Map matching algorithms for intelligent transportation system

Hou, Xiangting

2021

Hou, X. (2021). Map matching algorithms for intelligent transportation system. Doctoral thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/148923>

<https://hdl.handle.net/10356/148923>

<https://doi.org/10.32657/10356/148923>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 18:46:45 SGT

Map Matching Algorithms for Intelligent Transportation System

HOU XIANG TING

Interdisciplinary Graduate School
Complexity Institute

MAP MATCHING ALGORITHMS FOR INTELLIGENT TRANSPORTATION SYSTEM

HOU XIANGTING

Interdisciplinary Graduate School
Complexity Institute

A thesis submitted to the Nanyang Technological University in
partial fulfilment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

19 January 2021

.....

Date

Hou Xiangting

.....

Hou Xiangting

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

19 January 2021

.....

Date

.....

Prof. Cai Wentong

Authorship Attribution Statement

This thesis contains material from 3 papers published in the following peer-reviewed journal(s) / accepted (or submitted to) at conferences in which I am listed as an author.

Chapter 3 is composed of two publications:

Xiangting Hou, Linbo Luo, Wentong Cai, Masatoshi Hanai. Fast Online Map Matching for Recovering Travelling Routes from Low-Sampling GPS Data. IEEE International Conference on Ubiquitous Intelligence and Computing (2018). DOI: 10.1109/SmartWorld.2018.00165.

The contributions of the co-authors are as follows:

- Prof. Cai provided the initial project direction.
- I conducted literature review, proposed the map matching algorithm, designed and conducted all the experiments, and analyzed and interpreted experimental results with guidance of Prof. Cai and Prof. Luo.
- I wrote the manuscript drafts. The manuscript was revised by Prof. Luo, Prof. Cai and Dr. Hanai.

Linbo Luo, Xiangting Hou, Wentong Cai, Bin Guo. Incremental route inference from low-sampling GPS data: An opportunistic approach to online map matching. Information Sciences (2020). DOI: 10.1016/j.ins.2019.10.060

- Prof. Cai and Prof. Luo provided the initial project direction.
- I conducted literature review, proposed the map matching algorithm, designed and conducted all the experiments, and analyzed and interpreted experimental results with guidance of Prof. Cai and Prof. Luo.
- I wrote the manuscript drafts. The manuscript was revised by Prof. Luo, Prof. Cai and Prof. Guo.

Chapter 5 is submitted as Xiangting Hou, Linbo Luo, Wentong Cai, Bin Guo. E²T-CVL: An Efficient and Error-tolerant Approach for Collaborative Vehicle Localization. IEEE Internet of Things Journal (2020).

The contributions of the co-authors are as follows:

- Prof. Cai provided the initial project direction.
- I conducted literature review, proposed the localization approach, designed and conducted all the experiments, and analyzed and interpreted experimental results with guidance of Prof. Cai and Prof. Luo.

- I wrote the manuscript drafts. The manuscript was checked by Prof. Cai, Prof. Luo and Prof. Guo.

19 January 2021

.....

Date

Hou Xiangting

.....

Hou Xiangting

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Cai Wentong, my co-supervisor Prof. Cheong Siew Ann and my mentor Prof. Tang Xueyan. Their expertise, support and guidance gave me profound impact on my research work. Special thanks to Prof. Cai for his help, patience and encouragement during my PhD life. I would also like thank Prof. Luo Linbo for his research advise and help. In addition, I gratefully acknowledge the scholarship towards my PhD from Nanyang Technological University and would like to thank Interdisciplinary Graduate Programme for the invaluable research guidance. Also, I wish to thank Ms. Irene Ng-Goh Siew Lai for her support and friends in PDCL for their patient help. Last but not the least, I would like to thank my family, parents, my elders, my brothers and my sisters for their strong support and consistent help during my PhD journey.

Abstract

Map matching algorithms for route inference and vehicle localization are essential for Intelligent Transportation System (ITS) applications to reduce traffic congestion and accidents. Currently, the Global Positioning System (GPS) is widely used to obtain the absolute positions of vehicles. When the GPS has high measurement frequency, the travelling routes of vehicles can be inferred from the GPS positions. However, since the errors of GPS measurements can be large and the sampling rate can be low, these increase difficulties for route inference and vehicle localization. In this thesis, map matching algorithms for route inference and vehicle localization given GPS measurements are investigated.

For route inference, an opportunistic online map matching algorithm is firstly proposed. Unlike the existing online map matching algorithms which experience an inference delay between the GPS measurement and the inferred travelling route, the opportunistic algorithm is able to generate an immediate route inference when a new GPS measurement becomes available. In addition, in case when the opportunistic strategy produces an improper traveling route, a rollback mechanism that is able to correct the already inferred route is employed. The proposed algorithm is evaluated using the real dataset containing GPS trajectories. The experimental results show that the proposed algorithm outperforms the existing algorithms in terms of inference accuracy. More importantly, the proposed algorithm can yield much shorter inference delay and requires less execution time, which are critical for many real-time ITS applications.

Secondly, existing map matching algorithms usually adopt a driving cost model, which is formulated by considering travelling features and driving preferences (referred to as weights), to determine a travelling route. In general, the weights are estimated either empirically or through data-driven approaches. The empirical setting approaches determine the weights based on specific driving cost models. However, the weights need to be re-estimated empirically when the driving cost models are changed. Alternatively, the data-driven approaches determine the weights from

the selected historical routes of vehicles. However, estimation bias may be introduced due to inappropriate routes selection. To decrease the estimation bias, a distribution-based weights estimation approach is proposed, where weights are estimated by inferring their distributions through Bayesian inference, considering the traces generated from the drivers in a city scale. Then, a sampling-based strategy is adopted to determine the values of the weights from their distributions. Experiments are conducted to assess the map matching accuracy by comparing it with empirical setting and data-driven approaches. The experimental results show that the proposed approach is able to achieve higher map matching accuracy than the empirical setting and data-driven approaches.

For vehicle localization, approaches based on Vehicle-to-Vehicle (V2V) communication techniques are investigated to localize the target vehicle by leveraging the state information of its nearby vehicles. To determine the position of the target vehicle, most of the existing works need a large search space. This inevitably leads to large computational overheads. In addition, the measurement errors of state information can be large due to the external conditions. Therefore, the estimated position of the target vehicle may largely deviate from the real position. In this thesis, an efficient and error-tolerant target vehicle localization approach is proposed to increase the localization efficiency and accuracy even in the presence of large measurement errors. Unlike the existing approaches, a pruning-based strategy is proposed to prune the search space for the position of the target vehicle by considering the relative positions of its nearby vehicles. To determine the position of the target vehicle, a displacement-based selection strategy is also proposed to reduce the influence of the measurement errors of state information. Experiments are conducted to assess the localization accuracy and efficiency by comparing it with the state-of-the-art approaches. The experimental results show that compared with the existing approaches, the proposed approach is able to achieve higher localization efficiency and accuracy even with large measurement errors.

Similarly, localization approaches based on V2V communication techniques are also investigated to localize the nearby vehicles detected by a target vehicle. Different from the localization for the target vehicle, which is a self-localization process by using the state information of the nearby vehicles, the localization for the nearby vehicles is not processed by the vehicle itself. The geometry and trajectory information of the nearby vehicles are usually considered as the state information for

localization in the existing works. However, the diversity of the geometry information and insufficiency of the trajectories of the nearby vehicles may restrict the application of the existing localization approaches. In this thesis, a Bayesian-based localization approach is proposed to estimate the absolute positions of the nearby vehicles through incorporating the GPS measurements and relative positions. Different from the existing works, the proposed approach can be applied when the geometry information is diverse and the trajectories of the nearby vehicles are insufficient. The proposed approach is evaluated in terms of the localization accuracy based on a real dataset containing the trajectories of vehicles. The experimental results show that the proposed approach is able to achieve higher localization accuracy compared with the GPS measurements.

In summary, the major contributions of this thesis are: (1) An online map matching algorithm that reduces inference delays; (2) A distribution-based weights estimation approach that decreases the estimation bias; (3) An efficient, V2V-based target vehicle localization approach that tolerates measurement errors; and (4) A Bayesian-based nearby vehicle localization approach that can be applied when the geometry information is diverse and the trajectories of the nearby vehicles are insufficient.

Contents

Acknowledgements	xi
Abstract	xiii
List of Figures	xxi
List of Tables	xxiii
List of Notations	xxv
List of Terminologies	xxvii
1 Introduction	1
1.1 Motivation and Background	1
1.1.1 Map matching for route inference	3
1.1.2 Map matching for vehicle localization	4
1.2 Objectives	5
1.2.1 Map matching for route inference	5
1.2.1.1 Online map matching for route inference	5
1.2.1.2 Weights estimation for route inference	6
1.2.2 Map matching for vehicle localization	7
1.2.2.1 Localization for the target vehicle	7
1.2.2.2 Localization for the nearby vehicles	8
1.3 Summary of Contributions	9
1.4 Outline of the Thesis	11
2 Literature Review	13
2.1 Map matching for route inference	13
2.1.1 Offline map matching algorithms	13
2.1.2 Online map matching algorithms	15
2.1.3 Weights estimation for route inference	18
2.2 Map matching for vehicle localization	20
2.2.1 GPS measurement errors	21
2.2.2 Measurement errors of relative positions	23

2.2.3	Localization for the target vehicle	24
2.2.4	Localization for the nearby vehicles	27
2.3	Research Gaps	28
2.3.1	Map matching for route inference	28
2.3.2	Map matching for vehicle localization	29
3	Incremental Route Inference from Low-sampling GPS Data: An Opportunistic Approach to Online Map Matching	31
3.1	Framework	31
3.2	Detailed design of INC-RB	33
3.2.1	Offline preprocessing	33
3.2.2	General procedure for incremental online inference	34
3.2.3	Rollback mechanism	37
3.3	The experiment setup	39
3.3.1	Dataset description	40
3.3.2	Evaluation method and metrics	42
3.4	Evaluation results	44
3.5	Summary	50
4	Distribution-based Weights Estimation for Map Matching Algorithms	53
4.1	Framework	53
4.2	Detailed design of DBWES	54
4.2.1	Distribution-based Training Set Generation	54
4.2.2	Weights determination	56
4.3	Map matching inference	61
4.4	Experiments and results	63
4.4.1	Dataset preparation	63
4.4.2	Methods for comparison	64
4.4.3	Parameters settings:	64
4.4.4	Evaluation metrics	64
4.4.5	Experimental results	65
4.4.5.1	Weights distributions estimation	65
4.4.5.2	Map matching accuracy	67
4.4.5.3	Weights analysis	70
4.5	Summary	73
5	E²T-CVL: An Efficient and Error-tolerant Approach for Collaborative Vehicle Localization	75
5.1	Framework	75
5.2	Detailed design of E ² T-CVL	76
5.2.1	Nearby lanes preparation	76
5.2.2	Candidate lanes generation	77
5.2.3	Candidate zones generation	78

5.2.4	Optimal position determination	80
5.3	Performance evaluation and results	83
5.3.1	Simulation settings	83
5.3.2	Compared localization strategies	84
5.3.3	Parameters settings	85
5.3.4	Evaluation metrics	85
5.3.5	Experimental results	86
5.3.5.1	Accuracy results	86
5.3.5.2	Efficiency results	88
5.3.5.3	E ² T-CVL analysis	89
5.3.5.4	Sensitivity analysis	90
5.3.5.5	Cases analysis	92
5.4	Summary	95
6	Bayesian-based Absolute Positions Estimation for the Nearby Vehicles through Vehicle-to-Vehicle Communications	97
6.1	Framework	97
6.2	Detailed design of the proposed approach	98
6.2.1	Absolute position estimation for the target vehicle	98
6.2.2	Centers of sampling areas determination	99
6.2.3	Bayesian-based positions estimation	100
6.2.3.1	Scenario 1: One nearby vehicle	101
6.2.3.2	Scenario 2: num nearby vehicles (num > 1)	103
6.3	Experiments	105
6.3.1	Experiments setup	105
6.3.1.1	Dataset	105
6.3.1.2	Accuracy evaluation:	106
6.3.1.3	Parameters settings	107
6.3.1.4	Methods for comparison	107
6.3.2	Experimental results	107
6.3.2.1	The number of nearby vehicles	108
6.3.2.2	Localization accuracy	108
6.3.2.3	Sensitivity analysis	110
6.4	Summary	113
7	Conclusions and Future Work	115
7.1	Conclusions	115
7.2	Future Work	117
7.2.1	Map matching for route inference	117
7.2.2	Map matching for vehicle localization	117
A	Estimation of the travelling areas of nearby vehicle	119

B Determination of PDF of displacement difference	121
List of Publications	125
Bibliography	127

List of Figures

1.1	Map matching for route inference	3
1.2	Map matching for vehicle localization	4
2.1	HMM framework for map matching	14
2.2	Illustration of inference delay	16
3.1	INC-RB framework	32
3.2	RMF comparison with different trajectory features	46
3.3	Average inference delay under different window size	47
3.4	Average RMF under different window size	48
3.5	Impact of rollback depth dep under different sampling rates	49
4.1	Framework	54
4.2	Road network	63
4.3	Distribution fittings for linear route-related features	67
4.4	Distribution fittings for linear trajectory-related features	68
4.5	Distribution fittings for nonlinear route/trajectory-related Features	68
4.6	RMF for map matching algorithms for complex trajectories	70
4.7	ROC of the weights for different travelling cases	72
5.1	Framework	75
5.2	Approximate rectangular area	78
5.3	New York city	84
5.4	Average execution time with the number of nearby vehicles	89
5.5	Sensitivity analysis	92
5.6	Three driving cases	93
6.1	Framework	98
6.2	US 101 study area	106
6.3	Number of detected nearby vehicles given three distance thresholds	109
6.4	Average position error given three distance thresholds	110
6.5	Average position error with different value of α	111
6.6	Average position error with different value of σ_g	112
6.7	Average position error with different value of σ_d	113
6.8	Average position error with different value of σ_a	113

List of Tables

2.1	Summary of map matching algorithms for route inference	21
2.2	Summary of travelling features	22
2.3	Localization approaches based on V2V communications	24
2.4	Localization approaches based on V2V communications	27
3.1	Number of trajectories with different features	40
3.2	Inference Accuracy	45
3.3	Average Execution Time of 25 Cities	47
3.4	Ratios of 25 cities	50
4.1	Trajectory categories	64
4.2	Parameters of PDF for the linear combination of travelling features	66
4.3	Parameters of PDF for the nonlinear combination of travelling features	66
4.4	Weights estimates from our proposed approach and the distribution fitting results	69
4.5	map matching accuracy	69
5.1	Average accuracy results with respect to the number of nearby ve- hicles using different methods	87
5.2	Search space in terms of nearby lanes, candidate lanes and candidate zones	89
5.3	Sensitivity analysis in terms of ρ	91
5.4	Sensitivity analysis in terms of ω	91
5.5	Average accuracy results with respect to the number of the nearby vehicles under three cases	94
6.1	Average position error and variance	110

List of Notations

F	the set of travelling features
θ	the set of weights
g_i	the i^{th} GPS position
l	a travelling route
s	a road segment
L_i	the inferred partial route given the GPS position g_i
$sc(L_i)$	the selection score of L_i
α	balance factor
dep	the rollback depth
RT	the ranking table for route inference
RF	the rollback frequency
RI	the rollback improvement rate
G_l	the GPS trajectory corresponding to the travelling route l
$\mathbb{PR}$	the popularity-based training set
h	the number of travelling features
$ A^P $	the set that stores the popular route pairs
T_i	the target vehicle with the i^{th} GPS measurement
P_T^i	the real position of the target vehicle T_i given the i^{th} GPS position
g_T^i	the i^{th} GPS position of the target vehicle
num	the number of the nearby vehicles detected by the target vehicle
NL_i	the nearby lanes for the target vehicle T_i given the i^{th} GPS measurement
CL_i	the candidate lanes for the target vehicle T_i given the i^{th} GPS measurement
CZ_i	the candidate zones for the target vehicle T_i given the i^{th} GPS measurement
SP_i^m	the m^{th} sampled position of the target vehicle from CZ_i

P_i^{jm}	the corresponding position of the j^{th} nearby vehicle given SP_i^m
$SC(SP_i^m)$	the selection score of the sampled position SP_i^m of the target vehicle
N_i^j	the j^{th} nearby vehicle detected by the target vehicle T_i
P_N^{ij}	the real position of the nearby vehicle N_i^j
g_N^{ij}	the GPS position of the nearby vehicle N_i^j
B_T^i	the position bias of the target vehicle T_i caused by common errors
NB_T^i	the position bias of the target vehicle T_i caused by non-common errors
B_N^{ij}	the position bias of the nearby vehicle N_i^j caused by common errors
NB_N^{ij}	the position bias of the nearby vehicle N_i^j caused by non-common errors
$\hat{d}(T_i, N_i^j)$	the measured relative distance between the target vehicle T_i and nearby vehicle N_i^j
$\hat{a}(T_i, N_i^j)$	the measured bearing angle between the target vehicle T_i and nearby vehicle N_i^j
$\hat{PD}(T_i, N_i^j)$	the measured position difference between the target vehicle T_i and nearby vehicle N_i^j
$d(T_i, N_i^j)$	the error-free relate distance between the target vehicle T_i and nearby vehicle N_i^j
$a(T_i, N_i^j)$	the error-free bearing angle between the target vehicle T_i and nearby vehicle N_i^j
$PD(T_i, N_i^j)$	the real position difference between the target vehicle T_i and nearby vehicle N_i^j
ε_d	the noise of measured relative position $\hat{d}(T_i, N_i^j)$
ε_a	the noise of measured bearing angle $\hat{a}(T_i, N_i^j)$
ε^{PD}	the estimated error of the measured position difference $\hat{PD}(T_i, N_i^j)$
R	the radius of the estimated travelling area of nearby vehicles
P_{Sam}^{ij}	the center position of the sampling area for the nearby vehicle N_i^j
$\bar{P}_T^i$	the estimated position of the target vehicle T_i
$\bar{P}^j$	the estimated position of the nearby vehicle N_i^j

List of Terminologies

Road segment	a directed edge with one start node, one end node, and a length value
Route	a sequence of connected road segments
Candidate segments	a set of road segments that the GPS position most likely belongs to
Local route	a possible route travelled by the vehicle between two consecutive GPS positions
Emission probability	the probability for a GPS position mapped to a road segment
Transition probability	the probability for two mapped road segments to form a local route
Partial GPS trajectory	a sequence of GPS positions up to the current time
Inferred partial route	a route that a vehicle is believed to travel given the observation of a partial GPS trajectory
Route popularity	the number of travelling traces on a given travelling route
Nearby lanes	a set of lanes that the GPS position most likely belongs to
Candidate lanes	a set of lanes after filtering our impossible lanes from nearby lanes
Candidate zones	the zones that contain the real position of the vehicle
Displacement	the vector pointing from a vehicle position to its measured GPS position
Displacement difference	the difference of two displacements

Chapter 1

Introduction

This chapter firstly gives the motivation and background of the map matching algorithms for route inference and vehicle localization respectively. Then, the research contributions and organization of this thesis are provided.

1.1 Motivation and Background

In recent years, traffic congestion and accidents have become common problems worldwide. According to the World Health Organization, around 1,350,000 people die each year worldwide due to road traffic accidents, and injuries from such traffic accidents are the leading cause of death for people aged 5 to 29 [98]. For example, in the Kingdom of Saudi Arabia, the number of death caused by road accidents has increased from 17% to 24% per 10 million people over the last decade [2]. In addition to this high cost to human health, traffic accidents have large financial costs. According to [13], global congestion costs were estimated to be as high as USD 1 trillion in 2013. In U.S.A. there were around 40,000 traffic-related fatalities in 2018 and an estimated cost of USD 412.8 million due to injuries and property damage [98].

To reduce traffic congestion and accidents, most countries have turned to Intelligent Transportation Systems (ITS) [18, 43, 70] for innovative traffic management services. In general, ITS systems comprise a set of applications [1]. These applications are typically used for navigation, driving assistance, and analysis of driving

behavior. Navigation systems [75, 101], which have become ubiquitous in recent years, provide locations and route suggestions to drivers in real time. Advanced Driver Assistance Systems (ADAS) [20, 41] assist drivers while driving, alerting them to potential accidents and collisions. As part of Travelling Behavior Analysis (TBA) [22, 28], drivers' driving behaviors in varying traffic conditions are analyzed to inform city traffic management. To meet the requirements of these applications, travelling routes and absolute positions of vehicles are needed. For example, TBA leverages data on the travelling routes of vehicles to analyze driving behaviors, such as lane changes, accelerating, and braking [44]; while the absolute positions of vehicles are the fundamental outputs of navigation systems.

At present, the Global Positioning System (GPS) is widely used to obtain the absolute positions of vehicles in real time for ITS applications. The measured absolute positions determined by GPS are called GPS positions. When the GPS positions have a high measurement frequency, the travelling routes can be inferred from GPS positions reported by location-based devices [99]. However, there are two major limitations of GPS positions: measurement errors and low sampling rates.

- **GPS measurement errors:** Accurate measurement of the positions of vehicles is required to determine accurate driving decisions in ITS applications. For example, an error of few decimeters can cause a vehicle to localize itself on the wrong side of the road or can cause an accident [44]. According to [67, 80], the accuracy of GPS measurements depends on the quality of the measurements of the pseudo distance between satellites and GPS receivers. Satellite blockage due to high-rise buildings, bridges, and trees, etc. can result in large measurement uncertainty. As a result, the measurement errors of GPS can be as large as 10 m [26, 67, 81].
- **Low sampling rate:** According to [31, 32, 52, 54], many vehicles are equipped with GPS devices, and their GPS positions are periodically reported to data centers. The duration of time between two consecutive GPS measurements is called the sampling rate. In general, the sampling rates should be high (e.g., 2-5 seconds) for route inference [54, 99]. However, due to user concerns regarding privacy and energy consumption, the sampling

rates of GPS positions can be low (e.g., > 2 minutes). Using a dataset collected from 10,357 taxis in Beijing during the period of February 2-8, 2008, as an example, the average sampling rate was found about 177 seconds [96].

To estimate the absolute positions and travelling routes of vehicles from GPS measurements, map matching algorithms are proposed. According to [39], the basic form of map matching is the process of assigning the measured position of a vehicle to the position on a digital road map. In this thesis, the vehicle localization belongs to this form of map matching. Another form of map matching that aims to infer the travelling routes [32, 54] of vehicles is also proposed for ITS applications. In this thesis, the travelling routes inference belongs to this second form of map matching.

Due to the high accuracy requirements of ITS applications, vehicle localization is a much more challenging problem than route inference. In this thesis, map matching for vehicle localization is investigated and new localization approaches are proposed to achieve high localization accuracy. In general, given the absolute positions of vehicles after vehicle localization, travelling route can be easily influenced by identifying the road segments travelled by vehicles. In this thesis, we investigate the route inference problem first, assuming that only infrequent GPS positions are available. Then, vehicle localization based on GPS positions is studied.

1.1.1 Map matching for route inference

FIGURE 1.1: Map matching for route inference

Given the GPS trajectories with low sampling rates, the travelling routes are recovered from the map matching algorithm. The process for route inference is shown

in Fig. 1.1, where g_{i-2} , g_{i-1} and g_i are GPS positions at time t_{i-2} , t_{i-1} and t_i . The time duration between two consecutive GPS measurements is called the sampling rate. The yellow dash line represents the inferred route from map matching algorithms. To infer a travelling route, the local routes between consecutive GPS positions are firstly inferred. In Fig 1.1, the local route are the routes between GPS positions g_{i-2} and g_{i-1} , and between GPS positions g_{i-1} and g_i respectively. Then, the travelling routes are inferred by linking consecutive local routes based on an inference mechanism.

1.1.2 Map matching for vehicle localization

FIGURE 1.2: Map matching for vehicle localization

Given a GPS position with a large measurement error, the absolute position of the vehicle is estimated from the map matching algorithm. The map matching process for vehicle localization is shown in Fig 1.2, where g_{i-2} , g_{i-1} , and g_i are GPS positions at time t_{i-2} , t_{i-1} , and t_i respectively. The red dots represent the absolute positions of the vehicle estimated from the map matching algorithm.

To achieve high localization accuracy, map matching algorithms based on Vehicle-to-Vehicle (V2V) communication techniques [30, 85] are prevailing [3, 14, 55, 65] for ITS applications. In V2V communication techniques, the vehicle equipped with an Automated Driving System (ADS) is called the target vehicle. In addition, a Dedicated Short-Range Communication (DSRC) system is used for vehicles to form a vehicular ad hoc network (VANET) [14, 65]. Then, map matching algorithms are proposed to localize the vehicles by fusing the state information (e.g., driving directions and GPS positions) of nearby vehicles [44].

1.2 Objectives

The objectives in this thesis are to propose efficient and accurate map matching algorithms for route inference and vehicle localization respectively.

1.2.1 Map matching for route inference

In this subsection, map matching algorithms and weights estimation approaches for route inference are studied. In addition, the challenges and issues in map matching for route inference are summarized in the following subsections.

1.2.1.1 Online map matching for route inference

Based on the existing inference mechanisms, the existing map matching algorithms are typically classified as *offline* and *online* map matching algorithms [63]. The offline map matching algorithms generate the entire travelling routes by processing the entire sequence of observed GPS positions. Therefore, the offline map matching algorithms are able to provide accurate route inference, but need more computational time. In contrast, the online map matching algorithms incrementally infer the travelling routes using the partial GPS measurements. Therefore, the online algorithms usually have much shorter response time than offline map matching algorithms, and are important for the latency-sensitive applications (such as the real-time personalized route guidance).

The existing online map matching algorithms are based on *time-window* models [25, 86], where the latest GPS measurements within some time intervals are processed to infer the travelling routes. In other words, the travelling routes are conservatively generated after the GPS measurements are well analyzed to ensure the quality of the inference (e.g., upon meeting some user-defined convergence conditions). However, there are two limitations for window-based models. Firstly, since the algorithms conservatively generate the travelling route only after enough amount of GPS measurements are analyzed, the time gap between the observed GPS positions and the inferred route always exists. This brings the *inference delay*, which is the period between the time when the latest GPS position is observed and the time when the latest travelling route is inferred. Secondly, for each pair of two

consecutive GPS measurements, the algorithms usually require the determination of all the feasible local routes between two GPS positions. This leads to high computational costs for a large-scale complex road network.

To overcome the above limitations of the existing work, our objective is to design an online map matching algorithm to reduce the inference delay as well as the number of feasible local routes determined for route inference. To achieve the objective, an opportunistic inference strategy that is able to produce an inferred route up to the latest observed GPS position is proposed. Moreover, in case when an improper travelling route is inferred from our opportunistic inference strategy, a rollback mechanism that is able to correct the already inferred route is designed.

1.2.1.2 Weights estimation for route inference

In route inference, driving cost models are usually used for local routes determination. A local route is determined as the one with the smallest driving cost among the multiple possible routes. In general, a driving cost model is formulated with travelling features (e.g., road length [95], road type [38], etc), and driving preferences are modelled as weights that quantify the importance of the travelling features [60, 92]. As a result, the driving cost is largely affected by the weights given the available travelling features. To achieve a better understanding of driving behaviors for ITS applications, it is necessary to estimate the weights for map matching algorithms.

At present, the approaches of weights estimation for map matching algorithms are classified as empirical setting [33, 95] and data-driven approaches [38, 58]. The empirical setting approaches determine the weights based on specific driving cost models. However, the weights need to be re-estimated empirically when driving cost models are changed. Alternatively, the data-driven approaches determine the weights from the selected historical routes of vehicles. However, since traffic conditions are dissimilar in different areas and drivers may have misjudgments to these traffic conditions, therefore, estimation bias may be introduced from inappropriate training routes selection. In addition, some driving cost models are formulated with non-normalized travelling features and unbounded weights [50, 95], which increase difficulties to determine the optimal values of weights using the existing data-driven approaches.

Unlike the existing empirical setting approaches with low generality, and the existing data-driven approaches with estimation bias due to inappropriate routes selection, a distribution-based weights estimation approach is proposed. In the proposed approach, *routes popularity* are considered, where the popularity is defined as the number of travelling traces on a given travelling route. Since the popularity is generated from the drivers in a city scale, our weights estimation approach is able to decrease the estimation bias. In addition, our proposed approach can be used as an estimation framework, which is suitable for both linear and nonlinear driving cost models with either normalized or non-normalized travelling features.

1.2.2 Map matching for vehicle localization

In this subsection, vehicle localization approaches based on V2V communication techniques are studied. Then, the challenges and issues in map matching for vehicle localization are summarized in the following subsections.

1.2.2.1 Localization for the target vehicle

The position estimation for the target vehicle has been a crucial component for ITS applications, such as navigation systems [75, 101] which provide drivers with their absolute locations and route suggestions in real time.

In the existing approaches based on V2V communications, the position of the target vehicle is usually determined in a predefined area around the GPS position of the target vehicle [68, 77]. The area forms the *search space* for the position of the target vehicle. Sampled positions in the search space are called *candidate positions*. Existing approaches usually assign a selection score for each candidate position using the state information of nearby vehicles and determine the position of the target vehicle using the candidate position with the highest selection score [24, 26, 27, 67, 68, 77]. Since the search space needs to be large in order to contain the unknown real position of the target vehicle due to GPS measurement errors, the localization efficiency is usually low. In addition, the selection score for each candidate position also needs to be computed using probabilistic strategies. This consequently further increases the computational overheads. Therefore,

the existing localization approaches are in general not feasible for real-time localization. Moreover, most of the existing localization approaches assume that the measurement errors of state information of nearby vehicles are small [26, 27, 68, 77]. However, the measurement errors of state information of nearby vehicles can be large in practice due to the external conditions [17]. As a result, the existing approaches may generate unacceptable localization errors if large measurement error of nearby vehicles' information is present.

To address the above limitations of the existing map matching approaches, we propose an efficient and error-tolerant approach to localize the target vehicle. To achieve high localization efficiency, a pruning-based strategy is proposed to prune the search space for the position of the target vehicle, using the state information of nearby vehicles. To determine the position of the target vehicle from the search space with high localization accuracy and to decrease the influence of measurement errors, we sample the candidate positions of the target vehicle from the pruned search space and propose a displacement-based selection strategy to determine the selection score for each candidate position.

1.2.2.2 Localization for the nearby vehicles

The positions estimation of the nearby vehicles from the target vehicle has been a crucial consideration in advanced driver assistance systems and automated driving systems [78, 81, 90]. However, the measured relative positions of the nearby vehicles are not adequate to meet the requirements of the applications, such as collision avoidance [78, 90] and vehicle tracking [81]. For example, it is necessary to estimate the absolute positions of the nearby vehicles for the movement prediction to avoid collisions [90, 93]. For vehicle tracking system, the absolute positions estimation of the nearby vehicles plays a fundamental role for analysing the trajectories of the nearby vehicles [81].

In the existing localization approaches based on V2V communications, the state information of the nearby vehicles generally includes the geometry information around the nearby vehicles and the trajectory information of the nearby vehicles. The geometry information around the nearby vehicles (e.g., the width and curvature of the lanes) is utilized to reduce the measurement errors of the relative positions of the nearby vehicles [17, 36, 48], and the accuracy of the absolute

position estimation of the nearby vehicles can benefit from the improved relative position. However, the existing work is generally based on the empirical estimation by considering the specific road geometries (e.g., straight lanes and symmetrical curved lanes). Therefore, when the real road network is complex and diverse [4], the localization accuracy of the existing work using the geometry information may get worse [48]. Approaches are also proposed to estimate the absolute positions of the nearby vehicles by leveraging their historical trajectories (e.g., the positions and velocities over the past period of time) [81, 90, 91]. However, since the historical trajectories of the nearby vehicles may not be obtained by the target vehicle under certain circumstances, this consequently limits the application of these localization approaches [90].

To achieve our objective, a Bayesian-based localization approach is proposed, where the GPS measurements and relative positions of the nearby vehicles are considered as the 'prior' information, and the absolute positions of the nearby vehicles are considered as the 'posterior' information. The absolute position of each detected nearby vehicle is estimated using the Maximum A Posterior (MAP) estimation, where candidate positions are sampled for each nearby vehicle and the posterior probability is calculated for each candidate position using the GPS measurements and relative positions. In the existing work, candidate positions are sampled around the GPS position of each nearby vehicle [67, 68]. However, due to the existence of GPS measurement errors, these candidate positions may deviate from the real positions of the nearby vehicles, this consequently reduces the localization accuracy. In our study, we propose a strategy to determine the center of sampling area for each nearby vehicle by leveraging the GPS measurements and the absolute position of the target vehicle, and the measured relative position, instead of simply using the GPS position. As a result, the localization accuracy improves.

1.3 Summary of Contributions

Our main contributions can be summarized as follows:

- An opportunistic online map matching algorithm is proposed for route inference. Unlike the existing online map matching methods which often experience an inference delay between the GPS measurements and inferred

travelling route, our opportunistic algorithm is able to produce an immediate inference when a new GPS point becomes available. In addition, to ensure the inference accuracy, a rollback mechanism is provided to correct the already inferred route when some unusual conditions are detected.

- A distribution-based weights estimation approach for map matching is proposed, where the weights are determined by inferring their distributions. This approach differs from the existing empirical setting approaches that provide low generality, and the existing data-driven approaches that may suffer from the estimation bias due to inappropriate routes selection. Since the weights are derived using route popularity based on city-scale traces, it is able to decrease the estimation bias. In addition, our proposed approach can be used as an estimation framework and is suitable for both linear and non-linear driving cost models with either normalized or non-normalized travelling features.
- An efficient and error-tolerant localization approach is proposed to increase the localization efficiency and accuracy of the target vehicle even in the presence of large measurement errors. Different from the existing approaches with large search space, we propose a pruning-based strategy to prune the search space for the position of the target vehicle to achieve high localization efficiency. In addition, to determine the position of the target vehicle, we propose a displacement-based selection strategy to determine the selection score for each candidate position in order to decrease the influence of measurement errors of relative positions.
- A Bayesian-based localization approach is proposed to estimate the absolute positions of the nearby vehicles by incorporating GPS measurements and relative positions obtained from the target vehicle. Different from the existing approaches, our proposed approach can be applied when the geometry information is diverse and the trajectories of the nearby vehicles are insufficient. Therefore, our proposed approach can be widely applied to real driving scenarios. In addition, to increase the localization accuracy, we propose a strategy to determine the center of the sampling area for each nearby vehicle so that candidate positions are much closer to vehicle's real position than simply using the GPS position of the nearby vehicle as the center.
- Extensive experiments using real GPS trajectories are conducted to investigate the performance of our proposed algorithm for route inference and

approaches for vehicle localization. For route inference, a real dataset consisting of 100 GPS trajectories from 100 cities and a dataset consisting of GPS trajectories from 10,357 drivers are used in the online map matching algorithm and weights estimation approach, respectively. The performance of our algorithm in terms of map matching accuracy and inference delay relative to the state-of-the-art algorithms is then investigated. For vehicle localization, simulated trajectories from New York City and real trajectories of vehicles travelling on U.S. Highway 101 are utilized. The performance of our proposed approaches in terms of localization accuracy and efficiency is then investigated relative to the state-of-the-art approaches.

1.4 Outline of the Thesis

This thesis is organized into seven chapters

Chapter 2 provides a review of map matching for route inference and vehicle localization. Details regarding the research gaps are further elaborated based on the review. In Chapter 3, the proposed online map matching algorithm for route inference is introduced. In Chapter 4, the distribution-based weights estimation approach is introduced. In Chapter 5, the proposed efficient and error-tolerant localization approach is provided, where the absolute position of the target vehicle is estimated. In Chapter 6, the proposed Bayesian-based localization approach that localizes the nearby vehicles is provided. In Chapter 7, the conclusions and future work are discussed.

Chapter 2

Literature Review

This chapter provides a review of the literature on map matching for route inference and vehicle localization respectively. For route inference, route inference mechanisms and weights estimation approaches are summarized. For vehicle localization, approaches to localize the target vehicle and nearby vehicles are presented. Then, the research gaps are identified, which serve as the basis for the following chapters.

2.1 Map matching for route inference

In this section, the *offline* and *online* inference mechanisms of map matching are firstly discussed. Then, *weights estimation approaches* are summarized. After that, the existing map matching algorithms for route inference are summarized.

2.1.1 Offline map matching algorithms

The offline map matching algorithms infer the entire travelling routes of vehicles by processing the entire GPS trajectories all together. Therefore, the offline map matching algorithms are used for trajectory-based applications, such as travelling behaviour analysis [72, 89]. In the existing offline map matching algorithms, HMM is a widely used technique. In HMM, *emission probability* and *transition probability* are considered for route inference, where the emission probability refers to

the probability for a GPS position mapped to a road segment and the transition probability refers to the probability for two mapped road segments to form a local route.

The framework of HMM-based map matching algorithms is shown in Fig 2.1, where g_{i-4} , g_{i-3} , g_{i-2} , g_{i-1} , and g_i are the GPS positions from time t_{i-4} to t_i . For the GPS position g_i , s_i^1 , s_i^2 , and s_i^3 are the road segments near to g_i . The emission probabilities for these segments are determined from the geometry information, such as the distance between these road segments to the GPS positions. The dashed lines refer to the local routes between each pair of these road segments and the transition probabilities for these local routes are then determined by considering the geometry information or temporal information. Finally, the inferred route shown as the red arrow line (a sequence of connected local routes), can be inferred by maximizing the product of the emission and transition probabilities through dynamic programming. The details of HMM for route inference can be found in [58].

FIGURE 2.1: HMM framework for map matching

Based on the HMM framework, the algorithm proposed in [58] is the first work to infer the travelling routes of drivers, where the emission probabilities are determined from the distance between the road segments and GPS positions and the transition probabilities are determined from the length differences between the local routes and GPS trajectories. In this algorithm, the transition probability is assumed to follow an exponential distribution. The algorithm performs well with a high sampling rate of GPS measurements, but the accuracy decreases with a low sampling rate [94] because using the length differences to determine the transition

probabilities is insufficient to capture the driving behaviors of drivers with a low sampling rate. To increase map matching accuracy, a feature-based map matching algorithm is proposed in [94], where the local routes are obtained by considering a driving cost model formulated with travelling features (e.g., turning angles) and weights which quantify the importance of these features. Since the travelling features represent the factors that influence drivers' driving decisions, this algorithm achieves high inference accuracy. Similarly, the algorithm proposed in [54] infers the travelling routes of vehicles by considering the spatial and temporal features of road network. For the spatial features, the similarities between reachable routes and GPS trajectories are considered. For the temporal features, the similarities between the average travelling speeds of vehicles and speed constraints of road segments are considered.

2.1.2 Online map matching algorithms

With the development of real-time ITS applications, such as traffic sensing and route recommendation, it is critical to immediately infer the travelling route using incoming GPS data. The existing offline map matching algorithms are not suitable for this purpose. At present, online map matching algorithms are proposed to incrementally infer travelling routes by using partial GPS positions obtained in real time.

The existing online map matching algorithms are based on time-window models [25, 38, 57], where the latest GPS positions within some time intervals are processed to infer the travelling routes. In other words, the travelling routes are conservatively generated when convergence conditions (which will be explained later in this subsection) are met. To infer the travelling route, for each pair of consecutive GPS positions, the feasible routes between two GPS positions need to be determined. This results in high computational costs for a large-scale complex road network. In addition, since the online map matching algorithms conservatively generate the travelling route after enough amount of GPS measurements being analyzed, the time gap between the observed GPS positions and the inferred route always exists. This brings the *inference delay*, which is the time period between the time when the latest GPS position is observed and the time when the latest travelling route is inferred [56].

FIGURE 2.2: Illustration of inference delay

Fig. 2.2 illustrates the map matching process based on the time window model with inference delay. When a new GPS position g_i is available, three nearby road segments (i.e., s_i^1 to s_i^3) are prepared. The local routes (i.e., the dashed lines) between g_{i-1} and g_i are first generated based on driving cost models. Among these local routes, the routes shown as the black arrow represent the most probable local routes ending at s_i^1 to s_i^3 respectively. Then, the online Viterbi algorithm [86] is used to trace backward to check for a convergence point, that is, a road segment where all the most probable local routes of s_i^1 to s_i^3 converge. If such convergence point (e.g., s_{i-2}^3) is found, the inferred partial route up to the convergence point can be determined by selecting the series of the most probable local routes (i.e., the red lines) that ends at that convergent point. It is also possible that such convergence point cannot be found at the current step. In such a case, the extension of inferred partial route is postponed until a convergent point is found in the subsequent inferences with new GPS data. Note that this algorithm will fail if a convergence point can never be found.

As illustrated above, the strategy can only update the partial route when the convergence point is found. When a new GPS position g_i is added, the best case is that a convergence point is one of nearby road segments of g_{i-1} . Hence, the partial route can only be updated up to g_{i-1} . Therefore, the inference delay exists. Furthermore, there is no guarantee that a convergence point can be always found at the current step, this further delays the inference process.

According to [87], the inference delay can be decomposed into *algorithm execution time* and *decoding delay*. The algorithm execution time is the time needed for executing a given map matching algorithm and it often depends on the complexity of the designed algorithm and the processing power of running machine. The

decoding delay is the inherent delay that is due to the sliding window strategy [25] used by a given online map matching algorithm. The decoding delay is the most influential component in the overall inference delay and it varies based on different sliding window strategies.

- Fixed sliding window (FSW): for the algorithms which adopt FSW strategy, the partial route can be always obtained by moving the window forward one step. In this strategy, the route is inferred by following the offline inference mechanism using the GPS positions in the window. Therefore, the decoding delay increases with the increase of window size.
- Variable sliding window (VSW): for the algorithms which adopt VSW strategy, the sliding window can shrink and expand depending on the detection of convergence point. Therefore, the decoding delay can vary for different GPS positions, but the overall decoding delay is not affected by the change of window size.
- Bounded variable sliding window (BVSF): for the algorithms which adopt BVSF strategy, the decoding delay is the same as the one in VSW when the window size does not exceed the defined bound and when the window size exceeds the defined bound, the decoding delay will be the same as FSW.

A pioneer online map matching algorithm is proposed in [25] by applying HMM for route inference. In addition, the BVSF strategy is considered to extend the partial routes. An online learning map matching algorithm is proposed in [49]. The algorithm is based on HMM and uses an online learning method to dynamically tune the key algorithm parameters by leveraging a small piece of trajectory data and their matching result. In addition, the algorithm adopts the BVSF strategy to infer the travelling route. In [38], the travelling route is inferred by combining the route generated from HMM and the routes generated from a heuristic inference model with the consideration of travelling features such as travelling time. The partial route is determined through a route choice model [12] by selecting the best route from HMM and heuristic inference model. However, the reassessment of the route choice model inevitably introduces computational costs. Recently, a feature-based online map matching algorithm based on the FSW strategy is proposed in [95]. Similarly to the offline inference mechanism proposed in [94], this

algorithm considers a set of travelling features such as roads length, turning angles, and the distances between road segments and GPS positions.

2.1.3 Weights estimation for route inference

In this subsection, *features classification* is firstly introduced. Then, *driving cost models description* is given. Finally, the existing weights estimation approaches are classified into empirical setting and data-driven approaches and are discussed in details.

Features classification: Based on the existing driving cost models [33, 34, 50, 60], the considered travelling features can be classified into two types:

- **Route-related features:** These features are related to route properties. They can be extracted directly from the road network. For example, the travelling features in terms of routes length, turning angles, left/right turns made by the vehicle at road intersections, and the speed constraints of road segments are considered in the driving cost models in [33, 34, 60].
- **Trajectory-related features:** These features are related to routes and GPS trajectories such as the distance between a route and GPS trajectory. These features can be obtained when both of the routes and corresponding GPS trajectories of vehicles are available. For example, the map matching algorithm proposed in [50] considers the travelling features extracted from the routes and GPS trajectories of vehicles based on spatial positioning accuracy, spatial accessibility, middle-point spatial distribution, and the consistency of driving directions.

Driving cost models description: Given a GPS trajectory G_l , where l is the corresponding route, the driving cost model can be formulated as follows:

$$\begin{aligned} c^* &= F_{cost}(\theta, F, l, G_l) \\ \text{s.t. } \theta_v &> 0, \quad \forall \theta_v \end{aligned} \tag{2.1}$$

where c^* is the travelling cost of the route l . F represents the set of travelling features: $F = \{f_1, f_2, \dots, f_h\}$, where h is the number of considered travelling features. θ is the set of weights: $\theta = \{\theta_1, \theta_2, \dots, \theta_h\}$. To guarantee a non-negative

driving cost value, the weight value for each feature in this thesis is assumed to be non-negative. In general, the driving cost model can be formulated using the linear and nonlinear combinations of travelling features. In addition, the travelling features can be either normalized or non-normalized. For the normalized travelling features, the weights need to be normalized ($0 < \theta_v < 1$ and $\sum_1^h \theta_v = 1$). Non-normalized travelling features will introduce difficulty for weights estimation. More details about the driving cost models can be found in [34, 50, 60, 95] and Chapter 4.

Empirical setting approaches: For the empirical setting approaches, the weights are determined empirically according to the importance of travelling features. The empirical setting approaches are easy to implement, but the weights are determined based on the specific driving cost models.

For non-normalized travelling features, the map matching algorithm proposed in [94, 95] adopts the driving cost model with the nonlinear combination of travelling features in terms of segments length, turning angles, and the distance between road segments to GPS positions. Among these features, the feature of segments length is considered to be more important than the other two travelling features, therefore, a large weight value is set for the feature of segments length. Since the travelling features are not normalized, therefore $\sum_1^h \theta_v = 1$ is not required.

For consideration of normalized travelling features, the map matching algorithm proposed in [33] adopts the driving cost model with the linear combination of travelling features in terms of the historical travelling speeds of vehicles and the speed constraints of road segments. Since the two features are considered to be equally important, the weight for the two features are set to be identical. Similarly, the map matching algorithm proposed in [50] considers four travelling features in terms of spatial positioning accuracy, spatial accessibility, middle-point spatial distribution and the consistency of driving directions respectively. For the above driving cost models, the weights are normalized and empirically determined.

Data-driven approaches: In contrast to empirical setting approaches, data-driven approaches determine the weights by leveraging the historical travelling routes of vehicles. However, since the traffic conditions are dissimilar in different areas and drivers may have misjudgments to these traffic conditions, therefore the estimation bias are introduced due to inappropriate routes selection.

For non-normalized travelling features, the maximum likelihood estimation (MLE) strategy is widely adopted in the existing data-driven approaches. The real routes of vehicles and alternative routes with the same start/end segments are leveraged for weights estimation. The weights are determined by maximizing the probability of the real route being selected among all alternative routes. Among the existing map matching algorithms that use this strategy, the algorithm proposed in [38] adopts a driving cost model with the linear combination of travelling features such as road types. The map matching algorithms proposed in [35] and [51] adopt the conditional random field model (CRF) for route inference. In these algorithms, the travelling features are extracted from the routes and GPS trajectories. However, the efficiency of MLE is low since multiple alternative routes need to be considered for weights estimation.

For normalized travelling features, the map matching algorithms proposed in [84] and [62] estimate the weights through non-linear minimization (NM) [21] and genetic algorithm (GA) [16]. In NM and GA, the objective function is built according to the mapping accuracy from GPS points to road segments. Then the optimal weights can be obtained from the objective function. However, the two approaches cannot be used for non-normalized travelling features because the weights will become unbounded.

Based on the above analysis in this section, the summary of map matching algorithms in this thesis is provided in Table 2.1 and Table 2.2. In the two tables, the map matching algorithms, the inference mechanism, the weights estimation approaches, the adopted sliding window strategy, and the travelling features considered in their cost models are summarized.

2.2 Map matching for vehicle localization

In this section, the approaches for vehicle localization are provided. Firstly, *GPS measurement errors* and the *measurement errors of relative positions* are elaborated. Then, the review on the existing localization approaches based on V2V communications is discussed.

TABLE 2.1: Summary of map matching algorithms for route inference

Algorithms	Inference Mechanism	Weights Estimation Approaches	Sliding Window
[33, 54, 94]	Offline	Empirical setting	$\sim$
[35, 50, 51]	Offline	MLE	$\sim$
[84]	Offline	NM	$\sim$
[62]	Offline	GA	$\sim$
[25, 49]	Online	Empirical setting	BVSW
[38]	Online	MLE	VSW
[95]	Online	Empirical setting	FSW
[37]	Online	Empirical setting	VSW

2.2.1 GPS measurement errors

In general, the measurement errors of GPS are resulted from common error sources (e.g., atmospheric delay, satellite clock, and ephemeris errors) and uncommon error sources (e.g., the noise of GPS receivers). When vehicles are driving in a vicinity, these vehicles are assumed to have the GPS data coming from the same constellation of satellites. Therefore, the GPS position bias caused by common errors of these vehicles are generally correlated [67, 80]. However, as non-common errors vary from vehicle to vehicle, the bias in GPS positioning caused by uncommon errors is usually modelled following a vehicle-dependent zero-mean Gaussian distribution [68, 74]. So, given the target vehicle T_i , the measured GPS position of the target vehicle T_i can be formulated as:

$$g_T^i = P_T^i + B_T^i + NB_T^i \quad (2.2)$$

where P_T^i represents the real position of T_i , B_T^i represents the position bias caused by common errors, NB_T^i represents the position bias caused by non-common errors. NB_T^i is modelled following a Gaussian distribution $N(0, \sigma_{NB_T^i}^2)$, where $\sigma_{NB_T^i}$ is the standard deviation of NB_T^i .

TABLE 2.2: Summary of travelling features

Algorithms	Travelling Features
[54]	the length of road segments, the speed constraints of road segments, and the travelling speeds of vehicles
[33]	travelling speeds of vehicles and the speed constraints of road segments
[94]	the length of road segments, turning angles, the distance between GPS positions and road segments
[50]	spatial positioning accuracy, spatial accessibility, middle-point spatial distribution and the consistency of driving directions
[35]	the length of road segments, the number of stop signs, and the speed constraints of road segments
[51]	spatial positioning accuracy, spatial accessibility, middle-point spatial distribution and the consistency of driving directions
[84]	the angle differences between the driving direction and the directions of road segments, the distance between GPS positions and road segments
[62]	the angle differences between the driving direction and the directions of road segments, the distance between GPS positions and road segments
[25]	the travelling speeds of vehicles, the length of road segments
[49]	the number of connections between road segments, the distance between GPS positions and road segments
[38]	the free-flow travel time of vehicles, the number of traffic signals, the class of road segments, the number of class changes
[95]	the length of road segments, turning angles, the distance between road segments and GPS positions
[37]	the travelling time of vehicles, the length of road segments

Similarly, the measured GPS position of its j^{th} nearby vehicle N_i^j can be written as:

$$g_N^{ij} = P_N^{ij} + B_N^{ij} + NB_N^{ij} \quad (2.3)$$

where P_N^{ij} represents the real position of N_i^j , B_N^{ij} represents the position bias caused by common errors, NB_N^{ij} represents the position bias caused by non-common errors. NB_N^{ij} follows a Gaussian distribution $N(0, \sigma_{NB_N^{ij}}^2)$, where $\sigma_{NB_N^{ij}}$ is the corresponding standard deviation of NB_N^{ij} .

2.2.2 Measurement errors of relative positions

In this thesis, it is assumed that the relative positions (relative distances and bearing angles) of nearby vehicles can be obtained through on-board sensors, such as lidar and radar. According to [17], the measurement errors of relative positions can be large due to the external conditions. For example, the measurement error of relative distance usually ranges between 0.02 m and 0.5 m in good environmental conditions for lidar systems, but the performance starts to dwindle in the snowy, foggy, rainy and dusty weather conditions due to the influence of aerosols [17, 23, 88]. For radar systems, the field of view is usually small, which is around 10° . The angular errors usually ranges between 0.5° and 5° , which typically yields the distance errors from 87 cm to 4 m, respectively [17, 88]. According to [26, 40], the measured relative distance and bearing angle between the target vehicle T_i and nearby vehicle N_i^j can be represented as follows:

$$\begin{aligned} \hat{d}(T_i, N_i^j) &= d(T_i, N_i^j) + \varepsilon_d \\ \hat{a}(T_i, N_i^j) &= a(T_i, N_i^j) + \varepsilon_a \end{aligned} \quad (2.4)$$

where $d(T_i, N_i^j)$ and $a(T_i, N_i^j)$ are the error-free relative distance and bearing angle between the target vehicle T_i and the nearby vehicle N_i^j . The noise ε_d and ε_a are modelled using Gaussian distributions $N(0, \rho^2)$ and $N(0, \omega^2)$ respectively, where ρ and ω are the standard deviations of measurement errors.

2.2.3 Localization for the target vehicle

The existing localization approaches based on V2V communications are summarized in Table 2.3. In the table, the information leveraged for localization from the target vehicle and nearby vehicles is listed. In addition, whether the search space needs to be considered to localize the target vehicle is stated. Finally, the tolerance of the localization approaches for the measurement errors of relative positions is provided.

TABLE 2.3: Localization approaches based on V2V communications

Approaches	Information	Search Space	Tolerance
Error compensation [40]	GPS measurements, Relative positions	No	No
Weight-based [3, 9]	GPS measurements, Relative positions	No	No
Trilateration-based [6, 45, 64]	GPS measurements, Relative positions	No	No
Kalman filter [69], extended Kalman filter [11, 40]	GPS measurements, Relative positions	No	Yes
Sampling-based [24]	GPS measurements, Velocities, Relative positions	Yes	No
Bayesian-based [68, 77]	GPS measurements, Relative positions	Yes	Yes
Particle filter-based [66, 67]	GPS measurements, Road constraints	Yes	Yes

It can be seen from the table that these approaches can be classified into two types based on whether or not the search space is needed. The details of these approaches are given as follows:

Error compensation: For the localization approach proposed in [40], the position of the target vehicle is determined through error compensation by leveraging the GPS measurements and the relative positions of nearby vehicles. To estimate the GPS measurement errors for compensation, the absolute positions of nearby vehicle are firstly estimated from the GPS position of the target vehicle and the measured relative positions. The average GPS positions of nearby vehicles and the

average position of their absolute positions are then calculated. Next, the GPS measurement errors are estimated as the differences between the two average positions. Finally, the positions of the target vehicle can be determined by subtracting the GPS measurement errors from the GPS positions of the target vehicle. Due to the measurement errors, the estimated GPS measurement errors may deviate from the real GPS measurement errors, introducing large localization uncertainty.

Weight-based: The approaches proposed in [3, 9, 47] estimate the position of the target vehicle through the weighted average of the GPS measurements of nearby vehicles, where the weights are determined from the relative positions of nearby vehicles. However, the measurement errors of relative positions lead to inaccurate weights estimation.

Trilateration-based: Trilateration-based approaches are proposed in [6, 45, 64]. The position of the target vehicle is estimated through the geometry calculation using GPS measurements and the measured relative positions of nearby vehicles. However, due to the measurement errors of relative positions, trilateration-based strategies cannot be used in cases with unacceptable measurement errors. In addition, trilateration-based approaches cannot be applied when the number of nearby vehicles is less than three [6].

Kalman filter and extended Kalman filter: With the consideration of the measurement errors of relative positions, Kalman filter (KF) and extended Kalman filter (EKF) approaches are used for vehicle localization [11, 40, 69]. In general, the position of the target vehicle is determined by fusing the information of the target vehicle and nearby vehicles, such as the previous positions of the target vehicle, GPS measurements and the relative positions of nearby vehicles. However, the accuracy of KF and EKF approaches are affected by the estimation of previous positions [5, 69]. Therefore, these strategies are not suitable for the vehicle localization without precise motion models. In addition, as it is assumed that the measurement errors of relative positions follow Gaussian distributions, the existing KF and EKF approaches can tolerate the measurement errors of relative positions.

Sampling-based: The localization approach proposed in [24] determines the position of the target vehicle based on a sampling-based strategy. The candidate positions for the target vehicle are firstly sampled from the predetermined search

space. Then, the likelihood for each candidate position is determined by considering the velocity variances of nearby vehicles, GPS measurements, and the relative positions of nearby vehicles. In their approach, the candidate position with the maximum likelihood is selected as the optimal position of the target vehicle. However, since the measurement errors of relative positions are not accounted for in the localization process, the approach has no tolerance to the measurement errors of relative positions.

Bayesian-based: Different from KF and EKF approaches, Bayesian-based localization approaches are proposed so that the estimation of the target vehicle's previous position is not needed. In Bayesian-based approaches, the GPS measurements and relative positions of nearby vehicles [68, 77] are leveraged. In general, the position of the target vehicle is determined through the maximum a posteriori (MAP) estimation, where the candidate positions of the target vehicle are sampled from a search space. The probability for each candidate position is determined by considering GPS measurements and the relative positions of nearby vehicles. In [68], the relative distances between the target vehicle and each nearby vehicle are considered. In [77] the bearing angles between the target vehicle and nearby vehicles are additionally considered. However, since the search space can be large and probability needs to be computed for each candidate position, the localization efficiency is low. In addition, as it is assumed that the measurement errors of relative positions follow Gaussian distributions, the existing Bayesian-based approaches can tolerate the measurement errors [17, 66]. The details of Bayesian-based localization approaches can be found in Chapter 6 and [68, 77].

Particle filter-based: Similar to Bayesian-based approaches, particle filter-based approaches are proposed to determine the position of the target vehicle through the particle filter strategy [83] by considering the GPS measurements and the geometry information around nearby vehicles [66, 67]. In this approach, candidate positions (particles) of the target vehicle are firstly sampled from the search space distributed around the GPS position, then the weights for particles are updated with respect to the road constraints of both the target vehicle and nearby vehicles. Finally, the position of the target vehicle is determined as the weighted average position of the particles. The details of this localization approach is given in [67]. Since the relative positions of nearby vehicles are not considered for localization, the existing approach can tolerate the measurement errors of relative positions.

However, similar to Bayesian-based approaches, the large search space and the weight determination for each particle decrease the localization efficiency.

2.2.4 Localization for the nearby vehicles

Different from the localization for the target vehicle, which is a self-localization process by using the state information of nearby vehicle, the localization for the nearby vehicles is processed by the target vehicle. Based on the existing approaches that use V2V communications to localize the nearby vehicles detected by the target vehicle, the information obtained from target vehicle used for localization includes the geometry information around the nearby vehicles and the trajectory information of the nearby vehicles. The summary of the existing localization approaches based on V2V communications is given in Table 2.4, where the information utilized for localization and the limitations of these approaches are stated.

TABLE 2.4: Localization approaches based on V2V communications

Approaches	Information	Limitations
Geometry-based [7, 17, 36, 48]	GPS measurement of the target vehicle, Relative positions, Geometry information	Unsuitable for complex and diverse road network
Trajectory-based [46, 73, 81, 90, 91]	Historical trajectories of the nearby vehicles	Cannot be applied when the historical trajectories of the nearby vehicles are unavailable

Geometry-based: The geometry-based approaches are performed to infer the relative positions of the nearby vehicles. Then, the absolute positions of the nearby vehicles can be determined from their relative positions. The existing approaches utilize the geometry information around the nearby vehicles (e.g., the width and curvature of the lanes) through V2V communications to reduce the measurement errors of the relative positions of the nearby vehicles [7, 17, 36, 48]. In general, they are usually based on the empirical estimation by considering the specific road geometries (e.g., straight lanes and symmetrical curved lanes). For example, the strategy proposed in [36] estimates the relative position of nearby vehicles considering the geometry information such as road width and curvature. Therefore, when

the real road network is complex and diverse [4], the localization accuracy of the existing work using the geometry information is low [48].

Trajectory-based: The trajectory-based approaches are proposed to estimate the absolute positions of the nearby vehicles by leveraging their historical trajectories (e.g., the positions and velocities over the past period of time). The existing approaches are commonly based on regression models [46], Bayesian networks [73], Hidden Markov model [19], Random forest classifiers [71], and Kalman filter [81]. Among the above strategies, the Kalman filter-based strategies are commonly adopted to estimate the absolute positions of the nearby vehicles. In Kalman filter-based strategies, the information of the target vehicle and nearby vehicles, such as the previous positions of the nearby vehicle, GPS measurements and the relative positions of nearby vehicles are fused to determine the positions of the nearby vehicles. However, the application of these localization approaches is limited by the fact that the historical trajectories of the nearby vehicles may not be always obtained by the target vehicle under certain circumstances [90]. For example, when new nearby vehicles are detected by the target vehicle in a merging road network, the historical trajectories of these nearby vehicles would not be available for the target vehicle.

2.3 Research Gaps

Based on the review in this chapter, several key research gaps have been identified, which serve as the basis for the research in this thesis. The research gaps of map matching for route inference and vehicle localization are summarized in the following subsections.

2.3.1 Map matching for route inference

As discussed in section 2.1.2, the major limitation of window-based models is the inference delay between the GPS measurement and the inferred travelling route. To reduce the inference delay, an online map matching algorithm (referred to as *INC-RB*) that is able to produce the partial route up to the latest GPS position is proposed. In addition, to ensure the inference accuracy, a rollback mechanism is

provided in our proposed map matching algorithm to correct the already inferred route when some unusual conditions are detected. The details of the proposed online map matching algorithm can be found in Chapter 3.

For the weights estimation approaches discussed in section 2.1.3, the empirical setting approaches have low generality and the data-driven approaches may introduce estimation bias due to inappropriate training routes selection. To address the gaps, we propose a distribution-based weights estimation approach (referred to as *DBWES*). In our approach, routes popularity generated from drivers in a city scale is leveraged to decrease the estimation bias. In addition, our proposed approach has high generality and can estimate the weights for both linear and non-linear driving cost models with either normalized or non-normalized travelling features. The details of DBWES can be found in Chapter 4.

2.3.2 Map matching for vehicle localization

Different from the existing localization approaches discussed in section 2.2.3, where the position of the target vehicle is determined from a large search space and the calculation of selection scores for candidate positions is largely affected by the measurement errors of relative positions, we propose an efficient and error-tolerant approach to localize the target vehicle (referred to as *E²T-CVL*). To achieve high localization efficiency, a pruning-based strategy is proposed to prune the search space for the position of the target vehicle. In our pruning-based strategy, the search space is narrowed by identifying the area where the target vehicle cannot be located, which is termed the impossible area, using road geometry and relative positions between the target vehicle and nearby vehicles while considering measurement errors. Moreover, we propose a displacement-based selection method to determine the selection score for each candidate position in order to decrease the influence of measurement errors of relative positions, where the displacement is defined as the vector pointing from a vehicle position to its measured GPS position. In the calculation of the displacement difference between the target vehicle and its nearby vehicles, the measurement errors of relative positions can be largely cancelled out. The details of the pruning-based strategy and the displacement-based selection method are provided in Chapter 5.

For the existing approaches to localize the nearby vehicles based on V2V communications as discussed in section 2.2.4, the diversity of the geometry information and insufficiency of the trajectories of the nearby vehicles can restrict the application of these approaches. To address the research gaps, we propose a Bayesian-based approach that estimates the absolute position of the nearby vehicles by sampling from candidate positions. Our proposed approach can be applied when the geometry information is diverse and the trajectories of the nearby vehicles are insufficient. As a result, the proposed algorithm can be widely applied to real driving scenarios. Furthermore, to increase the localization accuracy, we proposed a strategy to determine the centers of sampling areas of the nearby vehicles by leveraging the GPS measurements, and the measured relative position of the nearby vehicles. In this manner, the candidate positions are much closer to their real positions than simply using the GPS position as in existing approaches. More details of the Bayesian-based localization approach can be found in Chapter 6.

Chapter 3

Incremental Route Inference from Low-sampling GPS Data: An Opportunistic Approach to Online Map Matching

This chapter describes the proposed online map matching algorithm (*INC-RB*). The algorithm framework, the detailed algorithm design, the experiments and results are provided in the following sections.

3.1 Framework

The overall design of our online map matching framework is shown in Fig. 3.1. It consists of two stages: *offline preprocessing* and *incremental online inference*. To speed up the online inference process, a road network analysis and transition cost initialization module are introduced at the offline preprocessing stage. In the road network analysis module, the network data in a given area of interest is first extracted and the road network information is then analyzed, aiming to extract road features such as length, turning angle, speed limit, etc. The transition cost initialization module is responsible for assigning the initial cost of the nodes in the extracted network based on the corresponding road features. These transition cost

values will be updated and utilized for determining the transition probability of local routes in the subsequent online inference stage.

The incremental online inference is the key part of our framework. When a new GPS position becomes available, the *nearby segments generation* module is responsible for generating the nearby road segments of the new GPS position. After that, the *local routes search* module is responsible for searching a set of local routes, each of which connects a nearby road segment of the new GPS position to the last road segment in the previous inferred partial route. Then, unlike the HMM map matching process, we adopt a *best-effort method to partial route extension* that aims to produce the partial route up to the newly added GPS position. During this process, the system also checks a set of rollback conditions that examine some abnormal situations in the inference (e.g., no local route is found). If any of such rollback conditions occurs, a rollback mechanism to correct the previous inferred partial route is performed to ensure the quality of inference. Otherwise, the system will proceed to next round of incremental inference when new GSP position becomes available again.

FIGURE 3.1: INC-RB framework

3.2 Detailed design of INC-RB

3.2.1 Offline preprocessing

The purpose of offline preprocessing is to perform some precalculations that are needed by the incremental online inference in the next stage. The analyses performed in this stage are mainly based on the road network information that are available offline. The two main tasks are the road network analysis and transition cost initialization which are described as follows:

Road network analysis: To infer travelling routes efficiently, the area in a given road network that is of interest of the study is first extracted. The network area extraction is performed on the road network data in OpenStreetMap where the road segments are stored in the ESRI Shapefile format. A user can specify the borders of the area as a two-dimensional polygon. For example, we can use a rectangle to specify the area of interest, where the borders of the area can be defined by the longitude and latitude of the four corner points of the rectangle. The road segments within the specified area are then extracted. Each road segment is represented as polyline associated with road attributes including segment name, segment length, longitude and latitude of the nodes in the segment and etc. To estimate the transition cost for route inference, a set of features are evaluated based on the extracted road attributes. The features can include information related to a segment itself such as length, historical traffic flow, speed limit and etc. They can also include the features regarding the relationship between different road segments such as turning angle between two consecutive road segments. It should be noted that we do not limit the choice of features in our framework and researchers can define and choose their own features depending on the availability of different road attributes and target applications. In our current work, we use a set of features including the road segment length and turning angle to initialize the transition cost in the offline preprocessing stage.

Transition cost initialization: This module is responsible for assigning the initial transition cost to the transition points in the extracted road network. The transition point is defined as the intersection that connects one incoming road segment with at least two or more outgoing road segments. The transition cost from the incoming road segment to one of outgoing road segment s_i is calculated based

on the input as the set of features being used: $c^{s_i} = \phi(f_1^{s_i}, f_2^{s_i}, \dots, f_n^{s_i})$ where ϕ denotes a function that returns the cost value based on the given feature values. A smaller transition cost value indicates the road segment s_i is more likely to be included in the inferred travelling route. To ensure the generality of our framework, we do not limit the form of the function ϕ . It can be weighted summation of the input features or non-linear combination of them. We adopt the transition cost assignment method in [94], which is a weighted summation of segment length feature and turning angle feature. The calculated transition costs for every transition point are then saved in our database organized with an R-tree indexing scheme. During the online inference stage, these initial transition costs are retrieved and dynamically updated based on additional information obtained in real time, such as distance to GPS positions.

3.2.2 General procedure for incremental online inference

Incremental online inference is the key part of our framework. It adopts an opportunistic approach which incrementally infers the route by extending the inferred partial route whenever a new GPS position is added. A rollback mechanism for route correction is provided if the current inferred partial route is found to deviate from the actual route based on the examination of a set of rollback conditions. The general procedure of our incremental online inference with rollback (INC-RB) is shown in Algorithm 1

Algorithm 1 takes the inputs of the new GPS position g_i , the partially inferred route L_{i-1} up to the previous GPS position g_{i-1} and a rollback depth that controls the level of rollback. The output is the updated partially inferred route L_i . To achieve this, the nearby road segments of g_i are firstly generated (line 1) using gaussian kernel as defined in Eq. 3.1.

$$p(s_i^f | g_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{dist(s_i^f, g_i)^2}{2\sigma^2}} \quad (3.1)$$

where σ is the estimated standard deviation of GPS measurement noise, $dist(s_i^f, g_i)$ indicates the the minimum great circle (MGC) distance [10] between the road segment s_i^f and the GPS position g_i . The generation of nearby road segments is performed in our nearby road segments generation module.

Algorithm 1: INC-RB

Input: next GPS position g_i , inferred partial route L_{i-1} , rollback depth dep ;
Output: Updated inferred partial route L_i ;
 Retrieve nearby road segments S_i of g_i ;
 Update the cost values of the transition points in road network;
if $i=1$ **then**
 Retrieve nearby road segments of g_0 ;
 Infer all the candidate local routes between nearby road segments of g_0 and those of g_1 ,
 then sort and save them to the ranking table RT_1 ;
 l_1^* = the route with the highest rank in RT_1 ;
 $L_i \leftarrow l_1^*$;
else
 Infer all the candidate local routes between the last segment in l_{i-1}^* and all nearby road
 segments of g_i , then sort and save them to the ranking table RT_i ;
 l_i^* = the route with the highest rank in RT_i ;
 $L_i \leftarrow L_{i-1} \cup l_i^*$;
 if *rollback conditions are met* **then**
 $L_i \leftarrow \text{Rollback}(L_i, dep, < RT_{i-dep}, \dots, RT_{i-1} >)$;

Next, given the new GPS position g_i , the cost values of the transition points in the road network are then updated (line 1) for the subsequent local route search operations. The determination of cost value considers both initial transition costs set in the offline preprocessing stage as well as the real-time cost that relates to the distances of road segments to the new GPS positions. In our current work, we use the same way to calculate the cost values as described in [94].

In the case that g_i is the second GPS position (line 3) in the whole GPS trajectory (i.e., there is no previous partially inferred route yet), the local route search is performed between each pair of nearby road segments $\{s_0^f, s_1^t\}$ of the first GPS position (i.e., g_0) and the second GPS position (i.e., g_1) respectively. The cost of a candidate local route $l_1^{(f,t)}$ is denoted as $c(l_1^{(f,t)})$, which is estimated based on the method described in [94]. The transition probability that the moving path of a vehicle from g_0 to g_1 follows the candidate local route $l_1^{(f,t)}$ is then calculated. Similar to [94], we consider both the cost of candidate local route and the GMC distance between g_0 and g_1 for determining the transition probability $p(l_1^{(f,t)})$ as follows:

$$p(l_1^{(f,t)}) = \exp(-\alpha \times \frac{c(l_1^{(f,t)})}{\text{dist}(g_0, g_1)}) \quad (3.2)$$

where $\text{dist}(g_0, g_1)$ is the MGC distance between g_0 and g_1 , α is the balance factor currently set to 0.1.

The selection probability of the candidate local route $l_1^{(f,t)}$ is determined through combining Eq. 3.1 and Eq. 3.2 as follows:

$$sc(l_1^{(f,t)}) = p(s_0^f|g_0) \times p(l_1^{(f,t)}) \times p(s_1^t|g_1) \quad (3.3)$$

These candidate local routes between g_0 and g_1 are then sorted based on their selection probability and are stored in the ranking table RT_1 . The route with the highest rank is chosen as the inferred local route l_1^* and added to L_1 (line 1 and 1).

In the case that there has already been a partially inferred route (i.e., $i > 1$), L_i will be generated by extending from the last road segment s_{i-1}^{lst} in l_{i-1}^* . Here, unlike the HMM-based conservative approach which requires pair-wise local route search, our local route search in this case is only performed between s_{i-1}^{lst} and the nearby road segments of g_i .

Similar to Eq. 3.2, we calculate the transition probability that the moving path of a vehicle from g_{i-1} to g_i follows the candidate local route $l_i^{(lst,t)}$. However, since the likelihood that the vehicle follows a candidate local route also depends on the previous decisions (i.e., the partially inferred route), we extend Eq. 3.2 to derive the transition probability for $l_i^{(lst,t)}$ as follows:

$$p(l_i^{(lst,t)}) = \exp(-\alpha \times \frac{c(l_i^{(lst,t)}) + c(L_{i-1})}{\sum_{j=1}^i \text{dist}(g_{j-1}, g_j)}) \quad (3.4)$$

where $c(l_i^{(lst,t)})$ denotes the cost of the candidate local route $l_i^{(lst,t)}$ connecting s_{i-1}^{lst} and s_i^t , $c(L_{i-1})$ is the cost of L_{i-1} , which is determined as $c(L_{i-1}) = \sum_{j=1}^{i-1} c(l_j^*)$ (l_j^* represents the inferred local route between g_{j-1} and g_j) and $\text{dist}(g_{j-1}, g_j)$ is the MGC distance between g_{j-1} and g_j .

The selection probability of $l_i^{(lst,t)}$ is determined through combining Eq. 3.1 and Eq. 3.4 as follows:

$$p^{\text{sel}}(l_i^{(lst,t)}) = p(l_i^{(lst,t)}) \times p(s_i^t|g_i) \quad (3.5)$$

These candidate local routes are sorted based on the selection probability and are stored in the ranking table RT_i (line 1). Then, the route with the highest rank is selected to extend L_{i-1} (line 1). If rollback conditions are met, a rollback

correction will be performed (line 1 to line 1), which will be further explained in the next sub-section.

3.2.3 Rollback mechanism

The design of our rollback mechanism contains 1) *detection of rollback conditions* that examine the abnormal situations to trigger the rollback operation, and 2) *rollback correction* that corrects the partially inferred route by rolling back to the previous inference states and performing the re-inference.

Rollback condition detection: This component decides when to perform rollback correction. That is, whenever one of the rollback conditions is met, rollback correction will be performed. Specifically, we consider three types of rollback conditions: no route condition, high cost condition and abnormal driving condition. The no route condition occurs when there is no reachable local route between the previous and current GPS positions. The high cost condition is triggered when all the candidate local routes between the previous and current GPS positions are of high cost values. The abnormal driving condition defines the situation where the inferred local route exhibits some abnormal driving patterns of a vehicle, such as unnecessary U-turn or detour. The detailed definitions of these rollback conditions are described below. Note that it is possible to include more rollback conditions as necessary to further improve the quality of inference in future.

Rollback condition 1 (No route) : There is no available candidate local route between g_{i-1} and g_i . This indicates that the previous inference has led to a dead end situation. Thus, it definitely requires a rollback.

Rollback condition 2 (High cost) : All the candidate local routes between g_{i-1} and g_i have large MGC distances to the line connecting g_{i-1} and g_i , exceeding the distance threshold ξ , and in the mean time, the costs of all these local routes exceed certain cost threshold ξ (Note that for each candidate local route we measure the maximum MGC distance between the route and the line of the consecutive corresponding GPS positions). This indicates that the local routes have deviated from the line connecting g_{i-1} and g_i which provides a rough estimation of actual route direction. At the same time, all these local routes have low probabilities (i.e., the high cost) to match the real route. In such condition, we also perform

a rollback. Note that the values of threshold η and ξ are dynamically determined during the online inference process.

Rollback condition 3 (Abnormal driving) : The newly inferred local route between g_{i_1} and g_i contains too many large angle turns which can be considered to be abnormal from driving behavior's perspective. This condition is evaluated based on the calculation of turning rate as follows:

$$tr_i = \frac{\sum \chi(\theta_k/0.5\pi)}{snum(l_i^*) - 1} \quad (3.6)$$

where θ_i^k is the angle of the turn k on the inferred local route l_i^* , $\chi(x)$ is a defined step function which returns 1 if $x \geq 1$ and 0 otherwise, the function $snum(l_i^*)$ returns the number of segments in the inferred local route l_i^* . In the case that $snum(l_i^*) = 1$, tr_i is set to 0. The rollback condition 3 is triggered when tr_i is greater than a threshold τ which is dynamically determined during the online inference process.

Rollback correction: The aim of rollback correction is to correct the partially inferred L_i by rolling back to the previous inference and search for the alternative feasible route. We adopt a best-effort approach for rollback operation. That is, we first rollback to the previous level (i.e., $i - 1$). If a new feasible route extending to g_i can be found, we replace the old route and stop rolling back further. Otherwise, we iteratively rollback to further levels. The process of rollback correction is shown in Algorithm 2.

In Algorithm 2, the rollback level k ($k \in [1, dep]$) indicates the current level where rollback correction is performed. At each rollback level, we first initialize L'_{i-k} with L_{i-k} (that is, $\langle l_1^*, \dots, l_{i-k}^* \rangle$, a subset of the existing inferred local route in L_i). Then, we retrieve the alternative local routes stored in the ranking table RT_{i-k} which were previously generated as shown in Algorithm 1. For each alternative local route l'_{i-k} , we replace the old l_{i-k}^* with it and re-generate L'_i according to the same procedure as described in Algorithm 1.

To compare the original solution L_i and the new solution L'_i , the joint probability of the inferred local routes is calculated as follows:

$$sc(L_i) = \prod_{j=i-k}^{j=i} sc(l_j^*) \quad (3.7)$$

Algorithm 2: Rollback

Input: Partially inferred route L_i , Rollback Depth dep , Ranking tables

$\langle RT_{i-dep}, \dots, RT_{i-1} \rangle$

Output: Updated inferred partial route L_i

```

for  $k = 1$  to  $dep$  do
     $L'_{i-k} \leftarrow \langle l_1^*, \dots, l_{i-k}^* \rangle$ ;
    for each alternative local route  $l'_{i-k}$  in  $RT_{i-k}$  do
        Replace  $l_{i-k}^*$  in  $L'_{i-k}$  with the alternative local route  $l'_{i-k}$ ;
        Re-generate a new  $L'_i$  that starts from  $l'_{i-k}$ ;
        Re-generate new ranking tables  $\langle RT'_{i-k+1}, \dots, RT'_i \rangle$ ;
        if  $sc(L'_i) > sc(L_i)$  according to Eq. 3.7 then
             $L_i \leftarrow L'_i$ ; //  $L_i$  is updated by  $L'_i$ 
             $\langle RT_{i-k+1}, \dots, RT_i \rangle \leftarrow \langle RT'_{i-k+1}, \dots, RT'_i \rangle$ ;
    if  $L_i$  has been updated then
        return  $L_i$ ;
return  $L_i$ ;

```

where $sc(l_j^*)$ is the selection probability of the inferred local route l_j^* ($j \in [i-k, i]$). If $p(L'_i)$ is higher than $p(L_i)$, the partially inferred route L_i will be corrected by L'_i . Also, the original ranking tables $\langle RT_{i-k+1}, \dots, RT_i \rangle$ are updated accordingly. After examining all the alternative local routes, the rollback process will be terminated if L_i has been corrected. Otherwise, the rollback process will proceed to the next level. If L_i is not updated through the rollback correction process, the algorithm will return the existing L_i .

3.3 The experiment setup

This section investigates the performance of the proposed INC-RB in terms of the inference accuracy and inference delay when dealing with large-scale dataset and the performance analysis of our rollback mechanism. We first describe the dataset being used and the evaluation method. We then show the experiment results of our INC-RB with the comparison of three state-of-the-art online map matching algorithms. Last, we conduct the detailed analysis of our rollback mechanism.

3.3.1 Dataset description

A large-scale real dataset consisting of GPS trajectories over 100 cities around the world [42] is used for evaluation. The dataset contains up to 247,251 geopoints with a sampling rate of 1Hz and the length of the trajectories varies from 5km to 100 km. In addition, different trajectories have different features, including loops, gaps, u-turns, and so on. The number of such trajectories are shown in Table 3.1. The details of these trajectory features in the dataset can be found in [42].

TABLE 3.1: Number of trajectories with different features

U-turns	Hives	Loops	Gaps	Congruence issues	No tags
25	3	24	73	20	19

The GPS trajectories under different sampling rates are generated by subsampling from the original dataset. The sampling rates being evaluated are set to 60s, 120s and 180s respectively. We evaluate and compare the proposed INC-RB with three state-of-the-art online map matching algorithms as described below. The selection of compared algorithms is based on two considerations. Firstly, the algorithms show the recent advancements that improve from the classical HMM-based model [25] in terms of inference accuracy and inference delay. Secondly, similar to our INC-RB, the algorithms leverage only a minimum set of information (i.e., GPS trajectories and road network topology data) for route inference. Therefore, the algorithms (e.g., [38, 79]) that require additional information such as vehicle’s velocity, free-flow travel time, road class, etc., are not selected for comparison.

- OLMM [49]: The algorithm is a HMM-based online map matching algorithm that uses an online learning method to dynamically tune the key parameter in map matching model. The online matching method utilizes a small piece of trajectory data and their matching result to tune the parameter value in the subsequent inference. The matching process in OLMM adopts the online Viterbi algorithm proposed in [25] to compute the partial inferred route based on the detection of convergence point.
- Heuristic Optimization (HEU) [37]: The algorithm is a HMM-based online map matching algorithm that proposes several heuristic optimization for

pruning low probability states, reducing the shortest path search space and detection of convergence conditions respectively. By adopting these optimization techniques, both computation speed and inference delay are improved from the original HMM-based model [25]. In HEU, the partial inferred route can be generated before the convergence point is found.

- Feature-based method (FEA) [95]: In this algorithm, a set of features that considers both GPS observations and human factors is proposed for estimating the cost of candidate routes during the matching process. The algorithm offers both offline and online versions.

To have a fair comparison, the standard deviation of GPS measurements σ is initially set to 10 and the maximum number of candidate segments for each GPS point is set to 10 for all the compared algorithms and our INC-RB. For OLMM, the parameter σ is dynamically adjusted from its initial value during the inference according to its online learning method [49]. For HEU, we use the same settings as described in [37]. Specifically, the error range of the location measurements is set to be 4 times the standard deviation σ . The parameters for pruning unlikely candidate segments and candidate local routes are set to 100 and 1.1 respectively. The threshold value, which is used for the early output of the partially inferred route to decrease the decoding delay is set to 0.9. For FEA algorithm, the balance factor ω which controls the transition angle is set to 100 the same as in [95]. For our INC-RB, the balance factor α , which is used to determine the transition score is set to 0.1 based on the analysis of the ground truth data. The rollback depth dep is set 2 according to our analysis. For the parameters η , ξ and τ in our rollback conditions, the values of these parameters are dynamically determined during the online inference process. Specifically, given the partial inferred route $L_{i-1} : <l_1^*, l_2^*, \dots, l_{i-1}^*>$, the values of η , ξ and τ are dynamically updated based on the average of the corresponding values of these inferred local routes (i.e., partially inferred route from l_1^* to l_{i-1}^*). For example, to determine η , as maximum MGC distance of each inferred local routes (l_1^* to l_{i-1}^*) to their corresponding lines connecting GPS points have been obtained, then, η is determined as the average value of these distance values. Similarly, ξ is determined as the average value of the costs of these inferred local routes from l_1^* to l_{i-1}^* . For τ , it is also determined as the the average value of the turning rates from l_1^* to l_{i-1}^* .

3.3.2 Evaluation method and metrics

In our experiments, the proposed INC-RB and the above algorithms are evaluated and compared from the three aspects: inference accuracy, inference delay and execution efficiency. In addition, the effectiveness of rollback mechanism in INC-RB is also evaluated. The evaluation metrics used in these experiments are described as follows.

Inference accuracy: To evaluate the accuracy of the proposed map matching algorithm, we use the metric named route mismatch fraction (RMF) proposed in [58] as

$$RMF = \frac{L_+ + L_-}{L_{Real}} \quad (3.8)$$

where L_+ is the total length of the mismatched road segments from the inferred route (i.e., the false positive road segments). L_- is the total length of the mismatched road segments from the real route (i.e., the false negative road segments). A smaller RMF value indicates a higher inference accuracy.

The map matching precision is measured as the ratio between the total length of the matched road segments and the total length of the inferred route. The map matching recall is measured as the ratio between the total length of the matched road segments and the total length of the real route. These two metrics are therefore formulated as follows:

$$precision = \frac{L_{matched}}{L_{Inferred}} \quad (3.9)$$

$$recall = \frac{L_{matched}}{L_{Real}} \quad (3.10)$$

where $L_{matched}$ is the total length of the matched road segments between the real route and the inferred route. L_{Real} is the total length of the real route. $L_{Inferred}$ is the total length of the inferred route.

The F-score, which is the harmonic mean of precision and recall, can be defined as

$$F\text{-score} = 2 \times \frac{precision \times recall}{precision + recall} \quad (3.11)$$

Inference delay: For our INC-RB algorithm, we do not adopt any of the above sliding window strategies and we use an opportunistic approach to infer the route up to the current GPS position. Therefore, we do not have decoding delay in

the most cases except the time when rollback mechanism is triggered. Therefore, we consider the algorithm execution time as well as the time required to perform rollback mechanism to measure the inference delay for our algorithm.

Execution efficiency: To evaluate the execution efficiency, we measure the algorithm execution time of a given online matching algorithm at each GPS position and the average value over all the GPS positions is recorded as the average execution time. The average execution time of INC-RB, OLMM, HEU and FEA are measured with a Dell Precision T3610 workstation with Intel(R) Xeon(R) CPU E5-1650v2@3.50GHz and 16GB RAM operated with a 64-bit Windows 10 system.

Performance of rollback mechanism: The design of rollback mechanism plays a key role in our algorithm to ensure the accuracy of the route inference while achieving minimal inference delay. Therefore, it is important to understand to what extent such rollback mechanism is triggered during the entire inference process and how the rollback correction performs when it is triggered. To this end, we propose two evaluation metrics, rollback frequency and rollback improvement rate. The rollback frequency (RF) counts the number of times that rollback correction is triggered during the inference process of an entire GPS trajectory where GPS positions are incrementally added (i.e., in an online manner). Thus, the rollback frequency can be calculated as follows:

$$RF = \frac{1}{N^{\mathbb{T}}} \sum_{tra=1}^{N^{\mathbb{T}}} \frac{N_{tra}^{rb}}{n_{tra} - 1} \quad (3.12)$$

where $N^{\mathbb{T}}$ denotes the the number of all trajectories being tested, N_{tra}^{rb} denotes the total number of rollback correction being triggered for a given trajectory tra , n_{tra} is the number of the GPS positions in the trajectory tra . A small RF indicates a small percentage of rollback corrections are triggered, which means less overhead and faster inference.

The rollback improvement rate (RI) measures how the accuracy of inference results can be improved by the rollback mechanisms when comparing with the alpha version of our INC-RB with rollback mechanisms being disabled. Here, we use RMF (i.e., smaller is better) to measure the accuracy and the rollback improvement rate RB can then be calculated as follows:

$$RI = \frac{1}{N^{\text{RT}}} \sum_{tra=1}^{N^{\text{RT}}} \frac{RMF_{tra}^+ - RMF_{tra}^-}{RMF_{tra}^-} \quad (3.13)$$

where N^{RT} denotes the number of trajectories among all trajectories being tested where the rollback operation(s) are triggered, RMF_{tra}^+ is the RMF value of a given trajectory tra with rollback mechanism is enabled and RMF_{tra}^- is the RMF value of a given trajectory tra with rollback mechanism is disabled. A positive RI indicates the rollback operation can improve the overall accuracy.

3.4 Evaluation results

Inference accuracy results: The average inference accuracy in terms of RMF, precision, recall and F-score over the total 100 GPS trajectories (i.e., one GPS trajectory per city) under different sampling rate is summarized in Table 3.2. It can be seen that the proposed INC-RB outperforms the compared algorithms in the most cases especially under the low sampling rates. In terms of RMF, INC-RB achieves up to 18.5% and 32.6% compared with OLMM, 15.8% and 24.8% compared with HEU and 14.4% and 13.3% improvement compared with FEA under the sampling rates of 120s and 180s. INC-RB can also obtained improved precision, recall and f-score under all the three sampling rates except for the case of precision under 120s sampling rate.

Among the compared algorithms, FEA can achieves the best inference accuracy. This is mainly because that unlike OLMM and HEU which take shortest path as the candidate local route between two nearby road segments, FEA considers several human behavior features to determine the path that can reflect real driver's route choices. Therefore, it can yield the more natural route when determining the local routes. Our INC-RB also adopts the similar features to derive the transition probability of candidate local routes. However, our INC-RB introduces the rollback mechanism to further ensure the accuracy of inference

Next, we investigate the inference accuracy of INC-RB for trajectories under different features. The average RMF of the trajectories belonging to each feature are shown in Fig. 3.2. It can be seen that our INC-RB can also obtain good performance for the trajectories under different features, even for the challenging

TABLE 3.2: Inference Accuracy

Sampling Rate (sec)	RMF			PRECISION		
	60	120	180	60	120	180
OLMM	0.079	0.124	0.184	0.946	0.937	0.907
HEU	0.066	0.120	0.165	0.968	0.953	0.913
FEA	0.065	0.118	0.143	0.970	0.964	0.910
INC-RB	0.064	0.101	0.124	0.973	0.957	0.943
Sampling Rate (sec)	RECALL			F-SCORE		
	60	120	180	60	120	180
OLMM	0.926	0.899	0.862	0.932	0.924	0.892
HEU	0.918	0.921	0.887	0.941	0.932	0.909
FEA	0.920	0.918	0.916	0.953	0.935	0.914
INC-RB	0.934	0.938	0.917	0.956	0.946	0.922

features such as u-turns and gaps. For the trajectories with hive which indicates a large volume of GPS positions are packed in a small area and the trajectories with congruence issues (CI) which indicates the map and the GPS trajectories are dissimilar, these trajectories may potentially contain many noises. It can be observed that our INC-RB achieves a much better RMF value compared to the other three algorithms for the trajectories with hive and CI. The trajectories with no-tags are those with the high quality GPS data. Therefore, all the algorithms can achieve relatively small RMF values for these trajectories.

Inference delay results: Figure 3.3 shows the inference delay with the increase of window size of sliding window strategies under different sampling rates. It can be seen that the output delay of FEA generally increases proportionally with window size because FSW is used as the sliding window strategy. By using BVSW, the average inference delay of OLMM remains unchanged once the window size reaches 6 for the sampling rates of 60s and 120s and 4 for the sampling rate of 180s. This means BVSW used in OLMM has no significant advantages at the window size of 4 and above. As HEU adopts sliding window strategy that functions similar to VSW, the inference delay remains constant under different window size. HEU specifically introduces the heuristic optimization to reduce the decoding delay. Therefore, the inference delay keeps lower compared with OLMM and FEA. Our proposed INC-RB achieves the best inference delay and it is also not affected by the window size. This is because we use the opportunistic approach in which the delay only occurs when the rollback mechanism is triggered.

FIGURE 3.2: RMF comparison with different trajectory features

Figure 3.4 shows the corresponding average RMF of each algorithm with the increase of window size. For OLMM and FEA, smaller RMF can be obtained when the window size increases from 2 to 6. When the window size is above 6, RMF value of OLMM remains unchanged which lies in the reason that the inferred partial route has been updated before the window bound is reached. For FEA, RMF value decreases slowly when the window size is larger than 8. For HEU and our INC-RB, the RMF value is not affected by the window size. Regardless of window size, our INC-RB can achieve the lowest RMF value for all the sampling rates. Compared to HEU which has the shortest inference delay among the compared algorithms, our INC-RB reduces the inference delay by 92.67%, 96.12% and 97.51% for the sampling rates of 60s, 120s, and 180s respectively.

FIGURE 3.3: Average inference delay under different window size

Execution time results: The average algorithm execution time is measured for evaluating the execution efficiency of each testing algorithm. Specifically, we select all 3 GPS trajectories of hive feature and randomly select 5 GPS trajectories of each of other features. The execution time results over the 28 GPS trajectories are shown in Table 3.3. We also conducted the independent two sample t-test which shows that the average execution of INC-RB is significantly shorter than other three algorithms.

TABLE 3.3: Average Execution Time of 25 Cities

Sampling Rate (sec)	60	120	180
OLMM	12.06 ^T	12.54 ^T	13.15 ^T
HEU	8.46 ^T	9.26 ^T	9.74 ^T
FEA	11.75 ^T	12.12 ^T	12.66 ^T
INC-RB	5.19	5.43	5.62

T: significantly better(t-test)

Among the other three algorithms, the average execution time of HEU is the shortest because the algorithms is optimized to reduce the total number of route search by pruning unlikely nearby road segments. Compared with HEU, the proposed

FIGURE 3.4: Average RMF under different window size

INC-RB still can achieve 38.7%, 41.3% and 42.2% improvements in terms of the average execution time under the three sampling rates respectively. The performances of OLM and FEA in terms of average execution time are similar as the size of search space for determining the local routes are similar in these two algorithms. Our INC-RB outperforms the other compared algorithms mainly because that its opportunistic inference strategy requires a smaller search space and is capable of producing a fast inference up to the current GPS point.

It should be also noted that the execution time of our INC-RB reported here also improves from our previous work [32]. This is mainly because that we have optimized the threshold values, such as η and ξ , in the rollback conditions so as to reduce the number of times that rollback needs to be triggered.

Analysis of rollback mechanism: In this section, we first present the analysis on how the setting of rollback depth dep parameter affects the inference accuracy and execution time. Then, we present the evaluations based on two metrics: rollback frequency (RF) and rollback improvement rate (RI) to examine the performance of rollback mechanism. The rollback depth dep is a key parameter in our INC-RB algorithm, which controls the maximum number of rollback operations to be performed. While a large dep can potentially help to improve the inference accuracy

FIGURE 3.5: Impact of rollback depth dep under different sampling rates

by allowing to roll back with more levels, it requires additional computational cost. To investigate the impact of dep values, we evaluate the inference accuracy in terms of RMF and average execution time with different dep values under sampling rate of 60s, 120s and 180s respectively. Fig. 3.5 shows the evaluation results with dep 's value set from 1 to 3.

It can be seen that the average execution time increases linearly with the increase of dep value. For the sampling rate of 60s, there is a slight improvement on the inference accuracy (i.e., the decrease of RMF value) with the increase of dep value. For the sampling rate of 120s and 180s, the inference accuracy decreases when $dep = 3$. This may be because that the distance between GPS positions increases as the sampling interval is larger. In such situation, the rollback operation requires an evaluation of more alternative routes, which may bring additional noises. Therefore, we find that the setting of $dep = 2$ can achieve a good balance between the inference accuracy and efficiency for our current dataset.

The average rollback frequency (RF) and rollback improvement rate (RI) of the trajectories being tested with different features under different sampling rates are shown in Table 3.4. It can be seen that RF is kept in a low level (less than 15%) under all the sampling rates, which means the opportunistic inference is successful

TABLE 3.4: Ratios of 25 cities

	LOOPS		GAPS		U-TURN	
Sampling Rates (sec)	RF	RI	RF	RI	RF	RI
60	6.2%	8.5%	6.0 %	9.5%	8.2%	5.1%
120	8.4%	11.7%	9.6 %	8.2%	10.3%	8.2%
180	8.5%	12.5%	10.6%	17.9%	12.1%	7.6%
	HIVE		CI		NO-TAGS	
Sampling Rates (sec)	RF	RI	RF	RI	RF	RI
60	5.0%	13.3%	11.0%	11.3 %	5.3%	7.2%
120	6.8%	10.2%	13.2%	10.4%	6.6%	5.9%
180	9.4%	18.5%	12.7%	7.2%	7.5%	6.4%

in the most of times and rollback corrections are only invoked occasionally. It can be also observed that RF generally increases from sampling rate of 60s to 180s. This is as expected because the low sampling rate generally introduces more uncertainties and it is thus more likely to have some false inferences that need to be corrected. RI measures how the inference accuracy in terms of RMF can be improved by the rollback mechanisms compared with rollback mechanism being disabled. It can be seen that RI values generally increase when the sampling rate becomes low (e.g., 180s). This means the rollback mechanism plays a more role when dealing with low-sampling data. RI values also vary for the trajectories of different features. For the trajectories with features of loops, gaps and hive that may contain more noisy data, RI values are larger compared with the ones for the trajectories with other features. This indicates the rollback mechanism is especially helpful to ensure the inference accuracy when noisy data are presented.

3.5 Summary

In this chapter, an incremental route inference algorithm with rollback for online map matching is proposed. To minimize inference delay, the proposed algorithm adopts an opportunistic approach that can infer the route up to the newly observed GPS position. To ensure the inference accuracy, a rollback mechanism is provided to correct the already inferred route when some unusual conditions are detected. We evaluated our proposed algorithm with a real-world dataset and compared it with three state-of-the-art online map matching algorithms OLMM, HEU and FEA.

The experimental results show that our algorithm can achieve better or at least comparable inference accuracy to the compared algorithms. More importantly, our algorithm has much shorter inference delay, which is critical for some latency-sensitive applications, such as real-time traffic sensing, navigation assistance and route recommendation.

In terms of local routes search, next chapter introduces the driving cost models and gives a distribution-based weights estimation approach for the determination of local routes.

Chapter 4

Distribution-based Weights Estimation for Map Matching Algorithms

To recover the travelling routes of vehicles, the local routes between consecutive GPS measurements are firstly inferred with the consideration of driving cost models. This chapter describes the proposed weights estimation approach (*DBWES*) for driving cost models. The approach framework, the detailed approach design, the experiments and results are provided in the following sections.

4.1 Framework

The framework of our proposed DBWES is shown in Fig. 4.1. It consists of two major components: *Popularity-based training set generation* and *Weights estimation*. The component of popularity-based training set generation generates the popular routes for extracting routes popularity. Firstly, the *trajectories preprocessing* module processes the GPS logs into sub-trajectories which contain high-sampling-rate GPS measurements by eliminating stationary GPS measurements. The *routes generation* module generates the routes from the high-sampling-rate sub-GPS trajectories by applying an existing map matching algorithm. Next, the *popular routes determination* module determines the popular routes from the travelling routes.

After that, the *travelling features extraction* module extracts the travelling features from the popular routes and GPS measurements. The component of weights estimation is responsible for determining the weights by inferring their distributions. Since the driving cost models with the nonlinear combination of travelling features increase difficulties for weights estimation, the module of *linear module encoding* converts these driving cost models to the linear-combination models with encoded weights and encoded travelling features. Then, the module of *Bayesian inference for distributions estimation* is conducted to determine the distributions of encoded weights. After that, the module of *linear model decoding* is responsible for determining the distribution of each weight, and the *weights determination* module determines the values of weights from their distributions. Finally, the *map matching inference* component is responsible for route inference by applying the estimated weights on the driving cost model of a given map matching algorithm.

FIGURE 4.1: Framework

4.2 Detailed design of DBWES

4.2.1 Distribution-based Training Set Generation

This component is responsible for generating popular routes from massive GPS logs, and the popular routes are leveraged as the training set for weights estimation.

Trajectories preprocessing: In practice, GPS logs record the movements of drivers for a long period of time. Therefore, GPS logs may contain stationary GPS measurements (drivers may stay at a geographical location for a long time) [100]. Therefore, if one stationary GPS measurement is identified, it implies a new trace for the driver has started. Therefore, this module splits each GPS trajectory into multiple sub-trajectories by identifying stationary GPS measurements.

Routes generation: Considering the GPS measurement errors, the GPS trajectory of a vehicle deviates from the real route. This module generates travelling routes from the sub-trajectories by applying an existing map matching algorithm [58]. The set of road segments mapped from the GPS positions is represented as $\hat{S}$, and the set of generated travelling routes from the sub-trajectories is represented as $\hat{R}$.

Popular routes determination: In this module, popular routes are extracted from the travelling routes $\hat{R}$ for weights estimation. To extract popular routes, the routes popularity, which is defined as the number of traces on a given travelling route, is calculated for each travelling route. Then a popular route is defined as the travelling route with high popularity (larger than the popularity threshold κ). The set of popular routes is represented as $\mathbb{PR}$.

Features extraction In this module, travelling features are extracted from the popular routes. Based on the existing driving cost models discussed in Chapter 2, the route-related and trajectory-related features are extracted.

Given the popular routes, route-related features can be obtained directly. However, since each popular route may be generated from multiple different GPS trajectories, this increases difficulties to obtain trajectory-related features. In our proposed system, we firstly collect the GPS trajectories for each popular route, and store the GPS points which are mapped to the start and end road segments. For example, given the popular route $l_m^{\hat{s}, \hat{t}}$ in $\mathbb{PR}$, the GPS points which are mapped to the start/end road segments $\hat{s}_s$ and $\hat{s}_t$ of $l_m^{\hat{s}, \hat{t}}$ are stored in the sets $\mathbb{GS}_s$ and $\mathbb{GS}_t$ respectively. We then generate the approximate GPS points g_s^* and g_t^* as follows:

$$g_s^*.lat = \frac{\sum_{p=1}^{|\mathbb{GS}_s|} \mathbb{GS}_s(p).lat}{|\mathbb{GS}_s|} \quad (4.1)$$

$$g_s^*.lon = \frac{\sum_{p=1}^{|\mathbb{GS}_s|} \mathbb{GS}_s(p).lon}{|\mathbb{GS}_s|} \quad (4.2)$$

where $|\mathbb{GS}_s|$ refers to the number of GPS points in the set $\mathbb{GS}_s$, $g_s^*.lat$, and $[g_s^*.lon, g_s^*.lat]$ are the geo-coordinates of the approximate GPS point for the road segment $\hat{s}_s$ by averaging the GPS positions in $\mathbb{GS}_s$. In this manner, the trajectory-related features in the existing driving cost models can be obtained approximately.

4.2.2 Weights determination

This component is responsible for determining weights by inferring their distributions. For a given driving cost model with the nonlinear combination of travelling features, the driving cost model is firstly converted to a linear combination model with encoded weights and encoded travelling features. Secondly, Bayesian inference is conducted to estimate the distributions of the encoded weights. After that, the distribution of each weight is inferred from the module of linear model decoding. Finally, the value of each weight is obtained through a sampling-based strategy.

Linear model encoding (LME): When the driving cost model is formulated with the nonlinear combination of travelling features, dependent weights are introduced. To decrease the difficulties of distributions inference, such driving cost models are converted to the ones with the linear combination of travelling features.

- **Linear combination:** For a driving cost model with the linear combination of travelling features, the driving cost model can be formalized as follows:

$$F_{cost}(\theta, F, l, G_l) = \theta F + \varepsilon_l = \sum_{j=1}^h \theta_j f_j + \varepsilon_l. \quad (4.3)$$

where ε_l is the noise, following a zero-mean normal distribution with standard deviation σ_{ε_l} . As there is no dependence among the weights θ , they are considered to be "separable", and their distribution can therefore be inferred directly from Bayesian inference.

- **Nonlinear Combination:** For a driving cost model with the nonlinear combination of travelling features, linear model encoding (LME) converts

the driving cost model to a model with the linear combination of travelling features.

$$F_{cost}(\theta, F, l, G_l) \xrightarrow{\text{LME}} \theta^* F^* + \varepsilon_l \quad (4.4)$$

where θ^* and F^* are the encoded weights and encoded travelling features respectively. Take the driving cost model $F_{cost}(\theta, F, l, G_l) = \theta_1 f_1 \times (\theta_2 f_2 + f_3) + \varepsilon_l$ as the example, the driving cost model after *LME* can be formulated as: $F_{cost}(\theta, F, l, G_l) = \theta_{1,2}^* f_{1,2}^* + \theta_1^* f_{1,3}^* + \varepsilon_l$, where $f_{1,2}^*$ and $f_{1,3}^*$ are the encoded features, $\theta_{1,2}^*$ and θ_1^* are the corresponding encoded weights. In this example, $f_{1,2}^* = f_1 f_2$, $f_{1,3}^* = f_1 f_3$, $\theta_{1,2}^* = \theta_1 \theta_2$, and $\theta_1^* = \theta_1$.

Bayesian inference for distributions estimation: The distributions of the encoded weights θ^* are estimated through Bayesian inference. In this module, popularity-transition equations are firstly built by utilizing popular routes. Then, Bayesian inference is conducted, where the posterior distribution of each encoded weight is determined from a Markov Chain Monte Carlo (MCMC) strategy.

Given a subset of the popular routes $\mathbb{PR}_{s,t}$ which stores the popular routes with the start/end segment pair $\langle \hat{s}_s, \hat{s}_t \rangle$, the popularity-transition equations are built by leveraging each pair of popular routes, i.e., $\langle \hat{l}_m^{s,t}, \hat{l}_n^{s,t} \rangle$. The popularity ratio between the route $\hat{l}_m^{s,t}$ and $\hat{l}_n^{s,t}$ is firstly determined as follows:

$$Pr(\hat{l}_m^{s,t}, \hat{l}_n^{s,t}) = \frac{P(\hat{l}_m^{s,t})}{P(\hat{l}_n^{s,t})} \times \frac{(P(\hat{l}_m^{s,t}) + P(\hat{l}_n^{s,t}))}{\sum_{k=1}^{|\mathbb{PR}_{s,t}|} P(\hat{l}_k^{s,t})} \quad (4.5)$$

where $P(\hat{l}_m^{s,t})$ and $P(\hat{l}_n^{s,t})$ refer to the popularity of $\hat{l}_m^{s,t}$ and $\hat{l}_n^{s,t}$ respectively. $\sum_{k=1}^{|\mathbb{PR}_{s,t}|} P(\hat{l}_k^{s,t})$ refers to the total popularity of the popular routes in $\mathbb{PR}_{s,t}$, therefore, $\frac{(P(\hat{l}_m^{s,t}) + P(\hat{l}_n^{s,t}))}{\sum_{k=1}^{|\mathbb{PR}_{s,t}|} P(\hat{l}_k^{s,t})}$ indicates the total popularity of the popular routes are considered to determine the popular ratios.

According to [60], given a route with cost C , the probability that the route matches the real travelling route is proportional to $\exp(C)$. This assumption has been adopted in map matching algorithms [32, 58, 95]. Therefore, the transition probability of a route with cost C can be determined as:

$$f(C) = \exp(-C) \quad (4.6)$$

The ratio of the transition probability of the route $l_m^{\hat{s},t}$ and $l_n^{\hat{s},t}$ is determined as follows:

$$Tr(l_m^{\hat{s},t}, l_n^{\hat{s},t}) = \frac{f(F_{cost}(\theta, F, l_m^{\hat{s},t}, G_{l_m^{\hat{s},t}}))}{f(F_{cost}(\theta, F, l_n^{\hat{s},t}, G_{l_n^{\hat{s},t}}))} \quad (4.7)$$

where $f(F_{cost}(\theta, F, l_m^{\hat{s},t}, G_{l_m^{\hat{s},t}}))$ and $f(F_{cost}(\theta, F, l_n^{\hat{s},t}, G_{l_n^{\hat{s},t}}))$ are the transition probabilities of $l_m^{\hat{s},t}$ and $l_n^{\hat{s},t}$ respectively. Then, by substituting Eq. 4.6 into Eq. 4.7, we get:

$$Tr(l_m^{\hat{s},t}, l_n^{\hat{s},t}) = \frac{\exp(-1 \times F_{cost}(\theta, F, l_m^{\hat{s},t}, G_{l_m^{\hat{s},t}}))}{\exp(-1 \times F_{cost}(\theta, F, l_n^{\hat{s},t}, G_{l_n^{\hat{s},t}}))} \quad (4.8)$$

Since a higher popularity of a route indicates a higher probability for drivers to choose. Therefore, the popularity-transition equations is built from the popularity ratio and the ratio of the transition probability as follows:

$$Tr(l_m^{\hat{s},t}, l_n^{\hat{s},t}) = \lambda \times Pr(l_m^{\hat{s},t}, l_n^{\hat{s},t}) \quad (4.9)$$

where λ is the scaling factor between the popularity ratio and the ratio of transition probability. Eq. 4.9 can be rewritten as follows:

$$\begin{aligned} \ln(\lambda \times Pr(l_m^{\hat{s},t}, l_n^{\hat{s},t})) = \\ F_{cost}(\theta, F, l_n^{\hat{s},t}, G_{l_n^{\hat{s},t}}) - F_{cost}(\theta, F, l_m^{\hat{s},t}, G_{l_m^{\hat{s},t}}) \end{aligned} \quad (4.10)$$

After the module of linear model encoding, Eq. 4.10 can then be rewritten as:

$$\begin{aligned} \ln(\lambda \times Pr(l_m^{\hat{s},t}, l_n^{\hat{s},t})) = \theta^*(F_n^* - F_m^*) + (\varepsilon_n - \varepsilon_m) \\ \text{s.t. } \theta_j^* \geq 0, \forall \theta_j^* \end{aligned} \quad (4.11)$$

where θ^* and F^* are the encoded weights and encoded features respectively.

Given Eq. 4.11, Bayesian inference is conducted to determine the posterior distributions of the encoded weights. For simplicity, we rewrite the left side of Eq. 4.11 as $y_{m,n}$, and $F_n^* - F_m^*$ is rewritten as $F_{m,n}$, and $\varepsilon_n - \varepsilon_m$ is rewritten as $\varepsilon_{m,n}$. Therefore, Eq. 4.11 can be rewritten as:

$$y_{m,n} = \theta^* F_{m,n} + \varepsilon_{m,n} \quad (4.12)$$

where $\varepsilon_{m,n}$ follows a zero-mean normal distribution population with standard deviation $\sigma_{\varepsilon_{m,n}}$.

Based on the Bayesian formula, the posterior distributions of the encoded weights θ^* are proportional to the prior density multiplied by the likelihood and can be written as follows:

$$p(\theta^*|F_{m,n}, y_{m,n}) \propto p(y_{m,n}, F_{m,n}|\theta^*) \times p(\theta^*) \quad (4.13)$$

where the likelihood is modelled as $p(y_{m,n}, F_{m,n}|\theta^*)$, and $p(\theta^*)$ represents the prior density.

To derive the likelihood function, we use A^P to represent the set that stores the popular route pairs. The likelihood function is determined as follows:

$$\begin{aligned} p(y_{m,n}, F_{m,n}|\theta^*) &= \prod_{k'=1}^{|A^P|} p(\varepsilon_{m,n}^{k'}) \\ &= \prod_{k'=1}^{|A^P|} \left[\frac{1}{\sqrt{2\pi\sigma_{m,n}^2}} \exp\left(-\frac{(y_{m,n}^{k'} - \theta^* F_{m,n}^{k'})^2}{2\sigma_{m,n}^2}\right) \right] \\ &= \frac{1}{(2\pi\sigma_{m,n}^2)^{\frac{|A^P|}{2}}} \exp\left(-\frac{\sum_{k'=1}^{|A^P|} (y_{m,n}^{k'} - \theta^* F_{m,n}^{k'})^2}{2\sigma_{m,n}^2}\right) \end{aligned} \quad (4.14)$$

where $|A^P|$ represents the number of route pairs, k' is index of the route pairs.

For the prior density of θ^* , we assume each encoded weight follows a uniform distribution to make the estimation be robust to different driving cost models. The uniform distribution for weight θ_j is as follows:

$$p(\theta_j^*) = \begin{cases} \frac{1}{UB(\theta_j^*) - LB(\theta_j^*)} & \text{if } LB(\theta_j^*) < \theta_j^* < UB(\theta_j^*) \\ 0 & \text{otherwise} \end{cases} \quad \text{where } UB(\theta_j^*) \text{ and } LB(\theta_j^*) \text{ are}$$

the upper and lower bound of θ_j^* inferred from the dataset or empirically determined, and $p(\theta^*)$ can be determined from the joint probability from each encoded weight.

To estimate the posterior distribution of $p(\theta^*|F_{m,n}, y_{m,n})$, the Metropolis-Hastings strategy [82] which belongs to Markov Chain Monte Carlo (MCMC) is adopted.

For the encoded weights θ^* , the moving probability p^{Acc} is calculated as the ratio of the new position and current position in the posterior distribution. An uniform random number within the interval $[0, 1]$ is then generated. If this random number is contained within the interval $[0, p^{Acc}]$, the movement will be accepted, otherwise it will be rejected. Details of Metropolis–Hastings strategy can be found in [15].

$$\begin{aligned} p^{Acc} &= \frac{p(\theta_{new}^* | F_{m,n}, y_{m,n})}{p(\theta_{current}^* | F_{m,n}, y_{m,n})} \\ &= \frac{p(y_{m,n}, F_{m,n} | \theta_{new}^*) * p(\theta_{new}^*)}{p(y_{m,n}, F_{m,n} | \theta_{current}^*) * p(\theta_{current}^*)} \end{aligned} \quad (4.15)$$

Linear model decoding (LMD): Given the posterior distributions of the encoded weights θ^* , this component determines the distribution of each weight as follows:

$$q(\theta^*) \xrightarrow{\text{LMD}} q(\theta) \quad (4.16)$$

where $q(\theta)$ is the distributions of the weights θ after the linear model decoding.

In this work, the Gibbs sampling strategy [76] is adopted. The LMD strategies are not limited, therefore alternative methods can be considered. Taking the driving cost model $F_{cost}(\theta, F, l, G_l) = \theta_{1,2}^* f_{1,2}^* + \theta_1^* f_{1,3}^* + \varepsilon_l$ as the example, the unknown posterior distribution of θ_2 can be estimated from $\theta_{1,2}^*$ and θ_1^* , through the rate of the fitting distributions of $\theta_{1,2}^*$ and θ_1^* by considering the Geary–Hinkley transformation [29].

Weights determination Given the distribution of each weight, a sampling-based strategy is adopted to generate the sampling set, and the value of each weight is determined by averaging the sampled values as Eq. 4.17.

$$\theta_j = \frac{\sum \theta_j^S}{N^{Sam}} \quad (4.17)$$

where θ_j^S is one sampled value of the weight θ_j , N^{Sam} is the number of the sampled values in the sampled set.

The general procedure of our algorithm is shown in Algorithm 3.

Algorithm 3: Distribution-based Weights Estimation**Input:** GPS logs, Road Network**Output:** Weights values

Trajectories preprocessing to generate sub-GPS trajectories;

Identify road segments $\hat{S}$ and routes $\hat{R}$ from the sub-GPS trajectories;Generate the popular routes $\mathbb{PR}$ and extract the travelling features;Let $A^P = \emptyset$ store the route pairs in $\mathbb{PR}$;**if** *Nonlinear combination driving cost model* **then**

└ Linear model encoding;

for each pair of road segment $\langle \hat{s}_s, \hat{s}_t \rangle$ in $\hat{S}$ **do**└ **for** each pair of route pair $\langle \hat{l}_m, \hat{l}_n \rangle$ **do**└ └ Store the route pair in A^P ;└ └ Calculate the popularity ratio and the rate of the transition probability for
└ └ $\langle \hat{l}_m, \hat{l}_n \rangle$ as Eq. 4.5 and Eq. 4.7;

└ └ Construct the popularity-transition equations as Eq. 4.11;

Bayesian inference to determine the weights posterior distributions;

if *Nonlinear combination driving cost model* **then**

└ Linear model decoding;

Determine the weights values as Eq. 4.17;

4.3 Map matching inference

In this component, routes are inferred by applying state-of-the-art map matching algorithms considering driving cost models. To illustrate the generality of our proposed DBWES, we apply DBWES on existing map matching algorithms, and compare DBWES with the existing empirical and data-driven approaches for weights determination. In these map matching algorithms, driving cost models are adopted with the following considerations:

- Whether is the considered driving cost model linear or nonlinear.
- Whether are the considered travelling features are route-related or trajectory-related.
- Whether are the considered travelling features normalized or non-normalized.

In the experiments, the driving cost models with the above three considerations are adopted. Note that the existing empirical and data-driven approaches cannot be applied on all the driving cost function given the above considerations, whereas our proposed DBWES is able to be applied. The details of the driving cost models are given as follows:

Linear combination: We adopt the driving cost models from the map matching algorithms [34, 50, 60] respectively. In these driving cost models, route-related features (LR) and trajectory-related features (TR) are considered.

For the LR features, the driving cost model is as follows:

$$F_{cost}(\theta, F, l, G_l) = \sum_{j=1}^4 \theta_j^{LR} f_j + \varepsilon_l \quad (4.18)$$

where f_1 to f_4 represent the travelling features of route lengths, turning angles, and the number of left and right turns at road intersections respectively. θ_j^{LR} refers to the weight for estimation. In this model, the travelling features are non-normalized, therefore the weights are unbounded. The details of the travelling features can be found in [34, 60].

For the LT features, we consider the driving cost model with the travelling features based on the map matching algorithm [50]. The driving cost model is as follows:

$$F_{cost}(\theta, F, l, G_l) = \sum_{i=1}^4 \theta_i^{LT} f_i + \varepsilon_l \quad (4.19)$$

where the four travelling features are obtained from the positioning accuracy, spatial accessibility, middle-point spatial distribution and consistency of driving directions respectively. θ_j^{LT} refers to the weight for estimation. In this model, the travelling features are normalized, therefore the weights are bounded between 0 and 1. The details of the travelling features can be found in [50].

Nonlinear combination: We adopt the driving cost model proposed in [95] with the nonlinear combination of both the route-related features (NLR) and trajectory-related features (NLT). The driving cost model is as follows:

$$F_{cost}(\theta, F, l, G_l) = \theta_1^{NLT} f_1 \times (\theta_2^{NLR} f_2 + f_3) + \varepsilon_l \quad (4.20)$$

where f_1 is the distance between the road segment and the GPS trajectory G_l . f_2 and f_3 are the features obtained from the turning angles and lengths of the segments from the route l . The details of the travelling features can be found in [95].

4.4 Experiments and results

In this section, we firstly describe the dataset and the methods for comparison. Then, the evaluation metrics in terms of map matching accuracy is provided. After that, the experimental results and analysis are presented.

4.4.1 Dataset preparation

Network Data: Experiments are conducted on a road network in Beijing extracted from OpenStreetMap [59] as shown in Fig. 4.2. The road network consists of 78334 road segments, the vertical length of the area is about 35 km, and the horizontal length is about 40 km.

FIGURE 4.2: Road network

Trajectory Data: In this work, the GPS trajectories are obtained from the T-drive dataset [97]. The trajectories are generated from 10,357 taxis. After the module of trajectory preprocessing, 1015 sub-GPS trajectories with high sampling rates (less than 15 s) are obtained. To evaluate the map matching algorithms for the low sampling rate GPS trajectories, the sampling rates in our experiments are set to 60 s, 120 s and 180 s respectively, and these trajectories are re-generated by subsampling from the original trajectories. We randomly select 40 GPS trajectories with travelling duration longer than 30 minutes. These trajectories are classified as *effortless* and *complex* trajectories as shown in Table. 4.1 based on the travelling features. The effortless trajectories refer to the routes considering the travelling features, such as small number of turns, etc. The complex trajectories refer to the trajectories with complex travelling features, such as U-turns, loops, etc., which introduce difficulties for route inference [95].

TABLE 4.1: Trajectory categories

Trajectory Type	Number of trajectories	Average Length (km)
Effortless	14	8.34
Complex	26	12.15
Total	40	10.81

4.4.2 Methods for comparison

To evaluate the weights estimation, we compare our proposed DBWES with empirical setting approach and data-driven approach respectively. For data-driven approach, we select the maximum likelihood estimation (MLE) approach [38]. In addition, the linear and non-linear combination of travelling features are considered in the driving cost models.

4.4.3 Parameters settings:

The standard deviation of the GPS measurements is set to 10, and the number of candidate segments prepared for each GPS measurement is set to 10 according to [95]. The number of routes for MLE is set to 5 according to [38]. In our approach, the threshold κ to determine the popular route is set to 5, the scaling factor λ is set to 1.0, and the variance of the white noise in Bayesian inference in Eq. 4.12 is set to 10. Moreover, we run Markov Chain Monte Carlo (MCMC) method with 20,000 iterations to obtain the posterior distributions of the encoded weights. In addition, for the sampling-based strategy, 50 values are sampled from each weight distribution.

4.4.4 Evaluation metrics

Weights distributions estimation: The weights distributions are fitted with the assumed probability density functions (PDF). To evaluate if each posterior distribution is well-fitted by the assumed PDF, the reduced chi-square and fitting status test are conducted.

Map matching accuracy: To evaluate the map matching accuracy, the matrices called the RMF and F-score [95] are adopted to measure the mismatched and matched percentages of road segments given the inferred routes and real routes. The RMF is defined as follows:

$$RMF = \frac{L_+ + L_-}{L_{real}} \quad (4.21)$$

where L_+ is the total length of the mismatched road segments from the inferred route, and L_- is the total length of the mismatched road segments from the real route. L_{real} is the total length of the real route. A smaller RMF value indicates higher map matching accuracy.

To compute F-score, the map matching precision and recall are computed as:

$$precision = \frac{L_{matched}}{L_{Inferred}} \quad (4.22)$$

$$recall = \frac{L_{matched}}{L_{Real}} \quad (4.23)$$

where $L_{matched}$ is the total length of the matched road segments between the real route and the inferred route. L_{Real} is the total length of the real route. $L_{Inferred}$ is the total length of the inferred route. Then F-score, which is the harmonic mean of precision and recall, can then be defined as:

$$F\text{-score} = 2 \times \frac{precision \times recall}{precision + recall} \quad (4.24)$$

where a larger F-score value indicates higher map matching accuracy.

4.4.5 Experimental results

4.4.5.1 Weights distributions estimation

The estimated weights histograms and corresponding PDF for the travelling features are shown in Fig. 4.3, Fig. 4.4 and Fig. 4.5 respectively.

From the fitting results, the weights in our experiments follow normal and exponential distributions with the following PDF respectively:

$$y = \frac{A_x}{\sigma_x \sqrt{2\pi}} \exp\left(\left(\frac{x - \mu_x}{2\sigma_x}\right)^2\right) \quad (4.25)$$

$$y = B_x \exp\left(\frac{-x}{t_x}\right) \quad (4.26)$$

where the estimated results of A_x , μ_x , σ_x , B_x and t_x are shown in Table 4.2 and Table 4.3.

TABLE 4.2: Parameters of PDF for the linear combination of travelling features

PDF Formula	$y = \frac{A_x}{\sigma_x \sqrt{2\pi}} \exp\left(\left(\frac{x - \mu_x}{2\sigma_x}\right)^2\right)$		
Parameter	μ_x	σ_x	A_x
θ_1^{LR}	2.42×10^{-2}	3.49×10^{-2}	1.53×10^{-3}
θ_2^{LR}	3.96×10^{-1}	5.72×10^{-1}	2.50×10^{-2}
θ_3^{LR}	1.85×10^{-1}	2.48×10^{-1}	2.01×10^{-2}
θ_4^{LR}	1.77×10^{-1}	2.32×10^{-1}	2.89×10^{-2}
θ_1^{LT}	2.74×10^{-1}	3.17×10^{-1}	1.49×10^{-2}
θ_2^{LT}	2.14×10^{-1}	2.71×10^{-1}	1.38×10^{-2}
θ_3^{LT}	2.36×10^{-1}	2.90×10^{-1}	1.47×10^{-2}
θ_4^{LT}	1.92×10^{-1}	2.56×10^{-1}	1.32×10^{-2}

TABLE 4.3: Parameters of PDF for the nonlinear combination of travelling features

PDF Formula	$y = \frac{A_x}{\sigma_x \sqrt{2\pi}} \exp\left(\left(\frac{x - \mu_x}{2\sigma_x}\right)^2\right)$		
Parameter	μ_x	σ_x	A_x
θ_1^{NLT}	1.45×10^{-2}	1.11×10^{-2}	1.29×10^{-3}
PDF Formula	$y = B_x \exp\left(\frac{-x}{t_x}\right)$		
Parameter	t_x	B_x	
θ_2^{NLR}	$1.84 \times 10^{+2}$	8.18×10^{-1}

FIGURE 4.3: Distribution fittings for linear route-related features

Table 4.4 gives the results of the weights estimation, and reduced chi-square values to examine the goodness of the distribution fittings. It can be seen that the chi-square values are small, this indicates good distributions fitting. In addition, the fitting status of the given PDF are successful. Moreover, it can be observed that the large gaps between the weight θ_1^{LR} and the three weights θ_2^{LR} to θ_4^{LR} , and between the weights θ_1^{NLT} and θ_2^{NLT} respectively, the reason behind is that the travelling features are non-normalized, and therefore the weights are not bounded.

4.4.5.2 Map matching accuracy

The average map matching accuracy in terms of RMF and F-score over 40 GPS trajectories under different sampling rates are shown in Table 4.5. In general, our proposed DBWES outperforms the empirical setting and MLE approaches for the driving cost models with the linear and non-linear combination of travelling features. Given the driving cost model with the route-related (LR) features, DBWES achieves 4.8%, 5.7%, and 10.4% accuracy improvements in terms of the RMF under the three sampling rates compared with the MLE approach. For the F-score, DBWES achieves 1.9% accuracy improvement when the sampling rate is 180s. For

FIGURE 4.4: Distribution fittings for linear trajectory-related features

FIGURE 4.5: Distribution fittings for nonlinear route/trajectory-related Features

the trajectory-related (LT) features, the MLE approach and our proposed DBWES achieve comparable map matching accuracy under the sampling rate of 60s, whereas DBWES achieves 4.5% and 13.1% accuracy improvements in terms of RMF under the sampling rate of 120 s and 180 s respectively. Given the driving cost model with the non-linear combination of the trajectory-related (NLT) and route-related (NLR) features, our proposed DBWES is compared with both of the empirical setting and MLE approaches. It can be observed that our proposed DBWES achieves up to 21.3% accuracy improvement under RMF for the sampling rate of 180 s compared with the empirical setting approach, and there is 6.2%

TABLE 4.4: Weights estimates from our proposed approach and the distribution fitting results

Features	Parameter	Mean	Std	RCS	FS
LR	θ_1^{LR}	2.54×10^{-2}	3.02×10^{-3}	1.86×10^{-5}	✓
	θ_2^{LR}	4.23×10^{-1}	2.40×10^{-1}	1.87×10^{-5}	✓
	θ_3^{LR}	2.01×10^{-1}	1.37×10^{-1}	2.24×10^{-5}	✓
	θ_4^{LR}	2.09×10^{-1}	1.72×10^{-1}	2.25×10^{-5}	✓
LT	θ_1^{LT}	2.88×10^{-1}	1.44×10^{-1}	4.78×10^{-6}	✓
	θ_2^{LT}	2.38×10^{-1}	1.28×10^{-1}	7.22×10^{-6}	✓
	θ_3^{LT}	2.57×10^{-1}	1.34×10^{-1}	7.02×10^{-6}	✓
	θ_4^{LT}	2.17×10^{-1}	1.21×10^{-1}	8.90×10^{-6}	✓
NLR/NLT	θ_1^{NLT}	1.48×10^{-2}	6.80×10^{-3}	1.02×10^{-6}	✓
	θ_2^{NLT}	$2.72 \times 10^{+2}$	$6.22 \times 10^{+1}$	8.21×10^{-6}	✓

accuracy improvement for the F-score when the sampling rate is 180s. When DBWES is compared with MLE approach, DBWES achieves 5.8% and 14.4% accuracy improvements under RMF for sampling rates of 60 s and 180 s. When compared the accuracy improvements under RMF and F-score, the accuracy improvements increase with the increase of the sampling rate, this indicates the weights obtained from our algorithm are more accurate than the empirical setting and MLE approaches with larger sampling rates.

TABLE 4.5: map matching accuracy

Sampling Rate (sec)	RMF			F-score		
	60	120	180	60	120	180
LR-MLE	0.208	0.227	0.270	0.920	0.882	0.858
LR-DBWES	0.198	0.214	0.242	0.934	0.902	0.874
LT-MLE	0.205	0.220	0.251	0.910	0.886	0.869
LT-DBWES	0.201	0.210	0.218	0.916	0.896	0.891
NLR-Empirical	0.216	0.240	0.284	0.906	0.873	0.844
NLR-MLE	0.206	0.222	0.260	0.918	0.886	0.865
NLR-DBWES	0.194	0.217	0.235	0.939	0.890	0.896

To evaluate our proposed DBWES for the complex trajectories, the map matching

accuracy in terms of RMF for the complex trajectories of three map matching algorithms are shown in Fig. 4.6. In general, the results are similar to the results in Table 4.5. It can be seen that our proposed DBWES significantly improves map matching accuracy under the three sampling rates. For the sampling rate of 60 s, DBWES achieves comparable accuracy with the MLE approach for *LT* features. When the sampling rate of 180 s, it can be observed in Fig. 4.6(c) that DBWES achieves up to 17.2% and 9.6% accuracy improvements compared with the empirical setting and MLE approaches respectively for the *NLR* and *NLT* features.

FIGURE 4.6: RMF for map matching algorithms for complex trajectories

4.4.5.3 Weights analysis

As weights quantify the importance of travelling features, in this experiment, we further investigate the weights determination regarding the travelling features given estimated weights distributions. Specifically, we investigate how the weight value of a given travelling feature varies for different travelling cases. For example, when comparing the case that origin and destination is far apart with the case that the two are close, the feature of route length may be less important (i.e., a smaller

weight value). In the case of origin and destination is far apart, it is likely that the lengths of all the feasible routes connecting O-D are far apart as well. Thus, a driver may be less concerned that one route is longer (probably only by a small amount) than the other. However, in the case that origin and destination is close, the driver may care more the route length compared with other travelling features. Thus, the feature of route length may be more important in this case. To investigate the weights determination regarding the travelling features, it is necessary to firstly identify travelling cases from historical routes and obtain the optimal weights for each case through sampling from our estimated weight distributions, then such relationships of weights towards different cases can be analyzed.

In this study, we identify travelling cases by grouping travelling routes regarding each travelling feature. For example, we collect and classify travelling routes into groups based on their route length, we then determine the optimal weights from the travelling routes in each group. To identify travelling cases for each travelling feature, we firstly obtain the value range of each travelling feature from the travelling routes and partition the value range of each travelling feature into multiple intervals. After that, we classify the travelling routes into multiple traveling cases, in each of which the value of traveling feature belongs to a partitioned interval. For example, we obtain the value range (0 m, 230 m) of the road length from the travelling routes and partition the value range into five sorted intervals: (0, 50 m], (50 m, 100 m], (100 m, 150 m], (150 m, 200 m], and (200 m, $\sim$), then the travelling routes belonging to each interval are collected.

To analyse how weights vary for different travelling cases, we calculate the average optimal weight value θ_j^w from the travelling routes for each travelling case given the estimated weights distributions, where w is the index of the travelling case for the travelling feature f_j . For the objective of comparing weights for different travelling features, we calculate the rate of change (ROC) for each travelling feature with the following definition:

$$RoC_j^w = \frac{\theta_j^w}{\theta_j^0} \quad (4.27)$$

where θ_j^0 represents the average optimal weight value from the interval with the smallest values of traveling feature f_j (e.g., the interval (0, 50m] for the feature of route length). As there are multiple travelling cases for each travelling feature, we examine the ROC of the weights with different travelling cases ($w = 1, 2, \dots$) for

each travelling feature, i.e., f_j . In this manner, a higher value of RoC_j^w indicates the weight θ_j increases with the increase value of the travelling feature f_j .

In this experiment, the weights for four route-related LR travelling features in terms of road length (f_1), road turning angles (f_2), number of left (f_3) and right turns (f_4) at road intersections for the driving cost model in Eq. 4.18 are studied. In addition, 100 travelling routes for all intervals are selected to determine ROC . The ROC of the four weights are shown in Fig. 4.7.

FIGURE 4.7: ROC of the weights for different travelling cases

In general, the ROC of the weight θ_j^{LR} decreases with an increase of f_j from Fig. 4.7(a) to Fig. 4.7(d). Specifically, the ROC of θ_1^{LR} decreases 40% when the value of road length (f_1) is larger than 200 m, whereas the ROC of θ_2^{LR} , θ_3^{LR} , and θ_4^{LR} increase in Fig. 4.7(a). This pattern can be explained that the importance of the road length decreases when the road length (f_1) becomes longer, and drivers begin to concern more about other travelling features. In addition, it can be found that the increase of ROC of θ_2^{LR} of road turning angles (f_2) is nearly 20%, which is larger than the ROC of θ_3^{LR} and θ_4^{LR} . This indicates that feature of road turning angles (f_2) is more considered by drivers than the number of left/right turns at road intersections (f_3 and f_4) when the roads become longer.

From Fig. 4.7(b), there is a nearly 70% decrement of the ROC of θ_2^{LR} when the travelling feature values related with turning angles (f_2) are larger than 0.2, whereas ROCs of θ_1^{LR} , θ_3^{LR} , and θ_4^{LR} increase with the increase of this feature value (f_2). This indicates the importance of the road turning angles becomes weak for large turning angles and drivers may pay more attention to other travelling features. Moreover, it can be observed that the distance of the decline of the ROC of θ_2^{LR} and the increase of the ROC of θ_1^{LR} is larger than that in Fig. 4.7(a), this indicates that drivers' driving preferences may be more likely to be changed when the travelling feature value of the road turning angle is large.

For the travelling features of the number of left/right turns at road intersections (f_3 and f_4), the ROC of θ_3^{LR} and θ_4^{LR} decrease up to 10% when f_3 and f_4 increase shown in Fig. 4.7(c) and Fig. 4.7(d) respectively. The decrements of the ROC of θ_3^{LR} and θ_4^{LR} are less than those of road length (f_1) and turning angles (f_2), therefore, we may infer that drivers may not consider the number of left/right turns as important as the road length and turning angles from this experiment.

In addition, from Fig. 4.7(c) and Fig. 4.7(d), both of the ROC of θ_3^{LR} and θ_4^{LR} decrease with the increase of the number of left/right turns (f_3 and f_4), this indicates a positive correlated relationship for θ_3^{LR} and θ_4^{LR} . Plus, the ROC of θ_1^{LR} and θ_2^{LR} remain constant, indicating that the number of left/right turns have slight influence on the weights for road length (f_1) and turning angles (f_2).

4.5 Summary

In this chapter, a distribution-based weights estimation system is proposed to estimate the weights in driving cost models, and therefore more precise local routes can be inferred for map matching algorithms. Different from the existing data-driven approaches with estimation bias due to inappropriate routes selection, our estimation from distributions inferred from city-scale traces is able to decrease the estimation bias. In addition, our proposed approach is suitable for both linear and non-linear driving cost models with either normalized or non-normalized travelling features. The experimental results show that our proposed DBWES is able to increase map matching accuracy significantly compared with the empirical setting and data-driven approaches.

Chapter 3 and Chapter 4 provide our proposed map matching algorithm and weights estimation approach for route inference. Chapter 5 and Chapter 6 give the proposed approaches for vehicle localization.

Chapter 5

E²T-CVL: An Efficient and Error-tolerant Approach for Collaborative Vehicle Localization

This chapter describes the proposed approach to localize the target vehicle (E^2T -CVL). The approach framework, the detailed approach design, the experiments and results are provided in the following sections.

5.1 Framework

FIGURE 5.1: Framework

The overall design of our proposed E²T-CVL is shown in Fig. 5.1. It consists of two main components: *preprocessing* and *position determination*. The component of preprocessing is responsible for generating the initial search space containing

the lanes near to the position of the GPS measurement g_T^i of the target vehicle T_i , considering the geometry information. The component of position determination determines the position of the target vehicle by using the relative positions and GPS measurements of nearby vehicles. Firstly, the candidate lanes generation module filters out impossible lanes (i.e., lanes that the target vehicle cannot be located in) by leveraging the road geometry and measured relative positions of nearby vehicles with the consideration of measurement errors. The remaining lanes (termed the candidate lanes) form the narrowed search space for the position of the target vehicle. Next, the candidate zones generation module further narrows the search space based on the measured relative positions of nearby vehicles and road geometry. Lastly, the optimal position determination module determines the position of the target vehicle. In this module, the candidate positions of the target vehicle are sampled from the candidate zones, and each sampled position is assigned a selection score calculated using a displacement-based selection method. The position with the maximum selection score is determined as the optimal position of the target vehicle. Simultaneously, the corresponding lane and road segment for the target vehicle are identified.

5.2 Detailed design of E^2T -CVL

5.2.1 Nearby lanes preparation

In this module, given the GPS measurement g_T^i , the nearby lanes NL_i for the target vehicle is prepared. To determine NL_i , the driving direction of the target vehicle and the distance between lanes and the GPS measurement g_T^i are considered. The lanes under one of the following conditions are considered impossible to be travelled by the target vehicle:

- The lanes are far from the GPS position of the target vehicle.
- The driving direction of the target vehicle does not satisfy the direction constraints of the lanes.

To determine NL_i for the target vehicle, the lanes in the city network are retrieved and the minimum great circle (MGC) distances between these lanes and the GPS

position of the target vehicle are calculated. When the lanes are within a given distance range and their direction constraints are satisfied by the driving direction of the target vehicle, they are selected as the nearby lanes for the target vehicle.

5.2.2 Candidate lanes generation

After obtaining the nearby lanes NL_i for the target vehicle T_i , this module filters out the impossible lanes from NL_i leveraging the road geometry and measured relative positions of nearby vehicles with the consideration of measurement errors.

To identify the impossible lanes from NL_i , we assume that the target vehicle drives on one of the nearby lanes NL_i , while obtaining the relative positions of nearby vehicles. Given the existence of the measurement errors of relative positions, whether nearby vehicles are driving on lanes can be determined by modelling their travelling areas containing the possible real positions. Next, each detected nearby vehicle can be identified as travelling on a lane when the intersections between lanes and the estimated travelling area are identified. If a detected nearby vehicle does not travel on a lanes with no intersection identified, it is impossible that the target vehicle travels on the assumed lane and it should be filtered out.

In this module, given an assumed real position of the target vehicle, the travelling area of each detected nearby vehicle is estimated as a circular area. More details of the proof is attached in Appendix A. The x-axis and y-axis positions of the circular area center are determined as follows:

$$\begin{aligned} Cen(N_i^j)_X &= X(T_i) + \hat{PD}(T_i, N_i^j)_X \\ Cen(N_i^j)_Y &= Y(T_i) + \hat{PD}(T_i, N_i^j)_Y \end{aligned} \quad (5.1)$$

In Eq. 5.1, $[X(T_i), Y(T_i)]$ is the assumed real position of the target vehicle, and $[\hat{PD}(T_i, N_i^j)_X, \hat{PD}(T_i, N_i^j)_Y]$ is the *measured position difference* ($\hat{PD}$) between the target vehicle T_i and nearby vehicle N_i^j determined from the measured relative distance $\hat{d}(T_i, N_i^j)$ and bearing angle $\hat{a}(T_i, N_i^j)$.

$$\begin{aligned} \hat{PD}(T_i, N_i^j)_X &= \hat{d}(T_i, N_i^j) \cos(\hat{a}(T_i, N_i^j)) \\ \hat{PD}(T_i, N_i^j)_Y &= \hat{d}(T_i, N_i^j) \sin(\hat{a}(T_i, N_i^j)) \end{aligned} \quad (5.2)$$

In addition, the radius of the circular area is determined as follows:

$$R = \sqrt{(-\varepsilon_d)^2 + (\varepsilon_a \hat{d}(T_i, N_i^j))^2} \quad (5.3)$$

where ε_d and ε_a are the measurement errors of the relative distance $\hat{d}(T_i, N_i^j)$ and bearing angle $\hat{a}(T_i, N_i^j)$ respectively, and ε_d and ε_a follow Gaussian distributions with the standard deviations ρ and ω respectively.

FIGURE 5.2: Approximate rectangular area

Since the target vehicle may travel along the assumed lane, the estimated circular areas with identical radius are consequently generated. Therefore, the travelling area of each detected nearby vehicle is estimated as a rectangular area parallel to the assumed lane, as shown in Fig 5.2 (the width of the rectangular area equals to the diameter of the circle areas), and the intersections between the rectangular area and lanes can be identified. Therefore, the assumed lane is considered one candidate lane for the target vehicle.

The process of the candidate lanes generation module is given in Algorithm 4.

5.2.3 Candidate zones generation

Since the length of some candidate lanes can be long (> 100 m), this module further narrows the search space on each candidate lane based on the measured relative positions of nearby vehicles and road geometry.

To determine the candidate zones for the target vehicle T_i , we see that the possible real position of the nearby vehicle N_i^j is contained within the intersections between

Algorithm 4: Candidate Lanes Generation

Input: Nearby lanes NL_i for the target vehicle T_i , Road geometry, Detected nearby vehicles N_i ;
Output: Candidate lanes CL_i for the target vehicle;
 $CL_i = \emptyset$;
for each lane $l_i^f \in NL_i$ **do**
 for each nearby vehicle $N_i^j \in N_i$ **do**
 Calculate measured position difference $\hat{P}D(T_i, N_i^j)$ between T_i and N_i^j as Eq. 5.2;
 Determine the parallel rectangular area for N_i^j ;
 Identify the intersections between the rectangular area and lanes for N_i^j ;
 if *Intersections are not identified* **then**
 break;
 if *Intersections can be identified for each nearby vehicle* **then**
 $CL_i = CL_i \cup l_i^f$;

the rectangular area and the lanes. Next, the zone Z_i^j for the target vehicle is identified as the area, from which the possible real position of the nearby vehicle N_i^j is contained within the intersections.

As multiple nearby vehicles may be detected by the target vehicle, therefore, the candidate zones for the target vehicle are obtained as the intersections of the zones from each detected nearby vehicle.

$$CZ_i = Z_i^1 \cap, \dots, \cap Z_i^{num} \quad (5.4)$$

where CZ_i is the nearby zones for the target vehicle, num is the number of nearby vehicles detected by the target vehicle, $Z_i^1, \dots, Z_i^{num}$ are the zones for the target vehicle generated from the nearby vehicle from N_i^1 to N_i^{num} respectively.

The process of the candidate zones generation module is given in Algorithm 5:

Algorithm 5: Candidate Zones Generation

Input: Candidate lanes CL_i for the target vehicle, Intersections of the nearby vehicles, Measured relative positions, Detected nearby vehicles N_i , Road geometry;
Output: Candidate zones CZ_i for the target vehicle;
Let $CZ_i = CL_i$;
for each nearby vehicle N_i^j in N_i **do**
 Determine the zones Z_i^j for the target vehicle from N_i^j ;
 $CZ_i = CZ_i \cap Z_i^j$;

5.2.4 Optimal position determination

In this module, the position of the target vehicle is determined by sampling candidate positions from the candidate zones. The selection scores for the candidate positions are determined by the proposed displacement-based selection method. Displacement is defined as the vector pointing from a vehicle position to its measured GPS position. Therefore, the displacement of the target vehicle in terms of a candidate position is obtained as the vector pointing from one candidate position to its GPS position. To decrease the influence of the measurement errors of relative positions, we infer the positions of nearby vehicles by offsetting the measurement errors. Similarly, the displacement of one detected nearby vehicle is obtained as the vector pointing from the inferred position and its GPS position. Next, the probability density function (PDF) of the displacement difference between the target vehicle and the nearby vehicles is derived in order to determine the selection scores.

Given the sampled candidate position set $SP_i = [SP_i^1, \dots, SP_i^m, \dots, SP_i^M]$ for the target vehicle from the candidate zones CZ_i the displacement of the target vehicle in terms of one candidate position SP_i^m is written as:

$$\text{Vec}(SP_i^m \rightarrow g_T^i) = g_T^i - SP_i^m \quad (5.5)$$

where g_T^i and SP_i^m are the GPS position and the sampled candidate position of the target vehicle T_i respectively.

To determine the displacement of each detected nearby vehicle, the position corresponding to each sampled candidate position of the target vehicle must first be inferred. We infer the position of each detected nearby vehicle by offsetting the measurement errors of relative positions as follows:

$$P_i^{jm} = \arg \min_{P_i^j} d(P_i^j, (SP_i^m + \hat{PD}(T_i, N_i^j) - \varepsilon^{PD})) \quad (5.6)$$

where $P_i^j \in \text{Intersections between the rectangular area and lanes}$

where P_i^j is the position contained within the intersections generated from the candidate lanes generation module, ε^{PD} is the estimated error of the measured position difference in order to offset the measurement errors of relative positions.

Following the proof in Appendix A, the x-axis and y-axis positions of ε^{PD} are determined as follows:

$$\begin{aligned}\varepsilon_x^{PD} &= R \cos(\hat{a}(T_i, N_i^j) + \beta) \\ \varepsilon_y^{PD} &= -R \sin(\hat{a}(T_i, N_i^j) + \beta) \\ \text{where } \beta &= \tan^{-1}\left(\frac{\varepsilon_a \hat{d}(T_i, N_i^j)}{\varepsilon_d}\right)\end{aligned}\tag{5.7}$$

where ε_d and ε_a are the measurement error of relative distance and bearing angle.

Therefore, $\hat{PD}(T_i, N_i^j) - \varepsilon^{PD}$ in Eq.5.6 indicates the position difference between the target vehicle T_i and nearby vehicle N_i^j after offsetting the measurement errors of relative positions. Since the nearby vehicle N_i^j drives on a lane, the position of N_i^j is determined by calculating the euclidean distance between the position P_i^j on a lane and the position $SP_i^m + \hat{PD}(T_i, N_i^j) - \varepsilon^{PD}$.

The displacement of the nearby vehicle N_i^j in terms of the inferred position P_i^{jm} and GPS position g_N^{ij} is determined as:

$$\text{Vec}(P_i^{jm} \rightarrow g_N^{ij}) = g_N^{ij} - P_i^{jm}\tag{5.8}$$

To determine the selection score for each candidate position, the displacement different (DD) between the target vehicle T_i and nearby vehicle N_i^j can be determined from Eq. 5.5 and Eq. 5.8 as:

$$DD(T_i, N_i^j, SP_i^m) = \text{Vec}(SP_i^m \rightarrow g_T^i) - \text{Vec}(P_i^{jm} \rightarrow g_N^{ij})\tag{5.9}$$

where $\text{Vec}(SP_i^m \rightarrow g_T^i)$ is the displacement for the target vehicle in terms of the candidate position SP_i^m , and $\text{Vec}(P_i^{jm} \rightarrow g_N^{ij})$ is the corresponding displacement for the nearby vehicle.

In this chapter, the probability density function (PDF) of the displacement difference in terms of the real position of the vehicle is derived to determine the selection scores. According to [67, 80], the accuracy of GPS measurements depends on the quality of the pseudorange measurement between the satellites and GPS receivers with the consideration of both common and uncommon error sources. However, as uncommon errors vary from vehicle to vehicle, the bias in GPS positioning caused

by uncommon errors is usually modelled as a vehicle-dependent zero-mean Gaussian distribution [68, 74]. Following the proof in Appendix B, the derived PDF of the displacement difference in terms of the real position of the vehicle follows a Gaussian distribution $N(0, \sigma_{NB_T^i}^2 + \sigma_{NB_N^{ij}}^2)$, where $\sigma_{NB_T^i}$ and $\sigma_{NB_N^{ij}}$ are the standard deviations of the position bias caused by non-common errors. Next, the selection score for each sampled candidate position is determined by considering the displacement difference and GPS measurements. As the target vehicle obtains the state information from all of the detected nearby vehicles, the selection score for each sampled candidate position is determined as follows:

$$SC(SP_i^m) = p(SP_i^m | g_T^i) \times \prod_{j=1}^{num} (p(DD(T_i, N_i^j, SP_i^m)) \times p(P_i^{jm} | g_N^{ij})) \quad (5.10)$$

where $p(DD(T_i, N_i^j, SP_i^m))$ is the score determined from the displacement difference $DD(T_i, N_i^j, SP_i^m)$ from Eq. 5.9 following the Gaussian distribution $N(0, \sigma_{NB_T^i}^2 + \sigma_{NB_N^{ij}}^2)$. num is the number of the nearby vehicles, $p(SP_i^m | g_T^i)$ and $p(P_i^{jm} | g_N^{ij})$ are the scores calculated from Eq. 5.11 based on the minimum great circle (MGC) distance from the position SP_i^m and P_i^{jm} to the GPS positions of the target vehicle T_i and nearby vehicle N_i^j as follows:

$$p(x | g_T^i) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{dist(x, g_T^i)^2}{2\sigma_g^2}} \quad (5.11)$$

where σ_g is the standard deviation of GPS measurement error, $dist(x, g_T^i)$ represents the minimum great circle (MGC) distance between one position x and the GPS position g_T^i .

Finally, the optimal position $\bar{P}_T^i$ of the target vehicle is determined from the sampled candidate positions as:

$$\bar{P}_T^i = \arg \max_{SP_i^m} (SC(SP_i^m)) \quad (5.12)$$

where $SP_i^m \in SP_i$

The process of the optimal position determination is given in Algorithm 6.

Algorithm 6: Optimal Position Determination

Input: GPS measurements of the target vehicle and nearby vehicles, Candidate zones, Measured relative positions, Detected nearby vehicles N_i , Road geometry;
Output: The optimal position $\bar{P}_T^i$ of the target vehicle
Sample candidate positions SP_i from the candidate zones CZ_i for the target vehicle T_i ;
for each sampled candidate position $SP_i^m \in SP_i$ **do**
 for each nearby vehicle $N_i^j \in N_i$ **do**
 Infer position P_i^{jm} of N_i^j as Eq. 5.6;
 Calculate displacement difference (DD) as Eq. 5.9;
 Calculate the selection score for the sampled candidate position SP_i^m as Eq. 5.10;
Determine the optimal position $\bar{P}_T^i$ of the target vehicle from SP_i as Eq. 5.12;

5.3 Performance evaluation and results

In this section, we investigate the performance of the proposed E^2T -CVL in terms of localization accuracy and efficiency. The trajectories of vehicles are generated from the traffic simulator SUMO [53]. Firstly, the simulation settings and two state-of-the-art collaborative vehicle localization strategies are described. The localization metrics used to evaluate localization accuracy and efficiency are then described. Lastly, the experimental results are discussed.

5.3.1 Simulation settings

Experiments are conducted on a road network from Midtown Manhattan, New York extracted from the OpenStreetMap [59] (Fig. 5.3). The road network comprises 12308 lanes, and the area has a vertical length of approximately 3.8 km and a horizontal length of approximately 2.7 km. For the target vehicle, 50 trajectories are randomly selected for evaluation. We randomly select 10 known positions from each trajectory with a constant sampling rate for localization evaluation. Therefore, a total of 500 known positions are prepared for evaluation. The positions of the detected nearby vehicles given each position of the target vehicle are randomly generated within a range of distance 50 m. Next, the GPS positions of the target vehicle and nearby vehicles are generated using the GPS Software Toolbox implemented in MATLAB with the consideration of GPS measurement errors.

FIGURE 5.3: New York city

5.3.2 Compared localization strategies

The localization accuracy and efficiency of the proposed E²T-CVL approach are assessed and compared relative to that of two state-of-the-art CVL approaches: Bayesian-based localization strategy and particle filter-based localization strategy. For the two existing CVL approaches, the GPS measurements of both the target vehicle and nearby vehicles are used. The Bayesian-based localization approach leverages the relative positions of nearby vehicles for localization, whereas the particle filter-based localization approach considers the road constraints around the nearby vehicles

Bayesian-based localization: The position of the target vehicle is determined through the MAP strategy, where the candidate positions of the target vehicle are sampled from the search space [77]. The probability for each candidate position is determined by considering the GPS measurements and relative positions of the nearby vehicles, with GPS measurement errors and measurement errors of relative positions modelled with Gaussian distributions.

Particle filter-based localization: In [67], the position of the target vehicle is determined through the particle filter-based strategy. In general, the candidate positions (particles) of the target vehicle are sampled from the search space around the GPS position. Next, the weight for each candidate position is determined by considering the road constraints around the nearby vehicles through V2V communications. Lastly, the position of the target vehicle is determined as the weighted average position of the candidate positions.

5.3.3 Parameters settings

For the nearby lanes preparation component, the distance range is set to 50 m. This means that the lanes with a MGC distance < 50 m to the GPS position of the target vehicle are considered to be nearby lanes. For the measurement errors of the relative positions, the standard deviation of the relative distances and bearing angles are set to 2.4 m and 0.1, respectively. Different standard deviations of the relative distances and bearing angles are investigated in the sensitivity analysis experiment. For the two localization strategies compared here, the number of sampled candidate positions of the target vehicle and nearby vehicles is set to 1000, and the number of iterations for the particle filter strategy is set to 20, in line with [67, 68]. In our proposed approach, the number of sampled candidate positions from the candidate zones is 50. This is appropriate considering the localization efficiency, where a larger value indicates higher localization accuracy but introduces additional computational overheads. The standard deviation σ_g of the GPS measurements in Eq. 5.11 is set as 10, in line with [56].

5.3.4 Evaluation metrics

In our experiments, the metrics of localization accuracy in terms of the position-level (PL), lane-level (LL) and segment-level (SL) are provided. The localization efficiency is then measured as the average execution time for localizing each GPS position .

Localization accuracy: To evaluate the localization accuracy, we adopt the matrices according to [54, 67] as follows:

$$PL = \frac{\sum_{i=1}^m |\tilde{P}_T^i - P_T^i|}{m} \quad (5.13)$$

$$LL = \frac{\text{The number of correct localized lanes}}{m} \quad (5.14)$$

$$SL = \frac{\text{The number of correct localized segments}}{m} \quad (5.15)$$

where m is the number of selected positions for each trajectory, which is set to 10 in our experiments. A smaller value of PL indicates higher localization accuracy. Different from the PL , LL and SL refer the ratios of the correct localization for each trajectory, therefore larger values of both LL and SL indicate higher localization accuracy.

Localization efficiency: The localization efficiency is measured as the average execution time for each GPS position as follows:

$$LE = \frac{\text{Total Execution Time}}{\text{Number of GPS positions}} \quad (5.16)$$

The average execution time of our proposed approach and compared CVL approaches are measured with a Dell Precision T3610 workstation with Intel(R) Xeon(R) CPU E5-1650v2@3.50GHz and 16GB RAM operated with a 64-bit Windows 10 system.

5.3.5 Experimental results

5.3.5.1 Accuracy results

The experimental results for localization accuracy in terms of PL , LL and SL are shown in Table 5.1. The number of nearby vehicles varies from 1 to 6. It is clear that the localization accuracy in terms of PL , LL , and SL for our proposed approach and the compared approaches increase with an increase number of nearby vehicles. This is due to the fact that a large number of nearby vehicles can significantly reduce the GPS measurement errors.

For the particle filter-based and Bayesian-based strategies, similar lane-level and segment-level accuracy are achieved. The particle filter-based localization strategy achieves 52% and 57% lane-level and segment-level accuracy, respectively, when the number of nearby vehicles is 6. For the same strategy, 41% and 46% in terms of lane-level and segment-level accuracy are achieved when the number of nearby vehicles is 1. Regarding position-level accuracy, the particle filter-based localization strategy achieves slightly better localization results than the Bayesian-based localization strategy when the number of nearby vehicles is larger than 2. This

is because the measurement errors of the relative positions of nearby vehicles introduce uncertainty in the Bayesian-based localization strategy, especially when the number of nearby vehicles is low. By contrast, the relative positions of nearby vehicles are not considered in the particle filter-base localization strategy.

In general, our approach outperforms the other two approaches with different number of nearby vehicles. For the position-level accuracy, our proposed E^2T -CVL approach is able to achieve 6.70 m compared with the accuracy of the compared two approaches when the number of nearby vehicles is 6. This shows that there are 16.5% and 14.5% accuracy improvements, respectively. For the lane-level and segment-level accuracy, our approach improves nearly 10% compared with the two approaches when the number of nearby vehicles is 6. When the number of nearby vehicle is 1, our approach achieves 8.21 m and 46 % in terms of the position-level and lane-level accuracy. The reason is that the selection sores in our strategy based on displacement differences are determined by offsetting the measurement errors of relative positions. Therefore our proposed E^2T -CVL tolerates the measurement errors of relative positions, especially when the number of nearby vehicles is small and the measurement errors are significant. Whereas the position of the target vehicle determined from the Bayesian-based and particle filter-based localization strategies are more likely to be influenced by the measurement errors.

TABLE 5.1: Average accuracy results with respect to the number of nearby vehicles using different methods

Approaches\Number of nearby vehicles		1	2	3	4	5	6
Bayesian-based	Position-level	8.92	8.75	8.44	8.26	8.14	8.02
	Lane-level	44%	45%	46%	48%	49%	49%
	Segment-level	51%	51%	52%	55%	56%	59%
Particle filter based	Position-level	9.03	8.84	8.41	8.22	8.06	7.84
	Lane-level	41%	43%	47%	49%	51%	52%
	Segment-level	46%	50%	52%	55%	57%	57%
E^2T -CVL	Position-level	8.21	8.09	7.14	7.02	6.84	6.70
	Lane-level	46%	50%	54%	58%	60%	61%
	Segment-level	52%	57%	60%	64%	64%	65%

5.3.5.2 Efficiency results

To investigate the efficiency of our proposed approach, the average execution time over 500 GPS positions from 50 trajectories is measured. The execution time of our approach includes four parts: nearby lanes preparation, candidate lanes generation, candidate zones generation and optimal position determination. The average execution times of our proposed approach and the compared two approaches with varying numbers of nearby vehicles are shown in Fig. 5.4.

It is shown that our proposed E²T-CVL has the smallest execution time compared with the two approaches with the different number of nearby vehicles. For example, our approach achieves approximately 0.22 s, whereas the average execution time of the Bayesian-based strategy is approximately 0.8 s, and the average execution time of the particle filter-based strategy is approximately over 0.3 s when the number of nearby vehicle is 6. In other words, our proposed E²T-CVL achieves approximately 4 times faster than the Bayesian-based strategy when the number of nearby vehicles is 6. The reason is that the search space for the position of the target vehicle has been significantly pruned through our pruning-based strategy (The experimental results in terms of the narrowed search space through the pruning-based can be found in Section 5.3.5.3). Due to the reason that the search space is smaller when there are more nearby vehicles, the execution time gaps between our proposed approach and the compared approaches become larger with the increase number of nearby vehicles.

For the proposed E²T-CVL approach, the increase of the execution time slows down slightly when the number of the nearby vehicle reaches 3, the reason is that the search space for the position of the target vehicle has been significantly decreased when the number of nearby vehicles reaches 3. For the Bayesian-based and particle filter-based strategies, the execution time increase linearly with the increase number of nearby vehicles. The reason is that the constant number of candidate positions are sampled for the target vehicle and nearby vehicles. It can be seen that the execution time of the Bayesian-based localization strategy reaches 0.4 s, and is the longest among the three approaches when the number of nearby vehicles is 3. The execution time of the particle filter-based localization strategy is shorter than the Bayesian-based localization strategy, the reason is that candidate positions are sampled for each nearby vehicle in the Bayesian-based localization approach.

FIGURE 5.4: Average execution time with the number of nearby vehicles

5.3.5.3 E^2T -CVL analysis

The search space for the position of the target vehicle from our pruning-based strategy, in terms of nearby lanes, candidate lanes, and candidate zones, is investigated with different number of nearby vehicles (5.2). Firstly, the average number and length of the nearby lanes for the target vehicle are summarized. Next, to examine candidate lanes generation, the number and length of the candidate lanes are summarized. Lastly, the average number and length of the candidate zones are reported. To evaluate the accuracy of the search space, the percentages of real positions contained within the candidate zones are calculated.

TABLE 5.2: Search space in terms of nearby lanes, candidate lanes and candidate zones

Lanes/Zones\ Number of the nearby vehicles		1	2	3	4	5	6
Nearby Lanes	Number	62	62	62	62	62	62
	Length (m)	2125.04	2125.04	2125.04	2125.04	2125.04	2125.04
Candidate Lanes	Number	22	18	16	15	13	12
	Length (m)	787.73	687.02	621.18	531.74	501.23	472.04
Candidate Zones	Number	10	7	7	6	5	4
	Length (m)	133.43	80.01	74.35	62.35	58.98	52.44
Percentages that the Real Positions being contained in the Candidate Zones		98%	97%	95%	95%	93%	93%

From Table 5.2, the number of nearby lanes and corresponding length remain constant with an increase number of nearby vehicles. The reason is that the search space is not narrowed. It can be seen that there are average 62 nearby lanes with a average length of 2125.04 m prepared for the target vehicle. When the number of

nearby vehicles increases from 1 to 6, the number of the candidate lanes decreases from 22 to 12, and the corresponding length decreases from 787.73 m to 472.04 m. Especially, the average length of the candidate lanes is narrowed from 2125.04 m to 472.04 m when the number of nearby vehicle reaches 6. The experimental results meet the objective of candidate lanes generation module, where more lanes are considered as impossible and filtered out with a larger number of nearby vehicles.

For candidate zones, it is clear the search space is largely narrowed. When the number of nearby vehicles increases from 1 to 6, the number of candidate zones decreases from 10 to 4, and the corresponding average length decreases from 133.43 m to 52.44 m. When there is 1 nearby vehicle, our approach is able to filter out 93.7% of the search space approximately. This guarantees high localization efficiency with a small number of nearby vehicles. When there are 6 nearby vehicles, 97.5% of the search space is approximately pruned. This guarantees high localization efficiency with a large number of nearby vehicles.

For the percentages of real positions contained within the candidate zones, it can be seen that the percentages are high with different number of nearby vehicles. When there is 1 nearby vehicle, the percentage is 98%. This indicates a high probability that the search space contains the real position. In addition, the percentage becomes lower when the number of nearby vehicles increases. For example, the percentage becomes 93% when the number of nearby vehicle is 6. The reason is that when the search space is smaller, the risk that the real position of the target vehicle being out of the candidate zones becomes higher.

5.3.5.4 Sensitivity analysis

In Eq. 5.3, the radius R of circular area is determined from the measurement errors of relative positions, and R becomes larger with an increase of measurement errors. Consequently, less number of candidate lanes are considered as impossible, and larger search space are generated. In this experiment, we investigate the influence of measurement errors on search space and localization accuracy, respectively.

According to the measurement errors of relative positions according to [17, 68, 77], the standard deviations of the measurement errors in terms of relative distances and bearing angles are varied from 0.8 m to 4.8 m, and from 0.01 to 0.31, respectively.

In addition, the number of nearby vehicles is set as 2 for comparison. Then the search space are shown in Table 5.3 and 5.4, respectively.

TABLE 5.3: Sensitivity analysis in terms of ρ

Lanes/Zones\ ρ (m)		0.8	1.6	2.4	3.2	4.0	4.8
Nearby Lanes	Number	62	62	62	62	62	62
	Length (m)	2125.04	2125.04	2125.04	2125.04	2125.04	2125.04
Candidate Lanes	Number	16	17	18	20	23	25
	Length (m)	570.20	620.73	687.02	762.29	823.85	902.63
Candidate Zones	Number	5	6	7	9	10	12
	Length (m)	45.23	62.88	80.01	98.04	128.90	171.41

TABLE 5.4: Sensitivity analysis in terms of ω

Lanes/Zones\ ω		0.01	0.07	0.13	0.19	0.25	0.31
Nearby Lanes	Number	62	62	62	62	62	62
	Length (m)	2125.04	2125.04	2125.04	2125.04	2125.04	2125.04
Candidate Lanes	Number	17	17	19	21	23	26
	Length (m)	620.73	622.61	732.12	791.31	824.32	934.62
Candidate Zones	Number	6	7	8	9	10	13
	Length (m)	62.88	64.65	88.38	105.47	129.21	182.56

From Table 5.3, the average number and length of the nearby lanes remain constant with the different values of ρ . The average number and length of the candidate lanes and candidate zones become larger with an increase value of ρ . When the value of ρ reaches 0.8 m, 16 lanes are prepared as the candidate lanes, and the average length of these candidate lanes is approximately 45.23 m. This indicates nearly 97.9% of the search space for the position of the target vehicle is filtered out. When the value of ρ is 4.8 m, both the number of the candidate lanes and candidate zones increase to 25 and 12 respectively, and nearly 91.9% of the search space is filtered out. From Table 5.4, the average number and length of the nearby lanes remain constant with the different values of ω . The average number and length of the candidate lanes and candidate zones become larger with the increase of ω . The candidate lanes are largely filtered out when the value of ω is 0.01. This indicates that there is up to 72.6% of the number of lanes are filtered out. It can be also inferred that approximately 97.0% of the search space in terms of the candidate zones is filtered out.

To investigate the influence of the measurement errors, the position-level accuracy of the GPS measurements, the Bayesian-based strategy, and our approach are reported with the different values of ρ and ω (Fig. 6.7 and Fig. 6.8). We can find the average GPS errors are approximately 13 m. For the Bayesian-based and our proposed E²T-CVL approaches, the localization accuracy decrease with an increase values of ρ and ω . When the value of ρ is 0.8 m, the localization accuracy of the Bayesian-based strategy is approximately 7.84 m, and our proposed E²T-CVL approach achieves 7.21 m. This indicates there is a approximate 8.03% accuracy improvement. When the value of ρ increases, the gap between the accuracy of the proposed E²T-CVL and the Bayesian-based strategies becomes larger. When the value of ρ is 4.8 m, our proposed approach achieves 25.3% and 11.8% accuracy improvements compared with the GPS measurements and Bayesian-based strategy. The reason is that the accuracy of the Bayesian-based strategy is influenced by the measurement errors of relative distances. Whereas our proposed displacement-based selection method is able to tolerate the measurement errors. Similarly, when the value of ω is 0.01, the localization accuracy of the Bayesian-based strategy is 8.21 m, and our proposed E²T achieves nearly 7.3 m. When the value of ω increases to 0.31, the accuracy of the Bayesian-based strategy achieves 11.15 m, and our proposed E²T achieves a nearly 10.8% accuracy improvement. This indicates that our approach is able to tolerate the measurement errors of bearing angles.

FIGURE 5.5: Sensitivity analysis

5.3.5.5 Cases analysis

In this section, we investigate the localization accuracy under different driving cases, which commonly result in localization errors. In this experiment, the localization accuracy for our proposed algorithm is compared with the two localization

algorithms. The three driving cases which may result in localization errors are shown in Fig. 5.6 as follows:

Case 1: Lanes with different direction constraints: When a driver approaches a road intersection linking multiple lanes with the different constraints of travelling directions, the position of the target vehicle is easily to be localized to a false lane near to the intersection.

Case 2: Parallel lanes: This case involves parallel lanes with the same driving direction and similar geometry information. When the GPS position of the target vehicle is measured between two parallel lanes, the vehicle can be localized to the position on a parallel lane.

Case 3: Roundabout: This case is a specific case of case 1. When a driver drives on a roundabout with curved lanes, the target vehicle is easily be localized to the false lane.

FIGURE 5.6: Three driving cases

Since drivers may travel at any position on a given lane, therefore, multiple positions on the lane are leveraged for investigation by assuming that these positions are from the target vehicle. To generate these positions, we uniformly sampled 50 positions on the given lanes as the real positions for the target vehicle. Similarly, the number of nearby vehicles is randomly generated and varies from 1 to 4. The position-level, lane-level, and segment-level localization accuracy for the three traveling cases are shown in Table 5.5.

It can be observed that our proposed approach outperforms the two localization strategies in terms of position-level, lane-level and segment-level for three cases. For case 1, our proposed approach achieves 4.91 m of the position-level accuracy when

TABLE 5.5: Average accuracy results with respect to the number of the nearby vehicles under three cases

Cases	Algorithms\Number of nearby vehicles		1	2	3	4
Case 1	Bayesian-based	Position-level	9.10	8.32	7.74	7.54
		Lane-level	68%	74%	78%	78%
		Segment-level	70%	82%	86%	88%
	Particle filter based	Position-level	9.17	8.66	8.24	7.94
		Lane-level	66%	74%	74%	76%
		Segment-level	68%	80%	82%	84%
	E ² T-CVL	Position-level	6.16	5.63	5.33	4.91
		Lane-level	84%	88%	88%	90%
		Segment-level	88%	90%	90%	94%
Case 2	Bayesian-based	Position-level	8.01	7.52	7.34	7.04
		Lane-level	72%	74%	74%	76%
		Segment-level	78%	80%	80%	84%
	Particle filter based	Position-level	8.02	7.43	7.08	6.87
		Lane-level	72%	74%	76%	80%
		Segment-level	76%	78%	82%	86%
	E ² T-CVL	Position-level	4.93	4.52	4.02	3.87
		Lane-level	88%	88%	92%	94%
		Segment-level	90%	90%	96%	96%
Case 3	Bayesian-based	Position-level	6.94	6.36	5.82	5.51
		Lane-level	82%	84%	88%	88%
		Segment-level	82%	84%	88%	90%
	Particle filter based	Position-level	6.93	6.33	5.84	5.63
		Lane-level	82%	84%	88%	88%
		Segment-level	84%	84%	88%	90%
	E ² T-CVL	Position-level	5.45	5.12	4.89	4.67
		Lane-level	90%	92%	94%	94%
		Segment-level	92%	94%	94%	96%

the number of nearby vehicle is 4, whereas the Bayesian-based strategy achieves 7.54 m, and the particle filter-based localization strategy achieves 7.94 m. For the lane-level and segment-level accuracy, the localization accuracy of our approach reaches 90% and 94% respectively, which show significant accuracy improvements compared with the two approaches. When the number of nearby vehicle is 1, our approach achieves 6.16 m in terms of the position-level accuracy, whereas over 9.0 m is achieved for the two compared approaches. Similarly, for the lane-level and segment-level accuracy, our approach is able to achieve 84% and 88% respectively, which are significantly better than other two approaches. The reason is that our pruning-based strategy significantly reduces the search space when the lanes are

connected by intersections, and the real position of the target vehicle can be more easily identified by our proposed strategy than the compared approaches. For case 2, the parallel lanes with similar geometry condition introduce localization difficulties. From the table, our proposed approach achieves 3.87 m, 94% and 96% in terms of the position-level, lane-level and segment-level accuracy respectively when the number of nearby vehicles is 4. When our proposed approach is compared with the Bayesian-based strategy, 18% and 12% improvements in terms of the lane-level and segment-level accuracy are achieved when the number of nearby vehicles is 4. In addition, our proposed approach achieves 14% and 10% improvements in terms of the lane-level and segment-level accuracy compared with the particle filter-based strategy. When the number of nearby vehicle is 1, our approach is able to achieve 88% of the lane-levels accuracy and 90% of the segment-level accuracy, which are significantly higher than the compared two localization strategies. The reason is that the parallel lanes with similar geometry information decrease the accuracy of the particle filter-based strategy, and the accuracy of Bayesian-based strategy is influenced by the measurement errors of relative positions of nearby vehicles. For case 3, the Bayesian-based and particle filter-based localization strategies have the similar position-level, lane-level, and segment-level accuracy with the different number of nearby vehicles. Our proposed approach achieves higher localization accuracy compared with the two approaches. For example, our proposed approach achieves 6% improvements for both the lane-level and segment-level accuracy compared with the Bayesian-based localization strategy when the number of nearby vehicle is 4. When the number of nearby vehicle is 1, there are 8% and 10% improvements for both the lane-level and segment-level accuracy compared with the Bayesian-based localization strategy. When our proposed is compared with the particle filter-based localization strategy, 8% improvements in terms of the lane-level and segment-level accuracy are achieved.

5.4 Summary

In this chapter, we propose an efficient and error-tolerant approach termed E²T-CVL to localize the target vehicle. We adopt a pruning-based strategy to narrow the search space for the position of the target vehicle with the consideration of the relative positions of nearby vehicles. To obtain the position of the target vehicle,

candidate positions are sampled from the candidate zones. Selection scores for the candidate positions are determined from the proposed displacement-based selection method, in which the measurement errors of relative positions are largely cancelled out in the calculation of displacement differences. In the next chapter, an approach to localize the nearby vehicles is provided.

Chapter 6

Bayesian-based Absolute Positions Estimation for the Nearby Vehicles through Vehicle-to-Vehicle Communications

This chapter describes the proposed localization approach to localize the nearby vehicles. The approach framework, the detailed approach design, the experiments and results are provided in the following sections.

6.1 Framework

The overall design of our proposed approach is shown in Fig. 6.1. It consists of three components: *Absolute position estimation for the target vehicle*, *Centers of sampling areas determination* and *Bayesian-based positions estimation*. After the first component, the absolute position of the target vehicle is estimated through the Kalman filter strategy, and the estimated absolute position of the target vehicle is utilized to determine the centers of sampling areas for the nearby vehicles. Since the GPS measurement and a motion model by considering vehicle's velocities and accelerations of the target vehicle are utilized for estimation, the accuracy of the

estimated absolute position of the target vehicle at the time t_i is higher than the measured GPS position. Then, the center of the sampling area for each nearby vehicle is determined from the second component, leveraging the GPS measurements and the absolute position of the target vehicle as well as the measured relative position. Next, the component of Bayesian-based positions estimation is responsible for estimating the absolute position of each detected nearby vehicle. In this component, candidate positions are sampled given the centers of sampling areas for both the target vehicle and each nearby vehicle, where the center of the sampling area for the target vehicle is the absolute position from the first component. After that, the posterior probability is calculated for each candidate position of each nearby vehicle by considering the relative positions and the GPS measurements. Finally, the candidate position with the maximum posterior probability is determined as the absolute position of the nearby vehicle.

FIGURE 6.1: Framework

6.2 Detailed design of the proposed approach

6.2.1 Absolute position estimation for the target vehicle

In this component, the absolute position of the target vehicle is estimated through the Kalman filter strategy with the consideration of a motion model and the GPS measurement. In the motion model, the driving velocity and acceleration of the target vehicle are leveraged.

In the motion model, the absolute position of the target vehicle at the time t_i is determined as follows:

$$S_i = FM * S_{i-1} + V_{i-1} \quad (1)$$

where $S_i = [p_x^i, \dot{p}_x^i, \ddot{p}_x^i, p_y^i, \dot{p}_y^i, \ddot{p}_y^i]$ is the vector of the target vehicle at time t_i , where $p_x^i, \dot{p}_x^i$ and $\ddot{p}_x^i$ are the absolute position, velocity and acceleration in the x-direction. $p_y^i, \dot{p}_y^i$ and $\ddot{p}_y^i$ are the absolute position, velocity and acceleration in the y-direction. FM is the transition matrix based on a constant acceleration model. V_{i-1} is the process noise, which describes the uncertainty of the position with the covariance matrix as follows:

$$\begin{aligned} Q_{i-1} &= E\{V_{i-1}V_{i-1}^T\} \\ &= \text{diag}(\sigma_{p_x^i}^2, \sigma_{\dot{p}_x^i}^2, \sigma_{\ddot{p}_x^i}^2, \sigma_{p_y^i}^2, \sigma_{\dot{p}_y^i}^2, \sigma_{\ddot{p}_y^i}^2) \end{aligned} \quad (2)$$

where diag represents the diagonal matrix.

The observation model of the target vehicle is expressed as follows:

$$Z_i = HS_i + W_i \quad (3)$$

where Z_i is the measurement vector representing the GPS measurement of the target vehicle, W_i is measurement noise representing the measurements uncertainty of Z_i . H is the observation matrix defined according to [77] as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4)$$

The absolute position $\hat{P}_T^i$ of the target vehicle is determined from Eq. 1 and Eq. 3 through the Kalman filter strategy. The details of the Kalman filter strategy can be found in [8].

6.2.2 Centers of sampling areas determination

This component is responsible for determining the centers of sampling areas for the nearby vehicles in order to make the sampled candidate positions be closer to the real positions of the nearby vehicles than their GPS positions.

To determine the center of sampling area for the nearby vehicle N_i^j (j is the index of the detected nearby vehicle), we firstly infer the conditional position of N_i^j from the

absolute position $\dot{P}_T^i$ of the target vehicle T_i using the measured relative position. The inferred conditional position $\dot{P}_N^{ij}$ of N_i^j is determined as:

$$\dot{P}_N^{ij} = \dot{P}_T^i + \begin{bmatrix} \hat{d}^{(ij)} \cos(\hat{a}^{(ij)}) \\ \hat{d}^{(ij)} \sin(\hat{a}^{(ij)}) \end{bmatrix} \quad (5)$$

where $\hat{d}^{(ij)}$ and $\hat{a}^{(ij)}$ are the measured relative distance and bearing angle of the nearby vehicle N_i^j . $\hat{d}^{(ij)} \cos(\hat{a}^{(ij)})$ and $\hat{d}^{(ij)} \sin(\hat{a}^{(ij)})$ are the relative positions in the x-direction and y-direction respectively.

To incorporate the GPS measurements of the nearby vehicles, the center of sampling area for the nearby vehicle N_i^j is determined by the linear combination of $\dot{P}_N^{ij}$ and the GPS position g_N^{ij} of the nearby vehicle N_i^j as follows:

$$P_{Sam}^{ij} = g_N^{ij} + \alpha \times (\dot{P}_N^{ij} - g_N^{ij}) \quad (6)$$

where $\dot{P}_N^{ij} - g_N^{ij}$ is the position deviation between $\dot{P}_N^{ij}$ and g_N^{ij} . α is called the balance factor between $\dot{P}_N^{ij}$ and g_N^{ij} . A large value of α indicates the center position is close to $\dot{P}_N^{ij}$, and a small value of α indicates the center position is close to g_N^{ij} . As a result, when the measurement errors of GPS measurements are large, the centers determined from our approach can be closer to the real positions of the nearby vehicles than their GPS positions.

6.2.3 Bayesian-based positions estimation

In this component, the absolute position of each detected nearby vehicle is determined through the Bayesian-based strategy, where the GPS positions and relative positions are considered as the ‘prior’ information, and the absolute positions of the nearby vehicles are considered as the ‘posterior’ information. Then the positions of the nearby vehicles can be estimated from the Maximum A Posterior (MAP) estimation, where candidate positions are sampled for each nearby vehicle and the target vehicle around the center of sampling areas (The center of sampling area for the target vehicle is determined as $\dot{P}_T^i$ in Eq. 5). Then, the posterior probability is computed for each candidate position of the nearby vehicles using the GPS measurements and relative positions.

To derive the Bayesian formula, the observed absolute positions of the nearby vehicles given an absolute position of the target vehicle is firstly given as:

$$p^{(ij)} = p^{(i)} + \begin{bmatrix} d^{(ij)} \cos(a^{(ij)}) \\ d^{(ij)} \sin(a^{(ij)}) \end{bmatrix} \quad (7)$$

where p^i is an absolute position of the target vehicle, and $p^{(ij)}$ is the corresponding absolute position of the detected nearby vehicle N_i^j . $d^{(ij)} \cos(a^{(ij)})$ and $d^{(ij)} \sin(a^{(ij)})$ are the error-free relative positions in the x-direction and y-direction respectively.

With the consideration of real driving scenarios, where the number of the nearby vehicles can be small or large, we give the positions estimation by considering the number of the nearby vehicles detected by the target vehicle: *one nearby vehicle* and *num nearby vehicles* respectively.

6.2.3.1 Scenario 1: One nearby vehicle

When there is one nearby vehicle detected by the target vehicle, the estimated absolute position $\bar{P}^{(j)}$ of the nearby vehicle N_i^j is obtained through the maximization of the posterior probability as follows:

$$\bar{P}^{(j)} = \arg \max_{p^{(j)}} f_{P^{(j)}|P^{(ij)}}(p^{(j)}|p^{(ij)}) \quad (8)$$

where $f_{P^{(j)}|P^{(ij)}}(p^{(j)}|p^{(ij)})$ is the posterior probability for one candidate position $p^{(j)}$ of the nearby vehicle N_i^j according to the Bayesian theorem as follows:

$$f_{P^{(j)}|P^{(ij)}}(p^{(j)}|p^{(ij)}) = \frac{f_{P^{(ij)}|P^{(j)}}(p^{(ij)}|p^{(j)}) \times f_{P^{(j)}}(p^{(j)})}{f_{P^{(ij)}}(p^{(ij)})} \quad (9)$$

where $f_{P^{(ij)}|P^{(j)}}(p^{(ij)}|p^{(j)})$ is the likelihood of the observed position of the nearby vehicle N_i^j , and $f_{P^{(j)}}(p^{(j)})$ is the prior probability for the candidate position $p^{(j)}$ of N_i^j following the Gaussian distribution from the GPS position g_N^{ij} as follows:

$$P^{(j)} = f_{P^{(j)}}^{GPS}(p^{(j)}) \sim N(g_N^{ij}, \sigma_{g^{(j)}}^2) \quad (10)$$

where $\sigma_{g^{(j)}}$ is the standard deviation of the GPS measurement errors.

To obtain the denominator $f_{P^{(ij)}(p^{(ij)})}$ in Eq. 9 according to the observation model in Eq. 7, the cumulative probability of $f_{P^{(ij)}(p^{(ij)})}$ is as follows:

$$F_{P^{(ij)}(p^{(ij)})} = \int_{-\infty}^{+\infty} \int_{-\infty}^{d(p^{(i)}, p^{(ij)})} \int_{-2\pi}^{a(p^{(i)}, p^{(ij)})} f_{P^{(i)} D^{(ij)} A^{(ij)}}(p^{(i)}, d^{(ij)}, a^{(ij)}) \times da^{(ij)} dd^{(ij)} dp^{(i)} \quad (11)$$

where $d^{(ij)}$ and $a^{(ij)}$ are the relative distance and bearing angle between the target vehicle and nearby vehicle respectively.

After differentiating with respect to $p^{(ij)}$, Eq. 12 can be obtained from Eq. 11 as follows:

$$f_{P^{(ij)}(p^{(ij)})} = \int_{-\infty}^{+\infty} f_{P^{(i)} D^{(ij)} A^{(ij)}}(p^{(i)}, d(p^{(i)}, p^{(ij)}), a(p^{(i)}, p^{(ij)})) \times dp^{(i)} \quad (12)$$

Since the position $p^{(i)}$, relative distance $D^{(ij)}$ and bearing angle $A^{(ij)}$ are obtained from different data sources, therefore they are independent, and we have Eq. 13 from Eq. 12 as follows:

$$f_{P^{(ij)}(p^{(ij)})} = \int_{-\infty}^{+\infty} f_{P^{(i)}}(p^{(i)}) f_{D^{(ij)}}(d(p^{(i)}, p^{(ij)})) f_{A^{(ij)}}(a(p^{(i)}, p^{(ij)})) \times dp^{(i)} \quad (13)$$

In this work, the candidate position $p^{(i)}$ of the target vehicle and the candidate position $p^{(j)}$ of the nearby vehicle are assumed to be independent, therefore the likelihood $f_{P^{(ij)}|P^{(j)}}(p^{(ij)}|p^{(j)})$ of the observed position of the nearby vehicle N_i^j can be rewritten as follows:

$$f_{P^{(ij)}|P^{(j)}}(p^{(ij)}|p^{(j)}) = \int_{-\infty}^{+\infty} f_{P^{(i)}}(p^{(i)}) f_{D^{(ij)}}(d(p^{(i)}, p^{(j)})) f_{A^{(ij)}}(a(p^{(i)}, p^{(j)})) \times dp^{(i)} \quad (14)$$

Therefore, the denominator $f_{P^{(ij)}(p^{(ij)})}$ in Eq. 9 can be derived from Eq. 14 as follows:

$$\begin{aligned}
 f_{P^{(ij)}(p^{(ij)})} &= \\
 &\int_{-\infty}^{+\infty} f_{P^{(j)}}(p^{(j)}) f_{P^{(ij)}|P^{(j)}}(p^{(ij)}|p^{(j)}) dp^{(j)} = \\
 &\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_{P^{(j)}}(p^{(j)}) f_{P^{(i)}}(p^{(i)}) f_{D^{(ij)}}(d(p^{(i)}, p^{(j)})) \\
 &\times f_{A^{(ij)}}(a(p^{(i)}, p^{(j)})) dp^{(i)} dp^{(j)}
 \end{aligned} \tag{15}$$

Since the error-free relative distances and bearing angles of the nearby vehicle are not available, we assume that the probability $f_{D^{(ij)}}(a(p^{(i)}, p^{(j)}))$ and $f_{A^{(ij)}}(a(p^{(i)}, p^{(j)}))$ follow Gaussian distributions as follows:

$$\begin{aligned}
 f_{D^{(ij)}}(d(p^{(i)}, p^{(j)})) &= f_{D^{(ij)}}^{Sen}(\hat{d}^{(ij)}) \sim N(\hat{d}^{(ij)}, \sigma_{d^{(ij)}}^2) \\
 f_{A^{(ij)}}(a(p^{(i)}, p^{(j)})) &= f_{A^{(ij)}}^{Sen}(\hat{a}^{(ij)}) \sim N(\hat{a}^{(ij)}, \sigma_{a^{(ij)}}^2)
 \end{aligned} \tag{16}$$

where $\hat{d}^{(ij)}$ and $\hat{a}^{(ij)}$ are the measured relative distance and bearing angle of the nearby vehicle N_i^j . $\sigma_{d^{(ij)}}$ and $\sigma_{a^{(ij)}}$ are the standard deviation of the measurement errors of the relative distance and bearing angle.

After that, by substituting Eq. 10, Eq. 14, Eq. 15 and Eq. 16 in Eq. 9, the posterior probability for the candidate position $p^{(j)}$ of the nearby vehicle N_i^j can be obtained from Eq. 17.

$$\begin{aligned}
 &f_{P^{(j)}|P^{(i)}}(p^{(j)}|p^{(i)}) \\
 &= \frac{\int_{-\infty}^{+\infty} f_{P^{(i)}}^{GPS}(p^{(i)}) f_{D^{(ij)}}^{Sen}(d(p^{(i)}, p^{(j)})) dp^{(i)} f_{A^{(ij)}}^{Sen}(a(p^{(i)}, p^{(j)})) dp^{(i)} \times f_{P^{(j)}}^{GPS}(p^{(j)})}{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_{P^{(j)}}^{GPS}(p^{(j)}) f_{P^{(i)}}^{GPS}(p^{(i)}) f_{D^{(ij)}}^{Sen}(d(p^{(i)}, p^{(j)})) f_{A^{(ij)}}^{Sen}(a(p^{(i)}, p^{(j)})) dp^{(i)} dp^{(j)}}
 \end{aligned} \tag{17}$$

6.2.3.2 Scenario 2: num nearby vehicles (num > 1)

We now consider the scenario when there are num nearby vehicles detected by the target vehicle, their absolute positions $\bar{P}^{1...num}$ can be estimated as follows:

$$\begin{aligned}
 \bar{P}^{1...num} &= \\
 &\arg \max_{p^{(1...num)}} f_{P^{(1...num)}|P^{(i1)}...P^{(inum)}}(p^{(1...num)}|p^{(i1)}...p^{(inum)})
 \end{aligned} \tag{18}$$

where $f_{P(1...num)|P(i1)...P(inum)}(p^{(1...num)}|p^{(i1)}...p^{(inum)})$ is the posterior probability for the num nearby vehicles according to the Bayesian theorem in Eq. 19.

$$\begin{aligned} & f_{P(1...num)|P(i1)...P(inum)}(p^{(1...num)}|p^{(i1)}...p^{(inum)}) \\ &= \frac{f_{P(i1)...P(inum)|P(1)...P(num)}(p^{(i1)}...p^{(inum)}|p^{(1)}...p^{(num)}) \times \Phi(p^{(1)}...p^{(num)})}{f_{P(i1)...P(inum)}(p^{(i1)}...p^{(inum)})} \end{aligned} \quad (19)$$

where $f_{P(i1)...P(inum)|P(1)...P(num)}(p^{(i1)}...p^{(inum)}|p^{(1)}...p^{(num)})$ is the likelihood of the observed position of the nearby vehicle from N_i^1 to N_i^{num} , and $\Phi(p^{(1)}...p^{(num)})$ is the prior probability for the candidate positions of the nearby vehicles.

Since the GPS measurements of the nearby vehicles are assumed to be independent, the prior probability for the candidate positions of the nearby vehicles can be written as:

$$\Phi(p^{(1)}...p^{(j)}...p^{(num)}) = \prod_{j=1}^{num} f_{P(j)}^{GPS}(p^{(j)}) \quad (20)$$

where $f_{P(j)}^{GPS}(p^{(j)})$ is the probability for the candidate position of the nearby vehicle N_i^j and it follows the Gaussian distribution $N(g_N^{ij}, \sigma_{g(j)}^2)$.

Similar, the candidate positions of the nearby vehicles are also assumed to be independent, the likelihood of the observations can be written as Eq. 21 according to Eq. 14:

$$\begin{aligned} & f_{P(i1)...P(inum)|P(1)...P(num)}(p^{(i1)}...p^{(inum)}|p^{(1)}...p^{(num)}) \\ &= \prod_{j=1}^{num} f_{P(ij)|P(j)}(p^{(ij)}|p^{(j)}) \\ &= \prod_{j=1}^{num} \int_{-\infty}^{+\infty} f_{P(i)}(p^{(i)}) f_{D(ij)}(d(p^{(i)}, p^{(j)})) f_{A(ij)}(a(p^{(i)}, p^{(ij)})) dp^{(i)} \end{aligned} \quad (21)$$

After that, the denominator of Eq. 19 can be determined from Eq. 20 and Eq. 21 as follows:

$$\begin{aligned} & f_{P(i1)...P(inum)}(p^{(i1)}...p^{(inum)}) = \\ & \int_{P(1)...P(num)} \Phi(p^{(1)}...p^{(num)}) \prod_{j=1}^{num} f_{P(ij)|P(j)}(p^{(ij)}|p^{(j)}) \\ & \times dp^{(1)}...dp^{(num)} \end{aligned} \quad (22)$$

Finally, by substituting Eq. 16, Eq. 20, Eq. 21 and Eq. 22 in Eq. 19, the posterior probability for the candidate positions of the num nearby vehicles can be determined from Eq. 22.

$$\begin{aligned}
 & f_{P^{(1...n)}|P^{(i1)}...P^{(inum)}}(p^{(1...num)}|p^{(i1)}...p^{(inum)}) \\
 &= \frac{\prod_{j=1}^{num} \int_{-\infty}^{+\infty} f_{P^{(i)}}(p^{(i)}) f_{D^{(ij)}}^{Sen}(\hat{d}^{(ij)}) f_{A^{(ij)}}^{Sen}(\hat{a}^{(ij)}) dp^{(i)} \times \prod_{j=1}^{num} f_{P^{(j)}}^{GPS}(p^{(j)})}{\int_{P^{(1)}...P^{(inum)}} \prod_{j=1}^{num} f_{P^{(j)}}^{GPS}(p^{(j)}) \prod_{j=1}^{num} f_{P^{(ij)}|P^{(j)}}(p^{(ij)}|p^{(j)}) dp^{(1)}...dp^{(num)}}
 \end{aligned} \tag{22}$$

The pseudocode of the proposed approach is given in Algorithm 7

6.3 Experiments

In this section, we investigate the localization accuracy of the proposed approach. The dataset and metrics for accuracy evaluation are firstly described, then the parameters setting and methods for comparison are given. After that, the experimental results of our proposed approach are discussed.

6.3.1 Experiments setup

6.3.1.1 Dataset

The real dataset consisting of the trajectories of the vehicles from the area of southbound US 101 [61] is used for evaluation. The area is shown in Fig 6.2, the length of the area is approximately 640 meters with five main lanes. In addition, there are eight synchronized video cameras recording the absolute positions of the vehicles passing through the study area. The details of the dataset can be found in [61].

FIGURE 6.2: US 101 study area

Algorithm 7: Bayesian-based absolute positions estimation

Input: The velocity, acceleration, and absolute position of the target vehicle at the time t_{i-1} , Detected nearby vehicles N_i at the time t_i , GPS measurements of the target vehicle and nearby vehicles at the time t_{i-1} ;

Output: The absolute position of each detected nearby vehicle;

Determine the absolute position of the target vehicle at the time t_i through the Kalman filter strategy;

for each nearby vehicle $N_i^j \in N_i$ **do**

 Obtain the relative distance and bearing angle between the target vehicle T_i and nearby vehicle N_i^j ;

 Determine the center of sampling area P_{Sam}^{ij} for the nearby vehicle N_i^j as Eq. 6;

 Sample candidate positions of the target vehicle T_i and nearby vehicle N_i^j ;

if The number of the nearby vehicles $num = 1$ **then**

 The absolute position of the nearby vehicle is determined from Eq. 8;

if The number of the nearby vehicles $num > 1$ **then**

 The absolute position of each nearby vehicle is determined from Eq. 18;

6.3.1.2 Accuracy evaluation:

The evaluation metrics to evaluate the localization accuracy are given as follows:

$$PL_i = \frac{\sum_j^{num} |\bar{P}_N^{ij} - P_N^{ij}|}{num} \quad (24)$$

$$PL = \frac{\sum_i^n PL_i}{n} \quad (25)$$

where $\bar{P}_N^{ij}$ and P_N^{ij} are the estimated absolute position and real absolute position of the nearby vehicle N_i^j , and num is the number of the nearby vehicles detected by the target vehicle. Therefore, PL_i refers to the average position error of the detected nearby vehicles at the time t_i . PL refers to the average position error given the trajectory of the target vehicle, where n is number of the absolute positions for evaluation from the trajectory of the target vehicle. Therefore, a smaller value of PL indicates higher localization accuracy.

6.3.1.3 Parameters settings

In this work, the distance thresholds for V2V communications are set to 15 m, 50 m and 80 m respectively. The x-axis and y-axis standard deviations of the GPS measurement are set to 2 m. The standard deviations of the relative distance and bearing angles are set to 2 m and 0.12 respectively. The value of balance factor α is set to 0.6. Different values of α are set from 0.2 to 0.8 in the experiment of sensitivity analysis.

6.3.1.4 Methods for comparison

To evaluate the localization accuracy, the accuracy of our proposed approach is compared with the accuracy of the GPS measurements. Moreover, to investigate the influence of the centers of sampling areas on localization accuracy, we compare the localization accuracy when the centers of sampling areas are determined as the GPS positions of the nearby vehicles, and determined from our proposed strategy respectively. In our experiments, *Bayesian* refers to our proposed approach, where the centers of sampling areas for the nearby vehicles are determined as the GPS positions, and *Bayesian+Kalman* refers to our proposed approach, where the centers of sampling areas for the nearby vehicles are determined in Eq. 6.

6.3.2 Experimental results

In this section, experiments are conducted to evaluate the accuracy of the proposed approach. To investigate the localization accuracy by considering the real driving scenarios with different number of the nearby vehicles, we firstly investigate the

number of the nearby vehicles detected by the target vehicle within the three distance thresholds. Secondly, the localization accuracy given the three distance thresholds is obtained. In addition, we investigate the influence of the centers of sampling areas on the localization accuracy. After that, sensitivity analysis is conducted to investigate the influence of the parameters on the localization accuracy.

6.3.2.1 The number of nearby vehicles

The number of the nearby vehicles detected by the target vehicle given the three different distance thresholds are shown in Fig. 6.3. The three distance thresholds are 15 m, 50 m and 80 m respectively. It can be seen that the number of the nearby vehicles detected by the target vehicle increases with the increase of the distance thresholds. For example, when the distance threshold is 15 m, the number of the nearby vehicles varies between 1 and 3. When the distance threshold is 50 m, the number of the nearby vehicles varies between 1 and 4, and the maximum number of the nearby vehicles reaches 6 when the distance threshold is 80 m.

6.3.2.2 Localization accuracy

The average localization accuracy given the three distance thresholds are shown in Fig. 6.4. In general, the accuracy of our proposed Bayesian+Kalman approach outperforms the GPS measurements for the three distance thresholds. The reason is that the GPS positions and relative positions of vehicles in the Bayesian strategy decrease the GPS measurement errors. In addition, the accuracy of the proposed approach is affected by the number of the nearby vehicles. From Fig. 6.4(a) to Fig. 6.4(c), it can be observed that the localization accuracy increases and becomes less volatile with the increase of the distance thresholds. The reason behind is that when the number of the nearby vehicles is small within a small distance threshold, the measurement errors of relative positions can be significant and volatile for localization. When the number of the nearby vehicles increases within a large distance threshold, the measurement errors can be reduced since the measurement errors can be cancelled out significantly from the Bayesian strategy.

FIGURE 6.3: Number of detected nearby vehicles given three distance thresholds

Table 6.1 lists the average position error and variance of the localization accuracy in terms of GPS measurements and our proposed *Bayesian* and *Bayesian+Kalman* approaches with the three distance thresholds. In general, the results matches the localization results in Fig 6.4. Our proposed Bayesian+Kalman approach outperforms the other two in terms of the PL and variance. The reason is that the centers of sampling areas for the nearby vehicles are closer to their real positions than the their GPS positions with the consideration of the motion of the target vehicle. For example, the accuracy of the Bayesian+Kalman approach is 4.39 m compared with 6.27 m of the GPS measurements, and 5.84 m of the Bayesian approach when the distance threshold is 15 m. When the distance threshold is 80 m, the accuracy of the Bayesian+Kalman approach is 4.21 m compared with the 6.28 m of the GPS measurements, and 5.40 m of the Bayesian approach. For the localization variance, the variances of our proposed Bayesian and Bayesian+Kalman approaches decrease with the increase of the three distance thresholds, while the variance of GPS measurements remains constant. The reason behind is that the influence of the measurement errors on the localization accuracy can be reduced in our Bayesian strategy when the number of the nearby vehicles increases.

FIGURE 6.4: Average position error given three distance thresholds

TABLE 6.1: Average position error and variance

Methods		15 m	50 m	80 m
GPS	PL	6.27	6.31	6.28
	Variance	2.12	2.35	2.02
Bayesian	PL	5.84	5.61	5.40
	Variance	1.63	1.12	1.04
Bayesian + Kalman	PL	4.39	4.24	4.21
	Variance	1.52	0.83	0.61

6.3.2.3 Sensitivity analysis

To investigate the influence of the parameters on the localization accuracy of our proposed Bayesian+Kalman approach, the sensitivity analysis in terms of the balance factor α , the standard deviation of GPS measurement errors σ_g , the standard deviation of the measurement errors of relative distance σ_d and bearing angle σ_a is conducted.

Balance factor α : As the balance factor α determines the centers of sampling areas for the nearby vehicles between the GPS measurements and the inferred

conditional positions the nearby vehicles. To investigate sensitivity of the proposed approach for different values of α , the localization accuracy for different values of α with the three distance thresholds is shown in Fig. 6.5.

In general, the localization accuracy increases with the increase of the value of α . This indicates that when the centers of sampling areas are closer to the inferred conditional positions of the nearby vehicles, the candidate positions are closer to the real positions of the nearby vehicles. As a result, the centers of sampling areas are closer to the real positions of the nearby vehicles. For example, when the value of α is set to 0.2, the centers of sampling areas for the nearby vehicles are dominantly determined from the GPS measurements. The localization accuracy given the three distance thresholds are more than 5 m. When the value of α reaches 0.8, the centers of sampling areas for the nearby vehicles are dominantly determined from the inferred conditional positions. The localization accuracy of the proposed approach given the three distance thresholds are less than 4 m. In addition, the accuracy gap given the three distance thresholds becomes less significant with the increase of the value of the α . The reason is that since the centers of sampling areas are closer to the real positions of the nearby vehicles with a large value of α , therefore, the influence of the number of nearby vehicle on the localization accuracy decreases.

FIGURE 6.5: Average position error with different value of α

Standard deviation σ_g : To investigate the influence of the GPS measurement errors on the localization accuracy, the localization accuracy for the different value of standard deviation σ_g of the GPS measurement errors is shown in Fig. 6.6. It can be seen that the localization accuracy decreases with the increase of the the value of σ_g . When our proposed Bayesian+Kalman approach is compared with the GPS measurements, our proposed approach achieves significantly higher accuracy than

the GPS measurements given the three distance thresholds. For example, when the standard deviation of GPS measurement errors is 4 m, the average GPS measurement errors achieves more than 11 m, whereas the proposed approach achieves approximately 7 m when the distance threshold is 80 m. When the standard deviation of GPS measurement errors is 1 m, our proposed approach enables to achieve approximately 20% accuracy improvements than the GPS measurements given the three distance thresholds. In addition, the accuracy gap given the three distance thresholds increases with the increase of the value of σ_g , the reason is that when the GPS measurement errors are large, the number of the nearby vehicles is able to decrease the measurement errors in the Bayesian strategy.

FIGURE 6.6: Average position error with different value of σ_g

Standard deviation σ_d and σ_a : The sensitivity analysis in terms of the standard deviation σ_d and σ_a are conducted. The localization accuracy for the different values of σ_d and σ_a are shown in Fig. 6.7 and Fig. 6.8 respectively. In Fig. 6.7, the localization accuracy significantly decreases with the increase of the value of σ_d from 0.5 m to 12 m. When the value of σ_d reaches 10 m, our proposed approach achieves comparable localization accuracy with the GPS measurements. For the three distance thresholds, the localization accuracy when the distance threshold is 80 m is slightly higher than the localization accuracy when the distance threshold is 15 m due to reason that the number of the nearby vehicles with the distance threshold of 80 m is larger than that when the distance threshold is 15 m. Similarly, when the value of σ_a is varied from 0.02 to 0.20, the localization accuracy with the three distance thresholds slightly decreases with the increase of the value of σ_a . In addition, there is no significant difference among the localization accuracy with the three distance thresholds.

FIGURE 6.7: Average position error with different value of σ_d

When comparing Fig. 6.7 and Fig 6.8, it can be observed that the localization accuracy is less sensitive to the measurement errors of the bearing angles than the relative distances. Therefore, we could infer that measured relative distances with higher accuracy is more significant to improve the localization accuracy than the bearing angles.

FIGURE 6.8: Average position error with different value of σ_a

6.4 Summary

In this chapter, we propose a Bayesian-based approach to estimate the absolute positions of the nearby vehicles. Unlike the existing localization approaches, our proposed approach can be applied when the geometry information is diverse and the trajectories of the nearby vehicles are insufficient. In addition, we determine the centers of sampling areas for the nearby vehicles to increase the localization accuracy. The experimental results show that our proposed Bayesian-based localization approach achieves higher localization accuracy compared with the GPS

measurements with different number of the nearby vehicles. In addition, the centers of sampling areas determined from our proposed strategy is able to increase the localization accuracy significantly.

Chapter 7

Conclusions and Future Work

This chapter concludes the thesis and discusses several directions for future work.

7.1 Conclusions

In this thesis, map matching for route inference and vehicle localization are studied. For route inference, an online map matching algorithm and a weights estimation approach are proposed. For vehicle localization, vehicle localization approaches to localize the target vehicle and nearby vehicles are proposed.

To minimize inference latency, the opportunistic online map matching algorithm proposed in Chapter 3 is able to produce an immediate inference when a new GPS point becomes available. Furthermore, to ensure the inference accuracy, a rollback mechanism is provided to correct the already inferred route when some abnormal conditions are detected. The map matching accuracy and inference delay of the proposed algorithm are assessed and compared relative to that of three state-of-the-art online map matching algorithms. The results show that the map matching accuracy of our algorithm is better or at least comparable to the compared algorithms. More importantly, our algorithm has much shorter inference latency, which is critical for some real-time ITS applications.

Different from the existing approaches that either provide low generality or suffer from the estimation bias, the distribution-based weights estimation approach proposed in Chapter 4 is able to decrease the estimation bias by considering the route popularity based on city-scale traces. In addition, our proposed approach can be used as an estimation framework and is suitable for both linear and non-linear driving cost models with either normalized or non-normalized travelling features. Experiments are conducted to evaluate the map matching accuracy by applying the estimated weights on the existing map matching algorithms which adopt either linear or non-linear driving cost models with normalized or non-normalized travelling features. The experimental results show that the weights estimated from our proposed approach are able to significantly increase map matching accuracy compared with the existing approaches.

Chapter 5 describes an efficient and error-tolerant approach for vehicle localization. A pruning-based strategy is proposed to narrow the search space for the position of the target vehicle with the consideration of relative positions of nearby vehicles. To determine the position of the target vehicle, candidate positions are sampled from the candidate zones. Then, a displacement-based selection method is proposed to determine the selection scores for these candidate positions. In the proposed displacement-based selection method, the measurement errors of relative positions are largely cancelled out in the calculation of displacement differences. The experimental results show that the search space for the position of the target vehicle is largely reduced even when the measurement errors of relative positions are large. Furthermore, the localization accuracy and efficiency of our proposed approach are assessed and compared relative to that of two state-of-the-art CVL approaches. The experimental results show our proposed approach is able to achieve higher localization efficiency and accuracy relative to the two existing CVL approaches.

Chapter 6 describes a Bayesian-based map matching approach to estimate the absolute positions of the nearby vehicles. Unlike the existing localization approaches, our proposed approach can be applied when the geometry information is diverse and the trajectories of the nearby vehicles are insufficient. In addition, to increase the localization accuracy, we proposed a strategy to determine the centers of sampling areas of the nearby vehicles by leveraging the GPS measurements, the absolute position of the target vehicle, and the measured relative position. The experimental results show that our proposed Bayesian-based localization approach

achieves higher localization accuracy compared with GPS measurements with the different number of the nearby vehicles. In addition, the experimental results show when the centers of sampling areas are determined using our proposed strategy, the localization accuracy can be significantly increased compared with the localization accuracy when the centers of sampling areas are determined simply using GPS positions.

7.2 Future Work

The key directions for future work are summarized in the following subsections.

7.2.1 Map matching for route inference

Research on online map matching is still an ongoing research area, and there are many challenges that need to be tackled. Firstly, for the driving cost models of map matching algorithms, real-time traffic-related features (e.g., real-time travelling time) can be considered as the factors that influence the driving behaviors of drivers. For example, the real-time travelling time on one given road segment can be estimated from the historical travelling time of drivers and real-time traffic conditions. Secondly, given the real-time traffic-related features, designing an online weights estimation strategy is another research direction. The weights could be determined according to the complex traffic conditions with the consideration of real-time features. Lastly, the inferred routes from map matching algorithms can be analysed for ITS applications. Using the application of personalized route recommendation as an example, it is important to understand drivers' driving behaviors towards the traffic conditions of city network by analysing the inferred travelling routes. In this manner, the optimal routes for drivers based on the driving analysis can be recommended.

7.2.2 Map matching for vehicle localization

Research on vehicle localization is active for ITS applications. The localization accuracy and efficiency are two main considerations for vehicle localization. To

further increase localization efficiency, a lane selection model can be proposed by considering the real-time travelling features. For example, travelling features such as drivers' driving velocities on lanes can be estimated from the historical driving velocities and traffic conditions. Then, a lane selection model can be proposed to filter out the impossible lanes for the target vehicle with the consideration of travelling velocities. In this manner, the search space can be further narrowed. To further improve localization accuracy, the information in terms of roads and traffic conditions can be considered in the proposed displacement-based selection method and Bayesian-based localization technique. To localize the target vehicle, the velocities of the target vehicle determined from their historical velocities and real-time traffic conditions can be leveraged to determine the selection scores for the candidate positions of the target vehicle. To localize the nearby vehicles, the velocity limitations of lanes can be used in Bayesian inference to derive the probability of the given lanes being travelled by drivers.

Appendix A

Estimation of the travelling areas of nearby vehicle

To estimate the travelling area of each detected nearby vehicle, the position relations between the real position of the target vehicle and its nearby vehicles are firstly provided. Given an assumed real position $P_T^i = [X(T_i), Y(T_i)]$ of the target vehicle T_i , the real position $P_N^{ij} = [X(N_i^j), Y(N_i^j)]$ of the nearby vehicle N_i^j can be expressed as:

$$\begin{aligned} X(N_i^j) &= X(T_i) + PD(T_i, N_i^j)_X \\ &= X(T_i) + d(T_i, N_i^j) \cos(a(T_i, N_i^j)) \end{aligned} \quad (\text{A.1})$$

$$\begin{aligned} Y(N_i^j) &= Y(T_i) + PD(T_i, N_i^j)_Y \\ &= Y(T_i) + d(T_i, N_i^j) \sin(a(T_i, N_i^j)) \end{aligned} \quad (\text{A.2})$$

where $[PD(T_i, N_i^j)_X, PD(T_i, N_i^j)_Y]$ is the *real position difference* (PD) between the target vehicle T_i and nearby vehicle N_i^j

The real position difference between the target vehicle T_i and nearby vehicle N_i^j can be expressed as:

$$\begin{aligned} PD(T_i, N_i^j)_X &= [\hat{d}(T_i, N_i^j) - \varepsilon_d] \cos(\hat{a}(T_i, N_i^j) - \varepsilon_a) \\ PD(T_i, N_i^j)_Y &= [\hat{d}(T_i, N_i^j) - \varepsilon_d] \sin(\hat{a}(T_i, N_i^j) - \varepsilon_a) \end{aligned} \quad (\text{A.3})$$

where ε_d and ε_a are the measurement error of relative distance and bearing angle, and ε_d and ε_a follow Gaussian distributions with the standard deviations ρ and ω respectively.

Eq. A.3 can be approximately estimated using the 1st Tayler expansion as follows:

$$\begin{aligned} PD(T_i, N_i^j)_X &\approx \hat{d}(T_i, N_i^j) \cos(\hat{a}(T_i, N_i^j)) + \varepsilon_x^{PD} \\ &= \hat{PD}(T_i, N_i^j)_X + \varepsilon_x^{PD} \\ PD(T_i, N_i^j)_Y &\approx \hat{d}(T_i, N_i^j) \sin(\hat{a}(T_i, N_i^j)) + \varepsilon_y^{PD} \\ &= \hat{PD}(T_i, N_i^j)_Y + \varepsilon_y^{PD} \end{aligned} \quad (\text{A.4})$$

where ε_x^{PD} and ε_y^{PD} are the estimated error of the measured position difference in order to offset the measurement errors of relative positions. ε_x^{PD} and ε_y^{PD} are determined as follows:

$$\begin{aligned} \varepsilon_x^{PD} &= -\varepsilon_d \cos(\hat{a}(T_i, N_i^j)) + \varepsilon_a \hat{d}(T_i, N_i^j) \sin(\hat{a}(T_i, N_i^j)) \\ \varepsilon_y^{PD} &= -\varepsilon_d \sin(\hat{a}(T_i, N_i^j)) - \varepsilon_a \hat{d}(T_i, N_i^j) \cos(\hat{a}(T_i, N_i^j)) \end{aligned} \quad (\text{A.5})$$

According to Trigonometric Equations, Eq. A.5 can be expressed as:

$$\begin{aligned} \varepsilon_x^{PD} &= R \cos(\hat{a}(T_i, N_i^j) + \beta) \\ \varepsilon_y^{PD} &= -R \sin(\hat{a}(T_i, N_i^j) + \beta) \end{aligned} \quad (\text{A.6})$$

where

$$\begin{aligned} R &= \sqrt{(-\varepsilon_d)^2 + (\varepsilon_a \hat{d}(T_i, N_i^j))^2} \\ \beta &= \tan^{-1}\left(\frac{\varepsilon_a \hat{d}(T_i, N_i^j)}{\varepsilon_d}\right) \end{aligned} \quad (\text{A.7})$$

Appendix B

Determination of PDF of displacement difference

To determine the probability density function (PDF) of displacement difference in terms of the real position of the target vehicle, the displacement of the target vehicle in terms of the real position is determined as:

$$\text{Vec}(P_T^i \rightarrow g_T^i) = g_T^i - P_T^i \quad (\text{B.1})$$

where g_T^i is the GPS position of the target vehicle T_i and P_T^i is the real position of the target vehicle.

Similarly the displacement of the nearby vehicle N_i^j in terms of real position is determined as follows:

$$\text{Vec}(P_N^{ij} \rightarrow g_N^{ij}) = g_N^{ij} - P_N^{ij} \quad (\text{B.2})$$

where P_N^{ij} and g_N^{ij} are the real position and GPS position of the nearby vehicle N_i^j .

As the accuracy of GPS measurements depends on the quality of the pseudorange measurement between the satellites and GPS receivers with the consideration of common error sources (e.g., atmospheric delay, satellite clock and ephemeris errors, etc.), and uncommon error sources (e.g., the noise of GPS receivers). When vehicles are driving in a vicinity, these vehicles are assumed to have the GNSS data coming

from the same constellation of satellites. However, as non-common errors vary from vehicle to vehicle, the bias in GPS positioning caused by uncommon errors is usually modelled following an vehicle-dependent zero-mean Gaussian distribution.

Therefore, given the target vehicle T_i , the measured GPS position of the target vehicle T_i can be formulated as:

$$g_T^i = P_T^i + B_T^i + NB_T^i \quad (\text{B.3})$$

where P_T^i represents the real position of T_i , B_T^i represents the position bias caused by common errors, and NB_T^i denotes the position bias caused by non-common errors. NB_T^i is modelled following a Gaussian distribution $N(0, \sigma_{NB_T^i}^2)$, where $\sigma_{NB_T^i}$ is the standard deviation of NB_T^i .

Similarly, the measured GPS position of its j^{th} nearby vehicle N_i^j can be written as:

$$g_N^{ij} = P_N^{ij} + B_N^{ij} + NB_N^{ij} \quad (\text{B.4})$$

where P_N^{ij} represents the real position of N_i^j , B_N^{ij} represents the position bias caused by common errors, NB_N^{ij} represents the position bias caused by non-common errors. NB_N^{ij} follows a Gaussian distribution $N(0, \sigma_{NB_N^{ij}}^2)$, where $\sigma_{NB_N^{ij}}$ is the corresponding standard deviation of NB_N^{ij} .

As the bias caused by common error sources $B_T^i \approx B_N^{ij}$ when the target vehicle T_i and nearby vehicle N_i^j in a vicinity. Therefore, by subtracting Eq. B.4 from Eq. B.3, we have:

$$g_T^i - g_N^{ij} \approx P_T^i - P_N^{ij} + NB_T^i - NB_N^{ij} \quad (\text{B.5})$$

where the difference of the GPS position of the target vehicle and nearby vehicle can be determined as the real position difference and the difference of the independent position bias caused by the uncommon errors.

Therefore, the displacement difference DD in terms of the real position of the target vehicle can be computed from Eq. B.1, Eq. B.2 and Eq. B.5 as follows:

$$\begin{aligned}
 DD(T_i, N_i^j, P_T^i) &= \text{Vec}(P_T^i \rightarrow g_T^i) - \text{Vec}(P_N^{ij} \rightarrow g_N^{ij}) \\
 &= (g_T^i - P_T^i) - (g_N^{ij} - P_N^{ij}) \\
 &= (g_T^i - g_N^{ij}) + (P_N^{ij} - P_T^i) \\
 &\approx (NB_T^i - NB_N^{ij})
 \end{aligned} \tag{B.6}$$

Since NB_T^i and NB_N^{ij} are assumed as independent Gaussian distributions $N(0, \sigma_{NB_T^i}^2)$ and $N(0, \sigma_{NB_N^{ij}}^2)$, therefore $(NB_T^i - NB_N^{ij})$ follows a Gaussian distribution $N(0, \sigma_{NB_T^i}^2 + \sigma_{NB_N^{ij}}^2)$, where $\sigma_{NB_T^i}$ and $\sigma_{NB_N^{ij}}$ are the standard deviations of NB_T^i and NB_N^{ij} respectively.

List of Publications

Journal Article

- Linbo Luo, Xiangting Hou, Wentong Cai, and Bin Guo. Incremental route inference from low-sampling GPS data: An opportunistic approach to online map matching. *Information Sciences*, 512:1407-1423,2020.16,83

Conference Proceeding

- Xiangting Hou, Linbo Luo, Wentong Cai, Masatoshi Hanai. Fast Online Map Matching for Recovering Travelling Routes from Low-Sampling GPS Data. *IEEE International Conference on Ubiquitous Intelligence and Computing* (2018). DOI: 10.1109/SmartWorld.2018.00165.

Bibliography

- [1] M. Alam, J. Ferreira, and J. Fonseca, “Introduction to intelligent transportation systems,” in *Intelligent Transportation Systems*. Springer, 2016, pp. 1–17. 1
- [2] A. Aldegheishem, H. Yasmeen, H. Maryam, M. A. Shah, A. Mehmood, N. Alrajeh, and H. Song, “Smart road traffic accidents reduction strategy based on intelligent transportation systems (tars),” *Sensors*, vol. 18, no. 7, p. 1983, 2018. 1
- [3] L. Altoaimy, I. Mahgoub, and M. Rathod, “Weighted localization in vehicular ad hoc networks using vehicle-to-vehicle communication,” in *2014 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2014, pp. 1–5. 4, 24, 25
- [4] J. M. Alvarez, T. Gevers, F. Diego, and A. M. Lopez, “Road geometry classification by adaptive shape models,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 459–468, 2012. 9, 28
- [5] I. Ashokaraj, A. Tsourdos, P. Silson, and B. White, “Sensor based robot localisation and navigation: Using interval analysis and extended kalman filter,” in *2004 5th Asian Control Conference (IEEE Cat. No. 04EX904)*, vol. 2. IEEE, 2004, pp. 1086–1093. 25
- [6] A. Benslimane, “Localization in vehicular ad hoc networks,” in *2005 Systems Communications (ICW’05, ICHSN’05, ICMCS’05, SENET’05)*. IEEE, 2005, pp. 19–25. 24, 25
- [7] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli, “Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 873–878. 27
- [8] G. Bishop, G. Welch *et al.*, “An introduction to the kalman filter,” *Proc of SIGGRAPH, Course*, vol. 8, no. 27599–23175, p. 41, 2001. 99
- [9] 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*. IEEE, 2007, pp. 1–6. 24, 25

- [10] B. Boots, A. Okabe, and K. Sugihara, "Spatial tessellations," *Geographical information systems*, pp. 503–526, 1999. 34
- [11] F. Bounini, D. Gingras, H. Pollart, and D. Gruyer, "Real time cooperative localization for autonomous vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 1186–1191. 24, 25
- [12] E. Cascetta, A. Nuzzolo, F. Russo, and A. Vitetta, "A modified logit route choice model overcoming path overlapping problems. specification and some calibration results for interurban networks," in *Transportation and Traffic Theory. Proceedings of The 13th International Symposium On Transportation And Traffic Theory, Lyon, France, 24-26 July 1996*, 1996. 17
- [13] P. Cramton, R. R. Geddes, A. Ockenfels *et al.*, "Using technology to eliminate traffic congestion," *Journal of Institutional and Theoretical Economics*, vol. 175, no. 1, pp. 126–39, 2019. 1
- [14] F. Cunha, L. Villas, A. Boukerche, G. Maia, A. Viana, R. A. Mini, and A. A. Loureiro, "Data communication in vanets: Protocols, applications and challenges," *Ad Hoc Networks*, vol. 44, pp. 90–103, 2016. 4
- [15] C. Davidson-Pilon *et al.*, "Probabilistic programming & bayesian methods for hackers," 2015. 60
- [16] L. Davis, "Handbook of genetic algorithms," 1991. 20
- [17] F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," *Sensors*, vol. 17, no. 2, p. 271, 2017. 8, 23, 26, 27, 90
- [18] D. Delling, A. V. Goldberg, M. Goldszmidt, J. Krumm, K. Talwar, and R. F. Werneck, "Navigation made personal: Inferring driving preferences from gps traces," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015, p. 31. 1
- [19] N. Deo, A. Ranges, and M. M. Trivedi, "How would surround vehicles move? a unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, 2018. 28
- [20] A. I. Dumitru, T. Girbacia, R. G. Boboc, C.-C. Postelnicu, and G.-L. Morgan, "Effects of smartphone based advanced driver assistance system on distracted driving behavior: A simulator study," *Computers in Human Behavior*, vol. 83, pp. 1–7, 2018. 2
- [21] M. C. Ferris, O. L. Mangasarian, and S. J. Wright, *Linear programming with MATLAB*. SIAM, 2007, vol. 7. 20
- [22] R. Feynman and F. Vernon Jr., "The theory of a general quantum system interacting with a linear dissipative system," *Annals of Physics*, vol. 24, pp. 118–173, 1963. 2

- [23] P. Fritsche, S. Kueppers, G. Briese, and B. Wagner, “Radar and lidar sensorfusion in low visibility environments.” in *ICINCO (2)*, 2016, pp. 30–36. 23
- [24] S. Fujii, A. Fujita, T. Umedu, S. Kaneda, H. Yamaguchi, T. Higashino, and M. Takai, “Cooperative vehicle positioning via v2v communications and on-board sensors,” in *2011 IEEE Vehicular Technology Conference (VTC Fall)*. IEEE, 2011, pp. 1–5. 7, 24, 25
- [25] C. Y. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, “Online map-matching based on hidden markov model for real-time traffic sensing applications,” in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 776–781. 5, 15, 17, 21, 22, 40, 41
- [26] K. Golestan, S. Seifzadeh, M. Kamel, F. Karray, and F. 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*, 2012, pp. 123–130. 2, 7, 8, 23
- [27] K. Golestan, F. Sattar, F. Karray, M. Kamel, and S. Seifzadeh, “Localization in vehicular ad hoc networks using data fusion and v2v communication,” *Computer Communications*, vol. 71, pp. 61–72, 2015. 7, 8
- [28] Y. Han, T. Zhang, and M. Wang, “Holiday travel behavior analysis and empirical study with integrated travel reservation information usage,” *Transportation Research Part A: Policy and Practice*, vol. 134, pp. 130–151, 2020. 2
- [29] J. Hayya, D. Armstrong, and N. Gressis, “A note on the ratio of two normally distributed variables,” *Management Science*, vol. 21, no. 11, pp. 1338–1341, 1975. 60
- [30] T. Higuchi, M. Giordani, A. Zanella, M. Zorzi, and O. Altintas, “Value-anticipating v2v communications for cooperative perception,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1947–1952. 4
- [31] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, “Preserving privacy in gps traces via uncertainty-aware path cloaking,” in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 161–171. 2
- [32] X. Hou, L. Luo, W. Cai, and M. Hanai, “Fast online map matching for recovering travelling routes from low-sampling gps data,” in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 2018, pp. 917–924. 2, 3, 48, 57

- [33] G. Hu, J. Shao, F. Liu, Y. Wang, and H. T. Shen, “If-matching: Towards accurate map-matching with information fusion,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 114–127, 2017. 6, 18, 19, 21, 22
- [34] T. Hunter, P. Abbeel, and A. Bayen, “The path inference filter: model-based low-latency map matching of probe vehicle data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 507–529, 2013. 18, 19, 62
- [35] —, “The path inference filter: model-based low-latency map matching of probe vehicle data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 507–529, 2014. 20, 21, 22
- [36] S. Hussain, M. Faizan, and M. Hayee, “Real-time relative lane and position identification of surrounding vehicles using gps and dsrc based vehicle-to-vehicle communication,” in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7. 8, 27
- [37] G. R. Jagadeesh and T. Srikanthan, “Heuristic optimizations for high-speed low-latency online map matching with probabilistic sequence models,” in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 2565–2570. 21, 22, 40, 41
- [38] —, “Online map-matching of noisy and sparse location data with hidden markov and route choice models,” *Proceedings of the 18th International Conference on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2423–2434, 2017. 6, 15, 17, 20, 21, 22, 40, 64
- [39] G. R. Jagadeesh, “Map matching imprecise trajectories for smart mobility applications,” Ph.D. dissertation, Nanyang Technological University, 2019. 3
- [40] H.-Y. Jeong, H.-H. Nguyen, and A. Bhawiyuga, “Spatiotemporal local-remote sensor fusion (st-lrsf) for cooperative vehicle positioning,” *Sensors*, vol. 18, no. 4, p. 1092, 2018. 23, 24, 25
- [41] F. Jiménez, J. E. Naranjo, J. J. Anaya, F. García, A. Ponz, and J. M. Armingol, “Advanced driver assistance system for road environments to improve safety and efficiency,” *Transportation research procedia*, vol. 14, pp. 2245–2254, 2016. 2
- [42] M. Kubička, A. Cela, P. Moulin, H. Mounier, and S.-I. Niculescu, “Dataset for testing and training of map-matching algorithms,” in *Intelligent Vehicles Symposium*. IEEE, 2015, pp. 1088–1093. 40
- [43] M. Kubička, A. Cela, H. Mounier, and S.-I. Niculescu, “Comparative study and application-oriented classification of vehicular map-matching methods,” *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 150–166, 2018. 1

- [44] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouza-kitis, “A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829–846, 2018. 2, 4
- [45] N. Lagraa, M. B. Yagoubi, S. Benkouider *et al.*, “Localization technique in vanets using clustering (lvc),” *International Journal of Computer Science Issues (IJCSI)*, vol. 7, no. 4, 2010. 24, 25
- [46] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Nègre, “Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety,” *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 4, pp. 4–19, 2011. 27, 28
- [47] C. Laurendeau and M. Barbeau, “Centroid localization of uncooperative nodes in wireless networks using a relative span weighting method,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, p. 6, 2010. 25
- [48] S. Leßmann, M. Meuter, D. Müller, and J. Pauli, “Probabilistic distance estimation for vehicle tracking application in monocular vision,” in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 1199–1204. 8, 9, 27, 28
- [49] B. Liang, T. Wang, S. Li, W. Chen, H. Li, and K. Lei, “Online learning for accurate real-time map matching,” in *Proceedings of the 20th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2016, pp. 67–78. 17, 21, 22, 40, 41
- [50] X. Liu, K. Liu, M. Li, and F. Lu, “A st-crf map-matching method for low-frequency floating car data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1241–1254, 2016. 6, 18, 19, 21, 22, 62
- [51] —, “A ST-CRF map-matching method for low-frequency floating car data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1241–1254, 2017. 20, 21, 22
- [52] Y. Liu and Z. Li, “A novel algorithm of low sampling rate gps trajectories on map-matching,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, pp. 1–5, 2017. 2
- [53] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, “Microscopic traffic simulation using sumo,” in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: <https://elib.dlr.de/124092/> 83
- [54] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, “Map-matching for low-sampling-rate gps trajectories,” in *Proceedings of the 17th*

- ACM IGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 352–361. 2, 3, 15, 21, 22, 85
- [55] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, “Connected vehicles: Solutions and challenges,” *IEEE internet of things journal*, vol. 1, no. 4, pp. 289–299, 2014. 4
- [56] L. Luo, X. Hou, W. Cai, and B. Guo, “Incremental route inference from low-sampling gps data: An opportunistic approach to online map matching,” *Information Sciences*, vol. 512, pp. 1407–1423, 2020. 15, 85
- [57] M. Narasimhan, P. Viola, and M. Shilman, “Online decoding of markov models under latency constraints,” in *Proceedings of the 23rd International Conference on Machine learning*. ACM, 2006, pp. 657–664. 15
- [58] P. Newson and J. Krumm, “Hidden markov map matching through noise and sparseness,” in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 336–343. 6, 14, 42, 55, 57
- [59] OpenStreetMap contributors, “Planet dump retrieved from <https://planet.osm.org>,” <https://www.openstreetmap.org>, 2017. 63, 83
- [60] T. Osogami and R. Raymond, “Map matching with inverse reinforcement learning,” in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013, pp. 2547–2553. 6, 18, 19, 57, 62
- [61] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, “On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (ngsim) program data,” *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1243–1262, 2011. 105
- [62] M. Quddus and S. Washington, “Shortest path and vehicle trajectory aided map-matching for low frequency gps data,” *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 328–339, 2015. 20, 21, 22
- [63] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, “Current map-matching algorithms for transport applications: State-of-the art and future research directions,” *Transportation research part c: Emerging technologies*, vol. 15, no. 5, pp. 312–328, 2007. 5
- [64] K. N. Qureshi and A. H. Abdullah, “Localization-based system challenges in vehicular ad hoc networks: survey,” *SmartCR*, vol. 4, no. 6, pp. 515–528, 2014. 24, 25
- [65] A. Rasheed, S. Gillani, S. Ajmal, and A. Qayyum, “Vehicular ad hoc network (vanet): A survey, challenges, and applications,” in *Vehicular Ad-Hoc Networks for Smart Cities*. Springer, 2017, pp. 39–51. 4

- [66] L. Rivoirard, M. Wahl, P. Sonni, D. Gruyer, and M. Berbineau, “A cooperative vehicle ego-localization application using v2v communications with cbl clustering,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 722–727. 24, 26
- [67] M. Rohani, D. Gingras, and D. Gruyer, “A novel approach for improved vehicular positioning using cooperative map matching and dynamic base station dgps concept,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 230–239, 2015. 2, 7, 9, 21, 24, 26, 81, 84, 85
- [68] M. Rohani, D. Gingras, V. Vigneron, and D. Gruyer, “A new decentralized bayesian approach for cooperative vehicle localization based on fusion of gps and vanet based inter-vehicle distance measurement,” *IEEE Intelligent transportation systems magazine*, vol. 7, no. 2, pp. 85–95, 2015. 7, 8, 9, 21, 24, 26, 82, 85, 90
- [69] A. H. Sakr and G. Bansal, “Cooperative localization via dsrc and multi-sensor multi-target track association,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 66–71. 24, 25
- [70] A. Sarker and H. Shen, “A data-driven misbehavior detection system for connected autonomous vehicles,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, p. 187, 2018. 1
- [71] J. Schlechtriemen, F. Wirthmueller, A. Wedel, G. Breuel, and K.-D. Kuhnert, “When will it change the lane? a probabilistic regression approach for rarely occurring events,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1373–1379. 28
- [72] S. Schönfelder, K. W. Axhausen, N. Antille, and M. Bierlaire, “Exploring the potentials of automatically collected gps data for travel behaviour analysis: A swedish data source,” *Arbeitsberichte Verkehrs-und Raumplanung*, vol. 124, 2002. 13
- [73] M. Schreier, V. Willert, and J. Adamy, “Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 334–341. 27, 28
- [74] M. Shen, J. Sun, H. Peng, and D. Zhao, “Improving localization accuracy in connected vehicle networks using rao–blackwellized particle filters: Theory, simulations, and experiments,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2255–2266, 2018. 21, 82
- [75] H. Slavin, Q. Yang, D. Morgan, A. Rabinowicz, J. Brandon, and R. Balakrishna, “Lane-level vehicle navigation for vehicle routing and traffic management,” may 2018, uS Patent 9,964,414. 2, 7

- [76] A. F. Smith and G. O. Roberts, "Bayesian computation via the gibbs sampler and related markov chain monte carlo methods," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 55, no. 1, pp. 3–23, 1993. 60
- [77] X. Song, Y. Ling, H. Cao, and Z. Huang, "Cooperative vehicle localisation method based on the fusion of gps, inter-vehicle distance, and bearing angle measurements," *IET Intelligent Transport Systems*, vol. 13, no. 4, pp. 644–653, 2018. 7, 8, 24, 26, 84, 90, 99
- [78] C. Su, W. Deng, H. Sun, J. Wu, B. Sun, and S. Yang, "Forward collision avoidance systems considering driver's driving behavior recognized by gaussian mixture model," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 535–540. 8
- [79] S. Taguchi, S. Koide, and T. Yoshimura, "Online map matching with route prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 338–347, 2018. 40
- [80] M. Tahir, S. S. Afzal, M. S. Chughtai, and K. Ali, "On the accuracy of inter-vehicular range measurements using gnss observables in a cooperative framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 682–691, 2018. 2, 21, 81
- [81] Z. Tian, Y. Cai, S. Huang, F. Hu, Y. Li, and M. Cen, "Vehicle tracking system for intelligent and connected vehicle based on radar and v2v fusion," in *2018 Chinese Control And Decision Conference (CCDC)*. IEEE, 2018, pp. 6598–6603. 2, 8, 9, 27, 28
- [82] L. Tierney, "Markov chains for exploring posterior distributions," *the Annals of Statistics*, pp. 1701–1728, 1994. 59
- [83] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan, "The unscented particle filter," in *Advances in neural information processing systems*, 2001, pp. 584–590. 26
- [84] N. R. Velaga, M. A. Quddus, and A. L. Bristow, "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 672–683, 2009. 20, 21, 22
- [85] A. Vinel, N. Lyamin, and P. Isachenkov, "Modeling of v2v communications for c-its safety applications: A cps perspective," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1600–1603, 2018. 4
- [86] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," in *The Foundations Of The Digital Wireless World: Selected Works of AJ Viterbi*. World Scientific, 2010, pp. 41–50. 5, 16

- [87] G. Wang and R. Zimmermann, “Eddy: an error-bounded delay-bounded real-time map matching algorithm using hmm and online viterbi decoder,” in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2014, pp. 33–42. 16
- [88] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8445–8453. 23
- [89] F. Witlox, “Evaluating the reliability of reported distance data in urban travel behaviour analysis,” *Journal of Transport Geography*, vol. 15, no. 3, pp. 172–183, 2007. 13
- [90] H. Woo, M. Sugimoto, J. Wu, Y. Tamura, A. Yamashita, and H. Asama, “Trajectory prediction of surrounding vehicles using lstm network,” in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2018. 8, 9, 27, 28
- [91] Z. Xiao, P. Li, V. Havyarimana, G. M. Hassana, D. Wang, and K. Li, “Goi: A novel design for vehicle positioning and trajectory prediction under urban environments,” *IEEE Sensors Journal*, vol. 18, no. 13, pp. 5586–5594, 2018. 9, 27
- [92] B. Yang, C. Guo, Y. Ma, and C. S. Jensen, “Toward personalized, context-aware routing,” *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 24, no. 2, pp. 297–318, 2015. 6
- [93] Y. Yao, M. Xu, C. Choi, D. J. Crandall, E. M. Atkins, and B. Dariush, “Egocentric vision-based future vehicle localization for intelligent driving assistance systems,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9711–9717. 8
- [94] Y. Yin, R. R. Shah, and R. Zimmermann, “A general feature-based map matching framework with trajectory simplification,” in *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*. ACM, 2016, pp. 1–10. 14, 15, 17, 19, 21, 22, 34, 35
- [95] Y. Yin, R. R. Shah, G. Wang, and R. Zimmermann, “Feature-based map matching for low-sampling-rate gps trajectories,” *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 4, no. 2, p. 4, 2018. 6, 17, 19, 21, 22, 41, 57, 62, 63, 64, 65
- [96] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, “T-drive: driving directions based on taxi trajectories,” in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2010, pp. 99–108. 3

- [97] J. Yuan, Y. Zheng, X. Xie, and G. Sun, “Driving with knowledge from the physical world,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 316–324. 63
- [98] M. Zhan-Moodie, “Spatio-temporal analysis of highway congestion and accidents: A case study of interstate 285 in georgia,” 2019. 1
- [99] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, “Reducing uncertainty of low-sampling-rate trajectories,” in *Proceedings of the 28th International Conference on Data Engineering*. IEEE, 2012, pp. 1144–1155. 2
- [100] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, “Mining interesting locations and travel sequences from gps trajectories,” in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800. 55
- [101] S. Zhu, S. Li, Y. Liu, and Q. Fu, “Low-cost mems-imu/rtk tightly coupled vehicle navigation system with robust lane-level position accuracy,” in *2019 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*. IEEE, 2019, pp. 1–4. 2, 7