

Article

MapSentinel: Can the Knowledge of Space Use Improve Indoor Tracking Further?

Ruoxi Jia ^{1,*}, Ming Jin ¹, Han Zou ², Yigitcan Yesilata ³, Lihua Xie ² and Costas Spanos ¹

¹ Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA; jinming@berkeley.edu (M.J.); spanos@berkeley.edu (C.S.)

² School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798, Singapore; zouhan@ntu.edu.sg (H.Z.); elhxie@ntu.edu.sg (L.X.)

³ Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara 06800, Turkey; yigitcan.yesilata@metu.edu.tr

* Correspondence: ruoxijia@berkeley.edu; Tel.: +1-510-693-7634

Academic Editor: Vittorio M. N. Passaro

Received: 25 January 2016; Accepted: 26 March 2016; Published: 2 April 2016

Abstract: Estimating an occupant's location is arguably the most fundamental sensing task in smart buildings. The applications for fine-grained, responsive building operations require the location sensing systems to provide location estimates in real time, also known as indoor tracking. Existing indoor tracking systems require occupants to carry specialized devices or install programs on their smartphone to collect inertial sensing data. In this paper, we propose MapSentinel, which performs non-intrusive location sensing based on WiFi access points and ultrasonic sensors. MapSentinel combines the noisy sensor readings with the floormap information to estimate locations. One key observation supporting our work is that occupants exhibit distinctive motion characteristics at different locations on the floormap, e.g., constrained motion along the corridor or in the cubicle zones, and free movement in the open space. While extensive research has been performed on using a floormap as a tool to obtain correct walking trajectories without wall-crossings, there have been few attempts to incorporate the knowledge of space use available from the floormap into the location estimation. This paper argues that the knowledge of space use as an additional information source presents new opportunities for indoor tracking. The fusion of heterogeneous information is theoretically formulated within the Factor Graph framework, and the Context-Augmented Particle Filtering algorithm is developed to efficiently solve real-time walking trajectories. Our evaluation in a large office space shows that the MapSentinel can achieve accuracy improvement of 31.3% compared with the purely WiFi-based tracking system.

Keywords: indoor tracking systems; non-intrusive; map-aided; WiFi; ultrasonic sensor networks; particle filters

1. Introduction

The indoor location sensing technology has emerged as an inherent part of the “smart buildings” as it provides great potential for building operation improvement and energy saving. For instance, an on-demand ventilation or lighting control policy must know the usage of the building spaces, which may involve when building occupants enter or exit the building, where they inhabit, what time they occupy the spaces, the duration of occupancy, *etc.* Such applications require the location sensing systems to provide real-time estimate of occupants' locations, which is also termed “indoor tracking”, in order to realize fine-grained, responsive building operations.

Most indoor tracking systems necessitate each occupant to carry or wear a powered device such as an infrared [1], ultrasonic [2–4], or Radio Frequency transceiver [5–7]. Even if the transceiver

is miniaturized into a convenient form, occupants are not willing or likely to carry it at all times. Another subset of tracking systems alleviate the need for carrying specialized devices by using the inertial sensors on smartphones to perform dead reckoning [8–10]. However, specialized programs are required to be installed on smartphones to continuously collect inertial sensing data, and thereby the associated energy issues or occupants' engagement become the main impediment.

On the contrary, we enable non-intrusive indoor tracking by developing an information fusion system that takes advantage of noisy measurements from various sensors, namely, WiFi access points and ultrasonic sensors. WiFi access points are beneficial for wide spatial coverage while WiFi signals transmitted in the indoor environments suffer from large variations [11]; ultrasonic sensors are able to accurately locate the occupants in their detection zones which are nevertheless limited spatially. Our vision is of occupants carrying some device with WiFi module, which can be smartphones, tablets, wearable devices, *etc.*, in the indoor space where ultrasonic sensors can provide opportunistic calibration of the location estimation. The location sensing system is operating in a passive way, *i.e.*, there is no need for specialized devices or programs for location inference.

In addition to the sensor measurements, another key input for our system is the floormap of the indoor space of interest. Floormap information has been used to refine walking trajectory estimates by eliminating wall-crossings or unfeasible locations [12–14]. There has also been efforts to use the floormap to reduce the complexity of the tracking task by properly quantizing the indoor space [15–18]. In effect, we can also acquire some prior knowledge of occupants' dynamic motion from the floormap. The indoor space comprises several typical components, such as cubicles, offices, corridors, open areas, *etc.*, where occupants' motion exhibit distinctive patterns. For example, when located at his/her office or cubicle, the occupant is very likely to keep static; the occupant walking on a particular corridor tends to continue the motion constrained along the corridor, while an occupant in an open space is free to move in any direction. Such information of space use is useful to track occupants' movement, notwithstanding it is less considered in previous work. Gusenbauer *et al.* [19] exploited different types of movements to improve the tracking model. This was done by introducing an activity recognition algorithm based on accelerometer data to model pedestrians' steps more reliably. Park [20] proposed incorporating the floormap information by "path compatibility", where occupants' motion sequences and motion-related information (e.g., duration and speed) are first estimated based on mobile sensing data, and then localization is achieved via matching occupants' motion sequences and the hypothetical trajectories provided by the floormap. Kaiser *et al.* [21] proposed a motion model based on the floormap, which weights the possible headings of the pedestrian as a function of the local environment. Our work differs from [19] and [20] in that our work does not rely on the inertial measurements to recognize the motion. Instead, the motion information is extracted from the floormap. We exploit the prior knowledge that the floormap endows us about the occupants' typical movement and activity, not merely the possible headings at each point of the floormap as in [21]. It is, therefore, the objective of this paper to propose MapSentinel, a non-intrusive location sensing system via information fusion, which combines the various sensor measurements with the floormap information, not only as a sanity check of estimating trajectories but as an input for occupants' kinematic models.

Our main contributions are as follows:

- We build a non-intrusive location sensing network consisting of modified WiFi access points and ultrasonic calibration stations, which does not require the occupants to install any specialized programs on their smartphones and prevents the energy and occupant engagement issues.
- We propose an information fusion framework for indoor tracking, which theoretically formalizes the fusion of the floormap information and the noisy sensor data using Factor Graph. The Context-Augmented Particle Filtering algorithm is developed to efficiently solve the walking trajectories in real time. The fusion framework can flexibly graft floormap information onto other types of tracking systems, not limited to the WiFi tracking schemes that we will demonstrate in this paper.

- We evaluate our system in a large typical office environment, and our tracking system can achieve significant tracking accuracy improvement over the purely WiFi-based tracking systems.

The rest of this paper expands on each of these contributions. We conclude the paper and discuss the future work in Section 6.

2. MapSentinel Architecture

Figure 1 presents the overall architecture of MapSentinel. There are three key components in MapSentinel: the non-intrusive sensing networks, the floormap processing engine, and the information fusion algorithm. The non-intrusive sensing networks, as the name suggests, generate location-related measurements without the need for computation on the smartphone end. Our sensing networks consist of WiFi access points (APs) and ultrasonic calibration stations, which track locations by relating the WiFi signal strength or the sound time-of-flight to the distance. The floormap processing engine converts the pictorial floormap to the information that can be directly combined with the sensor measurements in the fusion algorithm. The output of the floormap processing engine represents the prior knowledge obtained from the map, and can be computed in the offline phase. We will present the details of the main components of MapSentinel in this section.

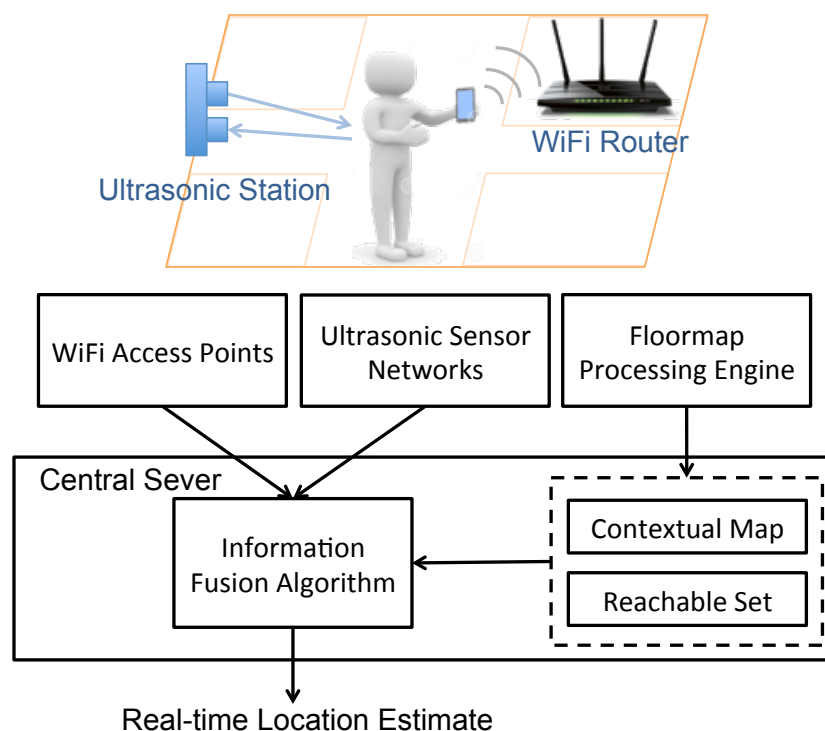


Figure 1. MapSentinel architecture—WiFi APs keep tracking occupants' locations, and the estimation is periodically refined using the ultrasonic stations deployed in the environment. Furthermore, the sensor measurements and the floormap information are combined via the information fusion algorithm to estimate location in real-time. The floormap processing engine helps transform the floormap to the information accessible to the fusion algorithm.

2.1. WiFi Access Points

IEEE 802.11 (WiFi) is the most commonly used wireless networking technology with widely available infrastructure in large numbers of commercial and residential buildings. Nearly every existing commercial mobile device is WiFi enabled. The common method to utilize WiFi for indoor location sensing is to enable the mobile device to collect WiFi Received Signal Strengths (RSS) of nearby WiFi APs by installing an application on the mobile devices. Our system, on the contrary, leverages

WiFi in a non-intrusive manner. Rather than modifying the hardware or software of occupants' mobile devices, we upgrade the software of the existing commercial WiFi APs to allow them to detect the RSS of each mobile device, while providing basic internet service to occupants as well. The RSS and media access control (MAC) address of each mobile device will be forwarded to the server and the occupant can be identified through the unique MAC address of the mobile device.

2.2. Ultrasonic Calibration Stations

Ultrasonic sensors measure the distance to the obstacle in the front to accurately position the object in its detecting range, which works by detecting the time of return, t , and the distance is given by:

$$d = \frac{v_{\text{sound}} \times t}{2} \quad (1)$$

where $v_{\text{sound}} \approx 340$ m/s is the velocity of sound in the air. The advantages include centimeter-resolution distance measurements and limited span of detection angles, which make it suitable for online calibration of indoor positioning systems. Figure 2 demonstrates typical traces of the ultrasonic sensor readings when the occupant moves across the detection zones. By properly thresholding the distance measurements, the ultrasonic sensor can be used as an indicator of occupant presence inside its detection zone.

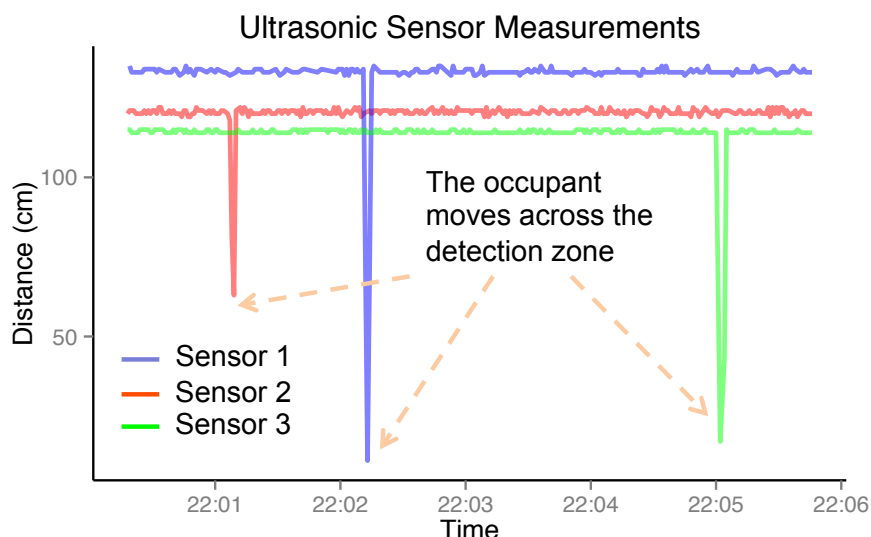


Figure 2. The measurements of ultrasonic stations deployed in the space. When the occupant is within the detection zone of the ultrasonic station, the sensor reading exhibits a smaller value.

The network consists of deployed ultrasonic stations and data collection center, which communicate with XBee radio modules operating the IEEE 802.15.4 standard, more specifically, the ZigBee protocols, as shown in Figure 3. The radios are low-power and can operate reliably in the indoor space, where the network can be automatically established by the coordinator, in our case, the data collection center. The data center controlled by Arduino enquires about the ultrasonic station for measurements periodically, so that the measurement frequency is 1 Hz, and transfers the data to the computer connected by serial ports. Each ultrasonic station is equipped with three ultrasonic sensors, whose directions are offset by 15° . As the measurement range spans 15° for each ultrasound, this covers an area of 45° in the front of the station, which is sufficient for indoor area localization.

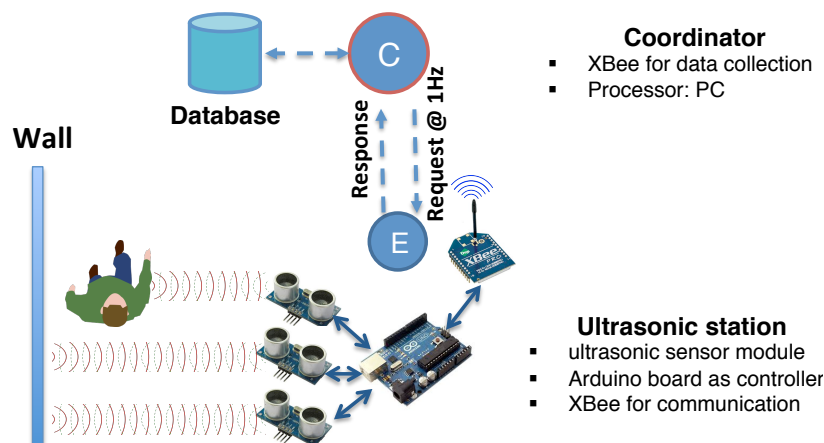


Figure 3. Illustration of the configuration of the ultrasonic calibration station. The coordinator requests measurements at 1 Hz frequency through the IEEE 802.15.4 protocol, and deposits collected data to the local database. The ultrasonic station takes three independent measures from its sensor points to detect occupant presence in the vicinity.

2.3. Floormap Processing Engine

The indoor space is well structured and typically organized into corridors, open areas, walls, rooms, *etc.* Depending on the occupant's present location, the motion is constrained by these external factors. For instance, an occupant on a particular corridor has high probability continuing its motion constrained along the corridor—or an occupant walking in the open area is free to move in any direction. Likewise, an occupant in his/her cubicle area is more likely to stay static. Based on different motion capabilities, we categorize the indoor space into several contexts, namely, open space, constrained space and static space. In addition, the floormap processing engine is designed to convert the original floormap into the *contextual floormap* that indicates the context of each point in the original floormap. The details of each component of contextual floormap is provided in Table 1. We use the word “canonical direction” to refer to the direction of constrained space along which the movement has more freedom.

Table 1. Components of contextual floormap.

Context	Symbols	Motion Characteristics
Free Space	FS	Move freely, e.g., rooms
Constrained Space	CS	Move along canonical direction, e.g., corridors
Static Space	SS	Stay static, e.g., cubicles

In addition, the occupant motion is also constrained by speed restrictions. Another function of the floormap processing engine is to compute the reachable set containing all the points visited with admissible speed from a given starting point. In the indoor space, the geographical distance between two positions in a floormap does not necessarily equal to the walking distance between them due to the block of walls and other obstacles. Hence, the physical features of the indoor environments would be ignored if the reachable set is confined within a fixed radius centered around the given starting point. The floormap processing engine addresses this problem by converting the floormap to a graph where all the non-barricade nodes connect to their neighboring non-barricade nodes and the barricade nodes do not have connections to any other nodes. In this way, the reachable set of a given node can be computed through finding the nodes within the maximum depth from the root node, which can be efficiently solved by breadth-first search algorithm [22].

3. Information Fusion Framework

In this section, we propose an information fusion framework that manages the heterogeneous sensor measurements as well as the floormap and occupants' context-related motion characteristics to provide an online estimate of occupants' location. There are two key components in the fusion framework: Context-Dependent Kinematic Models (CDKM) and Probabilistic Sensor Measurement Models (PSMM). CDKM is based on the observation that occupants' movements exhibit distinctive features in different parts of buildings as described in Section 2.3, and it captures this context-dependency by defining different kinematic models for distinctive contexts. PSMM models each sensor measurement as a probability distribution and multiple sensor data are combined via Bayes' rule to support the location inference.

3.1. Problem Formulation

Consider that the indoor space of interest is composed of M contexts, in each of which occupants exhibit a particular sort of kinematic patterns. Denote the context at time k as m_k where $m_k \in \{FS, CS_1, \dots, CS_R, SS\}$. The subscript of CS represents the index of the certain direction of constrained space and R is the total number of different directions. Let the state $\mathbf{x}_k = (\mathbf{z}_k, m_k)$ consist of the position and velocity components of the occupant in the Cartesian coordinates $\mathbf{z}_k = (x_k, y_k, \dot{x}_k, \dot{y}_k)$, as well as the context m_k . If the position is known, the context can be uniquely determined by the contextual floormap. We characterize this correspondence via a function $\mathcal{M} : \mathbb{R}^4 \rightarrow \mathbb{R}$ which assigns a specific context m_k for \mathbf{z}_k . The tracking problem can be viewed as a statistical filtering problem where \mathbf{z}_k is to be estimated based on a set of noisy measurements $y_{1:k} = \{y_1, \dots, y_k\}$ up to time k . Specifically, y_k is the measurements available at time k , and, in our case, it includes measurements from multiple sensors, $\{y_k^n\}_{n=1}^{N_s}$ where N_s is total number of sensors deployed in the space of interest. We model the uncertainty about the observations and the states by treating them as random variables and assigning certain probability distribution to each random variable. In this setting, we want to compute the posterior distribution of the state given the measurements up to time k , i.e., $p(\mathbf{z}_k | y_{1:k})$.

The impact of introducing context as an auxiliary state variable is manifold. Firstly, the transition of contexts m_{k-1} to m_k determines the type of motion executed during the time interval $(k-1, k]$. For instance, if the context remains the same, then the occupant should follow the motion type defined by the two identical contexts; on the contrary, if the context varies during $(k-1, k]$, then the occupant would execute the motion that is defined by neither of the contexts. For simplicity, we will assume a free motion. That is, the position/velocity state at time k , \mathbf{z}_k , depends on not only the past state \mathbf{z}_{k-1} and m_{k-1} , but also the current context m_k stochastically. Moreover, there is a deterministic mapping between \mathbf{z}_k and m_k as is specified by the contextual map. In order to facilitate visualization and analysis of the complex dependencies among the variables, we use a factor graph to represent the states, observations and the functions bridging these variables, as illustrated in Figure 4.

A factor graph has two types of nodes, *variable node* for each variable and *function node* for each local function, which are indicated by circles and squares, respectively. The edges in the graph represents the "is an argument of" relation between variables and local functions. For example, the function \mathcal{T}_k has four arguments, \mathbf{z}_k , \mathbf{z}_{k-1} , m_{k-1} and m_k . Three types of local functions are involved in our model:

- $\mathcal{T}_k(\mathbf{z}_k, \mathbf{z}_{k-1}, m_k, m_{k-1}) = p(\mathbf{z}_k | \mathbf{z}_{k-1}, m_k, m_{k-1})$: transition model, or the prior information on the state evolution over time. Inspired by Variable Structure Multiple Model Estimator in [23], we propose CDKM to capture the context-dependent characteristics of occupants' motion in the indoor space.
- $\mathcal{O}_k(\mathbf{z}_k, y_k) = p(y_k | \mathbf{z}_k)$: observation model, or how the unknown states and sensor observations relate. We will introduce PSMM where the relationship between locations and sensor observations is characterized by certain conditional probabilities and multiple sensor observations are combined via Bayes' theorem.

- $C_k(\mathbf{z}_k, m_k)$: characteristic function that checks the validity of the correspondence between \mathbf{z}_k and m_k using the contextual floormap.

Note that the prior knowledge abstracted from the floormap is inherently accommodated to this problem by defining characteristic function and parameterizing the transition model as will be elaborated in the following section.

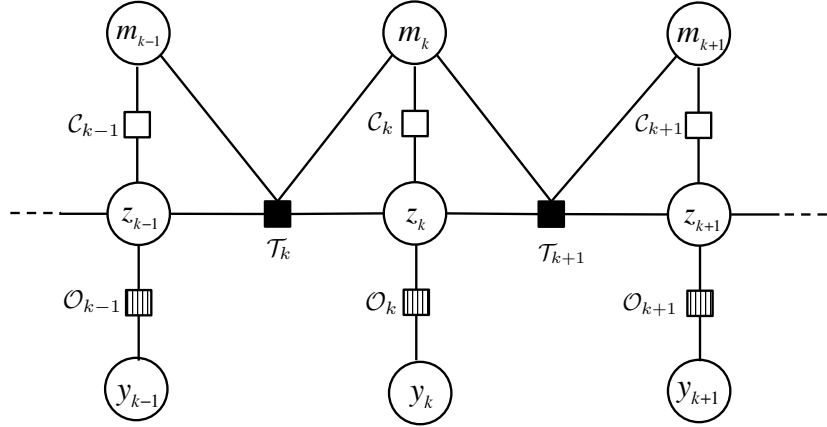


Figure 4. A factor graph model representation of the dependencies among location, velocity, context and observation.

3.2. Context-Dependent Kinematic Model

We assume that given \mathbf{z}_{k-1} , m_{k-1} and m_k , the current position/velocity \mathbf{z}_k follows a Gaussian distribution, of which the mean and covariance matrix are specified as

$$p(\mathbf{z}_k | \mathbf{z}_{k-1}, m_k, m_{k-1}) \sim \mathcal{N}(F(m_{k-1}, m_k)\mathbf{z}_{k-1}, GQ(m_{k-1}, m_k)G') \quad (2)$$

The equivalent state space model of Equation (2) is given by

$$\begin{aligned} \mathbf{z}_k &= F(m_{k-1}, m_k)\mathbf{z}_{k-1} + Gv(m_{k-1}, m_k) \\ v(m_{k-1}, m_k) &\sim \mathcal{N}(0, Q(m_{k-1}, m_k)) \end{aligned} \quad (3)$$

where $F(m_{k-1}, m_k) \in \mathbb{R}^{4 \times 4}$ determines the mean of the distribution of the next state. Let a denote the acceleration, we have the following kinematic equations,

$$x_k = x_{k-1} + \dot{x}_{k-1}T + \frac{1}{2}aT^2 \quad (4)$$

$$\dot{x}_k = \dot{x}_{k-1} + aT \quad (5)$$

where T is the sampling period. We will assume constant velocity in this paper, and model a as a Gaussian noise term. If we manipulate Equations (4) and (5) into matrix forms, then it can be identified that $F(m_{k-1}, m_k)$ has two possible values corresponding to moving or remaining static,

$$F_0 = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

F_1 imposes the velocity component of the state \mathbf{z}_k to be zero and $F = F_1$ when the context remains to be static space, i.e., $m_{k-1} = m_k = SS$; otherwise, $F = F_0$.

The matrix G is given by

$$G = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix} \quad (7)$$

$Q(m_{k-1}, m_k)$ stands for the process noise and, as the notation indicates, it is also a function of the context transition from $k-1$ to k . We will adopt the concept of directional noise to handle the constraints imposed by the contextual map. To see this, note that occupants in the free space ($m_{k-1} = m_k = FS$) can move in any direction with equal probability, therefore using equal process noise variance in both x and y direction, *i.e.*,

$$Q_0 = \begin{bmatrix} \sigma_f^2 & 0 \\ 0 & \sigma_f^2 \end{bmatrix} \quad (8)$$

For occupants moving on the constrained space ($m_{k-1} = m_k = CS_i, \forall i = 1, \dots, R$) such as corridors, more uncertainty exists along than orthogonal to the corridor. Denote the variances along and orthogonal to the corridor by σ_a^2 and σ_o^2 ($\sigma_a^2 > \sigma_o^2$), respectively, and the canonical direction of the constrained space CS_i is specified by the angle ϕ_i (measured clockwise from y -axis). Then the process noise covariance matrix corresponding to the motion in the constrained space is given by

$$Q_i = \begin{bmatrix} -\cos \phi_i & \sin \phi_i \\ \sin \phi_i & \cos \phi_i \end{bmatrix} \begin{bmatrix} \sigma_o^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} \begin{bmatrix} -\cos \phi_i & \sin \phi_i \\ \sin \phi_i & \cos \phi_i \end{bmatrix} \quad (9)$$

The preceding model specification incorporates the scenarios where the context remains the same during the time interval $[k-1, k]$ and the occupant will keep the motion type defined by the two identical contexts. On the contrary, if the context switches during the time interval $[k-1, k]$, we will assume a free motion pattern, *i.e.*, $F = F_0$, $Q = Q_0$. Table 2 summarizes our model given all possible context transitions.

Table 2. Context-dependent kinematic models.

Context Transition	Model Specification	
	$F(m_{k-1}, m_k)$	$Q(m_{k-1}, m_k)$
$m_{k-1} = m_k = FS$	F_0	Q_0
$m_{k-1} = m_k = CS_i$	F_0	Q_i
$m_{k-1} = m_k = SS$	F_1	Q_0
$m_{k-1} \neq m_k$	F_0	Q_0

3.3. Probabilistic Sensor Measurement Model

We construct probabilistic models for each sensor and multisensor fusion can be performed via Bayes' rule. Assuming that N_s different sensors function independently, then the observation model $p(y_k | \mathbf{z}_k)$ can be factored as

$$p(y_k | \mathbf{z}_k) = \prod_{n=1}^{N_s} p(y_k^n | \mathbf{z}_k) \quad (10)$$

This actually forms a convenient and unified interface to combine distinctive sensor data by projecting the heterogeneous measurements (y^n) to the probability space via *likelihood function*, $p(y^n | \mathbf{z})$. If one more sensor is added into the system, then the observation model can be simply updated by

multiplying the corresponding likelihood. Different likelihood functions requires being trained for different types of sensors.

WiFi Measurement. In the free space, the WiFi signal strength is a log linear function of the distance between the transmitter and receiver. However, due to the multipath effect caused by obstacles and moving objects in the indoor environments, the log linear relationship no longer holds. Previous work has proposed to adding a Gaussian noise term to account for the variations arising from the multipath effect; however, the simple model-based method can hardly guarantee a reasonable performance in practice. Another popular way is to construct a WiFi database comprising WiFi measurements at known locations to fingerprint the space of interest, but it requires onerous calibration to ensure the accuracy. We propose a novel WiFi modeling method based on a relatively small WiFi training set to accommodate for the complex variations of WiFi signals in the indoor space. The key insight is to use Gaussian process (GP) to model the WiFi signal where the simple model-based method provides a prior over the function space of GP.

We collect WiFi signal strength data at N_w reference points over the space and let $\{l^j, y_w^j\}_{j=1}^{N_w}$ denote the training dataset, where l^j is a vector containing the distances of j th reference point to each of the WiFi APs deployed in the field and y_w^j is the observed WiFi signal strengths. Assume the WiFi observations are drawn from the GP,

$$y_w \sim \mathcal{GP}(\mu(l), k(l, l')) \quad (11)$$

where the mean function $\mu(\cdot)$ is imposed to be a linear model with the parameters adapted to the training samples. The covariance function $k(\cdot, \cdot)$ takes the squared exponential form,

$$k(l, l') = \sigma_f^2 \exp(-\frac{1}{2r^2}(l - l')^2) + \sigma_n^2 \quad (12)$$

where σ_n^2 stands for the variance of the additive Gaussian noise term in the observation process, and σ_f^2 and r are the hyperparameters of the GP. These parameters can be tweaked according to the training data, and we set $\sigma_n = 4$, $\sigma_f = 2$, $r = 5$ in our experiments. At an arbitrary point l_* in the space of interest, the posterior mean and variance of the WiFi signal y_* are

$$\bar{y}_* = \mu(l_*) + K(l_*, \mathbf{L})[K(\mathbf{L}, \mathbf{L}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y}_w \quad (13)$$

$$\text{cov}(y_*) = K(l_*, l_*) - K(l_*, \mathbf{L})[K(\mathbf{L}, \mathbf{L}) + \sigma_n^2 \mathbf{I}]^{-1} K(\mathbf{L}, l_*) \quad (14)$$

where \mathbf{L} and \mathbf{y}_w are the vectors concatenated by $\{l^j\}_{j=1}^{N_w}$ and $\{y_w^j\}_{j=1}^{N_w}$, respectively. $K(l_*, \mathbf{L})$ denotes the $1 \times N_w$ matrix of the covariances evaluated at all pairs of training and testing points, and similarly for the other entries $K(\mathbf{L}, \mathbf{L})$ and $K(\mathbf{L}, l_*)$. In previous work using GP to model the WiFi signal strength [24], the WiFi signal is assumed to follow the Gaussian distribution with the mean and variance given by Equations (13) and (14), respectively. However, the posterior variance derived from GP is a indicator of estimation confidence. It depends largely on the density of training samples in the vicinity of the evaluated position. That is, if the evaluated point l_* happens to fall into the area that is densely calibrated, then the posterior variance will be relatively small. The posterior variance derived from GP cannot truly reflect the variations of WiFi signals over time. Therefore, instead of using the posterior Variance (14) in classical predictive equations, we model the likelihood as

$$y_* \sim \mathcal{N}(\bar{y}_*, \sigma_n^2) \quad (15)$$

Ultrasonic Measurement. Essentially, each of the ultrasonic sensors in the ultrasonic station can output the distance to the occupant passing in front of it. However, due to the missing data and measurement noise, the distance measurement is not always steady. Here, we will consider the

ultrasonic station to be a binary sensor to indicate the occupancy in its detection zone. To be specific, the likelihood function is modeled as

$$p(y_k < \eta | \mathbf{z}_k \text{ in the detection zone}) = 1 \quad (16)$$

where η is the threshold for ultrasonic measurements.

3.4. Characteristic Function

The characteristic function imposes constraints on the correspondence between the position and the context, and embodies the prior knowledge available from the floormap. In the preceding section, we have defined a function \mathcal{M} that sets up the relationship between the context and the position/velocity, i.e., $m_k = \mathcal{M}(\mathbf{z}_k)$, and \mathcal{M} can be readily read out from the contextual map. We thereby define the characteristic function to be

$$\mathcal{C}_k(\mathbf{z}_k, m_k) = \mathcal{I}[\mathcal{M}(\mathbf{z}_k) - m_k = 0] \quad (17)$$

where $\mathcal{I}[\cdot]$ is an indicator function. In other words, the characteristic function enforces the local correspondence defined by \mathcal{M} .

4. Context-Augmented Particle Filter

In this section, we will discuss how to perform inference on the underlying factor graph of the tracking problem we formulated previously. The particle filter is a technique for implementing a recursive Bayesian filter by Monte-Carlo simulations [25]. The key idea of particle filter is to represent the required posterior density function by a set of random samples or “particles” associated with discrete probability mass, and compute the state estimate based on these “particles”. The original particle filter proposed by Gordon *et al.* [26] was designed for a simple hidden Markov chain, which is also a cycle-free factor graph, using the Sampling Importance Resampling (SIR) algorithm to propagate and update the particles. However, the factor graph in our problem, as illustrated in Section 4, does have cycles due to the introduction of the context variable, and only approximate inference algorithms exist. We present a recursive approximate inference method for the cyclic factor graph by extending the particle filter and the resulting algorithm is termed *Context-Augmented Particle Filter* (CAPF).

To see the operation of the CAPF, consider a set of particles $\{\mathbf{z}_{k-1}^i, m_{k-1}^i\}_{i=1}^N$ that represents the posterior distribution $p(\mathbf{z}_{k-1}, m_{k-1} | y_{1:k-1})$ of the state. Note that m_{k-1}^i can be uniquely determined by \mathbf{z}_{k-1}^i via the characteristic function. At time k , we have some new measurement y_k . It is required to construct a new set of particles $\{\mathbf{z}_k^i, m_k^i\}_{i=1}^N$ which characterizes the posterior distribution $p(\mathbf{z}_k, m_k | y_{1:k})$. Now, suppose we have an “oracle” that is capable of providing the context value m_k^i of the corresponding \mathbf{z}_k^i even before we generate \mathbf{z}_k^i s, then our task is equivalent to draw samples from the distribution

$$p(\mathbf{z}_k | m_k, y_{1:k}) \quad (18)$$

This can be carried out in two steps: First, the historical density $p(\mathbf{z}_{k-1}, m_{k-1} | y_{1:k-1})$ is propagated via the transition model $p(\mathbf{z}_k | \mathbf{z}_{k-1}, m_k, m_{k-1})$ to produce the prediction density

$$p(\mathbf{z}_k | m_k, y_{1:k-1}) = \int p(\mathbf{z}_k | \mathbf{z}_{k-1}, m_k) p(\mathbf{z}_{k-1} | y_{1:k-1}) d\mathbf{z}_{k-1} \quad (19)$$

where $p(\mathbf{z}_k | \mathbf{z}_{k-1}, m_k) = p(\mathbf{z}_k | \mathbf{z}_{k-1}, m_k, m_{k-1})$ since m_{k-1} is completely determined conditioning on \mathbf{z}_{k-1} . Second, our interested density $p(\mathbf{z}_k | m_k, y_{1:k})$ can be updated from the prediction density using Bayes’ theorem,

$$p(\mathbf{z}_k | m_k, y_{1:k}) = \frac{p(y_k | \mathbf{z}_k) p(\mathbf{z}_k | m_k, y_{1:k-1})}{p(y_k | y_{1:k-1}, m_k)} \quad (20)$$

$$= \gamma p(y_k | \mathbf{z}_k) p(\mathbf{z}_k | m_k, y_{1:k-1}) \quad (21)$$

where γ is a normalization constant. Thus, Equations (19) and (20) form a recursive solution to Equation (18). In particle filter framework, the aforementioned prediction and update steps are performed by propagating and weighting the random samples.

Prediction Step. In the prediction phase, we generate the predicted particles by

$$\tilde{\mathbf{z}}_k^i \sim p(\mathbf{z}_k | \mathbf{z}_{k-1}^i, \tilde{m}_k^i, m_{k-1}^i) \quad (22)$$

where $\{\tilde{m}_k^i\}_{i=1}^N$ is a set of particles representing the estimates of m_k produced by the “oracle”. Given the different possible values of m_{k-1}^i and \tilde{m}_k^i , $\tilde{\mathbf{z}}_k^i$ will be sampled from different models, detailed in Table 2. We will then perform sanity check on newly generated particles, where the particles $\tilde{\mathbf{z}}_k^i$ absent from the reachable set of \mathbf{z}_{k-1}^i will be eliminated.

Update Step. To update, each predicted particle $\tilde{\mathbf{z}}_k^i$ is assigned with a weight proportional to its likelihood.

$$\tilde{w}_k^i = p(y_k | \tilde{\mathbf{z}}_k^i) \quad (23)$$

The weight is then normalized by

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i} \quad (24)$$

We resample N times with replacement from the set $\{\tilde{\mathbf{z}}_k^i\}_{i=1}^N$ using weights $\{w_k^i\}_{i=1}^N$ to obtain a new set of samples $\{\mathbf{z}_k^i\}_{i=1}^N$ such that $p(\mathbf{z}_k^i = \tilde{\mathbf{z}}_k^i) = w_k^i$. Correspondingly, the contexts m_k^i 's are obtained through the characteristic function, i.e.,

$$m_k^i = \mathcal{M}(\mathbf{z}_k^i) \quad (25)$$

“Oracle” Design. The oracle is supposed to be able to answer the query about the next possible contexts m_k , based upon which the position/velocity component of the state can be properly propagated according to different transition models. For computational efficiency, we adopt a simple discriminative model to produce \tilde{m}_k 's. Given a small database of WiFi fingerprints, we apply the K-Nearest Neighbors (K-NN) algorithm and a modified distance weighted rule to generate an empirical distribution of the context. To be specific, let the WiFi database be denoted by $\{(m^j, y_w^j)\}_{j=1}^{N_w}$, and N_w is the number of WiFi fingerprints. When the new WiFi observation y_k is querying the possible contexts, the K nearest neighbors of y_k are found among the given training set. Let these K nearest neighbors of y_k , with their associated context, be given by $\{(m^{j'}, y_w^{j'})\}_{j'=1}^K$. In addition, let the corresponding distances of these neighbors from y_k be given by $d^{j'}$, $j' = 1, \dots, K$. The weight attributed to the j' th nearest neighbor is then defined as

$$\tilde{q}^{j'} = \frac{d^K - d^{j'}}{d^K - d^1}, \quad j' = 1, \dots, K \quad (26)$$

We then normalize the weights, $q^{j'} = \frac{\tilde{q}^{j'}}{\sum_{j'=1}^K \tilde{q}^{j'}}$, and sample the context according to the following discrete probability distribution,

$$P(\tilde{m}_k = m^{j'}) = \begin{cases} q^{j'}(1 - \alpha) + \alpha, & m^{j'} = m_{k-1} \\ q^{j'}(1 - \alpha), & m^{j'} \neq m_{k-1} \end{cases} \quad (27)$$

where α is a context resilience factor and $\alpha \in [0, 1]$. We incorporate α to accommodate for the prior knowledge that the context will not change too often and to make the “oracle” more robust to the observation noise. Moreover, for the particles on the boundary of distinctive contexts, \tilde{m}_k is equally probable to be these contexts. The pseudo-code of the CAPF algorithm is provided in Algorithm 1.

Algorithm 1 Context-Augmented Particle Filter

```

function CAPF( $y_{1:T}$ ,  $wifi\_database$ ,  $reachable\_set$ )
  Initialization:
  Uniformly generate  $N$  samples  $\{\mathbf{z}_0^i\}_{i=1}^N$ 
  Set  $m_0^i = \mathcal{M}(\mathbf{z}_0^i)$ ,  $w_0^i = N^{-1}$ ,  $i = 1, \dots, N$ 
  for  $k = 1, \dots, T$  do
    for  $i = 1 : N$  do
      Context Estimate:
      if  $\mathbf{z}_{k-1}^i$  on the boundary of  $\{m_b\}_{b=1}^B$  then
        Uniformly sample  $\tilde{m}_k^i$  from  $\{m_b\}_{b=1}^B$ 
      else
        Sample  $\tilde{m}_k^i$  from Equation (27)
      end if
      Prediction Step:
       $\tilde{\mathbf{z}}_k^i \sim p(\mathbf{z}_k | \mathbf{z}_{k-1}^i, \tilde{m}_k^i, m_{k-1}^i)$ 
      Discard particles  $\tilde{\mathbf{z}}_k^i \notin reachable\_set(\mathbf{z}_{k-1}^i)$ 
      Update Step:
      Compute weight  $\tilde{w}_k^i = p(y_k | \tilde{\mathbf{z}}_k^i)$ 
    end for
    Normalize weights:  $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}$ 
    Resampling:
    Select  $N$  particle indices  $i' \in \{1, \dots, N\}$  according to weights  $\{w_k^i\}_{i=1}^N$ 
    Set  $\mathbf{z}_k^{i'} = \tilde{\mathbf{z}}_k^{i'}$  and  $w_k^{i'} = N^{-1}$ 
    Set  $m_k^{i'} = \mathcal{M}(\mathbf{z}_k^{i'})$ 
    Estimate:
     $\hat{\mathbf{z}}_k = \sum_{i=1}^N w_k^i \mathbf{z}_k^i$ 
  end for
  return  $\hat{\mathbf{z}}_{1:T}$ 
end function

```

5. Performance Evaluation

Our experiment was carried out in the Singapore–Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) located in CREATE Tower at the National University of Singapore campus, which is a typical office environment consisting of cubicles, individual offices, corridors and obstacles like walls, desks, *etc.* The total area of the testbed is around 1000 m². There are 10 WiFi routers and four ultrasonic stations deployed in the testbed in total. We utilize TP-LINK TL-WDR4300 Wireless N750 Dual Band Routers (manufactured in Shenzhen, China) as WiFi APs and HC-SR04 Ultrasonic Sensors (manufactured in Shenzhen, China) as the components of ultrasonic stations. The floormap and the corresponding contextual map are shown in Figure 5. Different contexts are colored differently in the contextual map. The static space contains the seating areas in the cubicles and offices, where occupants hardly move. The corridors of horizontal and vertical directions are considered to be two types of constrained spaces (HCS and VCS, respectively). The free space includes the open areas where occupants can freely move. We seek to answer the questions including how well MapSentinel is

able to track the occupant, and whether the map information exploited by way of MapSentinel can bring additional benefits to the tracking performance.

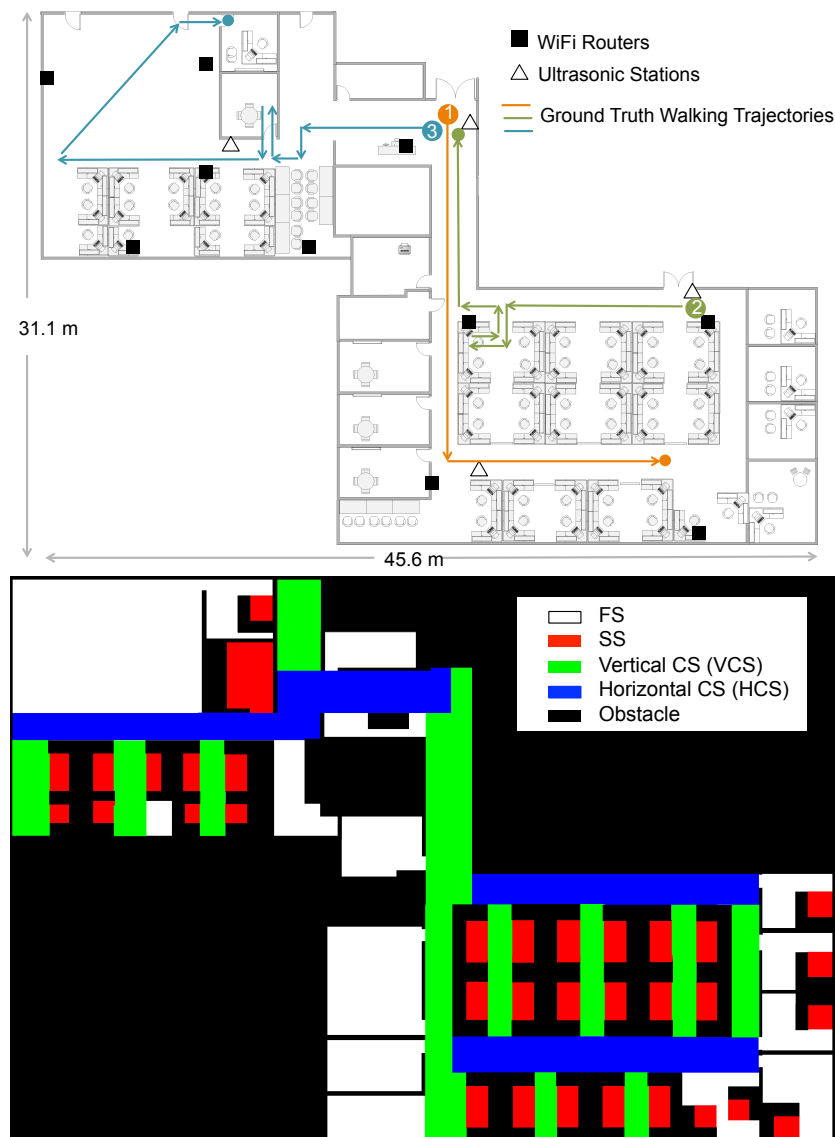


Figure 5. The floormap (**top**) and corresponding contextual map (**bottom**) of the testbed. Four different contexts (FS, SS, VCS, HCS) are defined and color coded as illustrated in the legend.

Experimental methodology. In a real-world setting, we expect the occupant to carry the smartphone as they walk through various sections of an indoor space. Moreover, occupants are unlikely to walk continuously; they would walk between locations of special interest and dwell at certain locations for a significant length of time. Our experiment aims at emulating these practical scenarios in an office environment and incorporating all the contexts defined in our model. Therefore, the following routes were designed as the ground truth for evaluation: (1) A enters the office from the front gate and walks through the corridors to find her colleague (different CSs are included); (2) B enters the office from the side door, walks to her own seat, stays there for a while and exits the office from the front gate (CSs, SS are included); (3) C enters the office from the front gate, walks through corridors, takes some time at her office and goes to the open area (CSs, SS, FS are included). We asked the experimenter to behave as usual when walking in the space. At the same time, the WiFi APs and ultrasonic stations constantly collect the measurements and send them to the central server.

To obtain the ground truth at the sampling time of the tracking system, we mark the ground with a 1 m grid on the pre-specified route and ask the experimenter to create lap times with a stopwatch when happening to be on the grid. By recording the starting time of the experiment, we can obtain the time stamp of each grid and then interpolate the ground truth at the sampling time.

Does the “oracle” work? The current context estimation done by the “oracle” is critical to the CAPF algorithm, as the tuple of the current and previous context jointly steer the states in our model. Here, we would like to evaluate the context prediction performance of the “oracle” we constructed in light of several design rules presented in the Section 4. Figure 6 illustrates the result of the context estimation for different walks. Since the context estimates are represented by a set of particles in the algorithm, we visualize the context estimate by the purple lines centered at the possible contexts, and the lengths of the purple lines are scaled by the proportions of the particles of different contexts. Ideally, the purple cloud should scatter around the ground truth context. Figure 6 suggests that the estimates given by the “oracle” can generally capture the ground truth. Evidently, the context estimate is not perfect, especially for the static space (SS). However, these approximate “ground truths” essentially present other possibilities of the current context and avoids particles trapping in the static space. We define the context estimation accuracy to be the ratio of the number of particles with correct context estimate to the total number of particles. The context estimation accuracy is calculated for each time step of the experiments, and the empirical distribution of the context estimation accuracy is illustrated in Figure 7, where the mean accuracy is 52.41%. With this noisy “oracle”, the system can achieve median tracking error of 1.96 m, while the tracking error would be 1.84 m if a perfect “oracle” was utilized. Therefore, our work has the potential to be further improved with a more advanced “oracle” design.

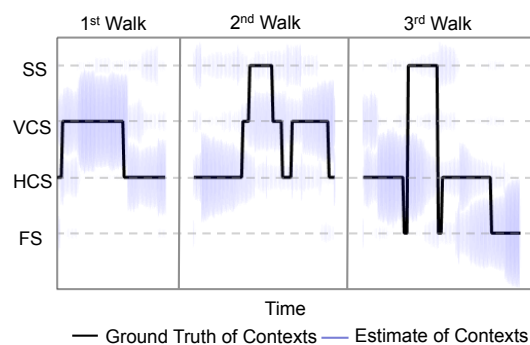


Figure 6. The context estimate produced by the “oracle” *versus* the ground truth context. The radius of the purple cloud is proportional to the number of particles of the estimated context which the cloud is centered around.

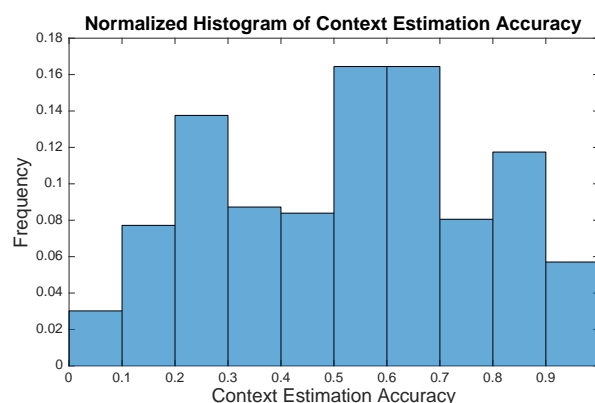


Figure 7. Normalized histogram of context estimation accuracy of the “oracle”. The mean accuracy is 52.41%.

Figure 8 demonstrates some snapshots of the CAPF algorithm in progress. At the beginning, the particles are initialized to be uniformly distributed in the space. In addition, the spread of the particles shrinks as the new WiFi observations come. When the ultrasonic station reports a detection, the particles are concentrated in the corresponding detection zone. As the occupant exits the detection zone, the particles spread out along the direction of the corridor. When the occupant sits in the cubicle, the particles distribute over the seating area as well as some possible routes through which the occupant might leave the seating area. The particles distribute evenly along different directions when the occupant is moving in the free space, in which case our model is identical to the traditional constant velocity dynamic model for the particle filter.

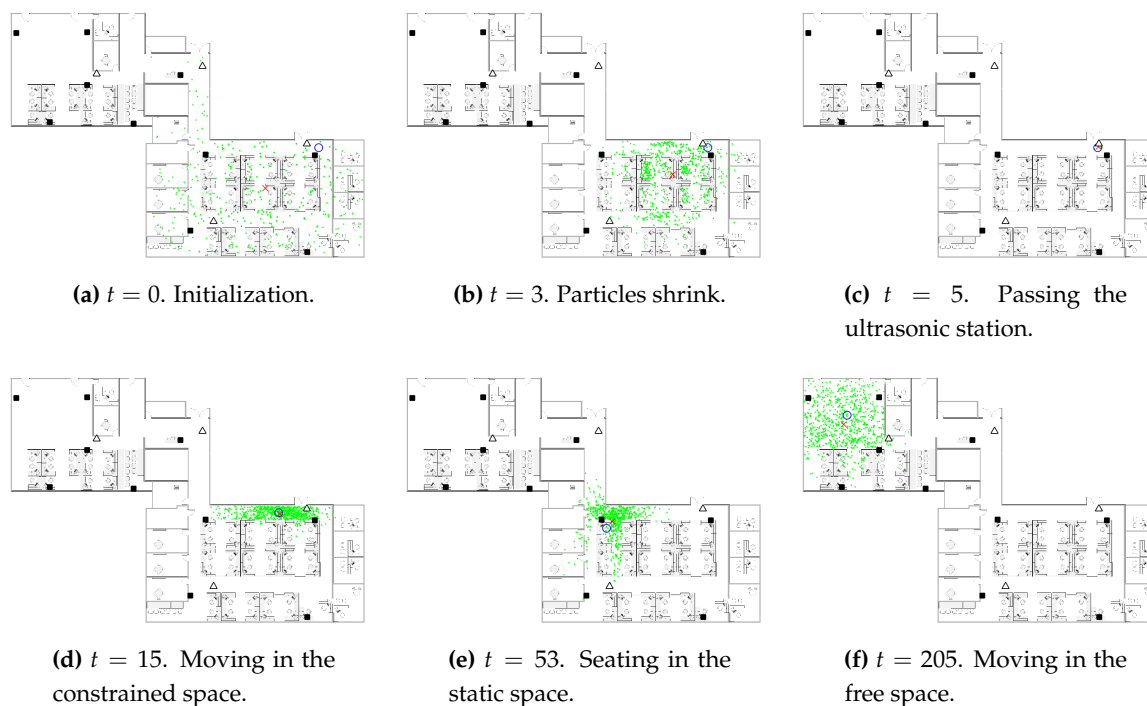


Figure 8. The snapshots of the intermediate steps of the CAPF algorithm visualized. The location estimate, ground truth location, particles are presented by the red cross, blue circle, green dots, respectively. As before, the black square and white triangles give the positions of WiFi routers and ultrasonic stations.

MapSentinel's tracking performance. We aggregate the data from different walks and compare the performance of MapSentinel against the fusion system of WiFi and ultrasonic station without leveraging the floormap information, as well as the purely WiFi-based tracking system. The tracking error distributions are depicted in Figure 9. As can be seen, the MapSentinel achieves an essential performance improvement, 31.3% over the WiFi tracking system and 29.1% over the fusion scheme. Note that adding the ultrasonic calibration into the WiFi system is able to realize a small amount of accuracy increment. Due to the high degree of uncertainty of WiFi signals, the effect of ultrasonic calibration will not last for long. The map information elongates the effect of the ultrasonic calibration via imposing additional constraints to the motion, and that is why MapSentinel greatly enhances the tracking performance compared with the purely WiFi-based system. We also evaluate the tracking performance in different contexts, and the result is shown by boxplots in Figure 10. Here, “without map” means using the WiFi and ultrasonic sensing systems without taking into account the reachable set as well as the context-dependent kinematic model. A unified dynamical model, the free space model, is applied in this case, and a traditional particle filter is implemented to estimate the location.

As can be readily read from the figure, the MapSentinel performs better in all contexts. More significant increase is achieved in constrained spaces and static spaces, as expected.

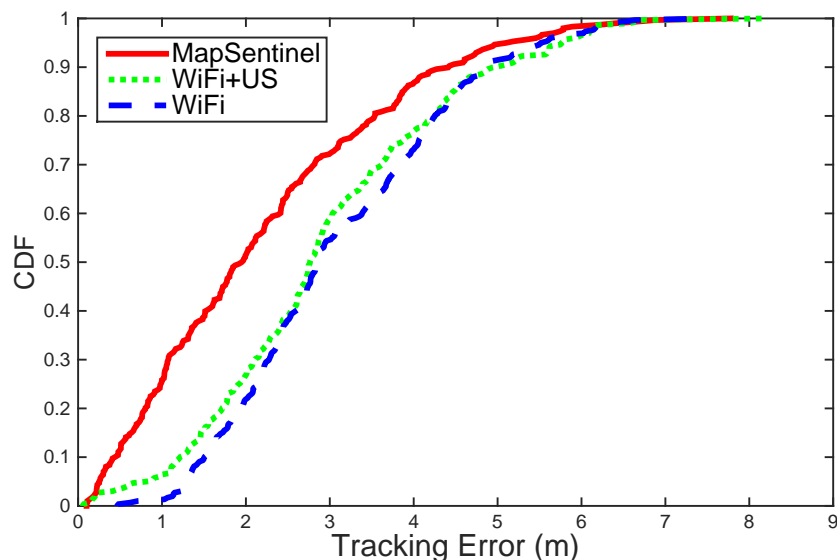


Figure 9. Tracking performance of MapSentinel, the fusion system of WiFi and ultrasound sensor, the pure WiFi system. The median tracking accuracy of the MapSentinel is 1.96 m, MapSentinel can achieve the performance improvement of 31.3% over the purely WiFi-based tracking system, 29.1% over the fusion system.

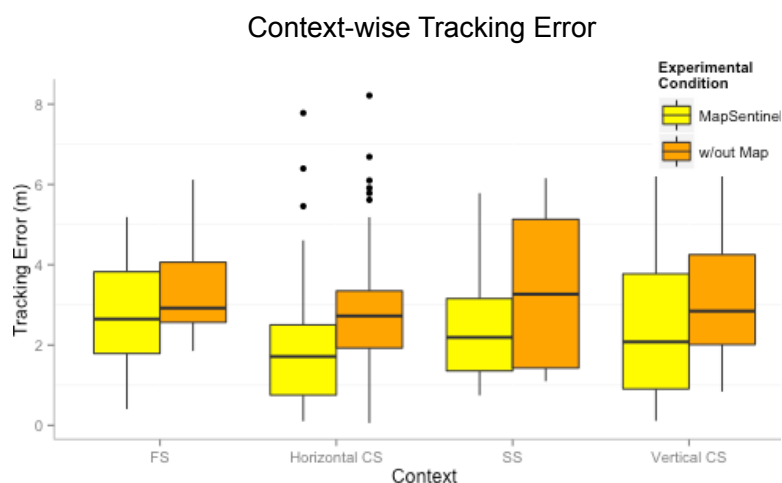


Figure 10. Tracking error in different contexts for the MapSentinel and the WiFi+Ultrasound system.

Figure 11 compares the performance of tracking systems with distinctive floormap usage. MapSentinel exploits the floormap information in two folds: first, MapSentinel integrates the context information into the kinematic model, and the movement patterns of people on different locations of the map are better captured. Secondly, MapSentinel takes into account the speed restrictions as well as physical obstacles in the indoor space by checking if the particles fall inside the reachable set at each time step. The second fold of the floormap information has been widely utilized in the previous work, while the context information is less explored. We therefore compare the tracking error of our system with the one that merely uses the reachable conditions. Figure 11 shows that incorporating information about physical constraints, as the previous work did, is surely beneficial to the tracking system. Particularly, the performance can be further improved by 19.8% by introducing the context information into the tracking system.

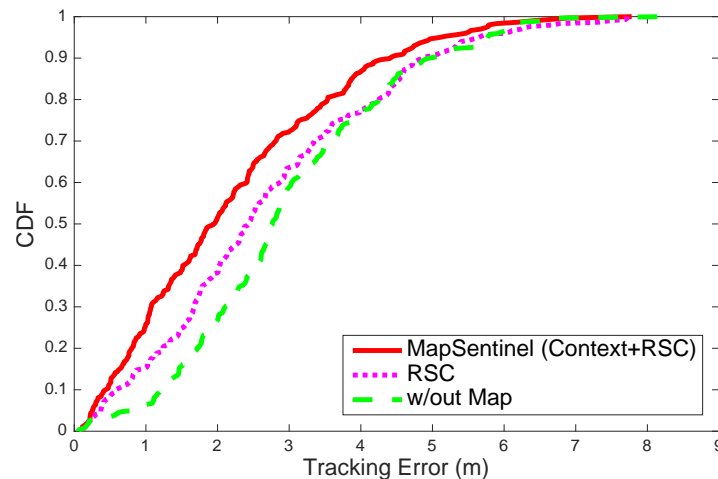


Figure 11. Tracking performance of different usage of floormap information. “RSC” stands for reachable set check. MapSentinel extracts the context information from the floormap, and simultaneously eliminates the particles falling outside the reachable set. MapSentinel is compared with the tracking system without using context information (*i.e.*, only performing RSC) and the one without using the map information at all. The median tracking errors of MapSentinel, the system only performing RSC, and the one without exploiting the floormap information are 1.96 m, 2.44 m and 2.77 m, respectively.

To better understand how the map helps improve the location estimation, we demonstrate the velocity estimation of different tracking schemes in Figure 12. Typically, the occupants will not perform complex motions in the indoor space due to the constraints of the wall and other barricades. The more the velocity estimate deviates from the canonical directions defined by the indoor environment, the worse the tracking performance can be. Using the fusion schemes of WiFi and ultrasonic calibration, only the location is the observable state. The velocity estimates depend largely on the location estimate and it has little effect in smoothing out the location estimate. Hence, extensive research has been focusing on using inertial measurements to perform dead reckoning, which makes the velocity observable. Analogously, the MapSentinel creates a *virtual* inertial sensor for the occupant, which mimics the actual inertial sensor to provide the possible walking speed and directions. As is shown in Figure 12, the velocity estimation without map information tends to point to any direction while the MapSentinel constrains the velocity via the context-dependent kinematic model.

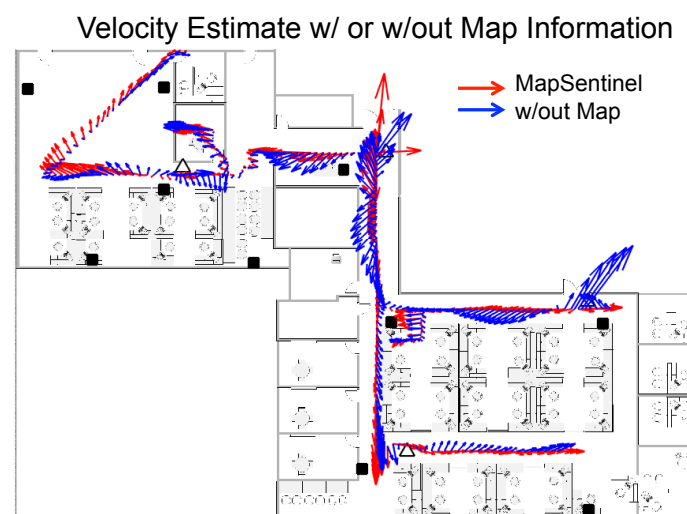


Figure 12. The velocity estimation for the MapSentinel and the WiFi+Ultrasound system. The vector indicates the speed and direction of the estimated motion.

6. Conclusions

This paper presents MapSentinel, a system for real-time location tracking that emphasizes both non-intrusiveness and accuracy. The non-intrusive sensing networks comprise the modified WiFi access points and the ultrasonic calibration stations. The MapSentinel makes novel attempts to exploit the floormap information by categorizing the indoor space into different contexts to capture the diversity of typical motion characteristics. This mimics having an inertial sensor attached to the occupant to obtain the knowledge of velocity. We formalize the fusion of floormap information as well as the noisy sensor readings using the Factor Graph, and develop the Context-Augmented Particle Filtering algorithm to efficiently solve real-time walking trajectories. Our evaluation in the large typical office environment shows that MapSentinel can achieve the performance improvement of 31.3% over the purely WiFi-based tracking system. MapSentinel is among the early attempts to obviate the need for the inertial sensors in indoor tracking, and our results are promising.

For future work, we would like to explore multiple occupant tracking. The ultrasonic sensor is essentially anonymous and cannot identify the occupant entering its detection zone. The WiFi access points are able to identify the occupant from the MAC address of the mobile device and can approximately tell which occupant is approaching the ultrasonic station. The ultrasonic calibration will work if the occupant can be identified with the MAC information without ambiguity; however, if the identity of the occupant within the range cannot be uniquely determined, as in the crowded scenario, the calibration may not work effectively. Further work to reliably track multiple occupants is necessary. Moreover, we would like to integrate our tracking method to the control of lighting and ventilation systems to improve energy efficiency of buildings.

Acknowledgments: This research is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore–Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

Author Contributions: Ruoxi Jia proposed the information fusion framework; Ruoxi Jia, Ming Jin, Han Zou and Yigitcan Yesilata conducted the experiment. Lihua Xie and Costas Spanos supervised the work and revised the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Want, R.; Hopper, A.; Falcao, V.; Gibbons, J. The active badge location system. *ACM Trans. Inf. Syst.* **1992**, *10*, 91–102.
2. Medina, C.; Segura, J.C.; De la Torre, A. Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy. *Sensors* **2013**, *13*, 3501–3526.
3. Priyantha, N.B.; Chakraborty, A.; Balakrishnan, H. The cricket location-support system. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, 6–11 August 2000; pp. 32–43.
4. Hazas, M.; Hopper, A. Broadband ultrasonic location systems for improved indoor positioning. *IEEE Trans. Mob. Comput.* **2006**, *5*, 536–547.
5. Weekly, K.; Zou, H.; Xie, L.; Jia, Q.S.; Bayen, A.M. Indoor occupant positioning system using active RFID deployment and particle filters. In Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina Del Rey, CA, USA, 26–28 May 2014; pp. 35–42.
6. Bekkali, A.; Sanson, H.; Matsumoto, M. RFID indoor positioning based on probabilistic RFID map and Kalman filtering. In Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, White Plains, NY, USA, 8–10 October 2007; pp. 21.
7. Saab, S.S.; Nakad, Z.S. A standalone RFID indoor positioning system using passive tags. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1961–1970.
8. Chen, Z.; Zou, H.; Jiang, H.; Zhu, Q.; Soh, Y.C.; Xie, L. Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. *Sensors* **2015**, *15*, 715–732.

9. Beauregard, S.; Haas, H. Pedestrian dead reckoning: A basis for personal positioning. In Proceedings of the Third Workshop on Positioning, Navigation and Communication, Hannover, Germany, 16 March 2006; pp. 27–35.
10. Wang, H.; Sen, S.; Elgohary, A.; Farid, M.; Youssef, M.; Choudhury, R.R. Unsupervised indoor localization. In Proceedings of the Tenth International Conference on Mobile systems, Applications, and Services, New York, NY, USA, 25 June 2012; pp. 197–210.
11. Bose, A.; Foh, C.H. A practical path loss model for indoor WiFi positioning enhancement. In Proceedings of the 6th International Conference on Information, Communications & Signal Processing, Singapore, 10–13 December 2007; pp. 1–5.
12. Evennou, F.; Marx, F.; Novakov, E. Map-aided indoor mobile positioning system using particle filter. In Proceedings of the 2005 IEEE Wireless Communications and Networking Conference, New Orleans, LA, USA, 13–17 March 2005; pp. 2490–2494.
13. Wang, H.; Lenz, H.; Szabo, A.; Bamberger, J.; Hanebeck, U.D. Enhancing the map usage for indoor location-aware systems. In *Human-Computer Interaction. Interaction Platforms and Techniques*; Springer: Berlin, Germany, 2007; pp. 151–160.
14. Evennou, F.; Marx, F. Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. *Eurasip J. Appl. Signal Process.* **2006**, 2006, 164.
15. He, C.; Guo, S.; Yang, Y. Voronoi diagram based indoor localization in wireless sensor networks. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 3269–3274.
16. Liao, L.; Fox, D.; Hightower, J.; Kautz, H.; Schulz, D. Voronoi tracking: Location estimation using sparse and noisy sensor data. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 27–31 October 2003; pp. 723–728.
17. Swangmuang, N.; Krishnamurthy, P. Location fingerprint analyses toward efficient indoor positioning. In Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications, Hong Kong, China, 17–21 March 2008; pp. 100–109.
18. Hilsenbeck, S.; Bobkov, D.; Schroth, G.; Huitl, R.; Steinbach, E. Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, WA, USA, 13–17 September 2014; pp. 147–158.
19. Gusenbauer, D.; Isert, C.; Krösche, J. Self-contained indoor positioning on off-the-shelf mobile devices. In Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Zurich, Switzerland, 15–17 September 2010; pp. 1–9.
20. Park, J.G. Indoor localization using place and motion signatures. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2013.
21. Kaiser, S.; Khider, M.; Robertson, P. A human motion model based on maps for navigation systems. *EURASIP J. Wirel. Commun. Netw.* **2011**, 2011, 1–14.
22. Skiena, S.S. In *The Algorithm Design Manual: Text*; Springer Science & Business Media: Berlin, Germany, 1998.
23. Arulampalam, M.S.; Gordon, N.; Orton, M.; Ristic, B. A variable structure multiple model particle filter for GMTI tracking. In Proceedings of the Fifth International Conference on Information Fusion, Annapolis, MD, USA, 8–11 July 2002; pp. 927–934.
24. Ferris, B.; Haehnel, D.; Fox, D. Gaussian processes for signal strength-based location estimation. In Proceedings of the Robotics Science and Systems, Philadelphia, PA, USA, 16–19 August 2006.
25. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, 50, 174–188.
26. Gordon, N.J.; Salmond, D.J.; Smith, A.F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F Radar Signal Process.* **1993**, 140, 107–113.

