Doorjamb: Unobtrusive Room-level Tracking of People in Homes using Doorway Sensors

Timothy W. Hnat
twhnat@memphis.edu

Erin Griffiths
erg3wb@virginia.edu

Ray Dawson
rcd4@virginia.edu

Kamin Whitehouse
whitehouse@virginia.edu

Abstract

Indoor tracking systems will be an essential part of the home of the future, enabling location-aware and individually-tailored services. However, today there are no tracking solutions that are practical for "every day" use in the home. In this paper, we introduce the *Doorjamb* tracking system that uses ultrasonic range finders mounted above each doorway, pointed downward to sense people as they walk through the doorway. The system differentiates people by measuring their heights, infers their walking direction using signal processing, and identifies their room locations based on the sequence of doorways through which they pass. Doorjamb provides room-level tracking without requiring any user participation, wearable devices, privacy-intrusive sensors, or high-cost sensors. We create a proof-of-concept implementation and empirically evaluate Doorjamb with experiments that include over 3000 manually-recorded doorway crossings. Results indicate that the system can perform room-level tracking with 90% accuracy on average.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and Embedded Systems

General Terms

Design, Experimentation

Keywords

Tracking, Smart Homes, Sensor Networks

1 Introduction

In homes of the future, indoor tracking systems will be a foundation for intelligent home applications such as: elderly and patient monitoring, activity recognition, and occupancy-driven lighting, heating, and cooling. Tracking will allow these systems to provide location-aware and individually-tailored services, even in multi-person homes. However, today there are no tracking solutions that are practical for "every day" use in the home. Most indoor tracking systems require each person to carry or wear a powered device such as an infrared, ultrasonic, or RF transceiver. Even if the transceiver is built into a convenient form factor such as a cell phone or iBracelet [1], people are not likely to carry it at all times. Other systems rely on cameras, microphones, or other sensors that some people would consider to be privacy invasive. Systems that use active floors or object-use sensors require extensive and costly installation, and/or have poor tracking accuracy.

In this paper, we introduce the *Doorjamb* tracking system: a convenient, non-intrusive, and inexpensive room-level tracking system for homes. Doorjamb uses ultrasonic range finders mounted above each doorway, pointed downward to measure the distance to the person as he or she walks between rooms. It estimates the height of a person by measuring the distance to the top of the head when the person is directly under the door frame and walking direction based on its ability to detect the person on either side of the door frame before and after walking through it. The system differentiates people based on their heights, and identifies room locations based on their walking direction and the sequence of doorways through which they pass. Doorjamb tracking will be an enabling technology because it provides room-level tracking without requiring any user participation, wearable devices, privacy-intrusive sensors, or high-cost sensors. Furthermore, the doorway sensor is in a molded plastic enclosure that mounts easily and discretely behind the door jamb. Doorjamb builds on results from an earlier paper that first proposed using height for non-intrusive biometric identification [2], and goes beyond those results by using ultrasonic sensors to infer walking direction, and by performing multi-person tracking. The principles underlying Doorjamb's design can easily be extended to include other passive sensors installed in doorways, such as weight sensors, motion sensors, color sensors, or any other sensors that can detect when somebody crossed the threshold and can provide some indication about the person's identity.

Doorjamb is a multi-target tracking system and, as such, the main challenge is *data association*: deciding which target is associated with each doorway crossing event. The process of measuring the height and direction of a moving person is subject to many causes of noise: a person's posture, multi-path reflections, and the natural undulation of gait, among

SenSys'12, November 6–9, 2012, Toronto, ON, Canada.

many others. Therefore, height and direction alone can be ambiguous and are not sufficient to identify who is walking through a doorway and into which room. The basic insight behind Doorjamb tracking is that a person cannot walk through an arbitrary sequence of doorways because the path is constrained by the floor plan of the home: *people cannot move through walls or teleport between rooms*. Doorjamb uses this insight to resolve height and direction ambiguities by choosing a path for each person that only includes legal room transitions. Unfortunately, a person's true path can also be ambiguous due to *false detections* and *missed detections*: reports of doorway crossings that did not actually happen, or the failure to report doorway crossings that did happen. Both types of detection errors can cause the appearance of discontinuous paths and teleportation. Therefore, Doorjamb considers sequences of multiple doorway crossings and performs a joint optimization that assigns a path to each person while (i) ensuring that each doorway crossing is assigned to only one person, (ii) maximizing consistency between the doorway assignments and the height and direction estimates, and (iii) minimizing the number of false detections or missed detections in each path.

In this paper, we present Doorjamb's hardware design, signal processing algorithms, and tracking algorithm. We evaluate Doorjamb tracking with controlled experiments in a home that contains 8 rooms and 7 doorways. Three test subjects with different heights walked through the home in pairs of two while collecting the exact timestamps, doorways, and identities of every doorway crossing. A total of over 3,000 doorway crossings were generated for analysis. Our results indicate that height measurements can vary by up to 15 *cm* from the true height, which is larger than that the difference in heights of our test subjects. Furthermore, direction estimates are only 81% accurate. Finally, our doorway sensors produce false detections at a rate of 7% and fail to detect true doorway crossings at a rate of 4%. Nevertheless, Doorjamb can perform room-level tracking with 90% accuracy on average, and accuracy improves as the difference in heights between people increases. In comparison, the Motetrack system achieved a 37% room-level tracking accuracy on the same set of experiments. Doorjamb tracking is computationally efficient: it only requires enough memory to store approximately 20-30 tracks for consideration, and can process our entire 3000-doorway experiment in 280 seconds. This indicates that the algorithm can be used on resource-limited embedded platforms such as an ARM processor.

The contributions of this paper include:

- A detailed presentation of design considerations for doorway sensors, including how to choose the beam angle, number, and position of ultrasonic range finders mounted on doorways. We also provide lessons learned from early prototypes.
- An in-depth analysis of the sources of height errors, direction errors, false detections, and missed detections when using ultrasonic range finders in doorways. We also present signal processing algorithms to overcome or ameliorate these errors.
- An analysis of the data association challenges and opportunities when using doorway sensors for tracking. We present the mathematical formulation of a tracking algorithm that exploits these opportunities, including a 4-step merging algorithm that enables efficient operation without compromising tracking accuracy.
- A proof-of-concept implementation, deployment, and empirical evaluation of Doorjamb tracking that reveals its strengths, limitations, and possibilities for future improvement.

2 Background and Related Work

Systems that track person movement in buildings can be roughly divided into three categories: (1) tracking devices, (2) vision or audio systems, or (3) non-intrusive tracking.

Most tracking systems require the people being tracked to carry a physical device, such as a GPS unit, a cell phone or an RFID tag. For example, the Active badges [3] system uses wearable badges with infrared transmitters that could localize people indoors, Cricket uses ultrasound transceivers [4] and RADAR, ActiveBats, and Motetrack require radio transceivers [5, 6, 7]. Other systems use wearable RFID readers [8] or tags on the body [9, 10] and can use signal strength or varying transmission power levels to localize these tags in space. These systems can track people in office or industrial environments, but wearable badges and tags are less like to be accepted in the home, particularly if they are battery powered and require charging. To address these concerns, several systems track people using cellular phones, which many people charge frequently and carry on their person anyway. These systems have been shown to achieve 2.5-5 *m* accuracy [11, 12]. However, people do not always carry their cellular phones in the home, particularly in the early morning, late at night, after showers, or shortly after returning from work [13]. These can be key times in the day for medical monitoring or energy management applications that want to monitor human activity.

People can be tracked in buildings without wearable tags or badges by using cameras, microphones, or other information-rich sensors. For example, one study uses ceiling mounted cameras to track the precise, long-term movement patterns of people in a single room [14], and other systems can identify people based on facial recognition [15, 16], gait analysis [17, 18, 19], or a combination of the two [17]. Microsoft's Kinect device uses a combination of infrared and video to infer the exact location and pose of a person in the room. Alternatively, iris patterns provide a unique texture that can be used for identification purposes [20, 21] and can be read without user interaction as a person moves about a space. However, several challenges arise when using a video-based localization system. Cameras require line-of-sight vision to the object being localized and do not operate in the dark. Thermographic cameras operate in the dark but are extremely expensive. Furthermore, many people would object to any type of cameras being placed in their home for fear of the privacy invasion, knowing that even trusted camera systems can be hacked.

Non-intrusive tracking systems can localize people without requiring tracking devices, cameras, or microphones. For example, ON-OFF contact switches on appliances, lights,

Figure 1. The doorway sensor is discretely mounted behind the door jamb in a molded plastic enclosure.

and water fixtures can be used to infer the location of a person [22, 23]. Motion sensors are typically used to detect simple room occupancy, but can also be used for tracking by using the intersection of different viewing angles to triangulate the position of a subject [24]. Radar [25] and device free localization (DFL) systems [26] can locate people by analyzing the reflection or attenuation of RF signals caused by the human body. However, these non-intrusive tracking systems cannot detect the identity the people being tracked: they can detect that a person is present, but cannot differentiate between different individuals. One exception are smart floor systems that track people based on their weights [27, 28, 29]. Weight is a weak biometric and has similar limitations to those of Doorjamb: it can differentiate people most of the time, but can easily be confused by changes in gait or carried objects. However, smart floors are more suitable to offices with raised floors than to homes, where it would be expensive to install an array of load cells throughout the floor, particularly as a retrofit in existing homes. In contrast, Doorjamb uses inexpensive and low power ultrasonic range finders that can be easily and discretely mounted behind the door jamb. The total cost for the Doorjamb hardware could be as low as \$20 USD per doorway at production scale quantities.

Doorjamb builds on an earlier paper by another group that first proposed using height for non-intrusive biometric identification [2]. This paper uses controlled and in-situ experiments to characterize the measurement errors when measuring the height of a moving person. Then, based on analysis of the heights of people in 2044 multi-person households, the authors conclude that height sensing can differentiate people in 85 percent of households. The authors note that the system could be extended to 95 percent of households if measurements could be averaged as each person walks through multiple doorways. However, the paper does not solve the data association problem, and this result is merely an upper bound on multi-person tracking accuracy. Doorjamb extends over this work by using range finders to infer walking direction and by developing new algorithms to track multiple people despite measurement noise, false positive, and false negatives.

3 Doorway Sensors

The primary goal of the Doorjamb hardware platform is to measure the heights and directions of people as they walk through a doorway. Secondary goals of the platform include cost, energy consumption, and a discrete and non-obtrusive aesthetic. In order to be cost effective for smart home applications such as energy management, the system should cost less than a few hundred dollars per home. A typical home has 10-15 doorways including exterior doorways, so the components of the system should total \$20-30 at production scale. The devices must also be able to operate for multiple years on battery power: although power is abundant in homes, it is typically not available above the doorways. Running wires from the top of the doorway would increase installation cost and cause a snagging hazard. Additionally, the sensors will be placed in a highly visible part of the home and must have a discrete design and a form factor that can be easily mounted.

3.1 Hardware Design and Operation

Doorjamb doorway sensors use ultrasonic range finding sensors that are mounted to the top of a doorway and aimed down toward the floor. When the doorway is empty, the sensors report the distance to the floor d_f. This baseline value indicates the height of the doorway, so in principle the sensors can be installed on doorways of various heights without requiring manual calibration. In practice, however, the baseline measurement is not always of the floor and we manually verify the distance d_f.

To detect a person's height, the ranging module tries to measure the distance to the top of the head d_p as a person walks through the doorway. The height of the person h can be calculated as the difference in these two values: $h = d_f - d_p$. The height estimate is most accurate if taken when the person is directly under the door frame. Typically, this corresponds to the tallest height observed, since the value d_p becomes larger as the person moves away from the door frame.

To detect a person's walking direction, the ranging module is angled into one room more than the other. This tilt causes an asymmetry in the sensing region, which changes the shape of the curve of d_p values as the person walks through the door. If the person walks in the same direction as the tilt, the system will detect the tallest heights first, followed by shorter heights. If the person walks opposite the direction of the tilt, the system will detect shorter heights followed by taller heights.

Our final doorway sensor design consists of five integrated components: (1) two to four Parallax PING ultrasonic range finders, (2) passive infrared sensors facing to each side to sense motion activity in the two adjacent rooms, (3) magnetic reed sensors to sense whether the door is open or closed, (4) a custom-designed power module to supply regulated 3V and 5V power to the devices, and (5) a Synapse Wireless SnapPY RF100 module, which consists of a Python programmable freescale processor and a 802.15.4 radio. The components are placed into a molded plastic enclosure that mounts discretely behind the door jamb. An image of our doorway sensor when installed in a doorway is shown in Figure 1.

Platform	Rate	Accuracy	Cost	Power (watts)	Range
Kinect™	30 Hz	1-2 *cm*	$150	15 *W*	120-335 *cm*
LIDAR	100+ Hz	2 *cm*	$2,000-$75,000	60+ *W*	3-120 *m*
Go!Motion	20 Hz	1-2 *cm*	$100	250 *mW*	15-600 *cm*
Maxbotix	10 Hz	1-2 *cm*	$120	150 *mW*	20-700 *cm*
PING	50-500 Hz	1 *cm*	$95	150-400 *mW*	2-300 *cm*

Table 1. Evaluation of a spectrum of height sensing solutions along five variables (sampling rate, accuracy, cost, power, and range). Alternative solutions have at least one metric that is not as good as the final solution, designated as PING.

3.2 Achieving Doorway Coverage

The main challenge for doorway sensor design is to operate under a wide range of doorway sizes, person sizes, and walking speeds while still achieving high height and direction accuracy, few false detections, and few missed detections. Doorway sensors must provide 1 *cm* resolution for people with heights ranging from 151 *cm* (5 ft) to 189 *cm* (6 ft, 2 in)[1], at walking at speeds of up to 3 m/sec[2], in doorways that range from 90-300 *cm* (3-10 ft) wide and 213-275 *cm* (7-9 ft) tall. To detect a person, the sensing region must cover the entire width of the doorway and the sensor must sample quickly enough to ensure at least one reading to the top of a person's head when the person is directly under the doorway. The sensing region must also extend outside the door frame in order to detect the direction from which the user enters or exits the doorway. At the same time, the sensing region must not extend too far from the door frame to avoid erroneous measurements such as a person walking by the doorway but not through it.

Because ultrasound sensors have a conical beam angle, it is most challenging to achieve adequate doorway coverage when a tall person walks through a short doorway: the sensing region is smaller at close range, so the person is more likely to walk to the side of the sensing region, or to walk through the sensing region too quickly to be measured. In our hardware design, we use the Parallax PING ultrasonic ranging modules that have a 40 degree beam angle, a minimum range of 2 *cm* (0.79 inches), and a maximum range of 300 *cm* (9 feet, 10 inches). When the tallest person we support walks through the shortest doorway we support, the gap between the head and doorway sensor is only 24 *cm* (10 in). At that distance, the 40 degree cone width of the PING produces a sensing diameter of 17 *cm* (6.7 in). At a speed of 3 m/sec, a head that is 15 *cm* (6 in) in diameter will pass through this sensing region in about 100 milliseconds, so the ranging module must take measurements at 10 Hz or higher to ensure at least one reading before the person passes out of the detection region, and a sampling frequency of 30-40 Hz is necessary to produce an accurate and high confidence detection event. The PING module samples at a nominal 50 Hz, but our doorway sensors only trigger one module at a time to reduce multi-path interference, which can cause a taller measurement than reality. Therefore, a person walking under only one sensor may only be measured at 25 Hz, producing only 2-3 readings before the person exits the sensing region.

[1] The 5th and 95th percentiles of adult human height [30].

[2] More than twice the average adult walking speed [31].

At short range, the PING's sensing diameter of 17 *cm* (6.7 in) is small compared to a typical doorway width of 90 *cm* (3 feet), and so multiple sensors must be used per doorway. This increases the potential for multi-path interference between modules, and also adds to the financial cost and the energy consumption of the doorway sensors. However, the head itself has a radius of about 7 *cm* (3 in), and so a person can be detected even if the center of the head is slightly outside the PING's sensing diameter. Therefore, the total detection diameter for a single sensor is approximately 30 *cm* (12 in), and two sensors would adequately cover 60 *cm* (24 in). Furthermore, the head is highly unlikely to pass through the outermost 15 *cm* (6 in) on either edge of the doorway because it would nearly hit the side of the doorway due to its 7 *cm* (3 in) radius. Therefore, a doorway width of 90 *cm* (3 ft) can be adequately instrumented with only two sensors, even in the shortest of doorways with the tallest of people.

In high door frames, the detection region becomes larger so even fewer sensors are required for full coverage of a doorways. However, a different set of problems are encountered. When the shortest person we support walks through the highest doorway we support, the gap between the head and the doorway sensor is 124 *cm* (4 ft, 1 in), which translates to a detection region with diameter of 83 *cm* (6ft, 11 in). This large detection region extends outside the door frame and into the neighboring rooms, thereby increasing the chance of false detections. In very wide doorways, on the other hand, the sampling rate is limited because only one ranging module is used at a time. One doorway that we instrumented was 300 *cm* wide by 275 *cm* high and required 4 range finders to cover the full span. This reduced the nominal sampling rate from 50 Hz to 12.5 Hz per sensor. A short and wide doorway is the worst case for sampling frequency: a doorway that is 300 *cm* wide and 213 *cm* would require 9 ranging modules, reducing the nominal sampling frequency to 5 Hz per sensor. At 3 m/sec, a person will move 60 *cm* between measurements and could easily pass through the ranging module's 17 *cm* sensing region without being detected. Thus, our doorway sensor design cannot support doorways that are both wide and short.

3.3 Early Prototypes and Lessons Learned

Doorjamb can operate with any ranging system, and we implemented or considered several alternatives to the PING ultrasonic range finder but found them inadequate for the needs of Doorjamb's doorway sensing. With further development, many of these technologies could be equal to or better than the ultrasonic solution we used, but not in the forms currently available. In this section, we summarize our findings and explain the shortcomings of each technology, in comparison to our final solution. A summary of the technologies can be found in Table 1.

We first considered high-end sensors that would provide very high accuracy for a prototype version of Doorjamb, even if not practical for long-term or large-scale deployments due to cost, bandwidth, or energy consumption. The Microsoft Kinect platform is a well known infrared and vision-based system that can accurately detect human location and position, but can also be used simply for range finding with accuracy of 1-2 *cm*. It has a low cost of about $150 but cannot detect objects closer than about 120 *cm*, making it impractical to mount above doorways. Laser range finders also provide very high accuracy but only measure distance at a single point, making it difficult to adequately cover a large doorway. LIDAR systems address this problem by pointing the laser in different directions to create a point cloud of distance estimates, and a 2D line scanner would work well in a doorway. However, these systems cost at least $2000 each and are cost prohibitive even for a prototype deployment in a home with 10-15 doorways.

We tried three different ultrasound ranging systems. The Go!Motion devices used in [2] use electrostatic ultrasonic range sensors that provide very high accuracy, but produce an audible click when taking measurements. This system could not be tolerated for long-term experiments in homes and needed to be turned off at night even for short duration experiments. Furthermore, the narrow 20 degree beam angle entails that many sensors must be deployed for full coverage of each doorway increasing cost and multi-path interference. Combined with a 20 Hz sampling rate, it also leads to missed detections when people walk quickly. The Maxbotics XL-EZ0 ultrasound ranging module uses a quiet, piezoelectric sensor. The low 10 Hz sampling frequency was compensated for by a wide 60 degree beam angle, so people could not pass through a doorway completely undetected. However, doorway crossings would often produce only one measurement, so it could not measure direction. Furthermore, the one reading was often taken when the person had not yet fully entered the door frame, and was therefore not necessarily an accurate height measurement. Additionally, the wide beam angle causes frequent false alarms when people walk near to but not directly under a doorway. Finally, the wide beam angle causes large multi-path errors because the acoustic signal bounces between doorways, even when across the room from each other. Low-power ultrasound ranging systems typically have difficulty measuring distances to the human head because hair and skin partially absorb the signal and reduce the effective range. Ultimately, we used the PING ultrasound range finder because the high sampling frequency would reduce missed detections and the cone width of 40 degrees was small enough not to cause inter-door interference yet wide enough not to require too many sensors to be deployed per doorway. As we discuss in the next section, the 40 degree cone width was wide enough to cause noise due to acoustic reflections from nearby objects such as door trim, hinges, and nearby furniture, and this became a challenge for our signal processing algorithms.

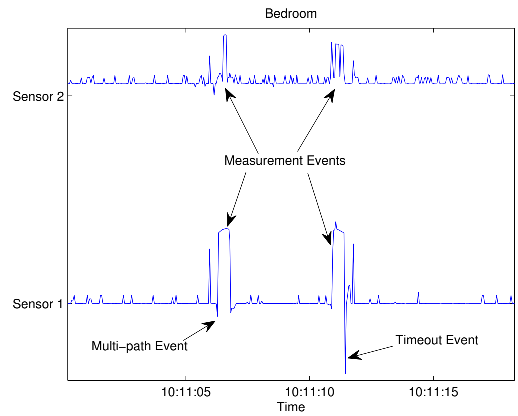

Figure 2. A person in the doorway can cause three different responses from the range finder: (1) Timeout events, (2) Multi-path events, and (3) Measurement events.

4 Signal Processing

A doorway sensor with k range finders generates a k-valued column vector X, where each element X_i at time t is a continuous stream of t height values produced by the ith range finder: $X_i = x_i^1, x_i^2, x_i^3, \ldots x_i^{t-1}, x_i^t$. The goal of Doorjamb's signal processing component is to convert this vector X into a single set of doorway events D where $d_j \in D = (t_j, h_j, v_j)$: a 3-tuple containing the timestamp of a doorway crossing t_j and the corresponding height and direction measurements h_j and v_j. Thus, Doorjamb's signal processing component fuses data from multiple range finders in a single doorway, but does not fuse data from multiple doorways.

Doorjamb uses four main signal processing algorithms. The first algorithm segments the continuous data streams into discrete doorway crossing events (Section 4.1). The second algorithm filters doorway events that were likely to be caused by noise (Section 4.2). Given a final set doorway events, the second and third algorithms estimate the height and walking direction of the person who triggered the event (Sections 4.3 and 4.4).

4.1 Doorway Crossing Detection

When a person walks under a PING range finder, one of three things will happen. First, if the sensor has already transmitted an ultrasonic pulse and is waiting for the response, the person may block the acoustic energy from reflecting back to the transceiver. In this case, the PING waits at most 18.5 msec before it times out, cancels the ranging process, and transmits another ultrasonic pulse. We call this a *"timeout"* event, and it can be detected because the PING modules output a consistent max-value ranging estimate of 285 *cm*. This distance is typically much farther than the distance to the floor, so the resulting height estimate $h = d_f - d_p$ is a large negative value. Second, the person sometimes blocks the line-of-sight path between the range finder and the floor, but acoustic energy still reaches the transceiver through a multi-path reflection. For example, the energy may go around the person after reflecting off the floor by bouncing off the side of the doorway and back to the transceiver.

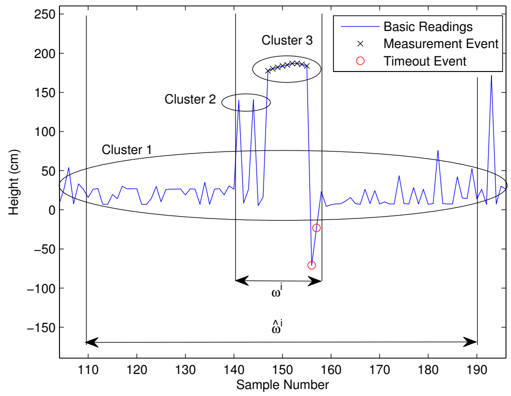

Figure 3. The noise filtering algorithm removes baseline values from window ω^i if they cluster with values from window $\hat{\omega}^i$.

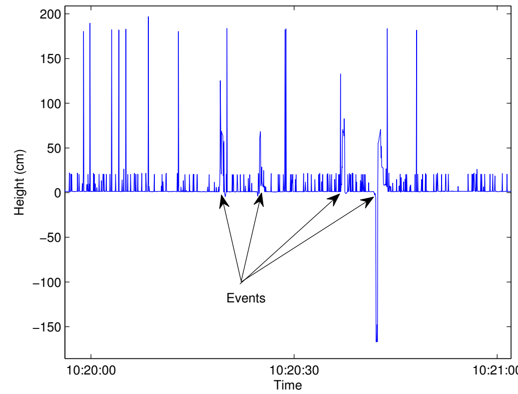

Figure 4. Nearby objects reflect acoustic energy and cause random noise at about 170-180 cm, similar to the heights of two of our test subjects.

We call this a *multi-path* event, and it can be detected by a small negative height estimate. Third, if the sensor has not yet transmitted an ultrasonic pulse when the person enters the doorway, the next pulse will reflect off the person and return to the transceiver. An example of each of these three types of doorway crossing events is depicted in Figure 2.

We detect doorway crossings by scanning all data streams in X for any timeout, multi-path or measurement event. In any such event is found, we create a *detection event* Y that contains all measurements near the event from all streams in X. More formally, we first search for any height estimates that are negative, or in the range 137-198 cm (4 ft, 6 in - 6 ft, 6in). An example of such values are depicted by x's and o's in Figure 3. The timestamps of all such height estimates are clustered using the DBSCAN clustering algorithm, resulting in a set of timestamp clusters C (the timestamp clusters are not shown in Figure 3; the clusters shown in that figure are height clusters, as discussed in the next section.). Each timestamp cluster $c_i \in C$ is a set of one or more event timestamps within 400 msecs of each other, denoted $c_i = c_i^1, c_i^2, \ldots$. A time window ω_i is defined around each cluster using the range $\omega_i = [min(c_i) - 200msec, max(c_i) + 200msec]$, and a new detection event $e_i \in E$ is defined to be all sensor data from that range: $e_i = X_{\omega_i}$. A window ω_i is illustrated by the inner bars in Figure 3: the event e_i includes all values within that window.

4.2 Noise Filtering

Ultrasonic range finders can be very noisy and a single outlier value will cause the algorithm above to produce a false detection. We reduce the effect of outliers by ensuring that every detection event has at least two values that are substantially different from the baseline measurements. More formally, we define a 200 msec region immediately before and after detection event e_i, and we remove any values in e_i that are similar to the values in that window. Any outlier detection algorithm could be used here, but we found a simple clustering algorithm to work well: we define a second time window that extends before and after each detection event by 200 msec: $\hat{\omega}_i = [min(e_i) - 200msec, max(e_i) + 200msec]$. This 400 msec time window will not merge events from two consecutive doorway crossings unless people are walking unusually close to each other. A window $\hat{\omega}_i$ is illustrated by the outer bars in Figure 3. We cluster all sensor values in $\hat{\omega}_i$, and remove all values from e_i that are clustered together with values in the region $\hat{\omega}_i - \omega_i$. In the example shown in Figure 3, the data in window $\hat{\omega}_i$ produces three height clusters: 1, 2, and 3. Only cluster 1 includes values in window $\hat{\omega}_i$ but not in window ω_i. Therefore, any values in cluster 1 are removed from e_i, thereby removing all baseline noise values and leaving only the measurement events (x's), the timeout events (o's) and the two values in cluster 2 remaining in e_i. If detection event e_i has fewer then 2 values after removing the baseline values, then it is eliminated.

In addition to random noise, the ultrasonic range finders are also subject to periodic noise due to reflections from nearby environmental features, such as door trim, a hinge, or a nearby shelf. Figure 4 depicts data from one doorway sensor that frequently picked up reflections from a nearby object and produced height measurements similar to a true person's height. The key feature of these environmental features is that they are consistently the same value and are never the negative height estimates cause by a timeout or multi-path event. Therefore, we filter these values by creating a third window that extends 30 seconds on either side of the detection event e_i. Any height measurement in e_i that is positive and identical to a measurement in this one minute window is removed from the event. Doorway events typically include a wide range of noisy values as a person enters and exits a doorway, so this filtering process would not affect most values in two doorway events that occur within 1 minute of each other.

4.3 Height Estimation

Once the detection events have been created, we estimate the height of the person that created each event. The best estimate of a person's height is typically the maximum height measurement (i.e. the shortest distance measurement d_p) within the event because it is taken when the person is closest to the doorway: as a person moves away from the doorway,

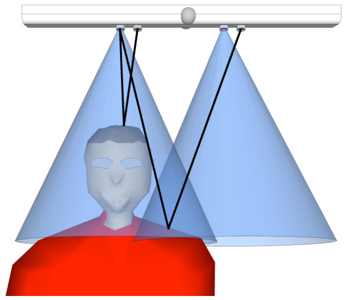

Figure 5. Incorrect height readings can be caused by multi-path reflections between ranging modules used in the same doorway.

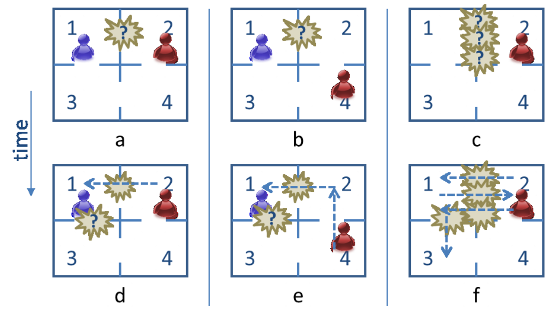

Figure 6. Doorway events can be ambiguous due to the possibility of sensor error (top row), but subsequent doorway events can help resolve ambiguities (bottom row).

the distance to the range finder increases and the person appears shorter. However, we also found that multi-path reflections would sometimes cause the maximum measurement to be taller than the person's true height. This is illustrated in Figure 5: the ultrasound signal from one range finder reflects off the head, which triggers the other range finder to begin taking a measurement. At that very moment, a multi-path reflection (perhaps off the shoulder) reaches the second range finder and causes it to estimate a very short distance (tall height).

Typically, multi-path errors only occur once in a single detection event and therefore do not get clustered with other values. To eliminate these errors, therefore, we use the maximum height in the maximum cluster for each event (in the event illustrated in Figure 3, the maximum cluster is Cluster #3). This approach ensures that outlier height measurements are not used. However, this approach will also filter the true height measurements if a person walks very quickly and the range finders only get a single reading of a person's true height. Therefore, our final height estimation algorithm is: set the height estimate h_i to be the maximum height of the maximum cluster in e_i – unless there is no cluster within the human height range, in which case it is set to be the maximum height measurement in e_i. The final height estimate is combined with the median timestamp from e_i, called t_i to form a new doorway event $d_i = (t_i, h_i)$.

4.4 Direction Estimation

The Doorjamb enclosure tilts the ultrasound sensor into the doorway, so that when a person *enters* the doorway the system detects the tallest heights first, followed by shorter heights. When a person *exits* the doorway, the system detects shorter heights followed by taller heights. Doorjamb applies three algorithms to estimate direction, each of which deals with noise and outliers in a different way. The three algorithms vote on each event e_i with a +1 to declare an enter event, a -1 to declare an exit event, or 0 if it cannot decide. A direction value v_i for event e_i is defined to be the sum of the votes from all three algorithms and is therefore in the range: $-3 < v_i < 3$.

The first algorithm uses robust least squares to fit a line to the height measurements in the event. If the slope is positive, it votes for an enter event, otherwise an exit event. This algorithm only uses values above 140cm to prevent outlier values from affecting the regression. The second algorithm compares the timestamp t_{max} of the maximum height value in e_i to the median timestamp t_i: if t_{max} occurred later than t_i then it votes for an enter event, otherwise an exit event. The third algorithm finds the timestamp t_{min} of the minimum height value in e_i and compares it to the median timestamp t_i: if t_{min} occurred earlier than t_i then it votes for an enter event, otherwise an exit event.

5 Tracking

While the signal processing algorithm operates on data from all range finders in a single doorway, the tracking algorithm operates on the detection events from all doorways. Thus, the sequences of detection events D produced by each doorway are all merged into a single sequence of events E that is processed by the tracking algorithm. Each event $e_i \in E = (t_i, d_i, h_j, v_j)$ is a 4-tuple that indicates the timestamp, doorway ID, height measurement and walking direction associated with the doorway event. The goal of the tracking algorithm is to convert this sequence of doorway events to a sequence of corresponding room states S where each state $s_i \in S = (r1_i, r2_i)$ is a 2-tuple that indicates the room locations of person $p1$ and $p2$ that result from doorway event e_i. For simplicity, we describe the Doorjamb tracking algorithm in the context of a two-person tracking scenario, but a straightforward extension to three or more people is possible. The complexity of multi-target tracking is well known to increase with the number of targets, but we expect Doorjamb to be installed in typical homes with 2-4 residents.

5.1 Analysis of Ambiguity and Constraints

The main challenge for the tracking algorithm is to decide which person caused each doorway event so that the person's track can be updated. In principle, height and direction measurements and/or path constraints should uniquely identify the person who walked through the doorway. However, this association is often ambiguous due to the possibility of height and direction errors, false detections, and missed detections. For example, assume person $p1$ is 160 cm and person $p2$ is 170 cm and the two people are in rooms 1 and 2, both adjacent to the same doorway as shown in Figure 6.a. An event detected in that doorway may indicate that $p1$ probably transitioned to room 2, but the possibility of a height or direction error makes this assignment ambiguous. Even if both people are not both adjacent to the doorway, as shown in Figure 6.b, the assignment is still ambiguous because of the possibility of false detections and missed detections: it is possible that $p1$ moved to room 2, that $p2$ transitioned to room 2 undetected and then moved to room 1, or that the detection is spurious and neither person moved. False detections and missed detections can also result in *sustained ambiguity*. For example, when a person moves back and forth between two rooms, as shown in Figure 6.c, a single detection error could cause the tracking system to consistently place the person in the wrong room, with no chance of disambiguation.

A key insight of Doorjamb is that ambiguities can often be resolved by future observations. To illustrate this, a second detection event is shown in the second row for all 3 examples in Figure 6. The second doorway event shown in Figure 6.d disambiguates the first event in Figure 6.a: $p2$ must have moved during the first event, because otherwise nobody would have been in room 1 to cause the second event. Notice that the second event is able to disambiguate the first event even through the second event is still ambiguous. Similar cases are illustrated in Figures 6.e and 6.f: the second detection event helps to disambiguate the first. To exploit this observation, Doorjamb must be able to process future events without committing to decisions about prior events. To achieve this, it uses a multiple hypothesis tracking approach (MHT) in which multiple alternative tracks are considered simultaneously, each with different doorway assignments and is based on Reid's MHT algorithm [32]. By using a MHT, Doorjamb is able to defer difficult assignments until a later time when disambiguating information becomes available. As new events are processed, tracks that are not consistent with the new information are evicted. The formulation of Doorjamb's MHT is described in the subsections below.

5.2 Models of Height, Direction, and Detection

In order to perform tracking, the Doorjamb system must first model the conditional probabilities $p(H|O)$ and $p(V|O)$: the probability of receiving a height measurement H or direction measurement V given the origin O, where O can be one of three things: Person A, Person B, or a false detection. The system must also model the probability of a missed detection $p(H = \emptyset)$: the probability that no doorway event is generated by the signal processing algorithms, even though a doorway crossing event did occur. In our current implementation, these models are learned during a training period where each individual walks under each doorway multiple times. For example, the observed height readings $\hat{H}_{p1}$ originating from person $p1$ are used to define the probability density function using frequency counting, as follows:

$$p(H = h|O = p1) \quad = \frac{|\hat{H}_{p1}==h)|}{|\hat{H}_{p1}|} \tag{1}$$

The height model for $p2$ and both direction models are learned using the same frequency counting technique. Instead of a training period, the height and direction models could alternatively be defined to be parametric functions of people's true heights and the sensors' tilt angle. At this early stage in this project, however, we use empirical models for simplicity. During this training period, the number of false detections and missed detections are counted and used to define the following models:

$$p(H = h|O = F) \quad = \frac{\text{\# false detections / 62}}{\text{\# doorway events detected}} \tag{2}$$

$$p(H = \emptyset) \quad = \frac{\text{\# missed detections}}{\text{\# true doorway events}} \tag{3}$$

where the number 62 is the total number of positive height values that can be produced by the signal processing algorithm. In other words, actual height values observed for the false detections are assumed to be random, and the overall probability of false detections is uniformly distributed over the range between 137-198 cm (4 ft, 6 in - 6 ft, 6in).

Once the height, direction, and event detection models are created, the probability of missed detections is converted to a *missed transition probability* $p(r_i, r_j)$: the probability of going from room r_i to room r_j without being detected by any doorway sensors. To calculate the transition probability, Doorjamb uses a binary *adjacency matrix* A, where $A(i, j)$ indicates whether r_i and r_j have an adjoining doorway. For example, the square, 4-room floor plan shown in Figure 6 has the following adjacency matrix:

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Using this adjacency matrix, Djikstra's shortest path algorithm is used to find the *path length* between r_i and r_j, which indicates the number of doors through which a person must pass to transition between these rooms: $l(r_1, r_2) = djikstra(A, i, j)$. Then, the missed transition probability is defined to be:

$$p(r_i, r_j) = p(H = \emptyset)^{l(r_1, r_2)} \tag{4}$$

Thus, the probability of staying in the same room without being detected by any doorway sensors is 1, and the probability of passing undetected through k doorways decreases exponentially with k.

5.3 Creating Tracks

In 2-person tracking, a *room state* s_i is defined to be a 2-tuple containing the room locations of persons p1 and p2 after the ith detection event: $s_i = (r1_i, r2_i)$. A *track* is a sequence of consecutive room states. Upon startup, the MHT algorithm creates a track for every possible combination of initial room states: in a 2-person home with k rooms, k^2 initial tracks are created:

$$\begin{aligned} T^1: &\quad (r1=1, r2=1) \\ T^2: &\quad (r1=1, r2=2) \\ &\ldots \\ T^k: &\quad (r1=1, r2=K) \\ T^{k+1}: &\quad (r1=2, r2=1) \\ &\ldots \\ T^{k^2}: &\quad (r1=k, r2=k) \end{aligned}$$

For each new doorway event between rooms i and j, five new room states are possible: either person $p1$ or person $p2$ moved into either room i or j, or the doorway event is a false detection and neither person moved. To consider these 5 possibilities, the tracking algorithm duplicates every existing track 5 times and advances them in each of these five ways. For example, consider track that ends with state $(r1=2, r2=4)$. Upon detecting a doorway event between Rooms 5 and 6, this track would be converted into five new tracks:

$$\begin{aligned} T^1: &\quad \ldots;(r1=2,r2=4);(r1=2,r2=4) \\ T^2: &\quad \ldots;(r1=2,r2=4);(r1=2,r2=5) \\ T^3: &\quad \ldots;(r1=2,r2=4);(r1=5,r2=4) \\ T^4: &\quad \ldots;(r1=2,r2=4);(r1=2,r2=6) \\ T^5: &\quad \ldots;(r1=2,r2=4);(r1=6,r2=4) \end{aligned}$$

The number of possible tracks increases exponentially with the number of doorway events. After processing m doorway events, a total of 5^m tracks will have been created. Furthermore, each track will have $m+1$ room states: $T^i = (s_0, s_1, s_2, s_3, \ldots, s_m)$, where s_0 is an initialization state and state s_i was created in response to doorway event e_i.

5.4 Weighting Tracks

Every track T^i is given a weight w^i that is proportional to the probability of the track given the doorway event observations: $w^i \propto P(T^i|D)$. Upon initialization, the weight of all tracks is set to 1. When room state s_m is added to a track in response to doorway event e_m, the track's weight is updated incrementally. First, the *origin* of event e_m is defined to be the person $p1$ or $p2$ who moved between state s_{m-1} and state s_m or, if nobody moved, a false detection. The origin's previous room location r_p is defined to be the room opposite doorway d_m of the origin's new location in state r_m, and the last observed room location r_{m-1} is defined to be the origin's old location in state s_{m-1}. The doorway direction u_m is defined based on the direction from r_p to r_m.

Given these definitions, the track's new weight is defined to be its old weight, multiplied by (i) the probability of the origin having moved through doorway d_m given height measurement h_m (ii) the probability of moving from r_p to r_m given the direction measurement v_m, and (iii) The probability of moving from the last observed room location r_{m-1} to the previous room location r_p without having been detected:

$$w^i_m = w^i_{m-1} * p(o_m|h_m) * p(u_m|v_m) * p(r_{m-1}, r_p) \tag{5}$$

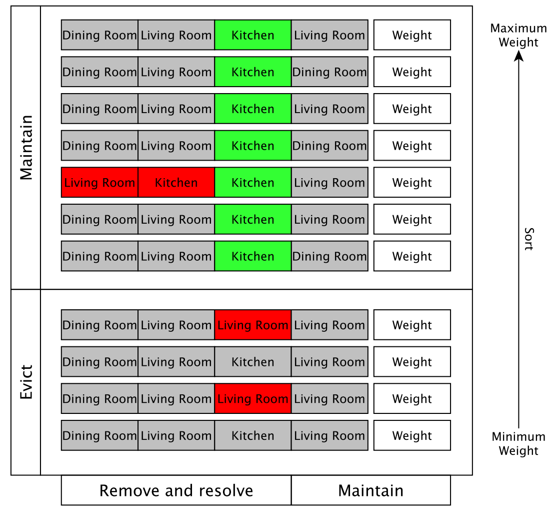

Figure 7. The MHT is composed of many tracks (rows). These are sorted according to their weights and a portion are evicted. The third column is resolved when all tracks agree on the event assignment. Tracks are resolved by removing all preceding state and shortening each individual event sequence.

where the probability of the origin given the height measurement h_m is calculated from the height models defined in Section 5.2, as follows:

$$p(o_m|h_m) = \frac{p(H=h_m|O=o_m)}{\sum_{o=(p1,p2,F)} p(H=h_m|O=o)} \tag{6}$$

and the probability $p(u_m|v_m)$ is calculated using the direction models in an identical fashion. The value $p(r_{m-1}, r_p)$ is defined in Section 5.2.

5.5 Merging and Evicting Hypotheses

The number of tracks under consideration grows exponentially with the number of doorway events detected, and will quickly exhaust memory and computational resources. To address this problem, we use a "n-best" eviction policy to eliminate the least likely tracks. After each doorway event is detected, Doorjamb temporarily increases the number of tracks to $5n$. These tracks are weighted, sorted, and the $4n$ tracks with the lowest weights are evicted, as illustrated in Figure 7. Thus, the number of tracks under consideration is limited to a constant number n after each time step. Real-time estimates of position are determined by reporting the latest entry in the highest weighted track. This corresponds to the most likely position given the previous information.

The main challenge for this eviction policy is that ambiguous doorway events cause existing states to be replicated many times, but do not help differentiate the replicas and therefore many of them have approximately the same weight. As more ambiguous events are encountered, these replicas quickly multiply and dominate the track buffer, causing viable tracks to be evicted. Because of the 5x multiplier for each detection event, the "n-best" eviction policy allows

Doorjamb to consider approximately $\lfloor \log_5(n) \rfloor$ ambiguous doorway events before it makes arbitrary assignments due to space limitations. Even at values of $n = 1000$, Doorjamb can only fully consider 4 ambiguous doorway events at a time.

In order to consider more ambiguous events simultaneously, we developed a track merging algorithm that finds and eliminates unnecessary replicas. The basic intuition is that future information will no longer help resolve ambiguous doorway assignments that occur before a fully known state. For example, if after event m all tracks agree on the state for step $m-1$, no new information can help resolve ambiguity about the states prior to $m-1$. Therefore, Doorjamb can resolve multiple hypotheses about any event prior to the known state. After the eviction step, Doorjamb follows a 4-step merging procedure: (1) it scans for any event about which all tracks share consensus (ii) it *commits* to the values of the track with the highest weight for all states $m-1$ and earlier, i.e. it outputs these state values to the user (iii) it truncates all tracks at step $m-1$, as illustrated in Figure 7, and (iv) it searches all remaining tracks for any exact duplicates: tracks that have exactly the same set of state assignments, and were only differentiated by the states that were just committed. For every pair of duplicate tracks i and j, duplicate j is evicted and duplicate i's weight is updated to be:

$$w^i = w^i + w^j - w^i * w^j \tag{7}$$

This 4-step merging algorithm effectively evicts ambiguous events that will never be resolved. By doing so, it makes space in the track buffer for new ambiguous events to be considered.

6 Experimental Setup

We built 43 ultrasonic doorway sensor platforms and deployed them across 4 different homes for periods of 6-18 months, and used them for development, testing, and iterative design. The results from these tracking tests cannot be used to evaluate the system because it is difficult to collect ground truth information about person location over long periods of time. Therefore, in this paper, we present the results of short-duration controlled tests where the exact timestamps and identity of every doorway event were manually recorded.

We performed 3 controlled experiments to test the effect of height difference on tracking accuracy with different pairs of test subjects ($p1$, $p2$), ($p1$, $p3$), ($p2$, $p3$), where each subject had a different height: $p1 = 190$ *cm*, $p2 = 176$ *cm*, and $p3 = 170$ *cm*. In each experiment, the subjects walked randomly through the house shown in Figure 8 that contains 7 interior doorways. Exterior doorways were not used in this experiment. The rooms in this home were separated by at most 4 doorways, and were 90 *cm* (3 ft) wide x 213 *cm* (7 ft) high and were covered by 2 ultrasonic range finders. Doorways (1) and (2) were 256 x 274 *cm* and 121 x 213 *cm*, respectively, and were therefore covered by 4 and 3 ultrasonic range finders, respectively. During each experiment, the subjects recorded at least 500 doorway events and produced a total of over 3,000 unique doorway events.

6.1 Ground Truth and Baseline

We collected ground truth using a custom hand-held, touch-screen interface showing an image of the house's floor

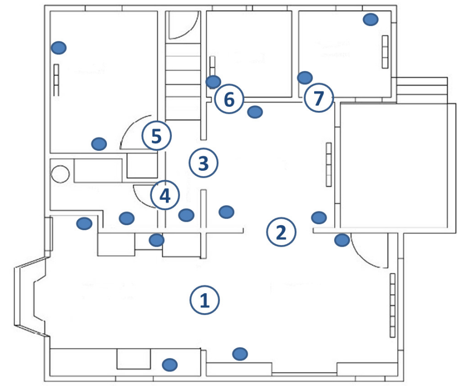

Figure 8. Numbered circles indicate the location of doorway sensors and solid circles indicate the location of Motetrack beacons.

plan. As a person transitioned into a room, he or she would touch the destination room to indicate the room transition. The system recorded the exact timestamp, the user, and the room ID. Because this system used a touch-screen, people occasionally touched the incorrect room by accident, particularly when trying to touch small rooms. These events were easily found and manually cleaned from the ground truth data set because they resulted in illegal paths. The hand-held interface provided visual feedback about the room that was touched and in the case of a mis-touch, the person immediately touched the correct room before performing a new room transition. Therefore, the final ground truth data sets consisted of legal and continuous paths. We estimate that a total of 20 mis-touches occurred out of the total 3000 doorway events recorded.

To compare with an existing system, we simultaneously deployed the Motetrack radio-based localization system [7]. Motetrack relies on a set of beacon nodes placed throughout the home that periodically transmit messages at varying transmission powers, and a person must carry a battery-powered wireless receiver in order to be tracked. Prior to running our experiments, we deployed 15 Motetrack beacons, as illustrated in Figure 8. Each beacon was powered by a AC-powered USB adapter and were plugged directly into AC power outlets. We tried to place beacons along the boundries of rooms in order to improve room-level differentiation, although the number and locations of the beacons were constrained by the locations of power outlets. After the experiment, we converted the Motetrack location estimates into room locations and room transitions.

6.2 Evaluation Metric

Both Doorjamb and Motetrack produce a sequence of events consisting of room transitions, and each transition is assigned to a test subject. To determine tracking accuracy, these sequences of events is compared to the room transitions as recorded by the ground truth system. However, the timestamps on these two event sequences will typically not match up perfectly due to a test subject's imperfect timing between walking and touch-screen recording, touch-screen mis-touches and re-touches, and wireless transmis-

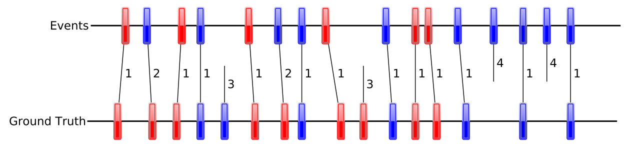

Figure 9. Bi-partite matching is used to address timestamp mis-matches between observed events and ground truth.

sion delays. Therefore, we define a one-to-one correspondence between the empirical and ground truth data using a max-weight bi-partite matching algorithm. Events are only matched if they include the same person and doorway, and the weight to match each pair decreases linearly as the difference in timestamps increases, up to a maximum difference of 10 seconds. The match produced by this algorithm is illustrated in Figure 9, where the top row represents empirical events and the bottom row represents ground truth. The red and blue boxes indicate whether the event is assigned to person A or B. This figure highlights four different types of tracking outcomes:

- **Type 1:** A correct state transition that matches ground truth.
- **Type 2:** An incorrect state transition because the tracking algorithm assigned the doorway event to the wrong person.
- **Type 3:** An incorrect state transition due to a false room transition.
- **Type 4:** An incorrect state transition due to a missed room transition.

We define tracking *accuracy* to be:

$$\text{accuracy} = \frac{\text{\# Type 1 matches}}{\text{\# ground truth events}} \tag{8}$$

In other words, we evaluate the fraction of room transitions that were detected and correctly assigned to a person. This evaluation does not consider the amount of time spent in each room and, thus, a brief walk through a hallway will have the same effect on accuracy as a long visit to a kitchen. This metric is most useful to determine the degree to which Doorjamb can be used to associate activities such as flicking a light switch with a person based on his or her room location. Other metrics would be more appropriate to evaluate, for example, how much time a person typically spends in each room. We could not meaningfully evaluate this aspect of Doorjamb using our experiments because they were not *in-situ* and therefore the time spent in each room is not representative of real home usage. We will further explore time duration once a long-term, in-situ ground truth system becomes practical.

6.3 Training the Noise Models

The height, direction, and event detection probabilities described in Section 5.2 are trained using 3-fold cross validation. In other words, the probabilities used for run ($p1$, $p2$) was derived from the ground truth data of runs ($p1$, $p3$)

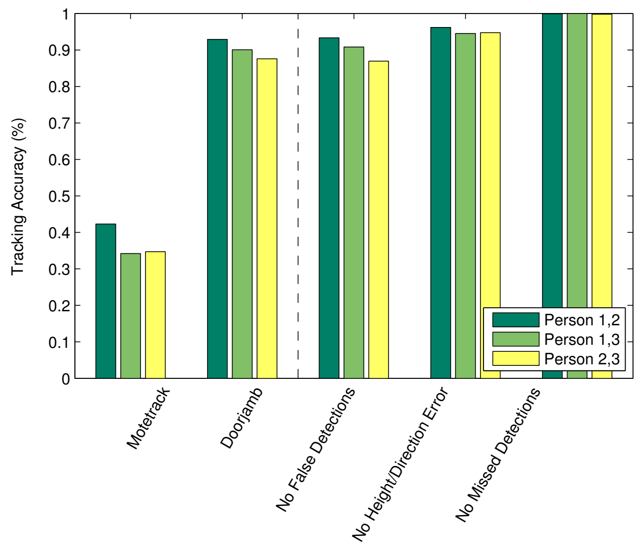

Figure 10. Motetrack achieves 37% tracking accuracy and Doorjamb achieves an average of 90% accuracy. Analysis indicates how much results improve when false detections, height/direction error, and missed detections are removed.

and ($p2$, $p3$). This avoided the need to perform explicit training experiments. The following constant values were used as parameters to the signal processing and tracking algorithms. The inner detection event window, w_i, was set to extend 200 msec around the initial detection event. The outer detection event window for noise filtering, $\hat{w}_i$, was set to extend 200 msec around the inner detection event window. The parameters for DBSCAN were a 400 msec radius and minimum of two values per cluster for timestamp clustering, and a 5 cm radius and minimum of two values per cluster for height clustering.

6.4 Tracking Accuracy

The tracking results for both Doorjamb and MoteTrack and shown on the left side of Figure 10, where each set of 3 bars represents the results for the three experimental runs. The 1st set of bars shows that Motetrack averages 37% room-level tracking accuracy. In comparison, the 2nd set of bars shows that Doorjamb achieves 90% room-level tracking accuracy on average.

Motetrack's performance can be explained by several factors. First, Motetrack performance increases with the number of beacon nodes, and its performance may have been improved with more than the 15 beacon we deployed (roughly 2 per room). Second, the Motetrack receiver was carried in pant pockets throughout the experiments, and rotation of the body would cause significant changes to the radio signal strength measurements, appearing as physical movement even if the person was only turning in place. Most importantly, however, Motetrack is not designed for room-level tracking: it locates a person in 2D space, and a small error in (x,y) coordinates can lead to large errors in room location because it would place a person on the wrong side of a wall or doorway. The results of this experiment do not

Doorway	Precision	Recall
1	0.933	0.842
2	0.957	0.975
3	0.959	0.927
4	0.809	0.958
5	0.927	0.973
6	0.996	0.979
7	0.968	0.983

Table 2. The system produces more than 93% precision and 96% recall for most doorways.

demonstrate that Doorjamb is a better tracking system than Motetrack, but only that it achieves higher room-level tracking accuracy, when used with approximately 2 beacon nodes per room.

The extent to which Doorjamb's accuracy is affected by false detections, missed detections, height errors, and direction errors is illustrated in Figure 10. The 3rd set of bars shows that tracking accuracy improves to 90.4% on average when false detections (Type 4 matches) are removed from the empirical data set. The 4th set of bars show that accuracy improves to 95% on average when ground truth height and direction values are used. The 5th set of bars shows that accuracy improves to 99.9% when missed detections (Type 3 matches) are replaced with ground truth events. This is equivalent to running the tracking algorithm on the ground truth data set, and illustrates that the tracking algorithm itself is not introducing errors. From this analysis, we conclude that false detections, height/direction errors, and missed detections account for 0.4%, 4.6%, and 4.9% of tracking accuracy, respectively.

6.5 False Detections and Missed Detections

Table 2 indicates the precision and recall for each doorway, where *precision* is the number of false detections divided by the number of total detections, and *recall* is the number of missed detections divided by the number of true doorway crossing events. Most doorways had precision levels above 93% and recall numbers above 96%. A few exceptions include doorway 4 with 81% precision and doorway 1 with 84% recall. Doorway 4 causes more false events because a shelf at roughly the same height as a person is picked up by the device. Doorway 1 sensors must cover a larger doorway and are mounted approximately 50 *cm* higher then the rest of the doorways. A small number of these are the result of packet loss between the doorway and base station. This results in a lower sample rate, thus causing more events to be missed.

6.6 Height Measurement Accuracy

Figure 11 shows the range of heights measured for each person. The modes of these distributions roughly correspond to the differences in the subject's actual heights: 6 *cm* and 14 *cm*. However, there is substantial overlap between the full range of measurements observed for each subject. Most measurements fall within a range of 172-195 *cm*, 160-184 *cm*, and 157-178 *cm* for subjects 1, 2, and 3, respectively. This overlap indicates that many doorway events will be ambiguous, and that height alone is not sufficient to identify

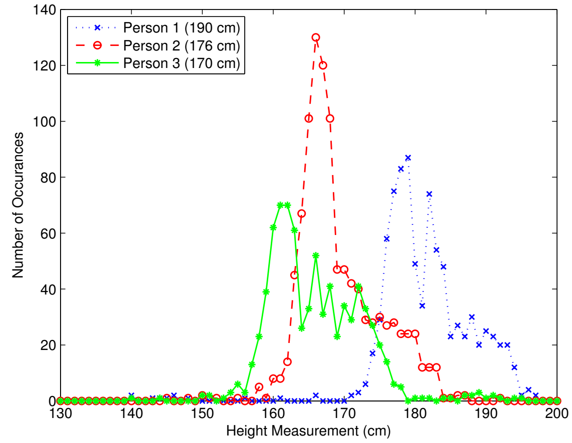

Figure 11. The range of height measurements from different people have large overlap, particularly between persons 2 and 3.

Measure	Correct	Incorrect	Percent
-3	881	133	86.9%
-2	0	0	-
-1	190	133	58.8%
0	-	-	-
1	230	84	73.3%
2	0	2	0%
3	714	111	86.6%

Table 3. Three algorithms produce direction measures between -3 and 3.

who is walking through a doorway. This is particularly true for subjects 2 and 3 that have similar heights. Height errors are not normally distributed, and measurements are more likely to be produced slightly above the mode than below. This error can be caused by many factors, including multipath errors, as described in Section 5.1.

6.7 Direction Measurement Accuracy

The Doorjamb sensors are able to deterine direction through a doorway with 81% accuracy overall. Three different algorithms are combined to form a score between -3 and 3 with a negative value meaning a person is exiting the doorway and a positive value indicating entering. Additionally, 0 is an unknown state where the algorithms have insufficient data and do not assign a direction. Intuitively, when all three algorithms agree, they produce more extreme measures, and are more likely to be correct. Table 3 shows the accuracy of the system for each value: values of -3 and 3 are correct 87% of the time, while values of 1 or -1 are correct 58% and 72% of the time, respectively. Very few doorway events produce values of 0, 2 and -2.

6.8 Systems Performance

In order to evaluate the trade-off between computational resources and tracking accuracy, we executed the tracking algorithm while varying the number of tracks that can be considered simultaneously. The results are illustrated in Fig-

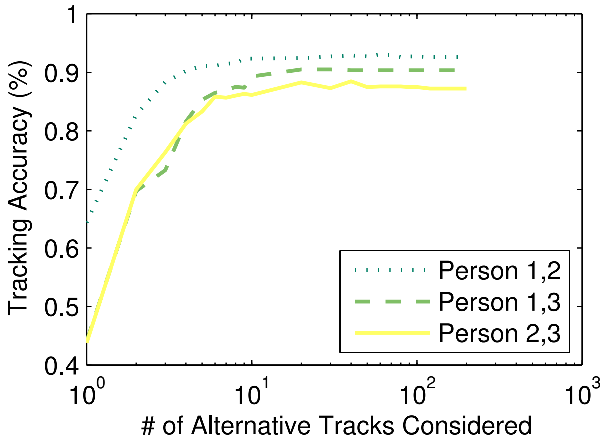

Figure 12. A 4-step path merging algorithm allows Doorjamb to perform well with only 20-30 simultaneous tracks.

ure 12, and demonstrate that Doorjamb can store, update, and compare using as few as 20 alternative tracks. While processing the 3000 doorway events in our experiments, tracks stored in memory were 24 states long on average, with a maximum length of 55 states. This indicates that Doorjamb can be executed on memory-constrained devices such as a cell phone without sacrificing tracking accuracy. Note that even 1000 tracks was not sufficient before we added the track merging algorithm described in Section 5.5.

The algorithm can also be executed in *real time*: using a 4-core Intel i7 processor running at 2.67 GHz, all 3000 doorway events in our experiments could be processed in a total of 280 seconds with 3100 seconds dedicated to signal processing and 100 seconds for a Python implementation of our tracking algorithm. Doorjamb is an on-line tracking algorithm, meaning that it does not need to process all data in batch. However, it does require look-ahead for noise filtering and for data association. The signal processing algorithm uses a 500 millisecond look-ahead window, which also adds a small delay to doorway event detection. In our experiments, the MHT tracking algorithm used a look-ahead of 14 doorway events on average before committing to a doorway assignment, with a maximum look-ahead of 49 doorway events. The length of this delay depends on how frequently people in a house move between doorways, and may even be several hours. However, just like other data association algorithms, Doorjamb can make an immediate estimate of each person's location based on the distribution of states in the tracks under consideration, and can use the variability in those states to provide a confidence value. In this usage mode, of course, the user may see changes in the estimated location if future events cause the system to change the tracking estimate.

7 Limitations and Future Work

During our experiments, subjects were instructed to walk with a natural pace and posture. However, these experiments fall short of true *in-situ* experiments that would involve natural activities such as carrying groceries while walking, wearing of shoes or hats, and people standing or lingering in doorways. The experiments also do not capture long-term effects such as the moving of furniture over time, opening and closing of doors, and the appearance and disappearance of household objects such as bags or laundry baskets. We refrained from intentionally introducing these noise sources in order not to create contrived experiments. Instead, we consider the experiments in this paper to provide a proof-of-concept for Doorjamb tracking: the system is demonstrated to work under the noise conditions tested. The effect of natural noise sources is outside the scope of this paper and will be the subject of future work once long-term, in-situ ground truth collection becomes practical.

The complexity of the Doorjamb tracking algorithm grows with the number of people, causing either an increase in computation resources or a decrease in accuracy. Our analysis shows that Doorjamb can track two people while considering as few as 20 tracks, and we expect satisfactory performance in typical homes with 3-4 people. However, performance will degrade with larger numbers of people, both due to height ambiguity and computational complexity. The Doorjamb sensor system is currently unable to detect two people crossing through a single doorway simultaneously, but the tracking algorithm can compensate by infering the missed doorway crossing in order to generate consistent paths.

Our current system requires calibration and training to create sensor models for the doorways. This type of extensive calibration and training may be practical for industrial systems or in-home medical monitoring systems that are deployed by trained professionals, but is rarely seen in products for the general population. In current work, we are developing new unsupervised learning algorithms to automatically learn system parameters and noise models in order to eliminate need for a training period. We believe this approach will also substantially improve results by allowing individual noise models to be learned for each doorway and person, instead of a single generic model for all doorways.

Our current system does not detect children due to an artificial constraint on the detectable height range. This is not a fundamental limitation of the system or the sensors. However, the introduction of children into the experimental method will introduce complications with pets or other low-laying sources of noise. Lastly, Doorjamb does not incorporate other non-intrusive sensors into the tracking algorithm, such as motion sensors and the open-closed status of the door. Our doorway sensors do include both types of sensors, but these sensors were not used in the current analysis. In future work we will incorporate this information, and we believe these sensors will only improve tracking accuracy.

8 Conclusions

In this paper, we present the Doorjamb tracking system that can track people in homes with room-level accuracy without requiring any user participation, wearable devices, privacy-intrusive sensors, or high-cost sensors. The system operates by sensing the heights and directions of people as they walk under each doorway, thereby differentiating between them and identifying which room they are entering.

The doorway sensors are small, low-power, and can be discretely mounted behind the door jamb. We evaluate this system in a home with 8 rooms and 7 doorways, using 3 controlled experiments totaling over 3000 doorway crossing events. Our results indicate that the system can achieve 90% tracking accuracy on average, and that accuracy is highest when the test subjects have different heights. Despite substantial measurement and environmental noise, Doorjamb can provide room-level tracking as well or better than existing technologies that rely on a wearable device.

Indoor tracking systems will be the basis for many future technologies and smart-home applications such as including elderly and patient monitoring, activity recognition, and occupancy-drive lighting, heating, and cooling. Tracking will allow these systems to provide location-aware services and individually-tailored services, even in multi-person homes. Doorjamb will be an enabling technology by enabling accurate indoor tracking while eliminating barriers faced by other tracking systems. The need for devices to be carried by participants is eliminated, as is the possibility for missing or incomplete data because the users forget to wear the devices. The chance for privacy invasion is significantly reduced because ultrasound range finders provide very little information other than height, direction, and location. This will make it easier to convince people to agree to an installation of the system in their homes. Finally, the system is low-cost, discrete, and easy to install.

9 Acknowledgments

We express great gratitude to the many volunteers in our sensing studies. This work is based upon work supported, in part, by the National Science Foundation under Grants EFRI-1038271 and CAREER-0845761.

10 References

[1] K.P. Fishkin, M. Philipose, and A. Rea. Hands-on RFID: Wireless Wearables for Detecting Use of Objects. In *Ninth IEEE International Symposium on Wearable Computers, (ISWC '05)*, 2005.

[2] V. Srinivasan, J. Stankovic, and K. Whitehouse. Using Height Sensors for Biometric Identification in Multi-resident Homes. In *The International Conference on Pervasive Computing, (Pervasive '10)*, 2010.

[3] R. Want and A. Hopper. Active Badges and Personal Interactive Computing Objects. *IEEE Transactions on Consumer Electronics*, 38(1):10–20, 1992.

[4] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-support System. In *The 6th Annual International Conference on Mobile Computing and Networking, (MobiCom '00)*, 2000.

[5] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The Anatomy of a Context-aware Application. In *The 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, (MobiCom '99)*, 2002.

[6] P. Bahl and V.N. Padmanabhan. RADAR: An In-building RF-based User Location and Tracking System. In *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM '00)*, 2000.

[7] K. Lorincz and M. Welsh. Motetrack: A Robust, Decentralized Approach to RF-based Location Tracking. *Personal and Ubiquitous Computing*, 2007.

[8] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall. Recognizing Daily Activities with RFID-based Sensors. In *The 11th International Conference on Ubiquitous Computing, (UbiComp '09)*, 2009.

[9] K. Chawla, G. Robins, and L. Zhang. Efficient RFID-based Mobile Object Localization. In *The 6th International Conference on Wireless and Mobile Computing, (WiMob '10)*, 2010.

[10] L.M. Ni, Y. Liu, Y.C. Lau, and A.P. Patil. LANDMARC: Indoor Location Sensing Using Active RFID. *Wireless Networks*, 10(6):701–710, 2004.

[11] W. ur Rehman, E. de Lara, and S. Saroiu. CILoS: A CDMA Indoor Localization System. In *The 10th International Conference on Ubiquitous Computing, (UbiComp '08)*, 2008.

[12] V. Otsason, A. Varshavsky, A. LaMarca, and E. De Lara. Accurate GSM Indoor Localization. In *The 7th International Conference on Ubiquitous Computing, (UbiComp '05)*, 2005.

[13] T.W. Hnat, V. Srinivasan, J. Lu, T.I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse. The Hitchhikers Guide to Successful Residential Sensing Deployments. In *The 9th ACM Conference on Embedded Networked Sensor Systems, (Sensys '11)*, 2011.

[14] T. Teixeira and A. Savvides. Lightweight People Counting and Localizing in Indoor Spaces Using Camera Sensor Nodes. In *The First International Conference on Distributed Smart Cameras, (ICDSC '07)*, 2007.

[15] V. Menon, B. Jayaraman, and V. Govindaraju. Biometrics Driven Smart Environments: Abstract Framework and Evaluation. pages 75–89, 2008.

[16] P. Chen, P. Ahammad, C. Boyer, S.I. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, et al. CITRIC: A Low-Bandwidth Wireless Camera Network Platform. In *The 2nd International Conference on Distributed Smart Cameras, (ICDSC '08)*, 2008.

[17] Z. Liu and S. Sarkar. Outdoor Recognition at a Distance by Fusing Gait and Face. *Image and Vision Computing*, 25(6):817–832, 2007.

[18] D. Gafurov and E. Snekkenes. Gait Recognition Using Wearable Motion Recording Sensors. *Journal on Advances in Signal Processing, (EURASIP '09)*, 2009:7, 2009.

[19] SME Hossain and G. Chetty. Next Generation Identity Verification Based on Face-Gait Biometrics. In *The International Conference on Biomedical Engineering and Technology, (ICBET '11)*, 2011.

[20] Y. Zhu, T. Tan, and Y. Wang. Biometric Personal Identification Based on Iris Patterns. In *The 15th International Conference on Pattern Recognition, (ICPR '00)*, 2000.

[21] R.P. Wildes. Iris Recognition: An Emerging Biometric Technology. In *Proceedings of the IEEE*, 1997.

[22] J. Schiff and K. Goldberg. Automated Intruder Tracking Using Particle Filtering and a Network of Binary Motion Sensors. In *The International Conference on Automation Science and Engineering, (CASE '06)*, 2006.

[23] Z. Wang, H. Li, X. Shen, X. Sun, and Z. Wang. Tracking and Predicting Moving Targets in Hierarchical Sensor Networks. In *The International Conference on Networking, Sensing, and Control, (ICNSC '08)*, 2008.

[24] Q. Hao, D.J. Brady, B.D. Guenther, J.B. Burchett, M. Shankar, and S. Feller. Human Tracking With Wireless Distributed Pyroelectric Sensors. *Sensors Journal*, 6(6):1683–1696, 2006.

[25] P.K. Dutta, A.K. Arora, and S.B. Bibyk. Towards Radar-Enabled Sensor Networks. In *The 5th International Conference on Information Processing in Sensor Networks, (IPSN '06)*, 2006.

[26] N. Patwari and J. Wilson. RF Sensor Networks for Device-Free Localization: Measurements, Models, and Algorithms. *Proceedings of the IEEE*, 98(11):1961–1973, 2010.

[27] W.H. Liau, C.L. Wu, and L.C. Fu. Inhabitants Tracking System in a Cluttered Home Environment via Floor Load Sensors. *The IEEE Transactions on Automation Science and Engineering, (T-ASE '08)*, 5(1):10–20, 2008.

[28] Y.L. Shen and C.S. Shin. Distributed Sensing Floor for an Intelligent Environment. *Sensors Journal*, 9(12):1673–1678, 2009.

[29] M.D. Addlesee, A. Jones, F. Livesey, and F. Samaria. The ORL Active Floor. *Personal Communications*, 4:35–41, 1997.

[30] National Health Statistics Reports. http://www.cdc.gov/nchs/data/nhsr/nhsr010.pdf, 2008.

[31] Improving Pedestrian Safety at Unsignalized Crossings. http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_rpt_562.pdf, 2006.

[32] D. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.