SemanticSLAM: Using Environment Landmarks for Unsupervised Indoor Localization

Heba Abdelnasser, *Student Member, IEEE,* Reham Mohamed, *Student Member, IEEE,* Ahmed Elgohary, *Student Member, IEEE,* Moustafa Farid, *Student Member, IEEE,* He Wang, *Student Member, IEEE,* Souvik Sen, *Member, IEEE,* Romit Roy Choudhury, *Senior Member, IEEE,* and Moustafa Youssef, *Senior Member, IEEE*

Abstract—Indoor localization using mobile sensors has gained momentum lately. Most of the current systems rely on an extensive calibration step to achieve high accuracy. We propose *SemanticSLAM*, a novel unsupervised indoor localization scheme that bypasses the need for war-driving. *SemanticSLAM* leverages the idea that certain locations in an indoor environment have a unique signature on one or more phone sensors. Climbing stairs, for example, has a distinct pattern on the phone's accelerometer; a specific spot may experience an unusual magnetic interference while another may have a unique set of WiFi access points covering it. *SemanticSLAM* uses these unique points in the environment as landmarks and combines them with dead-reckoning in a new Simultaneous Localization And Mapping (SLAM) framework to reduce both the localization error and convergence time. In particular, the phone inertial sensors are used to keep track of the user's path, while the observed landmarks are used to compensate for the accumulation of error in a unified probabilistic framework.

Evaluation in two testbeds on Android phones shows that the system can achieve 0.53 meters human median localization errors. In addition, the system can detect the location of landmarks with 0.83 meters median error. This is 62% better than a system that does not use SLAM. Moreover, *SemanticSLAM* has a 33% lower convergence time compared to the same systems. This highlights the promise of *SemanticSLAM* as an unconventional approach for indoor localization.

Index Terms—Unconventional localization, semantic SLAM, indoor localization, unsupervised localization.

1 Introduction

Although GPS is considered a ubiquitous outdoor localization technology, there is still no equivalent indoor technology that can provide similar accuracy and scale. This can be due to a number of reasons: First, a class of indoor localization technologies, e.g. [2]–[7] depends on special hardware installment, which in turn limits their scalability. Second, WiFi-based localization systems, e.g. [8]–[17], offer ubiquitous localization, however, they require tedious calibration effort. Third, to reduce this calibration effort, a number of systems have been proposed, e.g. [16], [18], [19]; nevertheless, in order to do that, they usually need to sacrifice accuracy.

Recently, techniques that leverage the inertial sensors (mainly the accelerometer, gyroscope, and compass) on cell phones have been proposed [20], [21]. Such techniques depend on dead-reckoning, where the accelerometer signal is used to count the user steps and the compass to determine the user direction. However, since dead-reckoning error accumulates quickly, re-calibration of the user location is required. This is usually performed using the GPS in outdoor environments. However, GPS is unreliable indoors, and hence, other approaches are required for error resetting.

In this paper, we propose *SemanticSLAM*, a system that leverages the smart phone sensors to detect unique points in the indoor environment, i.e. semantic landmarks, that can be used to reset the dead-reckoning error indoors. Starting from a building floorplan that is either manually entered or automatically generated [22]–[24], *SemanticSLAM* discovers the landmarks and their locations in a crowd-sensing approach based on the data collected from

- *Heba Abdelnasser and Reham Mohamed are with the Wireless Research Center, Egypt-Japan University for Science and Technology (E-JUST), Alexandria, Egypt.*
  E-mail: {heba.abdelnasser,reham.mohamed}@ejust.edu.eg
- *Ahmed Elgohary is with the University of Maryland, USA. This work was performed when the author was at E-JUST, Egypt.*
  E-mail: elgohary@cs.umd.edu
- *Moustafa Farid is with the University of California, Los Angeles, USA. This work was performed when the author was at E-JUST, Egypt.*
  E-mail: malzantot@ucla.edu
- *He Wang and Romit Choudhury are with the University of Illinois at Urbana-Champaign, USA.*
  E-mail: {hewang5,croy}@illinois.edu
- *Souvik Sen is with HP Labs, USA.*
  E-mail: souvik.sen@hp.com
- *Moustafa Youssef is with Egypt-Japan University for Science and Technology (E-JUST), Alexandria, Egypt. Currently on sabbatical from Alexandria University, Egypt.*
  E-mail: moustafa.youssef@ejust.edu.eg

An earlier version of this paper appeared in the proceedings of the ACM Tenth International Conference on Mobile Systems, Applications, and Services (MobiSys) 2012 [1].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2478451, IEEE Transactions on Mobile Computing

the building users and their dead-reckoned locations. These discovered landmarks are then used to reset the error in the dead-reckoning estimation and hence leads to better localization accuracy. Note that this recursive dependence between estimating the landmark location and the user location lends itself naturally to the Simultaneous Localization And Mapping (SLAM) framework commonly used in the robotics domain [25]. Therefore, at the core of *SemanticSLAM* is a novel SLAM framework that handles the characteristics of semantic landmarks while being robust to landmark recognition errors.

A semantic landmark is defined by two attributes: its sensors pattern and physical location. Based on this, *SemanticSLAM* identifies two types of semantic landmarks: seed landmarks and organic landmarks. When both attributes of a semantic landmark are known a priori, the landmark is defined as a seed landmark, which can be mapped to the physical environment. For example, a person using the elevator will have a unique known pattern affecting the phone acceleration. On the other hand, the attributes, i.e. pattern and physical location, of an organic landmark cannot be known a priori. Therefore, *SemanticSLAM* learns them in an unsupervised manner. For example, a point in the building with a dead cellular reception can be used as an organic landmark. Note that a seed landmark location and pattern can also be learned, if needed, in an unsupervised way using the same technique used for organic landmarks. However, entering them initially bootstraps the system and speeds convergence.

Evaluation of *SemanticSLAM* using Android phones in a university building and a mall shows that the system can reach 0.53m median accuracy for human location detection while localizing the landmarks to within 0.83m median error. In addition, *SemanticSLAM* leads to 33% enhancement in convergence time compared to systems that do not use the SLAM framework.

In Summary, the contributions of this paper are:

- We present the *SemanticSLAM* architecture and framework that leverages smart phone sensors to both dead-reckon the user location and identify semantic landmarks. These landmarks are used in a SLAM probabilistic framework to reset the accumulated localization error.
- We present supervised and unsupervised techniques for the automatic detection of both seed and organic landmarks. We show that adequate landmarks exist in indoor environments, leading to accurate localization with no calibration.
- We implement *SemanticSLAM* on different Android phones and evaluate it in two different testbeds quantifying its accuracy and fast convergence time.

The rest of the paper is organized as follows: Section 2 gives a background about the SLAM framework. Section 3 gives the architectural overview and the details of landmarks extraction. The proposed semantic SLAM framework is presented in Section 4. In Section 6, we present the system evaluation. Finally, we discuss related work and conclude the paper in sections 7 and 8 respectively.

2 Background

In this section, we provide a brief background on the Simultaneous Localization and Mapping (SLAM) framework as well as the advantage of the FastSLAM algorithm we choose as our implementation framework [26], [27].

2.1 Overview of the SLAM Framework

SLAM was originally used by mobile robots [25] to enable them to build an estimated map of an environment and, at the same time, use this map to deduce the robot location. To do that, the robot gathers information about sensed nearby landmarks and concurrently measures its own motion. Both types of measurements are noisy. SLAM provides a probabilistic framework to estimate both the map (Θ) along with the robot pose (location (x_t, y_t) and orientation (ϕ_t)). In particular, the goal in SLAM is to find the estimated pose $(\hat{s}_t)$ and map $(\hat{\Theta}_t)$ that maximize the following probability density function:

$$p(s_t, \Theta | u^t, z^t, n^t) \tag{1}$$

Where u_t is the robot motion update (displacement and heading) at time t obtained from the robot sensors, with $u^t = u_1, ..., u_t$ capturing the complete history, $z^t = z_1, ..., z_t$ are the history of landmark position observations relative to the user position, and $n^t = n_1, ..., n_t$ are data association variables, where n_t specifies the identity of the landmark observed at time t.

The traditional approach for estimating the probability density function in Eq. 1 was to use an Extended Kalman Filter (EKF) [28], [29]. The EKF approach represents the robot's map and pose by a high-dimensional Gaussian density over all map landmarks and the robot pose. The off-diagonal elements in the covariance matrix of this multivariate Gaussian represent the correlation between errors in the robot pose and the landmarks in the map. Therefore, the EKF can accommodate the natural correlation of errors in the map.

In the EKF approach, the probability density function $P(s_t, \Theta | u^t, z^t, n^t)$ is factorized into two independent models: a motion model $P(s_t | u_t, s_{t-1})$ and a measurement model $p(z_t | s_t, \theta_{n_t}, n_t)$, where θ_{n_t} is the location of landmark n_t observed at time t. The motion model describes how a control u_t, asserted in the time interval $[t-1, t)$, affects the user's pose. On the other hand, the measurement model describes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2478451, IEEE Transactions on Mobile Computing

how measurements evolve from state. Both models are traditionally modelled by nonlinear functions with independent Gaussian noise:

$$p(s_t|u_t; s_{t-1}) = h(u_t, s_{t-1}) + \delta_t \quad (2)$$

$$p(z_t|s_t, \Theta, n_t) = g(s_t, \theta_{n_t}) + \varepsilon_t \quad (3)$$

Here h and g are nonlinear functions, and δ_t and ε_t are Gaussian noise variables with covariance R_t and P_t, respectively.

One limitation of the EKF-based approach is the computation complexity, which is quadratic in the number of landmarks [27]. Another key limitation is the *data association problem*, i.e. how to determine the identity of the detected landmarks when multiple of them have a similar signature (e.g. two nearby stairs, elevators, or turns), which can lead to different maps based on the chosen association. Gaussians cannot represent such multi-modal distribution over the different candidate landmarks. The typical approach to handle this problem in the EKF-SLAM literature is to restrict the inference to the most probable landmark given the robots current map [30]–[32]. However, these tend to fail to converge when the estimated data association is incorrect. Other approaches have been proposed to interleave the data association decisions with map building to revise past data association decisions [33]–[36]. However, such approaches cannot be executed in real-time and hence cannot be used for human tracking.

The **FastSLAM** approach [26], [27] was introduced to address the issues of the EKF-SLAM approach. FastSLAM combines particle filters [37], [38] and extended Kalman filters. The idea is to exploit a structural property of the SLAM problem, where landmark estimates are conditionally independent given the robot path. In other words, correlations in the uncertainty among different landmarks arise only through robot pose uncertainty; If the robot's correct path is known, the errors in its landmark estimates are independent of each other. This observation allows FastSLAM to factor the posterior over poses and maps.

More formally, in FastSLAM the robot *path*, $s^t = (s_1, ..., s_t)$, is estimated as:

$$p(s^t, \Theta|z^t, u^t, n^t) = p(s^t|z^t, u^t, n^t) \prod_{n=1}^{N_L} p(\theta_n|s^t, z^t, n^t) \quad (4)$$

Where N_L is the number of landmarks. This factorization is exact and universal.

Since the user path is not known in advance, FastSLAM estimates the first term (the robot path (s^t)) by a particle filter, where each particle represents a possible path. Conditioned on these particles, the individual map errors are independent, hence the second term (mapping problem) can be factored into N_L separate problems, one for each landmark in the map. The individual landmark location probability density function ($p(\theta_n|s^t, z^t, n^t)$) is estimated using an EKF. More formally, the posterior of the m^{th} particle ($S_t^{[m]}$) contains a path $s^{t,[m]}$ and N_L landmark estimates denoted by the landmark type ($\hat{f}_{n,t}$), mean $\mu_{n,t}^{[m]}$ and covariance $\Sigma_{n,t}^{[m]}$:

$$S_t^{[m]} = s^{t,[m]}, \underbrace{\hat{f}_{1,t}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}}_{\text{Landmark } \theta_1}, ..., \underbrace{\hat{f}_{N,t}, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]}}_{\text{Landmark } \theta_{N_L}} \quad (5)$$

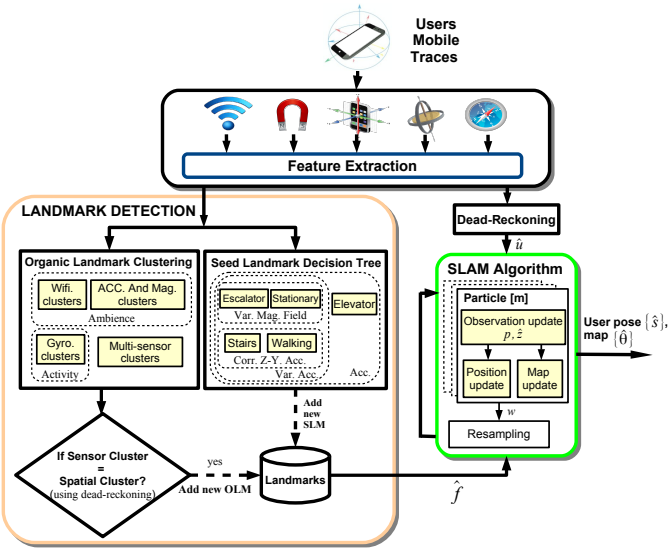


Fig. 1: SemanticSLAM System architecture.

2.2 Advantages of the FastSLAM Algorithm

The factorization employed by FastSLAM leads to an algorithm that is logarithmic in the number of landmarks, as compared to the quadratic time complexity for the EKF-SLAM. Moreover, data association decisions in FastSLAM can be made on a per-particle basis. Therefore, the algorithm maintains posteriors over multiple data associations, not just the most likely one as in the EKF-SLAM approach. This makes FastSLAM significantly more robust to data association problems [26], [27]. FastSLAM can also cope with non-linear models and is proven to converge under certain assumptions [27]. Therefore, we leverage the FastSLAM approach in *SemanticSLAM* due to these advantages.

3 System Overview

Figure 1 shows the system architecture. The system consists of four main modules: Sensor data collection and features extraction, landmark detection, dead-reckoning, and the *SemanticSLAM* framework. In the balance of this section, we give an overview of the different modules.

3.1 Sensor data collection and features extraction

Sensors data is collected from the users' mobile phones in a crowd-sensing manner. Collected sensors include inertial sensors (accelerometer, compass, and gyroscope) as well as WiFi and cellular access points and their associated signal strength. Note that inertial sensors have a low-energy profile while WiFi and cellular information is available during the phone normal operation. Therefore, *SemanticSLAM* has a minimal effect on the phone energy consumption.

Collected sensors data is then analyzed to extract the different features that can be used to identify the landmarks.

3.2 Dead-reckoning

Inertial sensors are combined to provide an estimate of the user location. Starting from a reference point, e.g. last GPS location of the person outside a building, the user next location is obtained based on the motion update measurement $u_t = \{l_t, \phi_t\}$, where l_t is the displacement and ϕ_t is the heading change at time t.

3.2.1 Displacement from the accelerometer

One possible solution to obtain the displacement is to double-integrate the accelerometer readings. However, due to the noisy cheap sensors on the phones, error accumulates quickly and can reach $100m$ within seconds [39]. A better approach [21], [39] is to use a step counting approach based on the human walking pattern. We use the UPTIME approach [39] as it adapts to the user step size based on her gait.

3.2.2 Orientation using compass/gyroscope

The magnetic field in indoor environments, due to ferromagnetic material and electrical objects in the vicinity, is very noisy, which can severely degrade the dead-reckoning performance. To address this issue, we fuse the gyroscope and magnetic sensor readings. The gyroscope provides accurate short term relative angle change while the magnetometer provides long term stability. In particular, we leverage the correlation between the two sensors readings to determine the points of time where the compass reading is accurate. We use these points as reference points (landmarks) to measure the relative angle from using the gyroscope until the detection of the next angular reference point [40].

3.3 Landmark Detection

Even though *SemanticSLAM*'s step counting approach reduces the dead-reckoning error accumulation, displacement error is still unbounded, which cannot be used for indoor tracking. Therefore, *SemanticSLAM* leverages a novel approach of detecting unique points in the environment, i.e. landmarks, that can be used to reset the errors. Specifically, whenever the user phone detects a landmark based on a unique multi-modal sensor signature, her position is reset to the position of this landmark, resetting the dead-reckoning error. We define two types of landmarks: seed landmarks (SLM) and organic landmarks (OLM).

Seed landmarks are landmarks that can be mapped to physical points in the environment and are used to bootstrap the system. Examples of SLMs include stairs, elevators, escalators, etc. Those SLMs have a unique effect on one or more of the phone sensors and hence can be uniquely detected.

On the other hand, organic landmarks do not necessarily map to an object and are detected based on their unique signature on the sensors. Usually, these are detected based on detecting consistent anomalies in one or more sensor patterns.

3.4 The SemanticSLAM Framework

Since the landmark location is estimated based on the user location, which in turn is a function of the detected landmark location; this recursive definition lends itself naturally to a SLAM framework. *SemanticSLAM* provides a novel framework that uses landmarks as observations to enhance both the user location estimation and the landmark identification.

In particular, the dead-reckoning state as well as the detected landmarks are fed into the SemanticSLAM algorithm which calculates the current pose of the tracked entity and updates the landmarks positions in a unified framework.

4 Landmarks Detection

Many points in indoor environments exhibit unique sensors signatures, which can be used as landmarks. Indoor environments are rich with ambient signals, like sound, light, magnetic field, temperature, WiFi, 3G, etc. Moreover, different building structures (e.g., stairs, doors, elevators) force humans to behave in certain ways.

In this section, we give the details of the detection of both the seed and organic landmarks.

4.1 Seed Landmarks

Seed landmarks (SLMs) are landmarks that can be associated with specific objects in the environment such as elevators and stairs. If the building floorplan is known (which is often necessary to visualize the user's location), then we can infer the locations of doors, elevators, staircases, escalators, etc. This implies that the locations of seed landmarks are immediately known. As long as the smartphone can detect these SLMs while passing through them, it can recalibrate its location. Thus, the goal of the SLM detection module is to define sensors patterns that are global across all buildings.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2478451, IEEE Transactions on Mobile Computing

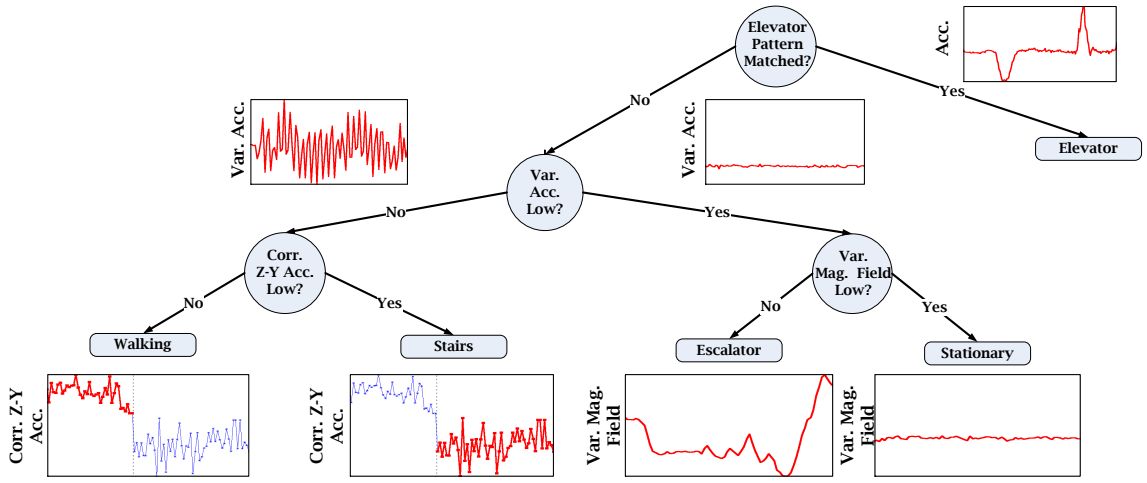


Fig. 2: Classification tree for detecting SLMs. The top level separates the elevator based on its unique acceleration pattern. The second level separates the constant velocity classes (stairs and escalator) from the other two classes (walking and stairs) based on the variance of the acceleration. The third level uses the variance of magnetic field to separate the escalator from the stationary case and the correlation between the Z and Y acceleration components to separate between the stairs and walking cases.

In this section, we discuss three inertial sensors-based of SLMs that are common in indoor environments: Elevators, Staircases, and Escalators. Inertial sensors have the advantage of being ubiquitously installed on a large class of smart phones, having a low-energy footprint, and being always on during the phone operation (to detect the change of screen orientation). Figure 2 shows a classification tree for detecting the three classes of interest and separating them from walking and being stationary.

Elevator: A typical elevator usage trace (Figure 3) consists of a normal walking period, followed by waiting for the elevator for some time, walking into the elevator, standing inside for a short time, 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, and finally a walk-out. The accelerometer shows distinct signatures in an elevator in the form of a pair of symmetric bumps in opposite directions, as shown in Figure 3. To recognize the elevator motion pattern, we use a Finite State Machine (FSM) that depends on the observed state transitions. Different thresholds are used to move between the states.

Evaluation over 22 traces shows that the thresholds are robust to changes in the testbed and can achieve 0.6% and 0% false positive and negative rates, respectively.

Escalator: Once the elevator has been separated, it is easy to separate the classes with constant velocity (escalator and stationary) from the other classes (walking and stairs) using the variance of acceleration. To further separate the escalator from stationarity, we found that the variance of the magnetic field can be a reliable discriminator (Figure 4) due to the motor of the escalator.

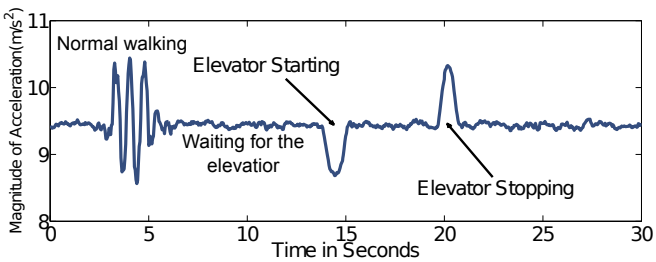


Fig. 3: Accelerometer signature inside the elevator (caused by the elevator starting and stopping).

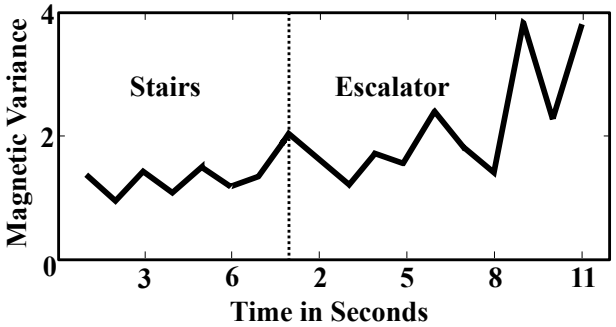


Fig. 4: Difference in magnetic variance when a user is climbing stairs and escalator.

Stairs: Once the scenarios with constant speed are separated, we need to differentiate between the stairs and walking case. The main observation here is that when the user is using the stairs, her speed increases or decreases based on whether the gravity is helping or not. This creates a higher correlation between the acceleration in the direction of motion and direction of gravity as compared to walking. As reported later, staircases can sometimes lead to false negatives

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2478451, IEEE Transactions on Mobile Computing

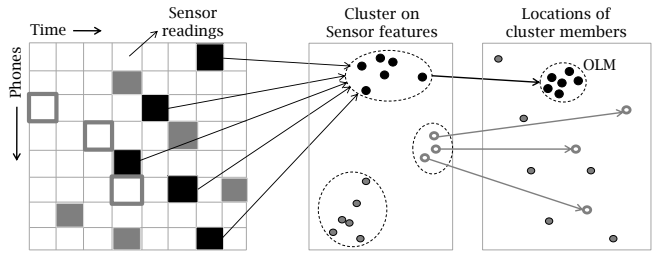


Fig. 5: A matrix showing sensor readings collected by devices across time. Readings are clustered and the location of cluster members is computed. If all cluster members fall within a small region, an OLM is detected.

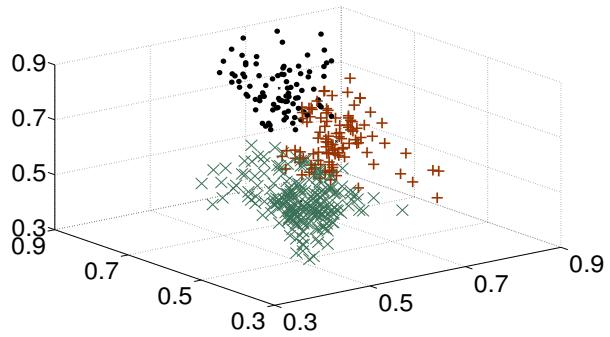


Fig. 6: Clusters identified by the K-means algorithm.

(1.8%).

4.2 Organic Landmarks

In addition to seed landmarks, some landmarks can be detected dynamically. Most of indoor environments offer some ambient signatures across one or many sensing dimensions. These signatures can be in the magnetic domain, where metals in a specific location may produce unique and reproducible fluctuations on the user's magnetometer near that location. Signatures could also be WiFi-based, a spot may overhear a set of WiFi base stations, but the set may change at short distances away from that spot. A few (dead) spots inside a building may not overhear any WiFi or GSM/3G signals, which by itself is a signature. Further, even a water-fountain could be a signature, users that stop to drink water may exhibit some common patterns on the accelerometer and magnetometer domains.

The task of discovering organic landmarks (OLMs) is rooted in (1) recognizing distinct patterns from many sensed signals, (2) and testing whether a given pattern is spatially confined to a small area. Figure 5 illustrates the flow of operations. All the sensor readings are gathered in a matrix: element <i,j> of the matrix contains sensor readings from phone i at time j. These sensor readings are essentially *features* of the raw sensed values (from the accelerometer, compass, gyroscope, magnetometer, and WiFi). Features for the magnetic and inertial sensors include *mean, max, min, variance, mean-crossings*, while for WiFi, they are *MAC ID and RSSI*.

These features are normalized between $[-1, 1]$ and fed to a K-means clustering algorithm. The clustering process is executed for each individual sensing dimension, as well as their combinations (such as accelerometer and compass together). Figure 6, for example, shows the clusters from the magnetometer readings for $K = 3$. The clusters were recorded for different values of K. The goal is to identify clusters that have *low similarity* with all other clusters; this will suggest a good signature. For this, we compute the correlation between a given cluster and all other clusters – if the maximum correlation is less than a *similarity threshold*, this cluster is considered as a candidate for landmark.

To qualify the candidate cluster as an OLM, it must also be confined to a small geographical area. For this, we first test whether the members of a cluster are within the same WiFi area (i.e., they overhear the same WiFi APs). While this is necessary, it is not sufficient because many WiFi areas are large. Therefore, for clusters within a WiFi area, we compute the locations for each of their members. If locations of all cluster-members are indeed within a small area (we use $4m^2$) then we declare this cluster as an OLM. We found that using the accelerometer, the points inside one of the sensor clusters were scattered all over the indoor space; upon investigation, we detected that this cluster roughly captured walking patterns. On the other hand, another cluster that proved to be within the $4m^2$ area was from a magnetic signature near an electrical service room in the building. The location of the OLM is obtained through the *SemanticSLAM* recursive framework (Section 4).

While the above describes the generalized version of the OLM detection algorithm, the different sensing dimensions require some customization, discussed next.

4.2.1 WiFi Landmarks

We use MAC addresses of WiFi APs and their corresponding RSSI values as features. Only APs with RSSI stronger than a threshold are considered. Applying K-means clustering, we identify small areas ($4m^2$) that have low similarity with all locations outside that area. We compute the similarity of two locations, l_1 and l_2, as follows:

Let us denote the sets of WiFi APs overheard at locations l_1 and l_2 as A_1 and A_2, respectively. Also, let $A = A_1 \bigcup A_2$. Let $f_i(a)$ denote the RSSI of AP a, $a \in A$, overheard at location l_i; if a is not overheard at

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2478451, IEEE Transactions on Mobile Computing

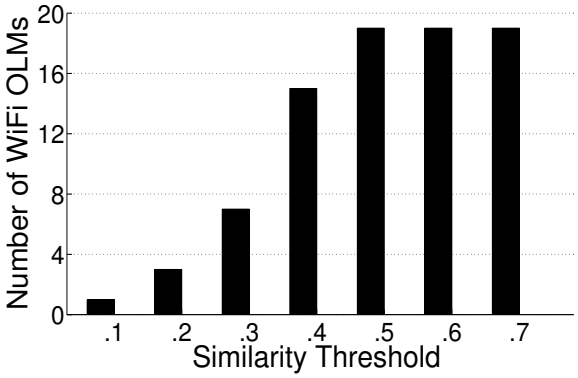


Fig. 7: Tradeoff between similarity threshold and number of WiFi landmarks.

l_i, then $f_i(a) = 0$. We now define similarity $S \in [0, 1]$, between locations l_1 and l_2 as:

$$S = \frac{1}{|A|} \sum_{\forall a \in A} \frac{\min(f_1(a), f_2(a))}{\max(f_1(a), f_2(a))}$$

The rationale for this equation is to add proportionally large weights to S when an AP's signals are similarly strong at both locations, and vice versa. We choose a threshold of 0.4 in our system to define a WiFi landmark, indicating that all locations within the WiFi landmark need to exhibit less than 0.4 similarity with any other location outside the landmark. Figure 7 shows this tradeoff using traces from the Engineering Building. We observed that 0.4 was a reasonable cut-off point, balancing quality and quantity of WiFi OLMs. In addition, we found that this value gives comparable performance for the two testbeds.

4.2.2 Magnetic and Inertial Sensor Landmarks

Indoor environments are characterized by at least a few turns (at the end of corridors, into offices, classrooms, stairs, etc.). Since the gyroscope offers reliable angular displacements, we recognize the opportunity to use them as organic landmarks. We design a special feature called the *bending coefficient*. Essentially, the coefficient captures the notion of path curvature, computed as the length of the perpendicular from the center of a walking segment to the straight line joining the end-points of the segment. We compute the bending coefficient over a sliding window on the user's walking path, and use them as a separate feature. Later, when we cluster on bending coefficient and WiFi together as features, similar turns within a WiFi area gather in the same cluster. The turns in the cluster could still be doors of adjacent classrooms in a corridor – these turns may very well lie within the same WiFi area. To avoid gathering all these turns into the same landmark, we check if the cluster is confined to within a $4m^2$ area; only then is the cluster declared a landmark.

Notation	Definition
$\hat{f}$	The detected landmark type (e.g. elevator pattern)
$\hat{u}_t$	The control data at time t (the dead-reckoning input)
$\hat{u}^t$	The history of control data t ($\hat{u}^t = \hat{u}_1, \hat{u}_2, ..., \hat{u}_t$)
$\hat{l}_t$	The estimated displacement at time t from the sensors
$[m]$	The particle index
$L_t^{[m]}$	The sampled step length for particle $[m]$ at time t
$\hat{\phi}_t$	The sensors heading change estimate at time t
$\Phi_t^{[m]}$	The sampled heading change for the particle $[m]$
s_t	The user's pose at time t ($s_t = \{s_t^x, s_t^y, s_t^\phi\}$)
$\hat{s}_t$	The predicted pose at time t
s^t	The posterior over the entire path ($s^t = s_1, s_2.....s_t$)
$S_t^{[m]}$	The posterior over all path and landmark positions for one particle ($S_t^{[m]} = s^{t,[m]}, \underbrace{\hat{f}_{1,t}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}}_{\text{Landmark } \theta_1}, ..., \underbrace{\hat{f}_{N,t}, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]}}_{\text{Landmark } \theta_{N_L}}$)
z_t	The measurement of landmark position at time t ($z_t = 0$)
$\hat{z}_{\hat{n}_t}^{[m]}$	The measurement estimation of particle $[m]$ with observed landmark $\hat{n}$ at time t
Θ	The landmark map
θ_{n_t}	The location of the landmark n at time t
Q_t	The landmark observation covariance matrix at time t
R_t	The measurement covariance at time t
P_t	The covariance matrix of the control data at time t
$\mu_{s_t}^{[m]}$	The mean of the estimated pose s_t at time t for particle $[m]$
$\sum_{s_t}^{[m]}$	The covariance matrix of the estimated pose s_t at time t for particle $[m]$
η	Normalization factor
p_n	The likelihood correspondence of landmark n with the observed pattern $\hat{f}$ at time t
p_0	The likelihood of observing a landmark for the first time
$K_t^{[m]}$	Kalman gain for particle $[m]$ at time t
$\mu_{\hat{n}_t,t}^{[m]}$	The mean of the estimated Gaussian position of landmark $\hat{n}_t$ at time t for particle $[m]$
$\sum_{\hat{n}_t,t}^{[m]}$	The covariance matrix of the estimated Gaussian position of landmark $\hat{n}_t$ at time t for particle $[m]$
$w_t^{[m]}$	The weight of particle $[m]$ at time t

TABLE 1: Notations used in the paper.

5 SemanticSLAM: Semantic Simultaneous Localization And Mapping

Once the landmarks have been detected, our system combines them with a dead-reckoning approach to both estimate the person's location and the landmark location. For that, we propose a modified FastSLAM algorithm [27] with *unknown* data association with changes to cope with the semantic landmarks detection. FastSLAM has the advantage of proved convergence even with a single particle, where it has a constant update time in this case, as well as it incorporates the measurement in the motion mode. This last feature allows it to handle the case when the noise in the person motion is large relative to the measurement noise. Table 1 summarizes the notations used in this section.

The algorithm consists of three steps: sampling, map update, and re-sampling. Without loss of gen-

erality, we assume that only a single landmark is observed at each time t.

5.1 Sampling the user's pose

The first step of SemanticSLAM algorithm is estimating the current pose of the tracked entity. Given the control data $\hat{u}_t$ obtained from the dead reckoning step, the measurement z_t, and the previous pose $s_{t-1}^{[m]}$, the current pose $s_t^{[m]}$ for each particle $[m]$ is sampled by the following probability distribution:

$$s_t^{[m]} \sim P(s_t|s^{t-1,[m]}, \hat{u}^t, z^t, n^t) \tag{6}$$

This distribution can be divided into the product of two factors: the next state distribution, and the probability of the measurement z_t as:

$$P(s_t|s^{t-1,[m]}, \hat{u}^t, z^t, n^t) = \underbrace{p(s_t|s_{t-1}^{[m]}, u_t)}_{s_t \sim \mathcal{N}(h(s_{t-1}^{[m]}, u_t), P_t)} .$$

$$\eta^{[m]} \int \underbrace{p(z_t|\theta_{n_t}, s_t, n_t)}_{z_t \sim \mathcal{N}(g(\theta_{n_t}, s_t), R_t)} \underbrace{p(\theta_{n_t}|s^{t-1,[m]}, z^{t-1}, n^{t-1})}_{\theta_{n_t} \sim \mathcal{N}(\mu_{n_t,t-1}^{[m]}, \Sigma_{n_t,t-1}^{[m]})} \tag{7}$$

The next state distribution depends on the estimated control data $\hat{u}_t = \{\hat{l}_t, \hat{\phi}_t\}$ with displacement $\hat{l}_t$ and heading change $\hat{\phi}_t$:

$$\hat{s}_t^{[m]} \sim P(s_t|s_{t-1}^{[m]}, \hat{u}_t) \tag{8}$$

Assuming Gaussian-distributed errors for both $\hat{l}_t$ and $\hat{\phi}_t$, the sampled displacement and heading for particle $[m]$ at time t is calculated as:

$$L_t^{[m]} \sim \mathcal{N}(\hat{l}_t, \sigma_l) \tag{9}$$

$$\phi_t^{[m]} \sim \mathcal{N}(\hat{\phi}_t, \sigma_\phi) \tag{10}$$

where σ_l and σ_ϕ are the variance of the displacement and heading estimation errors respectively.

Therefore, the user's sampled pose in Eq. 8 can be rewritten as:

$$\hat{s}_t^{[m],\phi} = s_{t-1}^{[m],\phi} + \phi_t^{[m]} \tag{11}$$

$$\hat{s}_t^{[m],x} = s_{t-1}^{[m],x} + L_t^{[m]} \cos(\hat{s}_t^{[m],\phi}) \tag{12}$$

$$\hat{s}_t^{[m],y} = s_{t-1}^{[m],y} + L_t^{[m]} \sin(\hat{s}_t^{[m],\phi}) \tag{13}$$

Note that equations 9 through 13 are the implementation of the *non-linear* $h(.)$ function in Eq. 2.

The probability of the measurement z_t involves an integration over all possible landmark locations θ_{n_t}, which is not possible in the general case. To address this issue, the FastSLAM framework approximates $g(.)$ (from Eq. 3) as a linear function, leading to a closed form solution as:

$$g(\theta_{n_t}, s_t) \approx \hat{z}_{n_t}^{[m]} + G_{\theta,n}.(\theta_{n_t} - \mu_{n,t,t-1}^{[m]}) + G_{s,n}.(s_t - \hat{s}_{n,t}^{[m]}) \tag{14}$$

where $\hat{z}_{n,t}^{[m]} = g(\mu_{n,t-1}^{[m]}, \hat{s}_{n,t}^{[m]})$ is the predicted measurement, $\hat{s}_t^{[m]} = h(s_{t-1}^{[m]}, u_t)$ is the predicted user's pose (from Eq. 8), and $\theta_{n,t}^{[m]} = \mu_{n,t-1}^{[m]}$ is the predicted landmark location. The matrices $G_{\theta,n}$ and $G_{s,n}$ are the Jacobians of $g(.)$ with respect to θ and s respectively.

Thus, the proposal distribution in Eq. 6 is Gaussian with the parameters:

$$\Sigma_{s_{n,t}}^{[m]} = [G_{s,n}^T Q_{n,t}^{[m]-1} G_{s,n} + P_t^{-1}]^{-1} \tag{15}$$

$$\mu_{s_{n,t}}^{[m]} = \hat{s}_{n,t}^{[m]} + \Sigma_{s_{n,t}}^{[m]} G_{s,n}^T Q_{n,t}^{[m]-1} (z_t - \hat{z}_{n,t}^{[m]}) \tag{16}$$

where $Q_{n,t}^{[m]} = G_{\theta,n} \Sigma_{n,t-1}^{[m]} G_{\theta,n}^T + R_t$ is the landmark observation covariance matrix, R_t is the measurement covariance matrix and z_t is the actual landmark position observation.

For the SemanticSLAM problem, we set:

$$\hat{z}_{n,t}^{[m]} = g(\mu_{n,t-1}^{[m]}, \hat{s}_{n,t}^{[m]}) = ||\mu_{n,t-1}^{[m]} - \hat{s}_{n,t}^{[m]}|| \tag{17}$$

reflecting that the measurement is the distance between the landmark and the current user's location. Moreover, we set $z_{n,t} = 0$ indicating that the landmark is observed when the user is at the landmark location.

Therefore, the final user position is sampled from the distribution $\mathcal{N}(\mu_{s_{n,t}}^{[m]}, \Sigma_{s_{n,t}}^{[m]})$

5.2 Map Update

Each particle has an independent map that contains the locations of the landmarks represented by their mean (μ), covariance matrix (Σ) and the associated landmark pattern (f). The purpose of the map update step is to update the location of the currently detected landmark. Due to the inherent sensors noise, there is ambiguity in landmark detection related to both the landmark locations and its type. Therefore, in SemanticSLAM, we compute the probability of actually observing each landmark (f) when the detected pattern is $\hat{f}$. This uncertainty is represented by a confusion matrix, where each cell (i, j) in the matrix represents $p(f_i|\hat{f}_j)$ for each landmark types i, j. Given this confusion matrix and the position uncertainty Q_n for each landmark n, the likelihood of correspondence with landmark n is calculated based on an EKF approximation [41] as:

$$p_n = \eta |2\pi Q_{n,t}|^{-\frac{1}{2}} \exp\{-\frac{1}{2} \hat{z}_{n,t}^T . Q_{n,t}^{-1} \hat{z}_{n,t}\} . p(f_t|\hat{f}_t) \tag{18}$$

where η is the normalization factor. Note that this landmark likelihood takes into account the distance between the landmark and the current user pose, the location uncertainty, and the confusion between the observed and actual landmark type.

We also assume that the probability of observing a new landmark given the detected pattern can be calculated as:

$$p_n = \eta p_0 . p(f_t|\hat{f}_t) \tag{19}$$

where p_0, the probability of observing a new landmark, is a constant determined empirically.

Finally, the landmark with the greatest probability ($\hat{n}_t$) is selected as the currently observed landmark and its location is updated using the standard EKF formulas as:

$$K_t^{[m]} = \Sigma_{\hat{n}_t,t-1}^{[m]} G_{\theta,\hat{n}_t}^T Q_{\hat{n}_t,t}^{[m]-1} \quad (20)$$

$$\mu_{\hat{n}_t,t}^{[m]} = \mu_{\hat{n}_t,t-1}^{[m]} - K_t^{[m]} \hat{z}_{\hat{n}_t,t}^T \quad (21)$$

$$\Sigma_{\hat{n}_t,t}^{[m]} = (I - K_t^{[m]} G_{\theta,\hat{n}_t}) \Sigma_{\hat{n}_t,t-1}^{[m]} \quad (22)$$

If it is more probable that the observed landmark is a new landmark, then a new landmark is added to the map, with the current position of the tracked entity $s_t^{[m]}$ as its location and the measurement uncertainty R_t as its covariance matrix.

Note that in *SemanticSLAM* inherits the property of FastSLAM that there are multiple hypothesis of the possible landmark, each corresponding to a different particle. In other word, each particle has its own belief in the currently observed landmark. This multi-hypothesis property provides *SemanticSLAM* with robustness to large errors as discussed in Section 2.

5.3 Resampling

Each particle represents an estimate of the tracked entity pose with a weight $w^{[m]}$ reflecting the confidence of the pose associated with this particle. The weights of the particles are initialized equally and updated at each step based on the likelihood of the detected landmarks, i.e. the landmark with the highest probability for each particle, as:

$$w_t^{[m]} = w_{t-1}^{[m]} . p_{\hat{n}_t}^{[m]} \quad (23)$$

The pose and the map of the particle which has the maximum weight are selected as the current estimate. Then, a resampling step is performed using the current weights in order to refine the particles and drop those that significantly deviated from the actual path.

Algorithm 1 summarizes the full algorithm.

Algorithm 1 SemanticSLAM(z_t, u_t, S_{t-1})

for $m = 1$ to M do **do**
  retrieve $\langle s_{t-1}^{[m]}, N_{t-1}^{[m]}, w_{t-1}^{[m]}, \langle f_{1,t-1}^{[m]}, \mu_{1,t-1}^{[m]}, \Sigma_{1,t-1}^{[m]} \rangle, ..., \langle f_{N,t-1}^{[m]}, \mu_{N,t-1}^{[m]}, \Sigma_{N,t-1}^{[m]} \rangle \rangle$ from S_{t-1}
  for $n = 1$ to $N_{t-1}^{[m]}$ do **do** ▷ calculate sampling distribution
    $\hat{s}_t^{[m]} \sim P(s_t | s_{t-1}^{[m]}, \hat{u}_t)$ ▷ predict pose
    $\hat{z}_{n,t}^{[m]} = g(\mu_{n,t-1}^{[m]}, \hat{s}_{n,t}^{[m]})$ ▷ predict measurement
    $\Sigma_{s_{n,t}}^{[m]} = [G_{s,n}^T Q_{n,t}^{[m]-1} G_{s,n} + P_t^{-1}]^{-1}$ ▷ Cov of proposal distribution
    $\mu_{s_{n,t}}^{[m]} = \hat{s}_{n,t}^{[m]} + \Sigma_{s_{n,t}}^{[m]} G_{s,n}^T Q_{n,t}^{[m]-1} (z_t - \hat{z}_{n,t}^{[m]})$ ▷ mean of proposal distribution
    $s_{n,t}^{[m]} \sim \mathcal{N}(\mu_{s_{n,t}}^{[m]}, \Sigma_{s_{n,t}}^{[m]})$ ▷ sample pose
    $p_n = \eta |2\pi Q_{n,t}|^{-\frac{1}{2}} \exp\{-\frac{1}{2} z_t^T . Q_{n,t}^{-1} z_t\} . p(f_t | \hat{f}_t)$ ▷ correspondance likelihood
  end for
  $p_{1+N_{t-1}^{[m]}} = \eta p_0 . p(f_t | \hat{f}_t)$ ▷ likelihood of new landmark
  $\hat{n} = \text{argmax}_{\{n \in 1,...,1+N_{t-1}^{[m]}\}} p_n$ ▷ ML correspondence
  $N_t^{[m]} = max\{N_{t-1}^{[m]}, \hat{n}\}$ ▷ new number of features
  if $\hat{n}_t = 1 + N_{t-1}^{[m]}$ **then** ▷ is new landmark?
    $\mu_{\hat{n}_t,t}^{[m]} = s_t^{[m]}$ ▷ initialize mean
    $\Sigma_{\hat{n}_t,t}^{[m]} = R_t$ ▷ initialize covariance
  else
    $K_t^{[m]} = \Sigma_{\hat{n}_t,t-1}^{[m]} G_{\theta,\hat{n}_t}^T Q_{\hat{n}_t,t}^{[m]-1}$ ▷ calculate Kalman gain
    $\mu_{\hat{n}_t,t}^{[m]} = \mu_{\hat{n}_t,t-1}^{[m]} - K_t^{[m]} \hat{z}_{\hat{n}_t,t}^T$ ▷ update mean
    $\Sigma_{\hat{n}_t,t}^{[m]} = (I - K_t^{[m]} G_{\theta,\hat{n}_t}) \Sigma_{\hat{n}_t,t-1}^{[m]}$ ▷ update covariance
  end if
  $w_t^{[m]} = w_{t-1}^{[m]} . p_{\hat{n}_t}^{[m]}$ ▷ importance weight
end for
loop M times
  draw random index m with probability $\alpha\ w^{[m]}$ ▷ resample
  add $\langle s_t^{[m]}, N_t^{[m]}, w_t^{[m]}, \langle f_{1,t}^{[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]} \rangle, ..., \langle f_{N,t}^{[m]}, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \rangle \rangle$ to S_t
end loop

6 Performance Evaluation

In this section, we present the performance evaluation of *SemanticSLAM*. *SemanticSLAM* can work in two modes of operation: offline and online. During the online phase, the user location is reported in each estimate. The offline mode is useful in applications that can tolerate delays, such as indoor user analytics. In this case, the entire path is determined based on the best particle at the end of the user movement trace instead of taking a local decision at every instant to select the best particle.

We start by describing our testbeds followed by the landmark and user location detection accuracy. We end the section by comparing the accuracy of the offline and online modes of operation as well as quantifying the advantages of using the proposed SLAM framework in terms of accuracy and convergence time compared to other systems.

6.1 Experimental Testbed

SemanticSLAM is implemented on different Android phones. The inertial sensors as well as WiFi information are sampled and sent to the server for processing.

We evaluated our system in two different testbeds: the Engineering Building in Duke university and a

Shopping Mall in Alexandria, Egypt. The **Engineering Building** area is $3000m^2$ and the breakup of its landmarks is: 9 magnetic, 8 turns, and 15 WiFi OLMs and 3 SLMs, as shown in Figure 8. The **Shopping Mall** area is $6000m^2$ and the breakup of its landmarks is: 9 magnetic, 12 turns, and 12 WiFi OLMs as well as 4 SLMs. Due to space constraints, we only present the results of the Engineering Building testbed and refer the reader to Appendix A for the detailed results of the Mall testbed.

Three different users move in the two testbeds to collect the data. Each user walked around arbitrarily in the building for 1.5 hours where each of them uses the landmarks detected by the previous user(s). The default setting of the system is the online mode.

6.2 Landmark Type Detection Accuracy

In this section, we evaluate *SemanticSLAM* ability to detect the landmarks type accurately.

6.2.1 Seed landmarks

Table 2 shows the confusion matrix for the detection of different seed landmarks. The matrix shows that some SLMs are easier to detect than others due to their unique patterns. Overall, *SemanticSLAM* can achieve 0.2% false positive and 1.1% false negative rates.

6.2.2 Organic landmarks

It is important that landmark signatures are stable over time and unique over distance. To evaluate these two points, we collected sensor readings on multiple days. We found sound consistency in the signatures. This is expected due to our design, where we use a low "similarity threshold" to detect an organic landmark as discussed in Section 4. Therefore, missing a few will affect performance much less than matching to an incorrect landmark.

6.3 Localization Accuracy

In this section, we evaluate the effect of the number of particles and landmark density on accuracy as well as the overall user localization accuracy.

6.3.1 Effect of the number of particles

Figure 9 shows the effect of the number of particles used for pose estimation on the overall estimation accuracy. The figure shows that the performance saturates around 50 particles achieving a median accuracy of about 0.53m. Therefore, we use 50 particles for the rest of this section.

6.3.2 Seed landmarks effect

Figure 10 shows the CDF of localization error for the Engineering building testbed with and without using the seed landmarks. The case of not using seed landmarks reflects running *SemanticSLAM* with no prior information at all. The figure shows that, even without the seed landmarks, *SemanticSLAM* can achieve high accuracy. Removing the seed landmarks reduces the localization accuracy due to the reduction of the overall number of landmarks. The Engineering Building has a higher accuracy compared to the Shopping Mall (Appendix A) due to the higher spatial density of landmarks in the Engineering Building. We note, however, that other common seed landmarks (e.g. turns, doors, and windows) can also be used in single floor buildings to compensate for the non-existence of "vertical transport landmarks".

6.3.3 User location accuracy

To evaluate the user localization accuracy, we calculated the Euclidean distance error at each step during the simulation. Figure 11 shows the pattern of the error in the Engineering Building. From the figure, we notice a saw-tooth pattern for the error, where the error range decreases with time. The justification for this pattern is that the error increases while the user is walking because of the noise in the sensors' readings. Whenever a landmark is observed, the error is reduced. As time passes, more landmarks are observed. Therefore, the map accuracy increases and therefore the user location error decreases.

6.4 Offline vs Online localization

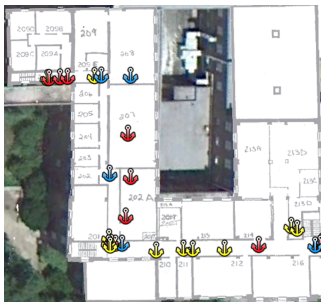
Figure 12 shows the CDF of the offline and online *SemanticSLAM* in the Engineering Building. The figure shows that the offline *SemanticSLAM* can significantly reduce the tail of the distribution compared to the online *SemanticSLAM* with a slight enhancement in median distance error.

6.5 Advantage of using a SLAM Framework

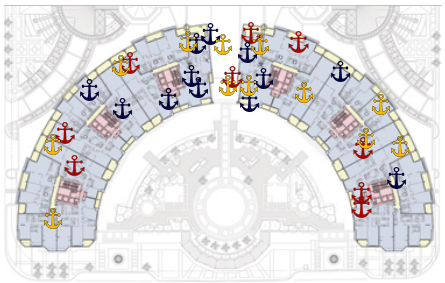
In this section, we quantify the advantage of the proposed semantic SLAM framework. For that, we compare *SemanticSLAM* with a previous version (our Unloc system [1]) that does not use the SLAM framework. Specifically, Unloc does not take into account the uncertainty of the landmark or the user's location which is considered in *SemanticSLAM*.

6.5.1 Convergence time

To measure the constructed map accuracy, at every time instance t we calculate the average location accuracy of all observed landmarks positions at the estimated map. The user movement was emulated to extend the simulation time to reach convergence by repeating the user's path trace in the simulation. Figure 13 shows the results for the Engineering Building. The figure shows that the *SemanticSLAM* accuracy converges quickly within 80 minutes and saturates at about an average accuracy of 0.83m. This is better than the Unloc system [1] that converges after 2 hours to around $1m$ average error. Note that even



(a) Testbed 1: An engineering building in a university campus.



(b) Testbed 2: A shopping mall in Alexandria, Egypt.

Fig. 8: Testbeds used in the evaluation. The blue, red and yellow anchors are the inertial, magnetic, and WiFi landmarks respectively.

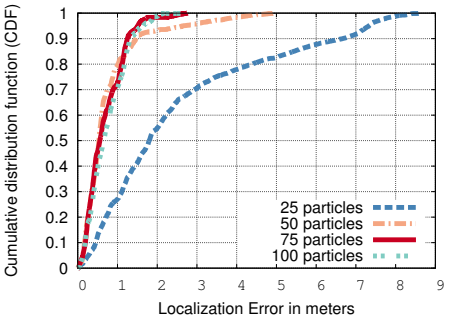


Fig. 9: CDF of user loc. accuracy for different number of particles.

	Elevator	Stationary	Escalator	Walking	Stairs	FP	FN	Traces
Elevator	24	0	0	0	0	0%	0%	24
Stationary	0	31	1	0	0	0%	3.1%	32
Escalator	0	0	22	0	0	0.6%	0%	22
Walking	0	0	0	39	0	0%	0%	39
Stairs	0	0	0	1	52	0%	1.8%	53
Overall						0.2%	1.1%	170

TABLE 2: Confusion matrix for classifying different seed landmarks.

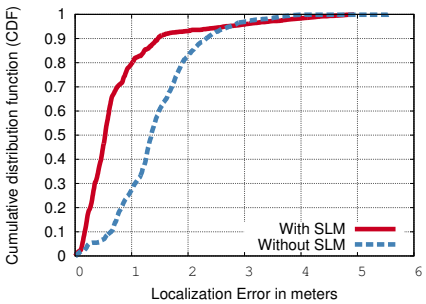


Fig. 10: CDF of loc. accuracy with and without using seed landmarks.

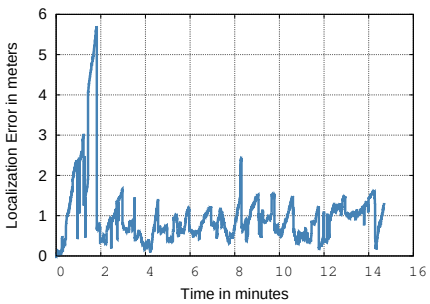


Fig. 11: User localization error over time.

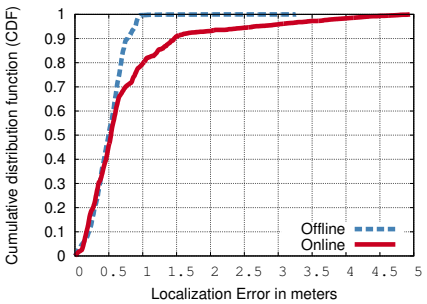


Fig. 12: CDF of loc. accuracy for online and offline *SemanticSLAM*.

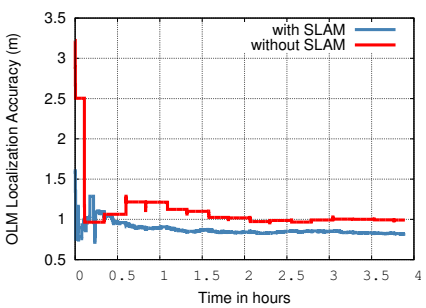


Fig. 13: Organic landmarks (OLM) loc. accuracy improvement with time.

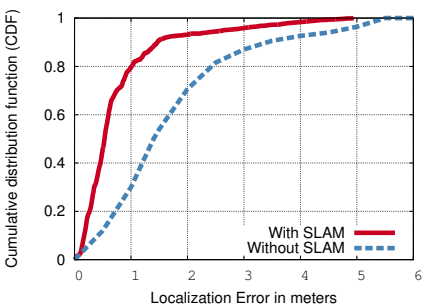


Fig. 14: Advantage of using the proposed SLAM framework on loc. accuracy.

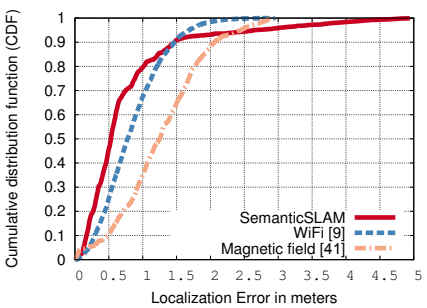


Fig. 15: Comparison between *SemanticSLAM* and other fingerprinting systems.

though *SemanticSLAM* requires some time to learn the organic landmarks, the effort to learn these landmarks is transparent to the user (collected through crowdsourcing) as compared to traditional fingerprinting techniques where the user has to perform the calibration manually.

6.5.2 Accuracy

In Figure 14, we compare our system accuracy with and without SLAM in the Engineering Building. The figure shows that the proposed SLAM framework in this paper can significantly enhance accuracy, achieving 62% enhancement in median distance error.

6.6 Comparison with Other Systems

In this section, we compare *SemanticSLAM* with two other systems: a WiFi fingerprinting system (similar to the Horus system [9]) and a magnetic fingerprinting system (similar to the MaLoc system [42]). The magnetic based system uses a particle filter to further improve the accuracy, which is not used in the WiFi-based system. Figure 15 shows that *SemanticSLAM* has better median accuracy as it uses more landmarks, both seed and organic. Fingerprinting-based systems, however, has a better worst-case error due to the initial manual calibration.

7 Related Work

Our work is related to prior work in both indoor localization systems and SLAM systems.

7.1 Indoor localization

Many systems have been proposed over the years to address the indoor localization problem [3], [5], [8], [9], [18], [43]–[46]. From these RF-based systems, especially those based on WiFi, have gained attention due to their ubiquitous deployment. Typically, these techniques build an RF fingerprint [8], [9] of the area of interest to compensate for the noisy wireless environment. However, building this fingerprinting is a tedious and time consuming process that needs to be repeated from time to time due to environment changes.

More recently, a number of systems have been proposed to reduce the calibration and re-calibration overhead of WiFi-based localization systems, e.g. [16], [18], [19]. These systems either depend on installing special hardware to monitor changes in the signal strength [16], crowd-sourcing [47] with active user participation, and/or propagation modelling tools [15], [48]–[50].

Dead-reckoning using smart phones inertial sensors has been also used in indoor localization [51]–[54]. However, dead-reckoning error quickly accumulates leading to complete deviation from the actual path. Therefore, GPS has been used in outdoor localization systems to recalibrate the dead-reckoned location [20], [21]. Since GPS is unreliable indoors, other techniques were used to improve the dead-reckoning errors [52], [54] such as constraining the resulting traces with the indoor floorplan layout.

SemanticSLAM, on the other hand, is unique in leveraging environment hints (i.e. landmarks) for resetting the error in dead-reckoning. These multi-sensors landmarks are learned in an organic way, with no calibration or active involvement from the user. Moreover, by combining the SLAM framework with landmarks as observations, *SemanticSLAM* can achieve high accuracy with guaranteed system convergence. In addition, since data is collected all the time through the system users, any changes in the organic landmarks locations or signatures can be captured in realtime, keeping the system up-to-date.

7.2 SLAM

The SLAM algorithm was originally introduced in robotics for navigation of autonomous agents in unknown environments. It has been applied recently in human tracking systems. *WiFi SLAM* [55] builds a landmark map of the WiFi signal strength based on a Gaussian Process Latent Variable Model (GP-LVM). GP-LVMs provide a framework for jointly modeling concurrent constraints on WiFi signal strength measurements (observation model) and a person's motion from the phone's inertial sensors (motion model).

Similarly, *ActionSLAM* [56] combines dead-reckoning based on special foot-mounted inertial sensors with observations of location-related actions.

SemanticSLAM generalizes the SLAM concept to work with landmarks that can be sensed by *any* of the phone sensors. In addition, it introduces the concepts of seed and organic landmarks. Moreover, it is based on leveraging the standard noisy cell phone sensors, without using any external hardware.

8 Conclusion

In this paper, we proposed *SemanticSLAM* as a calibration-free indoor localization system that provides a novel SLAM algorithmic framework. We provided the architecture and details of *SemanticSLAM* and how it combines dead-reckoning with semantic information discovered from multiple cell phone sensors about nearby landmarks to perform accurate localization and mapping simultaneously.

The system was evaluated on two testbeds: a university building and a shopping mall. Experimental results showed that *SemanticSLAM* can discover different multi-model landmarks with a low false positive and negative rate of less than 1%. In addition, it can achieve 0.53 meters median localization error in both testbeds with fast convergence time. This is even enhanced in the offline mode of operation. Compared to the state-of-the-art indoor localization systems, *SemanticSLAM* provides 62% enhancement in accuracy and 33% in convergence time, highlighting its promise for next generation indoor location-based services.

Currently, we are expanding the system in multiple directions including collaborative localization of multiple persons, exploiting other sensors (such as sound and light sensors), among others.

Acknowledgment

This work is supported in part by a Google Research Award to E-JUST and a grant from the Egyptian Information Technology Industry Development Agency (ITAC).

REFERENCES

[1] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 197–210.

[2] R. Want, A. Hopper, V. Falco, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, January 1992.

[3] R. Azuma, "Tracking requirements for augmented reality," *Communications of the ACM*, vol. 36, no. 7, July 1997.

[4] J. Krumm *et al.*, "Multi-camera multi-person tracking for Easy Living," in *3rd IEEE Int'l Workshop on Visual Surveillance*, Piscataway, NJ, 2000, pp. 3–10.

[5] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *6th ACM MOBICOM*, Boston, MA, August 2000.

[6] R. Fontana and S. Gunderson, "Ultra-wideband precision asset location system," in *Ultra Wideband Systems and Technologies, 2002. Digest of Papers. 2002 IEEE Conference on*, May 2002, pp. 147–150.

[7] D. Niculescu and B. Nath, "Vor base stations for indoor 802.11 positioning," in *MOBICOM*, 2004, pp. 58–69.

[8] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.

[9] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 205–218.

[10] M. Youssef, M. Abdallah, and A. Agrawala, "Multivariate analysis for probabilistic wlan location determination systems," in *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*. IEEE, 2005.

[11] M. Youssef and A. Agrawala, "Location-clustering techniques for WLAN location determination systems," *International Journal of Computers and Applications*, vol. 28, no. 3, 2006.

[12] A. E. Kosba, A. Abdelkader, and M. Youssef, "Analysis of a device-free passive tracking system in typical wireless environments," in *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, 2009.

[13] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 7, 2013.

[14] M. Seifeldin and M. Youssef, "A deterministic large-scale device-free passive localization system for wireless environments," in *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2010.

[15] K. El-Kafrawy, M. Youssef, A. El-Keyi, and A. Naguib, "Propagation modeling for accurate indoor wlan rss-based localization," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*. IEEE, 2010, pp. 1–5.

[16] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Gamt, "A system for LEASE: Location estimation assisted by stationary emitters for indoor rf wireless networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2004, pp. 1001–1011.

[17] M. Youssef, A. M. Youssef, C. Rieger, A. U. Shankar, and A. K. Agrawala, "Pinpoint: An asynchronous time-based location determination system," in *MobiSys*, 2006, pp. 165–176.

[18] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale wi-fi localization," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 233–245.

[19] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 173–184.

[20] M. Youssef, M. A. Yosef, and M. El-Derini, "Gac: energy-efficient hybrid gps-accelerometer-compass gsm localization," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–5.

[21] I. Constandache, R. R. Choudhury, and I. Rhee, "Towards mobile phone localization without war-driving," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[22] M. Alzantot and M. Youssef, "Crowdinside: Automatic construction of indoor floorplans," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '12. ACM, pp. 99–108.

[23] M. Elhamshary and M. Youssef, "CheckInside: A fine-grained indoor location-based social network," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14, 2014.

[24] ——, "SemSense: Automatic construction of semantic indoor floorplans," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, 2015.

[25] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics and Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[26] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *AAAI/IAAI*, 2002, pp. 593–598.

[27] D. Roller, M. Montemerlo, S. Thrun, and B. Wegbreit, "FastSALM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2003.

[28] P. Moutarlier and R. Chatila, "An experimental system for incremental environment modelling by an autonomous mobile robot," in *Experimental Robotics I*. Springer, 1990, pp. 327–346.

[29] ——, "Stochastic multisensory data fusion for mobile robot location and environment modeling," in *5th Int. Symposium on Robotics Research*, vol. 1. Tokyo, 1989.

[30] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 229–241, 2001.

[31] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," Ph.D. dissertation, University of Sydney, 2002.

[32] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 6, pp. 890–897, 2001.

[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[34] H. Shatkay and L. P. Kaelbling, "Learning topological maps with weak local odometric information," in *IJCAI (2)*, 1997, pp. 920–929.

[35] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Autonomous Robots*, vol. 5, no. 3-4, pp. 253–271, 1998.

[36] A. Araneda and A. Soto, "Statistical inference in mapping and localization for mobile robots," in *Advances in Artificial Intelligence–IBERAMIA 2004*. Springer, 2004, pp. 545–554.

[37] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer, 2001.

[38] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American statistical association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[39] M. Alzantot and M. Youssef, "UPTIME: Ubiquitous pedestrian tracking using mobile phones," in *WCNC*, 2012, pp. 3204–3209.

[40] N. Mohssen, R. Momtaz, H. Aly, and M. Youssef, "It's the human that matters: Accurate user orientation estimation for mobile computing applications," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, ser. MOBIQUITOUS '14, 2014.

[41] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[42] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lv, "Maloc: a practical magnetic fingerprinting approach to indoor localization using smartphones," in *Proceedings of the 2014 ACM International Joint*

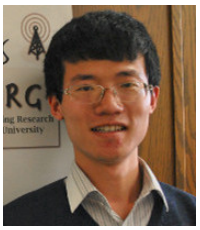
Conference on Pervasive and Ubiquitous Computing. ACM, 2014, pp. 243–253.
[43] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.
[44] A. Ward, A. Jones, and A. Hopper, "A New Location Technique for the Active Office," *IEEE Personal Communications*, vol. 4, no. 5, pp. 42–47, October 1997.
[45] M. Youssef, "Towards truly ubiquitous indoor localization on a worldwide scale," in *ACM SIGSpatial*, 2015.
[46] R. Elbakly and M. Youssef, "A calibration-free RF localization system," in *ACM SIGSpatial*, 2015.
[47] 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 *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '10. ACM, pp. 271–284.
[48] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "Ariadne: a dynamic indoor signal map construction and localization system," in *MobiSys*, 2006, pp. 151–164.
[49] A. Eleryan, M. Elsabagh, and M. Youssef, "Synthetic generation of radio maps for device-free passive localization," in *GLOBECOM*, 2011, pp. 1–5.
[50] H. Aly and M. Youssef, "New insights into wifi-based device-free localization," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, 2013, pp. 541–548.
[51] D. Gusenbauer, C. Isert, and J. Krosche, "Self-contained indoor positioning on off-the-shelf mobile devices," in *Indoor positioning and indoor navigation (IPIN), 2010 international conference on*. IEEE, 2010, pp. 1–9.
[52] J. A. B. Link, P. Smith, N. Viol, and K. Wehrle, "Footpath: Accurate map-based indoor navigation using smartphones," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*. IEEE, 2011, pp. 1–8.
[53] 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 *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 421–430.
[54] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 293–304.
[55] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM using Gaussian process latent variable models." in *IJCAI*, vol. 7, 2007, pp. 2480–2485.
[56] M. Hardegger, D. Roggen, S. Mazilu, and G. Troster, "ActionSLAM: Using location-related actions as landmarks in pedestrian SLAM," in *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*. IEEE, 2012, pp. 1–10.

Heba Abdelnasser is with the Wireless Research Center at E-JUST, Egypt. She received her B.Sc. in Computer and Systems Engineering from Alexandria University in 2013, and is currently preparing her M.Sc. in Computer Science in the same university. Her research interests include mobile computing, location determination systems, and NLP.

Reham Mohamed is with the Wireless Research Center at E-JUST, Egypt. She received her B.Sc. in Computer and Systems Engineering from Alexandria University in 2013, and is currently preparing her M.Sc. in Computer Science in the same university. Her research interests include mobile computing, location determination systems, and machine learning.

Ahmed Elgohary received a B.Sc. degree in Computer Engineering at Alexandria University, Egypt in 2010 and an M.A.Sc. degree in Electrical and Computer Engineering at the University of Waterloo, Canada in 2014. Ahmed is the recipient of the Best Paper Award Runner-Up, the 2013 IEEE Int. Conf. on Data Mining. He is currently pursuing a Ph.D. in Computer Science at the University of Maryland. Ahmed is broadly interested in data mining and machine learning.

Moustafa Alzantot is a PhD student in computer science at University of California, Los Angeles. His research interests include mobile sensing systems, context-aware systems, and privacy. Moustafa received his M.Sc. in computer Engineering from Egypt-Japan University of Science and Technology, and B.Sc. from Tanta University. He is the recipient of the 2013 COMESA innovation award. Contact him at malzantot@ucla.edu.



He Wang received his B.E. degree in Electrical Engineering from Tsinghua University in 2011 and M.S. degree in Electrical and Computer Engineering from Duke University in 2013. Since 2013, he has been pursuing his Ph.D. degree in Electrical and Computer Engineering at University of Illinois at Urbana-Champaign. He is interested in mobile computing. Currently, his research focuses on designing mobile systems involving various sensors.

Souvik Sen is a Principal Research Scientist in the Networking and Mobility group of HP Labs where he currently leads HPs initiatives on indoor location. His research on indoor location, wireless networking and mobile systems has been published at top conferences such as MobiSys, MobiCom and NSDI. He has won HP's Leading the Way Award in 2013 and 2014. He earned his Ph.D. in Computer Science from Duke University. He is the winner of ACM Student Research Competition in 2010 and Duke University Outstanding Ph.D. Preliminary Exam Award in 2011.

Romit Roy Choudhury is an Associate Professor of ECE at the University of Illinois at Urbana Champaign (UIUC). He joined UIUC from Fall 2013, prior to which he was an Associate Professor at Duke University. Romit received his PhD in the CS department of UIUC in Fall 2006. His research interests are in wireless protocol design mainly at the PHY/MAC layer, and in mobile computing at the application layer. Along with his students, he received a few research awards, including the 2015 ACM Sigmobile Rockstar Award, the NSF CAREER Award, the Google Faculty Award, Hoffmann Krippner Award for Engineering Innovations, best paper at Personal Wireless Communications conference, etc. Visit Romit's Systems Networking Research Group (SyNRG), at http://synrg.csl.illinois.edu

Moustafa Youssef is an Associate Professor at Alexandria University and Egypt-Japan University of Science and Technology (E-JUST), Egypt. He received his Ph.D. degree in computer science from University of Maryland, USA in 2004 and a B.Sc. and M.Sc. in computer science from Alexandria University, Egypt in 1997 and 1999 respectively. His research interests include mobile wireless networks, mobile computing, location determination technologies, pervasive computing, and network security. He has fifteen issued and pending patents. He is an associate editor for the ACM TSAS, a previous area editor of the ACM MC2R, and served on the organizing and technical committees of numerous prestigious conferences. Dr. Youssef is the recipient of the 2003 University of Maryland Invention of the Year award, the 2010 TWAS-AAS-Microsoft Award for Young Scientists, the 2012 Egyptian State Award, the 2013 and 2014 COMESA Innovation Award, multiple Google Research Awards, the 2013 ACM SIGSpatial GIS Conference Best Paper Award, among others. He is also an ACM Distinguished Speaker.