Hallway based Automatic Indoor Floorplan Construction using Room Fingerprints

Yifei Jiang†, Yun Xiang§, Xin Pan†, Kun Li◇ Qin Lv†, Robert P. Dick§, Li Shang◇, Michael Hannigan*

† Dept. of CS, ◇Dept. of ECEE, *Dept. of ME, University of Colorado Boulder, CO 80309 U.S.A.
§ EECS Department, University of Michigan, Ann Arbor, MI 48109 U.S.A.
† ◇ *{yifei.jiang, xin.pan, kun.li, qin.lv, li.shang, hannigan}@colorado.edu,
§{xiangyun, dickrp}@eecs.umich.edu

ABSTRACT

People spend approximately 70% of their time indoors. Understanding the indoor environments is therefore important for a wide range of emerging mobile personal and social applications. Knowledge of indoor floorplans is often required by these applications. However, indoor floorplans are either unavailable or obtaining them requires slow, tedious, and error-prone manual labor.

This paper describes an automatic indoor floorplan construction system. Leveraging Wi-Fi fingerprints and user motion information, this system automatically constructs floorplan via three key steps: (1) *room adjacency graph construction* to determine which rooms are adjacent; (2) *hallway layout learning* to estimate room sizes and order rooms along each hallway, and (3) *force directed dilation* to adjust room sizes and optimize the overall floorplan accuracy. Deployment study in three buildings with 189 rooms demonstrates high floorplan accuracy. The system has been implemented as a mobile middleware, which allows emerging mobile applications to generate, leverage, and share indoor floorplans.

Author Keywords

Indoor floorplan; context sensing; indoor localization

ACM Classification Keywords

C.3.3 Special-Purpose and Application-Based Systems: Real-Time and Embedded Systems

INTRODUCTION

On average, people spend approximately 70% of their time indoors [15], such as in offices, schools, and stores. New indoor mobile applications are being developed at a phenomenal rate, covering a wide range of indoor personal and social scenarios. Many of these mobile applications build upon indoor localization techniques, using location information to optimize and deliver on-the-fly personal and social services.

Indoor maps, i.e., indoor floorplans, are often required by indoor localization techniques to answer the following question *"so... I'm here but where is here?"* [11]. Simply put, the floorplan is typically in the form of a building blueprint, defining the structure and functionality of a specific indoor environment. Given an indoor floorplan, the personal and social functions supported by the indoor environment are then defined.

Acquiring indoor floorplan information is challenging – (1) many buildings do not have floorplans in easily-interpretable digital form; (2) even if available, building managers are often reluctant to share such information with the general public; furthermore, (3) buildings' internal structures and the corresponding functionalities often evolve over time, making the original floorplans outdated. Although it is possible to manually construct an indoor floorplan or correct an existing one, such process is slow, tedious, error prone, and difficult to scale.

This work tackles the problem of automated indoor floorplan construction. To accurately construct an indoor floorplan, a set of indoor features are required, including (1) unique identification of individual rooms, (2) estimated room geometric information, e.g., length and width, and (3) geometric relationship among rooms. Accurate determination of these indoor floorplan features faces the following challenges:

- **Heterogeneous indoor environments:** An indoor environment consists of rooms with diverse sizes. Room connections through hallways also vary significantly. Such heterogeneous indoor environments make accurate floorplan construction challenging.
- **Noisy Wi-Fi fingerprints:** Due to the complex multipath propagation problem, Wi-Fi fingerprints obtained by mobile phones are dynamic and noisy. The problem is particularly challenging when using Wi-Fi fingerprints to determine detailed floorplan features.
- **Mobile crowd sourcing:** Leveraging the built-in motion sensors of the mobile phones carried by occu-

UbiComp'13, September 8–12, 2013, Zurich, Switzerland.

http://dx.doi.org/10.1145/2493432.2493470

pants, "crowd sourcing" offers a potentially highly scalable solution for automatic floorplan construction. The primary challenge is how to accurately extract stable and representative floorplan structure from diverse and random occupant motion patterns.

In this work, we propose an automatic indoor floorplan construction system. Leveraging Wi-Fi fingerprints and user motion information collected via mobile crowd sourcing, the proposed system extracts indoor floorplan features, including room identity, geometry, and inter-room geometrical relationship, and then construct the indoor floorplan automatically. To tackle the aforementioned challenges, the proposed system uses the following main components.

- A *room adjacency graph construction* algorithm that identifies the adjacency of rooms and constructs a room adjacency graph that is robust to the spatial bias of room fingerprints and Wi-Fi noise;
- A *hallway layout learning* algorithm that determines the room arrangement along each hallway, e.g., room sizes and orders, using crowd-based motion sensing on smartphones; and
- A *force directed dilation* algorithm that adjusts the individual room structures globally to improve floorplan accuracy.

The proposed system builds upon existing room localization techniques [4, 6, 11]. The proposed system has been implemented and tested through deployment in five different buildings with diverse floorplan structures, including classrooms, research labs, libraries, offices, and shopping malls with 189 rooms in total. The deployment study shows that the proposed automatic indoor floorplan construction system yields an average room position accuracy of 91%, room area estimation error of 33% and room geometric aspect ratio error of 24%, requiring on average only 20 data points per location for the system to converge.

The rest of the paper is organized as follows. We start with an overview of the system architecture, then present the detailed technical design of the proposed system. Evaluation results are presented next. After surveying related work, we conclude this paper.

SYSTEM OVERVIEW

This section formulates the problem of indoor floorplan construction, and provides an overview of the proposed system.

A floorplan graphically displays the dimensions and positions of rooms inside a building. A floorplan includes three types of information: the number of rooms, dimensions of rooms, and relative position of rooms. In this work, we focus on rectangular-shaped rooms typically seen in most buildings. Each room can be represented by the coordinates of its center point (e.g., x and y values on a 2-D plane), along with its width and length information. The indoor floorplan construction problem can be defined as follows: Given a room set R containing n distinct rooms on the same floor in a building, determine the relative center coordinates, width, and length for each room $r_i (1 \leq i \leq n, r_i \in R)$ in a two-dimensional space.

In our system, the input data includes room Wi-Fi fingerprint r_i for each room, passively collected motion sensor data (accelerometer and compass), and Wi-Fi scan data when users are walking in the building. We assume that at least three Wi-Fi access points are visible from most positions in each room and hallway. This assumption is valid for most public and commercial buildings. A number of approaches exist for collecting room Wi-Fi fingerprints, such as manual room fingerprint collection [4], leveraging user feedback [8, 11] , and automated learning techniques [6]. Our work leverages the outputs of the aforementioned approaches and we assume that each room has been associated with a unique room ID and the corresponding Wi-Fi fingerprint. Note that the room IDs are used for identification and are necessarily related to real room names or numbers.

When users are walking, the motion sensor data and Wi-Fi scan data are collected by our system running on users' smartphones. Note that all data are passively collected from users without explicit actions on users' part. Users do not even need to be aware of the collection process. The accelerometer data are collected at a 5 Hz frequency and used to determine if a user is walking. When a user is walking, the compass and Wi-Fi data are collected at 5 Hz and 1 Hz frequencies, respectively. Note that this system only needs to be run in the training phase. Once the floorplan is constructed, the system no longer needs to be active, and the produced floorplan will be provided to other indoor mobile applications and services, thus saving smartphone energy.

Figure 1 illustrates the overall process of our indoor floorplan construction solution. The system consists of three main components: (1) room adjacency graph construction, (2) hallway layout learning, and (3) force directed dilation.

Room Adjacency Graph Construction. This component identifies adjacencies among rooms and constructs a room adjacency graph. In a room adjacency graph, the vertices represent individual rooms and the edges between vertices indicate adjacency between two rooms. We used k-means clustering to group Wi-Fi signals of two adjacent rooms into the same cluster. This approach is robust to Wi-Fi noise and unevenly distributed Wi-Fi scans within a room.

Hallway Layout Learning. Based on the room adjacency graph and users' walking traces in hallways, the layout learning algorithm first identifies rooms and their orders along each hallway, and then assembles the hallways based on the similarity score of Wi-Fi signals.

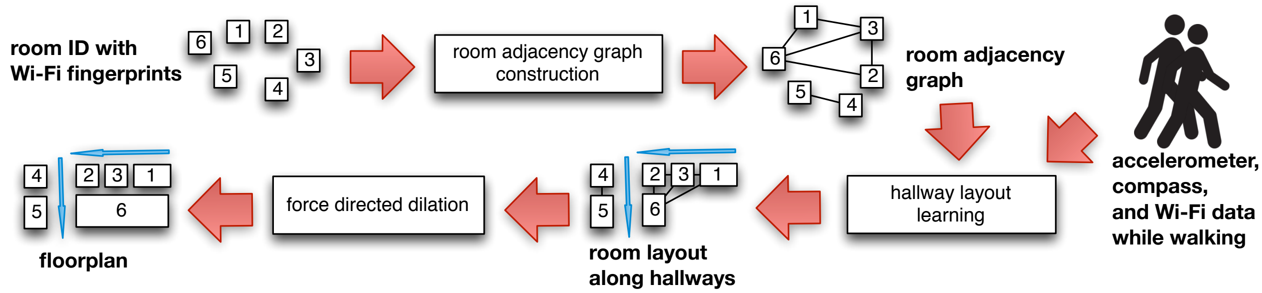

Figure 1. Automatic indoor floorplan construction system overview.

This component includes a novel and accurate indoor turn detection algorithm, which leverages compass sensor data and a noise-robust room layout detection algorithm.

Force Directed Dilation. This step optimizes the overall floorplan accuracy via global adjustment of the estimated room dimensions. We use a force directed dilation algorithm, in which the learned floorplan is defined as a mechanical system with springs between adjacent rooms, and the dimensions of rooms are automatically adjusted based on the forces among rooms.

ROOM ADJACENCY GRAPH CONSTRUCTION

Room adjacency graph construction is a process to determine whether each pair of rooms is adjacent for a given set of rooms, R. The adjacency information is represented using an undirected graph. For example, if two rooms are adjacent, the corresponding nodes in graph are connected by an edge as shown in Figure 1. The adjacency graph is a fundamental information for floorplan construction, and is used later in hallway layout learning and force directed dilation.

To determine the adjacency of two rooms, a straightforward approach is to compare the similarity of their fingerprints. However, such similarity can be affected by the relative positions of two rooms, the type of wall between rooms, and the distribution of fingerprints within rooms. Yet, Wi-Fi signal noise increases the uncertainty. Moreover, this approach requires the definition of a global threshold applicable to diverse room conditions, which is very difficult to derive.

In this work, we propose a clustering based approach to determine the relationship between pairs of rooms. It is based on the observation that Wi-Fi signals in adjacent rooms are similar, especially for those signals collected near the partition wall. Even though wall can decrease Wi-Fi signal strength, allowing differentiation of adjacent rooms, noise can blur the boundary between two rooms. On the other hand, signals collected near the wall can be differentiated by extreme RSS (received signal strength) values. The extreme RSS values for an AP (access point) are close to the minimum or maximum RSS value of that AP. Those signals are most likely at the edges (corners) of rooms. By leveraging such Wi-Fi signals, we can cluster adjacent rooms into the same group.

Based on the above observations, for each room fingerprint (containing a set of Wi-Fi scans collected in that room), we use the following procedure to remove Wi-Fi scans in which APs contain few extreme RSS values. First, for each access point ap_i that can be observed in a room r, we extract its RSS value from the room fingerprint. Second, we normalize the RSS values using minmax normalization to the range of $[0, 1]$. Given a Wi-Fi scan w, where $w = \{rss_{ap_1}, rss_{ap_2}, ..., rss_{ap_m}\}$, we calculate the average normalized RSS value nr_w as follows:

$$nr_w = \sum_{i=1}^{m}(rss_{ap_i})/m \qquad (ap_i \in w). \tag{1}$$

We remove all Wi-Fi scans whose nr_w values satisfy $0.2 \leq nr_w \leq 0.8$, and only keep Wi-Fi scans with more extreme RSS values.

After removing those Wi-Fi scans, we run the k-means clustering algorithm on the remaining Wi-Fi scans from all rooms. Based on our evaluation results, we set k to 4 times the total number of rooms. For each cluster, we calculate the total number of Wi-Fi scans n_{scan}, the number of unique rooms n_{room}, the number of Wi-Fi scans for each room $n_{scan/room}$, and the ratio n_{scan}/n_{room} as the threshold to determine if a room should be considered in this cluster. This is because if a room only has a few Wi-Fi signals in a cluster, e.g., $n_{scan/room} \leq n_{scan}/n_{room}$, those signals might be noise caused by environment changes or device heterogeneity. Thus if there are two rooms in one cluster that each satisfies $n_{scan/room} \geq n_{scan}/n_{room}$, these two rooms are classified as adjacent rooms.

HALLWAY LAYOUT LEARNING

A hallway is defined as a straight or curved path in a building that is adjacent to rooms. The end of a hallway can be a wall or an outlet. When a user walks through a hallway, he/she will pass all the rooms along the hallway sequentially. This indicates how rooms are arranged along the hallway.

Rooms vary by size and pairs of rooms can be positioned in many possible orientations. Based on these observations, our layout learning technique includes three sub-components: (1) *hallway motion detection*, which detects walking in straight hallways; (2) *layout learning along hallway*, which identifies the lengths of rooms and sequences along the hallway; and (3) *hallway assembling*, which is based on room overlap information on intersecting hallways.

Hallway Motion Detection

The goal of hallway motion detection is to identify user activities, such as walking straight and making a turn. Our detection algorithm first distinguishes traversing hallways from walking within rooms by leveraging Wi-Fi signals while walking and room fingerprints. It then identifies activities in different hallways via turn detection.

Hallway Traversal Detection

We leverage the technique proposed by Jiang et al.[6] for walking detection due to its high detection accuracy and energy efficiency. In order to determine whether a user is walking in hallway, our system periodically scans Wi-Fi signals at around 1 Hz. When a user is moving, the Wi-Fi RSS for access points change. However, as obstacles between phone and access points mainly affect Wi-Fi signal strengths, Wi-Fi signals within a room are relative stable (even for moving phones) compared to those during hallway traversal.

Based on the observation, we determine whether a moving user is in a hallway or room by comparing two consecutive Wi-Fi scans w_a and w_b. Due to Wi-Fi signal noise, the similarity score between two single Wi-Fi scans is quite unpredictable. In order to deal with this challenge, we first convert the single Wi-Fi scan w_i to a set of similarity scores d_{ij} for each room $r_j (r_j \in R)$. d_{ij} is defined as [5]

$$d_{ij} = \prod_k P(ngram_k(w_i)|r_j)P(r_j), \tag{2}$$

where

$$ngram_k(w_i) = (ap_k, ap_{k+1}). \tag{3}$$

That is, ap_k and ap_{k+1} are two APs in Wi-Fi scan w_i and are consecutive when ordered by RSS values. $P(r_j)$ is the probability of room r_j appearing in the system. We set all $P(r_j)$ equal to 1.0 since rooms are treated equally in our system.

The Wi-Fi scan w_i is then converted to a vector $D_{w_i} = d_{i1}, ..., d_{in}$, where n is the total number of rooms in R. The similarity computation between w_a and w_b is converted to the similarity computation between D_{w_a} and D_{w_b}. This conversion makes the similarity more reliable because d_{ij} is calculated using room fingerprint, which contains a set of Wi-Fi scans.

The similarity function between signal vector D_{w_a} and D_{w_b} is defined by Tanimoto Distance:

$$\begin{aligned} &simi(D_{w_a}, D_{w_b}) \\ &= \frac{D(w_a) \cdot D(w_b)}{|D(w_a)|^2 + |D(w_v)|^2 - D(w_a) \cdot D(w_b)}. \end{aligned} \tag{4}$$

We set a threshold τ. If $simi(D_{w_a}, D_{w_b}) \leq \tau$, this suggests motion within a room. $simi(D_{w_a}, D_{w_b}) > \tau$ suggests motion within a hallway. Based on our experimental data, we set $\tau = 0.3$.

Hallway Turn Detection

Compass-based turn detection [13] has been well studied in outdoor environments. However, in indoor environments, the accuracy of digital compasses on smartphones can be significantly affected by electrical cables and devices, construction materials, and other metallic objects nearby.

To deal with these challenges, we propose a technique combining accelerometer, compass, and Wi-Fi signals to detect turns in hallways. This is accomplished in three key steps: (1) coarse-grained turning point detection using accelerometer-based and compass-based turn detection; (2) clustering of turning points using Wi-Fi signals; and (3) fine-grained turning point detection when a user is close to a cluster of turning points. The intuition is that, we want to identify stable turning point clusters and utilize fined-grained (i.e., more sensitive) turning point detection only when a user is close to a turning point cluster. Next, we describe the three steps in detail.

Step 1: *Coarse-grained turning point detection*. Here, a turning point is detected whenever either accelerometer or compass detects a turn. Accelerometer-based turn detection is inspired by the walking detection approach [6], which assumes that when a user is walking, his body is in oscillation. However, when the user is turning, the oscillation is broken. If $m/m_{abs} \geqslant 0.15$ and $m_{abs} \geqslant 300$, we consider the user to be turning. Compass-based turn detection uses a 3 second long sliding window and calculates the mean value for each window. If the change of two consecutive values is larger than $\tau_{compass}$ (set to 30 in this work), the user is determined to be turning..

Step 2: *Clustering of turning points*. The individual turning points detected using either accelerometer or compass can be noisy. Given the physical constrains of hallways, users' turning points should be centered around certain regions such as the end of a hallway or hallway intersection. To identify these regions where turning is likely to occur, we leverage Wi-Fi scans collected at individual turning points and use density-based clustering to group Wi-Fi scans into clusters representing these turning regions. The similarity of two Wi-Fi scans is computed based on RSS value using Equation 4. The two parameters for density clustering algorithm, Eps and $MinPts$, are set to 0.8 and 15 respectively. Note that although the similarity of two single Wi-Fi scans is noisy, the density based clustering algorithm is robust to such noise and can produce stable Wi-Fi based turning regions.

Step 3: *Fine-grained turning point detection*. Given the stable turning regions, we can determine whether a user is close to any of these regions by calculating the similarity (Equation 2) between the current Wi-Fi scan and each of the Wi-Fi clusters produced in Step 2. If the user is close to a turning region, we lower the thresholds of accelerometer- and compass-based turning point detection. Specifically, m/m_{abs} is set to 0.11 and $\tau_{compass}$ is set to 20. The lower thresholds make turning point detection more sensitive near the turning regions. If both accelerometer- and compass-based approaches detect turning activity, the user is classified as turning.

Layout Learning Along Hallway

Given a specific hallways, there are two different types of room layouts along this hallway: (a) rooms on one side of the hallway only and (b) rooms on both sides of hallway. Not only do we need to identify each hallway and the rooms along the hallway, we also need to determine the rooms' orders and lengths along the hallway.

The hallway motion detection algorithm identifies the Wi-Fi scan sequences collected in a particular hallway. Those sequences may cover a whole hallway or segments of it. The hallway identification algorithm then clusters those Wi-Fi sequences into groups so that each group represents a unique hallway. All Wi-Fi sequences in one group are collected in the same hallway. A trace is not required to cover the whole hallway from one end to the other. A user can start and end a trace at any points on a hallway. The system can learn the complete hallway layout as long as the aggregated traces cover the whole hallway.

For each hallway h, we get a group of Wi-Fi scan sequences S_h and a room set s'_h by comparing room fingerprints with Wi-Fi scan sequences. To identify room order and length of rooms along the hallway, we first group rooms on the same side of hallway together based on the room adjacency graph – if rooms are on the same side of the hallway, their shortest path length to the group equals 1. Then for each side of the hallway, the following calculation is performed. For each Wi-Fi scan sequence $s_h \in S_h$, $s_h = \{w_1, \ldots, w_j\}$, we calculate the similarity between $w_i (1 \leq i \leq j)$ and each room r along the hallway based on Equation 2. Then we can get a time series shown as real dots (blue) in Figure 2, where x is the time point at which w_i was collected and y is the Wi-Fi similarity score. The time series tend to follow Gaussian distribution, and fitting the time series with a Gaussian model can help reduce noise in the Wi-Fi similarity calculation..

We repeat this process for all rooms on the same side of the hallway, producing results as shown in Figure 3. We compute the orders of rooms by compare the times-

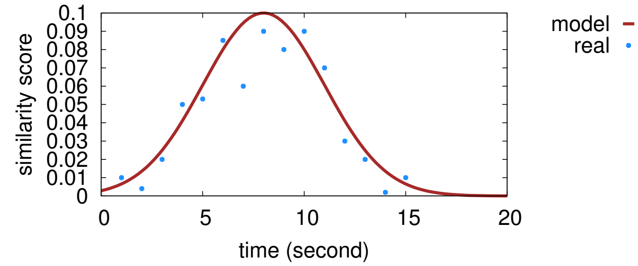

Figure 2. Hallway room similarity fitting.

tamps at the means of the Gaussian distributions. The time duration for crossing each room is also computed using intersection of adjacent distributions. For example, in Figure 3, the start time t_{s2} of crossing room 2 is the intersection of room1 (blue) and room2 (red) curves and the end time t_{e2} is the intersection of room2 (red) and room3(green) curves. The time duration of crossing room 2 is thus $t_{s2} - t_{e2}$. The room length along the hallway is computed by multiplying the duration with average walking speed, which is set to be 1.4 m/s. The final room length is the average of room lengths computed from all s_h in S_h. Using the average room length eliminates outliers caused by a few users who walk slower or faster than others.

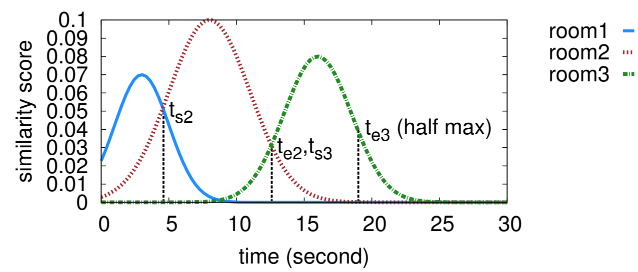

Figure 3. Hallway room order and size.

Hallway Assembling

We have learned a set of hallways H and the information of room order and lengths along each hallway $h (h \in H)$. Hallway assembling aims to identify connecting and/or intersecting hallways. The assembling algorithm starts with two hallways h_a and h_b ($h_a, h_b \in H$). If h_a and h_b share common rooms, they are assembled as one of the structures in Figure 4 (a), (b), and (c). The choice of structure is determined by the positions of the shared rooms in each hallway. For example, if the shared rooms are at one end of both hallways, the two hallways are assembled as the structure in Figure 4(a). We approximate the angle of two intersecting hallways as either 45° or 90°. These two intersection types allow us to determine whether two hallways are perpendicular, and in turn capture the geometric relations among rooms.

In order to estimate the angle between two hallways, our system calculates the changes in compass readings when a user turns from one hallway to the other. When the change is larger than a threshold τ, the angle of

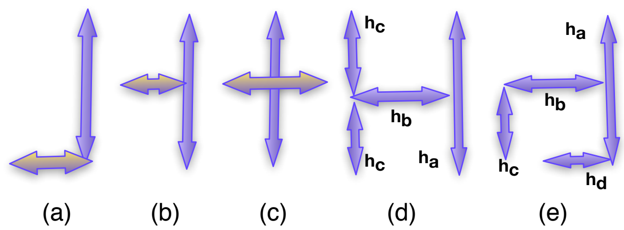

Figure 4. Hallway assembling: Connecting multiple hallways.

the two hallways is classified as 90°. Otherwise, it is classified as 45°. Due to noise resulting from magnetic material in buildings, a single estimation is unreliable. We use majority voting to determine the angle of two hallways by considering compass reading changes from different users and the same user at different times. We evaluated this approach in four different buildings with controlled experiments. The results show that the hallway angle classification accuracy is 93%.

Next, we pick another hallway h_c that intersects with h_a or h_b. The constraint of hallway intersection alone is insufficient for the placement of h_c. For example, in Figure 4(d), if h_c intersects with h_b at one end, h_c can be placed either above h_b or below h_b. In order to determine that, we select a room r_{h_c} from the side that does not intersect with h_b and another two rooms r_{above} and r_{below} which are above and below h_b, e.g., two rooms at each side of h_a. If the similarity between fingerprint r_{h_c} and r_{above} is greater than that of r_{h_c} and r_{below}, h_c is placed above h_b. Otherwise, it is placed below h_b. We define the similarity of two room fingerprints to be the average similarity of each Wi-Fi scan in one room to the fingerprint of the other room based on Equation 2.

Figure 4(e) shows one more scenario that needs to be addressed in hallway assembling. Here, h_d intersects with h_a and h_c, but the length of h_d is shorter than the distance already learned between h_c and h_d. In that case, we dilate hallway h_d to satisfy the conditions, and all rooms along hallway h_d are dilated by the same proportion.

In some large public buildings, e.g., airports and shopping malls, a few hallways are curved. Figure 12 shows an example from a shopping mall. In order to deal with these scenarios, our system calculates the hallway curve using the hallway length and direction angle changes between the two ends of a hallway. The direction angle changes are calculated using the compass sensor on smartphone. We use the average value over multiple users in order to reduce the impact of magnetic noise.

FORCE DIRECTED DILATION

Our hallway layout learning algorithm is able to estimate the room dimensions along each hallway (i.e., room length) but not the width (or depth) of these rooms. To estimate the perpendicular dimensions, e.g., room 105 and 150 shown in Figure 9(c), another technique is required.

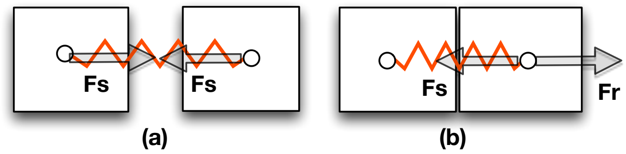

Figure 5. Spring system and the forces.

Algorithm 1 Force-Directed-Dilation(**input** G, D)

```
1: input G;            ▷ graph after hallway learning & assembling
2: inputl D;  ▷ directions set: left, right, up, and down
3: while G is not in equilibrium state do
4:     Compute F_vd on each v in d    ▷ v ∈ G and d ∈ D
5:     F_{v_j d_k} = Max(F_vd)
6:     Stretch v_j in direction d_k with Δ meters
7: end while
```

One might consider using user motion within rooms to estimate room dimensions [1]. However, this approach can be unreliable because it requires that users move to all edges and corners of the room. In reality, users are generally concentrated within part of a room. Thus, we propose to use a force directed dilation technique to estimate and automatically adjust the sizes of rooms.

Force directed algorithms [14, 3] have been applied to the problem area of graph drawing. The idea is to place graph nodes in 2D or 3D space with as few crossing edges as possible. The algorithm assigns an attractive and repulsive force to each edge in a graph. The values of the two forces are determined by the distances between the nodes. Once the forces have been defined, the algorithms gradually adjust the positions of nodes until each node experiences net zero force.

Our floorplan optimization algorithm responds to attractive F_s and repulsive F_r forces, between rooms. As shown in Figure 5, if two rooms are connected in the graph, we assign a spring system between their centers. The force F_s exerted by a spring is defined as [3]. Note that F_s also exists between hallways and rooms.

$$F_s(d) = c_1 * log(d/c_2), \tag{5}$$

where d is the length of the spring, and c_1 and c_2 are constants. As shown in Figure 5(a), each room receives the same amount of force F_s but in opposite directions. We define F_r as the force generated by walls between two adjacent rooms. Thus, if walls are not against each other, as shown in Figure 5(a), the force $F_r = 0$. Otherwise, F_r is equal to the opposite force on the wall, e.g., $F_r = -F_s$ in Figure 5(b).

Having defined the attractive force F_s and repulsive force F_r, we use the force directed dilation algorithm, as sketched in Algorithm 1, to determine the dimensions of the rooms. The inputs of the algorithm are the graph G and direction set D. G is learned in the first two components and nodes have been placed as learned/individual rooms. D represents the direction of force and consists of four directions: left, right, up, and down. A force F towards any other direction will be decomposed into these four directions.

The equilibrium state of graph is defined as follows: (1) the net forces F on all vertices are zero or (2) no vertex in the graph can be dilated. If the graph is not in equilibrium state, we compute the sum of forces for each vertex F_v and decompose it to each direction in D as F_{vd}. Then we find the largest force F_{vd} and dilate vertex v in direction d. Note that if there are multiple vertices with the same largest force F_{vd}, all those vertexes will be dilated. The algorithm repeats the above steps until the graph is in an equilibrium state.

The reason to dilate a room with the largest force F_{vd} is that when a vertex has a larger force, it means that the room has less similarity to the actual shape in the floorplan. For example, in Figure 9(c), room 150 has the largest force on the left direction, which is generated by the attractive forces from room 105, 116 and 137. The large force is caused by the deviation of room 105's currently assigned size from its actual size.

We assign higher priorities to rooms like 150 and 105 because (1) adjusting those rooms will make the learned floorplan look more like the real one and (2) avoiding block among rooms while dilation. For example, room 116 in Figure 9(c) also receives force in the down direction. If we change it first, it will block the dilation of room 150 in the following steps.

DEPLOYMENT STUDY

The system has been implemented as a mobile middleware on Android. A pilot indoor mobile application has also been developed built upon the floorplan construction middleware. Nine users have been recruited to conduct the deployment study at five different building sites: classrooms, research facilities, libraries, shopping malls, and offices with a total of 189 unique rooms. The user study lasted two months.

During the user study, the participants installed our mobile app on their personal Android phones and carried their phones normally without any change to their daily activities. The data collection function is automatically activated only when users are in the study buildings in order to minimize the impact on smartphone battery lifespan and protect participants' privacy. We implemented a state-of-the-art room fingerprinting algorithm [6] to generate the Wi-Fi based room fingerprints that are required by the system as inputs. The system automatically learned fingerprints for 91 unique rooms using the algorithm. We manually collected fingerprints for the remaining 98 rooms in order to generate complete floorplans. 11 phones of 6 types were used. We collected more than 25 motion traces from users for each hallway.

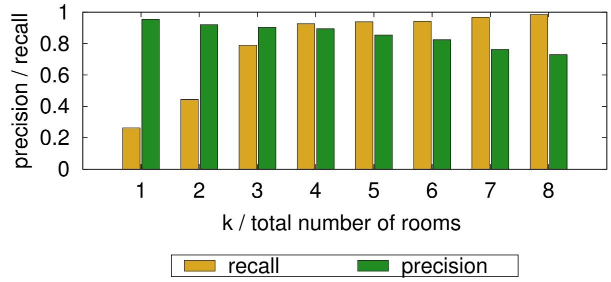

Figure 6. Quality of room adjacency graph with different k.

Room Adjacency Graph Construction

We first evaluate the proposed room adjacency graph construction algorithm, which provides fundamental information for use in floorplan construction. We adopt the precision-recall measure. More specifically, given the ground truth graph G and the produced graph G' of our algorithm, the *recall* is the number of links in G divided by the sum of the number of links in both G and $G's$. The *precision* is the number of links in G' divided by the sum of the number of links in both G and $G's$. The precision-recall measure is evaluated under different k-means clustering setting k. As shown in Figure 6, as k increases, recall increases and precision decreases. This is due to the fact that, with more clusters, the algorithm becomes more sensitive to the change of adjacency between rooms. However, as the corresponding size of each cluster decreases, the classification process becomes less robust. The deployment study shows that a k value equaling to 4 times of the total number of rooms provides a good precision-recall tradeoff.

Motion-Based Traversal/Turn Detection

We developed a motion detection technique for hallway traversal detection and turn detection. Appropriate thresholds need to be experimentally determined for high-accuracy motion detection.

Figure 7 evaluates the quality of hallway traversal detection using different thresholds. Using precision-recall measure, the *recall* is the total duration of correctly detected hallway traversals divided by the total duration of hallway traversals and the *precision* is the total duration of correctly detected hallway traversals divided by the total duration of detected hallway traversals. This study shows that the threshold settings for similarity between two consecutive Wi-Fi scans significantly affect the quality of motion detection. Small thresholds result in missing hallway traversals. Large thresholds lead to misclassification of in-room walking as hallway traversal. The deployment study shows that a threshold of 0.3 provides a good precision-recall tradeoff.

Figure 8 evaluates the quality of hallway turn detection using the precision-recall measure. The proposed method, which combines accelerometer, compass, and Wi-Fi fingerprint data, is compared against the fol-

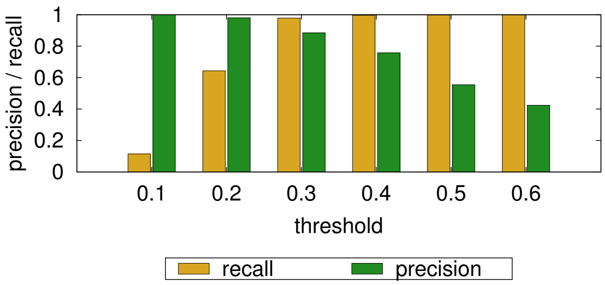

Figure 7. Traversal detection quality with different thresholds.

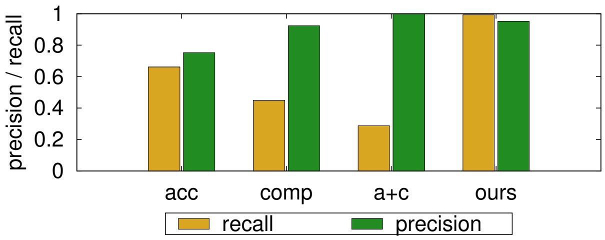

Figure 8. Turn detection quality comparison.

lowing three techniques, accelerometer-only, compass-only, and accelerometer-compass. It shows that, accelerometer-only based approach yields low recall and precision due to high dynamics and diversities of human motion. Compass-only based approach yields high precision but suffers from low recall when metallic objects are nearby. The combined accelerometer-compass approach further improves precision with the cost of low recall measure. On the other hand, leveraging the proposed Wi-Fi fingerprint-based turning point detection, our proposed solution offers both high precision and recall.

Floorplan Construction

Floorplans are the final output of the proposed system. We show four case studies of automatically constructed floorplans, a classroom building, a research lab building, a shopping mall, and a office builidng. The library building floorplan is not shown due to limited space and its similar structure to research lab building, but its results are included in the following quantitative analysis results.

Figure 9 provides a step-by-step walk through of the floorplan construction process for the classroom building – (a) is the ground truth floorplan; (b) is the room adjacency graph; (c) is the initial floorplan after hallway layout learning; and (d) is the final floorplan after force directed dilation. The results show that the automatically constructed floorplan offers excellent match to the ground truth.

Figure 10 shows the research lab building case study, which also shows excellent match to the ground truth. In addition, detected floorplan mismatches are indicated. The following room pairs (indicated in red) had their positions swapped: 19–20, 40–41, and 49–50. For rooms 44, 45, 46 (marked in blue), the width is estimated incorrectly, because their widths are different from the rest of the rooms along the same hallway. Room 10 and 11 are elevators and emergency exits, which are not accessible to our study and were therefore not detected and lead to overestimates of the sizes of rooms 8 and 9.

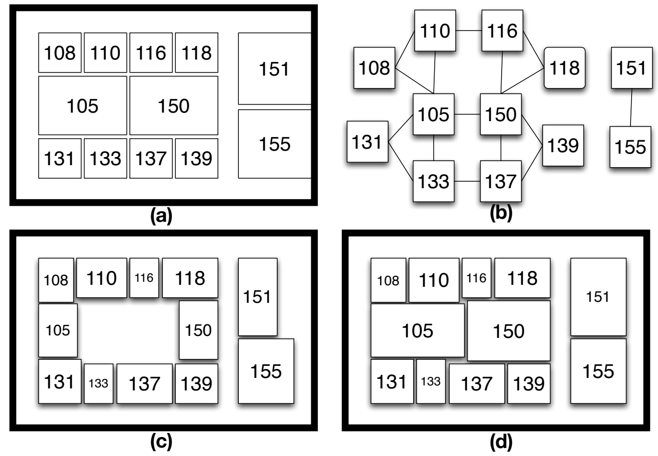

Figure 9. Classroom building floorplan case study.

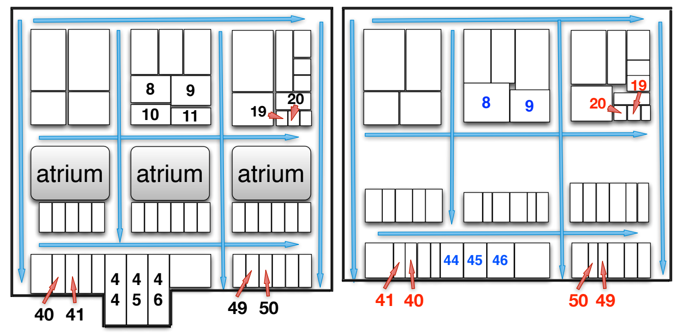

Figure 10. Research lab building floorplan case study. Left side is the original floorplan and right side is the learned one.

Figure 11 shows the office building case study, in which some rooms and hallways have irregular shapes. The result shows that our approach can still estimate room sizes and relative positions with reasonable accuracy even for irregular rooms and hallways. Figure 12 shows the shopping mall building case study, which has a main curved hallway and rooms along the hallway. The learned floorplan indicates that the room relations are learned correctly but the room size and shape are not exactly right. This is because we have no access to the hallway behind the stores and the room shape is mainly learned through our force-directed dilation algorithm instead of the hallway layout learning algorithm.

To quantitatively measure the quality of the constructed floorplan, the following three metrics are introduced:

- Position: A binary measurement for the relative position of two adjacent rooms with four directions: up, down, right, and left. Given two rooms, r_a and r_b, if

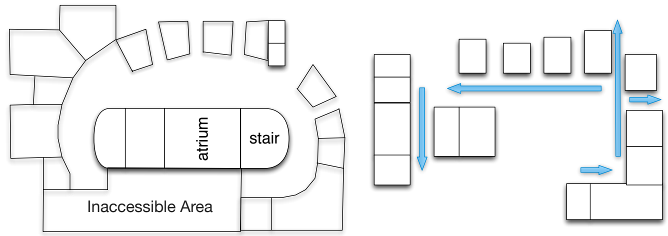

Figure 11. Office building floorplan case study. Left side is the original floorplan and right side is the learned one.

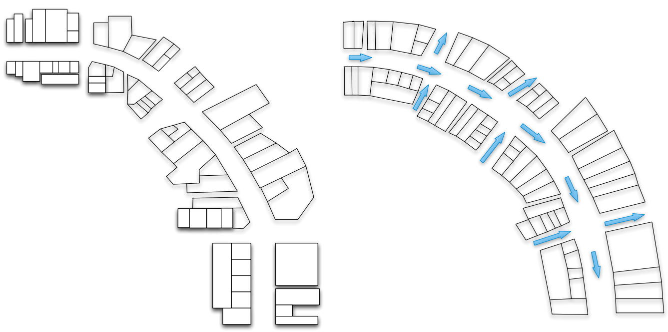

Figure 12. Shopping mall floorplan case study. Left side is the original floorplan and right side is the learned one.

any part of r_b is on the left side of room r_a's left wall, we say that r_b is on the left side of r_a, similar conventions are used for other directions. Also the position may be a combination of two directions, e.g., up-left. The position accuracy is defined as the total number of room pairs with correctly learned relative position divided by the total number of room pairs.

- Room area: The room length multiplied by width. Room area error is defined as the difference between learned room area and ground truth room area divided by the ground truth room area.

- Aspect ratio: The room geometric length-width ratio, which reflect the shape of a room. Ratio error equals to the difference of learned ratio and ground truth divided by real the ground truth.

Figure 13 shows the average position accuracy measure as a function of number of motion traces per hallway. Our system uses crowd-sourcing based learning method. In general, the more user data collected, the better its accuracy. Given the targeted accuracy measure, a good learning method should converge quickly, and therefore have a low crowd-sourcing data requirement. Given different numbers of motion traces collected per hallway, the corresponding floorplan accuracies are shown in this figure. When 20 traces are collected in each hallway, the position accuracy converges to 91%. Note that each motion trace can start and end at any point in a hallway.

Figure 14 shows the area error measure with and without the forced directed dilation. The results show that force directed dilation consistently improves room area

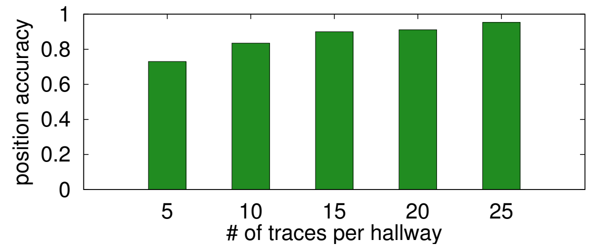

Figure 13. Position measure.

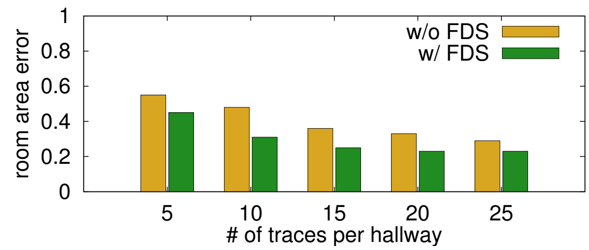

Figure 14. Area measure.

estimation accuracy by 10%, achieving a stable error measure of 33% with 20 data points per location.

Figure 15 shows the average aspect ratio measure. This study shows that, force directed dilation consistently helps reduce the error measure, converging to a 24% error rate with 20 data points per location.

RELATED WORK

In this section, we survey related work spanning the following areas: indoor mapping techniques, user behavior sensing techniques, and Wi-Fi fingerprint based indoor localization.

Floorplan construction

Simultaneous Localization and Mapping (SLAM) is a technique to build a map in an unknown environment, while simultaneously tracking the current location [2]. This technique is widely used in robotics. However, the map constructed in SLAM consists of a set of significant points, instead of floorplan. SmartSLAM applies the SLAM approach on the smartphones [12]. It utilizes the Wi-Fi signals collected when users are walking to generate maps of a building. However, SmartSLAM focuses on just hallway layout instead of the floorplan of the whole building. For example, it cannot derive the room dimension and position information. Alzantot et al. have proposed a floorplan construction approach, CrowdInside, based on user motion traces in the building [1].

Motion detection

There is a rich literature on user behavior sensing technologies [9, 7, 6, 10]. In this work, we implemented the walking detection technology proposed by Jiang et al. [6] because it has good energy efficiency. Turn detection has been studied in previous work [7] using accelerometer and compass on smartphones. However,

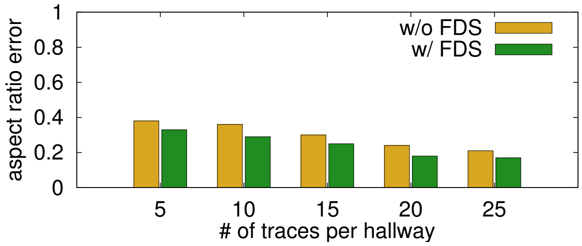

Figure 15. Aspect ratio measure.

their technique is targeted for outdoor scenarios and thus does not consider the impact of indoor environment on the accuracy of compass.

Comparisons

In contrast with our work, CrowdInside [1] does not require Wi-Fi coverage and it works well for irregular rooms and hallways. However, CrowdInside relies on landmarks (e.g., elevator and stairways), which are not always available. Moreover, it requires that user traces cover all edges of a room in order to estimate its shape. This requirement may not be met in practice because users typically concentrate in portions of rooms, e.g., the desk area. In addition, the edges of rooms may be inaccessible to users, e.g., blocked by furniture. In our future work, we would like to investigate leveraging their technique to further improve our system for irregular rooms and hallways while maintaining the benefit of our approach that only rely on users' motion in hallways.

We also compared our work with two state-of-the-art Wi-Fi based room localization systems: ARIEL [6] and LiFS [16]. ARIEL [6] is a room fingerprinting system that automatically clusters Wi-Fi signals by room. In ARIEL, rooms have automatically assigned unique room IDs and users can label rooms by their preference. It works well when the number of rooms is small for each individual user, but a large number of room labels would be difficult for a user to memorize. In addition, this approach only provides the room information without any contextual information, e.g., room size and relations to other rooms. Our system, built upon the ARIEL system, can automatically construct a floorplan for users to tackle the above problems.

LiFS [16] assumes a floorplan is available and it automatically maps Wi-Fi signals collected from users to the existing floorplan. Their approach works well in buildings whose floorplans are available. Otherwise substantial effort of manual floorplan construction may be required. Our system deals with the case when floorplan is not available. LiFS is appropriate when floorplans are available.

CONCLUSIONS

This paper has presented an automatic floorplan construction system targeting indoor environments. The proposed solution uses on-site Wi-Fi infrastructure. It also uses information on the motion of occupants gathered via crowd sourcing. Our analysis and optimization techniques determine the identities and geometries of individual rooms, as well as room adjacency information, and then construct an indoor floorplan through hallway layout learning and force directed dilation. The system functions for a variety of indoor environments, and is robust to Wi-Fi noise and variation in occupant motion patterns. Deployment at three building sites with 189 rooms yields an average room position accuracy of 91%, room area estimation error of 33% and room geometric aspect ratio error of 24%, using on average only 20 data points per location for the system to converge.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under awards CNS-0910995 and CNS-0910816.

REFERENCES

1. M. Alzantot and M. Youssef. Crowdinside: automatic construction of indoor floorplans. In *SIGSPATIAL '12*, 2012.
2. H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *Robotics Automation Magazine, IEEE*, 2006.
3. P. Eades. A Heuristic for Graph Drawing. *Congressus Numerantium*, 1984.
4. A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom '04*, 2004.
5. Y. Jiang, K. Li, L. Tian, R. Piedrahita, X. Yun, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. MAQS: A personalized mobile sensing system for indoor air quality monitoring. In *UbiComp '11*, 2011.
6. Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. ARIEL: Automatic Wi-Fi based room fingerprinting for indoor localization. In *UbiComp '12*, 2012.
7. M. B. Kjærgaard, S. Bhattacharya, H. Blunck, and P. Nurmi. Energy-efficient trajectory tracking for mobile devices. In *MobiSys '11*, 2011.
8. M. B. Kjærgaard, G. Treu, and C. Linnhoff-Popien. Zone–based RSS Reporting for Location Fingerprinting. In *Pervasive '07*, 2007.
9. M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Tröster. Detecting pedestrian flocks by fusion of multi-modal sensors in mobile phones. In *UbiComp '12*, 2012.
10. F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *UbiComp '12*, 2012.
11. J.-G. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. In *MobiSys '10*, 2010.
12. H. Shin, Y. Chon, and H. Cha. Unsupervised construction of an indoor floor plan using a smartphone. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 2012.
13. A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod. Accurate, low-energy trajectory mapping for mobile devices. In *NSDI'11*, 2011.
14. W. T. Tutte. How to draw a graph. *London Math*, 1963.
15. U.S. Environmental Protection Agency Green Building Workgroup. Buildings and their impact on the environment: A statistical summary, 2009.
16. Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: wireless indoor localization with little human intervention. In *MobiCom '12*, 2012.