likelihood. The better the proposals, the fewer particles necessary.

One could simply assign sensor readings uniformly between occupants, regardless of occupant state. This approach is inefficient because it will propose many unlikely or impossible associations (e.g., one occupant given sensor readings from different rooms). A quick improvement is to use *gating* to eliminate impossible associations.

A gated uniform method is still inefficient because it ignores the current state of each occupant. In reality, sensors are intimately tied to rooms and activities. Occupants that are in the same room as a sensor are more likely to have triggered it. Occupants engaged in certain activities are more likely to trigger associated sensors. A simple heuristic takes advantage of these properties. We currently assign measurements based on the posterior $p(\theta_t | x_{t-1}^{(j)})$. The proposed θ_t is constructed when each measurement is assigned independently based on the probability of being set off by each occupant $p(e_{it} | x_t) \forall i$. This method tends to choose likely assignments and avoids impossible assignments, but is not guaranteed to approximate the true distribution $p(\theta_t | z_{1:t})$.

We plan to implement a Markov Chain Monte Carlo (MCMC) technique called the Metropolis-Hastings algorithm that can generate samples proportional to the true distribution. This algorithm approximates the posterior by simulating a Markov chain with the correct stationary distribution, without building a full transition probability matrix. Specifically, a sequence of candidate assignments a' are proposed, given the current assignment a, according to some *proposal distribution* $Q(a'; a)$. A heuristic is used to either accept or reject the proposed assignment. Successive assignments are related to each other, and after some unknown time the Markov chain will converge to a sequence of (non-iid) samples from the true posterior distribution. The process is as follows :

1. Start with a valid initial assignment a.

2. Propose a new assignment a' using the *proposal density* $Q(a'; a)$.

3. Calculate the *acceptance ratio*

$$\alpha = \frac{p(a')}{p(a)} \frac{Q(a; a')}{Q(a'; a)} \tag{14}$$

4. If $\alpha >= 1$ then accept the proposed assignment, otherwise keep the old one.

In our case, the distribution $p(a)$ is the true distribution, conditioned on specific assignments $p(\theta_t|\theta^{(j)}_{1:t-1}, z_{1:t})$. The choice of proposal density $Q(a'; a)$ bears careful consideration. The efficiency of this approach is intimately tied to choosing a proposal distribution that will converge to the best answer the fastest. We intend to adopt and improve a a technique called *smart chain flipping*. Smart chain flipping was introduced by Dellaert et. al. [22], for data association in the structure from motion problem and has since been applied successfully to multiple target tracking [27]. It explores possible assignments by assigning sensors individually and then permuting associations at each step.

4.3.2 Adaptation

A conceptually simple modification greatly increases the efficiency of our particle filter. The inefficiency arises during the re-sampling step, when samples from the previous sample set are drawn *blindly* without considering the most recent observation. The particle filter is less wasteful with samples if the proposal distribution relies not only on the motion model, but also on the most recent measurement [60]. This improvement is known as *assignment lookahead* or *adaptation*.

The goal is to update the sample weights of the previous timestep by the sample's ability to predict the observation of the current timestep. This update is applied to every particle $j = 1...N$ as soon as a new measurement z_t is received, and before the resampling step. For sample j at time step $t-1$ we wish to update by,

$$\varpi^{(j)}_{t-1} \propto w^{(j)}_{t-1} p(z_t|x^{(j)}_{t-1}) \tag{15}$$

Equation 15 is intractable because $p(z_t|x^{(j)}_{t-1})$ requires a summation over every possible data association. There are too many data associations to enumerate. We can approximate by using the previously described MCMC technique to approximate $p(z_t|x^{(j)}_{t-1})$ with a set of R samples $s^{(j)}_{t-1}$ where $j = 1...R$ and the sampled data associations are denoted $\theta^{(j)}_{rt}$. We use these sample associations to re-weight each member of the previous sample set.

$$\varpi_{t-1}^{j} \propto \frac{w_{t-1}^{j}}{R} \sum_{r=1..R} p(z_t|\theta_{rt}^{(j)}, x_{t-1}^{(j)}) p(\theta_t^{(j)}|x_{t-1}^{(j)}) \tag{16}$$

4.3.3 Relaxing the Independence Assumption

Earlier, we introduced an independence assumption between occupants in order to make model learning tractable and to avoid having to learn a joint model for every combination of individuals. We plan to partially relax this assumption by expanding our motion and sensor models to situations when occupants are home alone, or home together. This division is attractive because the RFID sensor reports whether occupants are home alone or not. Thus, the model for occupants that are alone is learned by simple counting. The model for occupants that are home together will capture tendencies to avoid or gravitate to certain locations and activities. For example, this could model the tendency of people to eat together but to toilet individually. The home together model is learned online via the MCEM algorithm described in the next section.

4.4 Parameter Learning

Modeling the behavior of individual occupants can increase tracking and activity recognition accuracy and make data association more efficient. Motion models describe individual tendencies to transition between rooms and activities. Sensor models describe individual tendencies to set off specific sensors (e.g., short occupants may use high cabinet doors less often). Accurate sensor models improve the efficiency of sampling data associations. Models can be initialized generically for unknown occupants.

Motion model. We wish to learn individual parameters for the motion model.

$$\begin{aligned} p(X_t = x_t|X_{t-1} = x_{t-1}) &= p(a_t, r_t|a_{t-1}, r_{t-1}) && (17) \\ &= p(a_t|a_{t-1}, r_{t-1}) p(r_t|r_{t-1}, a_{t-1}). && (18) \end{aligned}$$

- $p(r_t|r_{t-1}, a_{t-1})$ is the probability of transition to a room given the previous room and whether the occupant was moving or not. Transition probabilities between contiguous rooms are initialized uniformly for moving occupants and set to small values for non-moving occupants.

- $p(a_t|a_{t-1}, r_{t-1})$ models the probability of whether or not the occupant is moving given the previous room and whether the occupant was moving during the last time step. This is initialized so that it is more likely for moving occupants to continue to move and non-moving ones to continue not to.

Sensor model. Individual sensor readings, called ***events***, are independent. For occupant m the sensor model can be rewritten:

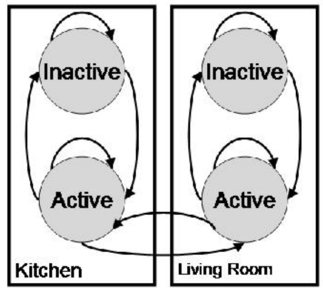

Figure 5: Active / inactive state representation.

$$p(z_t|X_t = x_t, \theta_t^{(j)}) = \prod_{m \in M} p(z_t|X_{mt} = x_{mt}, \theta_t^{(j)}) = \prod_{m \in M} \prod_i p(e_{it}|X_{mt} = x_{mt}, \theta_t^{(j)}). \tag{19}$$

This models the probability of observing each sensor measurement given the location of the occupant and whether or not the occupant is moving. This sensor model is initialized by assigning small probability to sensor readings occurring outside their designated room. Activity information contributes to the probability. For instance, motion detector readings are more likely from active occupants than from inactive occupants. Contact switches and break beam sensor readings are likely for active occupants but not inactive ones. Pressure mat readings are likely from inactive occupants and not active ones.

4.4.1 Monte Carlo EM

This system uses a non-metric, room based location representation and a discrete set of mutually exclusive activities (either moving or not moving). The result is a relatively small number of discrete states, even when confounded with additional activities. This simplicity helps make *unsupervised* learning of model parameters possible. It also invites an intuitive understanding of how transitions occur between rooms and activities (see Figure 6).

Training model parameters is simple when we know the true state of each occupant. In early experiments we trained parameters on data generated by occupants that were home alone. While a person is home alone we can assume that any sensor readings are generated by that person or a noise process. Parameter learning can be performed with simple counting. This method ignores a significant amount of training data because occupants are often home together. It also fails to learn the difference between how people behave alone versus in the presence of others.

Model Initialization:

1. Initialize model parameters with generic values.

E-step:

1. Generate N samples uniformly.
2. Forward filtering : for $t = 2...T$
(a) Generate N samples using the samples from the previous time step.
(b) Reweight each sample based on current observation z_t.
(c) Multiply or discard samples based on their weights.
(d) For each occupant m count and store $\alpha_t^m(r_t, a_t)$
3. Generate N samples uniformly.
4. Backward filtering : for $t = T..1$
(a) Calculate backward parameters $p(r_{t-1}|r_t, a_t), p(a_{t-1}|a_t, r_t)$
(b) Generate N samples using the samples from existing samples using backward parameter estimation.
(c) Reweight each sample based on current observation z_t.
(d) Multiply or discard samples based on their weights.
(e) For each occupant m count and store $\beta_t^m(r_t, a_t)$.

M-step:

1. Calculate γ_t^m and δ_t^m using equations (5) and (6) and then normalize.
2. Update parameters using equations (7) and (8).

Repeat

Table 1: Monte Carlo EM approach

Multiple occupants introduce uncertainty that could hurt the accuracy of learned models. A common method to minimize this uncertainty is to use the Expectation-Maximization (EM) algorithm [12]. The EM algorithm is an iterative approach to finding parameters that maximize a posterior density. The idea is to use current model parameters to estimate the expectations (E-step) of the distribution. The model parameters are then updated (M-step) using the expectations from the E-step. The steps are repeated and in each iteration the model parameters are improved. Eventually the algorithm converges to a local maximum.

A version of the EM algorithm called Monte Carlo EM [44, 73] takes advantage of the set of particles representing the posterior. Researchers at Intel used this technique with GPS readings to learn models of movement and transportation methods of a traveler in the city [59] . In this version both forward and backward updates are applied to the Bayes filter at each time step. At each forward and backward step, the algorithm examines each particle and counts the number of transitions between rooms and activities for each occupant. The counts from forward and backward phases are normalized and then multiplied and used to update model parameters. The learning algorithm is introduced thoroughly for Monte Carlo HMMs in [71].

$\alpha_t^m(r_t, a_t)$ is the number of particles in which occupant m is in room r and performing activity a during the *forward* pass.

$\beta_t^m(r_t, a_t)$ is the number of particles in which occupant m is in room r and performing activity a during the *backwards* pass.

$\gamma_{t-1}^m(r_t, r_{t-1}, a_{t-1})$ is the probability that occupant m will move from room r_{t-1} to room r_t in activity a_{t-1} at time $t-1$.

$\delta_{t-1}^m(a_t, a_{t-1}, r_{t-1})$ is the probability that occupant m will change from activity a_{t-1} to activity a_t from room r_{t-1} at time step $t-1$.

We define, [71]

$$\gamma_{t-1}^m(r_t, r_{t-1}, a_{t-1}) \propto \alpha_{t-1}^m(r_{t-1}, a_{t-1}) p(r_t^m | r_{t-1}^m, a_{t-1}^m) \beta_t^m(r_t, a_{t-1}) \tag{20}$$

and

$$\delta_{t-1}^m(a_t, a_{t-1}, r_{t-1}) \propto \alpha_{t-1}^m(r_{t-1}, a_{t-1}) p(a_t^m | a_{t-1}^m, r_{t-1}^m) \beta_t^m(r_{t-1}, a_t) \tag{21}$$

After the counting phase we update parameters as:

$$p(r_t^m | r_{t-1}^m, a_{t-1}^m) = \frac{\sum_{t=2}^T \gamma_{t-1}^m(r_t, r_{t-1}, a_{t-1})}{\sum_{t=2}^T \sum_{r_t \in contiguous r_{t-1}} \gamma_{t-1}^m(r_t, r_{t-1}, a_{t-1})} \tag{22}$$

and

$$p(a_t^m | a_{t-1}^m, r_{t-1}^m) = \frac{\sum_{t=2}^T \delta_{t-1}^m(a_t, a_{t-1}, r_{t-1})}{\sum_{t=2}^T \sum_{a_t \in \{moving,\ moving\}} \delta_{t-1}^m(a_t, a_{t-1}, r_{t-1})} \tag{23}$$

See Table 1. for a summary of particle filtering with MCEM model learning.

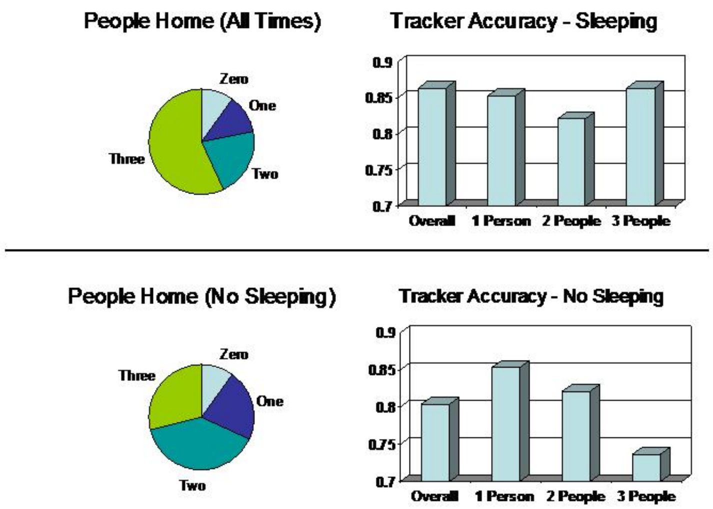

Figure 6: Results of tracking experiment based on the number of occupants at home during experiment, with sleeping periods present and removed. Results do not include the time when no occupants were at home.

4.5 Experiments

The tracker updates location and activity predictions every second. In the following experiments accuracy is measured as the number of seconds in which the maximum likelihood predictions of the tracker match the labeled location tag. In simulated experiments the location of each occupant is known, but experiments in the real environment required hand-labeling. Results are reported for real-time, online tracker performance.

4.5.1 Real Data

We conducted experiments using data generated by one to three occupants in an instrumented environment over the course of several months. The instrumented three story house contains twenty separate rooms, is 2824 square feet, and was home to two males (including the author), one female, a dog, and a cat. Detailed public information on the house can be referenced at the Allegheny County Real Estate web site [25]. The house contained one RFID reader, twenty four motion detectors, and twenty four contact switches. In these experiments we did not use break beam sensors or pressure mats, although they are included in later simulated data. Sensor and motion models were learned offline using data from when each person was home alone.

We used eight wireless keypads to help with hand-labeling. The keypads have one button for each of the three occupants. During experiments when anyone entered a room with a keypad, they pushed the button corresponding to their name. The wireless keypads were placed on the front door, the kitchen, the living room, the study, the downstairs bathroom, the upstairs bathroom, and each of the two bedrooms. They acted as road signs to help a human labeler disambiguate the data stream and correctly label the movements and identity of each occupant.

One person experiment. In the first experiment one person moved through the house. The occupant ultimately visited every sensor (including doors, drawers, and the refrigerator) and moved with varying speed and direction. The occupant conducted several common tasks, such as making a sandwich in the kitchen and pausing to use the computer in the study.

There were over 1200 sensor readings. The tracker used models trained for the occupant on previously collected data. Accuracy was 98.2%. We found that even for a single occupant accuracy was never 100% because of occasional lag between entering a room and triggering a sensor.

Two person experiment. In the second experiment two occupants moved through the house. Two ambiguous situations were scripted in which both occupants shared anonymous sensors and then separated. The scenario is as follows: two occupants enter the front door thirty seconds apart and move throughout the house without meeting. After fifteen minutes they meet in the living room. One occupant then moves to his bedroom and then returns to the living room. Next, the other occupant visits his own room and then returns.

The experiment spanned approximately thirty minutes, and occupant location was correctly predicted for over 98% of the total running time of the experiment. The tracker used individualized motion models, which was critical for recovering from the ambiguous situations. Both times the system recovered when the occupant arrived at their own bedroom door. We conducted another experiment in which generic models were used and found that one recovery was predicted correctly and the other not. In other words, the motion models contained information about who was more likely to visit a bedroom, and this information was used to recover identity.

Three person experiment. We measured tracker performance over a five day period for all occupants. There were no guests during this period. When the house was not empty, on average there was one occupant at home 13% of the time, two occupants home 22% of the time, and all three occupants home for 65% of the time. During the experiment every occupant slept in the house. The tracker used individual motion models for the three occupants. There were approximately 2000 sensor readings each day for a total of more than 10000 readings. We do not consider the time when no one was home.

On the whole, the tracker correctly classified 84.6% of the experiment. There was no significant difference in accuracy between occupants. The tracker was accurate 85.3% of the time when there was one occupant, 82.1% for two occupants, and 86.4% for three occupants. Accuracy for three occupants drops to 73.7% when sleeping periods are removed. See Figure 7 for a breakdown.

This experiment more than any other was affected by noisy sensors and animals. The sensors very rarely exhibited sporadic noise, but were sensitive to environmental factors. The dog in particular caused problems by introducing non-random noise when following occupants or moving between occupants in different rooms. The dog's location was highly correlated to occupant location and activity (e.g., the dog rushes to the kitchen for food when the owner wakes up in the morning). Also, for more than one day a motion detector on the back porch fired sporadically, due to movement caused by wind. Despite this noise, the overall accuracy was better than expected. Accuracy in the case of a single occupant, however, was lower than expected. We attribute this to the low density of sensors which contributed to significant periods of time between readings, especially with a single occupant at home. During these ”quiet” times, no new information arrived to help the learner recover from mistakes. In future experiments we plan to use many more sensors, as well as pressure mats and break beam sensors.

4.5.2 Simulated Data

The simulator. We implemented a program to simulate the data generated by occupants in an instrumented environment. The simulator can generate data from any number of motion detector, contact switch, and pressure mats per room, as well as break beam sensors on doors between rooms. The number of occupants, room structure, doorway location, and noise rates can be specified via command line parameters. ”Noise” is defined as a random sensor measurement. Each occupant obeys a first order HMM motion model that can be set by hand or initialized randomly.

Simulated occupants are introduced to the environment from the same starting state and identified correctly from this state, to imitate the RFID set up in the entry way of the real house. Henceforth, each occupant was unlikely to re-enter this ID state, with the average amount of time spent there being less than thirty seconds out of every hour. Unlike in reality, the movements of simulated occupants are truly independent. Simulated occupants were active (moving) approximately 15% of the time. There was a sporadic sensor reading about once every ten minutes. The simulated occupants followed hand crafted motion models in every experiment. The particle filter tracker used the same sensor model for each occupant and learned the parameters of motion models both online and offline. See the Appendix for more experimental details.

Experiment	**Accuracy**
same	0.99 ± 0.0001
opposite	0.99 ± 0.0003
middle	0.66 ± 0.002
uniform	0.46 ± 0.01

Table 2. Comparison of motion model experiments.

Motion model comparison. This experiment explored the value of unique motion models for tracking multiple occupants. We used six rooms, one entry way connected to a circular hallway composed of the other five rooms. Each room contained one motion detector, contact switch, and pressure mat. There were break beam sensors in every doorway. We tracked two simulated occupants for one hour with ten trials for each experiment. Occupants are identified in the entry room. They leave the entry room on the first time step and do not return.

We used four different motion models to generate the data and allowed the tracker to use the correct models. We ran four experiments (corresponding to the four motion models) called *same*, *opposite*, *middle*, and *uniform*. In the *same* experiment both occupants always walk in a clockwise direction. In the *opposite* experiment one occupant always walks clockwise and the other counter-clockwise. In the *middle* experiment, each occupant was %75 likely to transition clockwise. In the uniform experiment both occupants use the same uniform model in which they are equally likely to transition to any contiguous room. See the Appendix for the exact motion models.

We found that tracker accuracy depends heavily on occupant predictability. Accuracy was perfect in the *same* and *opposite* experiments, where each occupant performed the same action at each time step. Whether both occupants performed the same action (walking clockwise) or opposite actions (one clockwise and the other counter-clockwise) did not matter. Accuracy suffered in the *middle* experiment, when the transition probability was lowered to 75%. Accuracy was lowest for *uniform* models in which movements are completely unpredictable and identical between occupants. Results are summarized in Table 2.

Small house experiments. These experiments simulated a small house with ten rooms (three bedrooms, two bathrooms, a kitchen, living room, dining room, and hallways) in order to explore several questions. Motion models for five occupants were generated by hand (see Appendix). The models describe typical movements of occupants with the first three occupants having their own bedrooms and the last two occupants as guests. Each experiment tracked occupants for one hour and was run for ten trials. Figure 8 summarizes our results.

First, we looked at the impact of sensor configurations on tracking accuracy. See Figure 8a. In this experiment we used three sensor configurations to track three occupants. The *normal* configuration contains one motion detector, contact switch, and pressure mat per room, the *extra* configuration contains three of each type per room, and the *fewer* configuration contained only one motion detector per room. In general, more sensors improve accuracy. The *fewer* configuration exhibited high variability and didn't seem to be improved by adding more particles. This is because the data association was trivial for an environment with so few sensors. Also, with fewer sensors come fewer measurements, and longer periods before tracker recovery. In particular, motion detectors have an update rate of eight seconds. This means that after a motion detector triggers there is a window of time in which movement is undetected. This can lead to problems in an environment sparsely populated with motion detectors. The number of particles will need to grow for sensor configurations with hundreds of sensors per room, which will have much more complex data associations.

Second, we examine how different approaches to model learning affect accuracy. See Figure 8b. The number of particles is set to fifty, and parameter learning techniques are varied. We found earlier that an accurate motion model can greatly improve tracking accuracy. There are several methods available for learning these motion models. One method is to use simple counting to train a model using data from when the occupant is home alone. Alternately, we can use probabilistic methods to train a model online, while several occupants may be home. Three methods were used to train model parameters, (1) learning motion models off-line given one day of data generated by occupants that are alone (*offline*), (2) on-line via the Monte Carlo EM algorithm (*online*) discussed in section 4.4, and (3) the MCEM algorithm seeded by models already trained offline on one hour of single occupant data (*both*). In general, the *offline*

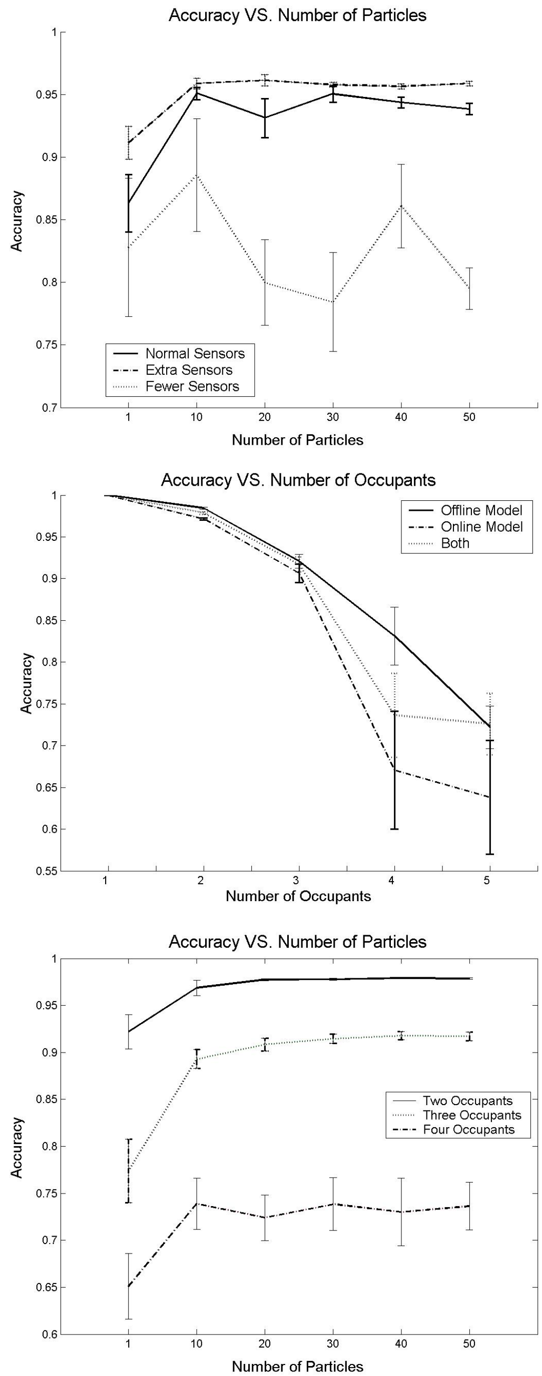

Figure 7: Results of simulated experiments.

method had highest accuracy, followed by *both* and with *online* learning last. These results make sense in a simulator, but in the real world occupants do not behave independently. Models learned offline will represent single occupants and will not be as effective for tracking multiple occupants.

We found the *both* method, of seeding a model with offline data and continuing to learn online, is the most promising route. As the number of occupants rises from two to three to four, we see the *online* method take a big accuracy hit. This is expected, as model learning will be confounded by multiple interfering occupants. However, as the number of occupants rises from four to five, we see that the *online* method takes a small hit, and the *both* method actually keeps up with the *offline* method. This could be because the fourth and fifth occupants are both guests, and have very similar motion models. Most accuracy is focused on the first three occupants, and the last two are equally unpredictable, lessening the importance of model learning.

Third, we examine how accuracy is affected as more occupants are added to the environment. See Figure 8b & 8c. In Figure 8b we varied the number of occupants and used fifty particles, and in Figure 8c we varied the number of particles and used offline model learning. We found that the accuracy plateaus as the number of particles are increased. As the number of occupants increases the step from one to ten particles is increasingly important. Due to efficient data association methods, the tracker does not need hundreds or thousands of particles, and in the future it will need even fewer. Obviously, accuracy drops as more occupants are tracked simultaneously. The difference between one and two occupants is much less than the difference between three and four occupants. The size of the environment and number of sensors directly impacts how many occupants can be accurately tracked.

In summary:

- **Motion models.** Highly predictive motion models (non-uniform) improve accuracy, regardless of whether occupants behave similarly. In reality, the differences between motion models show up in private areas, like bedrooms and bathrooms. The bigger these differences, the easier data association becomes and the more accuracy improves.

- **Particles.** The number of particles required depends on the complexity of the data association problem. More particles are only necessary for environments with many occupants and sensors. In our experiments we found negligible accuracy improvements after twenty or so particles, even for up to five occupants.

- **Sensors.** More sensors will increase accuracy, particularly when there are many occupants. Motion detectors are the most active sensors, but they have a ”down time” after detecting motion. Adding more sensors reduces this gap.

- **Occupants.** More occupants will decrease accuracy, particularly if parameter learning is performed completely online.

5 Proposed Research

5.1 Activity Recognition

We propose extending already completed work to include recognition of more complex activities chosen from the Activities of Daily Living and Extended Activities of Daily Living. Section 3.1.1 outlines the specific list of activities. Locomotion is incorporated into preliminary experiments in a simple manner, but more complex activities call for more complex models. These models must span households with different sensor configurations and use training data that can be collected and labeled easily and quickly by non-experts. They should also easily incorporate sensors that directly detect certain activities. We now describe how to integrate the results of any probabilistic activity recognition routine into our existing tracker. Next, we review some common approaches to activity recognition and describe a particularly promising technique. Finally, we introduce a novel approach to collecting labeled examples of activities.

5.1.1 Integration into Existing Framework

Activity recognition must be performed in real-time alongside our tracker within the framework of a dynamic Bayes net. Currently, the state of an occupant represents location and activity. The locations are the set of rooms in the environment and activities are either moving or not moving. We plan to incorporate new activities by incrementally expanding the set of mutually exclusive activities. This approach has two main benefits. First, having mutually exclusive activities ensures a linear increase in the number of model parameters as new activities are added. Second, we need only an arbitrary set of activity classifiers. Each classifier will independently provide a probability for the activity it has been trained to recognize, and can be easily switched out if better classifiers are developed.

This approach makes several assumptions : (1) activities can be described by sequences of the same atomic *events* used by the tracker, (2) activities are mutually exclusive (e.g., each occupant performs one activity at a time, although several occupants may perform activities simultaneously), (3) each activity is room specific and is not performed in more than one room, and (4) each occupant performs her own activity independently (activities are not performed in groups).

We clarify with a series of examples. First, only one activity will be most likely at any given time. To make this inclusive, one possible activity is that none of the other activities are occurring. Second, activities may occur in multiple rooms, but activities that occur across rooms are not recognized. For example, locomotion is recognized anywhere, but carrying laundry from the bedroom to the basement is not. Third, models trained to recognize individual activities will likely be confounded by observations from multiple activities (e.g., eating while watching television). One solution is to form separate models of common combinations of activities. Finally, although activities requiring participation of more than one person can not be modeled, we can model when the same activities occur simultaneously. For example, it is possible to recognize when a family eats dinner or when two people wash dishes together.

5.1.2 Approach

Our goal is to provide a real-time probability of each activity occurring. Probability should rise between the onset and finish of an activity. Specifically, we wish to incorporate this probability into our sensor model for each particle j:

$$p(z_t|r_t^{(j)}, a_t^{(j)}) \tag{24}$$

This is the probability of the most recent observation given that the occupant is engaged in activity $a \in A$ from the room $r \in R$. The presence of an activity can be highly discriminative when location alone is not. For example, if two occupants are in the same room but one is moving and the other stationary, data association is simplified and the two are identified, despite sharing the *same anonymous sensors*.

The task will be to find the probability that a sequence of observations over time were generated by a specific activity model. This is a fundamental problem for hidden Markov Models (HMMs). We propose to incorporate an HMM classifier for each possible activity. Activity recognition from discrete-valued time series data has been explored through a variety of hidden Markov model (HMM) approaches. They fit easily into a probabilistic framework that can account for dynamically time-varying activity sequences. See [61] for a detailed description of how HMMs work. We choose a promising variant of the HMM designed to minimize the amount of retraining necessary for new installations and sensor configurations.

Layered Hidden Markov Models (LHMMs) have been used for activity recognition over varying temporal granularity and with varying levels of abstraction [56]. LHMMs can be considered a cascade of HMMs. In LHMMs each layer of the architecture is connected to the next layer via its inferential results. The representation segments the problem into distinct layers that can operate at different temporal granularities. See Figure 9 for an example of a three-layer LHMM representing dish washing. LHMMs have several benefits:

- *LHMMs are robust to different instrumented environments.* Low level sensor configurations differ between installations, but higher level behavior often remains the same. The structure inherent in LHMMs can preserve higher level activity recognition, even when lower level sensors change between installations. For example, in Figure 9, *washing dishes* depends upon *using the sink*, but *using the sink* may depend on a *sink pressure mat*, *faucet sensor*, or any other sensor (or combination of sensors).

- *LHMMs require less training data.* Additional model structure lowers the number of parameters necessary to learn. Retraining is less rigorous in LHMMs because re-tuning is limited to isolated sub-HMM levels that may depend only upon specific sensor configurations.

- *LHMMs are highly discriminative.* The distance between the log-likelihoods of each activity occurring has been shown to be bigger for LHMMs than for standard HMMs. This makes the results of an LHMM more *discriminative* between possible activities.

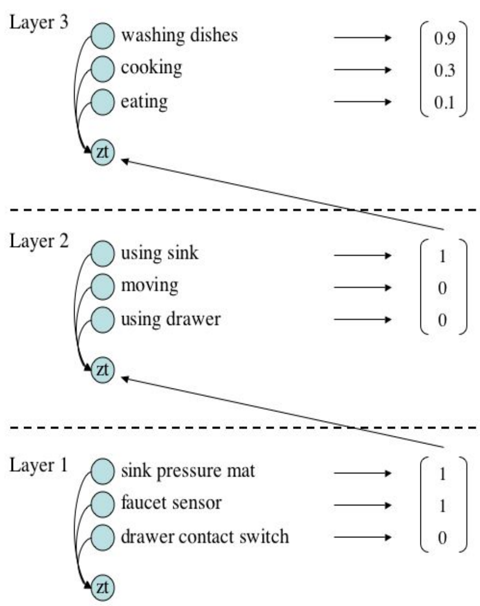

Figure 8: Outline of a typical LHMM for a kitchen for one time slice t. Each layer contains several HMMs trained to recognize observations z, generated by previous layers.

We propose to train a LHMM to recognize our list of ADLs. At each time step t for each particle j each occupant m will maintain a LHMM that is trained to recognize possible activities. The observations assigned to each occupant will be used to update the LHMM at each time step. The likelihoods of each ADL will be normalized and then used to update the tracker. These LHMMs will be trained off-line, so that at each time step we need only find the observation probability of each ADL, given the latest observation. A filter with 100 particles and 5 occupants requires 500 such updates at each time step.

5.1.3 Augmented Recall Survey

In order to learn a classifier, supervised learning algorithms require labeled training examples. Labeling activities is particularly expensive, requiring human expertise for each new instrumented environment. Unique sensor configurations between households will cause the same activities to appear different, although we can partially alleviate this problem by using LHMMs to abstract activity models into low and high levels. Our goal is to solve STAR with a minimal impact to daily routine. Ideally, our labeling technique should occur once or infrequently, be performable by a non-expert, require no additional instrumentation or change to environment, and not interrupt the tasks it gathers data about. Several standard methods exist for collecting this training data [38].

- **Interviews**. Occupants are interviewed individually or in groups.
- **Direct observation**. Trained observers watch the occupants (either live or via video) and hand-label events.
- **Self report : Recall Survey**. At the end of a time period the occupant records which activities occurred and when. This method is subject to the (often incorrect) memories of the occupant.
- **Self report : Time diaries**. Occupants write down what they do during the day as it happens or at set intervals. This improves recall accuracy but interrupts the daily routine.
- **ESM/EMA**. The experience sampling method (ESM), also called ecological momentary assessment (EMA), prompts the occupant for input through a device in the environment, either at timed intervals [34] or when activity levels are above a threshold [38]. This method requires a device placed in the environment and if used too often can irritate occupants.

We propose a novel approach, which we call the *Augmented Recall Survey* (ARS). The idea is to automatically group activities into episodes, convert the best episodes into English text summaries, and then to administer a multiple choice test to acquire labels.

Initially, data is collected for several days or weeks. Next, we segment rough examples of activities called *episodes* from the data. An episode is composed of a varying length time series of sensor measurements. Unsupervised segmenting of time series data into episodes is sometimes known as the *episode recovery* problem. A preliminary method is to pull out periods of time with sustained levels of activity over some threshold. More complicated methods include using mutual information statistics [20] or similarly, the use of a boundary entropy measure [18]. We plan to explore several approaches, simplifying the problem by using our assumption that activities occur one at a time in single rooms and using data collected only when one occupant is present.

Next, similar episodes are clustered into groups. Unsupervised clustering of time series data has been explored in a variety of domains. We plan to explore several methods, including a Bayesian approach used by researchers at the University of Massachusetts to cluster the activities of a mobile robot using onboard sensor input. They employ the Bayesian Clustering by Dynamics (BCD) approach, in which first-order Markov chains capture simple dynamics and a clustering algorithm groups episodes likely to have been generated by the same process [45]. The BCD approach has been successfully used to cluster robot activities as well as simulated war games, stock market behavior, and the fugues of Bach.

So far the ARS method has been performed in an *unsupervised manner*. The advantage is that no human intervention is required and numerous episodes can be collected and grouped. The disadvantages are that the examples may be inaccurately grouped and may not conform to intuitive ideals of activity. Regardless, we plan to investigate the use of anonymous activity groups for improving tracker accuracy. Activities can be very predictive of location, regardless of whether we know the name of the activity, or whether the activity represents an ADL at all. We will determine how much improvement is possible in a completely unsupervised system, without any call for labeled training data.

To train supervised classifiers for specific ADLs we would like to have every episode labeled. Unfortunately, this is a tedious and time consuming process that must be done by hand. Instead, we only choose a subset of the episodes in the order that improves activity recognition accuracy the fastest. Specifically, the learner is given a set of unlabeled training examples and allowed to choose a subset for labeling. The choice of each member of this subset may depend upon the examples already chosen. The problem of choosing the best training examples for (expensive) hand labeling is fundamental to the active learning community, and is called *selective sampling* or *design of experiment* (DOE) [52]. For reinforcement learning it is known as the *exploration vs. exploitation* tradeoff, where the learner must balance short term payoffs with learning.

We intend to investigate an approach called *query by committee* (QBC) [29]. In QBC several classifiers are generated and each independently classifies the training examples. The examples which cause the most disagreement between classifiers are chosen to be labeled. It has been shown that this approach can exponentially reduce the number of necessary labels. The algorithm has been extended to probabilistic classification [65] and the running time made tractable on large problems [62]. We expect to make a novel contribution due to several unique constraints. First, the number of classifiers is not given. Second, unlabeled examples may be collected in real-time and we wish to provide an ever improving (and expanding) set of classifiers. Third, episodes may be easy or difficult to hand label. We need to take into account the ability of the human labeler to correctly label episodes.

An unlabeled episode is chosen and then translated into English text via a tool which we call the Narrator [74]. The Narrator is a finite state machine that parses movement and activity information and generates a concise, readable summary. It is composed of a set of states, an input alphabet, and a transition function that maps symbols and states to the next state. The set of states represent English words and phrases, while the input alphabet is composed of sensor readings and times. To add some variety to the language, some states have more than one transition for a given symbol. A lookup table maps the room and occupant identities reported by the tracker to room and occupant names. The resulting summary represents important daily events in a compact format, although the tracker provides many thousands of second-by-second predictions.

The English summary is hand-labeled by the occupant (or a friend or relative) via a multiple choice test. As each episode is labeled (or not) the information is propagated through the committee of activity models, and the next unlabeled episode is chosen. We intend to conduct a user study to determine the effectiveness of our labeling strategy. Participants will be presented text descriptions generated by the Narrator and asked to choose one of several possible activities, including an ”I don’t know” choice. Each labeled example will improve the activity classifier, which can be used in turn to update the tracker results.

5.1.4 Preliminary Activity Recognition Experiment

To explore how well we could use a supervised learner to predict a common activity using simple sensors, we instrumented a graduate student apartment (6 rooms, 8 light switches, 2 residents) with motion detectors and instrumented light switches. We used a simple naive Bayes learner [46] on two months of data to predict (1) when a light switch would be manipulated and (2) which light switch it would be.

Our feature set was comprised of the number of times each motion detector (one per room) triggered and the number of times each light switch was manipulated over an array of time windows. Feature extraction was carried out automatically using mutual information scores. We assume that a parametric model (a multinomial mixture model) can describe light switch usage patterns. We use training data gathered from the environment to find Bayes-optimal estimates of model parameters. Straight classification on new instances is done by using Bayes’ rule to calculate the posterior probability of each class and choosing the most likely class. We use a threshold on the posterior probabilities already calculated to make event detection predictions in real time. The learning task is simplified by making the *naive Bayes assumption* (i.e., assuming independence between attributes), allowing parameters for each attribute to be learned separately.

The results were promising; the learner correctly predicted which switch was going to be flipped 75% of the time. In predicting when a light switch was going to be touched at all, the learner generated one false positive for roughly every four hours of data.

6 Contributions

This research defines the simultaneous tracking and activity recognition (STAR) problem. Our solution fills an important gap in existing research in ubiquitous computing by exploring a sensory modality that has been largely ignored in favor of vision and audition. There are several potential intellectual contributions of this work:

- New instrumentation techniques may be developed in the area of cost-effectively instrumenting entire buildings.We will explore realistic, effective ubiquitous systems by seriously attempting to create sensing systems at the building scale.
- New algorithms may be created or existing algorithms borrowed and improved to provide automatic activity recognition and room-level tracking from noisy sensor data.
- Cost effective tracking and activity recognition creates the possibility of fundamental new interfaces or ways of interacting with structures and information processing resources.

We will explore a neglected set of pre-existing sensors that are non-invasive, cheap, and easy to install and maintain in order to introduce cost-effective automatic health monitoring. We will push the boundaries of what these sensors can describe. Our research effort can collect data for months or years in a permanent home setting, rather than a permanent corporate setting or a temporary residential setting. Automatic health monitoring could help our growing elderly population to live independently longer. This research could also relatively instantly introduce ubiquitous computing services to thousands of corporate and residential buildings, possibly contributing to changing the way our society lives and works.

7 Research Plan

Table 3. contains a tentative plan of proposed research activities.

Activity	**Time**
Augmented Recall Survey and user study. Activity recognition models.	Summer 2004
Collaboration with Intel Research in Seattle. Expand, integrate activity recognition.	Fall 2004
Instrument a home health care facility.	Spring 2005
Gauge effectiveness of STAR prototype in instrumented facility.	Summer 2005
Write thesis and defend.	Fall 2005

Table 3. Rough timetable of thesis.

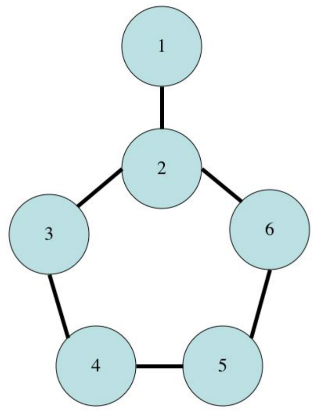

Figure 9: Physical layout of simulated experiment.

8 Appendix

8.1 Motion model experiment.

8.1.1 Physical structure

See Figure 9 for the physical layout of the experiment.

8.1.2 SAME model

$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \left| \begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right| \end{array}$$

8.1.3 OPPOSITE model

$$
\text{Occupant A} \quad \begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 1 \\
3 & 0 & 1 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 1 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 1 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 1 & 0
\end{array} \quad \text{Occupant B}
$$

$$
\begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 1 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 1 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 1 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 1 \\
6 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}
$$

8.1.4 MIDDLE model

$$
\begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0.25 & 0 & 0 & 0 & 0.75 \\
3 & 0 & 0.75 & 0.25 & 0 & 0 & 0 \\
4 & 0 & 0 & 0.75 & 0.25 & 0 & 0 \\
5 & 0 & 0 & 0 & 0.75 & 0.25 & 0 \\
6 & 0 & 0 & 0 & 0 & 0.75 & 0.25
\end{array}
$$

8.1.5 UNIFORM model

$$
\begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0.33 & 0.33 & 0 & 0 & 0.33 \\
3 & 0 & 0.33 & 0.33 & 0.33 & 0 & 0 \\
4 & 0 & 0 & 0.33 & 0.33 & 0.33 & 0 \\
5 & 0 & 0 & 0 & 0.33 & 0.33 & 0.33 \\
6 & 0 & 0.33 & 0 & 0 & 0.33 & 0.33
\end{array}
$$

8.2 Small house experiments.

8.2.1 Physical structure

See Figure 10 for the physical layout of these experiments.

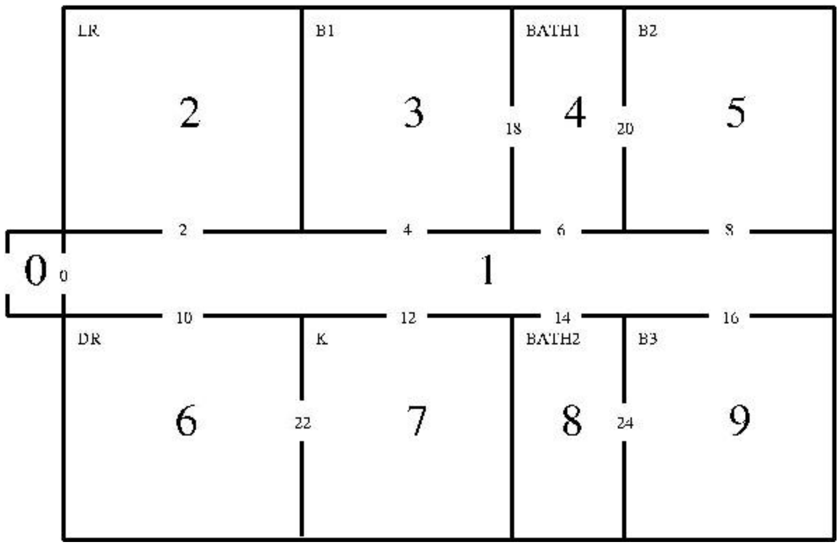

Figure 10: Physical layout of simulated experiment.

8.2.2 Occupant A

	0	1	2	3	4	5	6	7	8	9
0	0.1	0.9	0	0	0	0	0	0	0	0
1	0.1	0.04	0.2	0.04	0.04	0.04	0.1	0.05	0.19	0.2
2	0	0.05	0.95	0	0	0	0	0	0	0
3	0	0.85	0	0.1	0.05	0	0	0	0	0
4	0	0.7	0	0.1	0.1	0.1	0	0	0	0
5	0	0.6	0	0	0.1	0.3	0	0	0	0
6	0	0.1	0	0	0	0	0.7	0.2	0	0
7	0	0.6	0	0	0	0	0.3	0.1	0	0
8	0	0.1	0	0	0	0	0	0	0.7	0.2
9	0	0.1	0	0	0	0	0	0	0.2	0.7

8.2.3 Occupant B

	0	1	2	3	4	5	6	7	8	9
0	0.1	0.9	0	0	0	0	0	0	0	0
1	0.1	0.04	0.1	0.04	0.18	0.18	0.1	0.2	0.04	0.02
2	0	0.4	0.6	0	0	0	0	0	0	0
3	0	0.6	0	0.1	0.3	0	0	0	0	0
4	0	0.1	0	0.05	0.7	0.15	0	0	0	0
5	0	0.1	0	0	0.2	0.7	0	0	0	0
6	0	0.1	0	0	0	0	0.6	0.3	0	0
7	0	0.1	0	0	0	0	0.2	0.7	0	0
8	0	0.7	0	0	0	0	0	0	0.2	0.1
9	0	0.6	0	0	0	0	0	0	0.1	0.3

8.2.4 Occupant C

	0	1	2	3	4	5	6	7	8	9
0	0.1	0.9	0	0	0	0	0	0	0	0
1	0.15	0.02	0.2	0.2	0.17	0.02	0.1	0.1	0.02	0.02
2	0	0.2	0.8	0	0	0	0	0	0	0
3	0	0.05	0	0.8	0.15	0	0	0	0	0
4	0	0.05	0	0.2	0.7	0.05	0	0	0	0
5	0	0.3	0	0	0.6	0.1	0	0	0	0
6	0	0.1	0	0	0	0	0.6	0.3	0	0
7	0	0.1	0	0	0	0	0.4	0.5	0	0
8	0	0.9	0	0	0	0	0	0	0.05	0.05
9	0	0.9	0	0	0	0	0	0	0.05	0.05

8.2.5 Occupant D

	0	1	2	3	4	5	6	7	8	9
0	0.1	0.9	0	0	0	0	0	0	0	0
1	0.15	0.1	0.3	0.1	0.1	0.03	0.04	0.1	0.04	0.04
2	0	0.05	0.95	0	0	0	0	0	0	0
3	0	0.1	0	0.8	0.1	0	0	0	0	0
4	0	0.1	0	0.3	0.5	0.1	0	0	0	0
5	0	0.7	0	0	0.2	0.1	0	0	0	0
6	0	0.3	0	0	0	0	0.6	0.1	0	0
7	0	0.3	0	0	0	0	0.5	0.2	0	0
8	0	0.85	0	0	0	0	0	0	0.1	0.05
9	0	0.85	0	0	0	0	0	0	0.1	0.05

8.2.6 Occupant E

	0	1	2	3	4	5	6	7	8	9
0	0.1	0.9	0	0	0	0	0	0	0	0
1	0.2	0.03	0.6	0.01	0.03	0.01	0.07	0.01	0.03	0.01
2	0	0.1	0.9	0	0	0	0	0	0	0
3	0	0.9	0	0.05	0.05	0	0	0	0	0
4	0	0.7	0	0.05	0.2	0.05	0	0	0	0
5	0	0.9	0	0	0.05	0.05	0	0	0	0
6	0	0.2	0	0	0	0	0.7	0.1	0	0
7	0	0.3	0	0	0	0	0.6	0.1	0	0
8	0	0.75	0	0	0	0	0	0	0.2	0.05
9	0	0.9	0	0	0	0	0	0	0.05	0.05

References

[1] G. Abowd, C. Atkeson, A. Bobick, I. Essa, B. Mynatt, and T. Starner. Living laboratories: The future computing environments group at the georgia institute of technology. In *Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, 2000.

[2] G. D. Abowd and E. D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1):29–58, 2000.

[3] M. Addlesee. Implementing a sentient computing system. *IEEE Computer*, 34(8):50–56, August 2001.

[4] D. W. Albrecht, I. Zukerman, A. E. Nicholson, and A. Bud. Towards a Bayesian model for keyhole plan recognition in large domains. pages 365–376, 1999.

[5] S. Allin, A. Barucha, J. Zimmerman, D. Wilson, M. Roberson, S. Stevens, H. Watclar, and C. Atkeson. Toward the automatic assessment of behavioral disturbances of dementia. In *UbiHealth 2003: The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications. Seattle, USA*, 2003.

[6] R. N. Anderson. Method for constructing complete annual us life tables, national center for health statistics. In *Vital and Health Stat*, volume 2, page 129, 1999.

[7] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. In *IEEE Transactions on Signal Processing*, 2002.

[8] Y. Bar-Shalom and T. Fortmann. Tracking and data association. *Mathematics in Science and Engineering. Academic Press*, 179, 1988.

[9] T. Barger, D. Brown, and M. Alwan. Health status monitoring through analysis of behavioral patterns. In *AI*IA 2003 - 8th National Congress of Italian Association for Artificial Intelligence: Workshop on Ambient Intelligence*, 2003.

[10] M. Bennewitz, W. Burgard, and S. Thrun. Learning motion patterns of persons for mobile service robots. In *In Proceedings of the IEEE International Conference on Robotic and Automation (ICRA)*, 2002.

[11] N. Bergman and A. Doucet. Markov chain monte carlo data association for target tracking, 2000.

[12] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, 1997.

[13] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. Easyliving: Technologies for intelligent environments. In *Handheld and Ubiquitious Computing*, 2000.

[14] L. Burgio, K. Scilley, J. M. Hardin, and C. Hsu. Temporal patterns of disruptive vocalization in elderly nursing home residents. *International Journal of Geriatric Psychiatry*, 16(1):378–386, 2001.

[15] L. Burgio, K. Scilley, J. M. Hardin, J. Janosky, P. Bonino, S. Slater, and R. Engberg. Studying disruptive vocalization and contextual factors in the nursing home using computer-assisted real-time observation. *Journal of Gerontology*, 49(5):230–239, 1994.

[16] C. Burkey. Environmental interfaces: Homelab, 2000.

[17] B. Clarkson, N. Sawhney, and A. Pentland. Auditory context awareness via wearable computing. In *In the Proceedings of the Perceptual User Interfaces Workshop*, 1998.

[18] P. Cohen, B. Heeringa, and N. Adams. An unsupervised algorithm for segmenting categorical time series into episodes. In *Working Notes of the 2002 ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining*, 2002.

[19] D. Colbry, B. Peintner, and M. Pollack. Execution monitoring with quantitative temporal bayesian networks. In *6th International Conference on AI Planning and Scheduling*, 2002.

[20] M. M. D. Magerman. Parsing a natural language using mutual information statistics. In *Eight National Conference on Artificial INtelligence (AAAI 90)*, pages 984–989, 1990.

[21] L. Davis, K. Buckwalter, and L. Burgio. Measuring problem behaviors in dementia: Developing a methodological agenda. *Adv Nurs. Sci.*, 20(1):40–55, 1997.

[22] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Em, mcmc, and chain flipping for structure from motion with unknown correspondence. *Machine Learning - special issue on Markov chain Monte Carlo methods*, 50:45–71, 2003.

[23] A. Doucet, N. de Freitas, and N. Gordon. In *Sequential Monte Carlo in Practice*, 2001.

[24] P. Erickson, R. Wilson, and I. Shannon. Years of healthy life, us department of health and human services, cdc, national center for health statistics. In *Hyattsville Maryland Statistical Notes*, 1995.

[25] A. C. R. Estate. 2003.

[26] K. P. Fishkin, H. Kautz, D. Patterson, M. Perkowitz, and M. Philipose. Guide: Towards understanding daily life via auto-identification and statistical analysis. In *UBIHEALTH 2003 The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, 2003.

[27] D. Fox, J. Hightower, H. Kautz, L. Liao, and D. Patterson. Bayesian techniques for location estimation. In *Proceedings of The 2003 Workshop on Location-Aware Computing*, October 2003. part of the 2003 Ubiquitous Computing Conference.

[28] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, July-September 2003.

[29] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Sample selection using the query by committee algorithm. In *Machine Learning*, pages 133–168, 1997.

[30] S. Helal, B. Winkler, C. Lee, Y. Kaddoura, L. Ran, C. Giraldo, S. Kuchibhotla, and W. Mann. Enabling location-aware pervasive computing applications for the elderly. In *First IEEE International Conference on Pervasive Computing and Communications*, page 531, 2003.

[31] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, August 2001. This article is also excerpted in "IT Roadmap to a Geospatial Future," a 2003 report from the Computer Science and Telecommunications Board of the National Research Council.

[32] L. Holmquist, J. Falk, and J. Wigstrm. Supporting group collaboration with interpersonal awareness devices. *Personal Technologies*, 3:13–21, 1999.

[33] T. Huang and S. Russell. Object identification in a bayesian context. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1997.

[34] S. E. Hudson, J. Fogarty, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. C. Lee, and J. Yang. Predicting human interruptibility with sensors: A wizard of oz feasibility study, 2003.

[35] C. Hue, J. L. Cadre, and P. Perez. The (mr)mtpf: particle filters to track multiple targets using multiple receivers, 2001.

[36] S. Intille and A. Bobick. A framework for recognizing multi-agent action from visual evidence. In *AAAI/IAAI*, pages 518–525, 1999.

[37] S. Intille and K. Larson. Designing and evaluating technology for independent aging in the home. In *Proceedings of the International Conference on Aging, Disability and Independence*, 2003.

[38] S. Intille, E. M. Tapia, J. Rondoni, J. Beaudin, C. Kukla, S. Agarwal, and L. Bao. Tools for studying behavior and technology in natural settings. In *Proceedings of UBICOMP 2003*, 2003.

[39] M. Jug, J. Pers, B. Dezman, and S. Kovacic. Trajectory based assessment of coordinated human activity. In *ICVS 2003*, pages 534–543, 2003.

[40] T. Kanade, R. Collins, A. Lipton, P. Burt, and L. Wixson. Advances in cooperative multi-sensor video surveillance. *In Proceedings of the 1998 DARPA Image Understanding Workshop*, 1:3–24, 1998.

[41] R. Karlsson and F. Gustafsson. Monte carlo data association for multiple target tracking, 2001.

[42] Z. Khan, T. Balch, and F. Dellaert. Efficient particle filter-based tracking of multiple interacting targets using an mrf-based motion model. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03)*, 2003.

[43] S. Lee and K. Mase. Activity and location recognition using wearable sensors. In *First IEEE International Conference on Pervasive Computing and Communications*, pages 24–32, 2002.

[44] R. A. Levine and G. Casella. Implementations of the monte carlo em algorithm. In *Journal of Computational and Graphical Statistics*, 2000.

[45] P. R. C. Marco Ramoni, Paola Sebastiani. Bayesian clustering by dynamics. In *Machine Learning*, pages 47(1): 91–121, 2002.

[46] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[47] J. McNeil. Disabilities affect one-fifth of all americans: Proportion could increase in coming decades. In *U.S. Census Brief*, 1997.

[48] D. Meyer. Human Gait Classification Based on Hidden Markov Models. pages 139–146, 1997.

[49] A. Mihailidis, B. Carmichael, J. Boger, and G. Fernie. An intelligent environment to support aging-in-place, safety, and independence of older adults with dementia. In *UbiHealth 2003: The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, 2003.

[50] D. Moore, I. Essa, and M. Hayes. Exploiting human actions and object context for recognition tasks. In *In Proceedings of IEEE International Conference on Computer Vision 1999 (ICCV99)*, 1999.

[51] M. Mozer. The neural network house: An environment that adapts to its inhabitants. In *In Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, pages 110–114, 1998.

[52] R. Myers and D. Montgomery. Response surface methodology: Process and product optimization using designed experiments. In *Wiley*, 1995.

[53] M. Ogawa, R. Suzuki, S. Otake, T. Izutsu, T. Iwaya, and T. Togawa. Long-term remote behavioral monitoring of the elderly using sensors installed in domestic houses. In *Engineering in Medicine and Biology. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] EMBS/BMES Conference*, volume 3, pages 23–26, 2002.

[54] M. Ogawa and T. Togawa. The concept of the home health monitoring. In *Enterprise Networking and Computing in Healthcare Industry. Healthcom 2003. Proceedings - 5th International Workshop on*, pages 71–73, 2003.

[55] T. Ogawa, S. Yoshino, M. Shimizu, and H. Suda. A new in-door location detection method adopting learning algorithms. In *PerCom 2003*, pages 525–530, 2003.

[56] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Fourth IEEE Int. Conf. on Multimodal Interfaces*, pages 3–8, 2002.

[57] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

[58] D. Patterson, D. Fox, H. Kautz, and M. Philipose. Expressive, tractable and scalable techniques for modeling activities of daily living. In *UbiHealth 2003: The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*, 2003.

[59] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *UBICOMP 2003: The Fifth International Conference on Ubiquitous Computing (October 12-15, 2003)*, 2003.

[60] M. K. Pitt and N. Shephard. Auxiliary variable based particle filters. In *Sequential Monte Carlo Methods in Practice (Cambridge: Cambridge University Press, 2000)*, 1999.

[61] L. Rabiner. A tutorial on hidden markov models. *Proceedings of the IEEE*, 77:257–286, 1989.

[62] N. T. Ran Gilad-Bachrachranb, Amir Navotanavot. Kernel query by committee, 2002.

[63] D. B. Reid. The application of multiple target tracking theory to ocean surveillance. In *In Proceeds of the 18th IEEE Conference on Decision and Control*, pages 1046–1052, 1979.

[64] B. Rhodes. The wearable remembrance agent: A system for augmented memory. In *In Proceedings of the 1st International Symposium on Wearable Computers*, pages 123–128, 1997.

[65] I. D. Shlomo Argamon-Engelson. Committee-based sample selection for probabilistic classifiers. In *Journal of Artificial Intelligence Research*, volume 11, pages 335–360, 1999.

[66] H. Sidenbladh and M. Black. Learning image statistics for bayesian tracking. *In IEEE International Conference on Computer Vision (ICCV)*, 2:709–716, 2001.

[67] T. Starner. Visual recognition of american sign language using hidden markov models. Technical Report Master's Thesis, MIT, Feb 1995, Program in Media Arts & Sciences, MIT Media Laboratory, Also Media Lab VISMOD TR 316 [http://www-white.media.mit.edu/vismod/people/publications/publications.html 860Kb], 1994.

[68] K. Sycara, K. Decker, A. Pannu, M. Williamson, and D. Zeng. Distributed intelligent agents. *IEEE Expert*, 11(6):36–46, December 1996.

[69] A. P. Tanzeem Choudhury. Sensing and modeling human networks using the sociometer. In *Proceedings of the International Conference on Wearable Computing*, 2003.

[70] S. Thrun, D. Fox, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In *in Sequential Monte Carlo Methods in Practice, New York: Springer-Verlag*, 2001.

[71] S. Thrun and J. Langford. Monte carlo hidden markov models, 1997.

[72] K. VanHaitsma, P. Lawton, M. Kleban, J. Klapper, and J. Corn. Methodological aspects of the study of streams of behavior in elders with dementing illness. *Alzheimer Disease and Associated Disorders*, 11(4):228–238, 1997.

[73] G. Wei and M. Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal fo the American Statistical Association*, 85(411):699–704, 1990.

[74] D. Wilson and C. Atkeson. The narrator : A daily activity summarizer using simple sensors in an instrumented environment. In *The Fifth International Conference on Ubiquitous Computing (UBICOMP) 2003 Demonstrations, Seattle, USA*, 2003.

[75] P. A. P. S. J. T. D. W. A. Woodruff. The mad hatter's cocktail party: A social mobile audio space that supports multiple simultaneous conversations. In *Human Factors in Computing Systems (CHI) 2003*, 2003.

[76] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of human body. ieee trans. pattern analysis. *Machine Intelligence*, 19(1):780–785, 1997.