SonicDoor: A Person Identification System Based on Modeling of Shape, Behavior and Walking Patterns

NACER KHALIL, University of Houston
OMPRAKASH GNAWALI, University of Houston
DRISS BENHADDOU, University of Houston
JASPAL SUBHLOK, University of Houston

Non-intrusive occupant identification enables numerous applications in Smart Buildings such as personalization of climate and lighting. Current techniques do not scale beyond 20 people whereas commercial buildings have 100 or more people. This paper proposes a new method to identify occupants by sensing their body shape, movement and walking patterns as they walk through a SonicDoor, a door instrumented with three ultrasonic sensors. The proposed method infers contextual information such as paths and historical walks through different doors of the building. Each SonicDoor is instrumented with ultrasonic ping sensors, one on top sensing height and two on the sides of the door sensing width of the person walking through the door. SonicDoor detects a walking event and analyzes it to infer whether the Walker is using a phone, holding a handbag, or wearing a backpack. It extracts a set of features from the walking event and corrects them using a set of transformation functions to mitigate the bias. We deployed five SonicDoors in a real building for two months and collected data consisting of over 9000 walking events spanning over 170 people. The proposed method identifies 100 occupants with an accuracy of 90.2%, which makes it suitable for commercial buildings.

CCS Concepts: • **Computer systems organization → Special purpose systems**; • **Hardware → Sensor applications and deployments**;

Additional Key Words and Phrases: Wireless sensor networks, media access control, multi-channel, radio interference, time synchronization

ACM Reference Format:
Nacer Khalil, Omprakash gnawali, Driss Benhaddou, and Jaspal Subhlok. 2018. SonicDoor: A Person Identification System Based on Modeling of Shape, Behavior and Walking Patterns. 1, 1 (November 2018), 21 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

The ability to identify a person in an indoor environment without requiring the person to actively interact with the system or carry wearable or mobile devices will enable various building-related applications to support the users. In fact, providing identity information of building occupants would enable many applications including accurate and adaptive controls that would yield energy savings and increase comfort.

Achieving an accurate occupant identification is challenging because non-intrusiveness often leads to lower accuracy. Recent developments have exploited weak biometrics to identify occupants non-intrusively. These solutions exploiting

Authors' addresses: Nacer Khalil, University of Houston, nkhalil@uh.edu; Omprakash gnawali, University of Houston, gnawali@cs.uh.edu; Driss Benhaddou, University of Houston, dbenhad@central.uh.edu; Jaspal Subhlok, University of Houston, jaspal@cs.uh.edu.

parameters such as height [Ranjan et al. 2013], weight, footstep vibration [Subbu and Thomas 2014] or gait inferred from Wi-Fi RF signals [Jiang et al. 2012] do not accurately identify populations larger than 5-6 people making them unsuitable for commercial buildings where hundreds of people interact with the system every day.

In this paper, we propose a method that is a significant improvement over our previous work [Hnat et al. 2012] which accurately identifies up to 20 people by sensing height and width to infer the occupant's shape and movement using a door with Ultrasonic sensors. To achieve building-scale (100 or more people) identification, we design SonicDoor, a door instrumented with ultrasonic sensors, and data analysis techniques. The system uses a set of optimizations to the sensing, behavior detection, and corrections, and personalized Markov Chain Model for walking patterns in a network of sonicDoors. Enhancements in sensing platform resulted in increased sampling rate which leads to more accurate and consistent feature measurements with lower variances. Given this increase in accuracy, We propose a way to detect and correct a set of activities (that we call behaviors) that users perform as they walk through doorways. These behaviors, if not corrected, will bias the data and make identification more difficult to achieve. The behavior detection helps reduce the variance which leads to higher accuracy since clusters are less likely to overlap. We deploy a network of five SonicDoors where we learn about personalized users' patterns and filter candidates based on similarity to a set of clusters and also based on the frequent paths taken by the candidates. We model the participant's path patterns using personalized Markov Chain models for each participant which helped increase identification accuracy.

Compared to [Khalil et al. 2017], we added the following:

- Propose a new behavior namely detecting heels.
- Improve the identification model by taking into consideration the behavioral patterns that users perform and deriving the probabilities associated with performing the behaviors.
- Improve the decision function to take into consideration the behavioral patterns in deciding which user it is.
- Re-evaluate the model taking into consideration the new improvements and new behavior.

This paper makes the following contributions:

(1) We perform a real-world large-scale deployment of five door frames for two months collecting 9000 walking events spanning over 170 people. To our best knowledge, this is the largest deployment involving non-intrusive person identification in buildings.
(2) We made a set of improvements to the Ultrasonic Sensing which enabled us to increase the sampling rate by a factor of four from 35Hz to 132Hz.
(3) We propose a method to infer a set of common behaviors namely, using a phone, holding a handbag, wearing a backpack, and wearing high heels as the person walks through the door. By inferring such behaviors, we make the identification model more resilient to the observed variation by correcting the data from such behavioral bias.
(4) We propose a method that filters candidates based on the closest cluster candidates, filter using the network topology and build a per-person (personalized) Markov Chain model for every occupant to further filter based on the user's path probability. We retreive the set of behaviors observed as the person walks through the door and retreive the probability that the person performs such behavior.
(5) We also propose a "Decision score" that combines cluster closeness, the path probability and behavior probability. We show that this method is key to scale the system to accurately identify 100 people.

Table 1. Comparison of different non-intrusive people identification methods.

Paper	Sensor	Accuracy (%)	population
Hnat et al. [Hnat et al. 2012]	Ultrasonic	94	5
Pan et al. [Pan et al. 2015]	Geophone	96	5
Zeng et al. [Zeng et al. 2016]	Wi-Fi	93	4
Jenkins et al. [Jenkins and Ellis 2007]	Pressure	80	15
Khalil et al. [Khalil et al. 2016]	Ultrasonic	95	20

2 RELATED WORK

The ability to identify and track people has captured interest from the research community in the last decades. Different solutions have been proposed using different sensing technologies. Some sensing technologies require a user to carry a device [Hnat et al. 2012; Ranjan et al. 2013; Subbu and Thomas 2014] while others rely on sensing strong biometrics such as facial recognition [Lanitis et al. 1995], fingerprint [Hrechak and McHugh 1990], iris and hand geometry [Pan et al. 2013; Tisse et al. 2002]. Other technologies make use of weak biometrics such as height [Hnat et al. 2012; Srinivasan et al. 2010] and weight [Jenkins and Ellis 2007].

2.1 Carried and Wearable Devices

Different proposed technologies have been utilized including RFID-based wearables [Ranjan et al. 2013], users' smartphones [Subbu and Thomas 2014] and iBeacon technology [Hnat et al. 2012]. In RFID-based wearables, Ranjan et al. propose an RF Doormat which is an RF sensing system that can identify and track users' locations as they walk through doorways. Smartphones have been used to identify, track, and localize users in buildings using WiFi and Bluetooth access points. These systems identify occupants with high accuracy but suffer from missing a user if she does not carry the device. It can also misidentify the user if the wearable is carried by a different person.

2.2 Strong Biometrics

Different systems have been proposed using facial, fingerprint, iris and hand geometry and achieved high accuracy [Hrechak and McHugh 1990; Lanitis et al. 1995; Pan et al. 2013; Tisse et al. 2002] but they raise privacy concerns and some require user's active interaction with the system. Vision-based person identification is promising [Zhao et al. 2003] but they target different sets of applications ranging from entertainment and virtual reality to security and surveillance. Some vision-based identification methods rely on gait to identify occupants [Kale et al. 2004]. These vision-based systems are energy intensive, invasive and cannot be easily deployed in environments such as buildings as they are privacy infringing and places like nursing homes do not allow cameras to be deployed. Other systems such as those using fingerprint, iris, hand and retina sensing [Hrechak and McHugh 1990; Pan et al. 2013; Tisse et al. 2002] require a degree of engagement from the user to authenticate. This requirement is difficult to enforce and if users do not authenticate, then the identification process fails.

2.3 Weak Biometrics

Methods that sense height, weight, footstep vibration and step force have been proposed to identify occupants [Elrod and Shrader 2005; Hnat et al. 2012; Jenkins and Ellis 2007; Pan et al. 2015; Srinivasan et al. 2010]. These sensing systems are non-intrusive and do not require effort from the occupant in the identification process. Among these systems, footstep vibration based systems have been proposed to detect occupants identity [DeLoney 2008; Geiger et al. 2014;

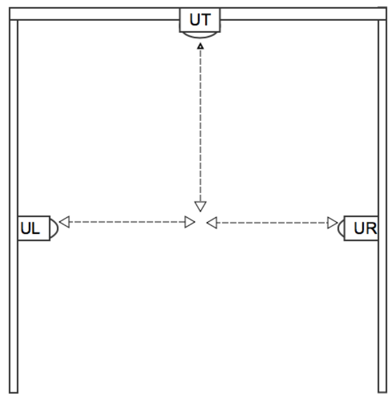


Fig. 1. Design of a SonicDoor doorframe.

Orr and Abowd 2000; Pan et al. 2015], some achieving 96% accuracy for a population of 4 to 5 people [Pan et al. 2015]. Height as a weak biometric has been proposed in the literature using ultrasonic sensing [Hnat et al. 2012; Srinivasan et al. 2010]. Doorjamb uses height information, walking direction, and tracking information to identify users achieved a high accuracy rate within a population of 2-4 people [Hnat et al. 2012]. Our previous system system leverages both height and width identify up to 20 people [Khalil et al. 2016]. Weight can also be used to identify occupants [Jenkins and Ellis 2007; Liau et al. 2008], sometimes achieving 80% accuracy for up to 15 people. RF signals have also been used to identify occupants (e.g., [Zhang et al. 2016]). WiWho [Zeng et al. 2016] identifies occupants by observing the signature of the RF signal reflection intensity on the human body. However, many of these systems fail to identify more than 4 with 90% and this quickly drops to 70% when the population is 6 people. Table 1 compares some of the work in this area. These systems are non-intrusive, do not require user's engagement to authenticate but many of them fail to identify large populations which make them limited and difficult to use in large commercial buildings. Our current system is able to identify up to 100 people with an accuracy of over 90%.

3 SYSTEM DESIGN

SonicDoor consists of a door frame instrumented with three ultrasonic sensors: One on top, UT, and two on the side, UR (for the right side), and UL (for the left side), as illustrated in figure 1. Each ultrasonic sensor measures the distance to the person passing by the door and generates a time series of data representing this distance. UT sensor generates data about the height, while UR and UL sensors generate data about the width. The following subsections discuss each component in detail and how the generated data is utilized to identify a person who passed by the SonicDoor.

3.1 System Overview

The identification system running SonicDoor is composed of the following components:

- Sensing and Calibration
- Event Recognition
- Event Data Correction
- Feature Extraction
- Filtering and Decision Making

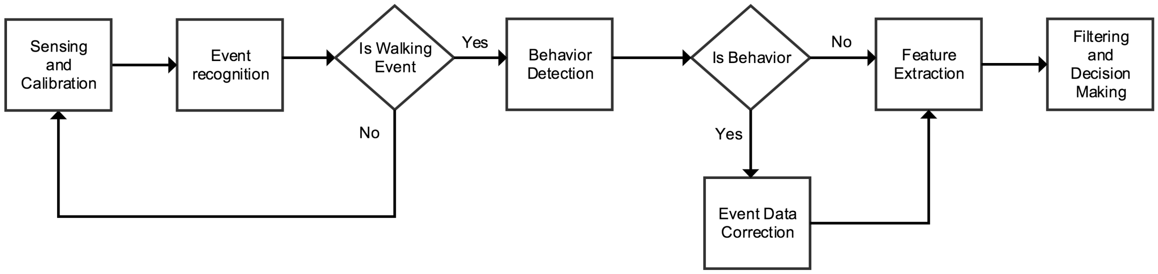


Fig. 2. Architecture of SonicDoor system showing components and processes from sensing to identification.

These components are organized as shown in Figure 2. Our system is composed of a network of SonicDoors deployed throughout a building. As the people walk through a SonicDoor, we first sense ultrasonic pings. We then decide if a person is walking through the door. We capture the walking event with sensor-data error correction and interpolation as described [Khalil et al. 2016]. We then analyze this data to detect any behavioral bias that may distort the data. If the walking event presents symptoms of such behavior coming from the user, we correct the data to make it less sensitive to these behavioral changes. Afterward, we extract a feature set from the walking event to map the walking event to a feature space. Each walking event is converted to a single point in the feature space. Each occupant is associated with one or multiple clusters. However, each cluster refers to one person. To determine the identity of the person, we first retrieve the five closest clusters to the point mapped from the walking event. We note the identity of these clusters and the distance of the clusters from our point. Afterward, we filter based on the network topology: We only keep the candidates seen in a previous node whose edge leads to the current node. If such an edge does not exist, the respective candidate is dropped from the list. Then we use the Markov Chain Model for every candidate to determine the probability that this person has taken such an edge. We combine both the edge probability and cluster distance into a decision score to rank the candidates and the identity of the person is determined by the highest decision score.

3.2 TestBed

Our testbed is composed of five SonicDoors deployed as shown in Figure 1 in an academic and research building in the University of Houston campus. Each SonicDoor is a wooden door frame mounted with three ultrasonic sensors as described in the previous section and illustrated in figure 1. These sensors are connected to an Arduino which runs the sensing and walking event detection algorithms. This data is then passed to a Raspberry Pi to which a camera is connected for ground truth collection. Each Raspberry PI is connected to the local area network. Once a walking event is detected, it is cleaned and errors are corrected. The data is then sent to our server through a message queue (RabbitMQ) that pushes the data to the database as well as to a local process that does the event behavior detection and correction, feature extraction and storing the raw event as a set of features.

3.3 Sensing

Each SonicDoor is equipped with three ultrasonic sensors.

3.3.1 Improving sensing sampling time. In prior work [Khalil et al. 2016], the system was sampling at 35Hz. Increasing the sampling rate is critical to provide sufficient data to enable using more accurate features which are used for

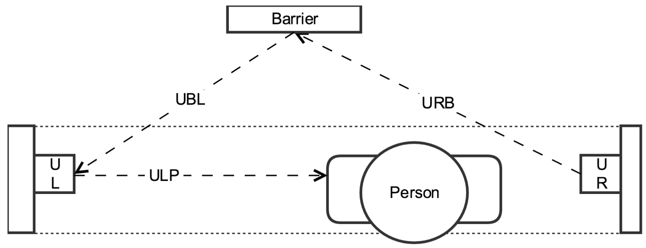


Fig. 3. A possible scenario where crosstalk between UR and UL happens in the case of concurrent sampling leading to erroneous width measurement. Crosstalk happens when distances $(URB + UBL) \leq 2ULP$.

identification and also for behavior detection. First, we must minimize sampling time allocated to each sensor to increase the sensor sampling rate. Each sensor is allocated enough time for its beam to get reflected back to it when it hits the farthest barrier: the ground in the case of UT and the sides of the door in the case or UL and UR. However, we can limit the time allocation for every sensor to allow every beam to travel just enough time to reach Walker regardless of his position under the door. The sampling rate depends on the door testbed dimensions. The door has height and width measure of respectively $d_h = 212cm$ and $d_w = 130cm$.

Each sensor is allocated enough time for the beam to travel forward and get reflected back. Given the speed of sound of $v = 341m/s$, we can compute the time required for the sound beam to traverse forward and backward. For example, the time needed to sense height using UT is $t_{ut} = t_{height} = \frac{2.d_h}{v}$. To operate all the sensors one at a time, the maximum sampling rate is:

$$f_{sample} = \frac{1}{t_{sample}} = \frac{v}{2.d_h + 4.d_w}(Hz)$$

With three sensors, this sampling rate is too slow to extract sufficient data for more sophisticated features. We use two techniques to speed up the sampling rate:

Truncating the sampling interval. When a person walks through the door, the beam reflects from the person and arrives back at the sensor much sooner than without the person. Without the person, the beam travels all the way to the bottom (in case of UT) or the other side (in case of UR or UL). We exploit this phenomenon by halving the sampling interval without compromising sensing correctness. Thus, we can effectively operate each sensor at double the sampling rate.

$$f_{sample} = \frac{v}{d_h + 2.d_w}(Hz)$$

Concurrent sensing. We can further speed up sensing if we operate the sensors concurrently. However, operating UR and UL concurrently causes sensor crosstalk. Even sampling height (UT) and width (UL+UR) in parallel cause crosstalk. Through experiments, we found that operating UT and one of UL or UR avoids crosstalk if there is a person walking through the door because that person functions as a shield between the beams from the sensors. This same person cannot be considered a shield between UR and UL because of a case where an external barrier reflects the signal coming from one sensor to another faster than the direct line as depicted in Figure 3. It is possible to operate these two sets of sensors (UT, UL) and (UR) at two different sampling rates because of the difference in distance the beam needs to travel for each case. To simplify the design and data processing, it is desirable to obtain the same number of samples from all the sensors so we simply use the smaller sampling rate allowed for these two sets.

3.4 Behavior Detection

We discuss how and why we detect three main activities: using a phone while walking, holding a handbag and wearing a backpack. Though there are many more activities that a walker performs as she walks in a building, we believe that these are some of the most common ones.

3.4.1 Rationale. A person's movement and posture changes as a result of behaviors performed when walking. Examples of behaviors are using a phone while walking, holding a handbag or holding a backpack. The person while performing such activities would be measured differently by SonicDoor than when not performing those activities leading to misidentification of the person. To make our system more robust to such behavioral biases, we detect such behaviors and correct their effect on the data. The idea of correction is to find a transformation function such that $f(feature) \rightarrow correctedfeature$

3.4.2 Detecting Person Using a Phone. Many people nowadays using their phones constantly, especially as they walk. As the walkers use their phone, their posture, as well as walking pattern, changes. For instance, a person using her phone will be more inclined forward and her head will be more leaned downward facing her phone compared to when the person is walking without distractions.

Given, we can sample at 132Hz, we can measure the head lean. In Figure 4, we illustrate the height measurements of two different walks by the same person: one where the person walks using her phone and another where the walker is not using the phone. One can observe that the curve related to using the phone appears to have her curve decrease at a more significant rate than the curve of when the user is not using her phone. To detect the phone, we compute the following feature:

(1) Locate the highest point
(2) Select all the point within 5cm of the highest point
(3) Fit a line using least square algorithm
(4) Compute gradient of the line with respect to x-axis

We determine if a person is using a phone based on this feature.

3.4.3 Detecting Handbag. Many walkers carry a handbag on one hand. This behavior distorts the data. The main sensors affected are the width sensors as the person would appear wider as a result. We detect this scenario by observing that the person on one side appears wider than on the other. As the person walks through the door, the ultrasonic sensor measures the distance to the hand and at other times, the distance to the waist. This is due to the fact that we naturally slide our hands as we walk. This results in two groups of distance measurements per side. The distance (we call body-hand) distance between these groups indicates how wide one's hands are as she walks. This calculation is done for both sides. We expect to see consistent body-hand distances from two sides. However, in the case of a handbag, the body-hand distance differs from one side to the other and indicates the person is possibly carrying a handbag. To identify the handbag, we propose the following feature for the walking:

- For each sensor UR, and UL, group the data distance measure into two groups based on the closeness between points. A point belongs to a group A if it's no farther than 2 cm.
- Calculate distance between two groups. This is referred to Body-hand (BH)distance. We calculate this distance for sensor UR and UL.
- Compute the difference between BH_{UL} and BH_{UR}

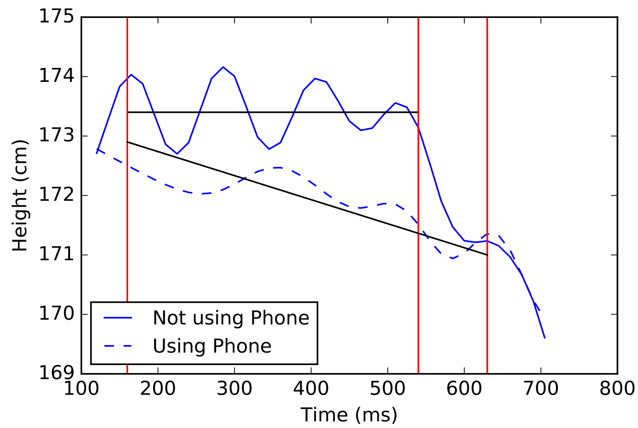


Fig. 4. Height readings (top of the head of the user) as a function of the sample number when using a phone and not using the phone. The vertical red lines show the area within 5cm of the highest points for each curve. The straight black lines are least square interpolation lines. We can observe a steeper descent for the person using a phone.

3.4.4 Detecting Backpack. Many users carry a backpack in everyday life. As the person walks through the door, it would be detected by the height sensor as the height would decrease at the less steep rate than in the absence of a backpack. In Figure 5, we illustrate two walks of the same person wearing a backpack and not wearing it. We observe that the decrease in height is not as steep when wearing a backpack. In addition, the number of points following the zone of highest points (when the head in under the door) is much larger.

To detect the backpack, we propose the following: First we identify the area with the highest points (head zone) which are all the points within 5cm of the highest point. We consider the list of height readings following the head zone. We note two features:

- Number of elements after the head zone as a ratio of the total number of elements
- Average rate of change between consecutive points after the head zone

Based on these two features, we train a Decision Tree Model to detect if a person is wearing a backpack.

3.4.5 Detecting Heels. Many people wear heels every day. Wearing heels affects the way a person walks. In fact, the person tends to bounce more and the footstep length decreases. This change in gait affects the extracted feature and introduces a bias that would lead to misidentification. To solve this issue, we propose an algorithm to detect whether a person is wearing heels or not.

To detect if someone is wearing heels, we use the footstep length and bounce as an indicator. The main observation is that people who wear heels will walk in smaller steps and bounce more than if they are not wearing heels. We observed experimentally this is a better indicator than relying on the bounce. To detect heels, we propose the following: We search for the highest and lowest point and measure the number of samples between the two. This is an indicator of the step length. Given we know the sampling rate,

- Let *sample* be the amount of time between the two consecutive samples. This is usually predefined. In our case the sampling rate is 132 Hz and sample is 7.5 ms
- Select point with largest height: P_{max}

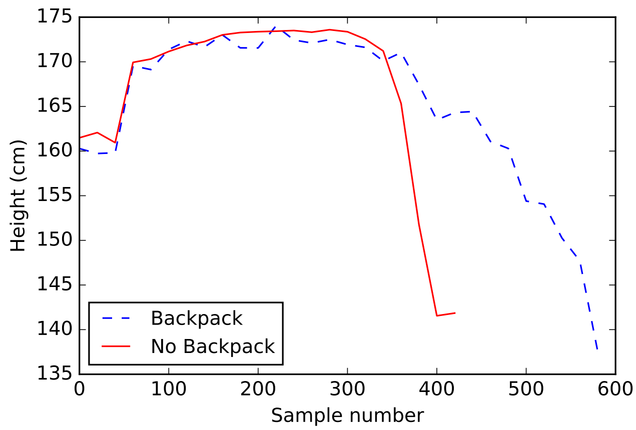


Fig. 5. Height readings as a function of the time of a person wearing a backpack vs. not wearing a backpack.

- Select point with lowest height: P_{min}
- Define *Bounce* $= \frac{P_{max} - P_{min}}{P_{max}}$
- Define Footstep length= $\frac{abs(Index(P_{max}) - Index(P_{max})) * sample}{P_{max}}$

3.5 Event Data Correction

Once we detect these behaviors, we need correct the walking event by removing such bias. For every behavior, we measure its impact on the walking event by measuring the rate of change of every feature when a behavior is present vs when it is not present. We then correct the walking event by transforming every feature based on the measured rate of change.

3.6 Feature Extraction

We proposed the following features:

- Minimum, Maximum, Average Height
- Minimum, Maximum, Average Width
- Time under the Door: The temporal length of the walking event
- Bounce: This is defined as Maximum Height - Minimum Height
- Girth: This is the waist perimeter of a person [Khalil et al. 2016]

The most successful features were girth and time because they were the least affected by the way people walked through the door. Therefore, finding a way to make our measured features resilient to the walkers' behavior is key to scaling our system. Our intuition is that by detecting certain behaviors and correcting the data, we can reduce the variance further and especially in other features which would strengthen the identification model.

3.7 Filtering and Decision Making

Now that our walking event is corrected from errors as well as corrected from behavioral biases, we decide which person walked through the door based on a number of criteria. We deployed a set of five doors throughout the building. Each time a person walks through one of the doors, we learn more about the person and use that information later to

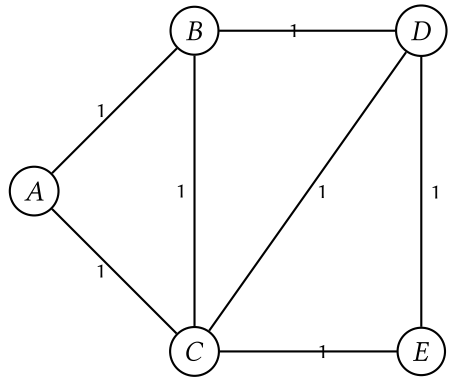


Fig. 6. Network topology of the deployed SonicDoors.

narrow the number of candidate identifiers. Our model does not require training data but requires us to approximate the number of people. Once we know the number of people, we cluster a subset of the data that is then used to create the initial clusters.

3.7.1 Clustering. We use HDBSCAN for clustering [Campello et al. 2013]. In [Khalil et al. 2016], we have shown how DBSCAN is appropriate for this problem but we found that it does not scale past 20 people. One of the main weaknesses of DBSCAN is that it does not cope well with varying density of clusters and expects a more conform density across different clusters. To solve this problem, HDBSCAN is based on DBSCAN but is able to cope with varying densities among the clusters. In fact, we do expect varying densities of clusters because, in commercial buildings, some people use the building more often than others.

HDBSCAN takes two parameters: the minimum number of samples per cluster and the maximum distance sparsity in a cluster which refers to how far can objects be to belong to the same cluster. As the number of participants grows, the feature space becomes denser and clusters become closer to each other. This pushes HDBSCAN to merge clusters and that causes misidentification as two different people would be mapped to the same cluster. Therefore, finding the distance sparsity parameter, "eps", is to be defined experimentally and should depend on the number of users as well as the sparsity in height and width of the population.

3.7.2 Candidate filtering using Clustering. Given a formed set of clusters as discussed in Section 3.7.1, and a new walking event, we first correct the walking event, detect behavior, extract features, and correct features from the behavioural biases. This new data point needs to be identified with the correct user or cluster. Instead of relying on our clustering algorithm to predict the correct cluster, we instead use it to filter a set of possible clusters.

We first compute the pairwise similarity between our new point and all points of our clustered dataset. For each pair of points, we use the Euclidean Distance between the two points. We rank all the points relative to the new point by distance. We then select the closest points that are no farther than 5cm. We chose 5cm experimentally because most of the shape-related features do not vary by more than 5 cm by person as we will see later in Figure 11. If no cluster is closer by 5 cm, then this is an indicator of a new person not seen by the system before. The identification stops at this point. But for the general case, we chose 5 closest clusters and note the respective distances to our point we want to identify.

3.7.3 *Candidate filtering based on SonicDoor Network Topology.*

After choosing the five closest candidates clusters, we eliminate further these candidates but this time based on the topology. The idea behind eliminating based on the topology is the following: As the person walks through the building, we gain more information about the Walker. If we see a person at doorway D_j at time t and we have a set of candidates C_i=$c_1, c_2, ...$. Then, each of these candidates must have been in a previous doorway at an earlier time whose edge leads to the doorway at time t. So eliminating by topology means keeping only the candidates who were previously seen at a door frame d_i whose edge leads to our door. So edge $D_i \rightarrow D_j$ exists in the topology. We note that this reasoning cannot apply to the start node as no previous door exists.

The process of eliminating candidates is as follows:

- For each candidate c_i in list of candidates at door d_t
- Select previous door location of candidate c_i d_{t-1} within the last 5 minutes
- if there is no edge from d_{t-1} to d_t, eliminate c_i from the list of candidates

3.7.4 *Decision Making using Markov Chain Model.*

§ We discuss how we learn the frequent paths of individual users by building a Markov Chain Model for every individual. Some paths are taken more frequently by some walkers rather than by others. As a result, our system needs to learn such path patterns and take them into consideration when deciding which user it is.

At this stage, we have the following information: A set of possible candidates, each with the respective distance from the walking event we are trying to identify. We also have an edge associated with every candidate.

Global Markov Chain Model. : One of the strategies is to build a global Markov Chain Model based on our door deployment topology as shown in Figure 6. Every time a person walks from one door to another, the respective edge frequency is incremented by one. This could help select the candidate based on the most frequently used edge. However, the most frequently used path among the whole population does not tell us much about the path of the individual candidates. In other words, the path taken depends on the person taking it. The main weakness of this approach is that it ignores the fact that different people take different paths and the most frequent path differs from one person to another.

Personalized Markov Chain Model. : Another approach is to build a Markov Chain Model for every person that will keep track of the different paths taken and enable the system to evaluate the probability that person P_i took edge E_i and select the candidate based on the highest probability of a person taking a specific path.

To build the personalized Markov Chain model, let's consider the case of a new person seen for the first time by the system (in this case a point farther than the closest clusters by a threshold distance and considered a new cluster). We first initialize a Markov Chain based on the topology in Figure 6 and set every edge weight to 1 as illustrated in Figure 6. We give equal likelihood to every edge since we have no information about the new person. Once a new event is assigned to such person, then we increment the edge in this person's graph by one.

To compute the probability that a person has taken such edge, we calculate the probability P_{ij} of an edge (i $\rightarrow$ j) with respect to the starting node i using $P_{ij} = \frac{Freq_{ij}}{\sum_{k=1}^{n} freq_{ik}}$. We divide the frequency of edge (i $\rightarrow$ j) by the sum of the frequencies of all edges starting at node i. Since we know person X was seen before in Node I, then this captures the probability person X appears in node j.

3.7.5 Behavior-based Identification. Given we are able to detect a set of behaviors as the person walks through the door, we noticed that some people are more likely to perform a behavior than others. For instance, women are more likely to wear high heels than men. Even though our system cannot and is not able to detect the gender of the candidates, the system is able to store a history of frequencies of various behaviors and derive probability associated with performing such a behavior. This probability helps filter and select the most likely candidate and therefore identify the person.

To perform this, we do the following. For each user, we associate a table of behaviors. For each behavior, we hold a counter recording the number of times the person performs the behavior and the number of times when she does not perform the behavior. Based on these two figures, we can infer the probability that a person may perform a behavior. We define the probability that person X performs behavior B as:

$P_{X,B} = \frac{Freq(X,B)}{Freq(X,B)+Freq(X,\ B)}$

This probability is used later in conjunction with other factors to identify the person. In addition, we initialize the probability table with all one as opposed to zero because if the person walked for the first time through the door, then the system would associate the probability $P_{X,B} = 50\%$ rather than 0 since the system has no idea whether one is more likely than the other.

3.7.6 Combining Cluster Distance, Path Probability and Behavior Probability. After filtering by cluster and retrieving the probability of every person taking his edge, we end up with three key pieces of information: the Euclidean distance between our point and a candidate cluster, the probability that candidate X has taken such an edge and the probabilities that person X did or not perform Behavior B. We have a probability associated with every behavior whether its performed or not. These pieces of information are independent of each other and giving more weight to one may influence the decision making further toward that one criterion as we repeat this operation which would lead to increased error as we run our system. In addition, there are numerous cases where a person takes an unusual path but walks in a very usual way and our clustering algorithm could point it with high likelihood or a person very distinct behavior or set of behaviors while rushing through the doors. We should, therefore, consider both pieces of information. We propose a decision score which combines the three pieces of information into a single score.

$$decisionscore = \frac{PathProbability}{Euclideandistance} * \prod_{b \epsilon B} BehaviorProbability_b$$

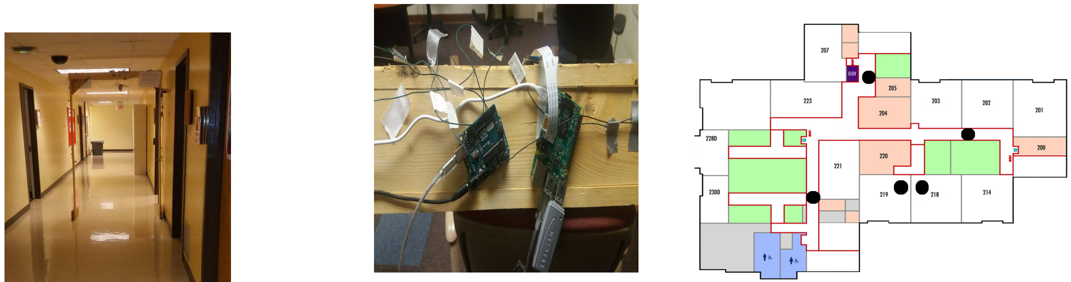
B refers is the set of Behaviors either performed or not whether depending on the user has performed them or not. Compared to the formula we presented in [Khalil et al. 2017], this one takes into consideration the probability that a person has performed a behavior or not. As we have mentioned the probability of a behavior can never be 0% as the behavior frequency table is initiated with one. Therefore, the fact that a person never performed a behavior will not lead to a decision score of zero.

4 SYSTEM EVALUATION

We describe the experimental setup and the results.

4.1 Deployment

We deploy five SonicDoors on the second floor (Figure 7c) of the Technology-2 building at the University of Houston for two months. The black dots on the map in Figure 7c represent the sensors' location. People were invited to participate in the experiment by simply passing through the doors and encouraged to walk as naturally as possible. No instructions



(a) Hallway showing a deployed SonicDoor.

(b) Top view of the door including an Arduino, a Raspberry PI and the wired ultrasonic sensors.

(c) The location of the SonicDoors (marked with black circles) in the Technology-2 Building at the University of Houston.

Fig. 7. Figure showing the SonicDoor testbed.

were given to the users. There was no direct communication with the participants with the exception of banners inviting them to participate. The experimental protocol was approved by the University of Houston Committee for the Protection of Human Subjects. Most of the participants were students, Faculty, and staff. Initially, 13,000 walks through the door were collected during the deployment. We discovered that almost 4,000 events were false positives (when a person does not walk through the door but the system thinks the person did). Fortunately, it was quite easy to spot the false positives where we can observe that they have no more than two samples per event. Also, after annotating the data, we found that over 215 participated in walking. We discarded from the dataset the people who participated once or twice as our model cannot predict anything based on one occurrence of a person. In Figure 8, we illustrate the distribution of Height and Width of all participants. We do not have accurate data about Gender but we observed that there was a homogeneous gender mix among the participants.

4.2 Data Annotation and Ground Truth

Given that 9000 walking events need to be evaluated for ground truth, manual annotation would be a tedious, error-prone and lengthy process. The process of annotating the data involves first associating an identifier with each person whenever she walks through the door, then evaluating all the events and grouping them along the 170 unique participants.

We opted for automated annotation using advances of computer vision in face recognition. We used Facenet [Schroff et al. 2015] which provides an algorithm that consumes an image composed of a face and provides the end user with a vector of 256 dimensions using Deep Learning. This vector is a mapping of the Face in the Euclidean space and similar faces have a Euclidean distance of less than 1. The algorithm achieved an accuracy of over 99.63% in the Labeled Faces in the Wild [Huang et al. 2007] dataset that is composed of 13,000 different faces. An implementation of Facenet is provided using Openface [Amos et al. 2016].

As a person walks towards the door, we extract 30 to 100 images containing the face, depending on the door position and how far was she initially. We collected a total of over 700,000 face images. Once, these images are collected, we clustered them such that each cluster refers to one person making sure that similar images of the same person are no farther than 1 Euclidean distance from another in the 256-dim feature space.

We convert every image to a 256-dim vector and note the ID of the walking event. We then search for the number of clusters in the dataset using Gaussian Mixture Model (GMM) estimation using Expectation-Maximization (EM). The parameter we vary is the distance between different images which is initially set to 1. For each walking event, we look for the percentage of the largest cluster of the set of captured images (30 to 100) and compute their average. We then vary distance value parameter passed to the GMM model until we find local maxima.

For a sanity check, we manually checked 3-4 images in every cluster to make sure the images refer to the same people. We checked 30 clusters. We did not observe any errors in the associations. The automatic association process is robust and can be trusted. We believe this annotation process could scale to a larger deployment at the multi-building level.

4.3 Annotation for Walking Event Evaluation

As explained in 4.2, we first receive the signal from the door that someone walked, then we look at the camera footage and annotate the data. Though this works, it doesn't enable us to evaluate the case where a person walks through the door undetected by our system.

To be able to evaluate walk-through-the-door events, we rely on the camera footage to indicate whether a person walked through the door or not. Using Openface [Amos et al. 2016], we extract all the face that appears in image taken by the camera. We extract their position and track their location in the camera. If they go through the center of image downwards (the camera is placed in the middle of the doorframe), then we conclude that someone walked through the door at time t. We then verify whether or not our door detected a walk-through event.

This type of annotation is important because it enables us to accurately measure the walk-through false positives (when someone walks through the door but the system is unable to detect this event). It is also important to measure because when building the Markov Model model we rely on accurate walk-through detection and if such assumption does not stand then we cannot retrieve or collect path probability information.

4.4 Evaluation Metrics

Our system uses a combination of clustering and filtering using Markov Chain Models. Even though we may refer to purity as a metric in clustering, we note that we have each walking event annotated with an identifier and our goal is to identify it. So we define accuracy as using the classical statistical definition:
$accuracy = \frac{TP+TN}{TP+FP+TN+FN}$.

4.5 Selecting Model Parameters

As the participant population grows, the clusters get closer to each other and the closer they get the more chance they will get merged into one making it difficult to distinguish between the clusters. However, by reducing the cluster's radius, we can fit more in the feature space. However, by reducing the radius, we run the risk of creating many more clusters and having many clusters per person which would in itself reduce accuracy. Ideally, we need to have one cluster per person and no two people in the same cluster. We experiment with the variable eps which refer to the maximum distance between a point and a cluster member in HBDSCAN. We found that a distance value eps=0.3 seemed to work best with our dataset leading to high cluster purity but also approximately 1.25 clusters per person. In a different setting, one must experiment with different values to find not only which one applies to the size of the population but also to the specific population targeted.

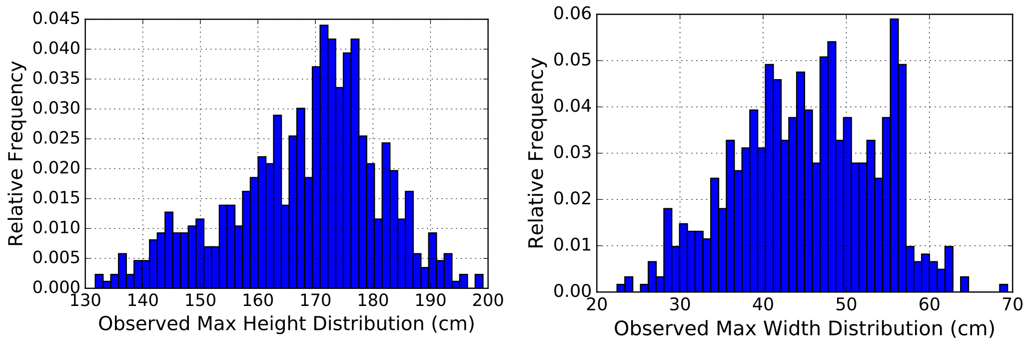


Fig. 8. Distribution of maximum height and width per walking pattern of the participants.

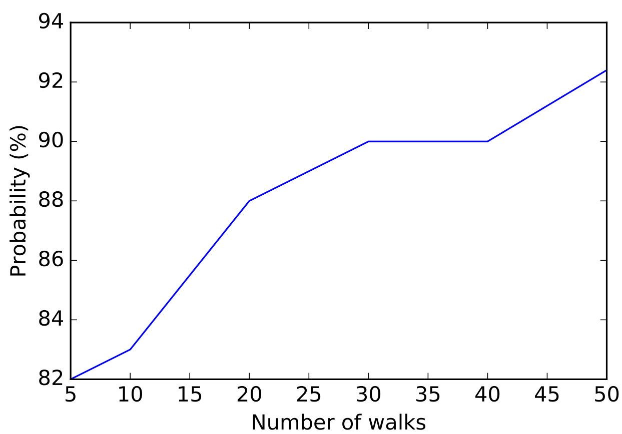


Fig. 9. The probability of identity being correctly chosen as we vary the numbers of walks of a person.

4.6 System Performance as a Function of Number of Walks by a User

As noted, the more a user walks through the door, the higher the probability of being accurately identified. To evaluate the hypothesis, we first cluster with the first 5 days of the dataset to create the initial model for the occupants. Then, we compute the number of times the person walked through any of the doors. As a person walks through the door network, we note if she was correctly identified or not and how many times was the person seen previously. We then map the number of walks through the door with the frequency of times being correctly identified. Figure 9 plots the probability that the system correctly identifying a person given the number of times it has seen her in the past. The more a person walks through the door, the higher the probability of being correctly identified which makes it useful for corporate buildings where the population does not change often.

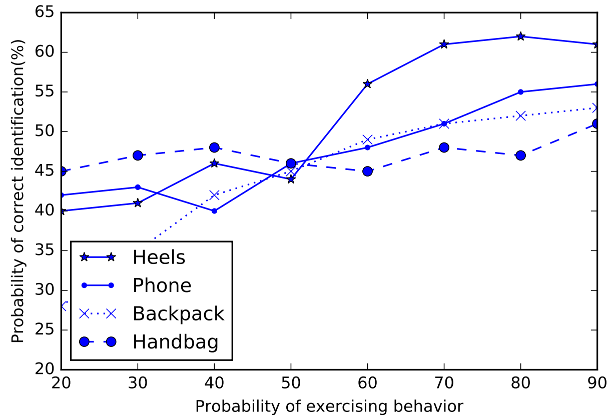


Fig. 10. The probability of identity being correctly chosen as we vary the numbers of walks of a person.

4.7 Evaluating Behavior-based Identification on System Performance

Our data shows that certain people perform certain behavior more consistently and at a higher frequency than others which makes this information potentially useful for occupant identification. To measure how much benefit the Behavior-based identification contributes to the overall model, we perform the following experiment. We cluster the first 5 days of the dataset to create the initial model for the occupants. Then, we note the behavior-related information for every occupant where we note the probability that this person performs a certain behavior and note whether she was correctly or wrongly identified. For instance, if a person walks through the door, we first look at the behaviors performed, the probability that this person performs such a behavior and whether he was correctly identified. We group these by behavior and percentiles. So we have a set of corrent and incorrect identification for every behavior and for every behavior probability. The goal is to measure what is the probability of correct identification as we vary the behavior probability and see if higher behavior probability leads to higher identification probability.

In Figure 10, we plot the probability of performing a behavior and its effect on correctly identifying the user. Each line refers to a behavior namely wearing heels, using phone, wearing a backpack and carrying a handbag. We observe that as the probability of all behaviors increases, accuracy of identification increases. However, it seems that most of them tend to converge around 50-60% correct identification probability. Therefore, though there is a benefit in using such a method, its overall contribution will tend to converge as we collect more accurate behavior probability information.

4.8 Evaluating Walking Event Detection

Relying on the door to trigger the walking event does not evaluate the system's ability in accurately detecting the walking event. In fact, there may be cases where a person walks through the door undetected and the system would fail to recognize the walk-through and therefore not identify this walker. We evaluate the performance of walk-through detection by using the annotation as described in 4.3. As we have annotated by relying on the camera to point to a walkthrough at time t, we check if the door detected a person at time t. Therefore, we use all the walking events.

We note that the number of walking events is 9000 when the number of walks detected by the camera is 9134. This shows that a few walking events are undetected by the system. Table 2 shows the accuracy with which the system can detect the events corresponding to walks through the door performance. The walk-through events are very accurately

Table 2. Walk-through detection performance for each of the five doors deployed

Door	Accuracy
1	99.1%
2	98.4%
3	99.3%
4	97.7%
5	98.4%

Table 3. Accuracy by clustering using different features.

Feature	Accuracy
Detecting Phone	89.1%
Detecting Handbag	90.1%
Detecting Backpack	84.6%
Detecting Heels	87.1%

detected. However, we note that many of the cases involved in missed detection have to do with groups of people walking through the door close to each other. The system in some of these cases thinks it's only one person with an unconventional shape. This may be solved by comparing the girth's shape to a more standard oval one and therefore find the delimiting line between the two.

4.9 Accuracy of Behavior Detection

We arbitrarily select 300 walking events from our dataset. This smaller dataset has walking events belonging to 40 people each with walking events varying between from 4 to 20. For each of these events, we manually annotate it by watching the video footage at the time of the event and look for the following:

- Is Walker using her phone?
- Is Walker holding a handbag?
- Is Walker holding a backpack?
- Is Walker wearing heels?

We divide the dataset into training and testing set. We train three different models (all three based on Decision Trees Models), one for each behavioral detector: detecting phones, handbags, backpacks and heels. The training set is composed of 1/3 of the data and the rest is used for testing. For each trained model, we evaluate its performance using the rest of the data. Table 3 shows the accuracy of each of the trained models.

4.10 Building Behavior Correction Models

In order to accurately identify occupants, we need to measure features that are consistent for the same people but differ across different people. Many of these features are affected by these behaviors but if detected, the walking event can be corrected. Given we can accurately identify the behaviors as shown in Table 3, we need to correct the walking event. To do so, we find a transformation function for every feature
$f : features_i \rightarrow correctedFeature_i$.

For every feature, we build a Linear Regression that corrects the feature. We use the dataset composed of 300 walking events each with annotated behaviors as described in Section 4.9. For every behavior, we group the walking events into

Table 4. The intercept α and slope β for every feature with respect to each behavior measured.

Detect:	**Phone**		**Backpack**		**Handbag**		**heels**	
	β	α	β	α	β	α	β	α
Mean Height	1.019	1.765	0.973	-1.28	1.005	-0.031	0.97	-0.451
Min Height	1.0345	0.72	1.452	0.742	0.992	0.752	0.989	-1.258
Max Height	1.025	1.523	1.009	0.873	0.991	1.023	0.93	1.45
Mean Width	1.001	0.023	0.987	1.173	0.965	0.458	1.001	0.45
Min Width	0.996	1.031	1.0145	1.02	0.969	2.346	1.003	0.248
Max Width	1.01	0.783	0.993	0.4	0.931	0.759	0.998	-1.091
Girth	1.016	-1.25	1.007	-0.563	0.874	-1.573	1.023	0.876
Bounce	1.029	-1.245	0.682	1.249	1.004	0.238	0.883	-1.245

two groups: those showing such behavior and those not showing them. We build a Linear Regression Model for every feature and every behavior. Table 4 shows the estimated regression parameters namely the intercept α, and the slope β. From the measured slopes, β, one can observe that some features significantly benefit from the model. For instance, the feature Minimum Height with respect to the backpack is increase by a factor of 1.4. This table also shows how some of these features are distorted by a person using a phone, holding a handbag or wearing a backpack.

4.11 Feature Variance with Behavior Detection and Correction

Feature Variance affects our models' accuracy. When the features' variance increases, clusters may start overlapping and eventually HDBSCAN will start merging them. As a result, the clusters' purity decreases and the identification accuracy decreases.

As shown in Sections 4.9 and 4.10, transforming the walking pattern with the aim of removing the bias due to the behaviors should help decrease feature variance and therefore increase the accuracy of the model. We evaluate the effect of modeling Behavior Detection and Correction on the feature variance. We evaluate how the feature variance changes as a result of behavior detection and correction. We selected 300 annotated walking events as used in Section 4.9. This dataset contains the walking events, the identifier of the person and whether or not the walker exhibited any of the behaviors namely: using a phone, holding a handbag, wearing a backpack and wearing heels. We group these walking events by person and for every person, we compute the variance with respect to every feature. We rerun this operation but this time by first detecting if there is an event and if so we correct it according to the correction model and then calculate the variance with respect to every feature. Figure 11 shows a comparison of the distribution of variances across people for every feature. We conclude that behavior detection and correction does reduce variance and will increase accuracy we see in section 4.12. We note that some points are outside the Boxplot and that is due wrongly identifying the behavior which could happen.

Compared to the initial results in [Khalil et al. 2017], we noticed that there was a slight improvement when including the heels detection algorithm. It appears that mean height, min height, and max height saw their variance decrease further when including such a behavior.

4.12 Impact of Network Filtering And Behavior Detection and Correction

We evaluate the benefit of using the topology filtering and Markov Chain Path Frequency Filtering, alone, then we augment the model with the behavior detection and correction, and finally, we augment the latter with Activity-based Recognition. We select 10 different groups of sizes 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. We have 11 sets of 10 groups each. The group members were selected randomly without replacement. Each group had at least 1000 walking

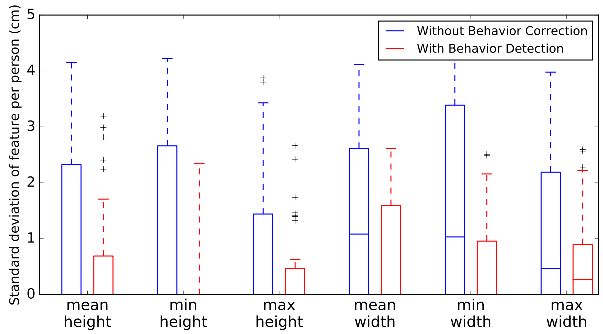


Fig. 11. Comparison of the observed variance of different walks of a person across different people. For each set of boxplot, the right boxplot shows the variance of features without behavior detection and correction and the left boxplots include the behavior detection and correction which show a clear decrease in variance across different walks for the same person.

events with the exceptions of groups of sizes 5 which only had 600 walking events. For each group, we sort the data chronologically. Then we take 1/4 of the sorted data and feed it to our clustering algorithm to create the initial clusters. We then build the individual Markov Chain models from this initial data by parsing it chronologically and using the clusters for identification. Then for each element in the remaining 3/4 of the data, we evaluate its accuracy with and without behavior detection. We extract the features for the walking event and extract the five closest clusters as explained in Section 3.7.2. For both cases, we retrieve the previous location of every candidate, eliminate a candidate based on if the edge exists in the topology, compute the probability of such edge from the candidate's Markov chain model and select the candidate with the based on the decision score as shown [Khalil et al. 2017]. To include activity-based identification, for every possible behavior, we note whether the user performed it or not. Then for every candidate, we select the probability that the user performed the activity or not. These probabilities are then used in the update decision score in 3.7.6. Figure 12 compares both methods. From Figure 12, we observe that using the topology and the network information greatly benefits the model. However, we note that for the larger populations where the benefit is more significant, the formed groups contain occupants with on average the highest number of walks compared to the smaller groups. This happens because, in the smaller groups, we have a larger set to choose from the groups have more participation frequencies. These methods outperform our previous method [Khalil et al. 2016] by a large margin.

5 DISCUSSIONS

Although the system achieved 90%-96% accuracy with 5 to 100 people, the accuracy degrades to 76% for 170 people. With more than 100 people, the cluster density increases and schedule overlap among people increases resulting in lower accuracy, however the performance is still better than the state of the art. With more doors, the accuracy likely would have increases allowing larger number of possible paths taken by people. Adding more ultrasonic sensors to each door could increase the accuracy but the benefits of additional sensors may be marginal because additional measurements of physical shape of the person brings more nuanced sampling of the user's body, not some fundamentally different aspect of physical shape and movement of the person from what we already capture (height, width, time). Although we explore the most common user behaviors in this paper, it may be possible to increase the accuracy by modeling more user behaviors.

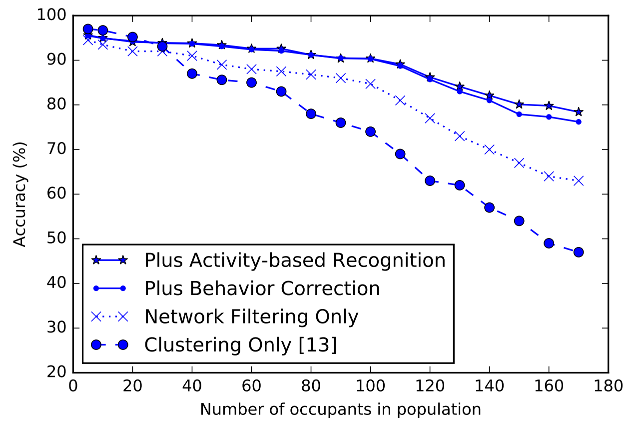


Fig. 12. Accuracy of using the SonicDoor network topology filtering model for identification to the accuracy with and without behavior detection and correction and comparison to previous work [Khalil et al. 2016].

Sonicdoor may complement other tracking systems such as cameras because they sense different things and therefore the data are independent of each other. Given both methods achieve high accuracy, there are cases when one system would fail when the other would succeed. For instance, if a person covers her face, the camera will fail to identify her but Sonicdoor would successfully identify her. Also, if more than one person walks through the door at the same time, sonicdoor would fail but the cameras would still detect the faces. There are cases where both systems would fail at once but the probability of this happening is lower than the probability of either on failing.

Adding more sensors may improve the identification but the margin of improvement would not necessarily justify the cost of adding one or more sensors per door. Since we are not adding a new degree of freedom, some of the data coming from the new sensors will correlate with data coming from already existing sensors and the information gain may not be significant enough. But it is useful to experiment with more sensors to come to a final conclusion. We found in a previous study [Khalil et al. 2016] that the height sensor only marginally improved the model which is why we did not use it in the model. But in this paper, we used it to detect behaviors and understand their impact on the walker.

6 CONCLUSIONS

In this paper, we propose SonicDoor, a method that uses Ultrasonic Sensor-Instrumented door to identify occupants. Our model uses a network of SonicDoors and builds a Markov Chain Model for every occupant and identifies the occupant by first filtering based on the topology. Then, we identify by combining cluster distance and individual path likelihood. With the new system design that allows fast sampling at 132 Hz, we infer three types of common user behaviors namely using the phone, holding a handbag or wearing a backpack. We deployed a network of five SonicDoors in a commercial building for two months and collected a total of over 9000 walking events by over 170 participants. To our knowledge, this is the largest deployment involving non-intrusive person identification. The SonicDoor system can identify people with an accuracy of 90.2% in a group of up to 100 people. This is five times greater than the state of the art which is limited to up to 20 people.

REFERENCES

Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. 2016. *OpenFace: A general-purpose face recognition library with mobile applications.* Technical Report. CMU-CS-16-118, CMU School of Computer Science.

Ricardo JGB Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 160–172.

Chasity DeLoney. 2008. Person identification and gender recognition from footstep sound using modulation analysis. (2008).

Scott Elrod and Eric Shrader. 2005. Smart floor tiles/carpet for tracking movement in retail, industrial and other environments. US Patent App. 11/236,681.

Jürgen T Geiger, Maximilian Kneißl, Björn W Schuller, and Gerhard Rigoll. 2014. Acoustic gait-based person identification using hidden Markov models. In *Proceedings of the 2014 Workshop on Mapping Personality Traits Challenge and Workshop*. ACM, 25–30.

Timothy W Hnat, Erin Griffiths, Ray Dawson, and Kamin Whitehouse. 2012. Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors. In *SenSys'12*.

Andrew K Hrechak and James A McHugh. 1990. Automated fingerprint recognition using structural matching. *Pattern Recognition* 23, 8 (1990), 893–904.

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. 2007. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.* Technical Report 07-49. University of Massachusetts, Amherst.

Jam Jenkins and Carla Ellis. 2007. Using ground reaction forces from gait analysis: Body mass as a weak biometric. In *International Conference on Pervasive Computing*. Springer, 251–267.

Yifei Jiang, Xin Pan, Kun Li, Qin Lv, Robert P Dick, Michael Hannigan, and Li Shang. 2012. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 441–450.

Amit Kale, Aravind Sundaresan, AN Rajagopalan, Naresh P Cuntoor, Amit K Roy-Chowdhury, Volker Kruger, and Rama Chellappa. 2004. Identification of humans using gait. *IEEE Transactions on image processing* 13, 9 (2004), 1163–1173.

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.* ACM, 1–10.

Nacer Khalil, Driss Benhaddou, Omprakash Gnawali, and Jaspal Subhlok. 2017. SonicDoor: Scaling Person Identification with Ultrasonic Sensors by Novel Modeling of Shape, Behavior and Walking Patterns. In *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys 2017).*

Andreas Lanitis, Christopher J Taylor, and Timothy F Cootes. 1995. Automatic face identification system using flexible appearance models. *Image and vision computing* 13, 5 (1995), 393–401.

Wen-Hau Liau, Chao-Lin Wu, and Li-Chen Fu. 2008. Inhabitants tracking system in a cluttered home environment via floor load sensors. *IEEE Transactions on Automation Science and Engineering* 5, 1 (2008), 10–20.

Robert J Orr and Gregory D Abowd. 2000. The smart floor: a mechanism for natural user identification and tracking. In *CHI'00 extended abstracts on Human factors in computing systems*. ACM, 275–276.

Shijia Pan, An Chen, and Pei Zhang. 2013. Securitas: user identification through rgb-nir camera pair on mobile devices. In *SPSM'13*.

Shijia Pan, Ningning Wang, Yuqiu Qian, Irem Velibeyoglu, Hae Young Noh, and Pei Zhang. 2015. Indoor person identification through footstep induced structural vibration. In *HotMobile'15*.

Juhi Ranjan, Yu Yao, and Kamin Whitehouse. 2013. An RF doormat for tracking people's room locations. In *Ubicomp'13*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 815–823.

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

Kalyan P Subbu and Nathan Thomas. 2014. A location aware personalized smart control system. In *IPSN'14*.

Christel-loic Tisse, Lionel Martin, Lionel Torres, Michel Robert, et al. 2002. Person identification technique using human iris recognition. In *Proc. Vision Interface*. 294–299.

Yunze Zeng, Parth H Pathak, and Prasant Mohapatra. 2016. WiWho: WiFi-based Person Identification in Smart Spaces. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).* IEEE, 1–12.

Jin Zhang, Bo Wei, Wen Hu, Salii S Kanhere, and Ariel Tan. 2016. Human identification using WiFi signal. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops).* IEEE, 1–2.

Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. 2003. Face recognition: A literature survey. *ACM computing surveys (CSUR)* 35, 4 (2003), 399–458.