An Empirical Design Space Analysis of Doorway Tracking Systems for Real-World Environments

ERIN GRIFFITHS, AVINASH KALYANARAMAN, JUHI RANJAN, and KAMIN WHITEHOUSE, University of Virginia

Doorway tracking systems track people's room location by instrumenting the doorways rather than instrumenting the rooms themselves—resulting in fewer sensors and less monitoring while still providing location information on occupants. In this article, we explore what is required to make doorway tracking a practical solution. We break a doorway tracking system into multiple independent design components, including both sensor and algorithmic design. Informed by this design, we construct a doorway tracking system and analyze how different combinations of these design components affect tracking accuracy. We perform a six-day *in situ* study in a ten-room house with two volunteers to analyze how these design components respond to the natural types and frequencies of errors in a real-world setting. To reflect the needs of different application classes, we analyze these design components using three different evaluation metrics: room accuracy, duration accuracy, and transition accuracy. Results indicate that doorway tracking can achieve 99.5% room accuracy on average in controlled settings and 96% room accuracy in *in situ* settings. This is contrasted against the 76% *in situ* setting room accuracy of Doorjamb, a doorway tracking system whose design implements only a limited number of components in our proposed doorway tracking system design space. We describe the differences between the data in the *in situ* and controlled settings, and provide guidelines about how to design a doorway tracking system for a given application's accuracy requirements.

CCS Concepts: • **Computer systems organization → Sensor networks**; **Sensors and actuators**;

Additional Key Words and Phrases: Doorway tracking systems, smart homes, sensor networks

ACM Reference format:
Erin Griffiths, Avinash Kalyanaraman, Juhi Ranjan, and Kamin Whitehouse. 2017. An Empirical Design Space Analysis of Doorway Tracking Systems for Real-World Environments. *ACM Trans. Sen. Netw.* 13, 4, Article 26 (September 2017), 34 pages.
https://doi.org/10.1145/3089157

1 INTRODUCTION

"Connected Home" technology provides remote access and control to many aspects of the home: the thermostat, door locks, smart appliances, and even smart egg cartons that alert users to buy more eggs. However, there is still a dearth of technology to monitor the most interesting things in the home: the people. People are biological objects and are fundamentally difficult to interface with the digital world: they don't consistently carry or wear electronics and, at least in their homes,

This work is supported by the National Science Foundation, under grant EFRI-SEED-1038271.
Authors' addresses: E. Griffiths, A. Kalyanaraman, J. Ranjan, and K. Whitehouse, Computer Science Department, University of Virginia, 85 Engineer's Way, Charlottesville VA, 22901; emails: {erg3wb, ak3ka, jr3cz, whitehouse}@virginia.edu.

https://doi.org/10.1145/3089157

don't typically want to be monitored on camera or microphone. As such, several research groups have recently begun to monitor people with *doorway tracking systems* that detect when a person passes through a doorway from one room to another. These systems have been built using ultrasound, infrared, radio-frequency identification (RFID), pressure sensors, and most recently using width sensors (Hnat et al. 2012; Lee et al. 2014; Patel et al. 2008; Ranjan et al. 2013; Khalil et al. 2016; Kalyanaraman et al. 2013). The goal of doorway tracking is to monitor the room location of each person in the home, which can help to understand their context because each room is equipped for specific activities: people cook in the kitchen, bathe in the bathroom, and sleep in the bedrooms. Doorways are small spaces with limited usage patterns, and so it is often easier to monitor room location by instrumenting the doorways than by instrumenting the rooms themselves.

Based on our initial experiences in this domain, however, we found that human body movement is far more varied in doorways than might be expected. For example, people don't just walk through doorways; they also lean against them to chat, pass groceries through them, put an arm through them to flip a light switch, and so on. We also found that nearby objects such as doors and furniture often cause noise or mimic human presence. Finally, we found that homes can have a variable number of people over time and occasionally have guests. These real-world issues will affect all doorway tracking systems, but each system will naturally be more robust to some errors than others.

In this article, we explore what is required to make doorway tracking a truly practical solution in terms of hardware design, requirements for training data, and realistic assumptions about occupancy. First, we design and build new sensing hardware to monitor four different aspects of the doorway: presence of a person, identity of a person, direction of a person, and the occupancy of neighboring rooms. Second, we design a new tracking algorithm that has three unique features: it does not require training data, it can dynamically track a variable number of people, and it can track short-term guests. Finally, we analyze these design components by combining them into different variants of a tracking system, each having one more design component than the last, and comparing the resulting tracking accuracy. This extensive design space analysis produces an understanding of how each design component affects the system's robustness to various types of errors.

For a complete analysis, we must examine our proposed design space, hardware, and algorithms in a natural, real-world setting. Therefore, we performed an *in situ* evaluation by deploying our systems in a ten-room house and recruited two volunteers who were not involved with the study to live in the home for six days. The participants performed all daily activities in the home such as cooking, showering, and sleeping. Their ground truth locations were collected by manually reviewing video recordings of their movements through the doorways using night-vision enabled cameras. Ground truth data collection in an *in situ* experiment is very challenging and few works perform *in situ* evaluations of a human tracking system over multiple days. Thus, to compare with the results of prior work, we also performed two controlled studies with two and three people, respectively, in which ground truth locations were collected manually.

Smart home applications such as home automation or elderly health monitoring use room location in different ways. Accordingly, we explore our design space and implementation in the context of three different evaluation metrics: room accuracy, duration accuracy, and transition accuracy. A tracking system has high *room accuracy* if it correctly estimates which room a person is in. This is necessary to answer questions such as "Has the toddler gotten into the kitchen?", "Who went into the basement yesterday to use the washing machine?", and so on. A tracking system has high *duration accuracy* if it correctly estimates how long a person was in a given room. This is necessary to answer questions such as "How long was the child in the TV room?", "Have my sleeping hours been declining over the past month?", and so on. Finally, a tracking system has high *transition accuracy* if it correctly estimates the exact time that a person enters or exits a room. This is the most rigid evaluation metric and is necessary to answer questions such as "Who was the last person to leave the living room, since the lights are still on?"

Results show that our implementation of the doorway tracking design space achieves 99%, 97%, and 97% accuracy for room, duration, and transition accuracy, respectively, in a controlled setting and peak accuracies of 96%, 91%, and 87% respectively in an *in situ* setting. The Doorjamb doorway tracking system, whose design implements only a subset of our design space with ultrasonic sensors and identifies individuals by height, only has accuracies of 76%, 39%, and 24% respectively in an *in situ* setting. To analyze these results, we describe the differences between the data from the *in situ* and controlled studies and how they affected each of the design components that we explored. Results also show that doorway tracking can achieve 94% room accuracy and 91% duration accuracy with only a limited number of design components—far above the Doorjamb baseline. These results serve as guidelines for how to design a doorway tracking system for a given application's accuracy requirements. The unique contributions of this article include:

- a new understanding of natural, real-world doorway environments and how they differ from a controlled laboratory setting;
- new hardware designs for doorway sensing, including presence sensing and neighbor room state sensing;
- new algorithm designs for doorway tracking, including the elimination of training data and the ability to track a variable number of people;
- a new understanding of how these design components affect different evaluation metrics for doorway tracking in a natural environment.

2 DOORWAY SENSOR DESIGN

To build a doorway tracking system, we must first identify the design components and decisions necessary to perform tracking. In any doorway tracking system, two main design elements are required to fulfill that goal: the doorway sensor hardware at each doorway monitoring transition and a tracking algorithm that uses transition information to track an occupant from room to room. The doorway sensor must sense at minimum three pieces of information: when a person enters the doorway, which direction they are going, and the identity of the person. To a doorway sensor the canonical example of a doorway transition is when a person walks through the doorway from one room to another. However, there are many more ways a person or object can interact with a doorway, including opening or closing doors, walking past a doorway but not through, reaching through a doorway to turn on a light, and when a person actually enters the doorway, perhaps to talk to someone in the other room, but then exits the way they came. Each of these exceptions must be considered with the canonical example when designing a doorway sensor.

To handle all possible interactions with the doorway, our doorway sensor design senses three zones around a doorway: the *presence zone*, which covers the area inside the doorframe; the *direction zone*, which covers the immediate area on either side of the doorframe; and the *room zone*, which covers either adjoining room, illustrated in Figure 1. Each of these zones plays a different role in sensing how and when a person interacts with a doorway. A doorway tracking system would implement some subset of these zones into a doorway sensor, fusing information from each zone as necessary to detect when a person crosses the doorway. Our final hardware design for the doorway system evaluated in this work implements all three of these zones with different selected sensors.

2.1 Presence Zone

The goal of a sensor monitoring the presence zone is to detect when a person is actually within the doorway. Any narrow region sensing technology, such as break beam sensors used at crosswalks,

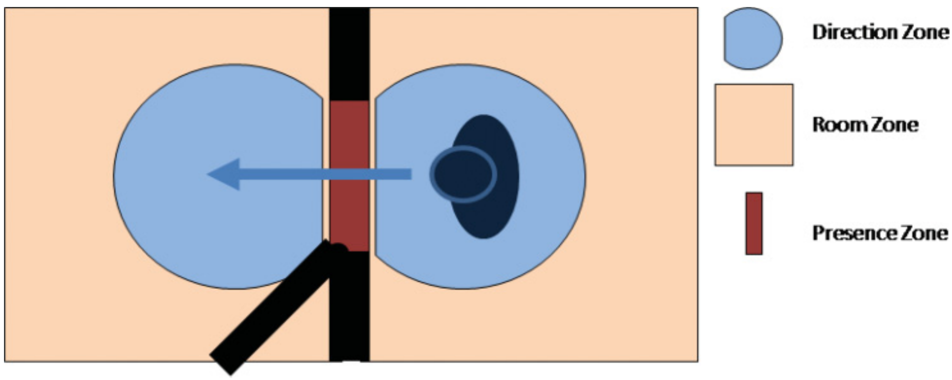

Fig. 1. The doorway sensor design has three zones: the direction zone (circular, blue) detects from which side of the door a person approaches; the presence zone (center, red) detects the exact moment when the person is inside the doorframe; and the room zone (square, yellow) detects when a person is in the adjacent rooms.

would fit the requirements of a presence zone sensor. This zone benefits the doorway sensor by not detecting objects outside of the doorway threshold, such as attached doors or a person walking by but not through. However, due to the narrow nature of the presence zone it cannot accurately detect the direction of a person crossing through the doorway. Alone, the presence zone could detect when someone was in the doorway, but would have no knowledge of the rooms around the doorway nor which one a person is going into.

2.2 Direction Zone

The goal of the direction zone sensor complements the presence sensor by determining the direction of a person. By sensing the immediate area on either side of the doorway, the direction sensor can determine where a person is coming from and where they are going to. Any sensors that can determine direction through a relatively small (2–4ft) region of space, such as short range radar, can act as a direction zone sensor. However, unlike a presence zone sensor, the direction zone sensor *will* detect a person walking nearby, or the attached door. Alone, the direction zone sensor might erroneously infer that a person has walked through the doorway when two people walk by simultaneously. Only by combining the direction zone sensor and the presence zone sensor can a doorway sensor detect both when a person has crossed and their direction.

2.3 Room Zone

The goal of the room zone is to handle the more complex doorway interactions that are not caught by the presence and direction zones. These situations include when a person enters a doorway for some reason, such as turning on a light switch or talking to someone, but then returns to their original room. These situations cause the presence zone sensor to incorrectly detect a transition of a person. The room zone sensor helps disambiguate these situations by sensing the movement of people in the room before or after an event. If the room zone sensor senses people in only one adjacent room, then a crossing during that period is unlikely. An example of such a sensor might be a directed motion sensor.

2.4 Identity Sensing

The final piece of information that the doorway sensor must detect is the identity of the person. A wide variety of sensors may be used here, including an additional sensor that does not monitor one of the above sensing zones. This may include weak biometrics, such as height or weight, or strong biometrics, such as long range retinal scan technology (Venugopalan et al. 2011).

3 TRACKING ALGORITHM DESIGN

A tracking algorithm is needed in part to deal with any missed or erroneous detections made by the doorway sensor. Essentially, it turns the crossings of every person through the doors of a home into a *spatio-temporal sequence*—a track containing the rooms that each person was located in over time. The benefit of such a tracking algorithm is that it can use a holistic view of the people in the home to correct any errors made by the individual doorway sensors. These errors manifest themselves in four main ways—(i) *false positives* (when the doorway sensor detects someone walking through a doorway when they have not), (ii) *false negatives* (FNs—when the doorway sensor misses someone walking through a doorway), (iii) *direction errors* (DEs—the doorway sensor indicates a person likely moved from *room1* to *room2*, but in reality the person moved from *room2* to *room1*), and (iv) *biometric errors* (e.g., data fusion reported an incorrect 155*cm* height value for a 177*cm* tall person). Tracking also infers the identity of a person from the biometric measurements of the identity sensor. Hence, tracking associates the biometric with an individual and corrects errors in the individual doorway sensor with a multi-event, holistic view of the system. To help in the selection of a tracking algorithm, we identify the main design decisions to be made below. Our final tracking algorithm implements each of these design decisions.

3.1 Dynamic Tracking

Ideally, a tracking system should operate in a dynamic, training-less way for practical use with a variable number of occupants. Training in this case refers to learning the distributions of false positives, false negatives, biometric errors, and direction errors that occur at each door for each resident (and any potential guests) in a home. As these distributions are human behavior dependent, can fluctuate over time (e.g., the height errors of a person wearing different size heels), and can be doorway usage dependent (e.g., false positives are more common when doors are often open and closed) learning these distributions for each home and every situation is generally infeasible. Overall, training for tracking is often infeasible as (i) training takes time, (ii) people's behaviors are time-variant, (iii) residents may change, (iv) guests frequent homes, (v) different guests exhibit different behavioral patterns, and so on. This warrants the desire for a training-less tracking algorithm that can track a variable number of people in a home. While this design decision does not necessarily increase accuracy, it does decrease the setup and installation effort of the tracking system while allowing the system to more easily generalize.

3.2 Perimeter Sensing

When including training-less, dynamic tracking, we found that a perimeter sensor greatly increased tracking accuracy. Perimeter sensors refer to the class of sensors that can detect the entry/exit of a person into a home, so the exact number of people in the home is known at a given time. These sensors help the tracking algorithm create person tracks by limiting the algorithm's search space (the number of people attributable to a given doorway event). In other words, in the absence of perimeter sensing, tracking must consider the possibility that any number of people are in the home and attempt to track them all simultaneously. With perimeter sensing, the tracking algorithm knows the number of people in the home and can more easily determine when errors occur in sensing. Perimeter sensing can take two forms: knowing when any person enters or exits the home and additionally knowing the identity of that person. However, we do not consider these sensors for an internal doorway sensor design, because such a sensor would need to be implemented with a more accurate, but far more intrusive, crossing and identity detection. This may include

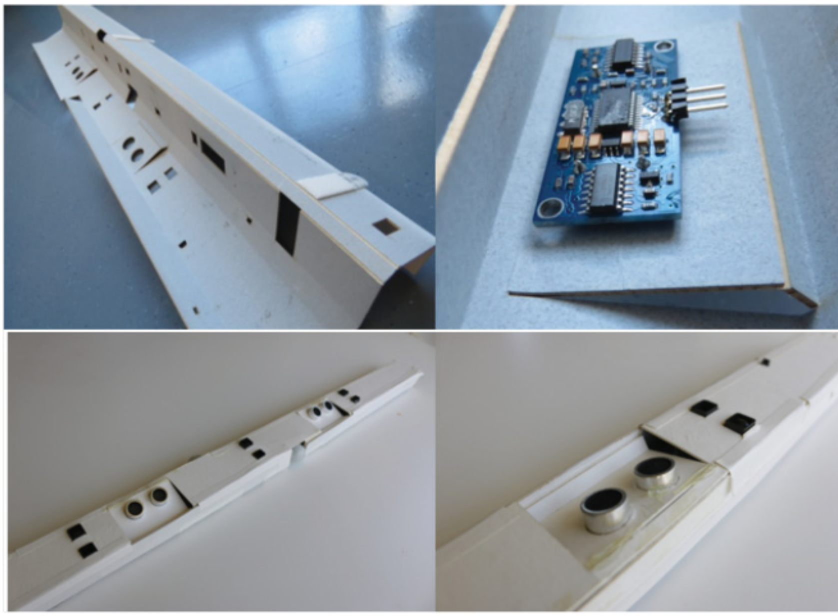

Fig. 2. The hardware enclosure was laser cut from backing board (top left). The PCB was placed inside (top right) along with sensors and the enclosure was folded. The pre-cut holes naturally hold all hardware firmly in place (bottom left). A folding technique achieved a consistent 20° tilt angle for ultrasound sensors (bottom right).

visual systems that can detect identity, such as external security cameras on the entrances and exits, or non-room-level carried-device-based system such as WiFi capable key or phone detection for when the occupant is on premises. While a resident may accept an external camera system to track their comings and goings, in-home visual tracking is generally unacceptable. In this article, we study the effect of both identity detecting and identity-less perimeter sensing, along with the absence of perimeter sensing, on tracking accuracy.

4 SENSOR IMPLEMENTATION

We implement two sensors to fulfill the design space: the *doorway sensor* (consisting of the presence, direction, room, and identity sensors) and *perimeter sensor*. We implement each zone from the doorway sensor design with a different physical sensor. Identity is obtained from the weak biometric height. For our implementation, all doorway sensors were built in a single enclosure that attached to the top of a doorway. A folding-based design was created for the enclosure and laser cut into a type of paper about 2mm thick called matboard, as shown in Figure 2. Pins were placed vertically across the entire enclosure to add stability and ensure the enclosure was square when folded. This prevented bumps from misaligning the sensors. When folded, some sensor ports fold up like an item in a pop-up book and hold the sensors at a consistent angle. The perimeter sensors, used only on external doorways, used a similar folded matboard design for their enclosure. The selection of the physical sensors for both the doorway and perimeter sensors is described in detail below.

4.1 Presence Zone Sensor

We implemented the presence zone sensor using three SHARP Infrared Distance Sensors (SHARP1487 2016) pointed downward from the top of the doorway. This sensor works well

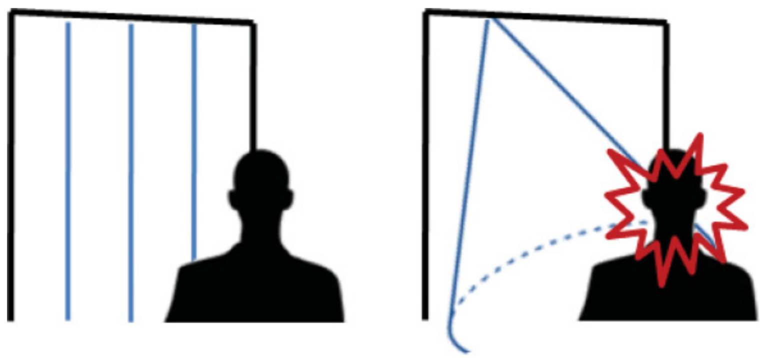

Fig. 3. The narrow-beam infrared range finders (left) detect people only in the presence zone while the wide-beam ultrasonic sensors (right) detect people outside of the doorframe in the direction zone.

for the presence zone, because it is narrow and can be directed to monitor only the presence zone (i.e., the threshold of the doorway) without detecting occupants in the direction or room zones. Each infrared sensor emits a narrow beam that can detect the distance to an object using angle of reflection. Each infrared sensor monitors a portion of the presence zone and together forms the presence zone sensor as seen in Figure 3. Each individual sensor is placed 15–18cm apart across the top of the doorway so at least one sensor is triggered when a person enters the doorway.

The infrared sensors deployed here measure in the 15–100cm range and have a precision of 1–2cm, although distances beyond 30cm are measured with decreasing accuracy. For our experiments this range is adequate, as we are only attempting to measure the upper half of a doorway were an adult's head would be. An implementation monitoring children would need a longer range sensor or an implementation that did not sense from the top of the doorway.

4.2 Direction Zone Sensor

PING))) Ultrasonic Distance Sensors (Parallax28015 2016) were used to detect the direction zone. These range finders have a wide, cone-shaped sensing region that detects the distance to an object based on time-of-flight of the signal. This is ideal for the direction zone, where a larger but still fairly contained zone needs to be monitored. Additionally, when placed at the top of a doorway, the conical sensing region will detect multiple measurements of a person as they walk through a doorway. When tilted at an angle of 20 degrees, this conical region can be used to obtain direction: when a person walks through the zone towards the doorway, the measured distance changes as they get closer to the sensor and indicates an approach. The person's absolute shortest distance reading will be when the person is inside the doorframe, because they are at the closest possible position to the sensor. Two ultrasound sensors placed evenly across the top of the doorway were used to detect the direction zone.

We leverage the conical nature of the ultrasonic sensor and its subsequent ability to sense direction with a single sensor to implement only one half of the direction zone. Alternate implementations might sense both direction zone regions.

4.3 Room Zone Sensor

The room zone sensor was implemented with the Parallax Passive Infrared (PIR) Sensor (Parallax28027 2016) with a sensing range of 7.5m. Two PIR sensors were used per doorway, each facing one of the adjoining rooms. A small amount of sensor overlap was possible in the threshold of the doorway, but otherwise the areas the sensors monitored were distinct. While this sensor might be used for the presence and direction zones if it could be made directional and limited to those smaller zones, it easily works out the box for the room zone sensor.

We leverage the room zone to disambiguate complex interactions with the doorway that are not actual crossings, such as someone reaching though a doorway to turn on a light. Two tests are performed on the room zone readings around any doorway crossing event. First, did the room zone sensors detect a person in both adjacent rooms in the 2s surrounding the event, indicating that someone might have crossed the threshold? Second, did the room zone sensors indicate that a person has actually crossed from one room to another, with greater indications of a person being in one room, followed by the crossing event, followed by greater indications of them being in the second room? These two tests are trained on data from a doorway sensor and combined using the naive Bayes statistical model. The probability of a true detection given these tests is then passed on to the tracking algorithm.

4.4 Identity Sensor

Our doorway sensors use height measured by the ultrasound sensors as a proxy for identity. This is a reasonable biometric in many homes due to the low likelihood of residents having similar heights (Srinivasan et al. 2010). However, in commercial buildings or homes with occupants of similar heights a stronger biometric should be chosen for implementation. We chose not to use the infrared sensors for this task due to their narrow beam angle, which would require a large number of sensors to be deployed to ensure the sensors measure the top of a person's head rather than a shoulder. Furthermore, the accuracy of infrared ranging depends greatly on the type of surface being measured; uneven surfaces such as human hair will not reflect a narrow beam signal cleanly. Because of the ultrasonic distance sensor's high ranging accuracy (+/−2cm) and the fact that they can reflect easily off the surface of a person's head, these sensors were chosen to provide height information for our implementation.

A height is detected from a doorway crossing event by clustering all ultrasonic readings from an event in time and height using DBSCAN(min_samples=1,eps=5)—a density-based clustering algorithm. From there, readings clustered at the start or end of an event that match readings outside of the event are removed. This helps filter measurements caused by non-people objects such as doors, as shown in Figure 6. The remaining clusters are compared to the time of the presence sensor firing, and the maximum height measurement temporally closest to the presence sensor is chosen as the height. However, if a person walks quickly enough, only one reading may be measured of their height and not placed in a cluster. To account for this a height measurement of an event has two readings: the height of the chosen cluster and the maximum reading from the event in the 90–180cm human height range. If there is no measurement that meets the above criteria, then the height is marked as "none." The detected heights are included in the final detected event output as shown in Figure 4.

4.5 Perimeter Sensor

The perimeter sensor was implemented using night vision cameras at all external doorways of the home. The same cameras were used throughout the home as ground truth data collection for evaluation, ensuring that the perimeter sensor was accurate for detecting both identity and presence of occupants. The cameras were constructed using Raspberry Pi Model Bs with attached infrared capable cameras (RaspberryPi 2016; RaspberryPiNoIR 2016). Diffused infrared LEDs were attached to the cameras to provided visibility in darkness and low light. The recorded video was then processed manually to extract a log of information about when occupants left and entered the home. This external doorway information was passed to tracking for its perimeter-based variant. While our implementation of the perimeter sensor is not ideal for a deployable system (manually

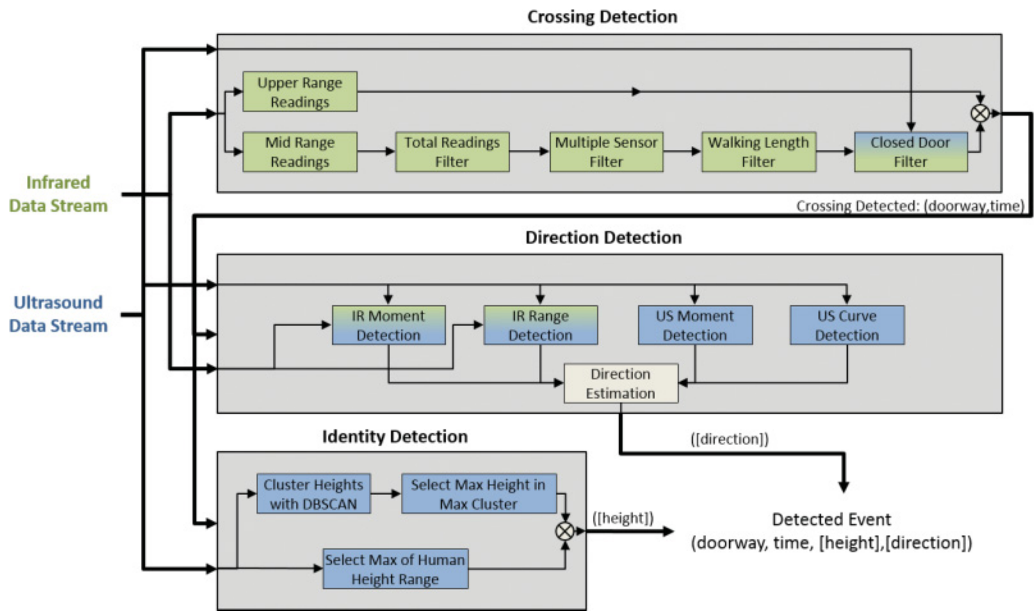

Fig. 4. A final event is detected through three processes, crossing detection, direction detection, and identity detection, to produce a final doorway event. Each process uses the ultrasound (US) and/or infrared (IR) data streams as input to its filtering and detection algorithms.

processing the video is not feasible long term), it allowed us to evaluate the effect of perimeter sensing on tracking accuracy. A full deployment might use computer vision or a carried device to identify individuals entering and exiting the home.

5 DATA PROCESSING AND TRACKING

A doorway tracking system process the data collected by the sensors in two major steps: doorway sensor event creation and person tracking. The main goal of our doorway sensor's event creation is to produce a set of events E where $e_i \in E = (doorway_i, time_i, [height_i], [direction_i],)$: a 4-tuple with information about what door the crossing occurred at, what time the crossing occurred at, a set of detected heights, and the probabilities of the two possible detected directions. The main goal of person tracking is to create tracks: a continuous sequence of room states of the occupants in the home. Tracking operates on the set of events produced by event creation and uses the deferred logic principle of Multiple Hypothesis Tracking (MHT) (Reid 1979) to correct any false positive, false negative, height, or direction errors in the events based on a holistic view of the events over time. An in-depth analysis of the general algorithm can be found in Kalyanaraman et al. (2016). The intuition behind such a deferred logic approach is that while a single doorway event can be erroneous, subsequent events can be used to correct errors by conforming data to a holistic track. For example, if a person moves through four adjacent rooms based on a multi-event view of their movements, an incorrect direction (e.g., one event inferred the person walking backwards through the second doorway) can be determined as unlikely and corrected to match the holistic track.

The doorway sensor's data fusion processes the raw sensor data in three components as shown in Figure 4: crossing detection, direction detection, and identity detection. These processing components create the crossing events $e \in E$. Tracking then processes these events into the holistic track. Such a tracking system has previously been designed to operate in a static manner (i.e., for a

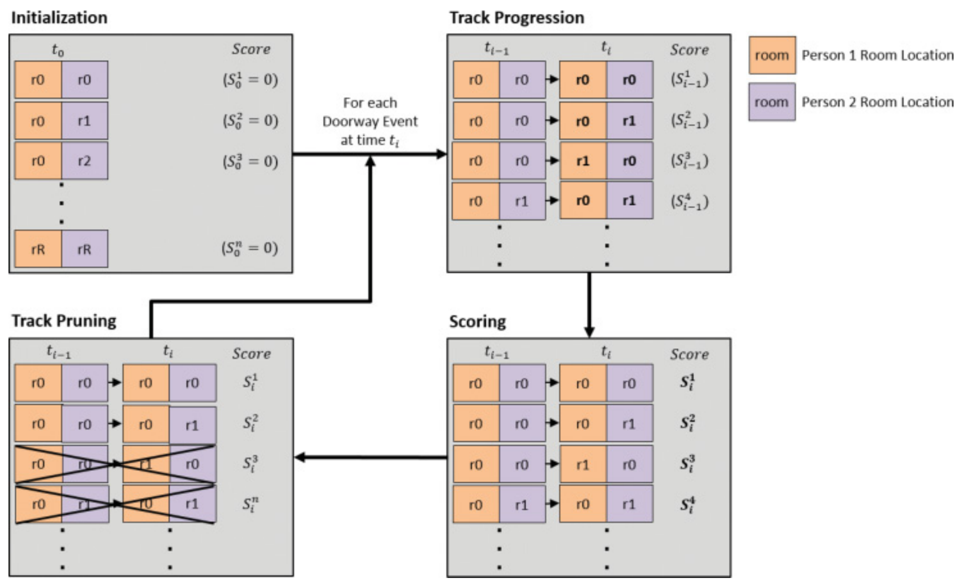

Fig. 5. Tracking first initializes all possible locations of people in the existing rooms (Initialization). Then, for each detected doorway event tracking expands the tracks of rooms states to incorporate that event (Track Progression), scores each new track based on its likelihood (Scoring), and prunes tracks that have low likelihood or identical end states (Pruning). In this way, tracking discovers the tracks of residents as they move from room to room.

fixed and known number of people) (Hnat et al. 2012). However, as mentioned in Section 3, it may not be possible to explicitly perform training for a static number of people, or assume that a static number of people are always moving throughout the home. Consequently, in this section, we also describe our dynamic tracking algorithms for multi-person tracking in homes. For dynamic tracking, tracking must first initialize potential tracks and expand those tracks based on new doorway events by considering all possible room movements. Since considering all possibilities results in an exponential growth of tracks, the algorithm must prune the least likely tracks to continue operation. This pruning requires a score function that ranks all current tracks. Therefore, tracking progresses using these four steps:

(1) *Initialization*: Defines the initial room states and potential tracks for tracking to consider.
(2) *Track Progression*: Expands the tracks based on each new doorway event.
(3) *Scoring*: Scores tracks based on their probability given doorway event data and the holistic view of the track.
(4) *Track Pruning*: Eliminates certain unlikely tracks to prevent state explosion of tracks. Additionally, selects the final track when tracking is complete.

An overview of these steps is shown in Figure 5.

We present the details of doorway sensor data fusion and event creation below. Additionally, we present two variants of dynamic tracking. First, we present an implementation of tracking without perimeter sensors, where the tracking algorithm has to implicitly estimate the number of persons in the home while simultaneously tracking individuals. Then, we present a variant of tracking in the presence of perimeter sensors where the number of people in the home is known to the tracking algorithm. For the former case, the tracking algorithm tries to explain the set of doorway

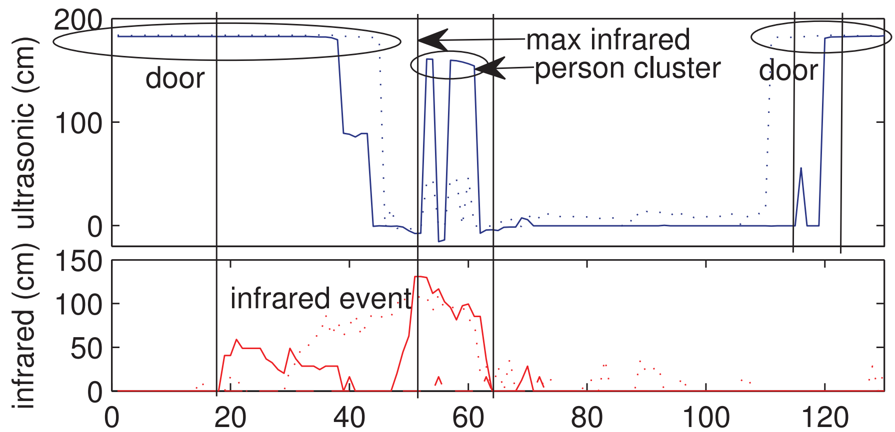

Fig. 6. An event is formed from successive infrared measurements. Height is inferred from the ultrasound cluster closest to the central infrared reading.

events with the minimum number of people. For the latter case, the tracking algorithm maximizes the likelihood of the data—that is, it chooses the track that best complies with the set of doorway events. We present both in detail below.

5.1 Doorway Sensor Data Fusion and Event Creation

To analyze the effect of different choices of sensing zones, we implement two different forms of data fusion between the zones: direction only and combined presence and direction. The direction zone only algorithm is pulled directly from the doorway tracking system Doorjamb (Hnat et al. 2012) and performs all processes necessary to doorway sensing with an ultrasonic-based direction sensor alone. The combined presence and direction algorithm is described below. Heights are obtained from the identity sensor as described in Section 4.4. Crossing detection and direction detection for the combined presence and direction zone algorithm are described below.

5.1.1 Crossing Detection. The goal of crossing detection is to detect the $doorway_i$ and $time_i$ of the event e_i. The presence sensor makes this a fairly straightforward process: any reading above the normal noise floor in the infrared stream must be caused by an object in the doorway. Hence, any sequence of abnormal readings produced by any of the three infrared sensors in a doorway is clustered together and marked as a candidate event. However, not all candidate events are actual crossing events in which a person walked through the doorway. Therefore, the key to crossing detection is to characterize, recognize, and reject false events as described below.

Readings above the normal floor that are not crossing events fall roughly into two main categories: environmental noise and human noise. *Environmental noise* occurs mainly from sensor noise, background noise (external lighting changes that effect infrared), and closed doors. Ideally, with the presence sensor, objects such as doors are not detected. However, doors in residential homes are not always perfectly perpendicular to the floor nor are the doorjambs always perfectly level. Slight imperfections in these areas can cause the infrared sensors to pick up a door when closed. *Human noise* occurs due to human behavior around the doorway. These events consist of a person entering the presence zone of a doorway sensor, but not actually crossing the threshold. This can occur for many reasons, including an arm opening or closing a door, an arm or shoulder brushing a door frame, or a non-crossing where a person enters the doorway but turns back into the room they came from. To remove false positives due to swinging arms, we simply merge events

Table 1. These Parameters for the Data Fusion Algorithms Are Derived from First Principles

Parameter	**Value**
upper range	0–24.5cm (10in)
mid range	24.5–50.8cm (20in)
mid_{length}	50s
$mid_{readings}$	5 triggers
$mid_{surrounding1}$	20 measurements
$mid_{surrounding2}$	40 measurements
mid_{max}	50.8cm (20in)
mid_{mean}	25.4cm (10in)
mid_{diff}	12.7cm (5in)
mid_{height}	120cm (47.3in)120
mid_{mode}	30
mid_{window}	0.5s

The same parameters can be used for all devices. Sensitivity analysis showed Low sensitivity to parameter values.

within 0.5s of each other. The rest of crossing detection deals with filtering out other sources of interference.

To filter out candidate events caused by environmental noise, we first define two sensor ranges for the infrared sensors. In essence, the infrared sensors measure how far away an object is from the top of the doorframe. We know that, typically, a person should be detected in the upper area, closer to the sensor itself. Hence, two ranges are defined, *upper* and *mid*, where readings in the *upper* range occur at about head height on an average adult and readings in the *mid* range are shoulder height or head height for a shorter person. Candidate events in the *upper* range are immediately labeled crossing events.

Candidate events in the *mid* range are closer to the floor and other environmental noise, so they must pass a number of filters to be considered crossing events. Therefore, candidate events in the *mid* range must have many supporting readings from the infrared sensors indicating that a human is in the doorway. This means there must be a good number of readings ($mid_{readings}$) for the *total readings filter*, there must be multiple infrared sensors that get triggered in and just outside the event for the *multiple sensor filter*, and, because crossing a threshold is an intuitively quick activity, the event must be fairly short (under mid_{length}) for the *walking length filter* for the event to be attributed to a person. Each of these filters can be seen filtering candidate events with mid range readings in Figure 4's Crossing Detection process. The values for these filters are specified in Table 1.

Closed doors also generate *mid* range candidate events and must be removed using a *closed door filter*. Data in the infrared readings that are indicative of doors have similar noise at the bottom of the *mid* range for long periods. Formally, a candidate event is too similar its surroundings if, when compared to the $mid_{surrounding2}$ surrounding measurements: the maximum infrared reading is within mid_{max} of the surrounding readings, the mean reading is within mid_{mean} of the surrounding readings, and the mean difference between successive readings is within mid_{diff} of the surrounding readings. Data in the ultrasonic readings that are indicative of doors shows "tall" (above mid_{height}), stable readings for a long period of time. More formally, if the mean of an ultrasonic sensor is above mid_{height} and the mode occurs more than mid_{mode}% of the time within the event and mid_{window}

seconds on either side of the event, then ultrasound indicates that there is a tall, stable, unmoving object present—a door.

The parameter values, presented in Table 1, were either chosen to represent human walking behavior or tuned based on samples of people walking through the doorway sensors. Parameter turning focused on doorways that had more complex physical setting (one or two attached doors) and was then applied uniformly to all doorway sensors in the studies. Doorways without doors are very robust to these parameter values, while doorways with doors are more sensitive. Duration-based parameters (mid_{length}, $mid_{readings}$, mid_{window}, $mid_{surrounding1}$, $mid_{surrounding2}$) were based on the general walking time of people (1.4m/s, where typical crossings take under 2s) and the sampling rate of the sensors (20Hz). Height-based parameters (*upper*, *mid*, mid_{max}, mid_{mean}, mid_{diff}, mid_{height}, mid_{mode}) were tuned using samples from recordings of the doorway sensors when the attached doors were closed, as well as samples of people walking through the doorway. While the effort required for tuning is a shortcoming of the system, all tuning was and can be applied uniformly to all doorways and need only focus on the noisiest (doorways with doors) doorways.

After a candidate event passes all the noise filters it becomes part of the final event set E. Crossing detection then fills in the $door_i$ and $time_i$ for each event $e_i \in E$ and passes this information on to direction and identity detection.

5.1.2 Direction Detection. Direction is determined by whether or not a person is detected by the direction zone (located on only one side of the doorway in our implementation) before they are detected in the presence zone. However, there are several ways to detect when someone is in the presence zone and the exact moment they crossed the threshold. As a result, we employ four direction algorithms. The output of each is passed on to tracking for further refinement. Each direction algorithm is described below.

(1) *IR Moment Detection.* This algorithm takes the time of the maximum infrared reading as the exact moment of crossing. This occurs when the infrared sensors detect the tallest object in the doorway. If a higher number of ultrasonic readings of heights occur before that exact moment, then the person entered the doorway from the side the ultrasound sensor is facing. We determine the absolute direction from this relative direction and knowledge of the sensor placement in the doorway.
(2) *IR Range Detection.* This algorithm considers the entire event duration, determined by the infrared sensor in crossing detection, to be the moment the person crosses the threshold. This provides a bound on when the person is in the doorway, rather than an exact moment. Like algorithm one, it then determines direction based on when the majority of ultrasonic readings occur relative to that period.
(3) *US Moment Detection.* This algorithm uses the intuition that the maximum reading in height detected by the ultrasound sensor should occur when the person is closest to the sensor, and therefore closest to the threshold of the door. This reading is defined as the exact moment of crossing. Again the algorithm determines direction based on when the most ultrasound readings occurred relative to the time of this reading.
(4) *US Curve Detection.* The fourth algorithm looks at the shape of the ultrasound curve. Because the ultrasound sensor is tilted, a person tends to cause a sequence of readings that begin low, gradually increase to their true height, and, within a reading or two, cease as the person exits the sensor's range. We determine direction from this pattern. If an ultrasound stream begins below 90cm in height, rises to the maximum height, then is non-increaseing for at least two values afterwards, we say that the person enter from the ultrasound side of the doorway. If the pattern is seen in reverse, then the person entered the doorway

in the opposite direction. To make the algorithm less strict and more noise resistant, the ultrasound stream is quantized into 2cm units and can have at most one noise value on the sharper side of the curve. We chose the 90cm threshold to be below the average human height and found it to be fairly robust (direction accuracy varied by at most 4% in the 40–100cm range).

The first two algorithms are generally applicable to a variety of direction and presence zone sensors. The second two algorithms rely on the particularly properties of the ultrasound sensors chosen for our implementation. From these four algorithms, direction detection creates the tuple $[direction_i]$, where the two entries represent the uniformly weighted probability of either an entry or exit. In this way, direction detection predicts the probable direction and passes this information on to tracking. Tracking then considers events occurring before and after this single event to better estimate and update this direction. Hence, the final direction of an event is calculated from the probabilities given by these four algorithms and tracking's holistic view of a person's track.

5.2 Dynamic Tracking without Perimeter Sensing

Without perimeter sensing tracking must estimate the number of persons in the home based on the detected events from the doorway sensors. This is challenging, because not all persons are sensed simultaneously—that is, unlike traditional multi-target tracking, targets (people) here are observed only when they transition from one room to another. Hence, one person may remain unsensed by any doorway sensor for hours at a time but still be present in the home. Tracking could explain these events by the number of people in the home equal to the number of doorway events (e.g., each person moves once and then never moves again). To mitigate this problem, tracking without perimeter sensing must minimize the number of people in the home, while still conforming their tracks to the doorway events.

5.2.1 Initialization. On start up, tracking first creates a hypothesized track for every possible combination of people in the rooms. In an R-room house with a maximum of P trackable persons, this results in the creation of $(R + 1)^P$ initial tracks (the +1 location is outside of the home). Hence, tracking starts with no knowledge of the number of people in the home or their location. It considers the possibility of each person in any of the "R+1" locations to be equally likely.

For example, consider a two-room {*r1, r2*} home (i.e., 3 locations with *"outside" (r0)*) having a maximum of three trackable persons {p1, p2, and p3}. The room location of each person is held in a 3-tuple (room of person 1, room of person 2, room of person 3). The initial set of 3^3 tracks, $t_1, \ldots t_n \in T$, created by tracking looks like the following:

$$t_1 : (r0, r0, r0) : \textit{All three persons hypothesized to be outside},$$

$$t_2 : (r0, r0, r1) : p1 \textit{ and } p2 \textit{ hypothesized to be outside}, \ p3 \textit{ is hypothesized to be in } r1,$$

$$\ldots,$$

$$t_{27} : (r2, r2, r2) : \textit{All three persons hypothesized to be in } r2.$$

This initial set of tracks is created only once. These tracks are then expanded for each new doorway event that occurs with a sequence of track progression, scoring, and pruning.

5.2.2 Track Progression. Tracks progress each time a new doorway event occurs. The progression expands each existing track in multiple ways, creating multiple new hypotheses. The multiple expansions represent possible direction and height errors caused by event detection. Additionally, they include the possibility that the event itself was a false detection. Hence, every new doorway event causes each existing track to duplicate itself and progress in the following manner:

(1) Someone inside the home has moved through the doorway of the event in either direction. They are now in a room adjacent to the doorway.
(2) Someone from the outside has come in and moved through the doorway of the event in either direction. There is now one more person in the home and they are in a room adjacent to the doorway.
(3) The detected doorway event was a false detection. The state of the home does not change.

Consider the two-room, three-person example with one exterior door leading outside (r0 ↔ r1). Upon detecting a doorway event between rooms r1 and r2, the track t_1 that ends in (r1, r2, r0) would duplicate itself (2P+1) times and progress them (in accordance with the above three possibilities) in the following way:

$$
\begin{aligned}
&t_1^1 : \ldots(r1, r2, r0) \xrightarrow{FP} (r1, r2, r0),\\
&t_1^2 : \ldots(r1, r2, r0) \xrightarrow{p1:\text{r1}\rightarrow\text{r2}} (r2, r2, r0),\\
&t_1^3 : \ldots(r1, r2, r0) \xrightarrow{FN} (r2, r2, r0) \xrightarrow{p1:\text{r2}\rightarrow\text{r1}} (r1, r2, r0),\\
&t_1^4 : \ldots(r1, r2, r0) \xrightarrow{FN} (r1, r1, r0) \xrightarrow{p2:\text{r1}\rightarrow\text{r2}} (r1, r2, r0),\\
&t_1^5 : \ldots(r1, r2, r0) \xrightarrow{p2:\text{r2}\rightarrow\text{r1}} (r1, r1, r0),\\
&t_1^6 : \ldots(r1, r2, r0) \xrightarrow{FN} (r1, r2, r1) \xrightarrow{p3:\text{r1}\rightarrow\text{r2}} (r1, r2, r2),\\
&t_1^7 : \ldots(r1, r2, r0) \xrightarrow{FN} (r1, r2, r1) \xrightarrow{FN} (r1, r2, r2) \xrightarrow{p3:\text{r2}\rightarrow\text{r1}} (r1, r2, r1).
\end{aligned}
$$

In this example, t_1^1 is the hypothesis that thinks the doorway event is a false detection. Tracks t_1^2, t_1^3, t_1^4, and t_1^5 move the people inside the home through the (r1, r2) doorway in either direction. Tracks t_1^6 and t_1^7 hypothesize that someone from the outside has come in and moved through the doorway in either direction. The false negatives inserted in the tracks ensure that the physical constraints of the home are obeyed (e.g., if a person is in r1, they must first walk into r2 before they can walk from r2 to r1). These hypothesized false negatives are inserted between the last detected event and the current one.

Furthermore, each hypothesis also next explores the possibility that someone inside has exited the home after the current doorway event via a missed detection. For example, hypothesis t_1^7 would duplicate itself three times (once for each person inside the home), and advance them the following way:

$$
\begin{aligned}
&t_1^{7'} : \ldots(r1, r2, r1) \xrightarrow{FN} (r0, r2, r1),\\
&t_1^{7''} : \ldots(r1, r2, r1) \xrightarrow{FN} (r1, r0, r1),\\
&t_1^{7'''} : \ldots(r1, r2, r1) \xrightarrow{FN} (r1, r2, r0).
\end{aligned}
$$

Such an exhaustive enumeration of possibilities ensures that all movements of people are considered. This allows tracking to judge when an event, direction, or height from the sensors was in error. Unfortunately, this expansion results in an exponential explosion of tracks. Said more precisely, given $(R + 1)^P$ initial tracks, upon processing d doorway events, $(R + 1)^P * (P * (2P + 1))^d$ tracks are created. This exponential explosion necessitates a need to maintain some tracks and delete others. Choosing one track over another results in the need of a scoring function.

5.2.3 Scoring. The intuition behind the scoring algorithm for dynamic tracking without perimeter sensing is to try and explain the events with the minimum number of people. As we model a variable number of people, maximizing the likelihood would always favor tracks with a larger number of people. This is because sensing errors can lead a track to introduce a *phantom person* (a person that does not correspond to an actual person in the real-world) into the home. Since a person can remain in a room for a long period of time without being observed, these phantom persons will be postulated to remain stationary until a doorway event occurs that no real person can explain. This creates a bias towards estimating a larger number of persons (as they explain the data better) and thus, the likelihood of any given data set will increase with a larger number of phantom targets.

To address this problem, we define a scoring function that penalizes a track based on the number of people it hypothesizes to be in the home. The intuition is to choose tracks with the minimum number of people required to explain the observed data. However, such a penalty function can result in idle real people (e.g., a person watching TV in the living room) being evicted out of the house to decrease the total. To mitigate this, we also penalize a track based on the number of moving people. More movers means a higher penalty. Hence, a track does not get penalized for having idle targets. Using the two penalty factors together prevents tracks from having different phantom people explain each doorway event. Note that this concept (the notion of person and mover penalty) could equally be incorporated into other tracking algorithms, such as the Hidden Markov Model (HMM) or particle filter. We chose to implement with the MHT because of computational tractability and ease of implementation, albeit at the expense of optimality (Kalyanaraman et al. 2016).

The scoring function incorporates these two penalties and any non-compliance with the set of doorway events to produce a score for each track. We refer to each non-compliance with a doorway event as an *"inferred error"* by a track. For example, if the detected doorway event says someone has moved "in" to the Kitchen, but the track moves a person out of the Kitchen, then it has inferred a direction error, and suffers a penalty. The other inferred error types are height errors (detected event says: height of the person involved is 173cm, but the track thinks the height of the person involved must be between 145–155cm based on past events), false positives (FPs) (track thinks the doorway event is a false detection), and false negatives (FNs) (track thinks there has been a missed doorway events).

To capture the aforementioned concepts, each track is scored according to the following score function:

$$S_{i+1} = S_i + (p + m + k) * \Sigma \epsilon_j w_j, \tag{1}$$

where: S_{i+1}: score of track after doorway event i+1,
S_i: score of track after doorway event i,
p: number of persons in the house during the error,
m: number of movers since the last inferred error,
k: constant offset to eliminate bias towards certain tracks,
ϵ_j: error penalty associated with the inferred error,
w_j: weight of the inferred error type.

We now explain each term in the above score function in detail:

— $\boldsymbol{\epsilon_j}$ *Penalty values*—Each non-compliance by a track with the given doorway event suffers a penalty depending on the inferred error type. Inferred direction errors are penalized based on the direction votes produced by the doorway sensor. Inferred FPs, FNs and height errors suffer a unit penalty. When the algorithm includes room zone sensing, tracking penalizes an inferred FP based on the probability of an event produced by the doorway sensor.

— w_j: *The weight of the inferred error type*—The weight of the error type aims to capture the likelihood of each error type across different inferred error types. For example, if missed detections are less likely than false detections, then we want the weight of an inferred FP to be lower than that of an FN.

— *p: The number of persons in the house during the error*—To capture our intuition of explaining the doorway events with the minimum number of persons, each track suffers a penalty of $p * \Sigma\epsilon_j w_j$. To eliminate any bias on the track that has all persons outside (i.e., $p = 0$), we use $(p + 1) * \Sigma\epsilon_j w_j$.

— *m: The number of movers since last inferred error*—To prevent static persons who are sleeping/ watching TV getting evicted out of the house in an attempt to minimize the number of people, we also penalize a large number of movers. As before, to eliminate any bias towards an all FP track, we use the factor of $(m + 1)$.

We include the constant value k to prevent bias toward tracks that keep people outside at all times ($p = 0$) and a track where no person moves ($m = 0$). To do this, we add 1 to both the p and m terms ($p + 1 + m + 1 = p + m + 2$ in our score function). We found that the addition of 1 to each parameter effectively removed this bias in our study, hence k was set to 2. This was the smallest value of k that eliminated the bias in our study. In general, the k value influences the number of errors the tracking algorithm can tolerate before incorrectly evicting a person from the home. More specifically, if there are P people in the home, an idle person in a peripheral room would get evicted out of the home after 2∗(P+k)+1 errors caused by the other persons (Kalyanaraman et al. 2016). For example, if there are four people in the home, with $k = 2$, a person seated in the peripheral room of a home will be evicted from the home after 13 errors are inferred in other people's tracks. Larger values of k will increase the minimum number of tolerable errors before evicting a person. Essentially, larger values of k would be required when the period of time (in terms of crossing events by other occupants) that a real person in the home remains sitting increases or if the error rates of the deployed doorway sensor are worse than those presented here.

5.2.4 Track Pruning. Once we have the scoring function, we can prune the hypothesized tracks (i.e., decide which tracks to retain and which to delete) to reduce the number of tracks held in memory. To ensure pruning does not remove the correct track moving forward, we only prune tracks that share their last state with another track. For example, if three different tracks end in the state *(r1, r2, r0)* (as seen in Section 5.2.2's example with tracks t_1^1, t_1^3, and t_1^4) then all tracks will progress forward in the same fashion. Therefore, only one such track needs to be retained for future track progression. Selecting the existing track to retain depends on the scoring function. By selecting the highest scoring track of all those that end in the same state, we ensure that the best track up to that point is retained. Then, tracking does not duplicate expanding that single state moving forward.

In addition to track pruning, we also "prune" states from within the tracks to decrease memory costs. This pruning is essentially a commit process, where tracking selects a prefix of states in the tracks that will not change with any new data and commits it to disk. Tracking commits a prefix of states when all existing tracks agree on a prefix (i.e., they agree on the room each person was in, from event e_0 to e_i). No tracks are removed in this process, just the selected prefix. After a commit, subsequent prefix checks occur from event e_{i+1}, since the states from e_0 to e_i are no longer held in memory. When tracking ceases, the sequential committed prefixes and the existing highest scoring track become tracking's inferred path of the household residents.

5.3 Dynamic Tracking with Perimeter Sensing

Tracking with perimeter sensing relies on the idea that the tracking algorithm can benefit from knowing the number of people in the home. This knowledge helps tracking as it reduces the algorithm's search space (i.e., the number of people that can potentially cause an event). Moreover, with a known number of people the tracking algorithm can perform data association by simply maximizing the likelihood of the data. The perimeter sensors that monitor the entry/exit of persons into the house obtain this person count for tracking. These perimeter sensors may or may not have a notion of identity (e.g., someone left the house via silhouette detection versus Bob left the house via a video camera sensor). Compared to perimeter-less tracking, track progression and scoring change the most with the addition of the perimeter sensor. The details of each step are described below.

5.3.1 Initialization. Initialization for with perimeter sensing is similar to the without perimeter version of this step with one main difference: perimeter tracking starts knowing the number of people in the home. Therefore, if there are three trackable persons, but only one is known to be in the R-room home, only R tracks are initialized (instead of $(R+1)^P$). If all trackable persons are in the home, then a maximum of R^P tracks are initialized. Like perimeter-less tracking this initial set of tracks is created only once. These tracks are then expanded for each new doorway event that occurs with a sequence of track progression, scoring, and pruning.

5.3.2 Track Progression. Track progression with perimeter sensing has one main difference from perimeter-less tracking: exterior and interior doorways are differentiated. Interior doorways are monitored by the doorway sensors, while exterior doorways are monitored by the perimeter sensor. Hence, every new interior doorway event causes each existing track to duplicate itself and progress in the following manner:

(1) Someone inside the home has moved through the doorway of the event in either direction. They are now in a room adjacent to the doorway.
(2) The detected doorway event was a false detection. The state of the home does not change.

Consider the three-person case in a two-room (r1, r2) house with one exterior door leading outside (r0 ↔ r1). Upon detecting an interior doorway event between rooms r1 and r2, the track that ends in (r1, r2, r0) duplicates itself 5 (= 2 * number of people inside + 1) times in accordance with the above two possibilities. Hence, interior doorway events progress the tracks in the following way:

$$
\begin{aligned}
t_1^1 &: \ldots(r1, r2, r0) \xrightarrow{FP} (r1, r2, r0),\\
t_1^2 &: \ldots(r1, r2, r0) \xrightarrow{p1:\mathrm{r1}\rightarrow\mathrm{r2}} (r2, r2, r0),\\
t_1^3 &: \ldots(r1, r2, r0) \xrightarrow{FN} (r2, r2, r0) \xrightarrow{p1:\mathrm{r2}\rightarrow\mathrm{r1}} (r1, r2, r0),\\
t_1^4 &: \ldots(r1, r2, r0) \xrightarrow{FN} (r1, r1, r0) \xrightarrow{p2:\mathrm{r1}\rightarrow\mathrm{r2}} (r1, r2, r0),\\
t_1^5 &: \ldots(r1, r2, r0) \xrightarrow{p2:\mathrm{r2}\rightarrow\mathrm{r1}} (r1, r1, r0).
\end{aligned}
$$

Like perimeter-less tracking, false negatives are inserted in the tracks to ensure that the physical constraints of the home are obeyed. When the perimeter sensor senses an exterior doorway event, each track either brings in or evicts a person from the home based on sensor data. The tracking algorithm now starts tracking an extra person (in-case of entry) or stops tracking a person (in event of exit). This entry or exit can included identity information depending on the perimeter sensor used.

Without identity sensing, tracking must expand tracks to consider the movement of each possible trackable person in the home. For example, upon a perimeter sensor exit detection tracking expands each track by the number of current occupants to hypothesis each person leaving the home. For a track that ends in (r1, r2, r0) with two residents inside the home, this progression looks like:

$$t_1^1 : \ldots(r1, r2, r0) \xrightarrow{\text{r1}\rightarrow\text{r0}} (r0, r2, r0), \tag{2}$$

$$t_1^2 : \ldots(r1, r2, r0) \xrightarrow{\text{r1}\rightarrow\text{r0}} (r1, r0, r0, \tag{3}$$

With identity sensing, tracking expands each track only once such that the detected person is moved outside. For instance, in the same example as above, if the perimeter sensor says p1 has exited, then only hypothesis t_1^1 is generated.

While perimeter sensing limits the number of tracks to be considered, the expansion of interior tracks is still extensive. Hence, an exponential explosion of tracks arises from tracking with perimeter sensing. Given R^P initial tracks, upon processing d doorway events, up to $R^P * (P * (2P + 1))^d$ tracks can be created. As with perimeter-less sensing, choosing one track over another necessitates a scoring function.

5.3.3 Scoring. The addition of the perimeter sensor greatly simplifies the scoring of tracks. Since there are a known number of people inside the house, we can score tracks by maximizing the likelihood of the data—that is, we choose the track that best explains the data (the one with the fewest non-compliances). This removes the notion of movers (demoted by m in perimeter-less scoring) and eliminates the need for the constant factor k. Hence, the perimeter sensor tracking scores tracks based only on their inferred errors. This creates the following score function:

$$S_{i+1} = S_i + \Sigma \epsilon_j w_j, \tag{4}$$

where: S_{i+1}: score of track after doorway event i+1,
S_i: score of track after doorway event i,
ϵ_j : error penalty associated with the inferred error,
w_j: weight of the inferred error type.

The description and calculation of the error penalty and weights is identical to the perimeter-less tracking case in Section 5.2.3. Without the need to determine the parameter k, this perimeter-based tracking can more easily generalized to different compositions of idle and moving people in the home. Additionally, since perimeter-based tracking cannot incorrectly determine the number of people in the home, it is more likely than perimeter-less tracking to correctly filter sensor errors and determine the true path of occupants in the home.

5.3.4 Track Pruning. Tracking with perimeter sensing performs the same pruning as perimeter-less tracking. It prunes tracks with the same end state and commits state prefixes that are shared between all existing tracks. As with perimeter-less tracking, we choose the sequential committed prefixes and the existing highest scoring track to become tracking's inferred path of the household residents.

6 EXPERIMENTAL SETUP

To assess what level of design implementation is needed for smart home applications, we evaluate the design choices in controlled and *in situ* environments. Both forms of experiments were performed at the same location, using the same ground truth and evaluation. The test home was

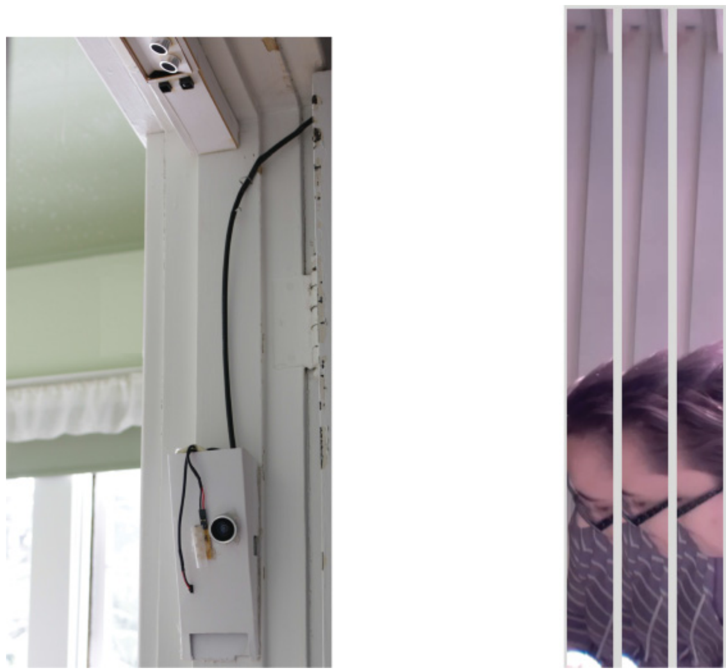

Fig. 7. The cameras (left) were placed in each doorway to record a narrow video (right) of only the doorway interior. This was used for ground truth detection of when people crossed between rooms.

instrumented with doorway sensors and ground truth cameras on every door as shown in Figure 7. Data is collected from the doorway sensors using off the shelf SNAPpy RF200 microcontrollers made by Synapse Wireless (RF200 2014). The RF200 provides wireless connectivity to a basestation located within the home. Readings are collected from the sensors at a rate of 20Hz and logged to the basestation. This data was then forwarded to a server external to the home for analysis. Each doorways has a 120cm threshold with none (four doorways), one (five doorways), or two (two doorways) attached doors. In general, any doorway with an 8–12cm space available for sensing where a door does not pass under the presence sensor can employ these sensors. Any configuration where the door swings under the presence sensor (sliding door, double hinged, revolving, or just mounting the sensor on the outside of the doorframe) can be addressed by sensing the door itself, for example, with magnetic reed sensors.

6.1 Ground Truth

Ground truth for the experiment was collected using cameras at each doorway in the home (Figure 7). To ensure that participants were as comfortable as possible and lived as naturally as possible the view of the cameras were restricted in hardware to only the doorjamb of the doorway. The cameras were constructed using Raspberry Pi Model Bs with attached infrared capable cameras (RaspberryPi 2016; RaspberryPiNoIR 2016). Diffused infrared LEDs were attached to the cameras to provided visibility in darkness and low light. These LEDs did not interfere with the infrared sensors on the doorway sensors. Recorded video was synced at 3 a.m. to a server external to the home. The recorded video was then processed to extract information about each crossing made by the participants, which participant made the crossing, and the direction they were traveling in. The recorded video also allowed for the identification of non-crossing doorway

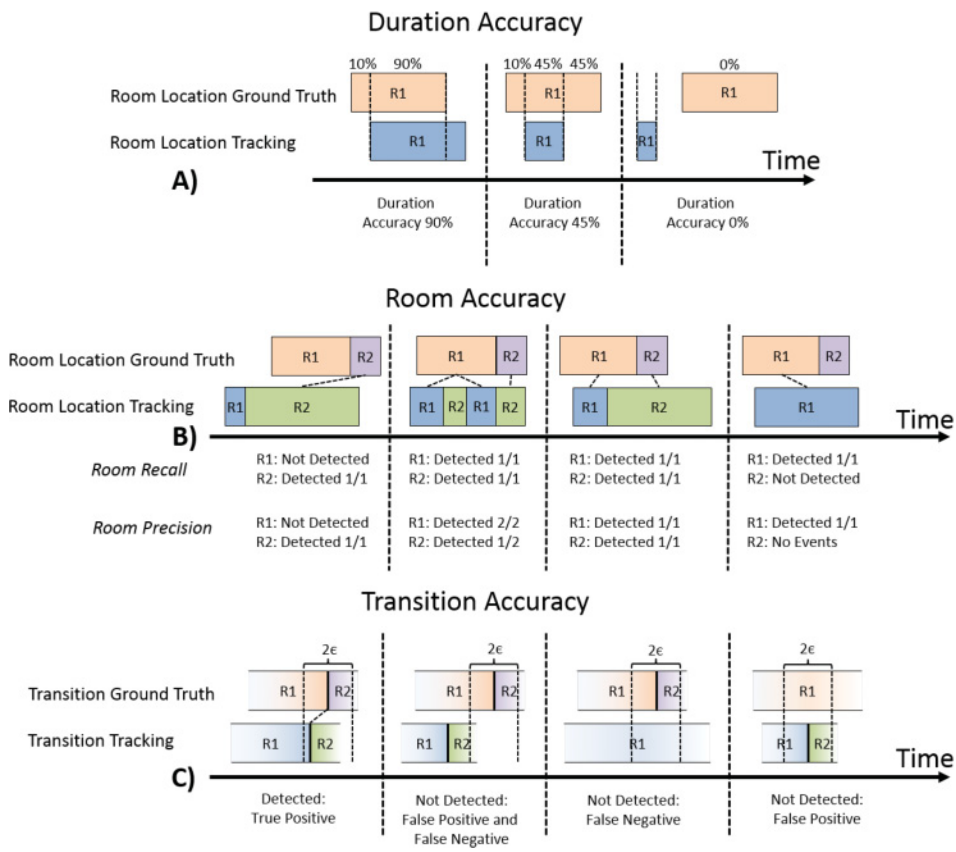

Fig. 8. Examples of the evaluation metrics (a) duration accuracy, (b) room accuracy, and (c) transition accuracy. Each metric allows an error of ϵ when comparing the time of ground truth an accuracy—though only (c) has this error diagramed.

activity, such as locking the front door, reaching in to close and open doors, hovers, and when the two participants crossed a doorway simultaneously.

Timestamps in the collection of video data skewed for each camera across any given day. Timestamps for ground truth were corrected by correlating the camera events with other sensors collecting data in the home. This ground truth is converted into a ground truth track: an ordered list of the times a person was in each room of the home.

6.2 Evaluation Metrics

We use three application class metrics to evaluate the potential design decisions of a doorway tracking system: room accuracy, duration accuracy, and transition accuracy. These metrics evaluate the questions: "Can the system detect if a person has been in a room?", "What percent of the time does the system detect the person in the right room?", and "How often does the system detect when a person moves from room to room?", respectively. Examples of the metrics are presented in Figure 8. When comparing timestamps between ground truth and the doorway tracking output an error of ϵ (15s) is allowed. Each of these metrics is described formally below.

The room accuracy metric evaluates if a person is detected at all in the correct room during the time they were in the room. This is calculated as the F-score of the room accuracy's room precision and room recall. *Room recall* is defined as the number of room occupancy periods (the time a person is in a specific room) in ground truth where tracking correctly places the person in that room at least once during the period, divided by the total number of room occupancy periods in ground truth. *Room precision* is the complement to room recall, where room occupancy periods found by tracking are evaluated. Examples of both values are shown in Figure 8(b). More formally, room recall is defined as the number of room occupancy periods in tracking where ground truth has the person in that room at least once during the period, divided by the total number of room occupancy periods in tracking. The final equation for room accuracy then becomes:

$$\text{Room accuracy} = \frac{2 * \text{room precision} * \text{room recall}}{\text{room precision} + \text{room recall}}.$$

The second metric, duration accuracy, takes into account the time spent in a room when determining accuracy. It evaluates how often, in second granularity, a person was placed in the correct room by tracking. Formally, we calculate duration accuracy to be:

$$\text{Duration accuracy} = \frac{\text{time in correct rooms}}{\text{total time in rooms}}.$$

The final metric, transition accuracy, evaluates how accurate a doorway tracking system is at detecting transitions. Such a metric requires the mapping of detected events to ground truth events despite timestamp differences between the two systems. We use a bipartite matching algorithm (Kalyanaraman and Whitehouse 2015) for this. We compare each event matched with ground truth and mark it correct only when the person identity, traversed doorway, and event direction are correct. *Transition recall* is defined as the number of these correct events, divided by the total number of ground truth transitions. *Transition precision* is defined as the number of correct events, divided by the total number of detected transitions. We define transition accuracy as the F-score of these two metrics:

$$\text{Transition Accuracy} = \frac{2 * \text{transition precision} * \text{transition recall}}{\text{transition precision} + \text{transition recall}}.$$

6.3 Controlled Study Design

Controlled studies were performed to test the hardware, ground truth collection, and provided additionally data for evaluation. In total, two controlled experiments were conducted. The first consisted of two participants (163 and 180cm) walking around the test home with 40 crossings per person per doorway (Controlled Study #1). The second study had three participants (152cm, 163cm, and 175cm) distinct from the two-person study with 20 crossings per person per doorway (Controlled Study #2). Participants kept a tally of how many times they had walked through a specific doorway on a mobile phone. Participants were asked to enter and walk around the test home as naturally as possible. They were asked to manipulate doors, including partially opening and closing the doors, and walk through the home in no specific pattern. Ground truth for Controlled Study #2 failed on one of the doorways in the test home. The data for this doorway was removed along with the leaf doorway adjacent to it, creating a layout of eight rooms. In total, 589 crossings were recorded for Controlled Study #2 and 800 crossings were recorded for Controlled Study #1.

6.4 *In Situ* Study Design

Participants were recruited from the university to live inside the instrumented home for a period of 3 weeks. We specifically recruited participants that would actively live in the home, cooking, showering, and sleeping there on a daily or near-daily basis. They were informed of the locations and general purpose of the sensors in the home, but were not informed of the specific details of how the sensors tracked or identified them. Participants were asked to go about their lives as naturally

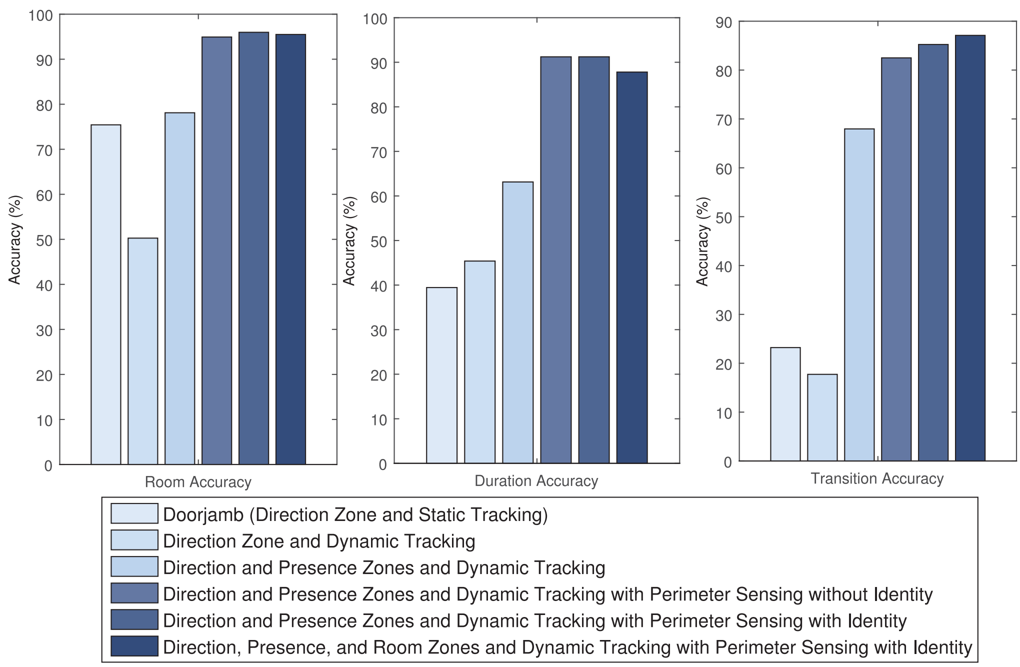

Fig. 9. Results from the *in situ* study show that each type of accuracy responds differently to the design space. Room accuracy and duration accuracy increase quickly as more design components are added transition accuracy, the most strict of the evaluation metrics, requires all design components to achieve peak values. We applied the training-based doorway tracking system Doorjamb to our *in situ* data as a baseline for comparison (Hnat et al. 2012). Direction Zone and Training-less Tracking implements the Doorjamb sensor design with training-less tracking and each bar after that cumulatively adds a design component.

as possible for the period of time they were living in the instrumented home. No participation beyond this was requested from the participants for this experiment. The two participants were a couple, one 152cm and the other 188cm. To help offset relocation costs, each participant was paid $200.

In a few cases throughout data collection the cameras went down and ground truth was lost, researchers went to the home for sensor maintenance, or the doorway sensor data failed to record. Because of this, only six days of the collected data is presented here. Of those six days, 6.7% of the data collected across all doors was lost during wireless communication. This corresponded to 1.7% lost crossing events (34/1,961) for the entire *in situ* study. We did not clean the data to remove these data losses. Instead, we mitigated the effect of data loss during tracking by not penalizing tracks for a false negative when an inferred crossing occurred during a data loss event. Such a data loss event had to be greater than 6s. We derived the 6s limit from the maximum amount of time in took any of our controlled participants to cross the doorway. With this data loss window for inferred events, tracking can handle a low percentage of lost data (such as the 1.7% of events in the study) that may be common in real world deployments.

7 RESULTS

We present the main results of our analysis in Figure 9. Each bar in the graphs shows a different selection of design components that have been implemented. The first bar represents the baseline

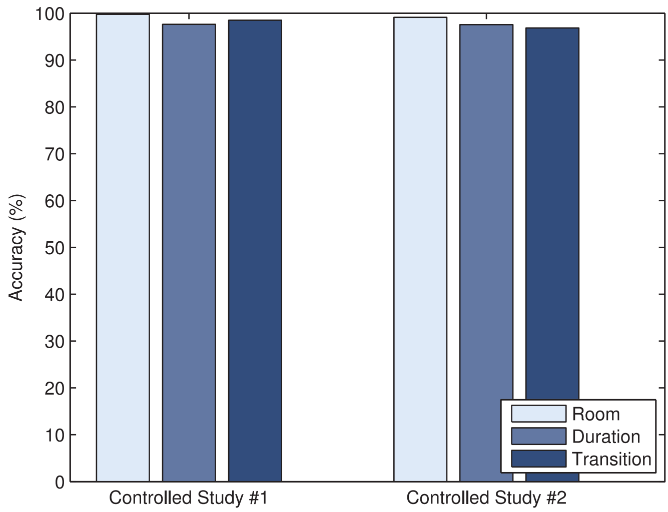

Fig. 10. In a controlled setting the doorway tracking system was easily able to track people with high accuracy. The results here are for a training-less perimeter-based tracking system, implementing the presence and direction zone sensors.

for comparison in doorway tracking algorithms, the training-based, direction zone only system known as Doorjamb (Hnat et al. 2012). Each following bar represents a point in the design space of a training-less doorway tracking system, beginning with the same direction zone only hardware design as Doorjamb (Direction Zone and Dynamic Tracking). The following bars add the presence zone sensor, perimeter sensing without identity, perimeter sensing with identity, and the room zone sensor cumulatively to the implementation.

Overall, results show that the inclusion of the presence zone and perimeter sensing provide the largest increase in accuracy for a training-less doorway tracking system. Both room accuracy and duration accuracy peak with these design elements, though room accuracy achieves nearly 78% with only the addition of the presence zone. Transition accuracy improves with every added design element. The doorway tracking system can track residents in a home with peak accuracies of 96%, 91%, and 87% accuracy for the room, duration, and transition metrics, respectively.

8 ANALYSIS

In this section, we analyze and discuss the effect of the design components on the accuracy of a doorway tracking system.

8.1 Controlled Studies Versus *In Situ* Behavior

We include select results from the controlled studies in Figure 10 as a basis for understanding the results found in the *in situ* studies in Figure 9. Unsurprisingly, the controlled studies show higher accuracy in all metrics. When transitioning this system from our controlled environment to a real environment, we gained many insights about the behavior of people and objects around a doorway as they interacted with our sensors. In preliminary testing, we deployed a version of the doorway tracking sensors in four homes for three months or longer per home and performed substantial testing in the lab. We could not perform an *in situ* evaluation due to the lack of ground truth, but our anecdotal experiences revealed several insights about the *in situ* environment.

First, non-human noise greatly increases in an *in situ* setting, particularly with non-intrusive range finding sensors like the ultrasonic and infrared sensors we used in our implementation. This noise occurs mainly from sensor noise, background noise (external lighting changes that effect infrared), and closed doors. Ideally, with the presence sensor, objects such as doors are not detected. However, doors in residential homes are not always perfectly perpendicular to the floor nor are the doorjambs always perfectly level. Slight imperfections in these areas can cause the infrared sensors to pick up a door when closed. Additionally, wood trim or doorknobs can reflect signals that can appear to be a human head, or simply create dynamic background noise.

While sensor and environmental noise may be filtered by more extensive sensor selection, the noise we found most common and troublesome for the doorway tracking system was caused by the people and not the environment. People interact very differently with a doorway in their own home compared to a controlled setting. Particularly, people don't only walk through a doorway to get from one room to another. The doorway tracking system can detect bags, arms, or other objects that are not whole people entering the doorway. This is particularly common when people put an arm through a doorway to shut a door or flip a light switch. Additionally, a person might walk close to a doorway but not through it, as is common in hallway doorways, brushing an arm or shoulder on the door frame. At times, the person might enter the doorway completely, perhaps to stand and talk to someone in the other room, but then exit the way they came. All of these interactions are uncommon in a controlled setting, but appear frequently and in a wide variety in a real home.

8.2 Variations in Occupant Behavior

While the controlled studies showed very consistent error rates, even across individuals, the *in situ* study showed highly variable errors over time even within the same two individuals. Essentially, clusters of errors appear in a real environment when a person's interactions with a doorway or the environment around a doorway changes for a period of time. This change in behavior is unique to the individual and the specific moment in time. A person might suddenly decide to pace right next to a doorway in the hallway, or run across the household at a speed that makes it difficult for the doorway sensors to detect a person in the noise given the present sampling rate. This difference between a controlled and real environment can be seen in Figure 11, where a boxplot shows the variability of the number of errors seen in a 20-event sliding window over the course of the studies. While the controlled studies are very consistent, ranging between 0 and 2 errors in 20 events, the *in situ* study is highly variable even across days. Indeed, the box plots for *in situ* days show several cases in which more than half of the events in a 20-event window are either false positives or false negatives. A tracking algorithm must be able to deal with this variability to operate in a real environment.

Occasional interactions between individuals in a doorway also occurred in the *in situ* setting, but did not greatly impact accuracy. These interactions included walking side by side through the doorway or crossing quickly in succession (e.g., with less than 0.5s between them). In these scenarios, only one person may be inferred by the doorway sensors and the other missed. This behavior was not prohibited in the *in situ* study (at least one such incident was recorded) but did not occur frequently. Hence, these behaviors did not have a major effect on our results but could effect accuracy in applications where the behavior occurs more frequently.

8.3 Adding the Presence Zone

The inclusion of the presence sensors helps correctly filter a significantly number of noisy doorway interactions at the doorway sensor level in both the controlled and *in situ* studies as shown in Figure 12. In the controlled studies included here, we attempted to induce some of the noise found

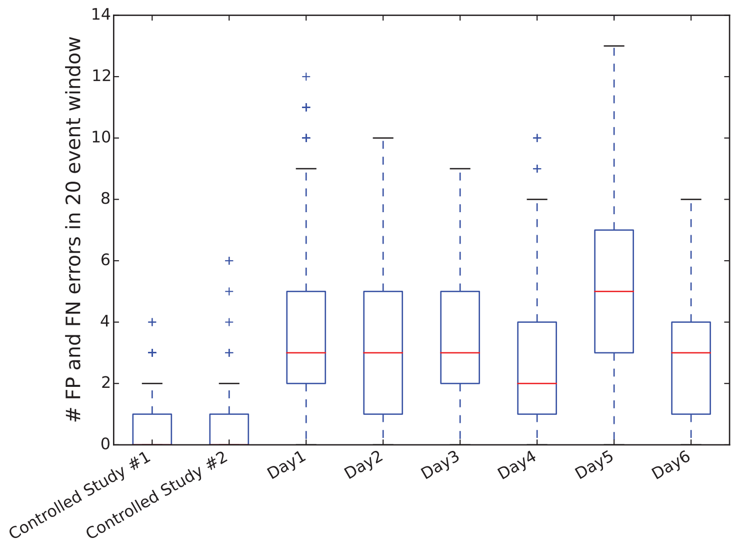

Fig. 11. Even though the average precision and recall for *in situ* was comparable to the controlled studies, the *in situ* data had many bursts of eight or more false detections or missed detections within a 20-event window.

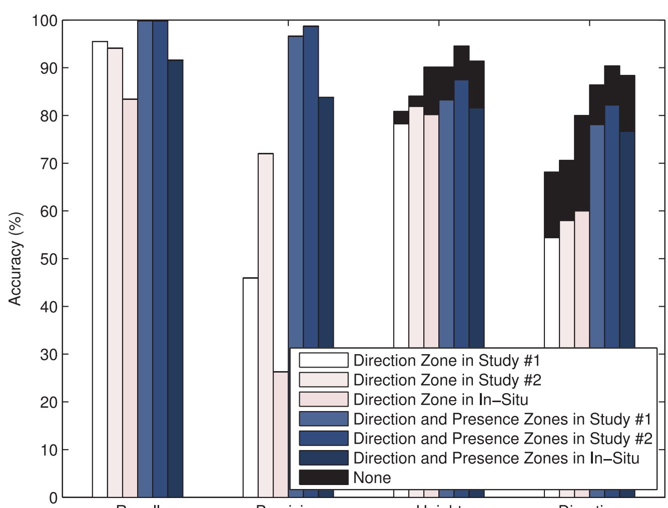

Fig. 12. In both the controlled (designated Study # 1 and # 2) and *in situ* studies the inclusion of the presence zone greatly increased the precision and recall of detected events produced by the doorway sensor. How often the direction of a true transition was correctly detected also improved significantly. This improvement in the doorway sensors detections feeds into the improved results in tracking.

in an *in situ* environment by asking participants to open and close the doors frequently as they walked around, particularly in Controlled Study #1. Additionally, we noticed that reflections off surrounding furniture caused false detections in all studies. However, including the presence sensor greatly reduces false detections as only objects that enter the doorway, be they people crossing or not, can be considered as transitions.

Direction estimation was also greatly improved with the presence sensor. This indicates that even when used for its intended purpose, direction, the ultrasonic direction sensor is inadequate alone. This may in part be caused by the single sided implementation of the direction zone used here. By implementing only one side of the doorway with the direction zone, we limit the area in which direction may be determined by half. Fusing the estimated direction with the temporal information of the presence zone (i.e., did the person enter the presence zone before or after the direction estimation?) allows the doorway sensor to more accurately determine the direction of a crossing.

Because the presence zone sensor plays a large role in the resulting accuracy of the doorway sensor, any limitations of the sensor can greatly affect the whole system. This was particularly apparent in the *in situ* study where one participant height was at the very edge of the presence sensors vertical sensing range. In the *in situ* study 21% of all of the shorter participant's events were missed by the doorway sensors, that is the presence sensors data fusion was unable to differentiate the participant from background noise. In comparison, only 5% of the other, taller participant's events were missed. Hence, our implementation of the presence sensor would not have been able to perform tracking on short people, including children. Another sensor choice or deployment location, such as at waist height, would mitigate this problem in future implementations.

8.4 The Effect of Perimeter Sensing

Knowledge of the number of people in the home greatly increases the accuracy of tracking in a training-less situation. As a result of this knowledge, the tracking algorithm can better reason out the errors happening within the home and prevent assignment of observations to phantom persons (persons tracking thinks are in the home, but actually aren't). The inclusion of perimeter sensing increases the three accuracy metrics room, duration and transition by 16%, 28%, and 14%, respectively, as shown in Figure 9.

Perimeter sensing can either detect the identity of a person entering/exiting the home, or detect that person without knowing their identity. At most, adding identity information increased accuracy by 2.5% in our *in situ* study. This increase was due to identifying exactly who was leaving the home, and at what time, when the two participants left together. When there is a sufficient time interval between the entries or exits of occupants, the internal doorway tracking algorithm can infer the identity of the entered (or exited) person using the heights of the occupants of the home, as they move around. This indicates that, especially in homes where occupants leave at different times, a perimeter sensor with no identity information is sufficient. Additionally, this means that guests entering the home (that the tracking system has no identity for) can be tracked. If the biometric used for identity can differentiate between a resident and a guest (i.e., they don't have exactly the same height), then the tracking algorithm can track the residents in conjunction with guests.

8.5 Filtering Complex Behaviors with the Room Zone

As the controlled and *in situ* studies were performed in the same test home, with no adjustments to the doorway sensors between studies, a drop of nearly 20% in precision indicates that more than doors or sensor noise were causing errors in the *in situ* case. We found many of these were related to the actions of the people, particularly as it related to a specific doorway in the home with a low, gate-like wall as show in Figure 13. When the person paced in front of this doorway the doorway sensor was unable to filter these events as they exhibited sensor data similar to a crossing event. This continuous pacing created as high as 25 false positives in a window of 29 observations. These false detections bring down transition accuracy without greatly impacting room or duration accuracy, since they often occurred in quick succession.

The addition of the room zone sensor allows tracking to filter these events by providing additional information on how likely any event created by the doorway sensor is to be an actual

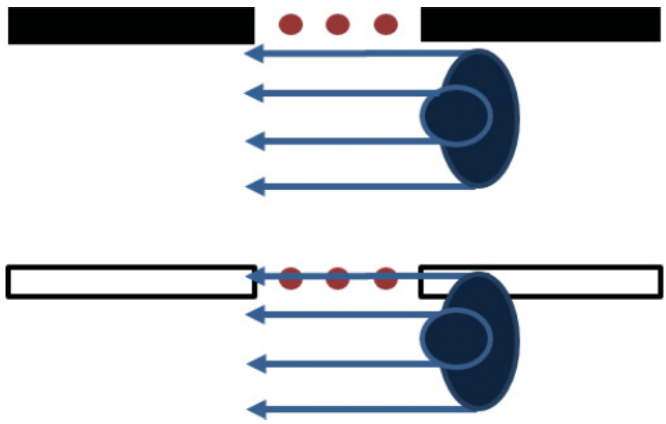

Fig. 13. One doorway was actually an opening in a low gate. This allowed residents to walk along the gate with one arm above it that enters the presence zone of the doorway (bottom) in a way that is much less likely with a full-height wall (top).

transition. If the room zone sensor detects a person only in one room, despite detected events at the doorway, then an actual transition is unlikely. We evaluate the upper bound of this intuition by training the room zone sensor per doorway on *in situ* data and performing a leave-one-day-out crossfold validation. The results of this are shown in Figure 9. We noticed that the addition of the room sensor was particularly useful in the two days with the most number of back-and-forth movements, namely Day1 and Day5 as shown in Figure 11. The addition of room sensing increased the transition accuracy by 9% and 4% respectively on these days. However, despite these two improvements, even training on per door *in situ* data only increased transition accuracy overall by 2%.

While the room zone sensor increased accuracy in terms of transitions, it caused duration accuracy to fall. The room zone sensor will sometimes erroneously filter true crossings if the event does not have sufficient room sensor support. When this event is followed by a long period of inactivity, duration accuracy can fall significantly. However, this has little effect on transition accuracy, since only 1 or 2 true events are filtered. This drop in duration and room accuracy, in addition to the only marginal increase in transition accuracy, showed that the room zone sensor was of little benefit to a doorway tracking system unless high transition accuracy is required for the final application.

8.6 Sensing Identity

Our chosen weak biometric of height was able to effectively distinguish between individuals in our studies. Figure 14 shows a histogram of the detected heights of participants in the three person controlled study. The participants can be easily told apart in the peaks, allowing tracking use this as an identifying feature. Because we are aware of the limitation of height as a biometric (many people have the same height) only individuals of dissimilar heights were chosen for our studies to ensure the limitations of our chosen biometric didn't greatly influence our design space analysis. All participants in the studies were easily differentiated by height.

To analyze the effect of our selected biometric in real world conditions, we also performed a sensitivity analysis to understand the effect of height difference on tracking accuracy. Figure 15 shows the accuracy for all three metrics for occupants with varying height differentials. The results are shown for the highest accuracy design implementation: Direction and Presence Zones and Dynamic Tracking with Perimeter Sensing with Identity. We created these results by simulating an increase or decrease in height for one of the *in situ* study participants using ground truth to identify

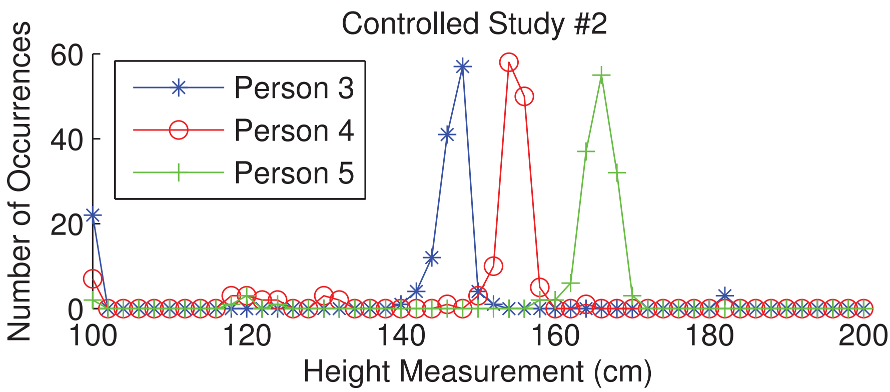

Fig. 14. This histogram shows all recorded heights for participants in our three person controlled study. The majority of the measurements for each person clustered around their actual height and are easily distinguishable from other participant's peaks allowing for identification based on height.

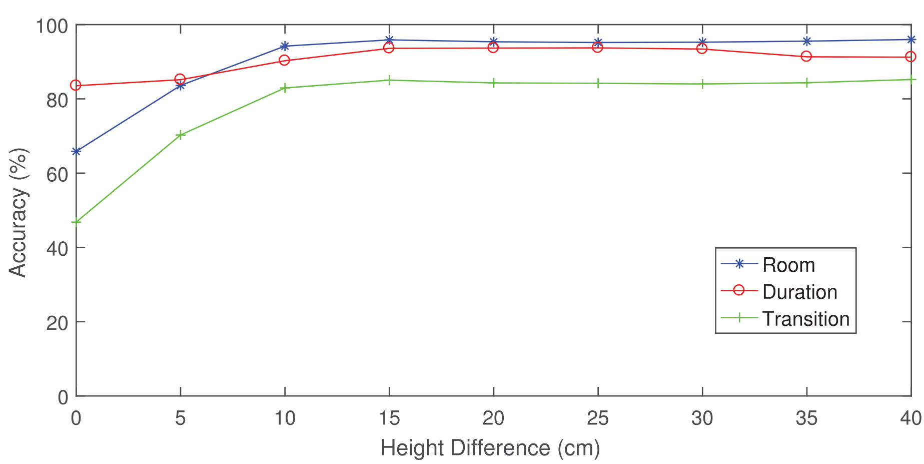

Fig. 15. With a difference of 0cm the tracking accuracy is essentially random but quickly improves, reaching its best performance with only 10cm (4in) of height difference. Height differences beyond that do not further improve accuracy.

which events belong to each individual. Two accuracies were calculated for each difference in height by holding one person's height steady and either increasing or decreasing the other's height. The average of the two runs for each height difference is presented in Figure 15. We simulated the results in this manner, because re-running the *in situ* study for each difference in height was infeasible. Hence, false positives retain the height of whatever caused them (e.g., resident or door) while true positives are modified to the simulated heights before tracking processes them. We present the average result when holding either height steady to include the effect of false positives as much as possible.

With a difference of 0cm the tracking accuracy is random but quickly improves, reaching its best performance with only 10cm (4in) of height difference. Height differences beyond that do not further improve accuracy. At a 0cm difference the system has 65% room, 83% duration, and 47% transition accuracy. Duration maintains high accuracy, because participants in our study both

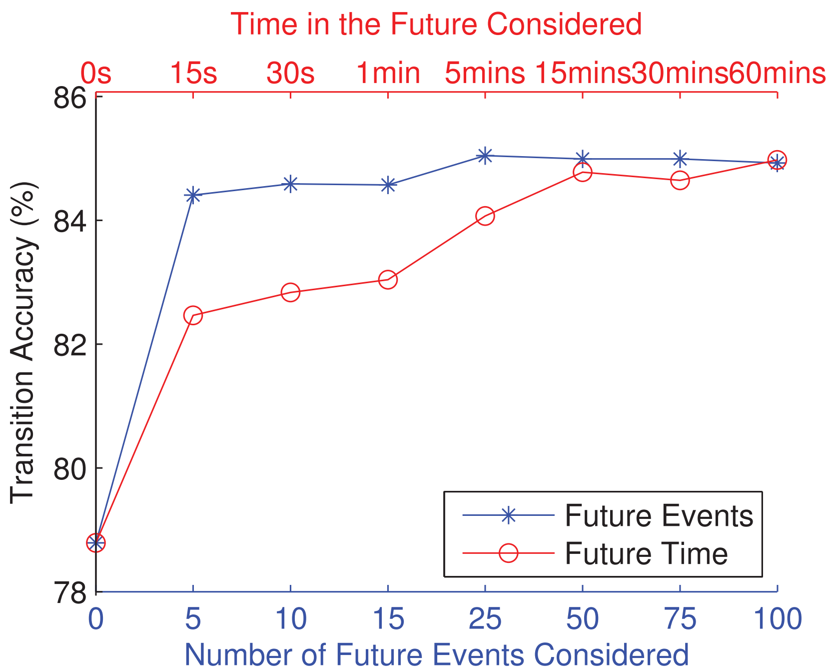

Fig. 16. A doorway tracking system also has the potential to perform real time tracking. Without any future information, tracking can correctly identify 79% of transitions. With only a 15s or five-transition event delay this accuracy jumps to between 82% and 85%. Potentially, this means tracking can provide lighting or HVAC control in real time.

spent the majority of their time in two rooms together (kitchen and bedroom). Transition accuracy suffers the most as tracking can no longer differentiate the two residents. At a 5cm difference, accuracies are 84% room, 85% duration, and 70% transition. If greater accuracies are required for an application, then a more accurate sensor or another biometric can be used to determine identity.

8.7 Tracking in Real Time

To understand how a doorway tracking system would operate in real time, we measure the transition accuracy of every event with different levels of future knowledge. In other words, since the tracking algorithm uses the *future to disambiguate the past*, we calculate how much future is required to correctly disambiguate an event. This future can take two forms: number of future crossing events and a period of time into the future. Figure 16 shows the transition accuracy of these two futures for the tracking system design with perimeter sensing. Both future types increase in accuracy with more data. However, future time depends on the behavior of the individuals in the home. Homes where people move more slowly between rooms, either due to room size or walking speed, will require longer time periods to have the same increase in accuracy. Future events should exhibit the same accuracy increases independent of human behavior.

In the absence of any future, tracking can correctly associate nearly 79% of the transition events. These results indicate that doorway tracking could support applications where an immediate response to a person's movements is needed. This includes applications such as lighting or HVAC control but may be limited to applications where weak identity association is acceptable. With just five future events or 15s of future, over 82%–84% of the transition events can be correctly associated with a particular individual. Hence, applications that have a small delay before they would query the system for an identity (e.g., asking who left the lights on in a room or who was last in the kitchen a few hours later) can be supported. In general, the longer the delay between an event and the application query, the more accurate the system response will be. This uncertainty could be communicated to the application and updated as more information becomes available.

9 RELATED WORK

The system that inspired this work is the Doorjamb system, which uses height as a weak biometric to track people at the room level (Hnat et al. 2012). Since its introduction, the Doorjamb hardware has been replicated by at least 5 other research groups and used for numerous studies, including energy apportionment and gait monitoring in the elderly (Lee et al. 2014; Nasir et al. 2015; Kalyanaraman et al. 2013). While this article looks specifically at non-device carrying, room-level, doorway transition-based tracking, a large number of other indoor localization systems have been created. These systems can be roughly classified into three categories: carried devices, visual systems, and non-invasive systems.

Most systems that track people in buildings require them to carry physical devices such as RFID tags, cell phones or battery powered transceivers. For example, all participants in the IPSN indoor localization competition fell into this category (Lymberopoulos and et al 2015). The Active badges system (Want and Hopper 1992), RADAR (Bahl and Padmanabhan 2000), and Cricket (Priyantha et al. 2000) are well-known systems based on infrared, wireless, and ultrasound, respectively. Newer systems use BLE transmitters for longer battery lifetime (Conte et al. 2014) or wearable RFID tags for battery-less operation (Ranjan et al. 2013). Some studies explore the use of unmodified carried devices, such as smartphones (Musa and Eriksson 2012), to perform indoor localization. In a commercial environment, these tracking solutions can be highly practical, but within homes tracking devices are often not feasible (Hnat et al. 2011). Residents may need or want to remove the devices for activities such as showering, changing clothes, or sleeping. They might be forgotten, require battery changes, or, if the device is a cell phone, left in a charging station while the residents perform other activities. In these cases, wearable device systems are forced to monitor the devices' location only, which may be very different from that of the person they are trying to monitor. Hours or days of tracking information can be lost in this situation, preventing intelligent applications such as space conditioning or medical monitoring from operating successfully.

Visual tracking systems collect information-rich video data so people do not need to wear or carry a tracking device. Some systems use ceiling-mounted sensors to track the movements of people within rooms (Teixeira and Savvides 2007) or between rooms (Erickson et al. 2011) but do not detect a person's identity. Vision alone often must deal with environmental problems, such as lack of coverage or occlusion, prompting many systems to use supplemental sensors such as audio, carried phones, or special occlusion algorithms (Zhang et al. 2014; Papaioannou et al. 2016; Zhang et al. 2014). Other systems use facial, iris, or gait analysis to recognize a person (Gafurov and Snekkenes 2009; Liu and Sarkar 2007; Venugopalan et al. 2011), but these systems require cameras to be placed at a lower level and closer to the person, where line-of-sight can become a problem. Pan-tilt-zoom cameras have been use to track specific individuals in a space, but are limited in the number of people they can track simultaneously (Cai and Medioni 2016). Unlike office environments, vision systems in homes also have problems with darkness. Night vision or depth cameras do not require light, but cost more and sacrifice resolution, possibly making it difficult to recognize individuals. Additionally, while video recording can be common and tolerated in public spaces, it is often considered a privacy invasion in the home, especially with the possibility that the data could be hacked.

Like visual systems, non-invasive tracking systems do not require wearable devices. Many non-invasive systems such as motion sensors, RF signals, and WiFi have been used to track people without inferring identity (Schiff and Goldberg 2006; Patwari and Wilson 2010; DSouza et al. 2013; Adib et al. 2015, 2014). These systems cannot differentiate between individual people and often cannot differentiate whether there is a single person or several people in a room. WiFi-based systems that do attempt to perform identity detection require extensive training and a very controlled space, such as narrow monitored hallway (Zeng et al. 2016; Wang et al. 2016). Non-invasive smart floor systems identify individuals by weight (Addlesee et al. 1997; Shen and Shin 2009) but require extensive and expensive deployment. These systems have not yet been shown to be cost effective for a typical home environment. Additionally, the weight detected by a floor tile can have high variability due to the person's motion, the location of the footstep, couches or chairs that span multiple tiles, and other objects on the tile or person. To our knowledge, none of the tracking systems described above have been evaluated in an *in situ* setting due to the difficulty collecting the ground truth location of humans.

10 CONCLUSIONS

Our findings indicate that doorway tracking systems can provide room-level identity-based tracking information to smart home applications in a real environment. We found that a doorway tracking system can achieve peak accuracies 96%, 91%, and 87% for room, duration, and transition accuracy, respectively, where each metric represents the needs of different applications classes. We discuss guidelines on how to achieve these accuracies and analyze what is required to make doorway tracking a true practical solution. We found that the behavior of people in a real environment and realistic assumption about occupancy in a home play the largest roles in attempting to design a doorway tracking system for the real world. However, by accounting for these behaviors in a doorway tracking system's design, real-world identity-based questions about our own lives and behaviors, such as "Has the toddler gotten into the kitchen?", "Have my sleeping hours been declining over the last month?", and "Who left the lights on?" can begin to be answered.

REFERENCES

M. D. Addlesee, A. Jones, F. Livesey, and F. Samaria. 1997. The ORL active floor [sensor system]. *Personal Commun. IEEE* 4, 5 (1997), 35–41.

Fadel Adib, Zachary Kabelac, and Dina Katabi. 2015. Multi-person localization via RF body reflections. In *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI'15)*. 279–292.

Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C. Miller. 2014. 3D tracking via body radio reflections. In *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14)*. 317–329.

P. Bahl and V. N. Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings (INFOCOM'00)*, Vol. 2. IEEE, 775–784.

Yinghao Cai and Gérard Medioni. 2016. Persistent people tracking and face capture using a PTZ camera. *Mach. Vis. Appl.* 27, 3 (2016), 397–413.

Giorgio Conte, Massimo De Marchi, Alessandro A. Nacci, Vincenzo Rana, and Donatella Sciuto. 2014. BlueSentinel: A first approach using iBeacon for an energy efficient occupancy detection system. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings.* ACM, 11–19.

Matthew D. Souza, Tim Wark, Mohanraj Karunanithi, and Montserrat Ros. 2013. Evaluation of realtime people tracking for indoor environments using ubiquitous motion sensors and limited wireless network infrastructure. *Pervas. Mobile Comput.* 9, 4 (2013), 498–515.

Varick L. Erickson, Miguel Á Carreira-Perpiñán, and Alberto E. Cerpa. 2011. OBSERVE: Occupancy-based system for efficient reduction of HVAC energy. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN'11).* IEEE, 258–269.

D. Gafurov and E. Snekkenes. 2009. Gait recognition using wearable motion recording sensors. *EURASIP J. Adv. Sign. Process.* 2009 (2009), 7.

T. W. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse. 2012. Doorjamb: Unobtrusive room-level tracking of people in homes using doorway sensors. In *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys'12).* ACM, 309–322. DOI: http://dx.doi.org/10.1145/2426656.2426687

T. W. Hnat, V. Srinivasan, J. Lu, T. I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse. 2011. The hitchhikers guide to successful residential sensing deployments. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys'11)*.

A. Kalyanaraman, E. Griffiths, and K. Whitehouse. 2016. TransTrack: Tracking multiple targets by sensing their zone transitions. In *Proceedings of the 2016 International Conference on Distributed Computing in Sensor Systems (DCOSS'16)*. 59–66. DOI: http://dx.doi.org/10.1109/DCOSS.2016.27

Avinash Kalyanaraman, Blaine Reeder, and Kamin Whitehouse. November 15–19, 2013. Poster abstract: Benefits and challenges of a height sensing approach for in-home gait speed detection. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA'13)*.

Avinash Kalyanaraman and Kamin Whitehouse. 2015. An event-based data fusion algorithm for smart cities. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 1575–1582.

Nacer Khalil, Driss Benhaddou, Omprakash Gnawali, and Jaspal Subhlok. 2016. Nonintrusive occupant identification by sensing body shape and movement. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*.

Seungwoo Lee, Daye Ahn, Sukjun Lee, Rhan Ha, and Hojung Cha. 2014. Personalized energy auditor: Estimating personal electricity usage. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'14)*. IEEE, 44–49.

Z. Liu and S. Sarkar. 2007. Outdoor recognition at a distance by fusing gait and face. *Image Vis Comput* 25, 6 (2007), 817–832.

Dimitrios Lymberopoulos et al. 2015. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Proceedings of the 14th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN'15)*. ACM Association for Computing Machinery. Retrieved from http://research.microsoft.com/apps/pubs/default.aspx?id=241638.

A. B. M. Musa and J. Eriksson. 2012. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems*. ACM, 281–294.

Nabeel Nasir, Kartik Palani, Amandeep Chugh, Vivek Chil Prakash, Uddhav Arote, Anand P. Krishnan, and Krithi Ramamritham. 2015. Fusing sensors for occupancy sensing in smart buildings. In *Distributed Computing and Internet Technology*. Springer, 73–92.

Savvas Papaioannou, Andrew Markham, and Niki Trigoni. 2016. Tracking people in highly dynamic industrial environments. *IEEE Trans. Mobile Comput.* (2016).

Parallax28015. 2016. Ping))) Ultrasonic Distance Sensor. Retrieved from http://www.parallax.com/product/28015 (2016).

Parallax28027. 2016. PIR Sensor (Rev B). Retrived from https://www.parallax.com/product/555-28027 (2016).

Shwetak N. Patel, Matthew S. Reynolds, and Gregory D. Abowd. 2008. Detecting human movement by differential air pressure sensing in HVAC system ductwork: An exploration in infrastructure mediated sensing. In *Pervasive Computing*. Springer, 1–18.

Neal Patwari and Joey Wilson. 2010. Rf sensor networks for device-free localization: Measurements, models, and algorithms. *Proc. IEEE* 98, 11 (2010), 1961–1973.

N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. 2000. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. ACM, 32–43.

Juhi Ranjan, Yu Yao, and Kamin Whitehouse. 2013. An RF doormat for tracking people's room locations. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 797–800.

RaspberryPi. 2016. Raspberry Pi—Model B DEV—11546. Retrieved from https://www.sparkfun.com/products/retired/11546 (2016).

RaspberryPiNoIR. 2016. Raspberry Pi NoIR Camera Board—1567. Retrieved from https://www.adafruit.com/product/1567 (2016).

Donald B. Reid. 1979. An algorithm for tracking multiple targets. *IEEE Trans. Auto. Control* 24, 6 (1979), 843–854.

SNAP RF200. 2014. SNAP Engine RF200. Retrieved from http://www.synapse-wireless.com/upl/downloads/resources/supporting-documents/data-sheet-rf200p81-and-rf200pu1-7f3a2a2e.pdf (2014).

J. Schiff and K. Goldberg. 2006. Automated intruder tracking using particle filtering and a network of binary motion sensors. In *Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE'06)*. IEEE, 580–587.

SHARP1487. 2016. Infrared Distance Sensor. Retrieved from http://www.sharpsma.com/webfm_send/1487 (2016).

Y. L. Shen and C. S. Shin. 2009. Distributed sensing floor for an intelligent environment. *IEEE Sensors J.* 9, 12 (2009), 1673–1678.

Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. 2010. Using height sensors for biometric identification in multi-resident homes. In *Pervasive Computing*. Springer, 337–354.

T. Teixeira and A. Savvides. 2007. Lightweight people counting and localizing in indoor spaces using camera sensor nodes. In *Proceedings of the 1st ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC'07)*. IEEE, 36–43.

Shreyas Venugopalan, Unni Prasad, Khalid Harun, Kyle Neblett, Douglas Toomey, Joseph Heyman, and Marios Savvides. 2011. Long range iris acquisition system for stationary and mobile subjects. In *Proceedings of the 2011 International Joint Conference on Biometrics (IJCB'11)*. IEEE, 1–8.

Wei Wang, Alex X. Liu, and Muhammad Shahzad. 2016. Gait recognition using wifi signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 363–373.

R. Want and A. Hopper. 1992. Active badges and personal interactive computing objects. *IEEE Transactions on Consumer Electronics* 38, 1 (1992), 10–20.

Yunze Zeng, Parth H. Pathak, and Prasant Mohapatra. 2016. WiWho: WiFi-based person identification in smart spaces. In *Proceedings of the 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 1–12.

Tianzhu Zhang, Kui Jia, Changsheng Xu, Yi Ma, and Narendra Ahuja. 2014. Partial occlusion handling for visual tracking via robust part matching. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1258–1265.

Received April 2016; revised January 2017; accepted May 2017