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

Murphy Z. Zhang, Gary W.-H. Cheung, S.-H. Gary Chan

Abstract—We study the challenging problem of navigating in an indoor carpark in the absence of GNSS (Global Navigation Satellite System) and cellular signals using an offline smartphone docked at the car dashboard. There is basic RF (radio-frequency) infrastructure in the venue but, due to signal attenuation by the car body, the location computed based on in-car signal is noisy and intermittent. Previous works on carpark navigation often require costly specialized equipment as on-car additional infrastructure (OCAIs), or suffer from error propagation stemmed from integrating IMU (inertial measurement unit) 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 HMM (Hidden Markov Model). 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 trade-off 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 computationally efficiency (less than 10ms per location).

Index Terms—fusion localization, carpark navigation

I. INTRODUCTION

Urbanization has led to increasing number of indoor carparks. In these carparks, 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, pre-installed 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 indoor carpark.

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 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, compu-
tation power and 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 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 dead-reckoning [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 HMM (Hidden Markov Model). 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
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 learnt by a offline training process proposed in
Section III-A. To represent similar constraints, a PF-based
method need to apply mass constrain-rules on a sufficient large
number of particles in the prediction and re-sampling step,
which spends more computation but perform worse.

Time is slotted in RICH with size in the order of a fraction
of a second (0.2 second 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. 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 paper are summarized as follows:
- **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.
- **Computational analysis of RICH**: We provide an analysis
on the trade-off between computation overhead and loca-
zation accuracy. Accordingly, users can adjust system
parameters for different mobile phones to balance among
phone computation time and accuracy in a carpark.
- **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 40% reduction in localization error and
60% reduction in computation time as compared with
other schemes.

The remainder of this paper 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: OCAI-based
indoor vehicle localization (Section II-A), RF-based localiza-
tion (Section II-B) and 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
(VANET) [13], [14], [30], whose basic idea is to perform
cooperative localization through vehicle-to-vehicle communi-
cation. With the rapid development of SLAM (simultaneous
localization and mapping), more works have been proposed
locating vehicles with wheel odrometers [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 lidar odometers. These SLAM-based methods are shown to achieve centimetre level localization accuracy. Another approach such as [25] proposes a fusion of WiFi, IMU and lidar applying a Gaussian-mixture particle filter model.

OCAl-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 sub-meter-level accuracy. Besides FP approach, several works such as 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 in-car signal is not strong. While 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

Probability-based fusion localization method fuses multiple sensor or localization system measurements based on probability. Kalman filtering and its variants, i.e., EKF and UKF are common methods studied and applied to localization. These works [7], [20], [22] adopt external signals such as visual, GNSS or RF signals to obtain a rough position and apply DR (Dead-Reckoning) 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 Dead Reckoning (DR). Moreover, the in-car RF signals are noisy and intermittent, leading to large and non-Gaussian observation noise. Therefore, Kalman filter is not applicable to the scenario.

To address the nonlinear filtering problem, particle filter (PF) [15] was introduced. Early pedestrian localization works [12], [19], [27] fuses external signals with PDR by particle filtering, achieving around 3m 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 particle filter 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-B, and III-C, respectively. We summarize the important symbols used in this work in Table I.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_t$</td>
<td>Total number of grid points (possible car states)</td>
</tr>
<tr>
<td>$s_t$</td>
<td>The car state at time (slot) $t$ ($s_t = 1, 2, \ldots, n_t$)</td>
</tr>
<tr>
<td>$q_{ij}$</td>
<td>2-D coordinate for grid point $j$, $j = 1, 2, 3, \ldots, n$</td>
</tr>
<tr>
<td>$F$</td>
<td>Car speed distribution</td>
</tr>
<tr>
<td>$S$</td>
<td>Total number of speed patterns</td>
</tr>
<tr>
<td>$p_t$</td>
<td>Vehicle speed pattern at time $t$</td>
</tr>
<tr>
<td>$o_t$</td>
<td>Vehicle heading at time $t$</td>
</tr>
<tr>
<td>$u_t$</td>
<td>Variance of heading error</td>
</tr>
<tr>
<td>$s_0$</td>
<td>Vehicle action at time $t$</td>
</tr>
<tr>
<td>$r_t$</td>
<td>2-D coordinate of RF localization result</td>
</tr>
<tr>
<td>$\sigma_r$</td>
<td>Variance of RF localization error</td>
</tr>
<tr>
<td>$l_t$</td>
<td>Boolean variable of whether the car is turning at time $t$</td>
</tr>
<tr>
<td>$z_t = [r_t, l_t]$</td>
<td>Observation vector at time $t$</td>
</tr>
<tr>
<td>$\gamma, \beta$</td>
<td>Precision and recall, respectively, for turn detection</td>
</tr>
<tr>
<td>$\alpha_t(s, j)$</td>
<td>Probability that a car is at grid point $j$ at time $t$</td>
</tr>
<tr>
<td>$H$</td>
<td>Maximum number of hops for a car transition in a slot</td>
</tr>
<tr>
<td>$v_{ma}$</td>
<td>Speed limit in the carpark</td>
</tr>
<tr>
<td>$d$</td>
<td>Distance between adjacent grid points (meters)</td>
</tr>
<tr>
<td>$K$</td>
<td>Kernal function for car transition probability</td>
</tr>
<tr>
<td>$\alpha_t(i, j)$</td>
<td>Transition probability for a car’s transition from $i$ to $j$ at time $t$.</td>
</tr>
<tr>
<td>$L$</td>
<td>Huber-loss function for fusion localization</td>
</tr>
</tbody>
</table>

A. Offline Training Phase

In the offline training phase, 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 hours of site survey to set up one typical carpark.

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

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 seconds. 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:

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 \( \rho_t \), while heading estimation detects the vehicle’s heading \( \phi_t \) 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, \( \rho_t, \phi_t, l_t \) 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.

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 \( z \). Let \( s_t \) be the vehicle state, given by any one of the grid points, at time \( t \). The vehicle’s action, defined as

\[
u_t = [\rho_t, \phi_t],\]

changes vehicle state from \( s_{t-1} \) to \( s_t \). Recall that \( \rho_t \) and \( \phi_t \) 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

\[z_t = [r_t, l_t].\]

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

\[p(s_t = j | u_{1:t}, z_{1:t}),\]

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

\[p(s_t = j | u_{1:t}, z_{1:t}) = \frac{p(s_t = j, u_{1:t}, z_{1:t})}{p(u_{1:t}, z_{1:t})} \propto p(s_t = j, u_{1:t}, z_{1:t}) = \alpha_t(j)\]

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

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

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

\[\alpha_t(j) = \sum_i p(s_t = j | s_{t-1} = i, u_t) \alpha_{t-1}(i)\]

such that

\[\alpha_t(j) = p(z_t | s_t = j) \overline{\alpha_t(j)}.\]

Given Equation (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 Equation (5) is also known as the "transition model".

2) **Refinement:** Refine the predicted location \( \alpha_t(j) \) with sensor observations \( z_t \). The term \( p(z_t|s_t = j) \) in Equation (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.,

\[
(\hat{x}_t, \hat{y}_t) = \frac{\sum_{\alpha_t(j) \in M} \alpha_t(j) q_j}{\sum_{\alpha_t(j) \in M} \alpha_t(j),}
\]

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

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

**Algorithm 1:** Localization process of RICH.

**Input:** IMU readings: \{a_k\}_{k=1}^K, \{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 \((\hat{x}_t, \hat{y}_t)\);

1. \( \rho_t \leftarrow \text{speed_classifier}(\{a_k\}_{k=1}^K, \{m_k\}_{k=1}^K); \)
2. \( \phi_t, l_t \leftarrow \text{orientation_estimation}(\{a_k\}_{k=1}^K, \{w_k\}_{k=1}^K, \{m_k\}_{k=1}^K); \)
3. \( r_t \leftarrow \text{RF_localization}(P); \)
4. \( \mu_t \leftarrow [\rho_t, \phi_t]; \)
5. \( z_t \leftarrow [r_t, l_t]; \)
6. **for** \( j = 1, 2, 3, ..., n \) **do**
7. \( \alpha_t(j) = \sum_{q_s \in \mathcal{S}} p(s_t = q_s|s_{t-1} = q_{t-1}, u_{t-1}) \alpha_{t-1}(i); \)
8. \( \alpha_t(j) = p(z_t|s_t = q_j) \alpha_t(j); \)
9. **end**
10. \( (\hat{x}_t, \hat{y}_t) \leftarrow \frac{\sum_{\alpha_t(j) \in M} \alpha_t(j) q_j}{\sum_{\alpha_t(j) \in M} \alpha_t(j)}, \)
11. **return** \((\hat{x}_t, \hat{y}_t)\);

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

Vehicle speed distribution with a regular pattern is regarded as location dependent. This is because the carpark infrastructure affects 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 \( F = \{f_i(q_1, v), f_i(q_2, v), ..., f_i(q_n, v)\} \), 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 estimates the vehicle speed distribution of irregular patterns with all samples regardless of the car’s location, i.e.,

\[
f_i(q_1, v) = f_i(q_2, v) = ... = f_i(q_n, v) = f_i(v).
\]

**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 \( \rho_t \). The key design motivation is that the linear acceleration and geo-magnetic field signals features vary with different speed patterns, as demonstrated in Figures 6a and 6b.

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

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on August 08,2023 at 07:45:15 UTC from IEEE Xplore. Restrictions apply.

WCL estimates the location of a node by a weighted average of the RSSI readings of RF emitters. Many RF localization methods [3], [8], [31], [35] can be adopted in our framework. In this paper, we adopt weighted centroid localization (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

$$\hat{s} = \frac{\sum_j w_j M_j}{\sum_j w_j},$$

where \(M_j = (x_j, y_j)\) denotes the coordinate of \(j\)-th RF emitter. The corresponding weight \(w_j\) is calculated as

$$w_j = \frac{P_j - P_0}{\Delta P},$$

where \(P_j\) is the RSSI reading of \(j\)-th emitter. \(P_0\), \(\Delta P\) and \(\eta\) are some constants set empirically.

VI. ONLINE HMM FUSION AND ITS COMPLEXITY

In this section we present 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 Equation (6) is modelled as

$$\bar{\alpha}_t(j) = \sum_i p(s_t = j|\phi_t, \rho_t, \omega_{t-1} = i) \alpha_{t-1}(i)$$

$$= \sum_i a_t(i,j) p(s_t = j|\phi_t, \omega_{t-1} = i) \alpha_{t-1}(i),$$

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

$$a_t(i,j) \propto \int \frac{\nu - v_{ij}}{\sigma_v} f_{\rho_t}(v, q_t) dv,$$

such that

$$\sum_j a_t(i, j) = 1,$$

where \(f_{\rho_t}(v, q_t)\) is the speed distribution of the pattern \(\rho_t\). \(K\) is a kernel function in which \(\sigma_v\) 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 \(\Delta t\), i.e.,

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

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.,

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

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 neighbour states within a period \(\Delta t\), i.e.,

$$a_t(i, j) = 0, \text{ if } ||q_i - q_j|| > H d.$$
The term \( p(s_t = j|\phi_t, s_{t-1} = i) \) represents the probability that a car has the heading \( \phi \) when \( s_t = j \) and \( s_{t-1} = i \). We illustrate how such probability is estimated with Figure 8. First, the observed vehicle's heading is expected to be close to the direction of vehicle's real transition. Let \( \sigma_\phi \) represent the variance of orientation estimation error. We assume that the orientation estimation error forms a Huber-loss distribution given by

\[
\ln(L(x, \sigma)) = \begin{cases} 
\frac{1}{2}|x|^2 + 2 \ln \sigma, & \text{if } |x| \leq \sigma; \\
\sigma \left(|x| - \frac{1}{2}\sigma \right) + 2 \ln \sigma, & \text{if } |x| > \sigma.
\end{cases}
\]  

(20)

Then we estimate the probability with

\[
p(s_t = j|\phi_t, s_{t-1} = i) = L(h(\phi, q_i, q_j), \sigma_\phi),
\]

(21)

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

\[
h(\phi, q_i, q_j) = h(\phi, (x_i, y_i), (x_j, y_j)) = \arccos \left( \frac{(x_j - x_i) \cos(\phi) + (y_j - y_i) \sin(\phi)}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \right).
\]

(22)

\[\text{B. Refinement}\]

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:

\[
\alpha_t(j) = p(r_t|s_t = j)p(l_t|s_t = j)\alpha_t(j).
\]

(23)

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

\[
p(r_t|s_t = j) = L(\|r_t - q_j\|, \sigma_r).
\]

(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 below:

\[
p(l_t|s_t = j) = g(q_j)^{l_t}(1 - g(q_j))^{1-l_t}, l_t \in \{0, 1\},
\]

(25)

where

\[
g(q_j) = \begin{cases} 
\beta, & \text{if } q_j \text{ is a junction point;} \\
1 - \gamma, & \text{if } q_j \text{ is not a junction point.}
\end{cases}
\]

(26)

where \( \beta \) and \( \gamma \) are the recall and precision of the turn detection algorithm.

\[\text{C. A Visualized Demo}\]

To elaborate more the proposed HMM model, we visualize the inference process by showing the probability distributions of a car in Figure 9. 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 Figure 9a. 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 Section VI-A and Section VI-B, shown in Figure 9b. After negotiating a corner, the probability distribution converges to the corresponding corner point, shown in Figure 9c.

\[\text{D. Computational Complexity}\]

The computational complexity in online localization is

\[O(F) + O(Hn),\]

(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 Equation (27) as follows. The online localization consists of the signal processing stage and the HMM fusion stage. As 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 \( V_{\text{max}} \), the minimum \( H \) achieving the highest localization accuracy is chosen as

\[
H = \left\lceil \frac{V_{\text{max}} \Delta t}{d} \right\rceil, \quad (28)
\]

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

\[
H \propto d^{-1}. \quad (29)
\]

Furthermore, the total number of states \( n \) satisfies

\[
n \propto d^{-1} \quad (30)
\]

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

\[
O(F) + O(Cd^{-2}), \quad (31)
\]

where \( C \) is a constant.

Grid size \( d \) is the critical hyper-parameter to balance the computation complexity and localization accuracy. We study the computation-accuracy trade-off through experiments in Section VII-F.

VII. ILLUSTRATIVE EXPERIMENT RESULTS

We have implemented RICH in smartphones and conducted extensive experiments in real car parks. 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 trade-off in Section VII-F.

A. Experiment Settings & Performance Metrics

The experiments are conducted in two typical indoor car parks, one in our university and the other one of a private apartment building. Both car parks are deployed with proper density of Bluetooth low energy 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 purpose.

We conduct site surveys to collect the data for both the offline and online phases. A total of 6 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 minutes 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 50Hz sampling frequency and updates the BLE readings at 5Hz 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:

- **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.
- **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:

- **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.
- **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. Afterwards, it acquires odometer information by fusing IMU and lidar. Finally, the WiFi and odometer readings are fused using a Gaussian-mixture particle filter. In
TABLE IV: Online signal processing settings.

<table>
<thead>
<tr>
<th>Module Specifications</th>
<th>Speed pattern classifier: LPF cut-off frequency: 15Hz; Number of 1D-CNN layers: 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn detection</td>
<td>Turn detection threshold: ( T_w = 10 )</td>
</tr>
<tr>
<td>Madgwick Filter</td>
<td>Regression step: ( \beta_r = 0.8 ).</td>
</tr>
<tr>
<td>WCL</td>
<td>( P_0 = -60dB, \Delta P = 20 )</td>
</tr>
</tbody>
</table>

TABLE V: Accuracy of speed classifier

<table>
<thead>
<tr>
<th></th>
<th>Stop</th>
<th>Low Speed Driving</th>
<th>Regular Driving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.87</td>
<td>0.91</td>
<td>0.76</td>
</tr>
<tr>
<td>Recall</td>
<td>0.85</td>
<td>0.79</td>
<td>0.92</td>
</tr>
<tr>
<td>F1-score</td>
<td>0.86</td>
<td>0.84</td>
<td>0.83</td>
</tr>
</tbody>
</table>

TABLE VI: Average computation time on 4-core i7-6560 CPU work station, Python 3.7.1.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>WCL</th>
<th>PF</th>
<th>RICH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation time (ms)</td>
<td>0.2</td>
<td>7.2</td>
<td>4.1</td>
</tr>
<tr>
<td>RMS error (m)</td>
<td>21.0</td>
<td>10.1</td>
<td>4.4</td>
</tr>
</tbody>
</table>

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 learnt in our offline training stage.

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 minutes length driving trajectories is applied to train the speed pattern classifier model. We apply 10-fold 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 figures 10a and 10b, 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 figures 10a and 10b 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 \( \beta r \), where \( r \) is the ratio of the total turning period over the driving period and \( \beta \) is the recall of turn detection algorithm. In our experiments, turning occupies 3.2% and 11.2% of the total driving period in the two fields respectively; \( \beta = 0.95 \).

We illustrate the superiority of RICH with temporal localization error plots. Figures 11a and 11b show 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 10m 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 Figure 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

<table>
<thead>
<tr>
<th>Scheme</th>
<th>WCL</th>
<th>PF</th>
<th>RICH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Distance Error (m)</td>
<td>15.22</td>
<td>17.62</td>
<td>6.61</td>
</tr>
</tbody>
</table>
TABLE VIII: Average computation time of RICH on various phones.

<table>
<thead>
<tr>
<th>Phone type</th>
<th>Computation time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung S8</td>
<td>8.2</td>
</tr>
<tr>
<td>Vivo Y12</td>
<td>4.2</td>
</tr>
<tr>
<td>iPhone 11</td>
<td>0.9</td>
</tr>
</tbody>
</table>

periods on the mobile phones range from 0.9ms to 8.2ms, all below the minimum response time, i.e. \( \Delta t = 0.2s \). Therefore, 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 rates (approximately 30%, 70%). The results are shown in the 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 Figure 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.

TABLE IX: Localization Error with Different Parking Occupancy

<table>
<thead>
<tr>
<th>Occupancy</th>
<th>Localization Error(m)</th>
<th>Heading Error(degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>3.3</td>
<td>17.6</td>
</tr>
<tr>
<td>70%</td>
<td>3.62</td>
<td>34.2</td>
</tr>
</tbody>
</table>

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

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

(a) Average computation time under different \( d \) and \( H \) settings.

(b) Average localization error under different \( d \) and \( H \) settings.

Fig. 14: System performance with variation of parameters \( H \) and \( d \).

F. Computation-Accuracy Trade-off

We show in Figure 14a 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 Figure 14b the average localization error versus 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 \( H \) increase until the accuracy achieves its optimal given by Equation (28). In our experiments, the maximum vehicle speed \( V_{\text{max}} \) is 9m/s. We perform localization every 0.2s, i.e., \( \Delta t = 0.2s \).

We study the computation-accuracy trade-off 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 Equation (28). As shown in Figure 15a, the average computation time decreases as the grid size \( d \) increases. Moreover, the average computation time is approximately inverse proportional to the square of \( d \). Figure 15a fits well with Equation (31). Figure 15b 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 Figure 15c the computation-accuracy trade-off 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 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 trade-off 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**


