Offline Docked-Phone Indoor Carpark Navigation: Fusing RF and IMU Signals With HMM

Murphy Zheng Zhang, Gary Wing-Ho Cheung, and S.-H. Gary Chan, *Senior Member, IEEE*

***Abstract*—We study the challenging problem of navigating in an indoor carpark in the absence of global navigation satellite system (GNSS) and cellular signals using an offline smartphone docked at the car dashboard. There is basic radio frequency (RF) infrastructure in the venue but, due to signal attenuation by the car body, the location computed based on the in-car signal is noisy and intermittent. Previous works on carpark navigation often require costly specialized equipment as on-car additional infrastructure (OCAI), or suffer from error propagation stemmed from integrating inertial measurement unit (IMU) signals over time. We propose RICH, a novel, simple, accurate, and cost-effective docked-phone approach to fuse RF and IMU signals for indoor carpark navigation using the hidden Markov model (HMM). RICH uses IMU signals to detect the speed level and turning of the car, which is then fused with the crude RF localization in an HMM framework to estimate the car's location distribution in real-time. We further present an analysis on the tradeoff between computation and accuracy of RICH. Our extensive experiments on smartphone in real carparks show that, as compared with the state-of-the-art, RICH achieves substantially lower localization error (by 40%) with high-computational efficiency (less than 10 ms per location).**

***Index Terms*—Carpark navigation, fusion localization.**

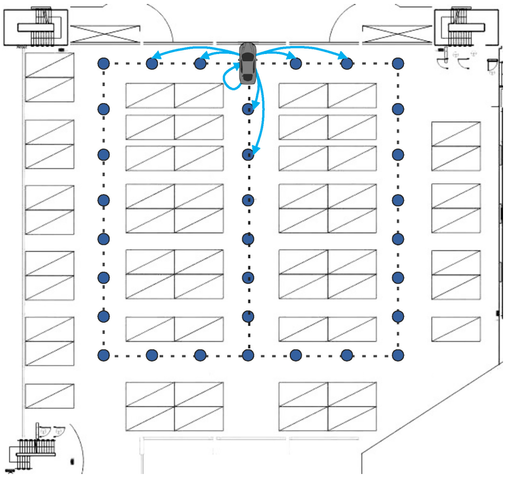


Fig. 1. Typical carpark floorplan with the HMM states corresponding to possible car positions. In a time slot, the car may move at most H hops away from its current position according to some transition probability.

I. Introduction

URBANIZATION has led to increasing number of indoor carparks. In these carparks, a global navigation satellite system (GNSS) and cellular signals are often weak or unavailable. We consider, in this work, the challenging problem of indoor carpark navigation, which is to direct the car driver to a designated spot (parking bay, exit, etc.) in the absence of GNSS and cellular signals. To achieve that, an app, preinstalled in the offline phone docked on the car dashboard, locates the vehicle and makes turn-by-turn instruction to the driver.

An indoor carpark is characterized by well-defined lanes with junctions. We illustrate in Fig. 1 a carpark floorplan in our university, where the vehicle is constrained in paths indicated in dotted lines. A car typically travels in a carpark with some rather regular or predictable speed patterns. For example, after negotiating a corner, a car usually accelerates to around the designated speed limit (e.g., 10–20 km/h) and slows down at the end of the lane or junction to make a turn. Despite such regular features, occasionally the car may unpredictably drop its speed to some slow level or even a complete stop caused by some irregular unexpected events, such as pedestrian crossing or backing of other cars in the front.

To support carpark navigation, we consider the usual case that a basic radio frequency (RF) infrastructure, such as Bluetooth beacons and/or WiFi access points (APs), has already been installed. However, because the RF signals are markedly shielded by the car's body, the ambient in-car RF signals as sensed by the offline docked phone is weak, intermittent, and noisy. Coupled with possibly low-RF scanning frequency, the localization is often not smooth, greatly hampering user navigation experience.

To improve navigation accuracy and smoothness, we propose to augment the RF signals with commonly available on-phone inertial measurement unit (IMU) with 9DOF (9 Degrees-of-Freedom), which consists of a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer to measure the acceleration, angular velocity, and geomagnetic field, respectively. The critical challenge then is how to efficiently and effectively fuse RF and IMU signals in an offline docked phone to locate and navigate the driver in an indoor carpark.

Manuscript received 30 March 2022; revised 14 September 2022 and 19 June 2023; accepted 21 July 2023. Date of publication 7 August 2023; date of current version 24 January 2024. This work was supported in part by the Hong Kong General Research Fund under Grant 16200120. *(Corresponding author: Gary Wing-Ho Cheung.)*

Murphy Zheng Zhang is with the Security Platform Department, Tencent, Shenzhen, Guangdong 518000, China (e-mail: zzhangdq@connect.ust.hk).

Gary Wing-Ho Cheung and S.-H. Gary Chan are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China (e-mail: whcheungam@connect.ust.hk; gchan@cse.ust.hk).

Digital Object Identifier 10.1109/JIOT.2023.3302378

Due to unfavorable in-car RF signal environment, previous works on carpark navigation often deploy on-car additional infrastructures (OCAIs) [17], [18], [28], such as wheel odometers, lidars, and surround-view cameras. Despite promising results, they require the installation of special and costly equipment, hence not applicable to general users. Visual-based approaches [2], [9], [21], [29] have also been proposed for carpark navigation. As they are mainly cloud-based, deploying such system is costly (due to high network bandwidth, computation power, and the number of cameras), and may have latency, occlusion, and privacy concerns. There have been quite a lot of works using RF signal for carpark navigation [1], [6], [8], [10], [24]. However, the works assume a high signal-to-noise ratio of in-car RF signals, which requires high AP density, signal power, and infrastructure cost. Another body of work uses dead reckoning (DR) to estimate object location by means of double integration of linear acceleration [4], [11], [34]. However, this is error-prone due to error propagation over time. To mitigate that, some require highly specialized, and hence highly accurate, IMU sensors beyond what is available in common smartphones today. Yet another body of localization works uses pedestrian pedometer and DR [12], [16], [19], [36]. They are, however, not applicable for docked phone because no steps can be detected in a running car.

We propose RICH, a novel docked-phone approach that fuses on-phone RF and IMU signals for indoor carpark navigation by means of the hidden Markov model (HMM). RICH is designed based on the observation that RF location and IMU information can complement each other to achieve higher localization accuracy. IMU detects car's heading and captures the car's speed pattern to predict the car's location over some upcoming time (in seconds). To mitigate error propagation and location drift, RF and turn landmarks are used to constrain the car location to a region.

To the best of our knowledge, this is the first piece of fusion work deployable by offline docked phone. Due to measurement noise and propagation error, RICH abolishes the traditional integration-based [11], [18] approaches on IMU readings. Instead, it adopts a simple yet efficient deep-learning approach to extract and classify accurately IMU signals corresponding to some distinct vehicular speed levels, such as coming to a full stop, driving at low speed, and traveling normally around the designated speed limit. Moreover, IMU provides valuable information on car heading and turning. To fuse this with the crude and intermittent location estimated from RF, HMM is the most suitable fusion model among typical probabilistic models for the specific car localization task because of a vehicle's regular motion pattern in well-defined driving lanes. Comparing with the particle filter (PF) approach (which is applicable but not optimal), HMM expresses more naturally a vehicle's motion regularities in a carpark by sampling the states only from a vehicle's drivable routes. Meanwhile, the speed regularities (transition probability) of a car in a specific carpark can be learned by a offline (training) process proposed in Section III-A. To represent similar constraints, a PF-based method needs to apply mass constrain-rules on a sufficient large number of particles in the prediction and resampling step, which spends more computation but performs worse.

Time is slotted in RICH with size in the order of a fraction of a second (0.2 s in our experiment). The driving lanes are divided into grid points (as shown in Fig. 1), which are the possible states of the car at a particular time slot. The RICH models the state transition with an HMM: a car may move to a grid point at most H hops away from its current one in the next slot according to some time-varying transition probability, which depends jointly on the speed pattern as observed from IMU readings, car heading, and turn, location as estimated from the RF readings, and the previous location distribution of the car.

The contributions of this article are summarized as follows.

1) *RICH, Novel Fusion-Based Carpark Navigation for Offline Docked Phones:* RICH is the first mass-deployable approach for offline docked phones to navigate cars in indoor carpark of any layout. It is OCAI-free, simple, memory and computationally efficient, and implementable in commodity smartphones. It fuses RF signals and phone IMU readings by means of an HMM to fully utilize a vehicle's motion constraints in an indoor carpark, and thus achieve real-time and highly accurate localization. Note that though we use RF signal in this work, RICH is an HMM framework to fuse any signal (such as GPS or cellular) with IMU for indoor carpark navigation.
2) *Computational Analysis of RICH:* We provide an analysis on the tradeoff between computation overhead and localization accuracy. Accordingly, users can adjust system parameters for different mobile phones to balance among phone computation time and accuracy in a carpark.
3) *Extensive Experimental Evaluation in Real Sites:* We have implemented RICH in mobile phones and compare it with existing state-of-the-art schemes in real carpark sites. Our results show that RICH achieves substantially better performance, with a 40% reduction in localization error and a 60% reduction in computation time as compared with other schemes.

The remainder of this article is summarized as follows. After discussing related work in Section II, we overview the offline and online phases of RICH and formulate and present the fusion model in Section III. Then, we present the speed collection in the offline phase in Section IV, and online signal processing of RICH in Section V, followed by HMM fusion and its complexity analysis in Section VI. We present our experimental results in Section VII and conclude in Section VIII.

II. Related Works

We discuss related works in three aspects: 1) OCAI-based indoor vehicle localization (Section II-A); 2) RF-based localization (Section II-B); and 3) probability-based fusion approaches (Section II-C).

A. OCAI-Based Indoor Vehicle Localization

OCAI-based localization locates a vehicle by installing specialized sensors or equipments in a car. Early vehicle localization methods focus on vehicular ad-hoc networks

(VANETs) [13], [14], [30], whose basic idea is to perform cooperative localization through vehicle-to-vehicle communication. With the rapid development of simultaneous localization and mapping (SLAM), more works have been proposed locating vehicles with wheel odometers [26], [32], lidars, surround-view cameras, and high-accuracy IMU sensors. A fusion of lidar and IMU is proposed in [17]. AVP-SLAM [28] proposes a visual semantic SLAM approach using surround-view cameras, IMU, and wheel odometers. These SLAM-based methods are shown to achieve centimeter level localization accuracy. Another approach, such as [25], proposes a fusion of WiFi, IMU, and lidar applying a Gaussian-mixture PF model.

OCAI-based methods often require special installation and hence are more customized and costly. RICH is a cost-effective solution based on offline docked smartphone for general car users without the need for additional installation of on-car sensors. The OCAI works are orthogonal and complementary to ours and may be used to further improve the localization accuracy.

B. RF-Based Localization

Received signal strength (RSS) of RF signals has been leveraged to locate targets. Typical RF signal includes WiFi, Bluetooth low energy (BLE), Zigbee, etc. Fingerprinting (FP) has become the most popular RF-based indoor localization method. The first FP system is RADAR [1]. Horus in [35] addresses the problem of the temporal variations in RF signals. By considering the channel state information (CSI), ArrayTrack [33] achieves submeter-level accuracy. Besides FP approach, several works, such as weighted centroid localization (WCL) localization [3], EZ [8], and EZPerfect [24], map the RSSI readings to physical distances to estimate the location with geometric methods. IncVoroni [10] proposes a Voronoi graph approach to refine user location over time.

While the above works are impressive, they cannot be directly applied for in-car docked-phone navigation because the in-car signal is not strong. While the RF-based methods are not accurate and responsive enough for real-time car navigation [6], they can provide crude first-order location estimation. RICH leverages that and fuses it with IMU using HMM to provide higher accuracy.

C. Probability-Based Fusion Approaches

The probability-based fusion localization method fuses multiple sensor or localization system measurements based on probability. The Kalman filtering and its variants, i.e., EKF and UKF are the common methods studied and applied to localization. These works [7], [20], [22] adopt external signals, such as visuals, GNSS, or RF signals, to obtain a rough position and apply DR by speed and direction for a more accurate position estimate. However, in the scenario of using a mobile phone for navigation inside a vehicle, the on-phone IMU signal is subject to self-noise and vehicle vibration interference, making it difficult to achieve DR. Moreover, the in-car RF signals are noisy and intermittent, leading to large and non-Gaussian observation noise. Therefore, the Kalman filter is not applicable to the scenario.

TABLE I
IMPORTANT SYMBOLS USED

Symbol	Meaning
n	The total number of grid points (possible car states)
s_t	The car state at time (slot) t ($s_t = 1, 2, \ldots, n$)
q_j	2-D coordinate for grid point j, $j = 1, 2, 3, \ldots, n$
F	Car speed distribution
S	Total number of speed patterns
ρ_t	Vehicle speed pattern at time t
ϕ_t	Vehicle heading at time t
σ_ϕ	Variance of heading error
$u_t = [\rho_t, \phi_t]$	Vehicle action at time t
r_t	2-D coordinate of RF localization result
σ_r	Variance of RF localization error
l_t	Boolean variable of whether the car is turning at time t
$z_t = [r_t, l_t]$	Observation vector at time t
γ, β	Precision and recall, respectively, for turn detection
$\alpha_t(j)$	Probability that a car is at grid point j at time t
H	Maximum number of hops for a car transition in a slot
V_{max}	Speed limit in the carpark
d	Distance between adjacent grid points (meters)
K	Kernal function for car transition probability
$a_t(i, j)$	Transition probability for a car's transition from i to j at time t.
L	Huber-loss function for fusion localization

To address the nonlinear filtering problem, PF [15] was introduced. Early pedestrian localization works [12], [19], [27] fuses external signals with PDR by particle filtering, achieving around 3 m accuracy. These works are not applicable to vehicle localization due to a lack of periodic patterns. Substitutions of PDR [5], [11], [18] are to estimate a vehicle's travel distance by double integration of forward acceleration. Such methods are also not applicable in our scenario, because mobile phone IMU often has limited accuracy and the localization error accumulates significantly. Another work [25] applies PF by assuming a constant vehicle speed which is not common practice.

Our proposed HMM differs from the aforementioned works by leveraging a vehicle's motion constraints (driving patterns) in a carpark as prior information. By embedding these motion constraints into HMM, the state space of the vehicle (velocity, position, direction) has been transformed from continuous, infinite space to finite, discrete space. Observations that do not conform to motion constraints (such as heading deviation, over-speeding, or position exceeding lane boundaries) will be corrected accordingly. At the same time, limiting the vehicle to a finite state space can reduce computational complexity.

III. System Overview

In this section, we overview RICH and formulate the fusion model, by presenting its offline training phase, online navigation phase, and the HMM fusion model in Sections III-A–III-C, respectively. We summarize the important symbols used in this work in Table I.

A. Offline Training Phase

In the offline training phase, the survey data is first collected in the carpark. Drivers drive naturally around the carpark while a surveyor in the passenger seat records driving trajectories and IMU signals with a mobile phone, where a driving trajectory refers to the car locations and their timestamps. Normally, it takes 1–2 h of site survey to set up one typical carpark.

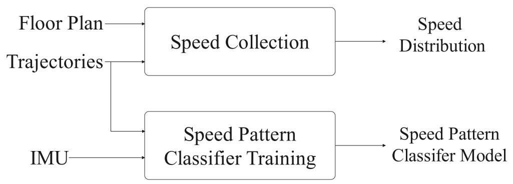

Floor Plan
Trajectories
IMU
Speed Collection
Speed Pattern Classifier Training
Speed Distribution
Speed Pattern Classifer Model

Fig. 2. Offline training phase.

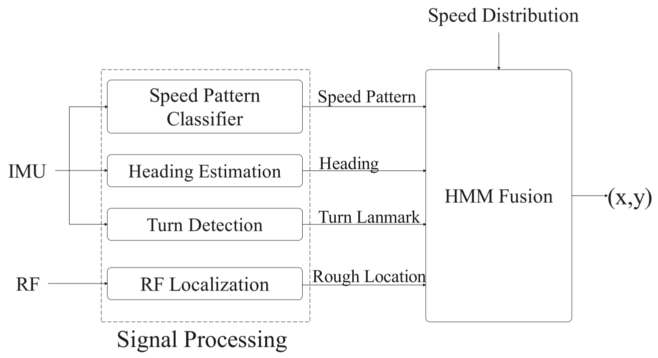


Fig. 3. Online navigation phase.

We show in Fig. 2 the offline phase, which consists of the following.

1) *Speed Collection:* Vehicle motion in a carpark is classified into S speed patterns. Here, a speed pattern is described by a collection of vehicle speed over a period of time, say 2 s. Let n be the number of grid points in the carpark whose coordinates are $q_1, q_2, \ldots, q_n$. The vehicle speed distribution $f_i(q_j, v)$ corresponding to each speed pattern $i \in [0, S-1]$ at grid point $q_j, j \in [1, n]$ is collected from the driving trajectories to estimate the transition probability of the HMM.
2) *Speed Pattern Classifier Training:* A speed pattern classifier model is trained with the sourced IMU signals. This is then saved to classify the car speed pattern in the online navigation phase.

B. Online Navigation Phase

We overview the online navigation phase in Fig. 3, which consists of the following.

1) *Signal Processing:* Raw IMU and RF signals are first processed in the signal processing step in the docked phone. Taking computation overhead into consideration, we apply simple but efficient algorithms for each individual algorithm instead of building an end-to-end model for all. At time slot t, the *speed pattern classifier* trained in the offline phase processes the raw IMU signals and outputs the vehicle's real-time speed pattern ρ_t, while *heading estimation* detects the vehicle's heading ϕ_t using IMU. We also perform *turn detection* using a gyroscope to identify the turn landmark l_t. At the same time, *RF localization* estimates the vehicle's rough location r_t independently.
2) *HMM Fusion:* All terms, ρ_t, ϕ_t, l_t, and r_t, are fused with the speed distribution $f(v, q_1), f(v, q_2), \ldots, f(v, q_n)$ in an HMM to estimate the vehicle location.

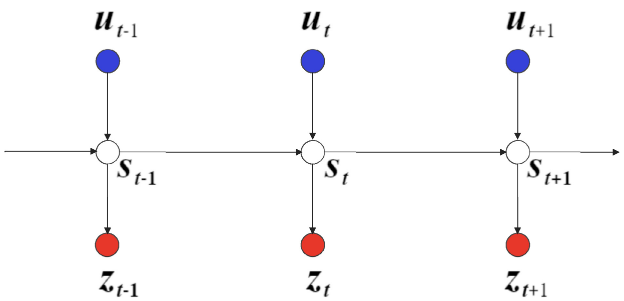


Fig. 4. Dependence graph of RICH. The vehicle state s_t at time t is dependent on the previous state s_{t-1} and the vehicle's action u_t. Meanwhile, the vehicle state can be observed by z_t.

C. Fusion Model Formulation

In this section, we formulate the HMM fusion model of RICH. The HMM in RICH is composed of the following elements: state s, action u, and observation z shown in Fig 4. Let s_t be the vehicle state, given by any one of the grid points, at time t. The vehicle's action, defined as

$$u_t = [\rho_t, \phi_t] \tag{1}$$

changes vehicle state from s_{t-1} to s_t. Recall that ρ_t and ϕ_t represent the vehicle's speed pattern and heading, respectively, both derivable from the IMU readings. Meanwhile, we infer the vehicle's location from an observation z_t given by

$$z_t = [r_t, l_t]. \tag{2}$$

Recall that r_t and l_t represent the RF localization result and the turn landmark, respectively. The fusion localization problem is to estimate the hidden state s_t given a sequence of actions $u_{1:t} = [u_1, u_2, \ldots, u_{t-1}, u_t]$ and observations $z_{1:t} = [z_1, z_2, \ldots, z_{t-1}, z_t]$. Formally, the fusion objective is to estimate

$$p(s_t = j | u_{1:t}, z_{1:t}) \tag{3}$$

the conditional probability that a car is at stage j for all $j = 1, 2, \ldots, n$. Applying the Bayesian rule

$$\begin{aligned} p(s_t = j | u_{1:t}, z_{1:t}) &= \frac{p(s_t = j, u_{1:t}, z_{1:t})}{p(u_{1:t}, z_{1:t})} \\ &\propto p(s_t = j, u_{1:t}, z_{1:t}) \\ &= \alpha_t(j). \end{aligned} \tag{4}$$

Usually, the term $\alpha_t(j)$ is also called the "forward variable," representing the joint probability that a car is at grid point j at time t. Applying the chain rule and the theory of conditional independence, we have

$$\alpha_t(j) = p(z_t | s_t = j) \sum_i p(s_t = j | s_{t-1} = i, u_t) \alpha_{t-1}(i). \tag{5}$$

Equation (5) shows that the joint distribution $\alpha_t(1), \alpha_t(2), \ldots, \alpha_t(n)$ can be recursively computed from the historical distribution $\alpha_{t-1}(1), \alpha_{t-1}(2), \ldots, \alpha_{t-1}(n)$. Let

$$\overline{\alpha_t(j)} = \sum_i p(s_t = j | s_{t-1} = i, u_t) \alpha_{t-1}(i) \tag{6}$$

such that

$$\alpha_t(j) = p(z_t | s_t = j) \overline{\alpha_t(j)}. \tag{7}$$

Algorithm 1: Localization Process of RICH

```
Input: IMU readings: acc. $\{a_k\}_{k=1}^K$, gyro. $\{w_k\}_{k=1}^K$ and
       mag. $\{m_k\}_{k=1}^K$, RF readings $P$, historical location
       distribution $\alpha_{t-1}[1{:}n]$;
Output: Estimated vehicle location $(\hat{x}_t, \hat{y}_t)$;
$\rho_t \leftarrow$ speed_classifier($\{a_k\}_{k=1}^K$,$\{m_k\}_{k=1}^K$);
$\phi_t$, $l_t \leftarrow$ orientation_estimation($\{a_k\}_{k=1}^K$, $\{w_k\}_{k=1}^K$, $\{m_k\}_{k=1}^K$);
$r_t \leftarrow$ RF_localization($P$);
$u_t \leftarrow [\rho_t, \phi_t]$;
$z_t \leftarrow [r_t, l_t]$;
for $j = 1,2,3\ldots,n$ do
    $\overline{\alpha_t(j)} = \sum_i p(s_t = q_j | s_{t-1} = q_i, u_{t-1})\alpha_{t-1}(i)$;
    $\alpha_t(j) = p\big(z_t | s_t = q_j\big)\overline{\alpha_t(j)}$;
end
$(\hat{x}_t, \hat{y}_t) \leftarrow \frac{\sum_{\alpha_t(j)\in M} \alpha_t(j) q_j}{\sum_{\alpha_t(j)\in M} \alpha_t(j)}$;
return $(\hat{x}_t, \hat{y}_t)$
```

Given (7), we perform HMM fusion in two steps, namely, prediction and refinement.

1) *Prediction:* Make a prediction of the vehicle location $\overline{\alpha_t(j)}$ based on the historical distribution $\alpha_{t-1}(1), \alpha_{t-1}(2), \ldots, \alpha_{t-1}(n)$ and the action u_t. The term $p(s_t = j | s_{t-1} = i, u_t)$ in (5) is also known as the "transition model."
2) *Refinement:* Refine the predicted location $\overline{\alpha_t(j)}$ with sensor observations z_t. The term $p(z_t | s_t = j)$ in (7) is also known as the "observation model."

Once every $\alpha_t(j)$ is known, we can finally estimate the vehicle location by a weighted average of highest probabilities, i.e.,

$$\big(\hat{x}_t, \hat{y}_t\big) = \frac{\sum_{\alpha_t(j)\in M} \alpha_t(j) q_j}{\sum_{\alpha_t(j)\in M} \alpha_t(j)} \tag{8}$$

where M is the set of top-k joint probabilities in the set $\{\alpha_t(1), \alpha_t(2), \ldots, \alpha_t(n)\}$.

Putting all the sections above together, we summarize the online localization algorithm in Algorithm 1.

IV. Offline Speed Collection

Vehicle speed while driving in an indoor carpark forms a certain number of S speed patterns. Given a collection of vehicle speed in a period (e.g., 2 s), we consider the peak value of vehicle speed v_p. The speed pattern is labeled as $\rho_t = i, i \in [0, S-1]$, if

$$v_l(i) \le v_p \le v_h(i) \tag{9}$$

where $v_l(i)$ and $v_h(i)$ are the lower and upper bound speed values corresponding to the speed pattern i.

In offline stage, we define these speed patterns and collect speed data from real driving trajectories. The set of vehicle speed distribution F corresponding to all the speed patterns is saved to compute the transition matrix of the HMM. RICH associates the offline vehicle speed data with these S speed patterns, where S is empirically set. We show an example of $S = 3$ in Fig. 5.

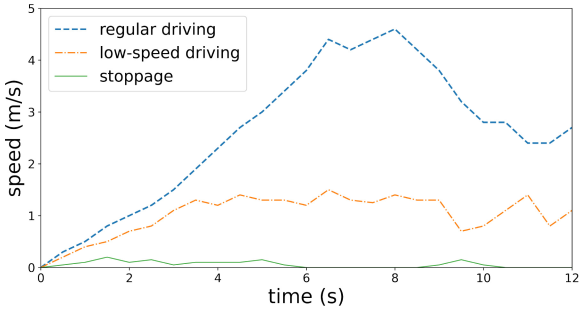


Fig. 5. Typical car speed patterns observed in an indoor carpark.

Based on the frequency of appearance, a speed pattern is defined as either a *regular* pattern or an *irregular* pattern. Vehicle speed distribution with a *regular* pattern is regarded as location dependant. This is because the carpark infrastructures affect driver's driving preference. For instance, a car typically speeds up to a certain speed level and slows down at the end of lane to make a turn. Therefore, the vehicle speed has some regularity and is location dependent. In this case, we collect the car speed at each grid point q_j separately and estimate the speed distribution of each point q_j accordingly. Vehicle speed distribution with the pattern i is collected as $F = \{f_i(q_1, v), f_i(q_2, v), \ldots, f_i(q_n, v)\}$, where $f_i(q_j, v)$ denotes the probability distribution of the vehicle speed with pattern i at the grid point q_j. On the contrary, vehicle speed with *irregular* patterns are regarded as location independent. This is because irregular events are often induced by a temporary changing of the environment (such as pedestrians and backing cars). Therefore, we estimate the vehicle speed distribution of irregular patterns with all samples regardless of the car's location, i.e.,

$$f_i(q_1, v) = f_i(q_2, v) = \cdots = f_i(q_n, v) = f_i(v). \tag{10}$$

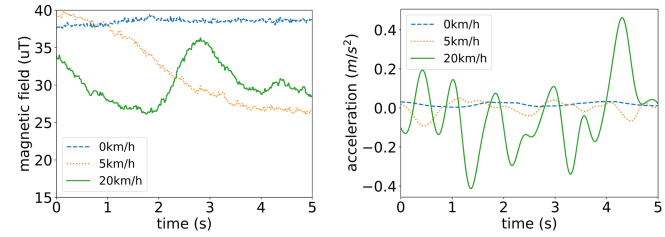
V. Online Signal Processing

In the signal processing module, raw IMU and RF signals are processed to extract vehicle actions u_t and location observations z_t for HMM fusion. In this section, we discuss the speed pattern classifier in Section V-A, heading estimation and turn detection in Section V-B, and RF localization in Section V-C.

A. Speed Pattern Classifier

Speed pattern classifier leverages the IMU signal readings to classify the vehicle speed pattern ρ_t. The key design motivation is that the linear acceleration and geo-magnetic field signals features vary with different speed patterns, as demonstrated in Fig. 6(a) and(b).

Based on the observations, we apply a 1-D convolutional neuron network (1D-CNN) model to extract IMU features for speed pattern classification shown in Fig. 7. We first use a second-order Butterworth filter to filter out the high-frequency noise in accelerometer readings. Afterward, the vehicle acceleration along the driving direction is estimated. With a sliding window of 1 s, a number of IMU readings are selected. After batch normalization and a dense layer, the geomagnetic field



(a) (b)

Fig. 6. IMU signals observed at different speed patterns. The variance of linear acceleration and geo-magnetic field is smaller in stoppage pattern and low-speed pattern. (a) Sensed geo-magnetic field with different speed patterns. (b) Filtered vehicle's linear acceleration with different speed patterns.

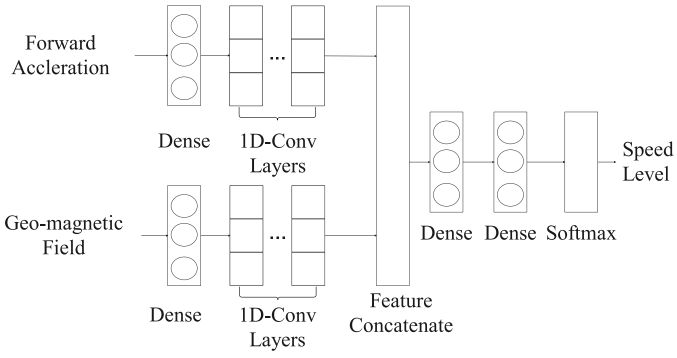


Fig. 7. Speed pattern classification model.

and forward acceleration are processed separately by 1D-CNN layers to capture motion features. Finally, the features are concatenated and processed with dense layers into speed level (pattern).

B. Heading Estimation and Turn Detection

Heading estimation detects a car's heading ϕ_t. In this article, we adopt an efficient algorithm Madgwick filter [23] to detect the vehicle's heading. Turn detection captures car turns to detect turn landmarks l_t. In RICH, turn detection is achieved by a simple threshold-based method, i.e.,

$$l_t = 1, \text{if} \quad \sum_i \omega_i \geq T_\omega \tag{11}$$

where $\sum_{i=1}^{K} \omega_i$ is the accumulation of vehicle angular velocity in a certain period and T_ω is a threshold set empirically.

C. RF Localization

RF localization estimates the vehicle's rough location $\hat{s}$. We can employ the typical RF localization methods to acquire $\hat{s}$ from the RSSI readings of RF emitters. Many RF localization methods [3], [8], [31], [35] can be adopted in our framework. In this article, we adopt WCL [3], a computationally efficient and fingerprint-free method. WCL estimates the location of a node by a weighted average of the coordinates of other RF emitters whose positions are inherently known, shown as

$$\hat{s} = \frac{\sum_j w_j M_j}{\sum_j w_j} \tag{12}$$

where $M_j = (x_j, y_j)$ denotes the coordinate of the jth RF emitter. The corresponding weight w_j is calculated as

$$w_j = \eta e^{\frac{P_j - P_0}{\Delta P}} \tag{13}$$

where P_j is the RSSI reading of the jth emitter. P_0, ΔP and η are some constants set empirically.

VI. Online HMM Fusion and Its Complexity

In this section, we present the HMM fusion model in terms of its prediction step (Section VI-A) and refinement step (Section VI-B). After that, we analyze its computation complexity in Section VI-D.

A. Prediction

The prediction step predicts a car's future location based on the past location distribution $\alpha_{t-1}(i)$ and the action u_t. The prediction step in (6) is modeled as

$$\begin{aligned}\overline{\alpha_t(j)} &= \sum_i p(s_t = j|\phi_t, \rho_t, s_{t-1} = i)\alpha_{t-1}(i) \\ &= \sum_i a_t(i,j) p(s_t = j|\phi_t, s_{t-1} = i)\alpha_{t-1}(i) \end{aligned} \tag{14}$$

where $a_t(i,j)$ is the transition probability estimated from the speed distribution F collected in the offline phase and the pattern ρ_t. We employ a kernel K to perform the task, given by

$$a_t(i,j) \propto \int_v K\left(\frac{v - v_{ij}}{\sigma_v}\right) f_{\rho_t}(v, q_i) dv \tag{15}$$

such that

$$\sum_j a_t(i,j) = 1 \tag{16}$$

where $f_{\rho_t}(v, q_i)$ is the speed distribution of the pattern ρ_t. K is a kernel function in which σ_v denotes the variance of the car speed distribution and, v_{ij} denotes the average speed required to drive across the states s_i and s_j within a unit time of Δt, i.e.,

$$v_{ij} = \frac{\|q_i - q_j\|}{\Delta t}. \tag{17}$$

There are many typical kernel functions that can be applied to the task. In our framework, we select a simple Gaussian kernel function, i.e.,

$$K(x) = e^{-\frac{1}{2}x^2}. \tag{18}$$

Usually, vehicle speed in an indoor car park is upper limited. We apply an H-hop constraint to reduce computation. We assume a vehicle is only capable of transferring into H hops neighbor states within a period Δt, i.e.,

$$a_t(i,j) = 0, \text{if} \quad \|q_i - q_j\| > Hd. \tag{19}$$

The term $p(s_t = j|\phi_t, s_{t-1} = i)$ represents the probability that a car has the heading ϕ when $s_t = j$ and $s_{t-1} = i$. We illustrate how such probability is estimated with Fig. 8. First, the observed vehicle's heading is expected to be close to the direction of vehicle's real transition. Let σ_ϕ represent the variance of orientation estimation error. We assume that

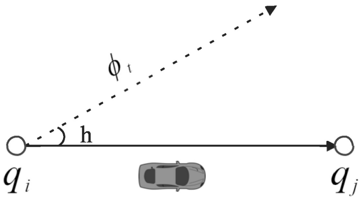


Fig. 8. h represents the angle between the direction of proposed transition ($s_{t-1} = q_i$ and $s_t = q_j$) and the observed vehicle's heading ϕ_t. A smaller h indicates a higher probability of proposed transition.

the orientation estimation error forms a Huber-loss distribution given by

$$\ln(L(x,\sigma)) = \begin{cases} \frac{1}{2}|x|^2 + 2\ln\sigma, & \text{if } |x| \le \sigma \\ \sigma\left(|x| - \frac{1}{2}\sigma\right) + 2\ln\sigma, & \text{if } |x| > \sigma. \end{cases} \tag{20}$$

Then, we estimate the probability with

$$p(s_t = j|\phi_t, s_{t-1} = i) = L\big(h(\phi, q_i, q_j), \sigma_\phi\big) \tag{21}$$

where the function h represents the angle between the direction of transition and the estimated vehicle direction, given by

$$\begin{aligned} h(\phi, q_i, q_j) &= h\big(\phi, (x_i, y_i), (x_j, y_j)\big) \\ &= \arccos\left(\frac{(x_j - x_i)\cos(\phi) + (y_j - y_i)\sin\big(\hat{\phi}\big)}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}\right). \end{aligned} \tag{22}$$

B. Refinement

We refine the predicted location with the RF localization result r_t and the turn landmark l_t. As l_t and r_t are conditionally independent with the state s_t known, the refinement step works as follows:

$$\alpha_t(j) = p(r_t|s_t = j)p(l_t|s_t = j)\overline{\alpha_t(j)}. \tag{23}$$

The term $p(r_t|s_t = q_j)$ represents the error distribution of the RF localization result. Knowing the variance of the RF localization error σ_s, we assume that the RF localization error forms a Huber-loss distribution, i.e.,

$$p(r_t|s_t = j) = L\big(\|r_t - q_j\|, \sigma_s\big). \tag{24}$$

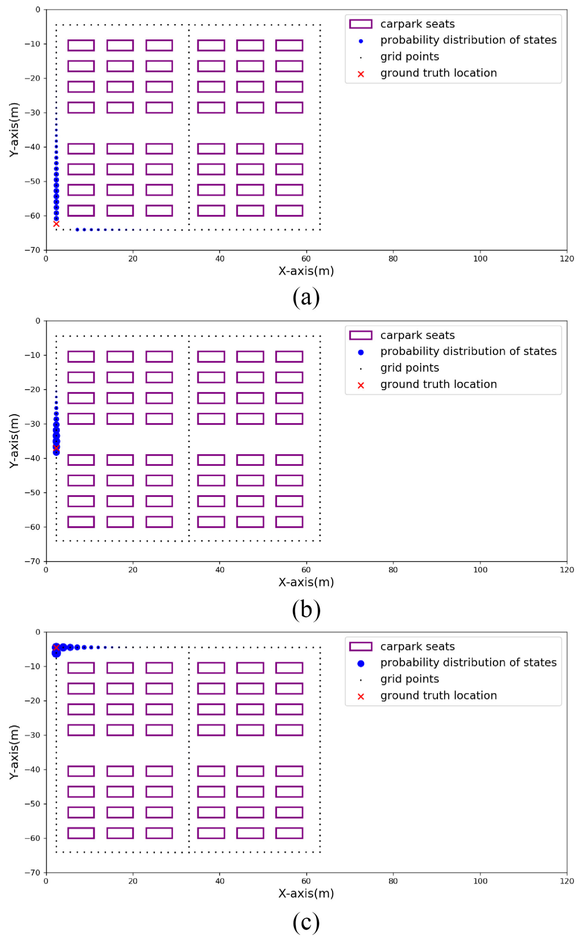
The term $p(l_t|s_t = j)$ represents the probability of observing a car turn l_t at j. We model the likelihood of observing a turn at the point q_j with the distribution

$$p(l_t|s_t = j) = g\big(q_j\big)^{l_t}\big(1 - g(q_j)\big)^{1-l_t}, \; l_t \in \{0, 1\} \tag{25}$$

where

$$g\big(q_j\big) = \begin{cases} \beta, & \text{if } q_j \text{ is a junction point} \\ 1 - \gamma, & \text{if } q_j \text{ is not a junction point} \end{cases} \tag{26}$$

where β and γ are the recall and precision of the turn detection algorithm.



(a)

(b)

(c)

Fig. 9. Visualized HMM's probability distribution. (a) Starting point. (b) When driving at middle points. (c) When negotiating a corner.

C. Visualized Demo

To elaborate more on the proposed HMM model, we visualize the inference process by showing the probability distributions of a car in Fig. 9(c). The red cross indicates the ground-truth location while the blue points show the probability distribution of the states. The size of a blue dot indicates the probability that a car is at the corresponding position. At the beginning, an RF packet and an IMU packet are received referring to a rough initial distribution with heading, as shown in Fig. 9(a). With more RF/IMU observations received, the distribution gradually converges and gets closer to the ground-truth location by iterating the prediction and refinement steps mentioned in Sections VI-A and VI-B, shown in Fig. 9(a). After negotiating a corner, the probability distribution converges to the corresponding corner point, shown in Fig. 9(b).

D. Computational Complexity

The computational complexity in online localization is

$$O(F) + O(Hn) \tag{27}$$

where F is the computation cost for signal processing, H is the number of hops, and n is the number of states.

We explain (27) as follows. The online localization consists of the signal processing stage and the HMM fusion stage. As

TABLE II
SPECIFICATION OF THE TWO CARPARKS

Car park name	University carpark	Apartment carpark
Area (m^2)	3600	9200
iBeacons deployed	19	35
Junctions	6	6

TABLE III
BASELINE PARAMETERS

Parameter	Value
d	1.2m
Δt	0.2s
H	5
S	3

the vehicle's transition probability a_{ij} with all speed patterns can be memorized in advance, the prediction step costs $O(Hn)$ computation. The refinement step costs $O(n)$ computation.

Note that if the vehicle speed is upper bounded by $V_{\max}$, the minimum H achieving the highest localization accuracy is chosen as

$$H = \left\lceil \frac{V_{\max}\Delta t}{d} \right\rceil \tag{28}$$

where Δt denotes the unit period to perform localization, d is the grid size, and $\lceil x \rceil$ denotes the ceiling of x. From (28), we have

$$H \propto d^{-1}. \tag{29}$$

Furthermore, the total number of states n satisfies

$$n \propto d^{-1} \tag{30}$$

because the total length of driveable paths in an indoor car park is fixed. Therefore, the computation complexity may be rewritten in terms of the grid size d as

$$O(F) + O\left(Cd^{-2}\right) \tag{31}$$

where C is a constant.

Grid size d is the critical hyperparameter to balance the computation complexity and localization accuracy. We study the computation-accuracy tradeoff through experiments in Section VII-F.

VII. ILLUSTRATIVE EXPERIMENT RESULTS

We have implemented RICH in smartphones and conducted extensive experiments in real carparks. In this section, we first discuss our experimental settings, performance metrics, and comparison schemes in Section VII-A. The speed classifier used in RICH is then evaluated in Section VII-B. We present the overall performance in Section VII-C. Then, we discuss how parking occupancy and AP density affects the localization in Sections VII-D and VII-E, respectively, and finally discuss the computation-accuracy tradeoff in Section VII-F.

A. Experiment Settings and Performance Metrics

The experiments are conducted in two typical indoor carparks, one in our university and the other one of a private apartment building. Both carparks are deployed with proper density of BLE beacons (iBeacons) with broadcasting interval of a second. Specifications of the two experiment fields are shown in Table II. Different brands of private cars are involved in the experiments, including Hyundai, BMW, Honda, and Nissan. Mobile phones involved in the experiments also vary in different types, including Samsung, Huawei, Vivo, and iPhone. Our system is implemented in both Android and iOS platforms with Dart language. We also implement the system on a 4-core i7-6560 personal workstation with Python language for evaluation purposes.

We conduct site surveys to collect the data for both the offline and online phases. A total of six volunteer drivers participated in the experiments. Drivers vary in driving age and gender. They drive in prescribed routes while the surveyor seated in the car collects data with a docked phone. We have collected a total of 276 min of driving trajectories in the two aforementioned car parks. The driving data covers various speed bands to cover most driving scenarios. Unless otherwise stated, we use the baseline parameters according to Table III. The parameters and settings in online signal processing step are specified in Table IV.

We developed an app for signal collection. The app collects the IMU signals at 50-Hz sampling frequency and updates the BLE readings at 5-Hz sampling frequency. Sensor (RF and IMU) data is automatically collected by the app. The car speed and the trajectories are annotated from recorded videos, frame by frame.

The performance metrics are as follows.

1) *Localization Error:* The localization error is defined as the distance between the estimated vehicle position and the ground truth. The overall performance is evaluated by the average localization error, final distance error, max distance error, and cumulative distribution function (CDF) error.
2) *Average Computation Time:* The average computation time is defined as the average time required to estimate one car location. Considering the computation power heterogeneity of various devices, we evaluated the average computation time on various devices, including a personal computer and various mobile phones.

Due to the uniqueness of our sensor settings, RICH is hard to compare directly with other works. We select the following comparable schemes.

1) *WCL [3]:* It is implemented as the baseline. It is also the RF localization algorithm adopted in the signal processing stage described in Section V.
2) *GMPF [25]:* It is a state-of-the-art approach which has similar sensor settings as RICH. GMPF applies ensemble WiFi fingerprinting to perform RF localization. Afterward, it acquires odometer information by fusing IMU and lidar. Finally, the WiFi and odometer readings are fused using a Gaussian-mixture PF. In our experiments, we replace the ensemble fingerprinting method with WCL. As lidar is unavailable in our sensor setting, we replace lidar-based odometer with the speed distribution learned in our offline training stage.

TABLE IV
ONLINE SIGNAL PROCESSING SETTINGS

Module	Specifications
speed pattern classifier	LPF cut-off frequency: 15HZ; Number of 1D-CNN layers: 3
turn detection	Turn detection threshold: $T_w = 10$
Madgwick Filter	Regression step: $\beta_r = 0.8$
WCL	$P0 = -60dB$,$\Delta P = 20$

TABLE V
ACCURACY OF SPEED CLASSIFIER

	Stop	Low Speed Driving	Regular Driving
Precision	0.87	0.91	0.76
Recall	0.85	0.79	0.92
F1-score	0.86	0.84	0.83

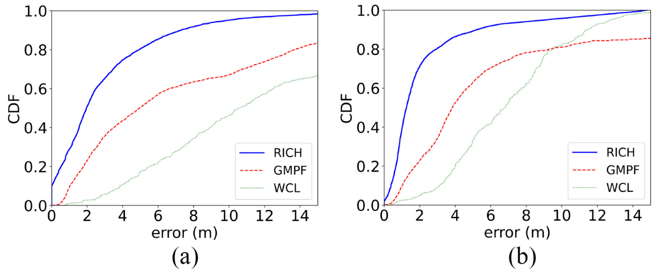


Fig. 10. CDF of different schemes. (a) CDF in University Carpark. (b) CDF in Apartment Carpark.

B. Speed Pattern Classifier

In our experiments, the number of speed patterns S is selected as 3: stoppage, low-speed driving, and regular driving. A training set containing 200-min length driving trajectories is applied to train the speed pattern classifier model. We apply tenfold cross-validation to validate the trained model.

Table V shows the accuracy of the speed pattern classifier. The average F1-score is around 85%. We observe a high recall in regular driving case because regular driving pattern appears most frequently in the training data.

C. Overall Performance

In Fig. 10(a) and (b), we show the CDF localization error in our two experiment fields. We observe that the pure RF-based WCL method has a limited localization accuracy, mainly due to the signal attenuation by the car body. Our scheme RICH significantly outperforms WCL and GMPF in localization error. We summarize the average computation time and average localization error of all schemes in Table VI. WCL consumes least the computation time, as both GMPF and RICH are implemented on top of it. Our RICH outperforms GMPF in both computation time and accuracy. Comparing the CDF curves in both sites, we observe RICH achieves better accuracy in University Carpark than Apartment Carpark. This is mainly because RF emitters in University Carpark is more densely deployed and therefore higher signal-to-noise ratio.

We observe at the bottom left of Fig. 10(a) and (b) that there is a proportion of the positioning error close to zero. Zero error happens when a car is correctly located at a turning point by turn detection. The cumulative density of 0 error is expected to be βr, where r is the ratio of the total turning period over the driving period and β is the recall of the turn detection algorithm. In our experiments, turning occupies 3.2% and 11.2% of the total driving period in the two fields, respectively; $\beta = 0.95$.

TABLE VI
AVERAGE COMPUTATION TIME ON 4-CORE I7-6560 CPU WORK STATION, PYTHON 3.7.1

Scheme	WCL	PF	RICH
Computation time (ms)	0.2	7.2	4.1
RMS error (m)	21.0	10.1	4.4

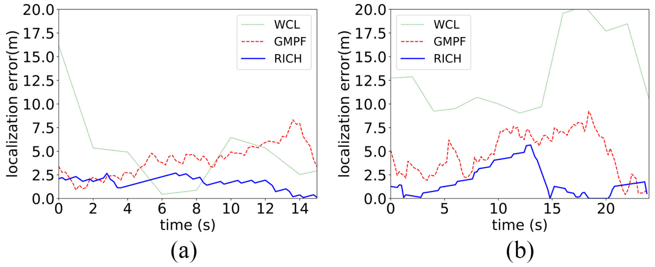


Fig. 11. Localization error over time under the (a) regular driving pattern and the (b) low-speed driving pattern.

TABLE VII
MAX DISTANCE ERROR

Scheme	WCL	PF	RICH
Max Distance Error(m)	15.22	17.62	6.61

We illustrate the superiority of RICH with temporal localization error plots. Fig. 11(a) and (b) shows the localization error over time at nominal vehicle speed and low-speed bands, respectively. We observe that both GMPF and RICH have reasonable localization accuracy at the nominal speed case. However, once the vehicle drops to a low speed for several seconds, the localization error of GMPF drifts to 10 m over time. RICH reduces the drift with the speed classifier and calibrates the location error with a detected turn. For the same reason, we observe that the final distance error shown in Fig. 12 and the maximum distance error shown in Table VII of RICH are also the lowest.

We also test the average computation time of RICH on various phones. As shown in Table VIII, the average computation periods on the mobile phones range from 0.9 to 8.2 ms, all below the minimum response time, i.e., $\Delta t = 0.2$ s. Therefore,

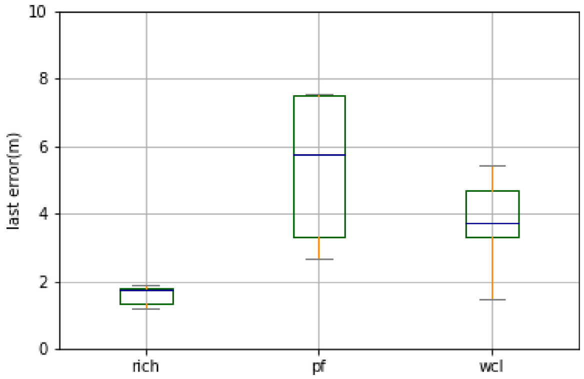


Fig. 12. Boxplot: final distance error of all schemes.

TABLE VIII
AVERAGE COMPUTATION TIME OF RICH ON VARIOUS PHONES

Phone type	Sumsung S8	Vivo Y12	iPhone 11
Computation time (ms)	8.2	4.2	0.9

TABLE IX
LOCALIZATION ERROR WITH DIFFERENT PARKING OCCUPANCY

Occupancy	Localization Error(m)	Heading Error(degree)
30%	3.3	17.6
70%	3.62	34.2

RICH is light-weight enough to deploy on a typical mobile phone.

D. Parking Occupancy

As the vehicle body is usually made of metal, the parking occupancy rate can affect the magnetic field in the parking lot. Therefore, we tested the positioning accuracy at different parking occupancy rate (approximately 30%, 70%). The results are shown in Table IX. We observe no significant change of average localization accuracy, but a double of heading estimation error. Therefore, the localization error is relatively robust to the parking occupancy.

E. AP Density

The deployed AP density may diverse in different carparks. We evaluate the influence of AP density on the localization error. Here, the AP density ratio is measured by the total number of APs participated in localization divided by the total of APs deployed in the carpark. As shown in Fig. 13, all schemes (WCL, GMPF, RICH) increase their accuracy with the increase of AP density. However, RICH is comparatively more robust to the AP density because of a better design of the turn detection, speed pattern classifier, and the motion constraints implied in the HMM model.

F. Computation-Accuracy Tradeoff

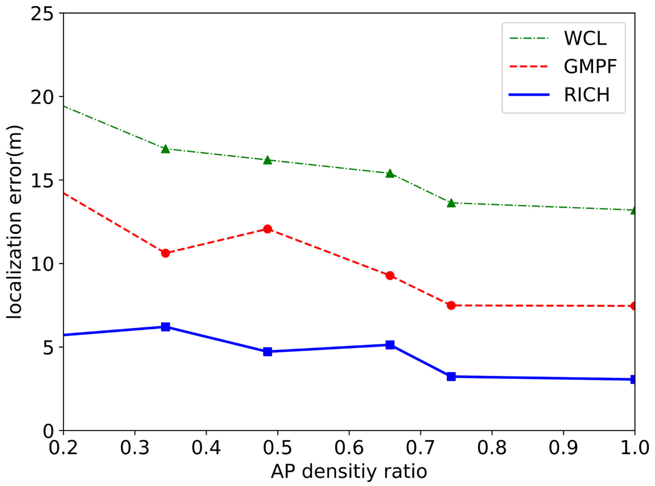
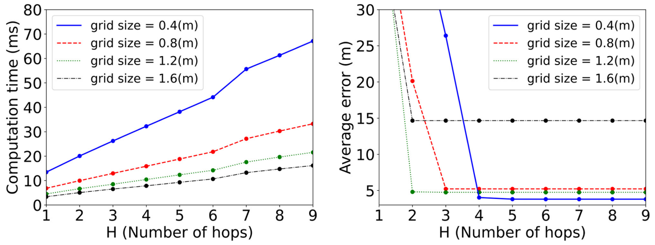


Fig. 13. Average localization error with different AP density ratios, where ratio = 1 means all APs deployed in the carpark are utilized in localization.



(a) (b)

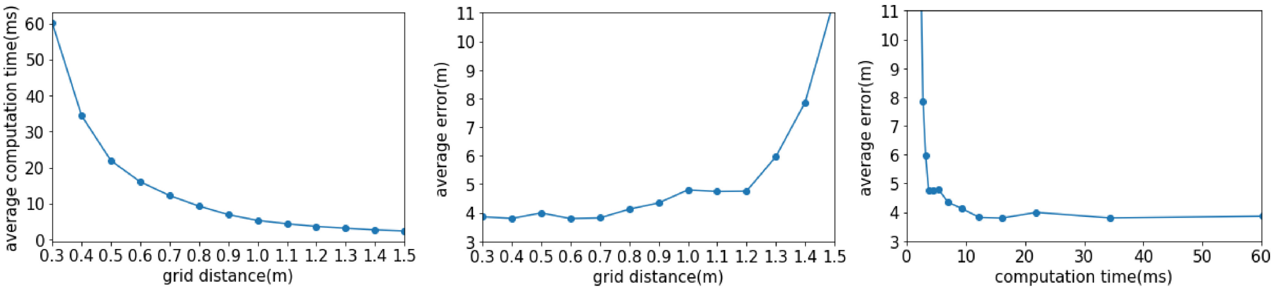
Fig. 14. System performance with variation of parameters H and d. (a) Average computation time under different d and H settings. (b) Average localization error under different d and H settings.

We show in Fig. 14(a) the average computation time versus the number of hops H under a variation settings of grid size d. The average computation time increases linearly with H. The computation cost increases as the grid size d decreases. We show in Fig. 14(b) the average localization error versus the number of Hops H under a variation settings of grid size d. Generally, the localization error decreases when the grid size is smaller. The localization error drops significantly as H increases until the accuracy achieves its optimal given by (28). In our experiments, the maximum vehicle speed $V_{\max}$ is 9 m/s. We perform localization every 0.2 s, i.e., $\Delta t = 0.2$ s.

We study the computation-accuracy tradeoff of RICH as follows. We observe how the computation time and the localization error change with the grid size d under the optimal setting of H determined by (28). As shown in Fig. 15(a), the average computation time decreases as the grid size d increases. Moreover, the average computation time is approximately inverse proportional to the square of d. Fig. 15(a) fits well with (31). Fig. 15(b) shows how average error changes with the grid size d. In general, localization error increase as the grid size increase. We observe a significant decrease of localization error when d is small (at d = 1.2m). This is because a smaller d means finer granularity such that the transition probability can be estimated more accurately. We show in Fig. 15(c) the computation-accuracy tradeoff curve. The average localization error decreases and converges to a minimum as we afford more computation time.

VIII. CONCLUSION

We study the problem of navigating in an indoor carpark under the general case of weak or no GNSS and cellular



(a) (b) (c)

Fig. 15. Performance of RICH under different grid size settings. H is set to be optimal in (28). (a) Average computation time over the grid side d. (b) Average localization error over the grid side d. (c) Tradeoff between the computation time and the localization accuracy.

signals. To address it, we propose RICH, a novel, simple, accurate, and cost-effective approach for an offline docked phone without costly OCAI installation and error-prone IMU integration. Using IMU signals, RICH classifies the car speeds and detects the car heading and turning. This information and the crude RF localization are then fused with an HMM to accurately compute the car location.

We analyze the computational complexity of RICH and its tradeoff with accuracy. We have implemented RICH in smartphones and conducted extensive experiments in two carparks in university and apartment complex. RICH achieves significantly lower (by more than 40%) localization error as compared with the state-of-the-art approaches. It is also computationally light-weight, deployable real-time in offline smartphones.

References

[1] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” in *Proc. IEEE 9th Annu. Joint Conf. Comput. Commun. Soc. (INFOCOM)*, vol. 2, 2000, pp. 775–784.

[2] D. Becker, F. Thiele, O. Sawade, and I. Radusch, “Cost-effective camera based ground truth for indoor localization,” in *Proc. IEEE Int. Conf. Adv. Intell. Mechatron. (AIM)*, Jul. 2015, pp. 885–890.

[3] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, “Weighted centroid localization in ZigBee-based sensor networks,” in *Proc. IEEE Int. Symp. Intell. Signal Process.*, Oct. 2007, pp. 1–6.

[4] M. Brossard, A. Barrau, and S. Bonnabel, “AI-IMU dead-reckoning,” *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 585–595, Dec. 2020.

[5] C. Chen et al., “Motiontransformer: Transferring neural inertial tracking between domains,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 8009–8016.

[6] C. Chen, C. Lee, and C. Lo, “Vehicle localization and velocity estimation based on mobile phone sensing,” *IEEE Access*, vol. 4, pp. 803–817, 2016.

[7] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, “Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization,” *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.

[8] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, “Indoor localization without the pain,” in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2020, pp. 173–184.

[9] J. Einsiedler, D. Becker, and I. Radusch, “External visual positioning system for enclosed carparks,” in *Proc. 11th Workshop Position. Navig. Commun. WPNC)*, Mar. 2014, pp. 1–6.

[10] R. Elbakly and M. Youssef, “A robust zero-calibration RF-based localization system for realistic environments,” in *Proc. 13th Annu. IEEE Int. Conf. Sens. Commun., Netw. (SECON)*, Jun. 2016, pp. 1–9.

[11] R. Gao et al., “Glow in the dark: Smartphone inertial odometry for vehicle tracking in GPS blocked environments,” *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12955–12967, Aug. 2021.

[12] Y. Gao et al., “XINS: The anatomy of an indoor positioning and navigation architecture,” in *Proc. 1st Int. Workshop Mobile Location Based Service (MLBS)*, 2011, pp. 41–50.

[13] K. Golestan, S. Seifzadeh, M. Kamel, F. Karray, and F. Sattar, “Vehicle localization in VANETs using data fusion and V2V communication,” in *Proc. 2nd ACM Int. Symp. Design Anal. Intell. Veh. Netw. Appl. (DIVANet)*, 2012, pp. 123–130.

[14] F. B. Günay, E. Öztürk, T. G. Çavdar, Y. S. Hanay, and A. U. R. Khan, “Vehicular ad hoc network (VANET) localization techniques: A survey,” *Arch. Comput. Methods Eng.*, vol. 28, pp. 3001–3033, Sep. 2020.

[15] F. Gustafsson, “Particle filter theory and practice with positioning applications,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 53–82, Jul. 2010.

[16] S. He, S.-H. Gary Chan, L. Yu, and N. Liu, “SLAC: Calibration-free pedometer-fingerprint fusion for indoor localization,” *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 1176–1189, May 2018.

[17] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2D LiDAR SLAM,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1271–1278.

[18] J. Jiao, H. Ye, Y. Zhu, and M. Liu, “Robust odometry and mapping for multi-LiDAR systems with online extrinsic calibration,” *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 351–371, Feb. 2022.

[19] W. Kang and Y. Han, “SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization,” *IEEE Sensors J.*, vol. 15, no. 5, pp. 2906–2916, May 2015.

[20] E. J. Krakiwsky, C. B. Harris, and R. V. C. Wong, “A Kalman filter for integrating dead reckoning, map matching and GPS positioning,” in *Proc. Position Location Navig. Symp. Rec. Navig. (IEEE PLANS)*, 1988, pp. 39–46.

[21] A. K. T. R. Kumar, B. Schäufele, D. Becker, O. Sawade, and I. Radusch, “Indoor localization of vehicles using deep learning,” in *Proc. IEEE 17th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, Jun. 2016, pp. 1–6.

[22] N. Li, Y. Gao, Y. Wang, Z. Liu, L. Guan, and X. Liu, “A low-cost underground garage navigation switching algorithm based on Kalman filtering,” *Sensors*, vol. 19, no. 8, p. 1861, 2019.

[23] S. O. H. Madgwick, *An Efficient Orientation Filter for Inertial and Inertial/Magnetic Sensor Arrays*, Univ. Bristol, Bristol, U.K., 2010.

[24] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, “Centaur: Locating devices in an office environment,” in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. Mobicom*, 2012, p. 281.

[25] D. Nguyen, T. Dao, E. Castelli, and F. Nashashibi, “A fusion method for localization of intelligent vehicles in carparks,” *IEEE Access*, vol. 8, pp. 99729–99739, 2020.

[26] X. Niu, Y. Wu, and J. Kuang, “Wheel-INS: A wheel-mounted MEMS IMU-based dead reckoning system,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 9814–9825, Oct. 2021.

[27] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, “Particle filter and smoother for indoor localization,” in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, 2013, pp. 1–10.

[28] T. Qin, T. Chen, Y. Chen, and Q. Su, “AVP-SLAM: Semantic visual mapping and localization for autonomous vehicles in the parking lot,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5939–5945.

[29] H. Sang, R. Jiang, Z. Wang, Y. Zhou, and B. He, “A novel neural multi-store memory network for autonomous visual navigation in unknown environment,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2039–2046, Apr. 2022.

[30] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar, “Broadcasting in VANET,” in *Proc. Mobile Netw. Veh. Environ.*, May 2007, pp. 7–12.

[31] J. Wang, P. Urriza, Y. Han, and D. Cabric, "Weighted centroid localization algorithm: Theoretical analysis and distributed implementation," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3403–3413, Oct. 2011.

[32] Y. Wu, X. Niu, and J. Kuang, "A comparison of three measurement models for the wheel-mounted MEMS IMU-based dead reckoning system," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11193–11203, Nov. 2021.

[33] J. Xiong and K. Jamieson, "ArrayTrack: A fine-grained indoor location system," in *Proc. 10th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2013, pp. 71–84.

[34] H. Yan, Q. Shan, and Y. Furukawa, "RIDI: Robust IMU double integration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 621–636.

[35] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, 2005, pp. 205–218.

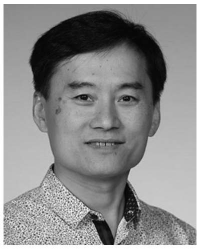
[36] C. Yu, N. El-Sheimy, H. Lan, and Z. Liu, "Map-based indoor pedestrian navigation using an auxiliary particle filter," *Micromachines*, vol. 8, no. 7, p. 225, Jul. 2017.

Murphy Zheng Zhang received the M.Phil. degree in computer science from The Hong Kong University of Science and Technology, Hong Kong, in 2021.

He is currently a Data Mining Engineer with Tencent, Shenzhen, China. His research interests include indoor localization, data mining, and natural language processing.

Gary Wing-Ho Cheung is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong.

His research focuses on developing innovative techniques and algorithms to enable accurate and reliable positioning of objects and individuals within indoor environments.



S.-H. Gary Chan (Senior Member, IEEE) received the B.S.E. degree (Highest Hons.) in electrical engineering from Princeton University, Princeton, NJ, USA, in June 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems, and the M.S.E. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, June 1994 and January 1999, respectively, with a Minor in business administration.

He is currently Professor with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong. He is also an Affiliate Professor of Innovation, Policy and Entrepreneurship Thrust with HKUST (Guangzhou), Guangzhou, China, and a Board Director of Hong Kong Logistics and Supply Chain MultiTech R&D Center, Hong Kong. He was a Visiting Professor or a Researcher of Microsoft Research with Princeton University, Stanford University, and University of California at Davis, Davis, CA, USA. At HKUST, he was the Director of Entrepreneurship Center, Sino Software Research Institute, and Computer Engineering Program and the Co-Director of Risk Management and Business Intelligence Program. He was a William and Leila Fellow of Stanford University. His research interests include smart IoT and sensing systems, location AI and mobile computing, video/user/data analytics, cloud and edge AI, technology transfer, and IT entrepreneurship.

Prof. Chan is the recipient of the Google Mobile 2014 Award and Silver Award of Boeing Research and Technology. He was the recipient of the Charles Ira Young Memorial Tablet and Medal and the POEM Newport Award of Excellence at Princeton University. Through technology transfer and entrepreneurship, he has successfully transferred and deployed his research results in industry and co-founded several startups. Due to their innovations, commercial, and societal impacts, his technologies have received numerous local and international awards. Notably, he received the Hong Kong Chief Executive's Commendation for Community Service for "Outstanding Contribution to the Fight Against COVID-19" in 2020. He has been an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA and a Vice-Chair of Peer-to-Peer Networking and Communications Technical Sub-Committee of IEEE Comsoc Emerging Technologies Committee. He has been a Guest Editor of *ACM Transactions on Multimedia Computing, Communications and Applications*, IEEE TRANSACTIONS ON MULTIMEDIA, *IEEE Signal Processing Magazine*, and *IEEE Communication Magazine*. He is a Steering Committee Member. He was the TPC Chair of IEEE Consumer Communications and Networking Conference and has been an Area Chair of the Multimedia Symposium of IEEE Globecom and IEEE ICC for many years. He is Fellow of Sigma Xi and a Chartered Fellow of The Chartered Institute of Logistics and Transport.