

**FUSING RF AND IMU SIGNALS WITH HMM IN
A DOCKED PHONE FOR INDOOR CARPARK
NAVIGATION**

by

ZHENG ZHANG

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy
in Computer Science and Engineering

December 2021, Hong Kong

Copyright © by Zheng ZHANG 2021

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature redacted

ZHENG ZHANG

FUSING RF AND IMU SIGNALS WITH HMM IN A DOCKED PHONE FOR INDOOR CARPARK NAVIGATION

by

ZHENG ZHANG

This is to certify that I have examined the above M.Phil. thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Signature redacted

PROFESSOR S.H- GARY CHAN, THESIS SUPERVISOR

Signature redacted

PROFESSOR RAYMOND WONG, THESIS CO-SUPERVISOR

Signature redacted

PROF.DIT-YAN YEUNG, HEAD OF DEPARTMENT

Department of Computer Science and Engineering

6 December 2021

ACKNOWLEDGMENTS

I would never have completed this work without the help from many people. First of all, I thank my advisor, Professor Gary Chan, for his years of patient mentoring, advice, and encouragement. I feel he treats us students like his children. I learned from him how to think critically and communicate effectively. These skills will be important through my life. I also thank my co-advisor Raymond Wong for providing valuable advice for my future career development.

I thank the members of my thesis committee, Professor S.C. Cheung and Professor Dan Xu, for their insightful comments on improving this work.

I thank my parents being supportive throughout my Mphil study. They have helped me to collect the raw data needed in the thesis.

I also thank my group members in HKUST – Tianlang He, Steve Zhuo, Guanyao Li and many others. They provided valuable advice to improve my work.

TABLE OF CONTENTS

Title Page	i
Authorization Page	ii
Signature Page	iii
Acknowledgments	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Abstract	ix
Chapter 1 Introduction	1
Chapter 2 Related Works	6
2.1 OCAI-based Indoor Vehicle Localization	6
2.2 IMU-based Tracking	6
2.3 RF-based Localization	7
2.4 HMM-based Vehicle Localization	8
Chapter 3 System Overview	9
3.1 Offline Training Phase	9
3.2 Online Navigation Phase	11
3.3 Fusion Model Formulation	12
Chapter 4 Speed Collection in Offline Phase	15

Chapter 5 Signal Processing in Online Phase	17
5.1 Speed Pattern Classifier	17
5.2 Heading Estimation and Turn Detection	18
5.3 RF Localization	19
Chapter 6 Online HMM Fusion and Its Complexity	20
6.1 Prediction	20
6.2 Refinement	21
6.3 Computational Complexity Analysis	22
Chapter 7 Illustrative Experiment Results	24
7.1 Experiment Settings & Performance Metrics	24
7.2 Evaluation of Speed Pattern Classifier	27
7.3 Overall Performance	27
7.4 Computation-accuracy Trade-off	29
Chapter 8 Conclusion and Future Works	32
References	33

LIST OF FIGURES

1.1	A 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 with some transition probability.	2
3.1	Workflow of offline training phase.	11
3.2	Online navigation phase.	12
3.3	Dependence graph of RICH. The vehicle state s_t at time t is depended on the previous state s_{t-1} and the vehicle's action u_t . Meanwhile, the vehicle state can be observed by z_t .	13
4.1	Typical car speed patterns observed in an indoor carpark.	15
5.1	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.	17
5.2	The speed pattern recognition model.	18
6.1	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.	22
7.1	Raw data collection app.	26
7.2	CDF of different schemes.	28
7.3	Localization error over time under the a) regular driving pattern and the b) low speed driving pattern.	29
7.4	System performance with variation of parameters H and d .	30
7.5	Performance of RICH under different grid size settings. H is set to be optimal in Equation (6.15).	31

LIST OF TABLES

3.1	Important symbols used.	10
7.1	Specification of the two car parks.	24
7.2	Baseline parameters.	25
7.3	Accuracy of speed classifier	27
7.4	Average computation time on 4-core i7-6560 CPU work station, Python 3.7.1.	28
7.5	Average computation time of RICH on various phones.	29

FUSING RF AND IMU SIGNALS WITH HMM IN A DOCKED PHONE FOR INDOOR CARPARK NAVIGATION

by

ZHENG ZHANG

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

Indoor carpark often suffers from unavailable or weak GNSS (Global Navigation Satellite System) and cellular signals. Under such condition, we consider the challenging problem of navigating a driver with an offline smartphone docked at the car dashboard. There is some basic RF (radio-frequency) infrastructure in the premise, but due to signal attenuation by the car body, the location is noisy and intermittent. Previous works on carpark navigation often require special measurement equipment such as on-car additional infrastructure (OCAI), or perform integration of IMU (inertial measurement unit) signals over time. These are either not cost-effective to deploy or prone to high propagation error.

We propose RICH, a novel, real-time, simple and cost-effective docked-phone approach to fuse **RF** and **IMU** signals for indoor carpark navigation using **HMM** (Hidden Markov Model). RICH is the first deployment-ready learning-based offline fusion approach executed completely in local phone without any OCAI or error-prone IMU integration. RICH uses IMU signals to classify

accurately the car speed pattern and detect its heading and turning. This information and the crude RF localization are then fused in an HMM framework to compute the location distribution of the car. We present an analysis of RICH complexity and present its trade-off between computation and accuracy. We implement RICH in smartphones, and conduct extensive experiments in real carparks. As compared with the state of the art, RICH achieves substantially lower localization error (lower by 40%), and is computationally light-weight and fast suitablefor docked phone navigation(less than 10ms per location).

CHAPTER 1

INTRODUCTION

Urbanization has led to increasing number of indoor carparks. In these carparks, global navigation satellite system (GNSS) and cellular signals are generally weak or unavailable. We consider in this work the challenging carpark navigation problem, which is to direct the car driver to a spot (parking slot, exit, etc.) in the absence of GNSS and cellular signals. To achieve that, an offline app, pre-installed in the phone docked on the car dashboard, makes turn-by-turn instruction to direct the driver within the carpark. An indoor carpark is characterized by well defined lanes with junctions. We illustrate in Fig. 1.1 a real carpark floorplan, where the vehicle is constrained in paths indicated in dotted lines. A car typically travels in the 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 irregular unexpected events such as pedestrian crossing or backing of other cars in front.

In order to support phone-based navigation, we consider the realistic scenario that a basic radio frequency (RF) infrastructure, such as Bluetooth beacons and/or WiFi access points (APs), has already been deployed in the carpark. However, due to the fact that the RF signals sensed by the docked phone are markedly weakened by the car's windshield or body, the ambient in-car RF signals is intermittent and noisy, therefore cannot offer prompt response to the vehicle's fast change of location and satisfactory localization accuracy. To make the matter worse, high in-car packet loss and possibly low RF scanning frequency make the localization intermittent, greatly hampering smooth navigation and user experience. To overcome these, we propose to use commonly available on-phone inertial measurement unit (IMU), which is with 9DOF (9 Degrees of Freedom) consisting of a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer to sense the acceleration,

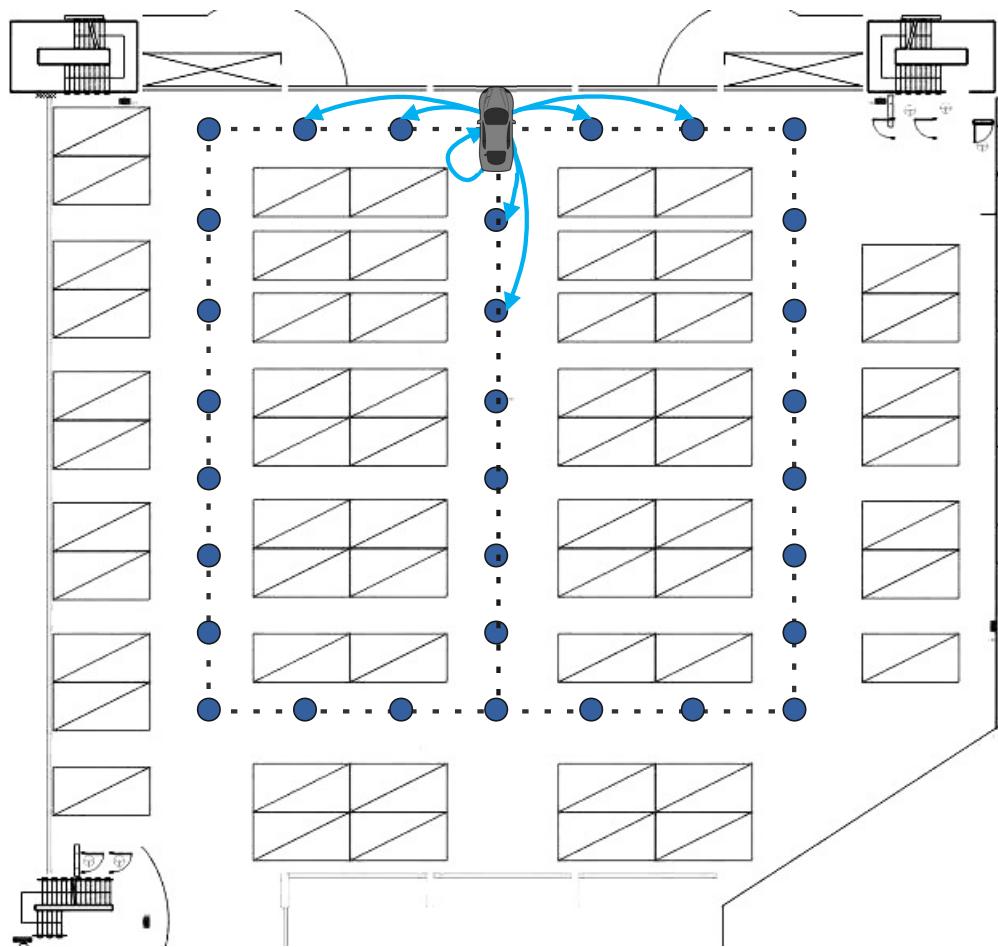


Figure 1.1: A typical carpark floorplan with the HMM states corresponding to possible car positions. In a time slot, the car may move at most H hops away from its current position with some transition probability.

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 navigate the driver in indoor carpark.

Due to unfavorable RF signal environment, previous works on carpark navigation often deploy on-car additional infrastructures (OCAIs) [15, 17, 27] 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, 19] have been proposed for indoor carpark navigation. As they are mainly cloud-based, deploying such system is costly, requiring high network bandwidth, computation power and number of cameras. There may also be latency, privacy and occlusion issues. There have been works using RF signal for navigation [1, 6, 8, 23]. Moreover, they rely on a rather good in-car signal-to-noise ratio, which requires high AP density, signal power and infrastructure cost. This is not practical or cost-effective in reality. Another body of works uses dead reckoning (DR) to estimate object location by means of double integration of linear acceleration [4, 10, 33]. However, this is error prone due to error accumulation over time. To mitigate that, some require highly accurate IMU sensors beyond what is available in common smartphones today. Yet another body of works uses pedestrian pedometer and dead-reckoning to localize users [11, 14, 18, 35]. They are, however, not applicable for docked phone navigation because no steps can be detected in such setting.

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 simple, real-time, memory and computationally efficient, and applicable to any offline commodity smartphone. It is a scalable fusion framework that can fuse any location sensors with IMU to perform indoor vehicle localization. To the best of our knowledge, this is the first piece of fusion work deployable for indoor carpark navigation with an offline docked phone. Due to measurement noise and propagation error, RICH abolishes the traditional integration on IMU readings. Instead, it adopts a simple yet efficient deep learning approach to extract and classify accurately distinctive IMU signal features corresponding to a number of vehicular speed patterns including coming to a full stop, driving at low speed, or the normal case of traveling around the designated speed limit. IMU also provides valuable information on car heading and turning. Fusing the above with the crude and intermittent

location estimated from RF, RICH then computes the car location by means of an HMM.

RICH is designed based on the observation that RF location and IMU information can complement each other to achieve high accuracy. Leveraging the motion regularity, IMU detects car turns and captures the car’s speed pattern to predict the car’s location over some short time scale (seconds). To compensate error propagation and location drift, RF is used to mitigate global ambiguity by constraining the car location to a region.

Time is slotted in RICH with slot size in the order of a fraction of a second (0.2 second in our experiment). As shown in Fig. 1.1, the driving lanes are divided into grid points, which is the state 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. The probability distribution of the car location in the current slot jointly depends 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 deployable approach for offline docked phones to navigate cars in indoor carparks. It is OCAI free, memory and computationally efficient, and implementable in commodity smartphones without bandwidth concerns. With a simple yet efficient deep learning model, it fuses RF signals and phone IMU readings by means of an HMM to achieve real-time and highly accurate localization.
- *Computational analysis of RICH:* We provide an analysis on the computation complexity of RICH, which captures the trade-off between computation and localization accuracy. Users can adjust system parameters to make RICH adaptive to mobile phones according to its computing power and venues of different size.
- *Extensive experimental evaluation of RICH:* We have implemented RICH in mobile phones, and compare it with existing state-of-the-art schemes in real carpark sites. Our experimental results show that RICH achieves substantially better performance, with 40% reduction in localization error and 60% reduction in computation time in our experiments.

The remainder of this thesis is summarized as follows. After discussing related work in Chapter 2, we overview the offline and online phases of RICH, and formulate the fusion model in Chapter 3. Then we present the speed collection in the offline phase and online signal processing of RICH in Chapters 4 and 5, respectively, followed by HMM fusion and its complexity analysis in Chapters 6. We present our experimental results in Chapter 7 and conclude with future works in Chapter 8.

CHAPTER 2

RELATED WORKS

In this chapter, we discuss related works in four aspects: OCAI based indoor vehicle localization (Section 2.1), IMU-based tracking (Section 2.2), RF-based localization (Section 2.3) and HMM-based vehicle localization (Section 2.4).

2.1 OCAI-based Indoor Vehicle Localization

OCAI-based localization locates a vehicle by installing various sensors or equipments in a car. Early vehicle localization methods focus on Vehicular Ad-hoc Networks (VANET) [12, 13, 28], whose basic idea is to perform cooperative localization through vehicle-to-vehicle communication. With the rapid development of SLAM (simultaneous localization and mapping), more works have been proposed locating vehicles with wheel odometers [26, 30], lidars, surround-view cameras and high accuracy IMU sensors. A fusion of lidar and IMU is proposed in [15]. AVP-SLAM [27] proposes a visual semantic SLAM approach using surround-view cameras, IMU and wheel odometers. These SLAM-based methods have achieved centimetre level localization accuracy. Another approach such as [25] proposed a fusion of WiFi, IMU and lidar applying a Gaussian-mixture particle filter model. OCAI-based methods often require special installation and hence are more expensive and customized. While not a general approach for all cars, these works are orthogonal and complementary to ours. RICH is a cost-effective solution, based on docked smartphone for general car users without additional installation of on-car sensors.

2.2 IMU-based Tracking

Phone IMU is often leveraged for various tracking tasks. PDR (pedestrian dead reckoning) [11, 18, 20] detects step events with an IMU’s acclerometer readings to track pedestrians. A typical

PDR system contains three components: 1) step detection: to detect pedestrian steps; 2) step length estimation: to estimate the length of one step; and 3) heading estimation: to estimate pedestrian’s heading. Due to a lack of periodic walking patterns, PDR is not suitable for vehicle tracking.

Works such as [5,10,17] estimate a vehicle’s travel distance by the double integration of forward acceleration. However, it requires highly accurate IMU sensors to achieve reasonable estimation accuracy, which is beyond what is available in common smartphones.

Semantic landmarks such as speed bumps and slopes can be detected using IMU to track vehicles. VeTrack [7] provides an IMU-based indoor vehicle localization scheme. It is complementary to our system if semantic landmarks are rich available in a carpark.

2.3 RF-based Localization

Received signal strength (RSS) of RF signals can be leveraged to locate targets. Typical RF signal types include WiFi, Bluetooth low energy, Zigbee and 5G. Fingerprinting (FP) has evolved over time to be the most popular RF-based localization method. The first FP system is RADAR [1]. Horus in [34] solves the problem of the temporal variations in RF signals. By considering the channel state information (CSI), ArrayTrack [31] achieves sub-meter-level accuracy. Besides the FP methods, several works such as WCL localization [3], EZ [8] and EZPerfect [23] map the RSSI readings to the physical distances between emitters and receivers to estimate the location with geometric methods.

RF-based methods are less accurate for car navigation [6]. This is because a car’s windshield is usually made of thick glass framed in metal. However, due to ineluctable attenuation of RF signals through a car’s body, using pure RF signal for navigation would require large number of AP deployment and power setting, leading to a costly system. Instead, RICH only requires a basic RF infrastructure where only rough car location is required.

2.4 HMM-based Vehicle Localization

HMM has been widely adopted in various road-level vehicle localization systems [16, 21, 24, 32]. Typical solutions adopt GPS [21] or GSM [16, 24] as their source of signal observation. RICH is not based on GNSS or cellular signals, and is more general because it considers unpredictable driving patterns such as slow driving and vehicle stoppage. Therefore, RICH achieves higher localization accuracy in the most general scenarios.

CHAPTER 3

SYSTEM OVERVIEW

In this chapter, we overview RICH and formulate the fusion model, by discussing the offline phase, online navigation phase, and the HMM fusion model in Sections 3.1, 3.2, and 3.3, respectively. We summarize the important symbols used in this work in Table 3.1, and the online localization algorithm in Algorithm 1 (explained below).

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) ;

- 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 $u_t \leftarrow [\rho_t, \phi_t]$;
- 5 $z_t \leftarrow [r_t, l_t]$;
- 6 **for** $j = 1, 2, 3, \dots, n$ **do**
- 7 $\overline{\alpha_t(j)} = \sum_i p(s_t = q_j | s_{t-1} = q_i, u_{t-1}) \alpha_{t-1}(i)$;
- 8 $\alpha_t(j) = p(z_t | s_t = q_j) \overline{\alpha_t(j)}$;
- 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)

3.1 Offline Training Phase

In the offline phase, a survey is firstly conducted in the carpark. We employ drivers to drive naturally around the carpark while a surveyor in the passenger seat collects driving trajectories and IMU signals with a mobile phone, where a driving trajectory refers to a sequence of car locations

Table 3.1: Important symbols used.

Symbol	Meaning
n	The total number of grid points (possible car states)
s_t	The car state at time (slot) t ($s_t = 1, 2, \dots, n$)
q_j	2-D coordinate for grid point j , $j = 1, 2, 3, \dots, n$
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
$\lceil \rceil$	Ceiling function

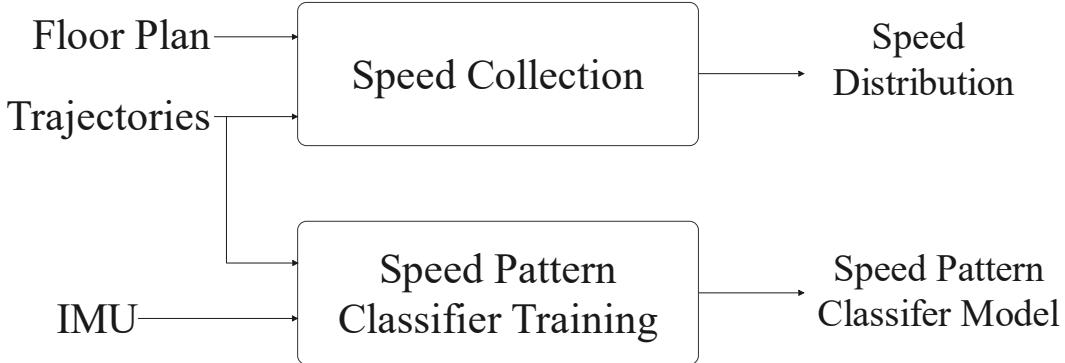


Figure 3.1: Workflow of offline training phase.

associated with their timestamps. Normally it takes 1-2 hours of site survey to setup one typical carpark.

We show in Fig. 3.1 the offline phase, which consists of:

- *Speed Collection*: Vehicle motion in a carpark forms S speed patterns. Here a speed pattern is described by a collection of vehicle speed over a period, say 2 seconds. 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.
- *Speed Pattern Classifier Training*: A speed pattern classifier model is trained with sourced IMU signals, saved to classify a car's speed pattern in online navigation phase.

3.2 Online Navigation Phase

We overview the online navigation phase in Fig. 3.2, which consists of:

- *Signal Processing*: Raw IMU and RF signals are processed in signal processing. Taking computation cost into consideration, we apply simple but efficient algorithms for each individual algorithms instead of build an end-to-end model for all. The *speed pattern classifier* trained in the offline phase processes the raw IMU signals and outputs the vehicle's real-time

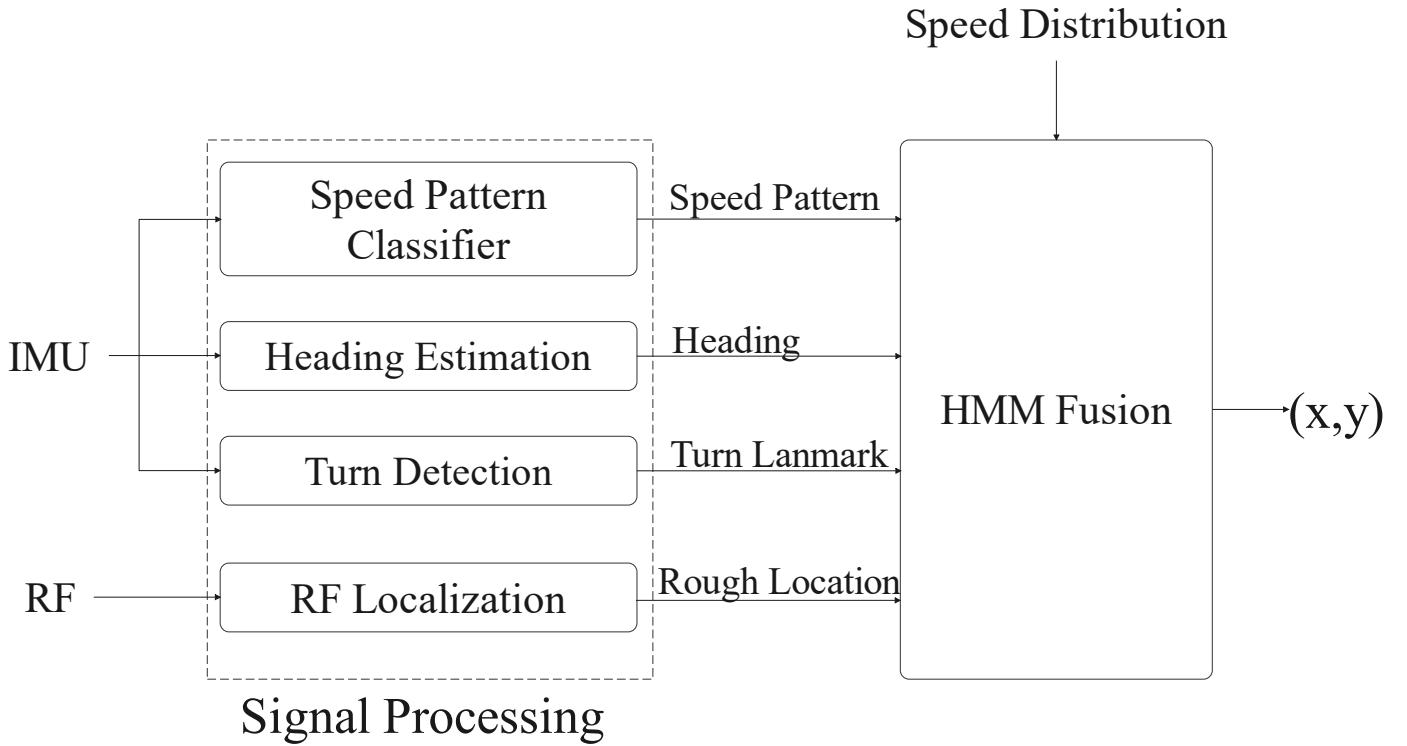


Figure 3.2: Online navigation phase.

speed pattern ρ_t . *Heading estimation* detects the vehicle’s heading ϕ_t using IMU. We also perform *turn detection* using a gyroscope to detect the turn landmark l_t . *RF localization* estimates the vehicle’s rough location r_t .

- *HMM Fusion*: All terms ρ_t, ϕ_t, l_t and r_t are fused with the speed distribution $f(v, q_1), f(v, q_2), \dots, f(v, q_n)$ in an HMM to estimate the vehicle location.

3.3 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 represent the vehicle state at time t , constrained at one of the n distinct grid points whose 2D coordinates are q_1, q_2, \dots, q_n . The vehicle’s action, defined as

$$u_t = [\rho_t, \phi_t], \quad (3.1)$$

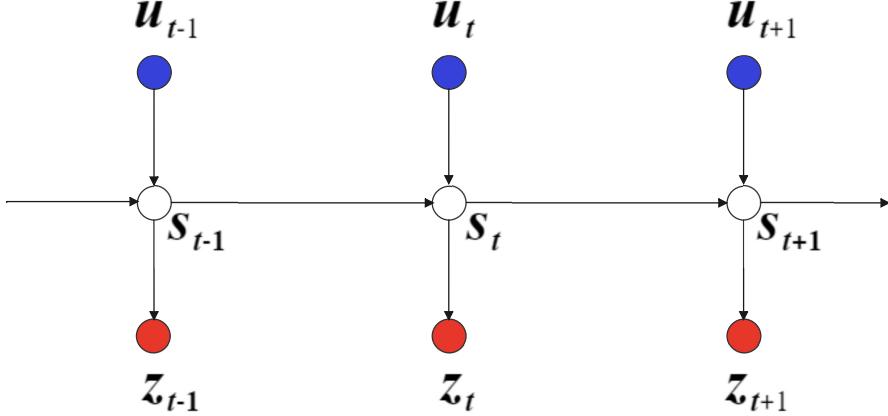


Figure 3.3: Dependence graph of RICH. The vehicle state s_t at time t is depended on the previous state s_{t-1} and the vehicle's action u_t . Meanwhile, the vehicle state can be observed by z_t .

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

$$z_t = [r_t, l_t]. \quad (3.2)$$

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 $u_{1:t} = [u_1, u_2, \dots, u_{t-1}, u_t]$ and observations $z_{1:t} = [z_1, z_2, \dots, z_{t-1}, z_t]$. Formally, the fusion objective is to estimate

$$p(s_t = j | u_{1:t}, z_{1:t}), \quad (3.3)$$

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

$$\begin{aligned} p(s_t = j | u_{1:t}, z_{1:t}) &= \frac{p(s_t = j, u_{1:t}, z_{1:t})}{p(u_{1:t}, z_{1:t})} \\ &\propto p(s_t = j, u_{1:t}, z_{1:t}) \\ &= \alpha_t(j). \end{aligned} \quad (3.4)$$

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

$$\alpha_t(j) = p(z_t | s_t = j) \sum_i p(s_t = j | s_{t-1} = i, u_t) \alpha_{t-1}(i). \quad (3.5)$$

Equation (3.5) indicates that the joint distribution $\alpha_t(1), \alpha_t(2), \dots, \alpha_t(n)$ can be recursively computed from the historical distribution $\alpha_{t-1}(1), \alpha_{t-1}(2), \dots, \alpha_{t-1}(n)$. Let

$$\overline{\alpha_t(j)} = \sum_i p(s_t = j | s_{t-1} = i, u_t) \alpha_{t-1}(i) \quad (3.6)$$

such that

$$\alpha_t(j) = p(z_t | s_t = j) \overline{\alpha_t(j)}. \quad (3.7)$$

Given Equation 3.7, 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 $\alpha_{t-1}(1), \alpha_{t-1}(2), \dots, \alpha_{t-1}(n)$ and the action u_t . The term $p(s_t = j | s_{t-1} = i, u_t)$ in Equation 3.5 is also known as the "transition model".
2. *Refinement*: Refine the predicted location $\overline{\alpha_t(j)}$ with sensor observations z_t . As the prediction drifts in a long run, predicted location is corrected by the sensor observations. The term $p(z_t | s_t = j)$ in Equation (3.7) 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.,

$$(\hat{x}_t, \hat{y}_t) = \frac{\sum_{\alpha_t(j) \in M} \alpha_t(j) q_j}{\sum_{\alpha_t(j) \in M} \alpha_t(j)}, \quad (3.8)$$

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

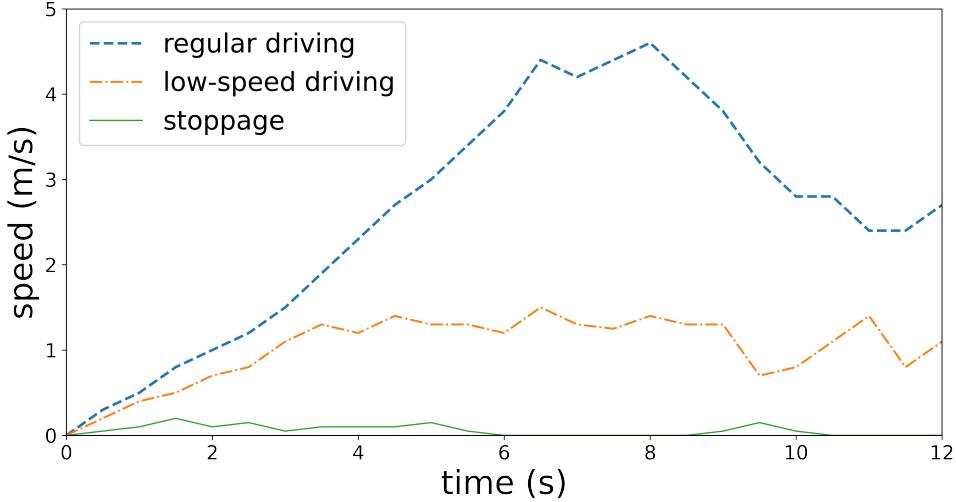


Figure 4.1: Typical car speed patterns observed in an indoor carpark.

CHAPTER 4

SPEED COLLECTION IN OFFLINE PHASE

Vehicle speed while driving in an indoor carpark forms several speed patterns. Here a speed pattern is described by an collection of vehicle speed in a period. To be specific, we consider the peak value of vehicle speed v_p in a period. The speed pattern is labelled as $\rho_t = i, i \in [0, S - 1]$, if

$$v_l(i) \leq v_p \leq v_h(i), \quad (4.1)$$

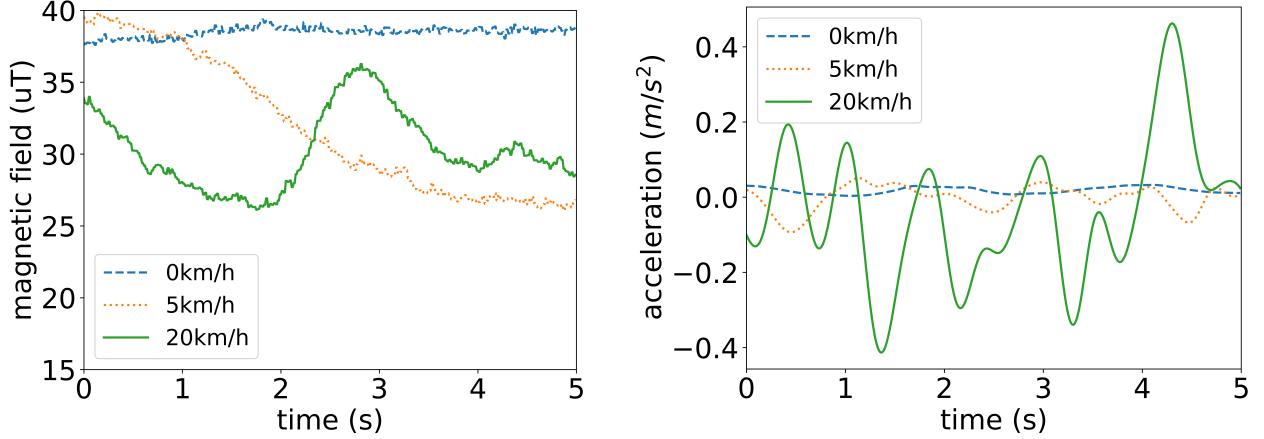
where $v_l(i)$ and $v_h(i)$ are the lower and upper bound speed values corresponding to the speed pattern i .

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 are 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 Figure 4.1.

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 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, as the driving lanes have been divided to grid points, we collect the car speed at each grid point q_j separately and estimate the speed distribution of each point q_j accordingly. Here we assume that the collected distribution is independent of the its driving direction. Vehicle speed distribution with the pattern i is collected as $F = \{f_i(q_1, v), f_i(q_2, v) \dots, f_i(q_n, v)\}$, 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.,

$$f_i(q_1, v) = f_i(q_2, v) = \dots = f_i(q_n, v) = f_i(v). \quad (4.2)$$



(a) Sensed geo-magnetic field with different speed patterns.

(b) Filtered vehicle's linear acceleration with different speed patterns.

Figure 5.1: 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.

CHAPTER 5

SIGNAL PROCESSING IN ONLINE PHASE

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 chapter, we discuss the speed pattern classifier in Section 5.1, heading estimation and turn detection in Section 5.2 and RF localization in Section 5.3.

5.1 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 Figures 5.1a and 5.1b.

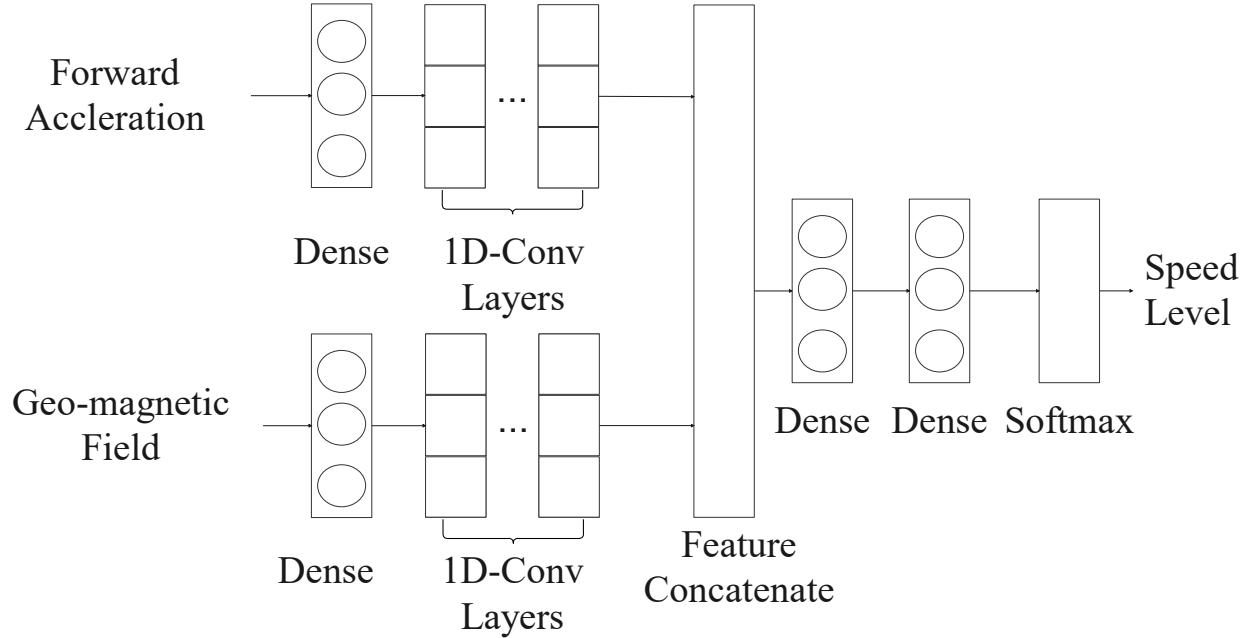


Figure 5.2: The speed pattern recognition model.

Based on the observations, we apply an 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, a number of IMU readings are selected. After batch normalization and a dense layer, the geomagnetic field and forward acceleration are processed separately by 1D-CNN layers to capture motion features. Finally, the features are concatenated and processed with dense layers.

5.2 Heading Estimation and Turn Detection

Heading estimation detects a car's heading ϕ_t . In this work, we adopt an efficient algorithm Madgwick filter [22] 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.,

$$l_t = 1, \text{ if } \sum_i \omega_i \geq T_\omega, \quad (5.1)$$

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.

5.3 RF Localization

RF localization estimates the vehicle's rough location \hat{s} . We can employ typical RF localization methods to acquire \hat{s} from the RSSI readings of RF emitters. Many RF localization methods [3, 8, 29, 34] can be adopted in our framework. In this work, 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

$$\hat{s} = \frac{\sum_j w_j M_j}{\sum_j w_j}, \quad (5.2)$$

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

$$w_j = \eta e^{\frac{P_j - P_0}{\Delta P}}, \quad (5.3)$$

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

CHAPTER 6

ONLINE HMM FUSION AND ITS COMPLEXITY

We present HMM fusion in this chapter. Recall that we have overviewed the HMM fusion model in Chapter 3 with two steps, prediction and refinement. We discuss the prediction step in Section 6.1 and the refinement step in Section 6.2. Afterwards, we discuss the computation complexity of RICH in Section 6.3.

6.1 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 (3.6) is modelled as

$$\begin{aligned}\overline{\alpha_t(j)} &= \sum_i p(s_t = j | \phi_t, \rho_t, s_{t-1} = i) \alpha_{t-1}(i) \\ &= \sum_i a_t(i, j) p(s_t = j | \phi_t, s_{t-1} = i) \alpha_{t-1}(i),\end{aligned}\tag{6.1}$$

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

$$a_t(i, j) \propto \int_v K\left(\frac{v - v_{ij}}{\sigma_v}\right) f_{\rho_t}(v, q_i) dv,\tag{6.2}$$

such that

$$\sum_j a_t(i, j) = 1,\tag{6.3}$$

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.,

$$v_{ij} = \frac{\|q_i - q_j\|}{\Delta t}.\tag{6.4}$$

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.,

$$K(x) = e^{-\frac{1}{2}x^2}. \quad (6.5)$$

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 Δt , i.e.,

$$a_t(i, j) = 0, \text{ if } \|q_i - q_j\| > Hd. \quad (6.6)$$

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 Figure 6.1. 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 the orientation estimation error forms a Huber-loss distribution given by

$$\ln(L(x, \sigma)) = \begin{cases} \frac{1}{2}|x|^2 + 2 \ln \sigma, & \text{if } |x| \leq \sigma; \\ \sigma(|x| - \frac{1}{2}\sigma) + 2 \ln \sigma, & \text{if } |x| > \sigma. \end{cases} \quad (6.7)$$

Then we estimate the probability with

$$p(s_t = j | \phi_t, s_{t-1} = i) = L(h(\phi, q_i, q_j), \sigma_\phi), \quad (6.8)$$

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

$$\begin{aligned} h(\phi, q_i, q_j) &= h(\phi, (x_i, y_i), (x_j, y_j)) \\ &= \arccos \left(\frac{(x_j - x_i) \cos(\phi) + (y_j - y_i) \sin(\hat{\phi})}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \right). \end{aligned} \quad (6.9)$$

6.2 Refinement

We refine the predicted location jointly 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

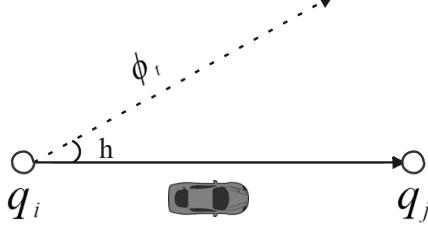


Figure 6.1: h represents the angle between the direction of proposed transition ($s_{t-1} = q_i$ and $s_t = q_j$) and the observed vehicle's heading ϕ_t . A smaller h indicates a higher probability of proposed transition.

follows:

$$\alpha_t(j) = p(r_t|s_t = j) p(l_t|s_t = j) \overline{\alpha_t(j)}. \quad (6.10)$$

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.,

$$p(r_t|s_t = j) = L(\|r_t - q_j\|, \sigma_s). \quad (6.11)$$

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:

$$p(l_t|s_t = j) = g(q_j)^{l_t} (1 - g(q_j))^{1-l_t}, l_t \in \{0, 1\}, \quad (6.12)$$

where

$$g(q_j) = \begin{cases} \beta, & \text{if } q_j \text{ is a junction point;} \\ 1 - \gamma, & \text{if } q_j \text{ is not a junction point.} \end{cases} \quad (6.13)$$

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

6.3 Computational Complexity Analysis

The computational complexity in online localization is

$$O(F) + O(Hn), \quad (6.14)$$

where F denotes the computation cost for signal processing, H is the number of hops and n is the number of states. We explain Equation (6.14) 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.

If the vehicle speed is upper limited at V_{max} , the minimum H that obtains the optimal localization accuracy is chosen as

$$H = \left\lceil \frac{V_{max} \Delta t}{d} \right\rceil, \quad (6.15)$$

where Δt denotes the unit period to perform localization, d is the grid size and $\lceil x \rceil$ denotes the ceiling of x . From Equation (6.15), we have apparently

$$H \propto d^{-1}. \quad (6.16)$$

Meanwhile, the total number of states n satisfies that

$$n \propto d^{-1} \quad (6.17)$$

because the total length of driveable paths in an indoor car park is fixed. Therefore, the computation complexity corresponding to the grid size d is

$$O(F) + O(Cd^{-2}), \quad (6.18)$$

where C is a constant with V_{max} acknowledged.

Grid size d is the critical hyper-parameter to balance the computation complexity and localization accuracy. A smaller grid size indicates higher computation time and higher latency, but results in higher localization accuracy. We discuss the computation-accuracy trade off with our experimental results in Chapter 7.

Table 7.1: Specification of the two car parks.

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

CHAPTER 7

ILLUSTRATIVE EXPERIMENT RESULTS

To validate RICH, we have implemented it in smartphones and conducted extensive experiments in real carparks. In this chapter, we first introduce our experimental settings, performance metrics and comparison schemes in Section 7.1. The speed classifier used in RICH is evaluated in Section 7.2. We present the overall performance in Section 7.3. Finally, the computation-accuracy trade-off is discussed in Section 7.4.

7.1 Experiment Settings & Performance Metrics

The experiments are conducted in two typical indoor carparks, a university carpark and a private apartment carpark. Both of the car parks are deployed with the proper density of Bluetooth low energy beacons (iBeacons). Specifications of the two carparks are shown in table 7.1. Different brands of private cars are involved in the experiments, including Hyundai, BMW, Honda and Nissan. Mobile phones used 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. To account for the driver and car heterogeneity, a total of 6 volunteer drivers participated in the experiments. Drivers also vary in driving age and gender. They drive in prescribed routes while the surveyor

Table 7.2: Baseline parameters.

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

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 7.2.

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 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 Chapter 5.

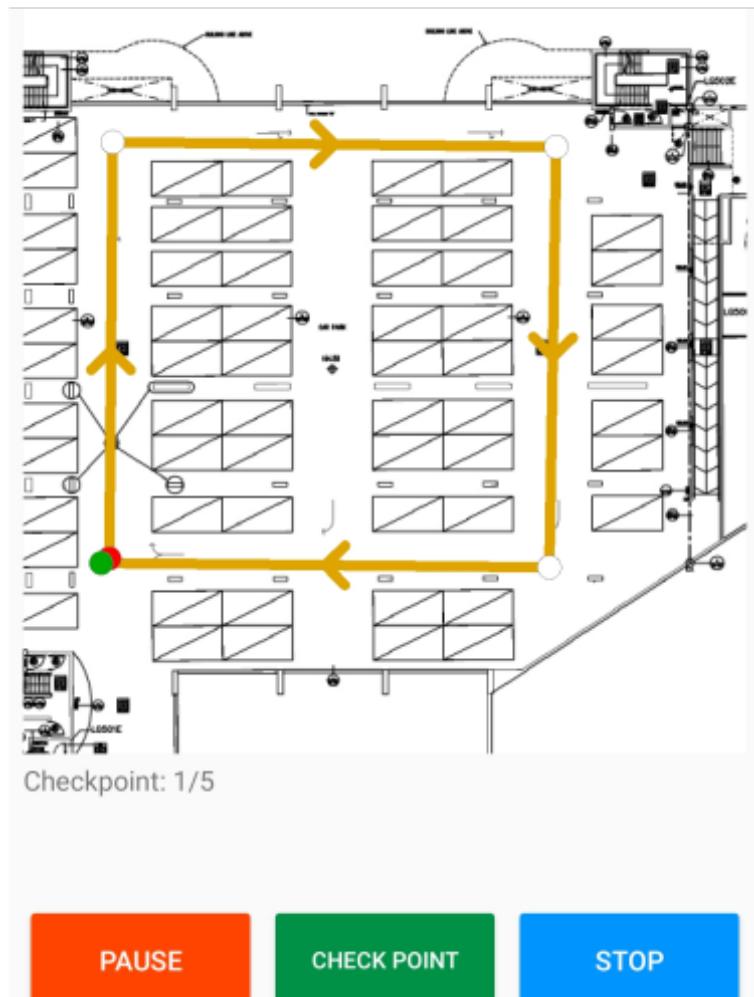


Figure 7.1: Raw data collection app.

Table 7.3: 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

- GMFP [25]: It is a state-of-the-art approach which has similar sensor settings as RICH. GMFP 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 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.

7.2 Evaluation of 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 7.3 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.

7.3 Overall Performance

In figures 7.2a and 7.2b, 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 GMFP in localization error. We summarize the the average computation time and average localization error of all schemes in Table 7.4. WCL consumes least the computation time, as both GMFP and

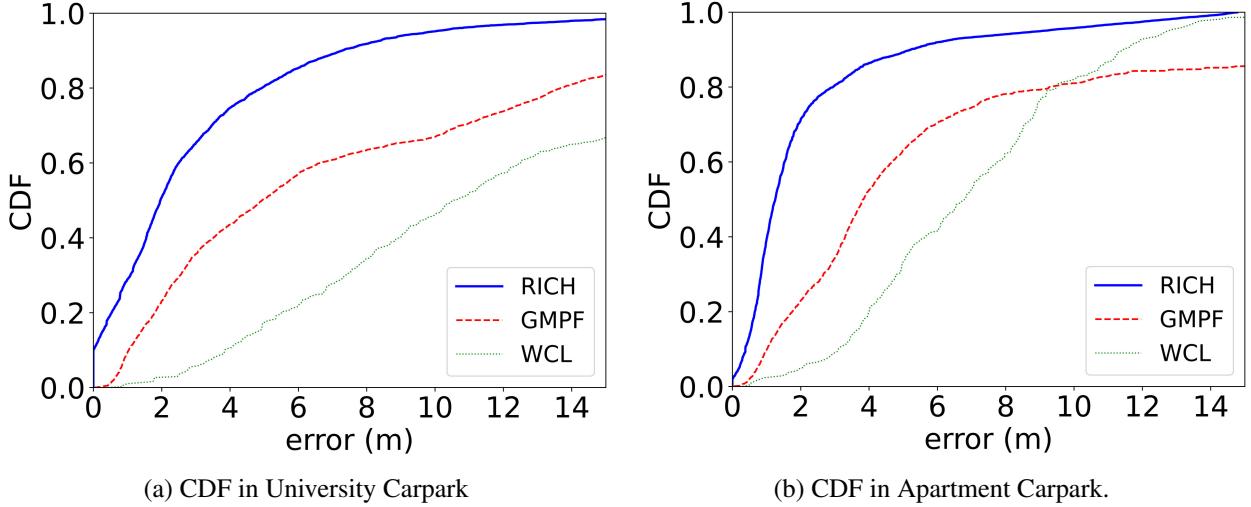


Figure 7.2: CDF of different schemes.

Table 7.4: 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

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 7.2a and 7.2b 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 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 7.3a and 7.3b 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

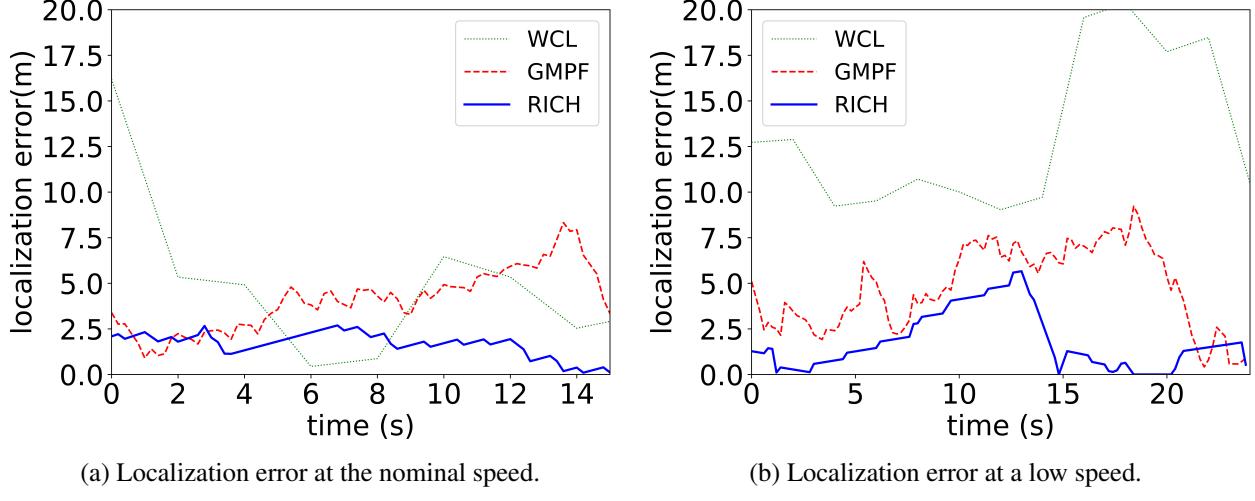


Figure 7.3: Localization error over time under the a) regular driving pattern and the b) low speed driving pattern.

Table 7.5: Average computation time of RICH on various phones.

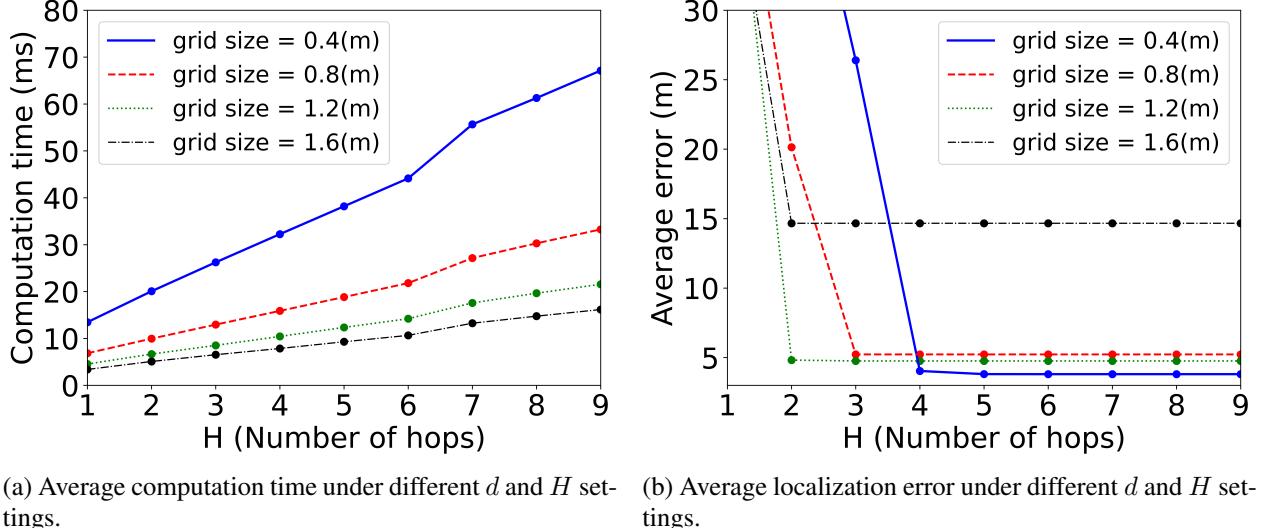
Phone type	Samsung S8	Vivo Y12	iPhone 11
Computation time (ms)	8.2	4.2	0.9

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.

We also test the average computation time of RICH on various phones. As shown in table 7.5, the average computation 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.

7.4 Computation-accuracy Trade-off

We discussed in Section 6.3 that the grid size d and the number of hops H have critical impacts on the system's performance. We evaluated how the parameters d and H affect the localization accuracy and computation time.



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

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

Figure 7.4: System performance with variation of parameters H and d .

We show in Figure 7.4a 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 7.4b 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 (6.15). In our experiments, the maximum vehicle speed V_{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 (6.15). As shown in Fig. 7.5a, the average computation time decreases as the the grid size d increases. Moreover, the average computation time is approximately inverse proportional to the square of d . Figure 7.5a fits well with Equation (6.18). Figure 7.5b 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 7.5c the computation-accuracy trade-off curve. The average localization error decreases and converges to a minimum as we afford more computation time.

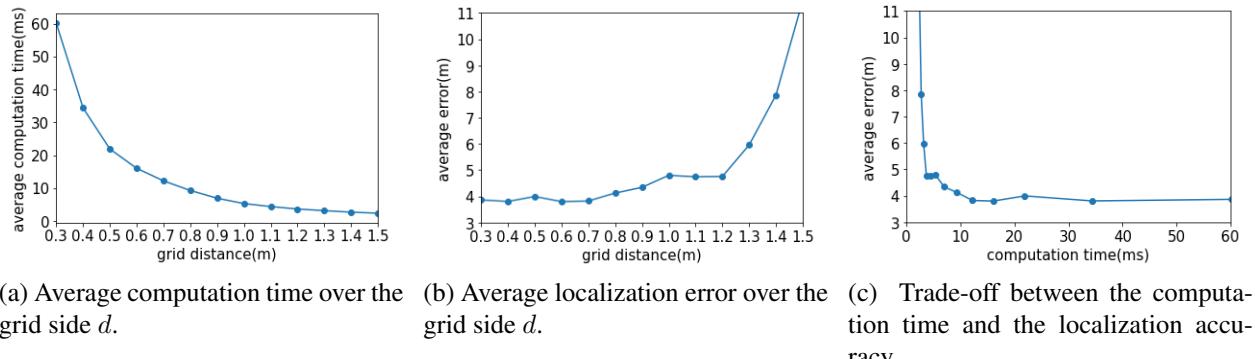


Figure 7.5: Performance of RICH under different grid size settings. H is set to be optimal in Equation (6.15).

CHAPTER 8

CONCLUSION AND FUTURE WORKS

We propose RICH, a robust and cost-effective offline approach to navigate indoor car with a docked phone. Without OCAI deployments and error-prone IMU integration, RICH combines on-phone RF and IMU signals with an HMM. Using IMU signals, RICH classifies the speed pattern of the car, and detects the car heading and turning. This information and the crude RF localization are then fused with an HMM to compute the car location.

We present the computational complexity of RICH, and have implemented RICH and have conducted extensive experiments in two real carparks. 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 to be deployed real-time in offline smartphones.

There are several future directions to improve this work:

- *Survey reduction:* RICH requires a survey. To reduce site survey efforts, we consider an implicit calibration-free crowdsourcing approach to collect driving trajectories and IMU signals.
- *Semantic landmark detection with IMU:* Currently, RICH does not detect semantic landmarks such as speed bumps and slopes. These semantic features can be further utilized to improve localization accuracy.
- *Fusion of other signals:* Other signal sources providing location information (such as lidar and camera) could be utilized to complement our system if it is available.

REFERENCES

- [1] P. Bahl and V. N. Padmanabhan. Radar: An in-building RF-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784 vol.2, 2000.
- [2] Daniel Becker, Fabian Thiele, Oliver Sawade, and Ilja Radusch. Cost-effective camera based ground truth for indoor localization. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 885–890, July 2015.
- [3] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann. Weighted centroid localization in Zigbee-based sensor networks. In *2007 IEEE International Symposium on Intelligent Signal Processing*, pages 1–6, October 2007.
- [4] Martin Brossard, Axel Barrau, and Silvère Bonnabel. AI-IMU Dead-Reckoning. *IEEE Transactions on Intelligent Vehicles*, 5(4):585–595, December 2020.
- [5] Changhao Chen, Yishu Miao, Chris Xiaoxuan Lu, Linhai Xie, Phil Blunsom, Andrew Markham, and Niki Trigoni. Motiontransformer: Transferring neural inertial tracking between domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8009–8016, July 2019.
- [6] Chi-Hua Chen, Chi-Ao Lee, and Chi-Chun Lo. Vehicle localization and velocity estimation based on mobile phone sensing. *IEEE Access*, 4:803–817, 2016.
- [7] L. Cheng and T. Qiao. Localization in the parking lot by parked-vehicle assistance. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3629–3634, December 2016.
- [8] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom ’10, pages 173–184, New York, NY, USA, 2010. ACM.

- [9] Jens Einsiedler, Daniel Becker, and Ilja Radusch. External visual positioning system for enclosed carparks. In *2014 11th Workshop on Positioning, Navigation and Communication (WPNC)*, pages 1–6, March 2014.
- [10] Ruipeng Gao, Xuan Xiao, Shuli Zhu, Weiwei Xing, Chi Li, Lei Liu, Li Ma, and Hua Chai. Glow in the dark: Smartphone inertial odometry for vehicle tracking in GPS blocked environments. *IEEE Internet of Things Journal*, pages 1–1, 2021.
- [11] Yuan Gao, Qingxuan Yang, Guanfeng Li, Edward Y. Chang, Dong Wang, Chengu Wang, Hang Qu, Pei Dong, and Faen Zhang. XINS: The anatomy of an indoor positioning and navigation architecture. In *Proceedings of the 1st International Workshop on Mobile Location-Based Service, MLBS ’11*, pages 41–50, New York, NY, USA, 2011. ACM.
- [12] Keyvan Golestan, Sepideh Seifzadeh, Mohamed Kamel, Fakhri Karray, and Farook Sattar. Vehicle localization in VANETs using data fusion and V2V communication. In *Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, DIVANet ’12*, pages 123–130, New York, NY, USA, October 2012. Association for Computing Machinery.
- [13] Faruk Baturalp Günay, Ercüment Öztürk, Tuğrul Çavdar, Y. Sinan Hanay, and Atta ur Rehman Khan. Vehicular ad hoc network (VANET) localization techniques: A survey. *Archives of Computational Methods in Engineering*, September 2020.
- [14] Suining He, S.-H. Gary Chan, Lei Yu, and Ning Liu. Slac: Calibration-free pedometer-fingerprint fusion for indoor localization. *IEEE Transactions on Mobile Computing*, 17(5):1176–1189, May 2018.
- [15] W. Hess, D. Kohler, H. Rapp, and D. Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, May 2016.
- [16] Mohamed Ibrahim and Moustafa Youssef. A Hidden Markov Model for Localization Using Low-End GSM Cell Phones. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5, June 2011.

- [17] Jianhao Jiao, Haoyang Ye, Yilong Zhu, and Ming Liu. Robust odometry and mapping for multi-lidar systems with online extrinsic calibration. *IEEE Transactions on Robotics*, pages 1–10, 2021.
- [18] W. Kang and Y. Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal*, 15(5):2906–2916, May 2015.
- [19] Anil Kumar Tirumala Ravi Kumar, Bernd Schäufele, Daniel Becker, Oliver Sawade, and Ilja Radusch. Indoor localization of vehicles using Deep Learning. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, June 2016.
- [20] Yi Lu, Dongyan Wei, Qifeng Lai, Wen Li, and Hong Yuan. A Context-Recognition-Aided PDR Localization Method Based on the Hidden Markov Model. *Sensors (Basel, Switzerland)*, 16(12), November 2016.
- [21] An Luo, Shenghua Chen, and Bin Xv. Enhanced map-matching algorithm with a Hidden Markov model for mobile phone positioning. *ISPRS International Journal of Geo-Information*, 6(11):327, November 2017.
- [22] Sebastian O H Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Technical report, Report x-io and University of Bristol (UK), 2010.
- [23] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, and Venkata N. Padmanabhan. Centaur: Locating devices in an office environment. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking - Mobicom '12*, page 281, Istanbul, Turkey, 2012. ACM Press.
- [24] Paul Newson and John Krumm. Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS ’09, pages 336–343, New York, NY, USA, November 2009. Association for Computing Machinery.
- [25] D. Nguyen, T. Dao, E. Castelli, and F. Nashashibi. A fusion method for localization of intelligent vehicles in carparks. *IEEE Access*, 8:99729–99739, 2020.

- [26] Xiaoji Niu, Yibin Wu, and Jian Kuang. Wheel-INS: A wheel-mounted MEMS IMU-based dead reckoning system. *IEEE Transactions on Vehicular Technology*, pages 1–1, 2021.
- [27] Tong Qin, Tongqing Chen, Yilun Chen, and Qing Su. AVP-SLAM: Semantic visual mapping and localization for autonomous vehicles in the parking lot. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5939–5945, October 2020.
- [28] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar. Broadcasting in VANET. In *2007 Mobile Networking for Vehicular Environments*, pages 7–12, May 2007.
- [29] J. Wang, P. Urriza, Y. Han, and D. Cabric. Weighted centroid localization algorithm: Theoretical analysis and distributed implementation. *IEEE Transactions on Wireless Communications*, 10(10):3403–3413, October 2011.
- [30] Yibin Wu, Xiaoji Niu, and Jian Kuang. A comparison of three measurement models for the Wheel-mounted MEMS IMU-based Dead Reckoning System. *IEEE Transactions on Vehicular Technology*, pages 1–1, 2021.
- [31] Jie Xiong and Kyle Jamieson. ArrayTrack: A fine-grained indoor location system. In *10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*, pages 71–84, 2013.
- [32] Xiangyu Xu, Jiadi Yu, Yanmin Zhu, Zhichen Wu, Jianda Li, and Minglu Li. Leveraging smartphones for vehicle lane-level localization on highways. *IEEE Transactions on Mobile Computing*, 17(8):1894–1907, August 2018.
- [33] Hang Yan, Qi Shan, and Yasutaka Furukawa. RIDI: Robust IMU double integration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 621–636, 2018.
- [34] Moustafa Youssef and Ashok Agrawala. The Horus WLAN location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, MobiSys ’05*, pages 205–218, New York, NY, USA, 2005. ACM.
- [35] Chunyang Yu, Naser El-Sheimy, Haiyu Lan, and Zhenbo Liu. Map-based indoor pedestrian navigation using an auxiliary particle filter. *Micromachines*, 8(7):225, July 2017.