Title
Arterial Traffic Activity Estimation

Permalink
https://escholarship.org/uc/item/97g732pd

Author
Yang, Qichi

Publication Date
2013

Peer reviewed|Thesis/dissertation
UNIVERSITY OF CALIFORNIA
RIVERSIDE

Arterial Roadway Traffic Activity Estimation

A Dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Qichi Yang

December 2013

Dissertation Committee:
Dr. Matthew J. Barth, Committee Chairperson
Dr. Jay A. Farrell
Dr. Amit K. Roy-Chowdhury
The Dissertation of Qichi Yang is approved:

__________________________________________

__________________________________________

__________________________________________

Committee Chairperson

University of California, Riverside
Acknowledgement

This dissertation could never have been completed without the help and support from many people. I owe my appreciation to all those who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I would like to give my deepest gratitude to my research advisor, Professor Matthew Barth, for his support and guidance during my Ph.D. career at CE-CERT. In addition to introducing me to many interesting research topics, he is always giving me enough time and room to learn new technologies and try my own ideas in research which led me not only to becoming a well-rounded researcher, but also an independent thinker. With his encouragement, I did my first technical report in a teleconference to my project sponsor. With his recommendation, I got my first internship opportunity in industry. His endless patience and wise advice helped me to overcome the obstacles and desperate situations during these past five years. I am also very appreciate with the time and effort he has spent listening and discussing research problems with me and has given me many suggestions to revise and improve many of presentations and papers including this dissertation.

I would like to give my appreciation to my committees: Prof. Jay Farrell and Prof. Amit Roy-Chowdhury for the time they spent in reading my dissertation and attending my defense. Thanks to Prof. Jay Farrell for his brilliant lectures in engineering parameter estimation class and mentoring me in the FHWA mapping project. Thanks go to Prof. Amit Roy-Chowdhury for introducing me the research
problems in Computer Vision, which remains one of the key areas of my research interest.

I would give my sincere thanks to my current and former colleagues at CE-CERT including Mr. Michael Todd, Dr. Guoyuan Wu, Dr. Kanok Boriboonsomsin, Mr. Alexander Vu, Mr. Daniel Sandez, Dr. Meng Cao, Dr. Weihua Zhu, Dr. Lili Huang, Dr. Anh Vu, and Ms. Kathy Vang. Without your help, many of my research projects would not have been completed. Thanks to Mr. Michael Todd for appearing in all of my research projects to get me out of many practical troubles. Thanks to Dr. Lili Huang for sharing with me her research experience which confirmed my determination to accomplish the degree in my early PhD career. Thanks to Dr. Anh Vu for introducing me to a variety of practical Computer Vision techniques. Thanks to Dr. Guoyuan Wu for many suggestive discussions and arguments we had on ITS research and sorry for postponing him several times to pick up his daughter because of the discussion. Thanks to Dr. Kanok Boriboonsomsin for his invitation to the intersection delay project, allowing for me to apply my travel time model in a real world large-scale dataset. Thanks to Dr. George Scora for his excellent work on the CMEM model, making my energy/emission research more convincing. Thanks to Mr. Daniel Sandez for the cooperation on the parking lot monitoring system which is the most beautiful monitoring system I ever seen in a parking lot.

I would express my thanks to all my friends in lab EBU II 369. Thanks to Dr. Anning Chen for her help in my graduate classes in first two years. Thanks to Dongfang Zhang, Mingyang Li, Yiming Chen for their discussions on various interesting research problems. Thanks to Dr Guowu Xie in Department of Computer
Science for introducing me my favourite programming language Python which makes data processing an enjoyable thing in my research.

I have received great help from Sensys Networks Inc. for the technical and financial support on my research and providing me the first internship opportunity in my life. Thanks to my mentor Dr. Christopher Flores for coordinating my work with other engineers and illustrating to me the relationship between development and research in a smart grid table.

Finally, and the most importantly, I would like to deeply thank my parents whose support, encouragement, patience and unwavering love are always the most valuable gifts at every stage of my personal and academic life. No matter what happened, they always stand behind me and offer their faith in me. In addition, I give special thanks to my girlfriend Dr. Fan Zhang whose understanding and company mean a lot to me. She is always the first person to encourage me in every possible way during the past four years. I will cherish the great time we had together and your continuous love and support.
ABSTRACT OF THE DISSERTATION

Arterial Roadway Traffic Activity Estimation

by

Qichi Yang

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2013
Dr. Matthew J. Barth, Chairperson

With advances in sensing technologies along with innovative modelling and estimation methods, a variety of Intelligent Transportation System (ITS) technology and applications have been developed to mitigate traffic congestion and associated environment pollution problems. In the last decade, several advanced sensing technologies have been developed for dedicated traffic measurements. In this dissertation, we focus on traffic activity estimation techniques and algorithms for three types of advanced traffic sensing systems: 1) sparse mobile sensors for arterial roadway travel time estimation; 2) wireless magnetic sensors for arterial roadway energy/emission estimation; and 3) 3-D LiDAR for lane-level vehicle trajectory estimation.

Due to the interrupted traffic flow caused by traffic control devices, it is very challenging to estimate average travel time of traffic flow along a signalized arterial corridor using conventional inductive loop detectors (ILD). Vehicle position samples from rapidly-growing smart phones and commercial navigator technologies turn out to be another promising data source for this task. However, one of the major obstacles
of using these technologies is the randomness of sampling location, which results in significant variation in measured distance between two consecutive samples, compared to the stationary infrastructure technology. In Chapter 3, we describe a novel probabilistic travel time model that has been developed to deal with this issue by decomposing the arterial travel time into two components: free-flow travel time and delayed time. Validated by field operational tests, the proposed model has exhibited a good fit on the travel time distribution under different congestion levels and has resulted in more reliable and robust vehicle’s activity classification to differentiate between stopped and free-flow maneuvers by each individual vehicle. With this benefit, we developed an unique arterial roadway energy/emission estimation approach that is described in Chapter 4, using wireless magnetic sensors which measure travel time directly for each re-identified vehicle. An approximated speed trajectory is then reconstructed for stopped vehicles and fed into a microscopic energy/emissions model to achieve more accurate energy/emissions estimation compared to today’s commonly used techniques.

Lane-level second-by-second vehicle trajectories are another important data source, which is particularly useful for traffic simulation models in order to calibrate their internal vehicle activity parameters. In Chapter 5, we present a novel mobile sensor platform consisting of a centimeter-level accurate positioning system and a 3-D LiDAR for detecting and extracting surrounding vehicle trajectories. A robust detection/tracking algorithm has been developed to extract a large number of trajectories from vehicles surrounding the sensor-equipped probe vehicle. Results from both freeway and arterial roadway types have shown great potential of such
innovative sensing systems in building high quality trajectory repositories for future research.
Table of Contents

Acknowledgement ........................................................................................................ iv

ABSTRACT OF DISSERTATION ........................................................................ v i

Table of Contents ................................................................................................... x

Table of Figures ..................................................................................................... xiii

Chapter 1. Introduction ........................................................................................... 1

1.1 Problem Statement ........................................................................................... 1

Arterial Travel Time Estimation/Modelling .......................................................... 2
Energy and Emissions Estimation ........................................................................ 8
Individual Vehicle Microscopic Behaviour ......................................................... 12

1.2 Contribution ..................................................................................................... 17

1.3 Organization ...................................................................................................... 18

Chapter 2. Research Background ......................................................................... 20

2.1 Arterial Travel Time Estimation ...................................................................... 20

2.1.1 Model-Based Approaches ...................................................................... 21

2.1.2 Data-Driven Approaches ...................................................................... 25

2.1.3 Hybrid Approaches .............................................................................. 27

2.2.4 Travel Time Distribution Modelling ...................................................... 28

2.2 Lane-Level Vehicle Trajectory Collection using Mobile Sensors ................. 30

2.2.1 Positioning and Mapping ...................................................................... 30

2.2.2 Detection and Tracking Moving Objects .............................................. 33

Chapter 3. Arterial Roadway Travel Time Decomposition and Modelling .......... 38

3.1 Overview .......................................................................................................... 38

3.2 Methodology .................................................................................................... 40

3.2.1 Travel Time Decomposition .................................................................. 40

3.2.2 Modelling Approach ............................................................................. 43

3.3 Experiments .................................................................................................... 49

3.3.1 Experimental Data ................................................................................. 49

3.3.2 Model’s Goodness-of-Fit Evaluation .................................................. 50

3.3.3 Vehicle Stop/Non-Stop Movement Classification ................................ 54

3.4 Summary .......................................................................................................... 58

Chapter 4. Arterial Roadway Energy/Emissions Estimation using Trajectory

Reconstruction ......................................................................................................... 59

4.1 Introduction ...................................................................................................... 59
Chapter 6 Conclusions and Future Work .................................................... 121
6.1 Conclusions .......................................................................................... 122
6.2 Future Work ........................................................................................ 123
    6.2.1 Arterial Route Travel Time Estimation ............................................ 124
    6.2.2 Apply Travel Time Model to Large-Scale Road Network ................. 124
    6.2.3 Vehicle Classification using Wireless Magnetic Sensors .................. 124
    6.2.4 Detection and Tracking using 3D LiDAR Data ............................... 125
    6.2.5 Lane-Level Map for Trajectory Collection ..................................... 126
Bibliography ............................................................................................... 127
Table of Figures

Fig. 1. 1 Travel Time Distribution (TTD)................................. 5
Fig. 1. 2 Average speed vs CO₂ on freeway and signalized arterial link........ 10
Fig. 1. 3 System Diagram for Arterial Energy/Emission Estimation........ 12
Fig. 1. 4 Trajectory Collection in NGSIM project..................... 15
Fig. 1. 5 System diagram of trajectory collection system................ 17
Fig. 3. 1 A typical arterial link with a signalized intersection........... 40
Fig. 3. 2 Travel time histogram 0am-5am over one month.................... 47
Fig. 3. 3 TTD estimate under different congestion level.................. 52
Fig. 3. 4 p-value from KS-test.............................................. 53
Fig. 3. 5 Travel time histogram 11AM-12PM over one month.............. 55
Fig. 3. 6 TTD estimate under different congestion level.................. 56
Fig. 3. 7 Correct rate of classification of GMM with 2-5 components... 57
Fig. 4. 1 Chula Vista Wireless Sensor Network.......................... 64
Fig. 4. 2 Raw data from a travel time detection site...................... 64
Fig. 4. 3 Travel time histogram with ground truth...................... 65
Fig. 4. 4 Platoon plot in pre-processing for trajectory reconstruction.... 71
Fig. 4. 5 Typical trajectory model for a vehicle stopped by a traffic light 72
Fig. 4. 6 Travel Time Histogram with Ground Distribution and mGMM.... 77
Fig. 4. 7 Example free flow vehicle trajectory........................ 79
Fig. 4. 8 Example delayed vehicle trajectory............................ 80
Fig. 4. 9 Total energy/emissions estimates for all trajectories for passenger vehicle.. 81
Fig. 5. 1 Two alternative approaches for detection and tracking for moving objects... 84
Fig. 5. 2 System Diagram.................................................. 85
Fig. 5. 3 Sensor Platform.................................................. 87
Fig. 5. 4 System sampling timing........................................ 90
Fig. 5. 5 Virtual table holds the 3D LiDAR measurements................ 91
Fig. 5. 6 Raw LiDAR measurements with virtual table.................. 92
Fig. 5. 7 Virtual table and virtual scan................................ 92
Fig. 5. 8 Ground point detection....................................... 95
Fig. 5. 9 Ground plane filtering....................................... 96
Fig. 5. 10 Point cloud segmentation.................................... 98
Fig. 5. 11 Illustration of a two-layer grid map........................ 99
Fig. 5. 12 Rectangular vehicle model.................................. 101
Fig. 5. 13 Illustration of motion evidence............................ 104
Fig. 5. 14 Motion evidence.............................................. 104
Fig. 5. 15 Line extraction................................................ 106
Fig. 5. 16 Motion detection.............................................. 108
Fig. 5. 17 Vehicle initialization....................................... 108
Fig. 5. 18 Vehicle geometric model................................... 110
Fig. 5. 19 Vehicle track in 3D view.................................... 116
Fig. 5. 20 Vehicle tracking at intersection........................... 118
Fig. 5. 21 Vehicle tracking during turning…………………………………………………………...119
Chapter 1. Introduction

1.1 Problem Statement

In recent years, we have become increasingly dependent on our transportation system to move both people and goods; however, since the transportation system capacity hasn’t increased significantly and travel demand continues to increase, traffic congestion has become a world-wide problem. It has been shown that traffic congestion wastes time, energy, and results in increased pollution. To solve this traffic congestion problem, Intelligent Transportation System (ITS) technology can be applied to improve overall system operations without building additional roadways. As part of the ITS solution, measuring, estimating and modelling traffic activity is an important component and has been investigated for decades. In last decade, with advances of ITS technologies and the large investment on the sensor deployment over the road network, more comprehensive traffic activity information is now available to be analyzed and evaluated, as well as used to estimate traffic state. This in turn will eventually contribute to better solutions to reduce congestion and environmental problems.

However, most of the modern traffic management systems and traveller guidance systems highly rely on a large number of sensor data which have their limitations. Due to these limitations, the approach of estimating the same traffic activity can vary based on the availability of sensor data in a certain area and time period. Therefore, different kinds of traffic activity have to be studied and integrated with traffic models, thereby requiring various sensing technologies that can be
adapted to different traffic networks. In this dissertation, we consider two kinds of traffic activity categories, at two levels: macroscopic and microscopic, which correspond respectively to aggregated traffic state and to highly detailed states of individual vehicles.

**Arterial Travel Time Estimation/Modelling**

From a macroscopic view, the traffic is considered as a consistent flow mostly ignoring the variation of individual vehicles in this flow. Speed, flow, density are basic macroscopic state parameters that describe flow within the famous fundamental diagram, which plays an important role in traffic analysis and forecasting for long periods, particularly for uninterrupted flow conditions (e.g., the case of freeways). However, in the case of interrupted flow conditions (e.g., the case of arterial roads), traffic flow requires different parameterization. This is because: 1) there is a higher degree of randomness of capacity supply and demand that is widely spread out on the urban road network; and 2) the flow is interrupted by control devices (e.g., signal lights, stop signs), pedestrians, and other events. These reasons make the basic speed, flow, density less useful compared to freeway analysis, and can sometimes even be misleading. To explain this further, we can consider having speed samples from a number of vehicles on one segment of freeway for a certain time period. Every individual vehicle in this uninterrupted traffic flow tends to follow the flow speed. From this we can estimate with high level of confidence the congestion level of traffic, in terms of speed, density and flow. But the same approach cannot be applied on arterial roads, for example because the slow speeds or even interrupted motion may be part of light traffic conditions when we sample vehicles that are waiting in a queue in front of a traffic light or stop sign. The formation and dispersion of these queues
makes speed and density on a single arterial link keep highly variable from the upstream of the link to the signal controlled intersection. Unless we can collect the basic parameters (speed, flow, density) in a very high spatial resolution, a reliable estimation of traffic conditions cannot be made. Hence other macroscopic parameters must be used, such as travel time, intersection delay, vehicle stop rate, and intersection saturation rate; all that are getting more attention from researchers and data collection systems serving arterial traffic applications. Among these parameters, *travel time* is one of the most used, primarily since it establishes a common perception between individual travellers and practitioners. More importantly, travel time is a range measurement that covers the whole link but only needs two spot measurements to be determined. Considering again the previous example, aggregated travel time over a well-defined range on a roadway link that includes signalized intersection can best represent current traffic conditions on that link that is much more realistic. Further, travel time can be applied to multiple distinct traffic applications. For example in Advanced Traveller Systems (ATS), travel time along a link or route (combination of links), aggregated over certain time periods, can be used as an intrinsic property of the links to serve for the traveller to make better route choice; While as a real-time metric, the average travel time at a specific intersection is a good parameter for an Adaptive Traffic Signal Controller (ATSC) to optimize. Despite the importance of travel time in arterial traffic estimation, its variability and measurement difficulty are two big problems in terms of the reliability and accuracy in practical use. Therefore in Chapter 3, our proposed estimation and modelling approach is intended to help solve these two problems.
The travel time variability represents the variation of various travel times over a specific path which increases the uncertainty of travel time estimation. Travel time variability can be investigated from multiple point of views [4]: 1) vehicle-to-vehicle variability which addresses the variance of travel time of different vehicles traveling the same route at the same time period; 2) period-to-period variability which corresponds to the vehicles traveling at the same route but at different time periods within a day; and 3) day-to-day variability corresponding to the trips on the same route in multiple days. Due to this variability, a deterministic estimation of travel time is not enough, a probabilistic travel time distribution (TTD) is a more comprehensive and reliable representation. In some previous research, TTD on a freeway is modelled as a single mode distribution while the TTD for arterial roads has been shown to be a multi-mode distribution that requires better modelling, which to date has not been fully investigated. As an example of this, refer to Figure 1.1. The single-state model such as Exponential or Gaussian distribution can fit the travel time data on freeway well [5] while the travel time on signalized arterial link shows an obvious multi-mode distribution. In Chapter 3, we propose a novel TTD model called a modified Gaussian Mixture Model (mGMM) which contains period-to-period variability and can represent the multi-modal properties.

In order to estimate the TTD, a considerable amount of travel time measurements is required. Measuring travel time with high accuracy over large areas with low cost is non-trivial. Advances in traffic monitoring techniques now offer a variety ways of arterial travel time collection but none of them can offer both wide coverage and high accuracy. We consider sensors of two types: fixed-location sensors that are installed on the road or roadside; and mobile sensors which are on-board
vehicles moving in the road network. The fixed sensors measure the travel time between two travel spots based on re-identification of vehicles. Two example sensors in this category are high-rate loop detectors and wireless magnetic sensors. The loop detector with detecting circuits operating with a high sampling rate has re-

Fig. 1. Travel Time Distribution (TTD) on freeway link (a) and signalized arterial link (b).
identification ability based on the electromagnetic signature of the vehicles passing over the sensor to provide reliable travel time measurement [6]. The wireless magnetic sensors apply the same measurement principle based on the magnetic signature of vehicles. A 70% vehicle matching rate can be achieved in the current deployment in real world traffic monitoring system [7]. Both of these typically are installed in the pavement which means a high cost for installation and maintenance. Radio Frequency Identification (RFID) and Bluetooth are two similar alternative fixed sensors which can also re-identify vehicles to measure travel time [8]. These sensors are usually installed along the roadside and travel time can be measured between any pair of sensors. However, the accuracy of position of detected vehicles is much lower when the RF signal or Bluetooth signal is stronger, which prevents it from providing accurate travel time measurement over a short link. The common limit of these fixed sensor with re-identification function is the system deployment coverage in urban network, and only a few sites contain these new technologies and there seems to be limited growth in the next few years.

The travel time measurement from fixed spot sensors always cover the same travelling length which is between the positions of the sensors. In the mobile sensor scenario, probe vehicles equipped with Global Positioning System (GPS) can record second-by-second position information on which travel time between any two spots can be calculated. This could be the best sensor for travel time measurement but the operation cost (information transmission cost) of a large number of probe vehicles is a big concern. Due to the rapid proliferation of GPS receivers and smart phone users, data streaming back from mobile devices have become a promising source of travel time information. For example companies like Garmin, INRIX, Microsoft, Google,
Apple, Nokia or Waze all use such data [9]. This fast growing data source can provide extensive coverage spatially and temporally, although there are also challenges due to their sparseness. The travel time from this type of sensor is measured between any two random sampled locations every 20 to 60 seconds. In order to utilize the travel time measurements from different data sources, we have developed a model that is designed to be a comprehensive model which can handle both fixed location travel time and sparse travel time samples. This model is described in detail in Chapter 3.

As described above, a good travel time estimation approach on signalized arterial roads should give consideration to the variability of travel time even on the same link at same time period caused by various stop level of vehicles and the right data source that can provide wide range and lasting measurements. The variability is mostly caused by the different stop modes of vehicles passing over a signalized link. To handle the variability problem, travel time was modelled as a Gaussian Mixture Model (GMM) in several previous research efforts since it provides flexibility on fitting multi-mode probabilistic distributions with a compact representation. While the traditional GMM has a great flexibility to fit on multi-mode distributions with a compact representation, it contains no restrictions on each components it carries. However, there are some relevance between different travel time modes. This flaw prevents the original GMM from providing final components that can be well interpreted. Our developed model modifies the original GMM by adding additional constraints. We found the travel time generated by different operation modes should contain a common free flow component. So we decompose the travel time into free flow component and delay component and add the restriction into our model to provide a more stable fitting results with easy inference to free flow and delay.
components in the travel time distribution. Our free flow component is modelled with a Gaussian free flow pace and a travelling distance which makes it available to fit on the data collected by sparse mobile sensors each sample of which contains a different distance. Due to the lack of large scale mobile data, our approach is first validated on a well-calibrated simulation under different volume-capacity (V/C) conditions generating simulated mobile data. Then we test the model on real field test data where a new generation fixed spot sensor is installed to record accurate travel time. We also generate virtual mobile data based on that dataset to verify our model. The proposed model passed all the experiments with very promising results.

Energy and Emissions Estimation

While the macroscopic traffic activities provide a useful view of traffic state for management and travellers, microscopic activities on individual vehicles can make distinct contributions in the traffic scenarios where variability of each individual vehicle cannot be ignored. Energy and emission estimation is one of the examples, and is surprisingly difficult to estimate for arterial roads. As fuel costs continue to rise, combined with increased concern on both greenhouse gas and pollutant emissions, there is a strong need to better estimate overall fuel economy and emissions for roadway traffic in general. To date, there has been a significant amount of traffic energy/emissions research carried out for roadways with uninterrupted flow, namely freeways and highways without traffic signals. As an example, one approach is to use the readily available regulatory emission models (i.e. the U.S. EPA’s MOVES model [10] and California’s EMFAC model [11]) to estimate energy/emission factors based on roadway link speeds. This link speed input can be refined by taking into account various roadway and traffic characteristics (e.g., speed
limit, curvature, access density, etc.), which results in emission factor output that better represents the traffic condition on the link [12]. For uninterrupted flow, energy and emissions can generally be estimated as a function of overall traffic speed, density, and flow for each roadway link, and then aggregated together. In contrast, estimating energy and emissions for roadways with interrupted flow (i.e., arterial roads that have traffic signals, stop signs, etc.) is much more difficult and cannot be simply determined by average traffic speed alone. There are several cases where traffic speed along a corridor can be the same for different scenarios, and yet their energy consumption and emissions can be drastically different, based on various vehicle modal operations (i.e., acceleration, deceleration and idling). The best solution is to estimate emissions at high time resolution (e.g., second-by-second) so that the effect of different modal operations can be captured. To estimate total vehicle emissions at this level of detail, velocity profiles of every vehicles on the roadway are required. Although the estimation results would be highly accurate, such input data are very difficult to acquire in the real world. In the current sensing technology set, vision sensors like surveillance camera and GPS on probe vehicles can provide such high spatial resolution measurements. The vision sensors requires robust detection and tracking algorithms to capture vehicle trajectories which only exist in state-of-art and their application in real world is very limited. The dense GPS data is limited on coverage as well since it is not available from user devices neither since the limited communication bandwidth and privacy. Therefore, the total microscopic approach has so far been restricted to be used in conjunction with microscopic traffic simulation model, which can generate vehicle velocity profiles needed for this approach of emission estimation.
To improve the arterial roadway energy/emission estimation in current sensing technologies, a method needs to be developed to connect the current observable traffic states to the vehicle modal operations that contains further information for the energy/emission estimate. As part of this dissertation, we have developed an approach to improve energy/emission estimates of roadways that builds upon a technique recently proposed for freeways. It was first proposed to create link-based emission factors as a function of link speed vehicle specific power [13]. Barth and Boriboonsomsin [14] extended this approach that combines a microscopic energy/emissions model with a large vehicle activity database to create functional relationships between link-based emission factors and a set of link-based explanatory variables such as speed, density, and road grade, as shown in Figure 1.2 (a). The black curve indicates the approximated relationship between CO$_2$ and average speed of vehicles aggregated within 30 seconds around the speed detector. If the same data can
be collected for the relationship between average speed of vehicles passing through a signalized link and the CO₂ emission being generated, a similar curve can be expected for the high average speed area since it indicates no interruption on the vehicle trajectory, so the energy/emission generated should be close to the vehicles on highway with same level average speed. In contrast, in the low average speed area, the relationship can hold due to a big variation of CO₂. This is because the vehicles delayed by the traffic control and queue at the intersection have significant modal operation comparing with those passing without delay. It seems hard to conclude a consistent functional relationship over the whole average speed range. However, it hints that if we can differentiate the vehicle with stop and non-stop, different speed to energy/emission relationship can be built on these two types.

To help solve this issue for arterial roads, we apply our developed travel time model which has easy inference for free flow (non-stop) distribution and delay (stop) distribution, so it can be used as a vehicle stop/non-stop or movement classifier. If enough travel time can be collected on the link to calibrate this model, it can differentiate those two types of behaviours based on travel time. The energy/emission generated by vehicles without stop can be estimated as the same way as [14], while a modal-based trajectory reconstruction approach is proposed in Chapter 4 to deal with a stopped vehicles for emission/energy estimation, linking macroscopic travel time to the microscopic vehicle activities. This method generates approximate vehicle trajectories on arterial links with traffic lights by explicitly modelling driving modes such as steady-state cruise, acceleration, deceleration, and idle. Then, these approximated trajectories are run through a microscopic energy/emissions model to estimate overall energy/emission impacts. This method has been tested on a real-
world test-bed arterial corridor. It is shown to estimate energy/emissions well within 10% of actual values, compared to standard speed-emission curve techniques that are only within 40% of the actual values. A system overview is shown in Figure 1.3.

**Individual Vehicle Microscopic Behaviour**

The most detailed microscopic traffic activity is a second-by-second vehicle state trajectory, where the state can include position, pose, and velocity. Although having the trajectory of every individual vehicle during all the time over the whole road network can let us know almost everything we want to know about traffic, under current sensing technology and reasonable cost, it is not practically achievable. However, real trajectories sampled from running vehicles over a certain time period or within a limited area are still very helpful. A single trajectory of probe vehicle following the traffic flow on the same link records the driver behaviours of the real-
time traffic. If more probe vehicles exist at the same arterial roads in the same signal cycle, queue dynamic within this cycle can be recovered approximately [15]. A perception sensor (LiDAR, camera) equipped probe vehicle can collect the trajectories of surrounding vehicles, which opens a window for studying the interactive driving behaviours among neighbouring vehicles. A larger set of trajectories can be collected in a fixed area where high resolution surveillance cameras are installed. Though the field of view of a single camera is limited, a group of cameras along the road can cover area of the monitored link with overlap. With robust vehicle tracking algorithms and data association over different cameras, more trajectories can be collected. These learned behaviours can be used in driving behaviour research to develop better driver assistant systems in the car, or applied in emission and energy estimate to evaluate the average modal trajectories in the studied situation. Microscopic traffic simulation might be the research area that most desires real vehicle trajectories where all kinds of individual or interactive vehicle behaviour models are needed for calibrated.

Despite the great value of vehicle trajectories, the collection of a large number of high quality trajectory data is not easily achieved due to the limit of sensing technologies. For surveillance cameras, the difficulty of trajectories collection depends on the installation. Most of cameras installed in a road networks often suffer with a low angle view that can hardly track the vehicles for a long distance after it is occluded by other objects. A high angle camera group can provide high quality trajectory collection if cameras are installed at enough height to avoid occlusion in the view and results from multiple cameras can be merged correctly [1]. However, it costs a lot on installation and data processing and the high installation spots, as shown
in Figure 1.3 (a) (b), are not easy to be found in most areas of road network. The NGSIM project [16] applied this technology to build several lane-level vehicle trajectory datasets that have been used as ground truth or model calibration data in a large number of research efforts. However, due to the difficulty of sensor setup, these datasets are from very few locations and last for only 15mins for each. This largely reduces the generalization of models or research based or validated on these datasets. In order to let traffic researchers collect real trajectories in more locations and for longer time, we developed a mobile sensing platform that can collect with the same level of accuracy (lane level, as shown in Figure 1.3 (c)) and resolution (second-by-second) of vehicle trajectories in the real traffic.

Comparing with the fixed vision sensors, a mobile sensing platform has more flexibility to collect trajectories over a wide range in the road network continuously. The probe vehicle only records ego state which cannot provide enough efficiency for trajectory collection, but perception sensors on a mobile platform can do this job much better. One big problem of mobile sensor platform is the motion of the sensor itself, ego-motion. The lack of high accuracy positioning technology inspires a large number of trials on the detection, tracking and localization research area. With a lower positioning accuracy, the system usually integrate moving object detection and mapping with the localization. Although some successful results have been reported especially in the autonomous driving challenges, it is still a hard problem in congested traffic scenarios since more surrounding vehicles can block the view of perception sensors to the environments which can affect the result of localization. Benefits from better positioning technologies, the lane level positioning and mapping is now
available on our platform, developed by Anh Vu and others [3] which makes the
detection and tracking easier.

In Chapter 5, we describe our mobile trajectory collection platform equipped
with high accuracy positioning device and high resolution range sensors 3D LiDAR.
The system consists of three components: ego-vehicle positioning, detection and
tracking, and a map. We aim to produce a lane level second-by-second trajectory

![Fig.1. 4 Trajectory Collection in NGSIM project. (a) (b) camera setup (c) lane level trajectory extracted by tracking [1]](image-url)
dataset with the same quality as that in NGSIM data warehouse collected by high-angle camera networks which is limited in one location. This requires our positioning and mapping should also achieve a lane-level accuracy. Previous researchers in our group have done the positioning work with a framework propagated by INS and aided by GPS and other feature perception sensors [3]. The work in this dissertation focuses only on detection and tracking using 3D LiDAR which, to the best of our knowledge, has never been applied for vehicle trajectory collection. Detection and tracking of surrounding vehicles by 2D and 3D LiDAR has been studied for over a decade. The main framework is based on feature detection, data association and track updating. It suffers from partially occlusion that commonly occurs in real world traffic and usually requires complicated and computationally complex data association algorithms to solve object merging and splitting. Recently, a model-based no-data association particle filter approach was invented and first applied on autonomous driving which provides more robust and consistent tracking for partially occluded objects. However, the detection step in their approach, served for fast moving object perception, satisfies the accuracy of recognizing vehicles. The false positive objects should be avoided in the large scale trajectory collection task as much as possible to keep the results clean. Utilizing accurate estimation of ego-vehicle state, our proposed method verifies and initializes the vehicle candidates with a high criteria which might lose vehicle trajectories for several frames though provides a higher quality of trajectories for further research applications. The system diagram is shown in Figure 1.5. The lane-level positioning technique is introduced in our previous work [3]. The trajectories collected from tracking module are stored in Earth-Center Earth-Fixed (ECEF) coordinates and registered (matched) to the corresponding lanes in the lane.
map. A lane map technique developed in our group is documented in [17] but has not been used in this work. Our work in Chapter 5 focuses on Vehicle Detection and Tracking module (yellow).

1.2 Contribution

There are several contributions in the dissertation in the area of ITS:
• A new arterial travel time model has been developed to estimate travel time distribution under different traffic conditions without losing the compact parametric representation. The model passes the validation on both simulation data and field test where the results reflect the proposed theory.

• The travel time model mentioned above also provides accurate classification for stop/non-stop of each vehicle with travel time measurement. Because a large portion of traffic energy and emissions are related to the stop-and-go traffic operation on arterial roadways, the classification leads to an improved emissions and fuel estimation approach which takes more detailed vehicle operation mode and shows a significant advance to these previous more macroscopic methods.

• A new mobile sensing platform is developed to collect high quality second-by-second vehicle state trajectories including position, pose and velocity. This mobile platform is equipped on a vehicle with a suite of sensors comprised of vision, range, inertial and GPS. The lane-level accuracy positioning system has been developed as part of a larger group effort. With this level of ego-positioning accuracy, an improved detection method is also developed to incorporate with geometric-based tracking method using 3D LiDAR to provide the same level of accuracy on trajectories of tracked vehicles. 3D LiDAR as a range sensor is first used for the vehicle trajectory collection in ITS and shows its great capability. Details of 3D LiDAR pre-processing are introduced as well. Complete trajectory extraction software has been developed to serve further research.

1.3 Organization
The dissertation is organized in the following way. Related research and background are provided in Chapter 2. Chapter 3 describes the arterial travel time modelling and Chapter 4 follows up with description of stop/non-stop classification incorporated with energy/emission estimation as an application of the travel time model. A trajectory collection system in challenging high quality vehicle state estimation is addressed in Chapter 5. Finally, the concluding remarks and future work are given in Chapter 6 which ties together the different components of the dissertation and provides insights into future directions for this work.
Chapter 2. Research Background

2.1 Arterial Travel Time Estimation

In previous studies, arterial (urban) travel time estimation has received a good deal of interest, with a major focus on the modelling approach. There are two general approaches, one a model-based method, the other a data-driven method. In recent work, hybrid approaches have also been taken, which combine the model- and data-driven methods. The model-based approach models the traffic flow process (at the macroscopic level) and has been inspired by hydrodynamics or queue theory, or explicitly models individual vehicle behaviour in the simulation to estimate the traffic state in the urban network and later the travel time on any links or routes. These approaches typically require a large set of parameters to be calibrated and depend on certain mathematical assumptions to hold between parameters. Although these models are stated to be generically applicable (in theory), when applying to different traffic scenarios and networks, they tend to lack flexibility resulting in decreased estimation and prediction accuracy. In contrast, the pure data-drive approach ignores the physical model assumptions and only focuses on the measurement data from sensor networks. When applying these methods to travel time estimation, relationships are developed between specific measurements of traffic parameters (e.g. speed, flow) and travel time based on existing statistical or artificial intelligence models. The flexibility of these pure data-driven approaches are their biggest advantage, but to achieve this flexibility, a vast amount of measurement data are required to train the statistical or artificial intelligence models. Without any physical constraint between parameters, a long-
existing problem of training and over-fitting can be a big drawback which makes the fitted model less generic. Another disadvantage of the pure data-driven method is its interpretability, as it is usually only a black-box mapping measurements to the desired targets, and no further information about the traffic state can be extracted from the fitted model.

Due to the flexibility and interpretability issues in the two approaches described above, researchers recently start exploring the combination of these two, i.e., a data-driven approach with additional constraint based on physical rules in the traffic network—a so-called a hybrid approach. In this chapter, we describe model-based approaches, data-driven approaches, and then hybrid approaches. In the last subsection of this chapter, the travel time or delay distribution models are also reviewed since they can potentially be embedded with the models which consider the travel time or delay as deterministic variables, resulting in a better solution.

2.1.1 Model-Based Approaches

The model-based approach can be classified as either macroscopic models that estimate the queue formation and discharge, or as microscopic models that simulate the detailed trajectories of all the vehicles in the whole network.

Macroscopic Models

In a hydrodynamic model, the traffic flow is represented as a continuous flow and parameterized by macroscopic parameters such as speed, flow, and density. The first order continuum theory of highway traffic to model the traffic dynamics, known as Lighthill-Whiteam-Richards model (LWR) [18, 19] was first proposed in the 1950s. In order to improve on specific deficiencies of this model, higher order models
were proposed starting in the 1970s [20]. However, some debate was generated on the usefulness of these models in the traffic research community [20, 21]. The application of LWR models for travel time estimation was first applied to freeways [22] where the on-ramp and off-ramp detectors are available at every freeway entrance. This makes the model very difficult to be transplanted to urban road networks due to the variety of movements on urban roads and limitations of sensor coverage. In [23], the existence of a fundamental diagram is discussed and data from multiple cities are shown. The author presents that even though the traffic parameters from individual detectors do not show a clear and consistent relationship, the aggregated occupancy and travel production (veh*km/hr) over the entire network results in a stable fundamental diagram. The authors also proposed that the traffic state estimation and prediction process can be applied to the entire network. The possibility of applying this aggregated approach on travel time estimation for a single link or route is not mentioned.

Transitioning from macroscopic to queue-level modelling, there are significantly more studies and applications for travel time estimation. In traffic engineering research, a queue is defined as vehicles delayed at a stop (e.g., an intersection) and is the major source of vehicle delay in an arterial road network. For an intersection, the formation of a queue starts from the beginning of a red light and discharging the queue starts at the beginning of a green cycle. Since the queue formation and discharge happen alternatively cycle-by-cycle, estimating the length of the queue at a single intersection can help to estimate the delay of vehicles at a certain location of the link given its arrival time. In early traffic research, a queue is simplified as a vertical queue, ignoring the distance on the link occupied by the
vehicles in the queue, and is assumed as a vertical stack locating in front of the stop bar. This assumption simplifies the calculations, but are obviously unrealistic when compared to a real traffic scenario, particularly when the queue length is comparable to the link length. The vertical queue application in traffic analysis was first proposed in [24]. An examples of its application on delay estimation was presented in [25] where the distribution of queue length and delay is modelled in a fixed-cycle traffic-light (FCTL) scenario. The bounds of the queue length and delay approximation under overflow situation is discussed in [26] using the same method.

A horizontal queue approach is a more realistic queue model which overcomes the spatial limitations of the vertical queue theory. The queue formation and discharge process is spatially and temporally modelled with the addition of shock-wave theory to approximate the queue length and vehicle travel time. Horizontal queue theory was first proposed in [27, 28], and applied to delay estimation for arterial intersections where the authors modified the vehicle dynamics during the queue discharge [29]. As with other model-based methods, the queue-based theories also require the calibration of parameters and has specific assumptions especially for the arrival distribution of vehicles from upstream of the intersection. In a complicated urban road network, vehicles always arrive as platoons due to the queue discharging in the previous intersection, a simple arrival distribution assumption which might mislead the derivation of the travel time estimation.

Recently, with the advances of traffic monitoring sensing technology, horizontal queuing theory can be incorporated with dedicated sensors. Liu et al. in [30] presented a real-time queue length estimation method based on high-resolution
loop detectors and signal status collectors. Then in [31] Liu et al. applied a so-called “virtual probe vehicle” to estimate the travel time along that link at a given time. This method overcomes the limitation of the vehicle arrival distribution assumptions but the coverage of such dedicated data is very limited. In [15], travel time data collected from probe vehicles are used to find the break-point of the traffic cycle and estimate the queue dynamics. This method has the same advantage as [30] but it required a high penetration rate of probe vehicles (more than 50%) and accurate travel time measurements from those probe vehicles. Operating a large size probe fleet is expensive and the more-populated cell phone positioning data are always sparse that cannot provide a direct travel time report within a specific distance.

**Microscopic Models**

In a microscopic modelling view, the vehicles in the road network should follow certain rules to move from one position to the next. If modelling the link as a continuous space, the movement of a single vehicle should be restricted by the geometry of the road, the vehicle in front of it, and traffic signal controls. Other irregular events like the crossing pedestrians or car accidents can also be modelled. So-called car following models are models that describe the position dynamics of the following car according to the current velocity and acceleration of itself and the distance to the preceding car. More complex models also take the dynamics of preceding car(s) into account. Examples of car-following models can be found in [32-34]. With these models and an origin-destination matrix, the traffic can be simulated. But the model is very sensitive to calibration parameters and consumes a large amount of computation, so it is rarely carried out for large scale networks. Many
applications of these models can be found, with simulation software such as PARAMICS [35], VISSIM [36], Transmodeler [37], and SUMO [38].

A Cell Transmission Model (CTM) is another type of microscopic model which discretizes the link as an array of cells, each of which holds a number of vehicles. In each simulation time step, the vehicles in one cell will be transferred to the next cell under a certain rule depends on the capacity of cells. This model was first developed in [39, 40]. Application for traffic state estimation based on CTM typically incorporates Extended Kalman Filters (EKF), as an example, [41] provided a general framework. The same framework on an urban road network is discussed in [42]. The nonlinear CTM is transcribed in a closed analytical state-space form for use within a general extended Kalman filtering framework to provide measurement models for the traffic state and model parameters for automatically estimating traffic conditions and model parameters in an online context. Another CTM-based delay estimation method in a signalized intersection network was propose in [43] where the delay at each time step was modelled as the difference between its current occupancy and the outflow. The step-by-step accumulative delay can then be formulated as an objective function for signal control optimization. Similar to car following models, it has high computational complexity and is difficult to implement for real-time traffic state estimation. However, CTM can be populated during offline planning and can be useful for signal optimization.

2.1.2 Data-Driven Approaches

Unlike model-based approaches, data-driven approaches directly connect available measurements with travel time or delay, avoiding the calibration or
assumption of complex physical models. Different statistical tools and artificial intelligence frameworks are applied to traffic research in a wide variety of literature. The common problem of the pure data-driven approach is that there is no model of traffic state, and travel time, delay, and flow are considered as single target values.

In time-series techniques, autoregressive integrated moving average (ARIMA) models are widely used for traffic state prediction which incorporate the traffic state or travel time in a previous time interval to the state in the current estimated or predicted interval; this way, evolving trends of traffic transitions can be captured. Particularly for travel time estimation and prediction, Yang et al. in [44] and Sisiopiku et al. in [45] apply the ARIMA model and use GPS-equipped vehicles to collect data for model fitting. The calculated section travel time data are treated as a realization of a time series. Vlahogianni et al. in [46] provide a critical discussion of short-term traffic forecasting techniques for both freeway and arterial roads. To fit such a model and use it for prediction, travel time over each link before the estimated time step must be collected. Due to the limitation of sensing technology of direct travel time measurement, these models not very suitable for current traffic monitoring system.

Artificial Neural Network (ANN), also known as a non-linear regression model for traffic prediction purposes, is the most widely applied approach for traffic prediction. Among all existing ANN models, State-Space Neural Network (SSNN), as a generic arterial of recurrent neural networks, is the best performing to predict arterial travel time, according to [47]. Abu-Lebdeh et al. in [48] presented a SSNN combined with conditional independence (CI) graphs to analyse the independence and
interaction among variables involved in the travel time process. Other researchers [49, 50] also present SSNN-based models on different network structures and input data representations. Although SSNN approaches described in these publications are shown to be promising for arterial travel time prediction, some difficulties exist for urban networks, such as correctly capturing turning fractions and complex traffic conditions along the roads, as discussed in [51].

The previous mentioned statistical tools (time series analysis) and artificial intelligence tools (SSNN) have similarities and differences. Although the goal of both approaches is the same, researchers frequently fail to communicate and even understand each other’s work [52]. Karlaftis et al. in [52] discuss differences and similarities between these two approaches and review relevant literature and attempt to provide a set of insights for selecting the appropriate approach.

2.1.3 Hybrid Approaches

While the model-based approach needs to extend flexibility to the real world data and reduce the complexity of model parameterization, and the pure data-driven approach is quite adaptive to the data but also relies too much on the data and has the lack of interpretability to use the trained model, there should be an opportunity to combine those two to solve the current high-dimension dynamic traffic problems for urban road networks.

The Mobile Millennium project is a great trial for a hybrid approach presented in multiple publications (see, e.g., [53…]). The sparse GPS data from smart phones over an urban road network (downtown San Francisco) are sampled every minute. The goal is to estimate the travel time on every single link (including delay on the
downstream intersection) in real time. Hofleitner et al. in [54] first developed a model for probability distribution of travel times between arbitrary locations based on kinematic wave theory. This travel time distribution is parameterized by basic intersection traffic states like such as cycle length, flow, and queue length. The data streamed back from probe vehicles are aggregated every 15 minutes and are used to fit the model. Herring et al. in [55] present a statistical approach by modelling the evolution of traffic states as a Coupled Hidden Markov Model (CHMM), which is a particular form of a probabilistic graphical model. This model connects the links with traffic states contained in the previous described travel time distribution. Finally in [53], the overall system was trained under an Expectation-Maximization (EM) algorithm based on all the data collected over the entire road network and all the time intervals. The overall estimation and prediction results are evaluated as the mean square error of estimated/predicted travel time to the travel-time collected in a three-day field tests. This project gives a good example of the combination of traditional traffic model and machine learning approach, but the complexity of the framework makes the large scale computation non-trivial.

2.2.4 Travel Time Distribution Modelling

As described in Chapter 1, Travel Time Distribution Modelling should be modelled as a random variable with specific probability distribution, due to the variability of travel time on arterial roads. Current travel time modelling can also be generally categorized into traffic-model-based and data-driven. Olszewski et al. in [56] proposed a cycle-average delay probability distribution model based on the sequential calculation of queue length and with the assumption of certain vehicle arrival
distributions. Hofleitner et al. in [54] and Zheng et al. in [57] expand this model but follow the same horizontal queuing theory but trained this model with sparse probe data. Zheng et al. in [58] also explores the uncertainty and predictability of the travel time under this model.

In addition to these models derived from queuing theories, there are data-driven models that fit the data with well-established probability distributions for a more compact representation and better flexibility. Emam et al. in [59] compared the different distributions (log-normal, Gamma, Weibull and exponential distributions) on freeways travel time data. However, these single-mode distributions are not very suitable for the travel time crossing intersection where multiple components can be contained in the travel time. Guo et al. in [60] presents a mixture of Gaussian for the travel time modelling in order to capture the multi-mode property of arterial travel time distribution. Loustau et al. in [61] present a mixture of log normal model with the same motivation. All of these models increase their flexibility but due to the lack of consideration of hidden constraint underlies the traffic physics, they suffer from initialization and results determination issues. A discrete travel time distribution model was proposed in [62], where the travel time space is divided into a finite set of states. This heuristic discretization is intended to make the travel time states easier to be combined over multiple links (along a route) based on Markov Chain model. The number of discrete states might vary with the distribution of travel time changes, otherwise the distribution estimation accuracy will decrease.
2.2 Lane-Level Vehicle Trajectory Collection using Mobile Sensors

As described in Chapter 1, to estimate lane-level vehicle trajectories, positioning, trajectory extraction, and maps are three necessary components. Positioning and mapping sometimes interact with each other so we give the review of these two parts together in the first subsection below. Vehicle trajectory extraction from range sensors usually depends on robust detection and track algorithms to locate the surrounding vehicles frame-by-frame. This is isolated to a high accuracy positioning technology and map, so we give the review of this part separately in the second subsection.

2.2.1 Positioning and Mapping

The goal of positioning and mapping in trajectory collection system is to build a bridge between a sensor platform and the geometric map so that the tracked vehicle trajectories with respect to the sensor platform can be mapped to the lane regions because they need lane number for driver interaction analysis. There are two main categories. The first category requires an accurate absolute position measurement or estimation of sensor-carried vehicle (ego-vehicle) in Earth Centered Earth Fixed (ECEF) frame-of-reference and the map or lane regions that are stored in an analytical format on the vehicle also in ECEF frame-of-reference. When the high accuracy global positioning measurements are not available on the ego-vehicle, the second category determines the position of the sensor platform relative to the map ranging from a sparse collection of features up to a dense set of 2D/3D points. The positioning
of the features can be either collected by a high-accuracy positioning device or averaged from multiple runs by the ego-vehicle with a lower accuracy.

The sensor fusion between Global Positioning System (GPS) and Inertial Navigation System (INS) is extensively studied to provide an absolute position measurement for vehicle positioning. Bayesian Filters like Extended Kalman Filters [63], Unscented Kalman Filters [64], and Particle Filters [65] are openly implemented to correct the INS estimation by the less frequent GPS measurements. The lane-level accuracy specifications are still under discussion, but are expected to be at the decimeter level (50cm or less) [17] which is much higher than the accuracy of a commercial GPS alone. Carrier Phase Differential GPS (CPDGPS)-aided INS or CODGPS-aided encoders [66] are capable of estimating vehicle ECEF positioning with centimeter level accuracy. In [3], an offline smoothing algorithm was proposed to increase the accuracy of position estimation by solving the carrier-phase integer ambiguities accurately and reliably by using multiple epochs of data, and spanning across long time intervals. Once the absolute positioning estimation is archived by the sensor platform, the map with absolute geo-referenced coordinates can be built by such a sensor-equipped vehicle running at the center of each lane during an extensity survey that covers all lanes in the road network. [17] provided such an example. The authors gave a representation of a single segment of single lane with a series of vertices which can be applied to the numerous lane segments that comprise any given roadway. In [67], an alternative representation “Emap” was proposed where instead of samples of shape points, the roadway data consists of using a series of straight lines, circles, and clothoids which all obey the same curvilinear 2D equations. The clothoids
fit the actual road shape better, which decreases the amount of information to be stored in the database.

When the lane-level accuracy is not available on positioning sensors or maps, the lanes need to be detected by the ego-vehicle so that it can map the trajectories of surrounding tracked objects and itself to the right lanes. This task should be realized by vision or range sensors which can differential lane makers from the environment or capture the surrounding area as an image. Given a sub-accurate initial position estimation, e.g. from GPS & INS sensors, the correction between the sensor measurements and map features (lane makers or map image) can provide a high-level position estimation. Lane maker detection by camera has been studied over decades especially used for lane-departure warning or early stage automated driving. In [68], the digital map with lane makers were generated from aerial imagery, and the lane makers detected by camera on the ego-vehicle were used for map matching. Centimeter-level accuracy was reported. A popular way to build a map under this condition is by simultaneously localization and mapping (SLAM) technique [69].

SLAM methods benefits from dense data, so the 3D LiDAR with a 360 degree field of view usually gives superior localization estimates and map quality. Levinson et al. in [70] proposed an approach that utilizes a platform of GPS, INS and 3D LiDAR to generate a high-resolution infrared remittance ground map that can be subsequently used for localization. The environment was represented, instead of as a spatial grid of fixed infrared remittance values, as a probabilistic grid whereby every cell was represented as its own Gaussian distribution over remittance values. By using offline SLAM to align multiple passes of the same environment, possibly separated in time by days or even months, it is possible to build an increasingly robust understanding of
the world that can be then exploited for localization. Moosmann et al. in [71] proposed a similar idea from localization except point clouds were represented in 3D and more shape features (3D position, normal vectors and normal confidence) were applied, but no intensity. The accumulated maps over multiple collections can also be refined to obtain a final map containing more details.

### 2.2.2 Detection and Tracking Moving Objects

The second component of trajectory collection system is the detection and tracking of moving vehicles, which can also be regarded as the distinction between background and the moving vehicles. In the real traffic scenario, background in LiDAR measurements is formed by multiple sources: reflection from road surface, road infrastructures like light poles, traffic signs, and buildings. Even vehicles parked on roadsides and pedestrians moving on the road are not the objects we are interested in tracking. The detection of vehicles are the first to be studied in the LiDAR application in ITS. Almost every paper contains detection algorithms to extract vehicle-like objects from the background.

The measurements of 2D LiDAR are an array of points represented in a polar coordinate system where each rotation angular interval contains only a single distance measurement. With this simple type of data type, vehicle detection is usually based on some basic geometric feature extraction, e.g. lines, arcs, arbitrary-shape clusters within a small area. Breaking point detection is first applied to determine sequences of measurements which are not interrupted by scanning surface changing. Line extraction is performed to each continuous scan sequence in a range image by applying line kernels. Borges et al. in [72] proposed a comprehensive summary on
line extraction in 2D range images. In [73], a simple line detection method was applied to a real traffic scene to detection and track obstacles around a bus. The lines and circles are tracked independently. A box model is another popular geometric shape feature for the tracked vehicle which consists of four lines and well fit the laser measurement from vicinity of the vehicle. One problem of the box model in tracking is the initialization. Because the vehicle body always blocks the other side of itself from the laser beam, only one or two sides of the vehicle will show in the LiDAR field of view which makes it very difficult to determine the orientation and accurate position of the vehicle. Streller et al. in [74] applied this model on vehicle tracking on roadway, and multiple box models with different initial orientation and position were created from each point cluster with valid group of lines. The real model can be discovered through following tracking with Multiple Hypothesis Track (MHT). The same initialization strategy was used in [75] as well, but coupling was used with an Interacting Multiple Model.

Recently, 3D LiDAR sensors appear more-and-more in ITS research papers, especially for automated driving [76-78]. An additional processing step required in 3D data processing is to segment objects above the road surface and the massive reflection points from the road surface. Leonard et al. in [76] and Montemerlo et al. [77] developed the same method independently to achieve this goal. The 3D point cloud is first aggregated to a virtual scan representation which is a 2D matrix representing azimuth and vertical angles in rows and columns with the aggregated distance in each cell. By going through each column in a virtual scan, a break point is defined to be the first point from the lowest vertical angle with a large angle between its two neighbors. All the points before the break point in each column are classified
as ground points. This method works fast and well to classify the obstacles in close range, but small measurement error in the ground reflection can cause the break point searching to stop earlier and therefore misses the vehicles at longer range. More complicated segmentation approaches were presented in other publications. Most of them are based on graph algorithms which represents the 3D point cloud as a weighted graph where the weights represent the similarity between two connected nodes. Different geometric features can be used to compute the similarity, such as edge distance [79], curvature [80], surface direction [81], local convexity [82] and local smoothness constraints [83]. The segmentation is then taken as to find an optimal clustering on this weighed graph. Some heuristic rules on local convexity were developed in [82] to archive fast segmentation. Graph cuts based algorithms play an important role, like the normalized cuts algorithm of [84], which minimize some cost function. This produces a global optimum at convergence while a high computational cost is a trade-off. Shi et al. in [84] alternatively generates a minimum spanning tree and apply recursive cutting which keeps nearly linear costs while still finding a good local minimum. Himmelsbach et al. in [85] grouped the points based their 2D projection on the ground plane, then a supervised classification algorithm Support Vector Machine (SVM) was applied to classify background and vehicles. Anguelov in [86] also applied a machine learning algorithm on the feature space generated from original point cloud to learn a Markov Random Field.

In addition to these explicit detection approaches, grid map-based approaches are also used with range sensors in traffic applications. By tracing all laser rays reaching the sensor, a space can be explicitly detected as free up to the measurement point. Storing this information within a 2-dimensional grid leads to the well-
established occupancy grids [87]. The moving vehicles, not staying on a fixed location, are identified as group of occupied grids. The tracking can also be taken on each single grid rather than a cluster. This method was originally developed for 2D applications [88] but proved to be efficient with 3D data in automated driving applications [77].

In the traditional tracking pipeline, data association between the detected objects and the existing tracks is the most challenging part because of the association ambiguities that arise. Multiple Hypothesis Tracking is typical approach to be carried out in this stage. [89] gives a comprehensive review of MHT based object tracking. A popular alternative of MHT is Joint Probabilistic Data Association [90] which associates multiple detection to a single track.

Tracking is dominated by variants of Extended Kalman Filter (EKF) frameworks due to their lower computation complexity [91]. Interacting Motion Model (IMM) can also be used with EKF as an augmentation [92]. Rao-Blackwellized Particle Filter are an alternative for multiple target tracking on simulated data [93].

A new framework of moving object tracking is based on the geometric model of the tracked object and does not require detection and data association before the track updating in every frame. It avoids the association ambiguity which frequently occurs in congested traffic scene. A suitable geometric model for the tracked objects is a precondition in this new framework and a good initialization for this model is always helpful to start the tracking procedure. Therefore, the detection step still takes its place in the new framework, just once for initialization not every frame. In each
frame the model will be propagated based on the current state of the track as a prior and the most likely state will be updated by maximizing the likelihood of predicted model with the measurement data. This method is first presented in [2] for autonomous driving. The box model mentioned above is the model for 2D vehicle tracking and a particle filter is applied due to the high non-linearity between the measurement (position of laser reflections on the box model) and the vehicle state. The shape parameters of the tracked vehicle can be estimated along with the vehicle pose. [94] proposed a same framework with different measurement likelihood computation formulas. Vu et al. in [95] applied this model-based method to an offline smoothing framework for searching optimal set of vehicle trajectories over a long time period. The performance of this approach depends on the measurement likelihood which differentiate the cost among correct reflection, occlusion, and free space. If the measurement model can be well designed, it is promising to be extended to 3D applications.
Chapter 3. Arterial Roadway Travel Time Decomposition and Modelling

3.1 Overview

Estimating arterial roadway travel time is crucial to the development and application of both Advanced Traveller Information Systems (ATIS) and traffic management systems which require detailed traffic congestion information to make better decisions for routing and signal operation. Compared with the case of freeway segments, estimating the link travel time in urban arterial network is much more challenging since the traffic conditions are more complicated and the movement of vehicles may be interrupted by control devices (e.g., traffic signals), pedestrians, or other events. Therefore, arterial link travel time distributions (TTD) have been getting more attention in recent years since they contain valuable information of arterial traffic state.

In order to estimate TTD, a considerable amount of travel time measurements is required. Advances in traffic monitoring techniques now offer a variety ways of arterial travel time collection. Fixed sensors like inductive loop detectors [6], wireless magnetic sensors [7], and Bluetooth [8] can measure the travel time based on vehicle re-identification algorithms. Mobile sensors such as probe vehicles equipped with GPS can provide second-by-second position information on which travel time can be calculated. Due to the rapid proliferation of smart phone users, data streaming back from mobile devices have become an important source of travel time information.
These data can provide extensive coverage both spatially and temporally, however there are also challenges due to their sparseness. Recent success on estimating TTD based on a sparse mobile data was achieved in [54, 57]. The authors developed similar mixture models to capture a two-mode travel time distribution when going through an intersection. Although those models fit the TTD well when the uniform vehicle arrival assumption holds, traffic flow interrupted by the upstream traffic signal can lead to the violation of this assumption frequently in practice. Furthermore, the complexity of the model distribution will lead to inefficient numerical computing for each data point which will be mentioned in II.B. Therefore, for a practical TTD estimation, a more flexible and efficient model is needed.

In this chapter, we describe a modified Gaussian Mixture Model (mGMM) to represent the arterial link travel time distribution which can be trained efficiently on both fixed sensor data and mobile sensor data. The resulting good fit of the probability density function and the high accuracy rate of vehicle stop/non-stop movement classification show great potential of this model for use in several traffic applications, such as arterial travel time prediction and arterial traffic energy/emission estimation [96]. The remainder of this chapter is organized as follows: Section 3.2 presents the approach for modeling link travel times from fixed and mobile sensor data; Section 3.3 describes the experiments using actual fixed sensor data and virtual mobile sensor data, along with the results of model performance in comparison with ground truth data; and Section 3.4 provides conclusions and recommendations for future research.
3.2 Methodology

3.2.1 Travel Time Decomposition

*Link Description*

In arterial travel time studies, one link segment is defined as the link between two signalized intersections, which also includes one of the two intersections, as depicted in Fig.3.1. In this paper, we define a link to include the downstream intersection. Vehicles may pass through the intersection with or without any delay. If a vehicle passes through the intersection during the green phase without any unintended deceleration, then we consider it to be under free-flow condition. Fixed sensors are usually installed right after each intersection in order to capture the travel time on the entire link without loss of much delay. Travel time can also be calculated from the time difference between two consecutive position data (e.g, \(x1\) and \(x2\) in Fig.3.1) sampled by mobile sensors every fixed time interval (e.g., 20 seconds). Note that the intersection delay region (orange segment) represents the segment on the link where queues usually occur. Some sampled mobile data points may fall inside this region (e.g., \(x2\)). Thus, when calculating travel time from mobile sensor data, the
calculation should be made based on pairs of data points that cover the intersection delay region. Otherwise some delayed time will not be captured [97].

**Travel Time Decomposition**

The variability in travel time that vehicles experience on arterial links is due to: 1) the variation of individual driver’s behaviour and 2) the waiting time for traffic signals or queue dissipation which depends on the arrival distribution of the vehicles, changing queue length, and signal timing. All these stochastic elements make it difficult for arterial travel time to be modelled as a single distribution. Decomposing arterial travel time into two components, free-flow travel time and delayed time, and modelling them with separate distributions can improve the estimation of the total travel time. Therefore, we formulate travel time \( tt_{x_1,x_2} \) between two positions \( x_1 \) and \( x_2 \) as

\[
\begin{align*}
    tt_{x_1,x_2} &= l_{x_1,x_2} \cdot p + d \\
\end{align*}
\]

where \( l_{x_1,x_2} \) is the distance between \( x_1 \) and \( x_2 \). For mobile sensor, \( x_1 \) and \( x_2 \) are the position of two data samples while for fixed sensor, they are always the static locations of two sensors. \( p \) is the free-flow pace (sec/meter) which is the reciprocal of free-flow speed, and \( d \) is the delayed time.

The delayed time was modeled in a previous research [54, 57] as a mixture of a Dirac Delta distribution and uniform distribution with the assumption of uniform arrival pattern upstream. The travel time \( tt_{x_1,x_2} \) turns out to be a mixture of Gaussian and quasi-uniform distribution, given that the probability of the summation of two random variables is the convolution of the probability of those two variables. When
fitting this model on a large historical dataset, the computation complexity is high due to having two integrals in the quasi-uniform distribution’s probability density function. Therefore, in our study we selected the Gaussian Mixture Model to fit the distribution of the delayed time, which made the model fitting process more efficient.

**Travel Time Data Description**

As mentioned previously, there are two major types of sensors for travel time data collection. Without loss of generality, we define the data from both types of sensors to have the format \((tt_{x1,x2}, l_{x1,x2})\). For fixed-location sensors, \(tt_{x1,x2}\) is the travel time measurement and \(l_{x1,x2}\) is the constant distance between two sensor locations, \(x1\) and \(x2\). For mobile sensors, \(x1\) and \(x2\) are the locations of two data samples from which \(tt_{x1,x2}\) and \(l_{x1,x2}\) are calculated.

The fixed-location sensors used in our study are the wireless magnetic sensors. Five to seven sensors were installed as an array 12 feet after each intersection, recording the magnetic signature and timestamp of vehicles passing over them. The sensor arrays find matched vehicles by comparing the peak features in the resulting signature while constrained by the order of vehicles in the flow. Approximately 70\% [7] of the vehicles can be matched when the vehicles pass through a single intersection. Due to the shifted relative horizontal position when vehicles pass the sensor array and to similar ambiguous signatures between vehicles of the same model, a 100 percent matching rate cannot be achieved. However, the existing matching rate, which is equivalent to the data observation rate, is already better than most probe vehicle data in the field [98].
Since a large-scale mobile sensor dataset is not available at the same site, we simulated virtual mobile data from the fixed-location data. According to Eq. (1), since we will only use mobile data that cover the entire delay region in order to guarantee the capture of full delay, the key difference between the mobile data and fixed-location data is the data sampling positions $x_l$ and $x_2$, or $l_{x_1,x_2}$ in the equation. If we can simulate the data sampling positions $x_l$ and $x_2$, and assume a reasonable estimate for free-flow pace $p$, then the virtual mobile travel time can be calculated as 

$$tt_{x_1,x_2} = tt_{fixed} + (l_{x_1,x_2} - l) \cdot \hat{p}. \quad (2)$$

Given that the delay region covers a factor of $\alpha$ of link $i$, $x_l$ is sampled between the start of the link and $(1 - \alpha_i) \cdot l_i$, and $x_2$ is sampled after the downstream intersection between the end of the link and $l_i + (1 - \alpha_{i+1})l_{i+1}$. With predefined $\alpha$ and $l$, we uniformly sample $x_l$ and $x_2$, and calculate $l_{x_1,x_2}$. For the free-flow pace, one can choose the free-flow pace based on the road speed limit or the speed during off-peak. In our case, we selected $\hat{p}$ from the free-flow component in our modified GMM fitted on fixed-location data. The results in Section IV.B and IV.C imply that this is a reasonable estimate.

### 3.2.2 Modelling Approach

#### A. Modified Gaussian Mixture Model

In this section, the formulation of the mGMM is presented. In Eq. (1), $tt_{x_1,x_2}$ and $l_{x_1,x_2}$ are measured variables while $p$ and $d$ are the variables we want to estimate from the dataset. We model $p$ as a Gaussian random variable, 

$$p \sim N(\mu_{ff}, \sigma_{ff}^2) \quad \quad (3)$$
and define $tt_{ff_{x1,x2}}$ as free-flow travel time between $x1$ and $x2$,

$$tt_{ff_{x1,x2}} \triangleq l_{x1,x2} * p \sim N(\mu_{ff} \cdot l_{x1,x2}, \sigma_{ff}^2 \cdot l_{x1,x2}^2).$$  \hspace{1cm} (4)

Here, we select a mixture of Gaussians [99] to model $d$ as

$$d \sim \sum_{k=1}^{K} \pi_k N(\mu_k, \sigma_k^2),$$  \hspace{1cm} (5)

$$\mu_1 = \sigma_1 = 0,$$

$$\sum_{k}^{K} \pi_k = 1.$$  

Notice that we keep the first component of delay to be $N(0,0)$ which corresponds to the vehicles without delay and will not change its mean and variance during the model fitting process. $\pi_k$ is the weight of $k$th component and has to be summed to 1.

Since the total travel time is a summation of $tt_{ff_{x1,x2}}$ and $d$, its probability density is the convolution of those two. The convolution of two Gaussians is still a Gaussian with the addition of mean and variance,

$$P(tt_{x1,x2}) = P(tt_{ff_{x1,x2}}) \ast P(d) =$$

$$\sum_{k=1}^{K} \pi_k N(tt_{x1,x2} | \mu_k + \mu_{ff} \cdot l_{x1,x2}, \sigma_k^2 + \sigma_{ff}^2 \cdot l_{x1,x2}^2).$$  \hspace{1cm} (6)

We can see that $tt_{x1,x2}$ is a mixture of Gaussian, with a special feature that the mean and variance of the first component are tied to the others, which embodies the free flow travel time in each of the delay components. The GMM has multiple solutions for the same training dataset (known as identifiability problem [99]) making it a challenging issue when interpreting each component after the model fitting.
process. However, this setting helps us identify the free-flow component easily since it will always be the one with the smallest mean.

The solution of our model follows the Expectation and Maximization (EM) algorithm framework [99] for solving the original GMM. In EM, we need to first initialize all the parameters and iteratively update the parameters using expectation (E-step) and maximization (M-step) until the error of parameter estimate between two consecutive iterations converges. In the following subsections, we introduce some special treatment in the initialization of the parameters, followed by the presentation of solution methods for both fixed and mobile sensor data.

**B. Initialization**

Since the EM algorithm only returns the local optimum, the selection of initial parameter set may have considerable impacts on the final results. In particular, since the free-flow Gaussian is interweaved with each components of the mixture model in our case, it makes the initialization of $\mu_{ff}$ and $\sigma_{ff}$ even more critical. Field observations show that the traffic volume was quite low in the early morning period from 1 a.m. to 5 a.m. and most vehicles were travelling at free-flow speed. Therefore, we used the data samples collected from this time period to identify the travel time distribution under free flow. Fig.3.2 shows the histogram of vehicle travel time from 1 a.m. to 5 a.m.

As shown in the Fig.3.2, most of the travel time data are located in the region with short travel times (towards the left side of the distribution). In order to reduce the effect from delayed travel time data, we applied Gaussian with RANSAC technique (red curve) [100] to estimate the free-flow travel time distribution and compared it
with the fitted Gaussian without RANSAC (green curve). For the mobile sensor data, due to having different travel distances, we first assumed the vehicle was travelling at free-flow speed, and calculated its virtual travel time on the link by multiplying the measured travel time with the ratio of link distance to travel distance. For the vehicles that were actually delayed, their virtual travel times would be larger than the true travel time, but those overestimated data would be eliminated by the RANSAC technique. We then used the fitted Gaussian distribution to derive the initial value for $\mu_{ff}$ and $\sigma_{ff}$.

Besides $\mu_{ff}$ and $\sigma_{ff}$, the initial value of $\mu_k$ and $\sigma_k$ were initialized by uniformly drawing from the range $(\mu_{ff} \cdot l_{x1,x2} + 3 \cdot \sigma_{ff} \cdot l_{x1,x2}, \max(tt_n))$. We selected the lower bound of the range in a way that keeps the delay components away from the free-flow data. Also, the number of components is another critical parameter in the GMM. In a previous study [101], the authors proposed 4 components, representing free-flow vehicles, slow free-flow vehicles, fast delayed vehicles, and delayed vehicles. Although this setting lacks theoretical support, we used it as a reference and test $k = 2, 3, 4,$ and 5 in our experiments.

**C. Solving Modified GMM**

Solving the modified GMM is done by maximizing the total log likelihood:

$$\arg\max_{\theta} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k N(tt_n \mid \mu_{nk}, \sigma_{nk}^2)$$

(7)

$$\mu_{nk} = \mu_k + \mu_{ff} \cdot l_n$$

$$\sigma_{nk}^2 = \sigma_k^2 + \sigma_{ff}^2 \cdot l_n^2$$

$$\theta = (\mu_{ff}, \sigma_{ff}, \mu_k, \sigma_k, \pi_k)$$
where $\theta$ is the parameter vector to be estimated; $l_n$ is the travel distance of $n$th data sample; and $\mu_{kn}$ and $\sigma_{kn}$ are the mean and standard deviation of $k$th Gaussian component of $n$th data sample. Note that if the measured travel distance is different for different data samples (e.g., from mobile sensors), each data sample has a unique probability distribution resulting in $n \cdot k$ pairs of different mean, $\mu_{nk}$, and variance, $\sigma_{nk}^2$. Fortunately, these new parameters consist of the same elements: original mean, variance and measured travel distance. We now show the different solutions of this model for both fixed and mobile sensor data.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{Travel time histogram 0am-5am over one month, Aug 2011, at intersection of San Pablo Ave and Solano St, Berkeley, CA.}
\end{figure}

\textbf{a. Data from Fixed Sensors}

When data are from fixed sensors, $l_n$ of all the samples is a constant distance $l$ between the upstream and downstream sensors, or the length of the link. This implies that the mean and variance of each component in the mixture of Gaussians are
identical for each sample. Let the shared mean and variance be $\mu_{lk}$ and $\sigma_{lk}^2$ respectively. Then, our model can be solved in the same way as original GMM using the EM algorithm from which each parameter has a closed form solution:

$$\hat{\mu}_{lk} = \mu_k + \mu_{ff} \cdot l = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \cdot tt_n \quad (8)$$

$$\hat{\sigma}_{lk}^2 = \sigma_k^2 + \sigma_{ff}^2 \cdot l = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \cdot (tt_n - \mu_{lk})^2 \quad (9)$$

$$\pi_k = N_k/N \quad (10)$$

where $\gamma_{nk}$ is the probability of sample $n$ generated by component $k$, which is also referred as the responsibility of $k$th component to $n$th sample and $N_k$ is the effective number corresponding to component $k$.

\[ \text{Data from Mobile Sensors} \]

Unlike the data from fixed sensors, each data sample from mobile sensors may correspond to its own distance, $l_n$, which leads to $n \cdot k$ different pairs of Gaussian parameters. This results in having no closed form solution for those parameters. If we still follow the EM algorithm similar to when we solved the original GMM, in the E-step the responsibility is calculated as

$$\gamma_{nk} = \frac{\pi_k \gamma^{N(t_n|\mu_{nk}, \sigma_{nk}^2)}}{\sum_{j=1}^{K} \pi_j \gamma^{N(t_n|\mu_{nj}, \sigma_{nj}^2)}} \quad (11)$$

which has not changed. In the M-step however, the objective turns out to be maximizing the posterior total likelihood given the responsibility $\gamma_{nk}$:

$$\log L(\theta|tt_n, l_n, \gamma_{nk}) =$$

$$\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \{ \log \pi_k + \log(N(tt_n | \mu_{nk}, \sigma_{nk}^2)) \}. \quad (12)$$
The weight \( \pi_k \) of each component will remain the same for each data sample so they can still be solved in closed form as in Eq. (9). However, because of the existence of \( l_n \) in \( \mu_{kn} \) and \( \sigma_{kn} \), the closed form solution cannot be achieved for \( \mu_{ff}, \sigma_{ff}, \mu_k, \sigma_k \).

Instead, we can apply the gradient descent technique to solve this problem. There are multiple methods for setting up the cost function in gradient descent technique:

1) Use the total log likelihood as in Eq. (7);

2) Keep the framework of the EM algorithm by finishing the E-step and deriving \( \pi_k \) at the beginning of the M-step. Then set Eq. (12) as the cost function to estimate the rest of the parameters;

3) The same as 2), but optimize Eq. (7) at the last step.

Methods 1 and 2 are prone to the singularity problem, which means one or more Gaussian components may collapse to a single data sample leading to its variance shrinking to zero during the optimization process. Finally, method 3 gives us the most stable results in practice. In this paper, L-BFGS-B algorithm [102] is used in the optimization process.

### 3.3 Experiments

#### 3.3.1 Experimental Data

In this study, the experimental data were collected from a wireless traffic sensor network installed along several consecutive links on San Pablo Ave, Berkeley,
California. For the fixed sensor experiment, we selected the link from Washington Ave (upstream) to Solano Ave (downstream) with the length of 0.3 miles.

The whole dataset covers 24 hours for two months from July 15\textsuperscript{th} to September 15\textsuperscript{th} in 2011. The data between 6AM and 9PM on weekdays were selected and divided into 15 groups (15 hours, each over 5 weekdays) so that the statistical travel time pattern of each hour of the day over a number of days or weeks can be learned separately. The virtual mobile data set was generated from this dataset following Eq. (2). We selected $\alpha_i = \alpha_{i+1} = 0.5$ and $l_i = l_{i+1} = 0.3$ miles. Again, the free-flow pace was randomly drawn from the free-flow Gaussian distribution in the mGMM fitted on the original fixed-sensor dataset of each group. Examples of distribution fitting under different congestion levels using one week data are shown in Fig.3.3 with classification results. In this section, data between Sep 5\textsuperscript{th} and Sep 9\textsuperscript{th} are used as a representative of one-week data.

### 3.3.2 Model’s Goodness-of-Fit Evaluation

The fitted mGMM can be used as a representation of the arterial traffic state which requires a stable TTD estimate. In this subsection, we applied the Kolmogorov-Smirnov test (KS-test) to evaluate the goodness-of-fit of our model on the empirical travel time distribution. The KS-test is a nonparametric test for the equality of continuous probability distributions that is based on KS-statistic. When a significance level is set, the hypothesis, $H_0$: The data follow the specified distribution, is rejected if the KS-statistic is larger than the critical value for that significance level, or if the corresponding $p$-value is smaller than the significance level. In other words, the smaller the $p$-value is, the less confidence in the correspondence between the
observed data and the test distribution. Here, we use the \( p \)-value as the measure of goodness-of-fit of our models.

Fig.3.4 shows the test results of models with the number of components varying from 2 to 5 fitted on fixed sensor data (blue curve) and *virtual mobile data* (green curve). In the top plot, we can see that the mixture model with two components has \( p \)-value smaller than 0.1 in most of the time periods, implying a poor fit on the data. The mixture models with 3 to 5 components fit the actual data distribution better and have smaller variation over different time periods. There is no obvious improvement when the number of components is increased from 4 to 5. Although a GMM with a larger number of components generally provides better probability density estimation, it may lead to more computational complexity (for gradient descent) and the risk of overfitting (which is not desirable since we will use the model for stop/non-stop movement classification later). Therefore, the model with 4 components is preferred in this case.

The green curves in the Fig.3.4 are the average \( p \)-value of the models fitted on *virtual mobile data*. We generated, fitted and tested the model on *virtual mobile data* for 30 times in each group (one hour period) of the dataset so that the randomness from this “simulation” would be reduced. The error bars on the curves show the standard deviation of the \( p \)-value over those 30 experiments in each group. These results show that our model has potential to work with mobile sensor data as well as fixed sensor data.
Fig. 3. TTD estimate under different congestion level. The mixture distribution learned on fixed data (blue) and virtual mobile data (black) in time period 5pm-6pm (top) and 6am-7am (bottom) over 1 week, Sep. 5th-Sep 9th.
Fig. 3. 4 p-value from KS-test. The models are trained on one week data between Sep. 5th and Sep. 9th, 2011 at intersection of San Pablo Ave and Solano St, Berkeley, California. Blue is used for fixed sensor data, green for virtual mobile data.
3.3.3 Vehicle Stop/Non-Stop Movement Classification

If the movement of vehicle with travel time measurement can be identified as stop or non-stop, the delay ratio of the traffic flow can be estimated. When applying the mGMM for stop/non-stop movement classification, the test sample \((tt_{x_1,x_2}, l_{x_1,x_2})\) is classified based on its responsibility \(y_{ik}\) from each component. In our case, if the free-flow component has the responsibility higher than the sum of others, the sample will be classified as free-flow. However, in Fig.3.5, some samples in the most left region have higher probability to be classified as delayed having very small travel time. In our case, assuming the travel time of a delayed vehicle cannot be smaller than the free-flow travel time in early morning, the additional criteria is added to guarantee a reasonable result. The data \((tt_n, l_n)\) is classified to be free-flow if

\[
Y_{n1} > \sum_{k=2}^{K} Y_{nk} \quad \text{or} \quad tt_n < \mu_{ff} \cdot l_n. \tag{13}
\]

In order to evaluate the delay classification performance, a large-scaled ground truth dataset was collected from 11AM to 4PM on September 8th, 2011 using a video camera placed along the link. The video of the vehicle movements were captured while the wireless sensors simultaneously recorded the travel time. The movement type, either free-flow or delayed, of 1,400 vehicles was visually verified.

We tested the models with the number of components from 2 to 5 and estimated the vehicle movement. Then, we compared the correct rate using the model fit on data from fixed-location sensor and virtual mobile data. As a statistical classifier, the performance of the model using different size of training data is also shown. We performed the model fitting on one month, one week, and one day worth of the experimental data. Fig.3.7 shows that the mixture model with only 2
components does not provide good results for vehicle movement classification. Its correct rate is around 60%. The models with more number of components achieved over 90% correct rate even on one day data which is high enough for most of the traffic applications that require information about percentage of stop vehicles. Note that it might be confusing that the correct rate of the model with 2 components fitted on virtual mobile data is always higher than the one fitted on fixed sensor data. A poor fit does not necessarily results in a bad delay classification boundary. The model may have a poor probability density estimate but it does provides a good classification boundary. Based on the low $p$-value for the 2-component model shown in Fig.3.4, we consider the results in the first row of Fig.3.7 to be not reliable. Two examples of classification are shown in Fig.3.6 for congested traffic (a) and uncongested traffic (b).

Fig.3. 5. Travel time histogram 11AM-12PM over one month, Aug 2011. The green curve is free flow component and the red curve is the delay distribution formed by all the other components. Green bins are the data classified as free flow and red bins are the delayed data.
Fig. 3. TTD estimate under different congestion level. The mixture distribution (blue) with free flow (green) and delay (red) component are shown with the histogram at time period 6pm-7pm (top) and 7pm-8pm (bottom) over 1 week, Sept. 5th-Sep 9th.
Fig. 3.7. Correct rate of classification of GMM with 2-5 components and learned on dataset of one day (Sept. 8th), one week (Sept. 5th - Sep 9th) and one month (Aug. 5th - Sept. 9th, 5 weeks).
3.4 Summary

In this chapter, we described a modified Gaussian Mixture Model for estimating the distribution of arterial travel time and classifying vehicle stop/non-stop movement. The goodness-of-fit of the model was evaluated using experimental datasets, which included travel time measurements from fixed-location wireless magnetic sensors over two months and virtual mobile data generated from the fixed sensor data. The accuracy of the vehicle movement classification by the model was evaluated against ground truth data of 1,400 vehicles. The evaluation results show that the proposed model, with a proper number of components, performs well on both travel time distribution estimation and vehicle movement classification.

In future work, the model should be tested on actual mobile data over a longer time period. And when estimating travel time in a large-scaled road network, the interactions among adjacent links should be considered. Finally, the potential of real-time traffic state estimation of this model using real-time data stream should be studied.
Chapter 4. Arterial Roadway

Energy/Emissions Estimation using Trajectory Reconstruction

4.1 Introduction

As fuel costs continue to rise, combined with increased concern on both greenhouse gas and pollutant emissions, there is a strong need to better estimate overall fuel economy and emissions for roadway traffic. Intelligent Transportation System (ITS) technology and methods can be used to improve these estimates. A variety of energy and emission modelling tools have been developed for specific vehicles and specific driving conditions, however, it is also desirable to estimate aggregate traffic energy/emissions for roadway networks.

In the previous chapter, we have introduced a reliable travel time model based on travel time decomposition and the portion of vehicles without stop can be inferred from this model. In this chapter, we use this model to classify the movement of a vehicle into stop and non-stop given a single travel time. The data to calibrate our model and validate the classification is collected by a state-of-the-art traffic sensors located near traffic signals along the corridor. For the stopped vehicles, it is possible to re-create approximate trajectories of the vehicles travelling along the corridor. Cycle breaking and queue length estimation technics are applied to make more
accurate trajectory reconstruction. These approximated trajectories can be run through a micro-scale energy/emissions model and then integrated for all vehicles to get an overall energy/emissions estimate for all traffic.

In Section 4.2 of this chapter, we briefly describe the various methods of roadway energy/emissions estimation. In Section 4.3, we describe new traffic sensors that have been developed to not only provide speed, density, and flow information, but also can re-identify a substantial fraction of the vehicles from one sensor to the next, providing overall estimates of individual vehicle travel times. In Section 4.4, we describe the overall estimation methodology using a modal-based trajectory reconstruction approach. In Section 4.5, real world data results have been collected and are described, followed by conclusions and future work in Section 4.6.

4.2 Roadway Energy/Emission Estimation

There are several methods that can be used to estimate total vehicle energy/emissions of a roadway. These methods have varying levels of detail in data need and analysis technique. Assume that there is a single type of vehicle on the roadway and that the length of the roadway is known, the total vehicle energy/emissions of the roadway traffic may be estimated using one of the three generic approaches described below:

- **Macroscopic approach** – In this simplistic approach, a single constant emissions factor is used and the total emissions are simply a product of that constant multiplied by the traffic volume. This approach may be used when only traffic count data are available, as in the case of Highway Performance...
Measurement System. However, the estimation error from this approach is likely to be very high.

- **Mesoscopic approach** – Because vehicle emissions vary substantially by speed, a more accurate approach would utilize emission factors that are a function of speed. There are several methods to estimate average traffic speed, and this can be used as an index for a speed-based emission factor. In this approach, the average traffic speeds are binned and the associated emission factors for each speed bin are multiplied by traffic volume in the corresponding speed bin. Then, the emission estimates from each speed bin are aggregated to result in total emissions for the roadway. This approach is commonly used when average traffic speed of roadways can readily be obtained from traffic sensors or derived from travel demand models.

- **Microscopic approach** – It is well known that vehicle emissions are sensitive not only to vehicle speed, but also to vehicle modal operation (i.e., acceleration, deceleration, idling). This approach estimates emissions at high time resolution (e.g., second-by-second) so that the effect of different modal operations can be captured. To estimate total vehicle emissions at this level of detail, velocity profiles of every vehicle on the roadway are required. Although the estimation results would be highly accurate, such input data are very difficult to acquire in the real world. Therefore, this approach has so far been restricted to be used in conjunction with microscopic traffic simulation models, which can generate vehicle velocity profiles needed for this approach of emission estimation.
In this chapter, we have designed a new approach that falls between a mesoscopic approach and a microscopic approach by reconstructing modal components of vehicle trajectories for interrupted traffic flow. It is possible to reconstruct these modal components using data from state-of-the-art traffic sensors described in the next section.

4.3 Data Collection

New wireless vehicle detection systems are being deployed in a variety of roadways around the world [7]. These sensors are not only able to measure lane occupancy, flow, and speed, but can also match “vehicle signatures” between different sensors (i.e., vehicle re-identification) to provide overall travel time estimates of individual vehicles. With these travel-time data, it is possible to further extract the information on platoon patterns of the vehicles.

For our research experiments, we have used a wireless traffic sensor network that is installed along a primary arterial corridor in Chula Vista, California (Telegraph Canyon Road; see Fig. 4.1a). This network consists of 18 sensors in both directions located at the 9 signalized intersections, spaced approximately 500 meters apart, along the corridor. Again, the upstream magnetic signature of a vehicle passing over a sensor can be matched to a downstream signature (see Fig. 4.1b), allowing for vehicle re-identification and a good estimate of travel time.

Approximately 70 percent of the total vehicles traveling over consecutive intersections in the lane(s) that have the sensors installed can be re-identified, providing an accurate travel time and also an absolute time stamp. A 100% match rate is not possible due to lane changing and vehicle ingress/egress patterns along the
corridor. The lane occupancy when vehicle crosses over each sensor can also be measured for use in estimating a spot speed for each vehicle, which can be aggregated to result in average traffic speed. Example travel time data are shown in Fig. 4.2, where the x-axis corresponds to different time stamps, and the y-axis corresponds to the travel time. The green vertical lines represent the re-identified vehicles with their corresponding travel times (height); yellow lines correspond to detected but unmatched vehicles.

Over a period of time, it is possible to create a travel time histogram of traffic between signalized intersections, as shown in Fig. 4.3. It is apparent that there is a wide range of travel times between two neighbouring intersections (in this example, intersections 5 and 6 along the study corridor). This wide range of travel times is due to a variety of factors, including traffic signal phase and timing (i.e., whether the vehicles are stopped by a red light) and driver behaviour.
Fig. 4.1 Raw data from a travel time detection site

Fig. 4.2 Chula Vista Wireless Sensor Network
Fig. 4. Travel time histogram with ground truth. Blue bins are stopped vehicles and green bins are free flow vehicles.
In order to determine actual vehicle movements along with traffic signal phase and timing information, a large-scale ground truth study was carried out using video cameras placed along the study corridor. Video imagery of the vehicle movements was captured simultaneously with traffic data from the wireless sensors for a wide range of traffic conditions. Given these time-synchronized data sets, traffic scenes in each traffic light cycle as well as the travel time data of each vehicle could be extracted. Vehicles traveling together in one cycle were grouped as one platoon. Meanwhile, whether a vehicle was stopped or significantly delayed by traffic lights was manually checked from the videos across different levels of traffic congestion. It was then possible to validate a variety of details associated with the wireless traffic sensor dataset. For example, the travel time histogram shown in Fig. 4.3 can be segmented into different groups. Green bins represent travel times of non-stop vehicles, while the blue bins belong to the stopped vehicles. Group 1 (green bins) vehicles were confirmed that passed through the intersection without stopping at the red light, Group 2 (blue bins) corresponds to the vehicles that were stopped for one red cycle, and Group 3 (red bins) corresponds to the vehicles that were stopped for two red cycles.

Using the Modified Gaussian Mixture Model introduced in chapter 3, we can now assign a probability for each intersection pair on how many stops a vehicle typically makes, based on their travel time. Furthermore, the location of each vehicle in the queue can be identified through the instantaneous travel time data. These data are critical in reconstructing an approximate vehicle trajectory, described in the next section.
4.4 Methodology

4.4.1 Overview

Using the ground truth derived data shown in Fig. 4.3, several observations can be made. For the Group 1 vehicles that are able to travel through the intersection without stopping, their travel times are simply a function of their average speed which we model as a Gaussian distribution. For the Group 2 vehicles, their travel times depend highly on the delay due to getting stopped by the red light and by the queue dispersion. The travel time is less dependent on the free flow speed prior to decelerating due to a red light; what is more important is the arrival time of the vehicle during the red cycle time. As expected, the vehicles at the head of the queue (which arrive at the intersection earlier) would wait for the green cycle longer than the vehicles arriving later in the cycle.

In our approach, matched vehicles are classified into stop and no-stop based on their travel time. The matched vehicles are only 70% percent out of all detected vehicles. In order to estimate energy/emission of overall traffic flow including the rest of unmatched vehicles, the movement of these vehicles should also be determined. Traffic signal breaks the traffic flow by each of the signal cycle. One assumption we can make is that the travel time of one vehicle should be constrained by the vehicles in front and behind it within the same cycle while the movement of vehicles in different cycles are independent. If we can group vehicles from the same cycle, we can approximate the travel time of unmatched vehicles so the same classification scheme can be applied. However, the timing of signal is usually unknown in most of the existing traffic control system, especially with those adaptive or actuated.
controlling strategies. Therefore we developed a clustering method to group the vehicle with close downstream detection time, which is called platoon extraction. The extracted platoon doesn’t necessarily include all the vehicles within that cycle, because vehicles in the queue and vehicles arrive at the intersection after the queue has been discharged might be separated. However, it won’t affect the final trajectory reconstruction since the late arrival vehicles are usually at free flow speed so that none of them requires trajectory reconstruction. The platoon extraction also helps to determine the position of each vehicle in the queue which is necessary to the trajectory reconstruction.

4.4.2 Platoon Extraction and Travel Time Interpolation

The raw data from travel time measurement system is a sequence of data tuples: \( D_i = (t^{u}_{i}, t^{d}_{i}, tt_{i}) \), where \( t^{u}_{i} \) and \( t^{d}_{i} \) are the detection time at upstream and downstream, \( tt_{i} \) is the travel time for matched vehicles only. An example of raw data of three consecutive platoons are shown in Fig. 4.4a, x-axis is the time at downstream and y-axis is travel time. The unmatched vehicles (red vertical lines) with no travel time measurement in raw data are set default 1 second travel time for visualization.

A. Platoon Extraction

Vehicles can only be matched if they pass both upstream and downstream sensors. So they must belong to the platoon that passes through the intersection. During the queue dispersion, the density in queue area is still large. Vehicles left from the queue will form a platoon with small headways. On the other hand, unmatched vehicles can be ones in the platoon or left or right turn vehicles from perpendicular
roadways. In most of the traffic control systems, there is a clearing time gap between the left turn signal and go-through signal. This gap should appear in the downstream time lines as a larger headway between individual unmatched vehicles and a group of vehicles containing matched vehicles. Based on this observation, empirical headway thresholds can be defined to group vehicles within a platoon with left turning vehicles separated. An example of resulting platoons are shown in Fig 4.4b from the raw measurements in Fig 4.4a. The vertical red lines connects all the vehicles in each single platoon.

**B. Travel Time Interpolation**

The travel time of a vehicle in a queen is usually constrained by the ones in front and behind it. It usually has longer waiting time than the one in front and shorter waiting time than the one after. The accurate travel time depends on the actual arrival time at upstream, but is known for the unmatched vehicles. The travel time of the matched vehicles at two ends of the unmatched vehicles provide a bound to the unknown travel times. Follow this intuition, we make pseudo travel times for unmatched vehicles using linear interpolation. An example of interpolation is shown in Fig 4.4c, where the red vertical lines are higher than the previous default height. The new height indicates the new interpolated travel time. Although the interpolated travel time is not the actual travel time, in this context, the purpose of interpolation of travel time is only for stop/non-stop classification and therefore the interpolation already guarantee the results.
4.4.3 Stop/Non-stop Classification

After interpolation, every vehicle in a platoon contains travel time which can be used as input to the mGMM classifier. The traffic states in peak hour and non-peak hour are different, so is the travel time distributions. Different mGMM should be trained using travel time data in these two time periods separately. About details of mGMM training and classification, the readers can refer Chapter 3 Section 3.2 and Section 3.3.3. An example of platoon plot with classification results is shown in Fig 4.4d. The letter “s” showing on top of vertical lines indicates “stopped vehicle” and “f” indicates “free flow vehicle”.

4.4.4 Trajectory Reconstruction

As described earlier, energy and emission estimates that are based solely on average speed along an arterial signalized link may not be accurate, particularly when vehicles are stopped by the traffic lights. To improve the estimate, we propose a modal decomposition method that divides the typical trajectory of a vehicle for a single signalized link into different modes. These modes include acceleration, deceleration, idle, and cruise components. Fig. 4.5 shows a typical trajectory representation of a vehicle that is stopped on an arterial link, illustrating the different modes. There is typically an acceleration event (t_acc1) as the vehicle departs from the previous traffic light, passing the upstream sensor. This is followed by a cruise event (t_c1), traveling at free-flow speed \( V_{ff} \). The vehicle maintains this speed until it reaches the next traffic signal or queue of traffic. There is a deceleration mode (t_dec), followed by an idle mode (t_idle) until the traffic signal turns green or the queue disperses. The vehicle then accelerates (t_acc2) followed by a second cruise (t_c2).
(a) Raw measurement, green matched, red unmatched

(b) Platoon Extraction, red horizontal bar covers each platoon

(c) Travel Time Interpolation, unmatched (red vertical lines) contains interpolated travel time

(d) Classification, “s”-stopped, “f”-free flow

Fig. 4 Platoon plot in pre-processing for trajectory reconstruction
The representation shown in Fig. 4.5 is a maximum model of the trajectory, we will refer to as *scenario-I*. A slight modification of this *scenario-I* trajectory called *scenario-II* is when a vehicle is already at the free-flow speed when it passes the upstream sensor. As a result, the initial $t_{\text{acc1}}$ mode doesn’t exist in this scenario. A third scenario (*scenario-III*) is characterized by the vehicle at the head of the queue and may not reach the free flow speed when it passes the downstream sensor; therefore, the second cruise event $t_{\text{c2}}$ does not exist. In the last case, *scenario-IV*, the vehicles passes the previous intersection at a speed of and does not reach after the stop, so that this trajectory has neither $t_{\text{acc1}}$ nor $t_{\text{c2}}$ modes.

Each of these trajectories can be parameterized with data from the wireless traffic sensing network. Each roadway sensor provides an occupancy time for each vehicle passing over it, and assuming an average length of a vehicle, an instantaneous velocity can be estimated at both the upstream and downstream sensors given by:

$$
\begin{align*}
V_0 &= \frac{L_v}{\partial T_{\text{up}}} \\
V_1 &= \frac{L_v}{\partial T_{\text{down}}}
\end{align*}
$$

(2)
where $L_v$ is the average length of a vehicle, and $OT$ is the occupancy time of the vehicle traveling over the sensor. These two instantaneous velocities are used to determine which trajectory scenario to apply. For example, if the upstream velocity estimate is higher than $V_{ff}$ and the downstream velocity estimate is lower than $V_{ff}$, then scenario-IV applies. Assuming that $V_{ff}$ is known (e.g., based on speed limit information), as well as assuming typical values for vehicle acceleration and deceleration, then $t_{acc1}$, $t_{dec}$, $t_{acc2}$ can first be derived from eqns. (3 - 5).

$$t_{acc1} = \begin{cases} 
0, & V_0 \geq V_{ff}, \text{Scenario II, IV} \\
\frac{V_{ff} - V_0}{acc}, & V_0 < V_{ff}, \text{Scenario I, III} 
\end{cases}$$

(3)

$$t_{dec} = \frac{V_{ff}}{dec}$$

(4)

$$t_{acc2} = \frac{V_1}{acc}$$

(5)

The durations of the remaining modes $t_{c1}$, $t_{idle}$, and $t_{c2}$ depend on the location of the vehicle when it stops, which can be determined from platoon information derived from the wireless traffic sensor network. Assuming an effective space of a queued vehicle is $L_{qv}$, the distance of the vehicle to the stop bar $D_q$ can be derived from eqn. (6) [103], given information on the location of the vehicle in the platoon (see Section III).

$$D_q = L_{qv} \times \text{count}(i)$$

(6)
where \( \text{count}(i) \) is the count of vehicle \( i \) in the queue. The area \( A_1 \) illustrated in Fig. 4.5 represents the distance from the upstream sensor to the stopped position of vehicle, \( D_1 \); similarly, the area \( A_2 \) should be equal to the distance from the stopped position to the downstream sensor, \( D_2 \). Since the locations of sensors are known as well as the location of stop bar, \( D_2 \) could be derived first using \( D_q \), and then \( D_1 = D_{\text{link}} - D_2 \), where \( D_{\text{link}} \) is the total distance of the link between the upstream and downstream sensors. With this information, all of the mode durations can now be derived using eqns. (7 - 9).

\[
t_{c1} = \frac{D_1 - \frac{V_0 + V_f L}{2} - t_{\text{acc}} - \frac{1}{2} t_{\text{dec}} - t_{\text{dec}}^2}{V_{ff}} \quad (7)
\]

\[
t_{c2} = \begin{cases} 
0, & V_1 < V_{ff}, \text{Scenario II, IV} \\
\frac{D_2 - \frac{1}{2} a_{\text{acc}} + t_{\text{acc}}}{V_{ff}}, & V_1 \geq V_{ff}, \text{Scenario I, III} 
\end{cases} \quad (8)
\]

\[
t_{\text{idle}} = T_T - t_{\text{acc}} - t_{c1} - t_{\text{dec}} - t_{\text{acc}} - t_{c2} \quad (9)
\]

**4.4.5 Fuel/Emissions Determination**

To calculate vehicle fuel consumption and emissions, we use a micro-scale emissions model (CMEM, see [14]) which requires second-by-second velocity as input. For a group-1 (non-stopped) vehicle, a smooth velocity trajectory is used centred around its average speed. For the group-2 vehicles (i.e., vehicles that are stopped by the traffic light), the velocity trajectory is constructed using the particular modes described in the previous section. Given the wireless traffic sensor-based travel time, instant speed, and vehicle count within the platoon, as well as using calibrated constants for average free flow speed, acceleration, and deceleration, the scenario
type and time length of each mode is estimated using the equations in Section IV. This approximated second-by-second velocity trajectory is then run through the microscopic emissions model to determine fuel consumption and emissions.

4.5 Experimental Results

To verify the performance of the proposed methodology, a variety of experiments have been carried out from field testing in Chula Vista California. Validation of the travel time histogram threshold selection method is first described, followed by the fuel consumption/emissions estimation.

4.5.1 Stop/Non-stop Classification Validation

As proposed in Section IV, we classify stopped and unstopped vehicles based on mGMM. The data used in this section was collected between 10am to 5pm from the test-bed corridor (link 5→6) with a total of 1344 matched vehicles. We divided the dataset into two parts: 10am-3pm as non-peak hour dataset and 3pm to 5pm as peak hour dataset. Two different mGMM are trained and used for classification separately on these two datasets.

We have verified the classification results with ground truth data recorded on video. The histogram of ground truth stop/non-stop vehicles are shown in Fig. 4.5 with trained mGMM model (red curves) and ground truth distribution (black curves). Based on this validation exercise, it was found that 88% of the vehicles were classified correctly in non-peak hour and 92% in peak hour.
4.5.2 Modal Decomposition for Fuel/Emission Estimation

In order to test the validity of the modal decomposition method, a probe vehicle was used extensively on two separate days on Telegraph Rd. in Chula Vista California (July 27th and Oct 19th, 2010). The probe vehicle was equipped with a GPS datalogger reporting second-by-second velocity along with position information. Given time and location information, the probe vehicle profiles were matched to the wireless traffic sensor data. A total of 58 second-by-second speed trajectories of the vehicle were extracted and used in this analysis, including 28 one-stop and 30 non-stop trajectories. Based on the 58 trajectories, the average free flow speed was set to be 50 mph, typical acceleration is at 2mph/sec, and deceleration is approximately 3mph/sec. The distance of the link 5→6 is 0.78 miles and the effective queuing vehicle length is set to be 5 meters.

An example trajectory of a group-1 unstopped trajectory is illustrated in Fig. 4.6a shows the actual vehicle trajectory in green, and the red line illustrates the corresponding trajectory estimate. Fig. 4.6b (typical passenger vehicle) show the fuel consumption and CO2 emissions for three cases: 1) the estimation using the standard speed-based emission factor approach (see Section II); 2) the estimation using the modal decomposition method; and 3) the ground truth energy/emissions based on the actual second-by-second trajectory of probe vehicle. For this example freeflow trajectory, it can be seen that case 1) and 2) are the same, and slightly underestimate actual energy/emissions due to not capturing small acceleration/deceleration perturbations around the average speed.
Another example trajectory is shown in Fig. 4.7, where the trajectory includes a stop. Using the same cases as before, it can be seen that the modal...
decomposition approach is approximately 93% of the ground truth, compared to the 68% estimate using the standard speed-emissions approach.

It is possible to compute and compare the total sum of all trajectories; this provides a general sense on how well the method performs. Fig. 4.8 illustrates these overall results, indicating that the modal decomposition method is approximately 8% less than the true energy/emissions whereas the standard speed-emissions approach underestimate the energy/emissions by nearly 40%.

4.6 Summary

Accurately estimating fuel consumption and tailpipe emissions from vehicles traveling on arterial corridors is difficult since the standard methods typically do not take into account differences between when a vehicle flows through the corridor without stopping at traffic lights, and when the vehicle is stopped at one or more red lights. In order to better estimate energy/emissions in these cases, we take advantage of new wireless traffic sensors that not only measure traffic speed, density, and flow, but also can perform vehicle re-identification to get an accurate link-to-link travel times. Furthermore, the sensor data also provide information on vehicle platoons, such as platoon length and where a particular vehicle is within a platoon. All of this information can be used in estimating an approximate vehicle trajectory that is more realistic than a simple average speed-based estimate. The acceleration and deceleration behavior and idle duration during one stop can be estimated, thereby better estimating energy and emissions associated with the corridor. Results show that this new method is typically within 10% of the true values, compared to the standard approach which falls within 40% of the actual values. Future work includes
improving the acceleration and deceleration properties of the reconstructed trajectory and extending this method to multi-stop trajectories.

Fig. 4. Example free flow vehicle trajectory; a) actual trajectory (green) and estimate (red); b) energy/emissions for passenger vehicle.
Fig. 4. Example delayed vehicle trajectory: a) actual trajectory (green) and estimate (red); b) energy/emissions for passenger vehicle.
Fig. 4. 9 Total energy/emissions estimates for all trajectories for passenger vehicle.
Chapter 5. Vehicle Trajectory

Extraction using 3D LiDAR

5.1 Overview

High quality vehicle trajectory data play an increasingly important role in modern traffic research and management, improving vehicle dynamic models and car following models, calibrating microscopic simulation software, and even providing real time traffic activities. Whether collecting data for research or for traffic management, utilizing probe vehicles can cost a huge amount of resources. Further, providing only second-by-second ego-motion trajectories from probe vehicles with positioning devices will hardly satisfy the high demand of trajectory data from traffic researchers and administrators. If the probe vehicles can carry environmental perception sensors with appropriate algorithms of detecting and tracking surrounding vehicles, the efficiency of trajectory collection from probe vehicles can be greatly improved. Taking advantage of the growth of robotic sensing technologies, a variety of sensors can be placed onto probe vehicles according to recent literatures in order to extend its trajectories collecting ability.

Among all onboard perception sensors for vehicle trajectory collecting, LiDAR is one of the most popular due to its high accuracy on both distance and angular measurement, wide coverage, and high frequency. In the last decade, traffic researchers have tried different types of LiDAR and different sensor mounting
methods for surrounding vehicle trajectory collecting. In today’s market, three major types of LiDAR can be found specifically for traffic applications: 2D, multilayer 2D and 3D. The 3D LiDAR is the most powerful member in this family and has been applied successfully in autonomous driving, however it hasn’t been readily utilized in the trajectory collection area. The main contribution of this chapter is applying the 3D LiDAR and specific algorithms on vehicle detection and tracking to show the revolutionary capability of trajectory collection on the LiDAR equipped probe vehicle.

The trajectory collection technology on probe vehicle depends on two parts: ego-positioning and detection and tracking of moving objects (DATMO). Since the trajectories collected from the road will be used for vehicle interactive activity analysis, the positions in the trajectories are required to contain absolute coordinates, which requires high accuracy positioning ability of probe vehicles. A centimeter-level positioning technology has been published in previous research [3]. We incorporate this technology without presenting those details, refer to [3]. The DATMO is the major focus of this chapter. The detection and tracking framework shown in most of the previous literature can be considered to follow roughly the same framework, with the approach shown in Fig. 5.1. This approach incorporates feature extraction and then data association with the tracker. Although vehicles can be tracked accurately under good sensing conditions (no partially occlusion, features are clearly detectable), disadvantage of splitting and merging of the parts belonging to the same object has been pointed out in [104] where a more complex data association algorithm is proposed. Recently in an autonomous driving competition, a model-based tracking approach was first proposed in [105], shown as the lower pipeline in Fig. 5.1. The new approach makes a geometric assumption on the shape of the tracked vehicles and
updates the tracker directly with the raw measurements. Since the relationship between raw measurements and vehicle state is highly nonlinear, a non-parametric filter is suggested for tracking in this approach. The traditional and model-based approach have their pros and cons in DATMO, the former relies on a good feature detector and high quality data association while the later requires a suitable geometric model for the tracked objects. Since the shape of most vehicles running in the real traffic have a rectangular-like shape in 2D view (i.e., tracking is a 2D problem even using 3D data), we decided to use model-based approach in our vehicle trajectory collection application.

We organized this chapter as follows: a system overview along with hardware and software setup is introduced in Section 5.2; Section 5.3 presents the core detection and tracking algorithms; The experimental results in two datasets are shown in Section 5.4. Section 5.5 provides conclusions and future work.

Fig. 5.1 Two alternative approaches for detection and tracking for moving objects. Both of them start from raw measurement data from sensor on left and produce tracks of moving objects on right side.
5.2 System Overview and Data Representation

In this section, all the software and hardware tools, terminologies, data structures and object models are introduced prior to the presentation of the main processing algorithms in the next section. An overview of the vehicle detection and tracking system can be found in Fig. 5.2.

![System Diagram](image-url)
5.2.1 Sensor Setup and Software Tools

For tracking purposes, only the relative position between sensor and tracked objects will be sufficient. However, in real traffic scenarios, the probe vehicle (ego-vehicle) is required to follow the traffic flow in the same lane to capture the realistic local traffic activity. Tracking moving objects relative to a moving sensor without knowing the sensor motion can drastically degrade the convergence of the tracking filter, since it is difficult to find an effective motion model to predict the relative motion between sensor and objects, both of which are moving. As mentioned in Section 1, vehicle trajectories collected by the probe vehicle will be used for interactive activity analysis, in which the absolute positions of tracked vehicles should be recovered from relative positions. For these reasons, an accurate position estimation system is incorporated with our tracking system to produce accurate trajectories of surrounding vehicles. A typical instrumentation platform for accurate positioning is installed on top of the probe vehicle, as shown in Fig. 5.3. The IMU provides high frequency (200Hz) inertial measurements (acceleration and angular rates) along three perpendicular axes and the DGPS unit provides raw dual-frequency GPS pseudo-range, Doppler and carrier-phase measurements. Our in-house aided-INS software integrates these measurements to estimate platform states at 20Hz, and centimeter-level position accuracy can be achieved as presented in previous research [3]. The Velodyne 64HL 3D LiDAR on the lower panel is our perception sensor with 64 lasers laid vertically, spinning at 10 rounds per second and generating 100Mb data per second. Each laser beam returns the distance and intensity of the obstacle hit within a maximum range (80m). In our application, only the distance data will be used and the raw data will be reduced to an approximate and compact data structure.
to drop redundant information and get faster processing speed. Note that the GPS antenna has to be installed above the lidar to avoid the inference from laser beams, which causes the four supporting stickers consistently block four certain angles of lidar view. This mounting flaw causes partial occlusion of tracked vehicles frequently, but our tracking method can effectively overcome this issue.

5.2.2 Coordinate System

Different referenced coordinate frames are used by the positioning system and tracking system. To make the discussion clear, we introduce these frames in this subsection and explain when the data is transformed between frames.

A. The navigation frame is denoted by \( \{N\} \). This is also known as local tangent frame with axes pointing to North, East and Down directions. The origin
is selected as a fixed location convenient for analysis. In our case, it is the DGPS base station antenna.

B. The body frame of vehicle at time $t$ is denoted by $\{B_t\}$ with its origin set at the GPS antenna’s phase-center. The three orthogonal axes are defined along the direction of forward, right and down w.r.t. the vehicle body.

C. The sensor frame of Velodyne 3D Lidar at time $t$ is denoted by $\{L_t\}$ with its origin at the spinning center of the device and axes along right-forward-up directions. All the raw measurements are represented in this reference frame.

The body frame $\{B_t\}$ is the reference frame that our GPS receiver and IMU rigidly attached. The pose estimation of the vehicle output from AidedINS at each time frame can form the transformation from $\{B_t\}$ to navigation frame $\{N\}$. And the origin of sensor frame $\{L_t\}$ is roughly approximated as 15 centimeter offset down from origin of body frame along its down axis. Since we are intending to estimate absolute position trajectories of tracked vehicles, the pose of the tracked vehicle is better represented in the navigation frame $\{N\}$. For convenience, all the data from the lidar will be first transformed into $\{B_t\}$ so the data representations we discussed in next subsection are all in $\{B_t\}$. The height map used for segmentation requires absolute position of each point measurement, the transformations from $\{B_t\}$ to $\{N\}$ are taken in that step. For detection, data is represented in body frames but needs to be transformed from $\{B_t\}$ at sequential time stamps $\{B_{t-1}\}, \{B_{t+1}\}$ for motion evidencing. In the tracking process, estimated vehicle states are kept in $\{N\}$ but transformed back to $\{B_t\}$ during updating in order to compare with measurements at each new time stamp.
5.2.3 Data Representation and Aggregation

The 3D LiDAR raw measurements represented as LiDAR data packet of tuples \([t, r, l, d, i]\), where \(t\) is timestamp, the \(r\) is the rotating angle of laser base, not the exact rotating angle of the laser beam, \(l\) is the index of laser from 1 to 64, \(d\) and \(i\) are distance and intensity measurement. By applying the calibration procedure of the Velodyne user manual, the angle and distance of reflection on each laser beam can be corrected and the raw packet data can be transformed to \([t, P]\), where \(P\) is point of reflection in 3D Cartesian coordinates w.r.t. the LiDAR Frame \(\{L_t\}\). We call the set of points as a \textit{point cloud}. Since intensity is not useful in our vehicle tracking application, it will not be utilized. As mentioned in the previous section, all the coordinate frames are available at any time, the resulting points cloud can be transformed between any other frame.

The \textit{point cloud} representing all the reflections around the ego-vehicle is, however, not convenient to access if we want to apply most of the existing vehicle detection and tracking algorithms for the range sensors. The representation widely used for 2D LiDAR is the discrete polar coordinate system, where the measurements are formed as an array of rays, each ray corresponds to an angle and distance. Under the assumption that vehicles are running on the ground plane, the vehicle tracking can finally be considered as a 2D application, so we choose to follow the similar representation as the 2D LiDAR: transform the points cloud in Cartesian coordinates into spherical coordinates. We call the data within one revolution (from 0 degree to 360 degree) as a single \textit{Lidar Frame}. 


Before the transformation, there is one more problem to be solved: the origin of each LiDAR Frame. In one LiDAR Frame, points are from multiple lidar packets which are sampled from different origins due to the ego motion of LiDAR. A common coordinate frame must be selected for all the points within one revolution. As an approximation, we link each LiDAR packet to the body reference frame \( \{B_t\} \) with the closest time, and use the median \( \{B_{t_{mid}}\} \) as one among all \( \{B_t\} \) linked with packets as the reference frame for this LiDAR Frame. Then all the points in one LiDAR Frame are transformed to \( \{B_{t_{mid}}\} \). This data sampling timing is illustrated in Fig. 5.4.

The discrete spherical coordinate system is represented as a 2D table (virtual table, shown in Fig. 5.5), and each grid of which represent an approximate distance measurement within certain azimuth and altitude grid in spherical coordinates. When multiple points fall into a single grid, the median distance is calculated as an aggregation. The important parameters for the virtual table are azimuth and altitude resolutions, thus we pick 0.5 degrees for both of them. The effect of this data reduction is shown in Fig. 5.6.

![Fig. 5.4 The sampling timing of ego-state estimation (blue lines) and LiDAR scanning (red lines represent start (0°) of each Lidar scanning).](image)
While the *virtual table* is extremely useful for the ground plane detection (section 3.1), we will only need a 2D data representation to apply the detection and tracking algorithms as mentioned before. From 3D to 2D, we simply collapse the *virtual table* by picking the smallest distance in each column, which is a reduction on the altitude dimension. An array is formed to hold the results, representing the discrete polar coordinate system. We call this form *virtual scan* and each element in the array represents a ray in a certain polar angle (azimuth). The angular resolution of *virtual scan* is the same as the azimuth resolution of *virtual table*. This procedure is shown in Fig. 5.5 and an example of data reduction effect is shown in Fig. 5.7.

![Diagram](https://via.placeholder.com/150)

**Fig. 5.5** Virtual table holds the 3D LiDAR measurements. Virtual Scan holds the 2D data of virtual rays reduced from Virtual Table.
Fig. 5. 7 The left image is the raw LiDAR frame from states aggregation. The right image is the same data represented in virtual table. The points cloud are denser in the raw data but the virtual table still retain the major features.

Fig. 5. 6 The left image is the virtual table after ground points and small connected components filtered out. The right image is the 2D virtual scan reduced from the virtual table on left. Through this transformation, each connected component in 3D virtual table representation only keeps the points on the contour that is facing to the LiDAR origin. Three vehicles (two on left side and one right behind) show “L” and “I” shapes in virtual scan.
5.3 Methodology: Segmentation, Detection and Tracking

In this section, we present our proposed pipeline of our trajectory collection system as shown in Fig. 5.2. With the data structure and reference frames being switched, segmentation, detection and tracking are applied sequentially on the data stream. We propose two different segmentation methods in subsection 3.1 as alternatives for different efficiency and storage requirements. Detection and tracking are described in subsections 3.2 and 3.3.

5.3.1 Object Segmentation

The 3D LiDAR produces an immense number of data points from the surrounding environment in each frame, which makes it capable for multiple applications. But in each particular application, the researchers only focus on the interesting parts of the data. In our vehicle tracking application, the interesting data components should be narrowed down to the objects above the ground but yet not too high (<2m in our case). So the ground elevation should be estimated first and then all the height data points can be evaluated. We introduce two different methods to segment the data points in different heights in the following subsections. We first introduce a ground elevation estimation method developed independently within two DARPA teams [106] for autonomous driving in 3.3.1 which is the basis of our first method, dynamic ground plane estimation in subsection 3.3.2. In 3.3.3, we introduce a grid map approach for object height segmentation. These two methods are both suitable for the data segmentation in real traffic. While the dynamic ground plane estimation requires less resource in computation and storage, it relies on the
assumption of a road surface without vertical variation. The height grid map, on the other hand, records the height information in each grid in all the whole area the probe vehicle and sensor covered which can cost a lot of space, but provides more compatibility on different road surface types.

5.3.1.1 Ground Points Detection

In autonomous driving applications, researchers have developed a fast and effective algorithm to classify the ground points from massive points ground [76, 77]. Like Fig. 5.8, if point A, B and C are the intersections of three rays with the same azimuth angle and the ground plane, the angle between AB and BC should be close to 180 degree. The assumption underneath is that the ground surface around the ego-vehicle is a plane, so the lasers scanning in the each azimuth angular degree, which is another a plane, should intersect the ground plane in a straight line. The laser measurements at the same azimuth angle are grids in one column in the virtual table as introduced in section 3.2. We can iterate through each column in virtual table and check the angle between each three consecutive points until the angle is smaller than a predefined angle threshold. Then the last point will be identified as the furthest ground plane before the laser hits the obstacle and the ground elevation is estimated by this point. Since the angle between two vectors can be fast calculated as dot product of the normalized vectors of these two, this method is popular in autonomous driving application which requires real-time processing. The main disadvantage of this method is that noise from laser measurements in the virtual table can be problematic. When we generate the virtual table, we first use an approximated origin and then aggregate the measurements within the same grid. This can lead to additional
noise which causes the angle check terminates earlier so that some ground points will be misclassified as obstacles. Another special case is when the ego-vehicle is running in a congested traffic environment, the lowest laser return with a certain azimuth angle might hit on the rear plane of a vehicle, then all the reflections from the rear plane might be counted as ground points.

![Diagram](image)

Fig.5. 8 three sequential laser readings. If they belong to ground plane, the dot product between AB and BC should be close to 1.

### 5.3.1.2 Ground Plane Estimation

In order to make the ground elevation more robust, we extend the ground point detection method to a ground plane estimation by fitting a ground plane using the ground points near the ego-vehicle. We can set the angle threshold high to make sure non-ground points will not be included. We apply principle component analysis (PCA) [99] and use the eigen-vector corresponding to the least eigen-value as the normal vector of the ground plane which minimizes the distance between all the data points to the plane. The obtained plane is the ground plane and the height of any point in the virtual table can be calculated as the distance from it to the plane along its normal vector. We do the same calculation in every frame. To overcome the congested traffic scenario described in last subsection, we will check the smallest eigen-value from PCA. It should be very small to prove all the nearby points are co-
planar, if it is larger than a threshold, we will decline the estimated ground plane in the current frame and keep using the previous result. The fitted ground plane is shown in Fig. 5.9. This method will work well if the vehicle is running in an environment where the surrounding ground surface is nearly a plane, for example on a freeway. But on some suburban roads or in a city like San Francisco, there is hilly terrain on which the elevation of surround ground plane cannot provide a reliable height filtering for the objects far from the vehicle. Therefore, we introduce a height grid map in the next subsection.

Fig. 5.9 Two example of ground plane filtering are shown. On the left, the vehicle is running on a very flat road surface, the elevation of fitted ground plane filters most ground points in the field of view. However, on the right, when the road surface appears to be a little hilly, large segments of ground points are failed to be filtered out. Furthermore, the terrace on the right road boundary are above road without being filtered out by the plane fitted from points around road center. Height grid map can do a better job in these cases.
5.3.3.3 Height Grid Map

Grid maps are widely used in probabilistic robotics to model the environment of the mobile robots. It represents the map of environment as evenly spaced field of some kind of variable that holds the important map information, e.g. random variable for occupancy [87]. In our application, we put the height information in the grid map in order to estimate the height of road surface in more complicated driving environment. In each LiDAR Frame, we first transform the point from body frame $\{B_i\}$ to the navigation frame $\{N\}$. The point fall into a certain grid based on its $(N, E)$ coordinates, and its third coordinate along down-direction is used as height. For more robust estimation, we always keep $k$ lowest height measurements in each grid and take the average for the inference. Once we generate the height grid map, we run through all the LiDAR data stream again and the height of points can be calculated with the height in its corresponding grid. Although moving objects like vehicles or pedestrians will cover some grids in a number of frames, the LiDAR will eventually hit the ground once those objects move away. For those vehicles parked on the roadside or at the intersection that didn’t move during the data collection period, their contour will be recognized as the height of map in that small area. This is certainly misclassification, but since they are stationary objects that we don’t need to track, it is not harmful to filter them out as ground points. Some fixed obstacles with low height on side of the road like brushes or curbs will also be misclassified as the ground points but not harmful to be filtered out as well. In other words, our goal is to find the height of the road surface where moving vehicles may appear. The result of height filtering is shown in Fig. 5.10.
Fig. 5. The vehicle is running on an urban road. The blue points are within 0.3 meters from the ground surface estimated by the height grid map. The green points are the points with height between 0.3m to 1.5m, which is the height range where most of vehicles can be observed. The terrace on the right road boundary is successfully classified as ground points. Only trees or walls are kept and can be filtered out by motion later.
One more point to be mentioned is the size of the map. In the indoor robotics application, the environment usually only consists of a small room so that the whole grid map can fit in computer memory. However, when dealing with an outdoor traffic data environment, the probe vehicle usually covers a huge area, e.g. multiple blocks in the city or several miles of freeway. To make the grid map applicable in practice, we create a two-layer grid map, shown in Fig. 5.11. A high-level map matrix covers the whole area and consists of a number of small low-level grid maps which only cover small range areas and are only generated when the sensor reach that area. The size to the low-level grid map is assigned to be twice of the sensor range. So at any time, the sensor on the ego-vehicle can only cover at most four small grid maps. This

Fig.5.11 Illustration of a two-layer grid map. The big matrix covers a large area while the small cells are real grid maps holding height information. Only the cells reached by the sensor along the vehicle trajectory (blue curves) actually take space (grey cells). All the other cells are never allocated. If the sensor is now at the right-bottom corner (blue triangle), only four cells it covered (red circle) need to be updated in memory.
means at any time, only four small grid maps needs to be held in the memory. As soon as the sensor reaches a new grid that is not in the memory, the corresponding small grid map is loaded or allocated if it doesn’t exist. Each cell in the new/loaded small grid map will be updated by the measurements until the sensor range leaves this area. The small grid map with newest height information will be dumped to the disk for further inference.

5.3.2 Vehicle Detection

Given the points within the right height range extracted, there are still several steps to find the suitable candidates for tracking. In autonomous driving applications, the detection criteria is loose, since any possible moving objects should be considered as obstacles to avoid potential collision. But in trajectory collection applications, the probe vehicle is required to be running on the road for a long time and providing high quality trajectories of surrounding vehicles. If too many misclassified vehicles are sent to track, the final trajectory results will be incorrect. So in this application, we built the detection module by multiple algorithms with high criteria so that the tracking starts only when we have high confidence in a candidate.

All the processing is applied on the virtual table and corresponding 2D virtual scan with ground and high measurements filtered out. The output of the detection module is the initialized rectangular vehicle models as shown in Fig. 5.12.

5.3.2.1 Clustering using connected components

The mid-height measurements consist of reflections from vehicles, pedestrians and buildings and roadside objects like trees and light poles. Most of buildings and road infrastructures are stationary and can be filtered out in the next sub-module
motion detection. But because of the motion of leaves caused by the wind, the
reflections from flickering trees will mislead the motion detector resulting in a false
positive. Furthermore, in practice, some reflections from the road surface still exist in
sporadic frames due to the error from ego-vehicle motion estimations or LiDAR
calibration, which also appear to be a misclassified as moving objects. In order to
focus on the real moving vehicles, these kinds of noise should be filtered out before
motion detection.

Measurements of flickering leaves and inaccurate ground reflections are both
sparse and within a small size. If we cluster all the points in one LiDAR Frame based
on the distance and only retain ones with a large enough size, we can filter the noise
out. Connected component clustering algorithms are a well-known and popular
method in computer vision to group adjacent pixels as components. The connectivity
of two pixels are defined as four-connected (with up, down, left, right neighbors) or
eight-connected (including diagonal neighbors). In our method, we apply the
connected component algorithm on the virtual table which is a 2D matrix, just like the
image. The connectivity is defined as any two points within a k-by-k neighborhood
and within a 3D distance threshold (we set k=3 and distance threshold=0.3m). A mask
is made to mask out any components containing less than 10 points and will subsequently be ignored.

As mentioned in section 2, vehicle tracking is a 2D application, therefore the virtual table is reduced to the virtual scan data structure with the connected component mask. In each column of virtual table, the point with smallest distance and masked by the connected component mask will be used as virtual measurement in a virtual scan. Meanwhile, all the connected components extracted in the same frame will also be collapsed to be 2D segments keeping only the points on the contour facing to the lidar origin. Let the point in \( i \)th row and \( j \)th column of virtual table to be \( P_{ij} \), the contour points for \( k \)th 2D segments in virtual scan is \( P_{jk} = \arg\min_j (\text{Distance}(P_{ij}), P_{ij} \in \text{Seg}_k) \), where \( \text{Distance}(\cdot) \) is denoted as the Cartesian distance from origin to this point. A set of segments extracted in frame \( t \) is created for further processing, noted as \( \{\text{Seg}_k\}_t \). The example of result has already been shown in Fig. 5.6.

### 5.3.2.2 Motion Detection

In the 2D view of LiDAR measurements, a vehicle appears to be similar to a small segment of wall or some other on roadside obstacles with rectangular or line shapes. The motion is the most significant feature to differentiate the moving objects from the background objects. If a rigid object is moving, some space occupied by it in the current frame must be cleared in the next frame, and some space cleared in the current frame must also be occupied in the next frame. In the virtual scan, the rays always end up at the first hit point, so the vehicle body usually blocks one side of itself when it is facing the other side to the LiDAR. It means in the virtual scan, only
one cleared/occupied space can be observed in each consecutive frames, as shown in Fig. 5.13. However, we can check motion of a segment in the current frame with both the previous virtual scan and the next virtual scan separately. If both verify the same motion for this segment, then this moving segment will be a validate candidate of a moving vehicle.

In order to quantify the motion, we define a new property for each segment, i.e., motion evidence. Each point in the segment is first transformed to previous/next body frame and an expected laser measurement is evaluated for this point at that frame. If the \( \text{ith} \) point in segment \( k \) at time \( t \) be \( P_{ki}^t \) has an expected measurement on \( \text{jth} \) laser \( \hat{L}_{j}^{t-1} \) at time \( t-1 \), the previous motion evidence is calculated as \( \text{prev}_{ki}^t = L_{j}^{t-1} - \hat{L}_{j}^{t-1} \), where \( L_{j}^{t-1} \) is the real measurement on \( \text{jth} \) laser known in virtual scan at time \( t-1 \). The next motion evidence is calculated in the same way \( \text{next}_{ki}^t = L_{j}^{t+1} - \hat{L}_{j}^{t+1} \).

The motion evidence is saved with each segment and utilized by vehicle model initializer in subsection 3.2.4. In Fig. 5.14, we can see the vehicle shows consistent motion evidence in previous and next frame, while the motion of roadside obstacles only exist in a single frame therefore will be filtered out.
5.3.2.3 Line Extraction and Adjustment

The last validation on the candidate of a vehicle is based on its shape. Since we extract the measurements between 0.3m to 1.5m, the parts in this range of most...
vehicles should appear to be one line or two perpendicular lines in 2D view. The well-known Iterative End-Point Fit (IEPF) [107] algorithm is an efficient and effective way to find lines within a sequential laser points. IEPF is a recursive algorithm, where a line is defined by the first and the last point of a given segment. Then, the point with the maximum distance to this line is detected. This point divides the segment in two intervals and the algorithm starts recursively again until the maximum distance is smaller than a certain threshold, which can be empirically determined. The result after line extraction is shown in Fig. 5.15 with vehicle models. We keep the segment with a single line. But for ones with two lines, we only keep the segments with two lines which form an angle close to perpendicular and within reasonable length. Segments with more than two lines are ignored.

If the contour of a vehicle in 2D view is a perfect rectangle, the lines extracted from IEPF can be directly used to initialize the rectangle vehicle model. However, a lot of passage cars contain curvy front boundaries which makes the extracted line(s) far away from the side of the real bounding rectangle. The line segments extracted by IEPF are just the links of two break-points. We adjust the raw line by applying Principle Component Analysis (PCA) on the subset of points between the two break-points. Given the 2D points, PCA produces two eigen-vectors and eigen-values as output. The eigen-vector with larger corresponding eigen-value is the adjusted line segment that minimizes the sum of normal distance from each point to it. When two lines exist in the segment, the two adjusted line segments are still not perpendicular to each other in most of time, we have to select one of them. Each time two perpendicular vectors from PCA from one subset of points, we make a rectangle based on these two vectors and the middle break-point of the whole segment (segment
with two lines extracted must contain three break-points), like Fig. 5.15. The rectangle with smaller sum of distance to all the points in the segment will be used for the vehicle model initialization.

### 5.3.2.4 Vehicle Model Initialization

Filtering out small connected components, stationary segments and bad shapes, the remaining segments can be recognized as measurements from the new detected vehicle. The vehicle model is described in the beginning of this section. The center, size, orientation and speed are necessary to define a vehicle model for tracking. Given the motion evidence and line segment(s) in each candidate, the front side of the vehicle must be determined which is critical to determine all the other properties of the model. In previous research, multiple vehicle models which fit on the segment with different possible orientations are initialized for tracking together in the framework of Interactive Multiple Model method. In congested traffic, when vehicles are quite close to each other, this method will make lots of false positive trajectories due to this interference.
Given a segment with one or two perpendicular line segments, the orientation of moving vehicle is hard to determined. Fortunately, the motion evidence we saved in previous step can help to solve this problem with initializing the moving direction at the same time. First of all, we have to assume the vehicle to be tracked is moving forward; not backward or crosswise. The motion evidence on a vehicle moving towards the lidar origin (getting closer) should show a positive motion in previous virtual scan and a negative motion in next virtual scan. The contrast motion should be shown for vehicles moving away from lidar origin (getting further). Motion evidence always points from lidar origin to the measured point so it is not the actual moving direction, but knowing the vehicle is moving further or closer is the most critical part to find front/rear line segment, then vehicle orientation and finally speed. Since the vehicle motion has been assumed to be forward, so that most of the motion evidence should be observed on front or rear line segment, not on side line segment. However, if the relative speed between measured vehicle and ego-vehicle is large and the angle between motion of measured point and laser ray is also large, motion evidence can be observed on side line segment, shown in Fig. 5.15 (b) (c). After observation, we found if the vehicle is moving further, the motion evidence on previous virtual scan should be checked, and next virtual scan for the ones moving closer. The line segment with more motion evidence is recognized as front/rear part of the vehicle depends on the relative between the vehicle and lidar. Once the front/rear is determined, the orientation should be found on its normal vector agreed with motion evidence. And the speed can be calculated as the motion evidence projected along the orientation.

The center and size the vehicle model is easy to be calculated since the line segments and orientation of the rectangular box have been determined. When the
The length of observed line segment is too small, a default vehicle width/length is used.

The results of vehicle model initialization is shown in Fig. 5.16.

Fig.5. 16 The line extraction and vehicle model initialization steps are demonstrated. Measurements in current frame and next frame are shown in white and blue points. Green line segment is line indicator connecting two end points. Cyan line segment is the adjusted line segment used for model initialization. In (b) and (d), the white box is the initialized vehicle model with an inside arrow representing its orientation. The yellow box is the predicted model in the next frame based on the speed estimated by the motion evidence. In (c), the moving objects are not initialized as a car because of the large angle between two lines, but it is quickly captured again after several frames and initialized in (d). As mentioned in the paper, the motion evidence are not guaranteed to show the real speed, the predicted model in (d) is a little behind the real measurements.

Fig.5. 17 A moving vehicle is detected by the motion evidence (red line segments) w.r.t the previous frame (a) and next frame (b) (both blue points in two figures). From the motion evidence, the motion of vehicle can be determined as moving away from the LiDAR origin. And the rear face can be differentiated from its side face by checking the motion on next frame. The opposite motion is detected in (c) and (d), the vehicle is moving closer and the front face can be determined in previous frame.
5.3.3 Tracking

There are two categories of measurement models used in LiDAR object tracking. The model found in most of the literature is a virtual measurement model, which extracts features or object states from raw LiDAR measurements and calculate the tracking residual between tracked states and those features. This provides a linear or slightly non-linear measurement model for filters. Although more efficient filters like Kalman filter or EKF can be applied in this case, stable and accurate features extraction in each frame to guarantee a high rate filter updating. This is very difficult in real traffic scenario where occlusions and vehicle-like environmental structures always exist. On the other hand, a new approach has been developed to interpret the raw measurements directly from geometric object model [105]. As shown in Fig. 5.18 (a), given the pose and shape in state and a virtual scan S, we can compute the likelihood as follows. We position the rectangular box defined by vehicle state in body frame and expect a set of laser measurements on this vehicle \{L\}. We compare this expected measurements with real measurements in S. Each ray in S is considered independent and the score of the comparison is first referred from cost function shown in Fig. 5.18 (b).
Fig. 5. 18 Vehicle geometric model (a) and the cost function of measurement (b) [2].
5.3.3.1 Particle Filter

A Particle filter, also known as Markov Chain Monte Carlo simulation, is a sampling solution for state tracking problems [108]. Let the state to be $x_t$, measurement to be $z_t$, the problem is to determine the posterior probability of state conditional on measurements $p(X_t | Z_t)$, where $X_t$ and $Z_t$ represent the states $x_k$ and measurements $z_k$ up to time $t$. Assuming an Markov process, Bayes’ rule can be applied on the posterior:

$$p(x_t | Z_t) = \frac{p(z_t | x_t) \cdot p(x_t | Z_{t-1})}{p(z_t | Z_{t-1})} \propto p(z_t | x_t) \cdot p(x_t | Z_{t-1})$$

$$p(x_t | Z_{t-1}) = \int p(x_t | x_{t-1}) p(x_{t-1} | Z_{t-1}) dx_{t-1}$$

Which yields a recursive solution. In general, the computations in the prediction and update cannot be carried out analytically, hence approximate methods such as Monte Carlo sampling is needed.

The basic idea about sequential Monte Carlo simulation is to represent the posterior distribution through a definite number of particles. Each particle represents a sample of state corresponding to its weight:

$$w_t^i \sim \frac{p(x_{1:t}^i | Z_t)}{q(x_{1:t}^i | Z_t)}$$

, where $x_{1:t}^i$ is $i$th particle representing states up to time $t$ and $q(\cdot)$ is the proposal distribution the samples are drawn. In order to find a recursive expression, it turns out to be:

$$w_t^i \sim w_{t-1}^i \cdot \frac{p(z_t | x_t^i) \cdot p(x_t^i | x_{t-1}^i)}{q(x_t | x_{t-1}^i, z_t)}$$
Assuming the proposal distribution to be state transition distribution \( p(x_t|x_{t-1}) \) and applying resampling every iteration, the update function of particles reduce to:

\[
\begin{align*}
x_t^i &= p(x_t|x_{t-1}) \\
w_t^i &\sim p(z_t|x_t^i)
\end{align*}
\]

This is known as sampling importance resampling algorithm and is applied in our approach.

### 5.3.3.2 Model-based Vehicle Tracking

In model-based vehicle tracking, a geometric model is assumed to accurately represent the state of tracked vehicle. Based on the rectangular box model, we define six dimensional state: \( X = (x, y, \varphi, v_t, w, l) \) as center \((x, y)\), orientation \((\varphi)\), speed \((v_t)\) along the orientation and shape (width, length). Due to the difficulty of shape observation, we use a fixed width and length for all the vehicles, only first 4 dimensions are updated in the tracker. We apply the SIR algorithm, so the particles are resampled during every updating cycle.

#### A. Propagation

A linear motion model is applied for the dynamic. Given a prior pose and velocity, the linear motion model assumes a constant velocity for the duration of each time interval from \( t-1 \) to \( t \). To model the dynamic on orientation, a perturbing orientation is added to the beginning of motion at \( t-1 \) and a final adjustment to orientation is made after the model moves forward at \( v \) along \( \theta \). Given the maximum value of initial
angular rate, final angular rate and acceleration, the probabilistic motion can be
sampled.

Let the position and orientation at time $t$ to be $x_t$ and speed to be $v_t$, the motion
propagation or state transition distribution can be decomposed as:

$$
p(x_t, v_t | S_{t-1}) = p(x_t | x_{t-1}, v_t, S_{t-1}) \cdot p(v_t | v_{t-1}, x_{t-1}, S_{t-1}) \cdot p(x_{t-1}, v_{t-1} | S_{t-1})
$$

In $p(v_t | v_{t-1})$, new speed in time $t$ is assigned to old speed plus a noise
sampled from $(a_{\min} \cdot \Delta t, a_{\max} \cdot \Delta t)$ and in $p(x_t | x_{t-1}, v_t)$ and initial and final
orientation is perturbed by a random angular rate sampled from $(\theta_{\min} \cdot \Delta t, \theta_{\max} \cdot \Delta t)$.

**B. Update**

The weights of particles are updated by likelihood of the new virtual scan
given previously. Since each particle is a sample of state that can form a vehicle
model hypothesis, given the vehicle’s pose and shape in state and a virtual scan $S_t$, we
can compute the likelihood as follows. We position the vehicle model (rectangular
box defined by vehicle state) in body frame and calculate a set of expected laser
measurements on this vehicle hypothesis

$$
\hat{L}_k = \{ \hat{l}_k^i \} = f(x_k)
$$

We compare this expected measurements with real measurements $L_k$. Each
ray in $S$ is considered independent, given a cost function on the difference of real and
expected laser measurements, the score of each particle is

113
\[ score_k = \prod \exp(-\text{cost}(t_k - l_k)) \]

The likelihood is the normalized score:

\[ p(z_t|x_t^k) = \frac{score_k}{\sum score_i} \]

5.3.3.3 Track Initialization

Since the threshold vehicle model initialization is already set high based on its motion and geometric features in the detection module, we only check the track confliction in track initialization. If the new detected vehicle has any parts enter the vicinity of a tracked vehicle model, it will be ignored. The pose of the tracked vehicle is referred by the average vehicle model of all the particles with more than \(k\) occupied measurements on it.

5.3.3.4 Track Termination

The termination of a track in particle filter is usually indicated by the effective variance of particles. However, a good threshold of effective variance is not easy to determine in practice since it is not straightforward to the human observation. We instead use a more heuristic termination criteria. If the number of trustable particles is lower than a percentage of total particles, the track is lost. If it is lost for \(k\) sequential frames, the tracking is terminated.

5.4. Experimental Results
We collected two datasets to test our approach, one on arterial road focusing on congested signalized intersection and big turning movement, the other on freeway focusing on high speed. The statistic of vehicles appeared, detected and consistently tracked is shown in table 5.1. An example of tracking results are shown in Fig. 5.19.

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Filtered</th>
<th>False Positive</th>
<th>Missing Vehicles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arterial</td>
<td>432</td>
<td>215</td>
<td>2</td>
<td>35</td>
</tr>
<tr>
<td>Freeway1</td>
<td>96</td>
<td>53</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>Freeway2</td>
<td>108</td>
<td>58</td>
<td>0</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 1. Qualities of trajectories collected by our system from one loop of arterial roads and twice on one section of freeway CA215N.

5.4.1 Urban Traffic

In urban traffic, congestion at signalized intersections is often encountered. When the probe vehicle is stopping at the intersection, all surrounding vehicles on the same direction also become stationary or pass by slowly. Due to the high density over this small area, partially occlusion frequently occurs on the tracked vehicles, which causes target lost. A good vehicle trajectory collecting system should keep tracking the vehicles in this scenario since the vehicle trajectories during stopping and queue dispersion are usually the most important part for traffic activity analysis.

In our experiment, we drove our probe vehicle on Iowa street and University Avenue in Riverside, CA which are two busy arterial roads at peak conditions. We took the experimental data at around 4pm when the traffic is congested.
In Fig. 5.19, our probe vehicles stops in the middle lane at an intersection. In the left figure, one vehicle in the right lane is coming up and is being tracked. Though its mid-body is occluded by a supporting post on the sensor platform (see Fig. 5.3), the vehicle model can fit the measurements very well. In middle figure, the front body of that vehicle was occluded by another supporting post which generates even some error measurements shown as blue points between ego-vehicle and the tracked one. These error measurements were within the bound area of the vehicle model therefore led to higher cost to the likelihood which prevented vehicle model moving forward until the front body of tracked vehicle showed up again. In the right figure, the tracked vehicle kept moving forward and the vehicle model caught up the correct position as soon as the front body was being measured again. This example shows the
tracking algorithm really robust to the partially occlusion and thus our trajectory collecting rarely gets interrupted during the congested intersection.

Turning movement is another potential cause for target lost since during the turning movement the measured parts of a tracked vehicle can change drastically. In Fig. 5.20, from left to right, the measured parts of vehicle changed from rear to side-rear and back to rear again, but the tracking algorithms worked very well.

### 5.4.2 Freeway

On freeway, the traffic flow is usually higher than on the arterial roads. We took experiment data on a section of CA215 north to verify the performance of the system under higher speed. The speed of our probe vehicle is at an average of 60mph. It is hard to show the performance in images, but the reader can refer our online video for the whole procedure.

### 5.4.3 Trajectory Processing

Although we have high quality detection and robust tracking, there are still bad and short trajectories in the output due to some poor measurements. We filter out the trajectories that are too short in time and distance. After filtering out trajectories shorter than 3 seconds and 5 meters, we have 50% trajectories left. The quality of these remaining trajectories are shown in table 1 based on our manual checking. False positive trajectories among the filtered are the ones tracked as vehicle but belong to road structures or the tracks left the initially tracked vehicles attaching to other measurements. We can see our high detection criteria and post processing control the number of this type of error very well. Missing vehicles are the vehicles in the field of view but failed to be detected or being detected too later so only a small portion of
trajectories were collected. This type of error is on the opposite of the false positive. If we set the bar of detection or post processing too high, some short but valid tracks are filtered out. Especially on high density traffic, vehicles frequently occlude others which cuts the real trajectories into several short parts. Overall, our system provides very robust and efficient results on the data set collected from real traffic.

Fig. 5. 20 Probe vehicle (while rectangular box) stopped at intersection. Vehicles on the same lane and left lane were also stopping while vehicles on right lane kept passing by. From left to middle and to right figures, the passing-by vehicle was detected and tracked, then stopped moving due to the occlusion on its front body, and finally got tracked back after the front body appeared again even though a large portion of body were still occluded. Trajectories are shown in green. Particles are shown in white points.
5.5 Summary

In this chapter, we proposed a new automatic platform for vehicle trajectory collection. Impressive results have been shown in both urban and freeway, free flow and congested traffic environments.

By applying the vehicle model, we obtained the tracked states not only with the vehicle position as the previous methods, but also the orientation. This allow us to take research on more detailed vehicle activities.

Although the current detection and tracking method shows an accurate states estimation for most of the vehicles with shapes close to average vehicle shape dimension, a more flexible and dynamic shape estimation should be included in this method. The shape should be adjusted in each frame when the vehicle is tracked, so
that big track or bus can be covered with models in their real dimensions which yields to more accurate state estimation.
Chapter 6 Conclusions and Future Work

The research developed in this dissertation spanned several different areas, from arterial travel time modelling to energy/emission estimation and finally to vehicle detection and tracking on mobile sensing platform. However, the main focus has certainly been to estimate detailed traffic activities more accurately. Different sensing technologies have been utilized in this dissertation for different level of traffic activity measurement. These sensing technologies have all been deployed in the current traffic monitoring system and will be more and more populated in the near future. Therefore new traffic estimation and modelling methodologies based on these cutting edge sensing system will become more practical. Some of these methodologies addressed in this dissertation are: 1) arterial travel time modelling using sparse probe vehicle data, 2) arterial energy/emission estimation using travel time measured by magnetic sensors, and 3) vehicle trajectory collection using 3D lidar and lane level positioning sensing platform. The topic aimed at a number of applications in the areas of ATIS, traffic management system, traffic model calibration, and even autonomous driving.

Section 6.1 provides the conclusions and Section 6.2 provides some of the future directions for the research.
6.1 Conclusions

An arterial travel time model has been proposed in Chapter 3 which models the travel time over a single signalized intersection based on Gaussian Mixture Model with additional constraint that is derived from travel time decomposition. This model keeps the advantage of compact representation as a parametric model. It can be calibrated by travel time data from either fixed-location travel time sensors or sparse mobile sensors. We showed that this model can provide a good fit on the real traffic data from very congested condition to light traffic condition. Furthermore, the mean and variance of free flow travel time and portion of free flow vehicles in the total flow can be easily inferred from this model, which implies great potential on more detailed traffic state estimation and traffic control.

Chapter 4 describes a novel approach to estimate energy/emission from arterial traffic by analysing the data collected by wireless magnetic sensors that can measure accurate travel time of vehicles matched by a pair of sensors. The collected data is first used to train our model proposed in Chapter 3. Then the model can be used as a classifier to differential the stopped vehicles and free flow vehicles given their travel time. Although only the travel time of vehicles matched by the sensors can be measured, the method assigns pseudo travel time to those unmatched vehicles based on their neighbouring matched vehicles in the same platoon. Following the classification, a trajectory reconstruction algorithm approximates the second-by-second speed of stopped vehicles based on their positions in the queue and predefined acceleration and deceleration. Finally, the approximated trajectories will be running through CMEM to calculate energy and emission. We showed in the results that the
energy/emission estimation can be improved by 40\% in the proposed approach from the traditional method that only uses the link average speed.

Finally, the accumulation of work on traffic activity estimation is complemented by a mobile sensor platform to collect high-quality trajectories of surrounding vehicles of a probe vehicle. The sensor platform consists of Differential GPS, IMU and Velodyen 3D LiDAR. Based on our previous work on lane-level positioning technology, the position of probe vehicle can be estimated at a centimetre-level accuracy and at a high rate (200 HZ). This accuracy and resolution helps the development of clear ground point segmentation and reliable vehicle detection algorithms. A geometric-model-based tracking pipeline is then applied to track the vehicles once they are detected without further data association. The tracked trajectories are refined and filtered in the post processing. Finally, examples of second-by-second lane-level vehicle trajectories are collected on both freeway and arterial roadways. Although the true trajectories of those tracked surrounding vehicles are unknown, from the visual checking, we found the detection and tracking algorithm is robust in terms of high detection rate, low false positive rate and long term tracking capability.

6.2 Future Work

While we have gotten a lot of great results from the research presented in this dissertation, there are still a lot more to be done to improve on the current research and to bring them into the application space. The following subsections offer some of the future directions and potential applications for the research addressed in this dissertation.
6.2.1 Arterial Route Travel Time Estimation

Despite of good fitting result and easy inference, our mGMM only models the travel time over a single intersection. For some applications like ATIS or route planning, the route travel time will be more useful than individual link travel time. In previous researches of road network routing, the link travel time is usually modelled as a deterministic variable, therefore the route travel time can be calculated as the sum of link travel times along the route. However, as a random variable, the link travel time brings new challenge to the calculation of route travel time. One possible solution is to assume a Markovian relationship (conditional independency) between the neighbouring links along the route so that the total travel time distribution can be solved recursively.

6.2.2 Apply Travel Time Model to Large-Scale Road Network

The major challenge of applying the current mGMM in a large-scale road network is the different data availability over the network. In the large-scale road network, there are always some links without a large number of travellers which implies the number of sample data will be too small to train the model reliably. However, the links are all connected with each other which indicates some dependency among the links within the same neighbourhood. Traffic state transmission models are worth to be studied to spread the information from links with enough number of samples to their neighbours with few samples.

6.2.3 Vehicle Classification using Wireless Magnetic Sensors

In our proposed energy/emission estimation method, all vehicles are considered as simple passenger vehicles when passed into CMEM. Although we can
improve that by calculating the energy/emission as all available vehicle categories in CMEM and weighted the results by the vehicle type distribution profile in a certain area, it is still worse than an actual measurement of vehicle type. The wireless magnetic sensor records the magnetic signature of every passage vehicle. The big repository of vehicle signatures is a potential training dataset for a classification algorithm. For supervised learning algorithm, the vehicle type in such training data must be labelled which is very time consuming work and it might not be easy to find a location that all categories of vehicles can be collected. An alternative way is to find a good feature space on the raw signature data so that an unsupervised learning algorithm can produce a good result.

6.2.4 Detection and Tracking using 3D LiDAR Data

An important potential aspect we didn’t make use of is the 3D dimension data, illustrated in Fig. 5.19. Since the 2D data are easier to process and 2D geometry is enough for inference of vehicle state, most of the existed methods only use the 2D data for applications like trajectory collection and autonomous driving. However, 3D data can provide more information. For example, when the vehicle is present on the side of the sensor, in 2D, it appears as a single line and sometimes partially blocked by the instrumental parts around the sensor, this will lead to uncertainty in the position estimation of the tracked vehicle like the aperture problem in other computer vision scenarios. If we can consider all the distance and intensity on the side the vehicle in 3D data, this problem can be solved by including features on the side face of the vehicle in the measurement model to improve likelihood evaluation. Several 3D methods introduce in Chapter 2 for vehicle detection are worth to be studied here.
6.2.5 Lane-Level Map for Trajectory Collection

In our proposed trajectory collection system, there are three components: positioning, detection and tracking, and map. The first two components have been accomplished, while the mapping is still under developing. With the current centimetre-level positioning platform, the probe vehicle can collect the central coordinates along all lanes by running in the middle of every lane for multiple times. This is very time-consuming. Cameras or intensity measurements of 3D LiDAR can be used here to extract multiple lane makers on the same road, so that the data collection and mapping procedure can be more efficient. Building a lane-level road map can also help the detection and tracking and the lane information of trajectories are necessary for some applications, e.g. lane change modelling. If a road map can be built, we can recede the detection criteria without worrying about the false alarm detections from roadside. Then a faster detection and longer trajectories can be collected.
Bibliography


[45] V. P. Sisiopiku, N. M. Rouphail, and A. Santiago, Analysis of correlation between arterial travel time and detector data from simulation and field studies, 1994.


