People Tracking Using Anonymous, Binary Sensors

Daniel Wilson

The Center for Automated Learning & Discovery
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
dan.wilson@cs.cmu.edu

Abstract. Automatic health monitoring helps enable independent living for the elderly by providing specific information to caregivers. In this paper we use simple binary sensors in a Bayesian framework to provide simultaneous room-level location estimation and rudimentary activity recognition (ambulation). Results from experiments using a Bayes filter and simple data association are demonstrated. We discuss several improvements, including a particle filter approach to the data association problem and the Monte Carlo EM algorithm for online parameter learning, and demonstrate results from a simulated environment.

1 Introduction

In 2002 national health expenditures were estimated at US$1.5 trillion, or almost 15 percent of the GDP [55]. About 1 in 5 Americans have some kind of disability, and 1 in 10 have a severe disability [35]. People aged 65 and older are the fastest growing segment of the US population and by 2030 over 4 million Americans will be over the age of 85 [5]. Over 20% of people 85 and over have a limited capacity for independent living [19], and as a result they require continuous monitoring and daily care. Automatic health monitoring of the elderly and those with disabilities can improve the accuracy of pharmacologic interventions, track illness progression, and lower caregiver stress levels [17]. Additionally, [51] has shown that movement patterns alone are an important indicator of cognitive function, depression, and social involvement among people with Alzheimer's disease.

The basic goals of ubiquitous computing are aligned with the needs of automatic health monitoring. They include 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 on aware environments, including our own, has focused on implementing machine vision and auditory systems to do these tasks [4].

This research uses state-of-the-art machine learning techniques to exploit many "simple" sensors in order to automatically recognize human activity. Existing sensor infrastructures, particularly those employed by security systems, are used to provide automatic recognition of human behavior. These new abilities come at minimal privacy, monetary, and computational cost, and could be used on a large scale in homes and businesses. We examine the utility of applying anonymous, binary sensors such as motion detectors, contact switches, break-beam sensors, and pressure mats to automatic activity recognition and room-level tracking. We show that such sensors can tell us which rooms are occupied, count the occupants in a room, identify the occupants, track occupant movements, and recognize occupant activities.

2 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 [54, 12, 47], as well as cameras and microphones for activity recognition [16, 37]. Wearable sensors have been used for health monitoring [33], the facilitation of group meetings [25], memory augmentation [46], and augmented reality [48].

People tracking in particular 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, 9, 16, 30, 38, 22, 47]. See [24] 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 [13], and a target population of institutionalized Alzheimer's patients frequently strip themselves of clothing, including any wearable sensors [14]. A distributed network of many low cost sensors has several advantages over co-located sensors on a single platform (e.g., wearable sensors). 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. 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. The goal is to determine if a newly observed object is the same as a previously observed object. In AI, this problem is known as *object identification*. The solution offered by [42] has been applied to tracking automobile traffic using cameras, extending the technique introduced by [26] to accommodate many sensors. Neither approach deals well with sensor noise. In the multi-target tracking community the problem is well known as *data association*. Bayesian techniques, particularly particle filters, have been introduced as effective solutions to this problem [6, 31, 27]. In a recent experiment [21], a particle filter implementation used laser range finders and infrared badges to track six people simultaneously in an office environment for 10 minutes. The range finders provided anonymous, high granularity coordinates while the badge system identified occupants. This research uses a similar particle filter approach to solve the data association problem. However, we use a single

ID sensor and rely upon individual motion and activity models to resolve ambiguity. Data collected over the long-term provides an ever-improving model of the unique patterns of each occupant. Motion and activity models can identify occupants in lieu of additional ID-sensors.

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 [39, 41, 40]. The raw data generated during these experiments was made available to physicians, but has not been used for machine learning applications such as automatic tracking, behavior recognition, or pattern discovery. 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) [8]. They cluster sensor readings into rough groups based on room, duration, and time of day and demonstrate that many of the clusters correspond to ADLs. However, recognition and prediction of high-level behavior such as ADLs and location estimation has not been described with this set of simple sensors.

People tracking and activity recognition experiments typically occur in a laboratory setting in a corporate or educational building [29, 12, 15]. Recently, there has been an increase in the number of stand-alone instrumented home environments. [2], have built a home ubiquitous computing laboratory for a variety of experiments. [28] purchased a house and instrumented it with their own brand of generic, simple sensors for off-line activity recognition. [23] have instrumented a house with ultrasound localization and displays. [41] has instrumented real homes for months at a time. The Neural Network House [38] used neural networks to negotiate between energy conservation and comfort. None of these laboratories has had a permanent resident. Other groups, including our own, have instrumented health care facilities for a variety of experiments [4, 8, 36]. Our instrumented environment is unique in that we use cheap, off-the-shelf sensors for real-time, simultaneous tracking and activity recognition over a long period of time. The specific applications explored by other groups are exciting and inspirational. Our critical goals are to observe, recognize, and predict behavior, rather than to modify the environment.

3 What Are We Trying To Sense?

Physicians and caretakers (including relatives) need certain information to enable elders to live independently at home without being institutionalized. We wish to provide this information at minimal cost and with minimal impact to the occupant's life routine. The automatic collection of this information, called *automatic monitoring*, is predominantly composed of **location** and **activity** information. Below is a list of what we wish to automatically recognize.

3.1 Location & Activity Information

- **Occupancy.** Determine if there is anyone in a room.

- **Counting.** Determine how many people are in a room.
- **Individual Identification.** Determine the identity of each person in the home.
- **Room-Level Tracking.** Identify where each person is in the home.
- **Movement.** Recognize whether ambulation is occurring.

4 Anonymous, Binary Sensors

We are concerned with people tracking and activity recognition of several people in a home environment. In order to collect this information we must instrument the environment with sensors. There are two critical issues at hand : (1) choice of sensors, and (2) sensor placement. Ideally, the sensors we choose to use 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 activity or daily routine. Most people are not interested in dealing with new devices without a large perceived benefit.
- **Private.** Sensor data alone should not provide information the user wishes to keep private, 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 simply mounted to a surface, while wired sensors may require running cable to a central location. We envision a system available off-the-shelf to be fully 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 whether sensors are functioning properly.
- **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 via the data 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 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 Sensor Properties

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

- **Coverage.** There is a *volume of interaction* around each sensor. This radius can convey important information about occupant location. For instance, the motion detector has a coverage of around ten square feet, or the size of an average room. Break beam sensors cover a linear area. Contact switches and pressure mats must be directly manipulated through a physical 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 standing and moving their body. For instance, a motion detector is most likely to trigger due to a person walking in a room. 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 specifically placed on doors and drawers are manipulated only by active occupants. Break-beam sensors placed in doorways are triggered by someone actively walking through a doorway. On the other hand, pressure mats placed only in seating areas are triggered passively by seated occupants that are staying put.

- **Continuity.** 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.

4.2 Sensor Choice and Placement

There are many simple binary sensors to choose from. In fact, any sensor can be anonymous and binary with the proper thresholds. We chose four : motion detectors, contact switches, break-beam sensors, and pressure mats. These four sensors have different properties which, when exploited, can reveal a surprising amount of information. See Figure 1. for an overview of a typically instrumented room.

- **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. 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 attached to the ceilings in order to maximize room coverage. The detectors are wireless, pet-resistant, require both heat and movement to trigger, and run on battery power for over one year.

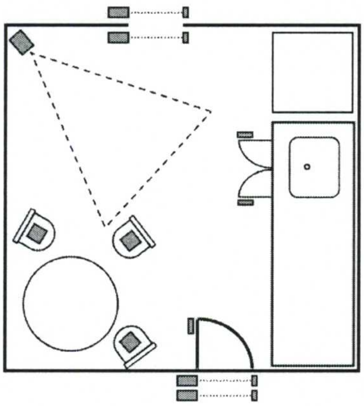


Fig. 1. Overview of typically instrumented kitchen. Grey squares represent contact switches, motion detectors, pressure mats, and break beam sensors.

- **Contact switches.** These inexpensive magnetic contact switches indicate a closed or open status. They are installed on every interior and exterior door, cabinet drawer, and the refrigerator.

- **Pressure mats.** We use these sensors to 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 it when it is reflected back. While the beam is interrupted the sensor changes state.

- **Radio Frequency Identification (RFID).** Although anonymous sensors can be used to maintain identity and location of occupants, at some point the identity must be given. 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 front or back door, as well as identify the occupant entering or leaving the environment.

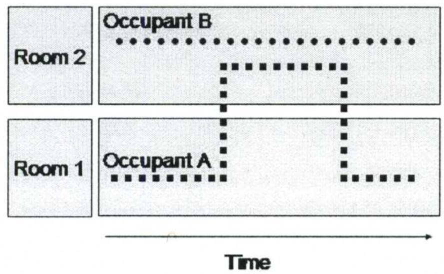


Fig. 2. Movement of two occupants through two rooms over time.

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.

5 People Tracking & Preliminary Activity Recognition

Room-level tracking provides occupancy, counting, and individual identification information. There are two main problems when tracking multiple people, (1) where are the people and, (2) which person is which? In the first problem observations are used to update the location of each occupant. In the second problem, known as the *data association* 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 share the same room and 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, occupant A enters the same room as B. Without further information, the identity of each occupant is uniform after A exits the shared room. There are several ways to reduce this ambiguity :

- **Increase the number of ID sensors.** The simplest approach solves the problem by using sensors that identify occupants outright. Unfortunately, ID sensors fail to satisfy the requirements for our domain (see Section 4.1).

- **Increase the sensor granularity.** The more sensors there are, the smaller the probability that multiple occupants will share the same anonymous sensor. For example, [49] tracks people with high granularity laser range finders; sensor collision does not occur unless people occlude each other very closely. This problem has extra significance for room level tracking with low granularity sensors. Clever sensor choice and placement can maximize granularity. For example, placing contact switches out of reach of pressure mats separates two occupants when one is seated and the other opens a drawer.

- **Learn individual movement patterns.** Over time, motion models represent particular habits of select individuals. Specific models can help the tracker recover from ambiguity as occupants follow regular habits (i.e., sitting in favorite chairs or sleeping in their own beds). This could resolve the ambiguity in Figure 2. If occupant B rarely enters room 1, person A is more likely to have left.

- **Recognize activities.** Activity recognition is performed alongside tracking, providing a novel method of recovery. Preliminary experiments model occupant ambulation, which in turn helps predict occupant location. For example, if person B is not moving and person A is moving, then person A is correctly identified as exiting room 2.

5.1 Discrete Bayes Filter

Bayes filters estimate the state of a dynamic system from noisy sensor data in real world domains [18]. The *state* commonly represents occupant location and sensors provide information about the state. A posterior probability distribution, called the *belief*, describes the probability that the occupant is in each state. A Bayes filter updates the posterior probability density over the state space at each time step, conditioned on the data. They model systems over time through the Markov assumption that the current state depends only on the previous state.

We estimate the state $x_t = \{x_t^1, x_t^2, ..., x_t^M\}$ of M occupants using the sensor measurements collected so far $z_{1:t}$. At each time step we receive the status of many binary sensors. The measurement $z_t =< e_t^1, e_t^2, ..., e_t^E >$ is a string of 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 provides initial categories of activity. The update equation is analogous to the forward portion of the forward-backward algorithm used in hidden Markov models (HMMs). See [44] for a detailed description of how HMMs work.

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \sum_{x_{t-1}} p(x_t|x_{t-1}) p(x_{t-1}|z_{1:t-1}). \tag{1}$$

The *sensor model* $p(z_t|x_t)$ represents the likelihood of measurement z_t occurring from state x_t. The *motion model* $p(x_t|x_{t-1})$ predicts the likelihood of transition from the last state x_{t-1} to the current state x_t. Currently, a simplifying independence assumption between occupants means that this motion model is factored as:

$$p(x_t|x_{t-1}) = \prod_{m \in M} p(x_t^m|x_{t-1}^m) \tag{2}$$

In later work we intend to utilize two models, one for occupants that are alone and another for occupants close to each other in an environment. This abstraction would

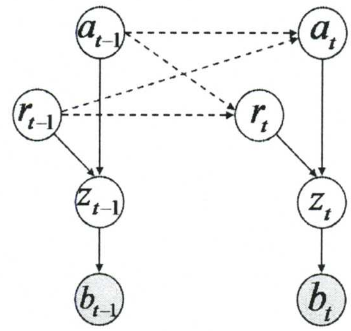


Fig. 3. A dynamic Bayes net describing tracking and activity recognition. Arcs indicate causal influences, with dotted arcs representing causality through time. Circles represent variables. Shaded variables are directly observable, the rest are hidden.

avoid the exponential blow up resulting from joint models of combinations of specific individuals. An interesting approach to this problem has been applied successfully to tracking multiple interacting ants in [32].

In most location estimation applications the state space $x \in X$ is the range of possible occupant locations. We integrate activity recognition into our model, so the state of an occupant m becomes $x_t^m =< r_t^m, a_t^m >$, where $r \in R$ denotes which **r**oom the occupant is in, and $a \in \{\text{MOVING,NOT MOVING}\}$ denotes occupant **a**ctivity. This node represents whether or not the occupant is moving; later work will integrate more activities. The dependencies involved are shown in a graphical model in Figure 3. The raw sensor values (in the non-hidden **b**$inary$ node) are the only given information; the rest must be inferred. These raw values are converted into four types of *events* and appear as observations $< e_t^1, e_t^2, ..., e_t^E >$ in the z_t node. 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.

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 [53]. The update equation is analagous to the backward portion of the forward-backward algorithm.

$$p(x_t|z_{t+1:T}) \propto \sum_{x_{t+1}} p(x_{t+1}|z_{t+2:T})p(x_{t+1}|x_t)p(z_{t+1}|x_{t+1}). \quad (3)$$

5.2 Data Association

Each sensor measurement z_t is composed of E binary valued events $z_t =< e_t^1, e_t^2, ..., e_t^E >$. The goal is to assign these events to the occupants that generated them. An assignment matrix θ_t of size $[E\text{x}M]$ indicates these assignments. For instance, $\theta_t(i,j)$ is 1 if event e_t^i belongs to occupant j. For each occupant $m \in M$ a new observation z_t^m is generated using the original measurement z_t and the assignment matrix θ_t. Each measurement z_t^m is used by a Bayes filter to update the posterior of occupant m.

In our domain data association can become a severe problem. In an environment with m occupants and k sensors, there are m^k possible assignments. There could easily be hundreds of cheap sensors monitoring several occupants, resulting in too many data assignments to enumerate (e.g., 5^{100}). Several classical approaches to data association exist in the multiple target tracking literature. In preliminary experiments we limited tracking to up to three occupants with around fifty sensors, and settled on a simple approach called the *nearest neighbor standard filter* (NNSF) [7]. NNSF uses only the closest observation to any given state to perform the measurement update step. Unfortunately, this approach requires exhaustive enumeration of every possible association. To cope, we use pruning and gating to eliminate less likely hypotheses. For instance, it is impossible for one occupant to set off sensors in more than one room during one time step. It also helped that in our experiments we limited the number of occupants to three, and multiple sensors were rarely triggered at the same time.

There are several classical data association methods, see [45] for a survey. Another method is the *joint probability data association* (JPDA). It estimates the states by a sum over all the association hypotheses weighted by the probabilities from the likelihood. The most general method is called *multi hypothesis tracking* (MHT), which calculates every possible association hypothesis over time. These approaches are feasible for room level tracking in a small environment for a few occupants, but become computationally intractable for simultaneous tracking and activity recognition with many sensors and people. Particle filters have been introduced as a solution to this problem, and in the next section we introduce a particle filter approach to data association adapted to our system.

5.3 Off-line Parameter Learning

The easiest way to train model parameters is to use information gathered when only a single occupant is home (as indicated by the RFID sensor). While a person is home alone we can assume that any sensor readings are generated by that person. In this way, models can be trained through simple counting. Note that this method, used in preliminary experiments, 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. We discuss solutions to these problems in the next section.

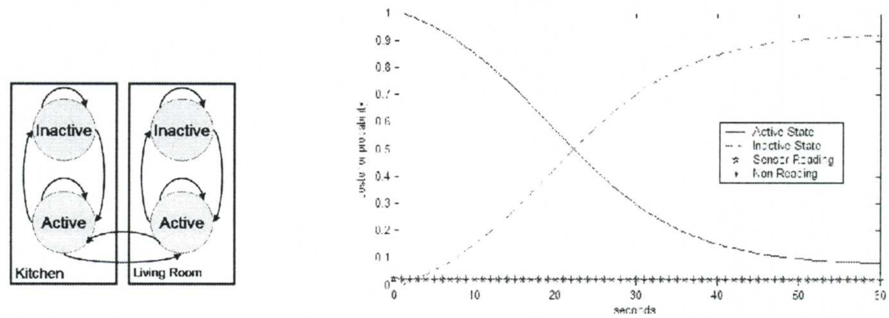


Fig. 4. (a) Active / inactive state representation. (b) The probability of being in an active state recedes as a full minute of sensor non-readings accumulate. The slope is learned for each room, and is more gradual for hallways and steeper for bedrooms.

Modeling the behavior of individual occupants can increase accuracy and reduce ambiguity. Individual behavior is represented in transitions between rooms and activities. These are the most important probabilities to learn, although each must be initialized generically.

- $p(r_t|r_{t-1}, a_{t-1})$ is the probability of transition to a room given the previous room and whether the occupant was moving or not. A generic set of transition probabilities is gathered by running a simple diagnostic exercise. First, each sensor is assigned to the correct room by hand. Next, a single occupant walks through the entire environment, manipulating every door and drawer and walking through every doorway. This provides the set of contiguous rooms. For occupants that are moving the transition probabilities between contiguous rooms are set uniformly. Static occupants can not transition to new rooms and probabilities are set much lower. See Figure 4a for an example transition between a kitchen and living room.
- $p(a_t|a_{t-1}, r_{t-1})$ models the probability that the occupant is moving given the previous room and whether the occupant was moving during the last time step. This models individual behavior: (1) each occupant has a different amount of activity in each room, (e.g., some people take shorter showers). (2) each room is associated with certain activities (e.g., hallways have more walking and bedrooms less). These probabilities are initialized uniformly. See Figure 4b for an example.

The perceptual model $p(z_t|x_t)$, can be broken down to $p(z_t|r_t, a_t)$. This models the probability of observing a set of sensor measurements given the location of the occupant and whether or not the occupant is moving. Thoughtful sensor choice and placement make this a straightforward calculation. In preliminary experiments these probabilities are fixed beforehand and are the same for every occupant. We incorporate a set of common-sense rules regarding (1) occupant location (e.g., must be in the same room as the sensor to trigger it), (2) sensor placement (e.g., occupant can not trigger both a contact switch and a pressure mat) and (3) occupant activity (e.g., must be moving to trigger a break beam sensor) into an ideal measurement.

Model Initialization:
1. Initialize model parameters with generic values.
2. Use data collected off-line to update the models of each occupant.

Tracking
1. Initialize the Bayes filter for occupants in range of the RFID reader.
2. Given an observation z_t use NNSF to find the best assignment matrix θ_t.
3. Use θ_t and z_t to generate z_t^m for $m \in M$.
4. Update each Bayes filter with each new observation z_t^m.

Repeat

Table 1. Bayes Filter approach.

See Table 1 for a summary of this approach.

5.4 Experiments - Real Data

Experiments were conducted using data generated by one to three occupants in an instrumented environment over the course of several months. The instrumented three story house contained twenty separate rooms, is 2824 square feet, and is home to two males, one female, a dog, and a cat. The house contained one RFID reader, twenty four motion detectors, and twenty four contact switches. In this experiment we did not use break beam sensors or pressure mats, although they are included in later simualted data. Detailed public information on the house can be referenced at the Allegheny County Real Estate web site [20].

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. The tracker used a motion model trained for the occupant being tracked with previously collected data. There were 1288 sensor readings from a 2 day period. Accuracy was 98.2%.

In the second experiment two occupants moved through the house according to a pre-planned script. The two occupants enter the front door thirty seconds apart and move throughout the house for fifteen minutes while following a script. At some point during this time each occupant visits every room in the house, including the other occupant's bedroom. After approximately fifteen minutes the two 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 tracker used two individual motion models for the two occupants. The data consists of 225 sensor readings, spanning thirty three minutes and fourteen seconds. 219 of 225 were classified correctly, corresponding to over 98% of the total running time of the experiment. The tracker was

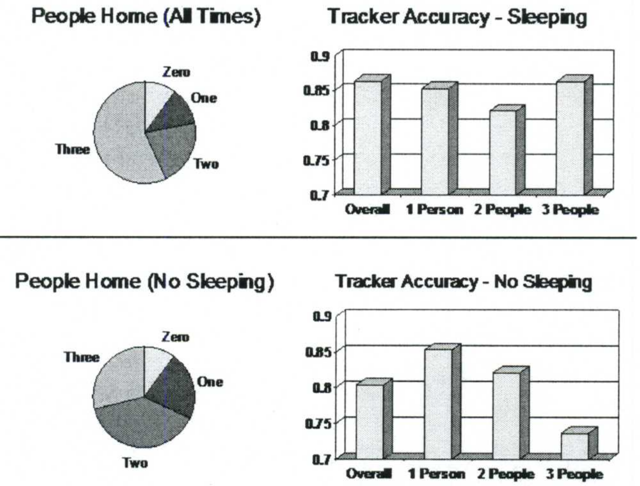


Fig. 5. Results of tracking experiment based on the number of occupants at home during experiment, with sleeping periods present and removed. Results do not include the time when no occupants were at home.

incorrect for 66 seconds. We scripted two ambiguous situations when two occupants share one anonymous sensor and then separate. In both instances, when the occupant arrived at his bedroom door the system recovered. In another experiment in which generic motion models were used one recovery was predicted correctly and the other not.

We measured tracker performance in the real world over a five day period for all occupants. The tracker used individual motion models for the three occupants. There were no guests during this period. To gauge performance we had to hand-label the data. To make hand-labeling feasible we gathered additional information from eight wireless keypads. The keypads have one button for each of the three occupants. During that week 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 the human labeler disambiguate the data stream and correctly label the movements and identity of each occupant. There were approximately 2000 sensor readings each day for a total of 10441 readings. 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. Note that each night every occupant slept in the house. 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 for 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 5.

6 Improvements

Particle filters can approximate the Bayes filter and solve tracking and data association problems. The key idea is to sample possible states and data associations via particle filtering and choose the one with maximum likelihood. The particle filter maintains a variety of hypotheses regarding the identity of each occupant, with each particle representing a hypothesis of occupant identities, locations, and activities. The data association problem can quickly become intractable as more occupants and sensors are added, so the approximation provided by particle filter methods becomes crucial.

6.1 Data Association

A particle filter approach to data association appeared in [10] for an over the horizon radar application. Particle filters use simulation to solve the data association problem of assigning anonymous sensor readings to the appropriate occupants. They can approximate a large range of probability distributions, unlike Kalman filters which are limited to Gaussian distributions. With resampling a sample set focuses on a narrow range of hypotheses, becoming increasingly computationally efficient with resources gathered on areas of state-space with high likelihood. The number of samples can be dynamically adjusted according to available computational resources, so that particle filters can be run in real-time.

At each time step a set of N weighted samples $S_t = \{s_t^j, w_t^j\}$ where $j = 1 \ldots N$ is generated from the current observation z_t and the set of samples from the previous time step S_{t-1}. Each sample $s_t^j =< x_t^j, \theta_{1:t}^j >$ is composed of the current state of all occupants and the history of assignments. Occupant location and identity are gathered by RFID upon entry and exit. During these time steps the assignment matrix θ_t is updated accordingly. When there is no ID sensor reading the sensors are assigned via sequential importance sampling with re-sampling [18]. The following steps generate a sample s_t^j:

- **Sampling**
  Sample a state x_{t-1}^i by drawing from the sample weights w_{t-1}^i of the previous time step. Recall that each state x_{t-1} is composed of the location and activity of M occupants $x_{t-1}^{(i)} = \{x_{t-1}^{(i),1}, \ldots, x_{t-1}^{(i),M}\}$.
  Use the sample x_{t-1}^i to sample a current state x_t^j from the motion model $p(x_t|x_{t-1})$. Recall that this motion model is calculated independently for each occupant (3).
  Sample a possible sensor assignment matrix θ_t^j from $p(\theta_t|\theta_{t-1})$.

- **Importance Sampling**
  Weight the sample s_t^j by the importance factor :

$$w_t^k = p(z_t|x_t^j, \theta_t^k) = \prod_{m \in M} p(z_t^m|x_t^{(j),m}) \tag{4}$$

This is the likelihood of the assigned observations z_t^m given the state of each independent occupant $x_t^{j,m}$.

- **Re-sampling**
  Multiply or discard samples by drawing samples with replacement according to the distribution defined through the importance weights w_t^j.

This procedure approximates the Bayes filter update (1) using a sample based representation [18].

6.2 On-line Parameter learning

This system uses a non-metric, room based location representation and a discrete set of mutually exclusive activities. The result is a relatively small number of discrete states, even when confounded with possible activities. This simplicity helps make *unsupervised* learning of model parameters possible. It also invites an intuitive understanding of how transitions occur between rooms.

In earlier experiments we trained parameters on data generated by occupants that were home alone. This is because multiple occupants introduce uncertainty that could detrimentally affect the accuracy of learned models. Training would be simple if we knew the true location and activity of each occupant. A common method to minimize this uncertainty is to use the Expectation-Maximization (EM) algorithm [11]. 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.

The EM algorithm is applicable because of the discrete number of possible locations and activities. A version of the EM algorithm called Monte Carlo EM [34, 52] takes advantage of the set of particles representing the posterior. A similar approach was used by [43] to learn the position of a traveler using GPS readings. 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 are normalized and then multiplied at the corresponding time slices. The learning algorithm is introduced thoroughly for Monte Carlo HMMs in [50].

$\alpha_t^m(r_t, a_t)$ 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 $t-1$.

A derivation [50] results in,

$$\gamma_{t-1}^{m}(r_t, r_{t-1}, a_{t-1}) \propto \alpha_{t-1}^{m}(r_{t-1}, a_{t-1}) p(r_t | r_{t-1}, a_{t-1}) \beta_t^{m}(r_t, a_{t-1}) \tag{5}$$

and

$$\delta_{t-1}^{m}(a_t, a_{t-1}, r_{t-1}) \propto \alpha_{t-1}^{m}(r_{t-1}, a_{t-1}) p(a_t | a_{t-1}, r_{t-1}) \beta_t^{m}(r_{t-1}, a_t) \tag{6}$$

We update parameters as:

$$p(r_t | r_{t-1}, a_{t-1}) = \prod_{m \in M} p(r_t^m | r_{t-1}^m, a_{t-1}^m) = \prod_{m \in M} \frac{\sum_{t=2}^{T} \gamma_{t-1}^{m}(r_t, r_{t-1}, a_{t-1})}{\sum_{t=2}^{T} \sum_{r_t \in Neighborofr_{t-1}} \gamma_{t-1}^{m}(r_t, r_{t-1}, a_{t-1})} \tag{7}$$

and

$$p(a_t | a_{t-1}, r_{t-1}) = \prod_{m \in M} p(a_t^m | a_{t-1}^m, r_{t-1}^m) = \prod_{m \in M} \frac{\sum_{t=2}^{T} \delta_{t-1}^{m}(a_t, a_{t-1}, r_{t-1})}{\sum_{t=2}^{T} \sum_{r_t \in \{MOVING, NOTMOVING\}} \gamma_{t-1}^{m}(a_t, a_{t-1}, r_{t-1})} \tag{8}$$

See Table 2. for each step.

6.3 Simulated Experiments

We have conducted a series of simulated experiments using particle filters for data association and online parameter learning. The simulator generates data from one motion detector, contact switch, and pressure mat per room, as well as break beam sensor data for doors between rooms. The number of rooms, occupants, and doorways can be specified via command line parameters.

Our experiments contained five rooms and from one to five occupants. Each occupant was initialized with a random transition model, although all occupants obeyed the contiguity of rooms. On average occupants were active (moving) 25% of the time. Each simulated occupant was 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. Every experiment was run on one hour of activity in the simulated environment. The tracker used two methods to train parameters, (1) learning off-line given one day of data generated by single occupants (OFFLINE), and (2) on-line via the Monte Carlo EM algorithm (MCEM). We also show results with and without data association (DA). When data association is turned off, every occupant receives every reading that is generated. Results are in Figure 6.

The results for one occupant are nearly the same between all methods. In this case data association has no impact because every reading already belongs to the same occupant. The motion models have nearly identical performance as well, because with only

Model Initialization:

1. Initialize model parameters with generic values.

E-step:

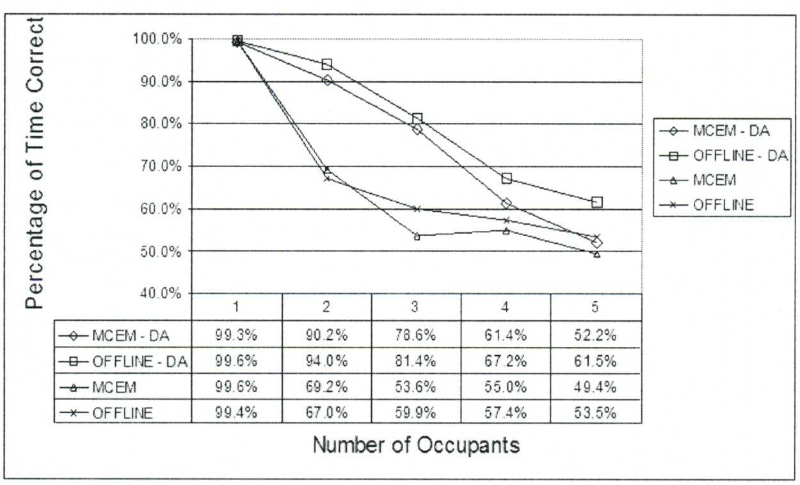
1. Generate N samples uniformly.
2. Forward filtering : for $t = 2...T$
(a) Generate N samples using the samples from the previous time step.
(b) Reweight each sample based on current observation z_t.
(c) Multiply or discard samples based on their weights.
(d) For each occupant m count and store $\alpha_t^m(r_t, a_t)$
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 2. Monte Carlo EM approach



	1	2	3	4	5
MCEM - DA	99.3%	90.2%	78.6%	61.4%	52.2%
OFFLINE - DA	99.6%	94.0%	81.4%	67.2%	61.5%
MCEM	99.6%	69.2%	53.6%	55.0%	49.4%
OFFLINE	99.4%	67.0%	59.9%	57.4%	53.5%

Fig. 6. Results of simulated experiments.

one occupant they need not be discriminative. For more than one occupant these factors become more important. The tracker that uses motion models trained off-line has the best performance. This is because it has the most accurate models, although in real experiments this data is harder and slower to obtain (e.g., we must wait until occupants are alone). Also, simulated occupants behave independently, which is not the case for real data. The online learning approach stays competitive with offline models, although accuracy falls off more as the number of occupants is increased. This is probably because the individual models become more difficult (and slower) to learn as interference between occupants increases. Removing data association entirely significantly lowers accuracy in all cases. The results are still better than we expected. Often, the problem with giving every sensor reading to every occupant is that inactive occupants will be influenced by the actions of other active occupants. In simulation the occupants were active and inactive about the same amount of time, which could make the effect of false positives less drastic. Also, because there were only five rooms, chance dictates that more occupants will more often share rooms, and in turn the sensors in those rooms.

7 Conclusion

This research fills an important gap in existing research in ubiquitous computing. It uses a sensory modality that has been largely ignored in favor of vision and audition to explore simultaneous recognition of location and activity. The data is generated from a permanent home setting, rather than a corporate setting or a temporary residential setting. This research exploits advances in home security, specifically sensors which are non-invasive, cheap, and easy to install and maintain, to introduce cost-effective automatic health monitoring. Specifically, automatic monitoring could be used to help our growing elderly population live independently longer.

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. 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.
5. 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.
6. 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.
7. Y. Bar-Shalom and T. Fortmann. Tracking and data association. *Mathematics in Science and Engineering. Academic Press*, 179, 1988.

8. 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.
9. 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.
10. N. Bergman and A. Doucet. Markov chain monte carlo data association for target tracking, 2000.
11. J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, 1997.
12. B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. Easyliving: Technologies for intelligent environments. In *Handheld and Ubiquitious Computing*, 2000.
13. 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.
14. 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.
15. C. Burkey. Environmental interfaces: Homelab, 2000.
16. B. Clarkson, N. Sawhney, and A. Pentland. Auditory context awareness via wearable computing. In *In the Proceedings of the Perceptual User Interfaces Workshop*, 1998.
17. L. Davis, K. Buckwalter, and L. Burgio. Measuring problem behaviors in dementia: Developing a methodological agenda. *Adv Nurs. Sci.*, 20(1):40–55, 1997.
18. A. Doucet, N. de Freitas, and N. Gordon. In *Sequential Monte Carlo in Practice*, 2001.
19. 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.
20. A. C. R. Estate. 2003.
21. 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.
22. 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.
23. 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.
24. 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.
25. L. Holmquist, J. Falk, and J. Wigstrm. Supporting group collaboration with interpersonal awareness devices. *Personal Technologies*, 3:13–21, 1999.
26. 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.
27. C. Hue, J. L. Cadre, and P. Perez. The (mr)mtpf: particle filters to track multiple targets using multiple receivers, 2001.
28. 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.

29. 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.
30. 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.
31. R. Karlsson and F. Gustafsson. Monte carlo data association for multiple target tracking, 2001.
32. 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.
33. 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.
34. R. A. Levine and G. Casella. Implementations of the monte carlo em algorithm. In *Journal of Computational and Graphical Statistics*, 2000.
35. J. McNeil. Disabilities affect one-fifth of all americans: Proportion could increase in coming decades. In *U.S. Census Brief*, 1997.
36. 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.
37. 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.
38. 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.
39. 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.
40. 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.
41. 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.
42. 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.
43. 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.
44. L. Rabiner. A tutorial on hidden markov models. *Proceedings of the IEEE*, 77:257–286, 1989.
45. 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.
46. 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.
47. H. Sidenbladh and M. Black. Learning image statistics for bayesian tracking. *In IEEE International Conference on Computer Vision (ICCV)*, 2:709–716, 2001.

48. T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. Picard, and A. Pentland. Augmented reality through wearable computing. *Presence*, 6(4):386–398, 1997.
49. 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.
50. S. Thrun and J. Langford. Monte carlo hidden markov models, 1997.
51. 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.
52. 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.
53. 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.
54. 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.
55. www.cms.org.