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

Daniel Wilson & Chris Atkeson

Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
[dan.wilson, cga]@cs.cmu.edu

Abstract. In this paper we introduce the simultaneous tracking and activity recognition (STAR) problem, which exploits the synergy between location and activity to provide the information necessary for automatic health monitoring. Automatic health monitoring can potentially help the elderly population live safely and independently in their own homes by providing key information to caregivers. Our goal is to perform accurate tracking and activity recognition for multiple people in a home environment. We observe a "bottom-up" approach that primarily uses information gathered by many minimally invasive sensors commonly found in home security systems. We describe a Rao-Blackwellised particle filter for room-level tracking, rudimentary activity recognition (i.e., whether or not an occupant is moving), and data association. We evaluate our approach with experiments in a simulated environment and in a real instrumented home.

1 Introduction

Advances in modern health care are helping millions of people live longer, healthier lives. As a result, people aged 65 and older are the fastest growing segment of the US population (set to double in the next two decades) [5]. Current health-care infrastructure is inadequate to meet the growing needs of an increasingly older population. Clearly, there is a growing need to develop technological solutions.

One solution is to use automatic health monitoring to enable *aging in place*, in which elders live independently and safely in their own homes for as long as possible – without transitioning to a care facility. Automatic health monitoring uses data from ubiquitious sensors to infer location and activity information about at-risk occupants. 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 [13]. Additionally, [30] 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 paper we introduce the simultaneous tracking and activity recognition (STAR) problem. The key idea is that people tracking can be improved by activity recognition and vice versa. Location and activity are the *context* for one another and knowledge of one is highly predictive of the other. We seek to provide the information that is vital for automatic health monitoring: identifying people, tracking people as they move, and knowing what activities people are engaged in. This research takes the first steps toward solving STAR by providing simultaneous room-level tracking and recognition of locomotion (which we loosely categorize as an activity). Please note that in this paper we do not attempt to provide tracking at higher than room-level granularity, and activity recognition is limited to whether or not an occupant is moving.

Automatic health monitoring necessarily occurs in a home environment where privacy, computational, and monetary constraints may be tight. We proceed from the "bottom-up," using predominantly anonymous, binary sensors that are minimally invasive, fast, and cheap. 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. These sensors can be found in existing home security systems and include: motion detectors, contact switches, break-beam sensors, and pressure mats.

We employ a particle filter approach that uses information collected by many simple sensors. Particle filters offer a sample-based approximation of probability densities that are too difficult to solve in closed form (e.g., tracking multiple occupants in a home environment via several hundred anonymous, binary sensors). Particle filters are desirable because they can approximate a large range of probability distributions, focus resources on promising hypotheses, and the number of samples can be adjusted to accommodate available computational resources. We show that a particle filter approach with simple sensors can tell us which rooms are occupied, count the occupants in a room, identify the occupants, track occupant movements, and recognize whether the occupants are moving or not.

This paper is organized as follows: In section 2 we discuss our rationale for choosing simple sensors. In section 3, we we introduce our approach, including the details of our learner. Section 4 contains experimental results from a real instrumented environment and from simulations. In section 5, we review existing instrumented facilities and discuss the state of the art in location estimation and activity recognition. We discuss our findings in section 6. In sections 7 and 8 we make our conclusions and acknowledgments, respectively.

2 Instrumenting the Home

In this section we describe which sensors we use and why. First, we discuss several challenges faced when placing sensors in a home and we then describe the ideal properties of sensors that would meet these criteria. Second, we list the sensors used in these experiments and illustrate how they work together.

2.1 Sensor Constraints & Issues

In fieldwork, we have found that 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. In particular, elderly individuals are often very sensitive to small changes in environment [10], and a target population of institutionalized Alzheimer's patients frequently strip themselves of clothing, including any wearable sensors [11]. As a result, there is a great potential for simple sensors to 1) "fill in the blanks" when more complex sensors can not be used and 2) to reduce the number complex (and possibly expensive) sensors that are necessary to solve a problem.

Like other researchers in academia and industry, we envision an off-the-shelf system installed and configured by a consumer [6, 7, 28]. Ideally, the sensors we choose should offer solutions to the following issues: sensors and monitoring systems should be *invisible* or should fit into *familiar* forms. Sensor data should be *private* and should not reveal sensitive information, especially identity. Arguably equally important – sensors should not be perceived as invasive. Sensors should be *inexpensive*, preferably available off-the shelf. Sensors should be *easy to install*. Wireless sensors can be mounted to any surface, while wired sensors may require running cable to a central location. Processing sensor data should require *minimal computational resources* (e.g., a desktop computer). Finaly, sensors should be *low-maintenance*, easy to replace and maintain. Sensors will be neglected and should be robust to damage. Sensors should be *low-power*, requiring no external power or able to 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.

2.2 Sensor choice.

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 [1]. Perceived privacy issues are minimalized by the fact that anonymous, binary sensors are already present in many homes as part of security systems. Binary sensors, which simply report a value of zero or one at each time step, satisfy computational constraints. These sensors are valuable to the home security industry because they are cheap, easy to install, computationally inexpensive, require minimal maintenance and supervision, and do not have to be worn or carried. 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.)

In this research, we chose four commonly available anonymous, binary sensors: motion detectors, break-beam sensors, pressure mats, and contact switches. These four sensors are selectively placed so that they are triggered by gross movement, point movement, gross manipulation, and point manipulation, respectively. In addition, we use an ID sensor to capture identity as occupants enter and leave the environment. In experiments, we substituted house keys with unique radio frequency identification (RFID) tags. Instead of a lock and key, an RFID reader near the doorway "listens" for the key (an RFID tag) and automatically records identification while it unlocks the door for a few seconds. (The RFID reader can detect multiple keys simultaneously from a distance of about a foot.) Afterwards, the door locks itself and the occupant need not continue to carry the key. See Figure 1 for an overview of a typically instrumented room.

[1] Our decision to use simple sensors provides inherent privacy at the physical layer, but does not directly address higher-level privacy issues, such as dissemination of information.

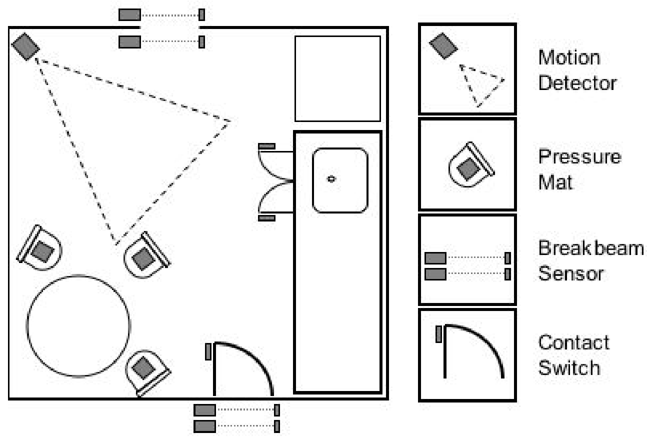

Fig. 1. Overview of typically instrumented room in which grey squares represent sensors.

3 Approach

In this section we introduce the STAR problem, discuss why it is difficult to solve with simple sensors, and consider several simplifications. We discuss a Bayes' filter approach and show why it fails to accomodate multiple occupants. We then describe a Rao-Blackwellised particle filter that is able to handle multiple occupants by performing efficient data association. Finally, we discuss how to learn model parameters both online and offline.

3.1 Simultaneous Tracking & Activity Recognition

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 (i.e., their activity and location). 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).

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. It is more desirable to employ many cheap sensors in lieu of expensive sensors. Second, we could *increase the sensor granularity.* The more sensors there are, the smaller the probability that multiple occupants will share the same anonymous sensor. In experiments, we intentionally

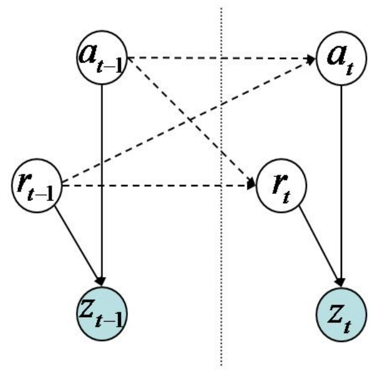

Fig. 2. A dynamic Bayes net describing room-level tracking and activity recognition. Arcs indicate causal influences, with dotted arcs representing causality through time. Circles represent variables. Shaded variables are directly observable, the rest are hidden.

placed sensors so that they would detect different properties, which increases granularity. For example, ceiling-mounted motion detectors detect gross movement while chair-mounted pressure mats detect static occupants. Similarly, noting which contact switches are out of reach of pressure mats can potentially separate 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., sleeping in their own beds).

3.2 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 [14]. 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. The update equation is analogous to the forward portion of the forward-backward algorithm used in hidden Markov models (HMMs). See [25] 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 3.4.

The graphical model in Figure 2 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:

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

This assumption could be partially relaxed 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 [19].

The Data Association Problem 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). However, this approach struggles to track multiple occupants because other occupants do not behave like noise processes. The tracker becomes 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. In a reasonable scenario with several dozen cheap sensors monitoring a handful of occupants for a week, there are too many data assignments to enumerate.

3.3 Particle Filter Approach

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 mea-

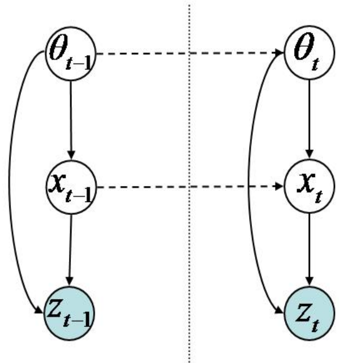

Fig. 3. A dynamic Bayes net describing tracking and activity recognition (combined into the state x) as well as data associations θ.

surements 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 3 for the updated graphical model. We must expand the posterior of Equation 1 to incorporate data association. We accommodate our expanded posterior efficiently by using a Rao-Blackwellised particle filter (RBPF) [14]. By the chain rule of probability,

$$p(X_{1:t}, \theta_{1:t}|z_{1:t}) = p(X_{1:t}|\theta_{1:t}, z_{1:t})p(\theta_{1:t}|z_{1:t}). \tag{3}$$

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.

$$s_t^j = \{x_t^{(j)}, \theta_{1:t}^{(j)}, w_t^{(j)}\}. \tag{4}$$

Note that for filtering purposes we may store only the latest association $\theta_t^{(j)}$. 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,

$$p(\theta_{1:t}|z_{1:t}) \approx \sum_{j=1}^{N} w_t^{(j)} \delta(\theta_{1:t}^{(j)}, \theta_{1:t}). \tag{5}$$

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,

$$\begin{aligned} p(X_t|z_{1:t}) &= \sum_{\theta_{1:t}} p(X_t|\theta_{1:t}, z_{1:t}) p(\theta_{1:t}|z_{1:t}) && (6) \\ &\approx \sum_{\theta_{1:t}} p(X_t|\theta_{1:t}, z_{1:t}) \sum_{j=1}^{N} w_t^{(j)} \delta(\theta_{1:t}^{(j)}, \theta_{1:t}) && (7) \\ &= \sum_{j=1}^{N} w_t^{(j)} p(X_t|\theta_{1:t}^{(j)}, z_{1:t}). && (8) \end{aligned}$$

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.

$$p(X_t = x|z_{1:t-1}, \theta_{1:t-1}^{(j)}) = \sum_{x'} p(X_t = x|X_{t-1} = x') p(X_{t-1} = x'|z_{1:t-1}, \theta_{1:t-1}^{(j)}). \tag{9}$$

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

$$\begin{aligned} p(X_t = x|z_{1:t}, \theta_{1:t}^{(j)}) &= \frac{p(z_t|X_t = x, \theta_t^{(j)}) p(X_t = x|z_{1:t-1}, \theta_{1:t-1}^{(j)})}{\sum_x p(z_t|X_t = x, \theta_t^{(j)}) p(X_t = x|z_{1:t-1}, \theta_{1:t}^{(j)})} && (10) \\ &\propto p(z_t|X_t = x, \theta_t^{(j)}) p(X_t = x|z_{1:t-1}, \theta_{1:t-1}^{(j)}). && (11) \end{aligned}$$

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 samples $s_t^{(j)}$ where $j = 1...N$) [14].

During *initialization* occupant location and identity are gathered by RFID and sensor measurements are assigned automatically. In each iteration there are four steps. First, during *re-sampling* we use the sample set from the previous time step S_{t-1} to draw with replacement a random sample $s_{t-1}^{(j)}$ according to the discrete distribution of the importance weights $w_{t-1}^{(j)}$. Next, we *sample* a possible sensor assignment matrix $\theta_t^{(j)}$. We discuss how to propose sensor assignments in the next section. Next, we use the association $\theta_t^{(j)}$ to perform an *analytical update* of the state of each occupant in sample j via Equation 12. Finally, during *importance sampling* we 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,

$$w_t^{(j)} = \eta \sum_x p(z_t | X_t = x, \theta_t^{(j)}) p(X_t = x | z_{1:t-1}, \theta_{1:t}^{(j)}), \tag{12}$$

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

The Data Association Problem During the sampling step a possible assignment of sensor readings to occupants (a data association) must be proposed for the new sample. Choosing an impossible association will cause that particle to have a zero weight and wastes computational time. For example, foolishly assigning two sensors from different rooms to the same occupant will result in a particle with negligible probability. A more efficient particle filter will propose data associations in areas of high likelihood. The better the proposals, the fewer particles necessary.

Assigning sensor readings uniformly (regardless of occupant state) 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, but a gated uniform method is still inefficient because it ignores the current state of each occupant. 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 and 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 assignment matrix θ_t is constructed by independently assigning each measurement to an occupant based on the probability that she triggered it $p(e_{it} | x_t) \forall i$. This method tends to choose likely assignments and usually avoids impossible assignments, but is not guaranteed to approximate the true distribution $p(\theta_t | z_{1:t})$.

3.4 Parameter Learning

Modeling the behavior of individual occupants can increase tracking and activity recognition accuracy and make data association more efficient. In a system with few ID sensors (like ours) these models are vital to disambiguate the identities of many occupants. Motion models describe individual tendencies to transition between rooms and activities. Sensor models describe individual tendencies to set off specific sensors (e.g., shorter occupants may use high cabinet doors less often). Models can be initialized generically for unknown occupants.

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

$$p(X_t = x_t | X_{t-1} = x_{t-1}) = p(a_t, r_t | a_{t-1}, r_{t-1}) \tag{13}$$

$$= p(a_t | a_{t-1}, r_{t-1}) p(r_t | r_{t-1}, a_{t-1}). \tag{14}$$

– $p(r_t|r_{t-1}, a_{t-1})$ 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:

$$p(z_t|X_t = x_t, \theta_t^{(j)}) = \prod_{m \in M} p(z_t|X_{mt} = x_{mt}, \theta_t^{(j)}) = \prod_{m \in M} \prod_i p(e_{it}|X_{mt} = x_{mt}, \theta_t^{(j)}). \tag{15}$$

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.

Training model parameters is simple when we know the true state of each occupant. When possible, we train parameters on data generated by occupants that are home alone. We assume that sensor readings are generated by that person or a noise process and use simple counting for parameter learning. This method ignores a significant amount of training data because occupants are often home together.

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, an iterative approach to finding parameters that maximize a posterior density [9]. We use a version of the EM algorithm called Monte Carlo EM [20, 31], which takes advantage of the set of particles representing the posterior. 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 [29]. An application of this technique appears in [23] for learning the motion models of people walking, riding the bus, and driving in cars.

4 Evaluation

In this section, we evaluate the performance of our approach on simulated and real datasets. In every experiment, location and activity predictions are updated every second and 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.

4.1 Simulated Data

We implemented a simple program to simulate the data generated by occupants in an instrumented environment [2]. 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 an independent first order HMM motion model that is set by hand or initialized randomly. Sensors also obey a hand-set sensor model in which the likelihood that a given sensor will trigger depends upon the number of occupants in the room and whether they are moving or not.

Simulated occupants are introduced to the environment from the same starting state and identified correctly from this state, to imitate an RFID set up in the entry way. Henceforth, each occupant is unlikely to re-enter this state. Unlike in reality, simulated occupants behave truly independently. Simulated occupants were active (moving) approximately 15% of the time. There was a sporadic sensor reading about once every ten minutes. The particle filter tracker used the same sensor model for each occupant. Parameters of motion models were either learned offline via counting, or online (i.e., during the experiment) via the EM Monte Carlo method.

Small house experiments. These experiments simulated a small house with ten rooms (three bedrooms, two bathrooms, a kitchen, living room, dining room, and hallways). Motion models for five occupants describe typical movements, 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. In Figure 4, 5, and 6 the variance bars reflect variations over the ten trials. Figure 4 summarizes our results.

Sensor configurations. First, we looked at the impact of sensor configurations on tracking accuracy (see Figure 4). In this experiment we tracked two occupants with generic motion models, using three different sensor configurations: 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 had so few sensors that the number of particles ceased to matter. Also, with fewer sensors come fewer measurements, and longer periods before tracker recovery. The number of particles will need to grow for sensor configurations with hundreds of sensors per room, which will pose a much more complex data association problem.

Parameter learning. Second, we examined how different approaches to model learning affect accuracy (see Figure 5). In this experiment, the number of particles is set

[2] The simulator can be downloaded from *www.danielhwilson.com*

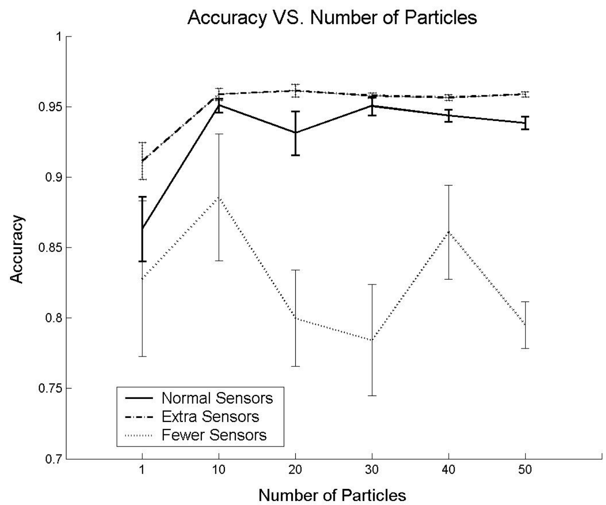

Fig. 4. Accuracy vs. number of particles for three simulated sensor infrastructures.

to fifty and we compare three techniques for learning motion model parameters. 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*), and (3) a combination in which the MCEM online parameter learning algorithm was seeded by a model already trained offline on one hour of single occupant data (*both*). In general, the *offline* method had highest accuracy, followed by *both* and with *online* learning last. Although the offline method performed best, this is in part due to the simplicity of our simulator, in which occupants behave independently. We feel that 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 online model learning will be confounded by multiple interfering occupants.

Number of occupants. See Figure 5 and Figure 6. In Figure 5 we varied the number of occupants and used fifty particles, and in Figure 6 we varied the number of particles and used offline model learning. 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 of particles. Accuracy does not drop linearly as more occupants are tracked simultaneously; the difference between one and two occupants is much less than the difference between three and four occupants.

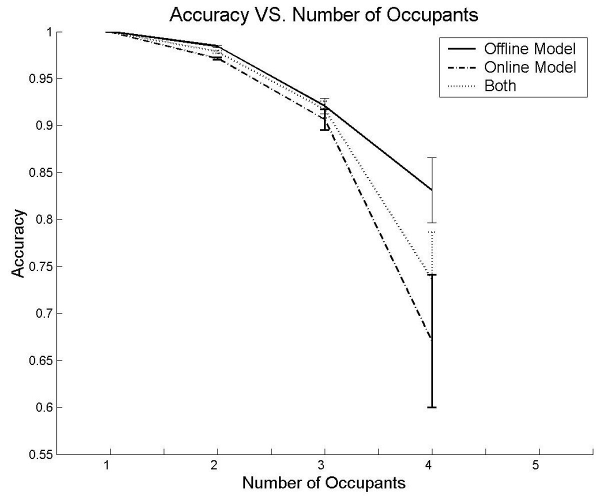

Fig. 5. Accuracy vs. number of occupants for three different parameter learning methods.

4.2 Real Data

We conducted experiments using data generated by one to three occupants in an instrumented environment. The instrumented three story house contains twenty separate rooms, is 2824 square feet, and was home to two males (including the first author), one female, a dog, and a cat. The house contained one RFID reader located in the front doorway (the back door was not used during the experiment). There were twenty four motion detectors, with at least one per room. Twenty four contact switches were distributed to every doorway, the refrigerator, and in many of the kitchen cabinets and drawers. In these experiments we did not use break beam sensors or pressure mats. Sensor and motion models were learned before the experiment began (offline) using several days worth of data from when each occupant was home alone.

A human hand-labeled the location of each occupant using information gathered by eight wireless keypads. During the experiment, 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. This process was unwieldy and has led our group to conduct further research concerning in-home data collection techniques [**?**].

Two person experiment. In order to understand how the tracker performs with occupants that are co-located versus occupants that are in different places, we scripted two ambiguous situations in which both occupants shared the same set of 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

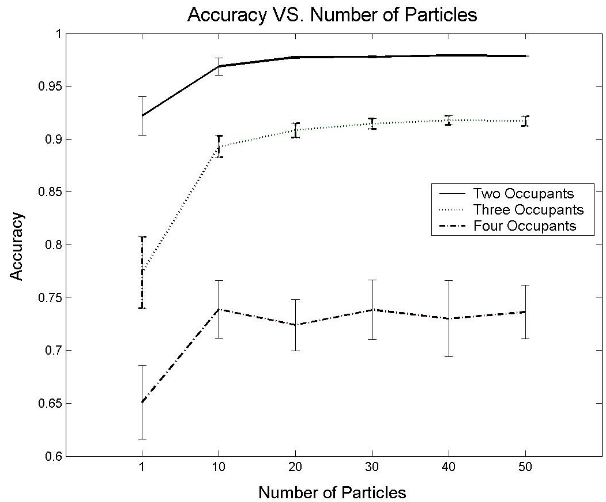

Fig. 6. Accuracy vs. number of particles for varying numbers of occupants.

returns to the living room. Next, the other occupant leaves to visit his own bedroom and then returns to the living room.

The tracker was accurate for over 98% of the thirty minute experiment. The bulk of the experiment was spent with occupants either moving separately (the first fifteen minutes), or co-located (meeting in the living room). We found near-perfect accuracy when 1) occupants were not co-located and had not recently shared the same sensors, and 2) when occupants were co-located. For example, it is easy to track two people while they watch television together. The difficulty arises when one or both occupants leave the room (the tracker must predict who left). There were two such ambiguous situations in this experiment, and in both cases the ambiguity was resolved as soon as the occupant reached his bedroom. In this case, the motion model contained information about who was more likely to visit a bedroom, and the tracker used it to recover identity. In a similar experiment using generic motion models, we found that one recovery was predicted correctly and the other not.

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. Two of the occupants shared a bedroom and one had a separate bedroom. Every occupant had a separate "study." The tracker used individual motion models for the three occupants. There were approximately 2000 sensor readings each day for a total of 10441 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. The system was quite good at tracking sleeping occupants (all three occupants were home together each night). Accuracy for three occupants drops to 73.7% when sleeping periods (all data between midnight and 8 AM) are removed.

5 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 automatic health monitoring, activity recognition, and people tracking.

Automatic health monitoring is being explored via several stand-alone instrumented laboratories, using a variety of sensors to approach a set of highly-interrelated problems, mostly a subset of location awareness and activity recognition. The Aware Home project at Georgia Tech has built a house and instrumented it with a variety of sensors with the goal of helping elderly adults live independently [2]. Researchers at the RERC Center on Aging at the University of Florida instrumented a house with ultrasound localization and displays to provide timely information to (possibly confused) occupants [16]. Other groups have instrumented actual health care facilities with a variety of complex sensors such as cameras and microphones for a variety of experiments [4]. There is also significant interest from industry; several existing products already advertise ADL monitoring systems that use sensors such as motion detectors [1].

An impressive amount of research falls under the umbrella of activity recognition. In particular, researchers have used GPS readings to infer walking, driving and bus riding behaviors [23], laser range finders to learn motion paths in a home [8], and combinations of audio and video to recognize behavior in an office environment [22] and interactions between individuals [12]. Recently, researchers at Intel Research Seattle have used scores of radio frequency identification tags to recognize dozens of ADLs [24]. Simple sensors are also being explored; several research groups have instrumented homes with binary sensors and collected the resulting data. At the Tokyo Medical and Dental University raw data was made available to physicians who were able to pick out patterns of activity by hand [21]. Researchers at the Medical Automation Research Center (MARC) at the University of Virginia clustered sensor readings into rough groups based on room, duration, and time of day and demonstrated that many of the clusters corresponded to ADLs [6]. Finally, researchers with the House_*n* project at MIT have developed their own version of generic, simple sensors which they deploy for weeks at a time, collecting data that is later used for off-line activity recognition [17, 27]. Clearly, simple sensors have solid potential for solving activity recognition problems in the home.

People tracking is a fundamental problem in ubiquitous computing and has also been approached via a variety of sensors, including cameras, laser range finders, wireless networks, RFID (Radio frequency identification) badges, and infrared or ultrasound badges [2, 3, 8, 12, 18, 15, 26]. 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 [15]. The range finders provided anonymous, high granularity coordinates while the badge system identified occupants. We also use a particle filter approach to solve the multi-target tracking 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 (in lieu of additional ID-sensors).

6 Discussion

We have shown that tracking multiple occupants in a home environment is feasible via a set of simple sensors. In summary:

– We found that highly predictive motion models improve accuracy, regardless of whether occupants behave similarly. In practice, the differences between motion models show up in private areas, like bedrooms and bathrooms, or during personal activities, like sitting in a favorite easy chair. The bigger these differences, the easier data association becomes and the more accuracy improves.
– Parameter learning is straightforward when an occupant is alone, however, occupants behave differently in groups. Learning models online can mitigate this discrepancy. In simulations, we found that the accuracy of models trained online falls as the number of occupants rises. One promising solution is to combine online and offline approaches.
– The number of particles required depends on the complexity of the data association problem. More particles are necessary for environments with many occupants and sensors. We found negligible accuracy improvements after twenty or so particles, even for up to five occupants. This number may change depending on the efficiency of the particle filter approach and the data association proposal scheme.
– More sensors will increase accuracy, regardless of the number of occupants. A low sensor density contributes to significant periods of time between readings (especially with only one occupant). During these "quiet" times no new information arrives to help the tracker recover from mistakes (such as the lag between entering a room and triggering a sensor). Motion detectors are the most active sensors, and a lack of them hurts accuracy the most.
– More occupants will decrease accuracy, particularly if parameter learning is performed completely online and motion models are generic. The accuracy suffers most when data association becomes difficult, i.e., immediately after co-located occupants separate. In general, accuracy is high for co-located occupants and for occupants who have not come into contact with other for some time.

7 Conclusion & Future Work

We have introduced the STAR problem and shown the potential of simple sensors for providing simultaneous location awareness and activity recognition. Automatic health monitoring ultimately requires recognition of complex ADLs, and we intend to incorporate new sensors and models as needed in order to meet this goal. New models must span households with different sensor configurations and use training data that can be collected and labeled easily and quickly by non-experts. More advanced DBNs, such as hidden semi-Markov models, could better incorporate time information. Models should also easily incorporate new sensors, including those that can directly detect certain activities. Additionally, we are interested in determining just how far simple sensors can go towards solving the STAR problem. Through extensive simulation and precise instrumentation of real environments, we plan to reveal how many and what type of sensors are necessary to solve increasingly complex location and activity recognition problems. Finally, we recognize that simple sensors are best used in conjunction with more complex sensors. For instance, an RFID gate placed in a crowded hallway could improve the performance of a system that relies mainly on many simple sensors. We intend to explore which additional sensors should be chosen and where they should be placed, in order to maximize accuracy when used in conjunction with a network of simple sensors.

8 Acknowledgments

We would like to thank our anonymous reviewers for their (many) valuable comments. Comments from Stephen Intille also helped improve this document. Thanks also to Jason Hong for a discussion on privacy. This material is based upon work supported by the National Science Foundation under Grant Numbers IIS-0312991 and DGE-0333420.

References

1. Living Independently - QuietCare system TM. www.livingindependently.com, 2005.
2. 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.
3. M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. 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*, 2003.
5. R. N. Anderson. Method for constructing complete annual US life tables, national center for health statistics. *Vital and Health Stat*, 2:129, 1999.
6. T. Barger, D. Brown, and M. Alwan. Health status monitoring through analysis of behavioral patterns. In *AI*IA 2003: Workshop on Ambient Intelligence*, 2003.
7. C. Beckman and S. Consolvo. Sensor configuration tool for end-users: Low-fidelity prototype evaluation # 1. Technical Report IRS-TR-03-009, Intel Research Seattle, July 2003.
8. M. Bennewitz, W. Burgard, and S. Thrun. Learning motion patterns of persons for mobile service robots. In *Proceedings of ICRA*, 2002.
9. J. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models. Technical Report ICSI-TR-97-021, University of Berkeley, 1997.
10. L. Burgio, K. Scilley, J. M. Hardin, and C. Hsu. Temporal patterns of disruptive vocalization in elderly nursing home residents. *Int. Journal of Geriatric Psychiatry*, 16(1):378–386, 2001.
11. 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.
12. B. Clarkson, N. Sawhney, and A. Pentland. Auditory context awareness via wearable computing. In *Proceedings of the Perceptual User Interfaces Workshop*, 1998.

13. L. Davis, K. Buckwalter, and L. Burgio. Measuring problem behaviors in dementia: Developing a methodological agenda. *Advances in Nursing Science*, 20(1):40–55, 1997.
14. A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag Telos, New York, 2001.
15. 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.
16. 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.
17. S. Intille and K. Larson. Designing and evaluating technology for independent aging in the home. In *Proceedings of the Int. Conference on Aging, Disability and Independence*, 2003.
18. T. Kanade, R. Collins, A. Lipton, P. Burt, and L. Wixson. Advances in cooperative multi-sensor video surveillance. *In Proceedings of DARPA Image Understanding Workshop*, 1:3–24, 1998.
19. 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 ICRA*, 2003.
20. R. A. Levine and G. Casella. Implementations of the Monte Carlo EM algorithm. *Journal of Computational and Graphical Statistics*, 2000.
21. M. Ogawa and T. Togawa. The concept of the home health monitoring. In *Proceedings of Enterprise Networking and Computing in Healthcare Industry. Healthcom 2003*, pages 71–73, 2003.
22. N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Fourth IEEE International Conference on Multimodal Interfaces*, pages 3–8, 2002.
23. D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *Proceedings of UBICOMP 2003*, October 2003.
24. M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing Magazine: Mobile & Ubiquitous Systems*, 3(4):50–57, October-December 2004.
25. L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
26. H. Sidenbladh and M. Black. Learning image statistics for bayesian tracking. *In IEEE International Conference on Computer Vision (ICCV)*, 2:709–716, 2001.
27. E. M. Tapia. Activity recognition in the home setting using simple and ubiquitous sensors. Master's thesis, Massachusetts Institute of Technology, September 2003.
28. E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home setting using simple and ubiquitous sensors. In *Proceedings of PERVASIVE 2004*, volume LNCS 3001, pages 158–175, 2004.
29. S. Thrun and J. Langford. Monte Carlo hidden markov models. Technical Report TR CMU-CS-98-179, Carnegie Mellon University, 1997.
30. 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.
31. G. Wei and M. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.