The existence of a worldwide indoor floorplans database can lead to significant growth in location-based applications, especially for indoor environments. In this paper, we present CrowdInside: a crowdsourcing-based system for the automatic construction of buildings floorplans. CrowdInside leverages the smart phones sensors that are ubiquitously available with humans who use a building to automatically and transparently construct accurate motion traces. These accurate traces are generated based on a novel technique for reducing the errors in the inertial motion traces by using the points of interest in the indoor environment, such as elevators and stairs, for error resetting. The collected traces are then processed to detect the overall floorplan shape as well as higher level semantics such as detecting rooms and corridors shapes along with a variety of points of interest in the environment

Implementation of the system in two testbeds, using different Android phones, shows that CrowdInside can detect the points of interest accurately with 0.2% false positive rate and 1.3% false negative rate. In addition, the proposed error resetting technique leads to more than 12 times enhancement in the median distance error compared to the state-of-the-art. Moreover, the detailed floor- plan can be accurately estimated with a relatively small number of traces. This number is amortized over the number of users of the building. We also discuss possible extensions to CrowdInside for inferring even higher level semantics about the discovered floor plans.

During the last decade, there has been a rapid growth in location-based applications, including location-enabled social networking, direction finding, and advertisement. This has been driven by the flourishing of smart phones and mobile devices, location determination technologies, and wireless Internet connectivity. A key requirement to many of these location-based applications is the avail- ability of a map to display the user location on. This map can be a street map, in case of outdoor applications, or a floorplan, in case of indoor applications. Traditionally, outdoor location-based services providers, such as Google Maps, Bing Maps, FourSquare, etc, provide outdoor street maps for almost all regions around the globe. However, the indoor equivalent floorplans are currently very limited, affecting the ubiquity and spread of indoor location-based applications. Recently, a number of commercial systems for indoor direction finding have started to emerge, e.g. Point Inside and Micello Indoor Maps. In late 2011, Google Maps started to pro- vide detailed floorplans for a few malls and airports in the U.S. and Japan. Nevertheless, all these systems depend on manually building the floor plan. Manual addition/editing of all buildings floorplans around the world requires an enormous cost and effort which may be unaffordable. In addition, keeping these floorplans up to date is another challenge.

In this paper, we introduce CrowdInside as a automatic floorplan construction system. CrowdInside leverages the ubiquity of smart phones to infer information about the building floorplan along with other semantic information. In particular, today’s smart phones have an array of sensors, e.g. inertial sensors (accelerometers, com- passes, and gyroscopes), that can be used to construct traces of movement in a transparent manner to the users. People walking in their homes, offices, and even visitors collect these traces and send them for processing by CrowdInside. Using this crowdsourcing approach, CrowdInside can provide the general layout of a building, identify the rooms and corridor locations and shapes, along with identifying other points of interest, such as elevators, stairs, and escalators.

* We present the CrowdInside system architecture for leveraging the smart phones sensors in a crowdsourcing approach to automatically estimate the indoor floorplans for virtually any building around the globe.
* We provide techniques for estimating points of interests (or anchor points) in the environment (such as building entrances, elevators, stairs, and escalators) based on the phones inertial sensors with high accuracy.
* We provide a novel technique for constructing accurate in- door user traces based on the noisy inertial sensors in today’s commodity smart phones. The proposed technique depends on resetting the accumulation of error by leveraging the detected anchor points.
* We employ classification techniques to separate corridors from rooms and further apply clustering techniques to separate the rooms from each other.
* We show how to identify the rooms shapes using computational geometry techniques.

Finally, we implement the system on different Android phones (Samsung Nexus S, Nexus One, Galaxy Ace and Galaxy Tab) and evaluate it in a campus building and a mall.

We start by giving an overview of the CrowdInside system and how it can construct accurate traces in Section 2. Section 3 gives the details of the floorplan estimation module.

图示

描述已自动生成

Figure 2 shows our system architecture. The system consists of three main module: (a) the Data Collection Module is responsible for collecting measurements from users’devices, (b) the Traces Generation Module is responsible for building accurate motion traces based on a novel anchor-based error resetting technique and (c) the Floorplan Estimation Module that separates the corridors from the rooms and detects the rooms boundaries. We describe the details of the first two modules in this section and leave the details of the Floorplan Estimation Module to Section 3.

## 2.1 Data Collection Module

The time-stamped measurements collected can be buffered and then sent opportunistically to server in the cloud for later processing when a connection is available to reduce the communications cost and/or save energy. Data collected are measurements from sensors including: accelerometers, magnetometers, gyroscopes, and the received WiFi signal strength values from available access points. The GPS is also queried with a low duty cycle to detect the user’s transition from outdoors to indoors. The duty cycle can be set adaptively according to user’s current position (e.g. more frequently when the user approaches the border of the building.)

## 2.2 Traces Generation Module

To address this problem, we rely on a dead-reckoning based approach.

the current location (Xk; Yk) is estimated with the help of the previous location (Xk; Yk), distance traveled (S), and direction of motion (θ) since the last estimate as:

图示

中度可信度描述已自动生成

θ can be estimated from the magnetometer and/or the gyroscope.

while the displacement S can be obtained from the accelerometer.

The initial position is the last known GPS coordinate, detected by the loss of the GPS signal.

Anchor points are points in the environ- ment with unique sensor signatures that can be used to reset the trace error when the user hits one of them as shown in Figure 4. In particular, we identify two classes of anchor points: those based on the GPS sensor (building entrances and windows) and those based on inertial sensors (stairs, elevators, escalators, room doors, etc).

图表, 折线图

描述已自动生成

1. *GPS-based Anchor Points s*

In particular, the building entrance location is uniformly dis- tributed in the interval between the last obtained GPS position and the first loss of the GPS signal (Figure 5). Using the law of large numbers, the building entrance position can be estimated with high accuracy by averaging a large number of samples as quantified in Section 4.2.1.

Therefore, whenever the loss of the GPS signal is detected, the user position can be reset based on the position of the nearest build- ing entrance/window, enhancing the trace accuracy. This also helps in reducing the error in the trace starting point.

图示, 日程表

描述已自动生成

1. Inertial-based Anchor Points

Our focus in this section is on defining a set of rules that enable us to clearly identify elevators, escalators, and stairs as an example of the anchor points that can be identified using the inertial sensors and separating them from other patterns such as normal walking and being stationary.

图示

描述已自动生成

Figure 6 shows a classification tree for detecting the three classes of interest: elevators, escalators, and stairs. Note that a false positive leads to errors in estimating the location of the anchor point while a false negative leads to missing an opportunity for synchronization. Therefore, high accuracy in detection with low false positive and negative rates are highly desired. We note also that different features can be used to detect the same class accurately, as we show below.

图示

描述已自动生成

### Elevator:

The elevator has a unique acceleration pattern that makes it easily distinguishable with high accuracy (Figure 7). A typical elevator usage trace consists of a normal walking period, followed by waiting for the elevator for some time, walking into the elevator, standing inside, an over-weight/ weight loss occurs (depending on the direction of the elevator), then a stationary period which depends on the number of the floors the elevator moved, another weight-loss/over-weight period, and finally a walk-out. To recognize the elevator motion pattern, we developed a Finite State Machine (FSM) that depends on the observed state transitions (Figure 8). The detected direction of motion (based on the order of the weight-loss over-weight events) and the number of floors traveled (based on the time or displacement during the inside-elevator period), can be used to further enhance the accuracy.

图示

描述已自动生成

### Escalator:

Once the elevator has been separated, the key observation that distinguishes the constant speed scenarios (escalator/stationarity) from the dynamic scenarios (stairs/walking) is that users do not move their legs in the constant speed scenarios. Moving legs has a significant effect on the variance of the acceleration pattern (Figure 9).

To further separate the escalator from stationarity, we found that the variance of the magnetic field when the user is stationary is much less than the case when she is using an escalator (Figure 10). We believe that this is due to the change of location in the case of the escalator and the presence of the powerful motor of the escalator.

图表, 折线图

描述已自动生成

### Stairs

1. To distinguish walking and stairs, using Y/Z axis. Figure 11 shows that the correlation between the acceleration in the Y and Z axes can be a good mea- sure to separate the two cases. The intuition is that when the user is using the stairs, her speed increases or decreases based on whether the gravity is helping her or not. This creates a higher correlation between the acceleration in the direction of motion and direction of gravity as compared to walking.

图形用户界面, 图表

描述已自动生成

1. To distinguish between going upstairs and going downstairs.

Furthermore, our measurements show that climbing downstairs exhibit a higher motion intensity than climbing up (as the gravity is helping the user in the former case). This is reflected in different features. For example, Figure 12 shows that the peak of the acceleration magnitude can be used to differentiate between the stairs up/down case.

图表, 折线图

描述已自动生成

## FLOORPLAN ESTIMATION MODULE

(1) the overall shape

(2) the room-corridors details

### 3.1 Overall Floorplan Shape

In order to automatically estimate the overall floorplan shape, we represent each user step by a point. The goal is to estimate the best shape that represents the point cloud generated from all collected traces (Figure 13(b)).

图示, 工程绘图

描述已自动生成

We found that alpha shapes is a general tool that can capture the building shape with high accuracy.

Use α-shape to get the shape of the building

### 3.2 Detailed floorplan

#### 3.2.1 Traces segmentation and filtering

The first step in our approach is to break the continuous motion traces into segments. Segments are straight parts of the trace that are separated by either turns or pauses (periods of in activities).

In particular, consecutive segments are separated by significant changes in the direction of motion (we have chosen the threshold to be 45). The intuition is that a segment will be inside the same area (corridor/ room/ hall)

Finally, we filter the segments by excluding short segments in terms of both time and/or distance as we found that those segments are not descriptive.

#### 3.2.2 Segments classification

To distinguish corridors and rooms.

* **Average time spent per step in the segment:** This feature represents the average time spent between individual steps in the segment. The intuition is that, typically, the user walks faster through corridors than rooms.
* **Segment length:** Since we start a segment at each significant change in direction, this feature captures the intuition that the segments in corridors should be longer than rooms.
* **Neighbor traces density:** The intuition here is that the segments in the corridors are more dense (as more users use them) than the segments in rooms (as shown in the point cloud in Figure 13(b)).

图示

描述已自动生成

#### 3.2.3 Segments clustering

We use a density-based clustering algorithm (DBSCAN) to group segments that lie close to each other into one cluster. To prevent segments from adjacent rooms to be grouped together and reduce the effect of the traces noise that may cross the walls be- tween rooms, we use the center point of each segment for the clustering (rather than all the points in the segment). The similarity measure used for clustering is the distance between the location of center points and the similarity between the measured WiFi signals at these points. Figure 14(c) shows the clusters generated using the segments center points.

#### 3.2.4 Shaping

we calculate the α-shape of the points corresponding to all the segments that belong to each room separately (as generated by the clustering module). further smoothing can be applied to the obtained rooms and corridors, e.g. to make them rectangular.

#### 3.2.5 Estimating doors positions

图示

描述已自动生成

To estimate the locations of the room-doors, we extract all the intersection points of two segments; one of which is of a corridor type while the other is of a room type. The distribution of those points of intersection is shown in Figure 15(a). We apply a spatial clustering algorithm (DBSCAN) on these points based on the Euclidean distance between points as a similarity measure. Each cluster corresponds to a door whose centroid is taken as the esti- mated door location. Figure 15(b) shows the estimated locations of doors.