

Fusing Heterogeneous Traffic Data: Parsimonious Approaches using Data-Data Consistency

Qing Ou

This thesis is the result of a Ph.D. study carried out from September 2006 to September 2010 at the Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology. From 2006 to 2009, this research was sponsored by the *ICIS* Data fusion project, Technology Program of the Ministry of Economics Affairs Under project DCB.7814 *Reversing Traffic Jam* and *NWO* funded *VENI* project. From 2009 to 2010, this research was sponsored by the Next Generation Infrastructures Foundation.

Fusing Heterogeneous Traffic Data: Parsimonious Approaches using Data-Data Consistency

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College van Promoties,
in het openbaar te verdedigen op maandag 9 mei 2011 om 12:30 uur
door Qing OU
Master of Science in Engineering, University of Science and Technology of China
geboren te Zunyi, China.

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. S.P. Hoogendoorn

Copromotor: Dr. ir. J.W.C. van Lint

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. S.P. Hoogendoorn	Technische Universiteit Delft, promotor
Dr. ir. J.W.C. van Lint	Technische Universiteit Delft, copromotor
Prof. dr. R.L. Bertini	Portland State University
Prof. dr. R.E. Wilson	University of Southampton
Prof. dr. ir. J.H. van Schuppen	Technische Universiteit Delft
Prof. dr. ir. B. De Schutter	Technische Universiteit Delft
Dr. L. Mihaylova	Lancaster University

TRAIL Thesis Series T2011/5, the Netherlands TRAIL Research School.

TRAIL Research School
P.O. Box 5017
2600 GA Delft
The Netherlands
Telephone: +31 (0) 15 278 6046
Telefax: +31 (0) 15 278 4333
E-mail: info@rsTRAIL.nl

ISBN: 978-90-5584-143-1

Keywords: data fusion, traffic state estimation.

Copyright © 2011 by Qing Ou

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission of the author.

Printed in the Netherlands

To my wife and my parents: Ji Xia; Huanfen Yao & Mingxiang Ou

Preface

After four years of interesting and ill-organized work, my thesis is presented here. During these four years, I have learnt a lot about traffic data fusion. Also I was challenged by many people and organizations, and some of my potentials are exploited. I am really proud and satisfied with some of results. This is indeed a enjoyable period. In this preface, I would like to express my thanks to some relevant people.

First of all, I would like to thank all the committee members. I can imagine that all of you have suffered a lot when you were reading this thesis written in *Chinglish*. But I am glad that you all survive in reading this thesis. Following that, I want to thank Professor Hen van Zuylen, who hired me in China and gave me the opportunity to work as a PhD in Delft University of Technology.

Secondly, I would like to thank my promoter Professor Serge Hoogendoorn and co-promoter Hans van Lint. They gave me quite a lot of inspiration and careful guidance. They also grant me enough freedom so that I could spread my imagination freely the same time they keep me on the right track. I feel lucky to have such supervisors.

Thirdly, I should thank my colleagues such as Chris, Frank, Adam, Victor, etc. It is really fun to work and chat with them. I often get some inspiration from them.

Fourthly, I want to thank my family—my parents, my wife and my sister. They always encourage me and cook nice food for me. I have developed some important habits and personalities being with them.

Last but not least, I would like to thank all the friends who care about me and are sincere to me. I am very lucky to have such friends that teach me how to become a good person of a warm heart and strong will.

Sincerely yours,
-Qing Ou, 2011

Contents

1	Introduction	1
1.1	Research background	1
1.1.1	Context & background: the need for traffic data fusion methods .	1
1.1.2	Multi-sensor data fusion	3
1.2	Problem formulation	4
1.3	Research scope and objective	4
1.4	Theoretical and practical contributions	5
1.4.1	Scientific contributions	5
1.4.2	Methodological contributions	6
1.4.3	Practical contributions	7
1.5	Thesis Outline	8
2	Traffic data collection and importance of data fusion	9
2.1	Introduction	9
2.2	Definitions of basic traffic variables	9
2.2.1	generalized variables with Edie definition	10
2.2.2	Eulerian and Lagrangian measurements	11
2.2.3	Issues in computing traffic variables from available data sources .	15
2.3	Traffic data	15
2.3.1	Brief overview	15
2.3.2	Loop data	18
2.3.3	Travel time and trajectory data	21
2.3.4	Low-resolution positioning data	23
2.3.5	Summary	24
2.4	Conclusion	25
3	The state-of-the-art in traffic data fusion	27
3.1	Data fusion and traffic data fusion	27
3.1.1	Levels on data fusion	27
3.1.2	Traffic data fusion	29
3.2	Assimilation techniques in the first level of traffic data fusion	31

3.2.1	Kalman filters and its variations	31
3.2.2	Particle filters	37
3.2.3	Linear programming	39
3.2.4	Treiber filter	41
3.2.5	Nudging Technique	44
3.2.6	Summary	45
3.3	Discussion on further challenges	48
3.3.1	The spatio-temporal alignment problem	48
3.3.2	Measurability and parameter identification	50
3.3.3	Computation cost	50
3.4	Conclusion	51
4	Towards a new approach	53
4.1	From data-model consistency to data-data consistency	53
4.2	Benefits of the new approach	56
4.3	Methods based on this approach	57
5	Trajectory reconstruction by using travel time and local speed	59
5.1	Introduction	60
5.1.1	Analysis of fusing travel time and local traffic data	60
5.2	Methodology: PISCIT	64
5.2.1	Framework of PISCIT	64
5.2.2	Step one: reconstruction of individual vehicle trajectories	64
5.2.3	Step two: speed re-estimation	67
5.3	Validation	70
5.3.1	Experiment setup & data generation	70
5.3.2	Results	71
5.4	Conclusion and recommendations	74
6	General approach for speed reconstruction by fusing travel times	77
6.1	Introduction	79
6.1.1	Floating car data used for speed estimation	79
6.1.2	Challenge from floating car data	79
6.2	Theoretical background	81
6.2.1	Analysis of simple cases	82
6.2.2	Analysis of more complicated cases	84
6.2.3	Implied difficulties in complicated cases	84
6.3	Methodology: TravRes	85
6.3.1	Basic ideas	86
6.3.2	Methodology framework	87
6.3.3	Inner loop: trajectory reconstruction	88

6.3.4	Outer loop: time-space speed reconstruction	90
6.4	Method validation	94
6.4.1	Experiment setup and generation of virtual FCD	95
6.4.2	Speed reconstruction by FCD only	95
6.5	Conclusion and recommendations	99
7	Speed estimation by fusing low-resolution positioning data	101
7.1	Introduction	102
7.1.1	Cellular networks and topology positioning data	102
7.1.2	Challenge and objectives	104
7.2	Methodology	105
7.2.1	Prototype and basic idea	106
7.2.2	Probe vehicle count process and speed re-estimation	108
7.2.3	Physical explanation and improved re-estimation	110
7.2.4	Analysis of variance	112
7.3	Validation	114
7.3.1	Validation by synthetic data	114
7.3.2	Impact of penetration of probe vehicles	117
7.3.3	Impacts of different prior speed probability distributions	121
7.3.4	Network-wide traffic speed estimation	122
7.4	Conclusions and recommendations	122
8	An integrated algorithm for fusing travel times, local speed and flow	125
8.1	Introduction	126
8.1.1	Basic relationships between traffic variables	126
8.1.2	Impact of traffic flow data	128
8.2	Methodology	129
8.2.1	Framework	130
8.2.2	Fusion part one: trajectory reconstruction	130
8.2.3	Fusion part two: speed and density reconstruction	133
8.3	Validation	135
8.3.1	Experiment setup & data generation	135
8.3.2	Results	136
8.4	Conclusion	141
9	Synthesis: a data fusion framework	143
9.1	Synthesis and comparison of data-data consistency methods	143
9.2	Data fusion framework	148
9.3	Summary	150

10 Conclusions and recommendations	151
10.1 Conclusions	151
10.1.1 Main conclusions	151
10.1.2 Conclusions related to the presented algorithms	152
10.2 Implications and recommendations	154
10.3 Future research	155
Bibliography	161
Summary	163
About the Author	165
List of TRAIL Thesis Series publications	167

<i>Version 18th April 2010</i>

List of Figures

1.1	Schematic representation of traffic state estimation and prediction	2
1.2	The Layout of this thesis	8
2.1	Trajectories in gridded time-space domain	10
2.2	Illustration of local measurements	12
2.3	Illustration of instant measurements	13
2.4	Illustration of Lagrangian measurements	13
2.5	Examples of some traffic data and their sources	17
2.6	Illustration of vehicles traveling around a ring-road	18
2.7	The impact of difference between time mean and harmonic mean speed .	19
2.8	The accumulated errors in loop counting	20
2.9	Relationship between vehicle trajectories	22
2.10	An example of low-resolution positioning data	23
3.1	The flow chart of traffic data processing on different fusion levels	29
3.2	Components in level one–data fusion of raw data	30
3.3	Literature review: the state of the art in traffic data fusion.	32
3.4	The estimation result by using Extended Kalman filter	36
3.5	The estimation result by using Particle filters	39
3.6	The estimated travel times by using Linear Programming	41
3.7	The estimation result by using Particle filters	44
3.8	Traffic flow estimation by Nudging method	46
3.9	Input data and output data types in data fusion techniques	46
3.10	Numerous trajectories correspond one travel time record	49
4.1	New approach: Data-Data Consistency approach	55
4.2	The new methods are based on different physical models	58
5.1	Illustration of how data-data consistency approach works	59
5.2	Travel time and local speed data have different levels.	61
5.3	Data fusion of travel times and local traffic data	63
5.4	The framework of PISCIT	65

5.5	Recursive reconstructions of vehicle trajectories	67
5.6	Illustration of individual vehicle trajectories	69
5.7	Illustration of the study road and how the ground-truth data are perturbed	71
5.8	Ground-truth time-space speed plot	72
5.9	Comparison with observed speeds using PISCIT	73
5.10	Comparison with ground-truth speeds on segment 12	73
5.11	Comparison with ground-truth speeds on segment 7	74
5.12	Compare travel times between before and after using this algorithm	74
5.13	The impact of FCD added into camera data	75
6.1	Illustration of internal consistency	77
6.2	Comparison with ground-truth data	80
6.3	Illustrations of trajectories running through time-space cells.	83
6.4	Evolution and convergence of time-space speeds plots	85
6.5	The whole framework of TravelRes	87
6.6	Illustration of trajectory reconstruction algorithm	89
6.7	Comparison of the ground-truth trajectory and the reconstructed trajectory	91
6.8	Assumed time-space contour plots for benchmark	94
6.9	Global MAREs under different polling and penetration rates	97
6.10	Mare under different cases	97
6.11	Coverage rate and Convergence of Mare	98
6.12	Comparison of estimated travel times with different methods	98
7.1	Illustration of the consistency rule in this chapter	101
7.2	Example cellular network.	103
7.3	Illustration of location displacement of vehicles	104
7.4	Model comparison of water container and roadway.	106
7.5	Illustration of all vehicles moving and probe vehicles moving	107
7.6	Illustration of a count process $\{\Psi(x), x \geq 0\}$	108
7.7	Physical explanation of Equation 7.8.	111
7.8	Probe vehicle in inflowing and outflowing traffic.	111
7.9	The layout of road infrastructure, segment and scenario	115
7.10	Comparison with observed speeds using FlowrRes	116
7.11	Speed time series plots	118
7.12	The influence of vehicle number and defined deviation	119
7.13	Theoretical relations	120
7.14	The influence of the forms of probability distribution	123
7.15	Road network and throughput of probe vehicles.	124
8.1	Illustration of Data-data consistency in this chapter	125
8.2	Time-space contour plot and its functions	127

8.3	The accumulated traffic counts from loop detectors	128
8.4	Fusion framework	131
8.5	Illustration of PISCIT algorithm	132
8.6	Fusion part two	134
8.7	Illustration of the study road and how the ground-truth data are perturbed	135
8.8	Ground-truth time-space speed plot	137
8.9	Comparison with observed speeds using ITSF	138
8.10	Comparison with ground-truth speeds on segment 10	139
8.11	Ground-truth time-space density plot	139
8.12	Comparison between observed density and estimated density	140
8.13	Comparison with ground-truth density on segment 10	141
8.14	Compare travel times between before and after using this algorithm . . .	141
9.1	MARE on speed estimation under different cases	145
9.2	Time-space consistency when FlowRes works with other methods	147
9.3	Application of fusion tools in the traffic system	149

List of Tables

2.1	Taxonomy of traffic measurements	11
2.2	Classification of data from traffic sensors with some examples.	17
2.3	Summary of data characteristics	24
3.1	Three-level data fusion	28
3.2	Examples in three-level traffic data fusion	29
3.3	The performance of EKF and UKF	37
3.4	Summary of varieties of assimilation techniques in traffic data fusion . . .	47
5.1	Symbol list	61
5.2	Comparison of MARE before and after using the proposed algorithm. . .	72
5.3	The impacts of added floating car data	75
6.1	Symbol list	78
6.2	MAREs of speed estimations by Naive and TravRes	96
6.3	MARE of speed estimation	96
7.1	The posterior distributions fit into Normal distributions	121
8.1	Symbol list	126
8.2	Comparison of MARE before and after using the proposed algorithm. . .	137
9.1	Brief on the proposed algorithms	144
9.2	The result from PISICT	145
9.3	The result from TravRes globally	146
9.4	The result from TravRes for low-speed region	146
9.5	The result from ITSF.	146

Chapter 1

Introduction

1.1 Research background

1.1.1 Context & background: the need for traffic data fusion methods

Traffic data collection and archiving systems are essential tools for online (and real-time) dynamic traffic management (DTM) applications, such as adaptive intersection control, routing, ramp metering, and traffic information services. Put simply, without data from sensors neither traffic management and control nor traffic information services are possible. However, the availability of data from a multitude of different traffic sensors do not necessarily amount to consistent, coherent and meaningful information on the state of a traffic network, for example in terms of speed, density or flow. Traffic state estimation and data fusion techniques are required to translate these available data into a consistent and complete description of the state in a traffic system. Figure 1.1 illustrates the position of traffic data fusion and state estimation (and prediction) in the context of real-time dynamic traffic management. In this thesis, the traffic states to be estimated are traffic speeds, flow and density. Later on in this thesis, some of the used terms and definitions will be given.

Traffic state estimation and data fusion are not only required for the online applications but also essential for *offline* applications, such as policy / measure evaluation studies and the development, calibration and validation of the tools necessary to perform these tasks (e.g. traffic simulation models). Finally, the advancement of scientific research itself relies heavily on the availability of large amounts of detailed and reliable empirical data

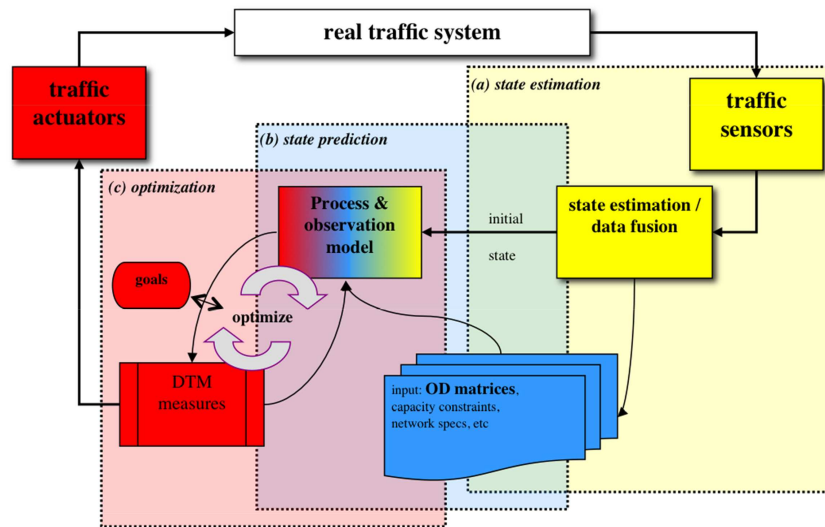


Figure 1.1: Schematic representation of traffic state estimation and prediction in the context of (dynamic) traffic management (DTM) & control

and on techniques to extract consistent and reliable information from these data.

In the last few decades the amount of empirical data becoming available for both online and offline use has steeply increased, particularly in terms of the wide range of sensor technologies developed and applied to collect these data. Traffic sensors may range from inductive loop detectors, radar, microwave, ultrasonic sensors to infrared cameras and in-vehicle GPS/GSM receivers/transmitters (“floating car data”), to name a few. The motorway network in the Netherlands for example (around 6600 km), has an inductive loop based monitoring system installed (with loops about every 500 meters), however, this only holds for around 1/3 of the network. Another 1/3 has only limited traffic monitoring, while the rest 1/3 has nothing at all. Besides the limited spatial coverage in some areas, a second major issue is that of the available data on average 5-10% is missing or otherwise deemed unreliable, with regular extremes over 25 or 30%. Also in the Netherlands other data sources (than loop detectors) are already available or will become available in the near future. Data from these different sensors (cameras, induction loops, or in-car GPS/GSM devices) are typically characterized by different formats, semantics, temporal and spatial resolution and accuracy, and also differ in availability and reliability as a function of location, time and circumstances (Van Lint et al. (2005), Van Lint (2006)). From technical points of view, all kinds of currently available sensors can not provide traffic measurements which have enough accuracy and coverage for solid traffic state estimation. However, economical situations may not allow the people to invest too much in equipment to collect traffic information. For this reason, scientists have to put more focus on the methodological part. For methodological points of view, the integration of

such heterogeneous data into comprehensive and consistent data warehouses is a complex and challenging task. This chapter focuses predominantly on the second challenge, that is, on the methodological tools to fuse heterogeneous traffic data. As we will see later in this thesis, particularly the *semantical* differences over space and time between these data impose strong constraints on the applicability of data fusion techniques.

1.1.2 Multi-sensor data fusion

In many fields of science, such as robotics, medical diagnosis, image processing, air traffic control, remote sensing and ocean surveillance (see e.g. Yager (2004); Xiong and Svensson (2002); Varshney (1997); Piella (2003); Linn and Hall (1991)), the de facto method for state-estimation is multi-sensor data fusion, a technique by which data from several sensors are combined by means of mathematical and/or statistical models to provide comprehensive and accurate information. A wide variety of approaches have been put forward for multi-sensor data fusion, based on, for instance, (extended) Kalman filters, Bayes methods and Artificial Neural Networks, Dempster-Shaefer theory or Fuzzy Logic. Which of these is suitable for the problem at hand is governed largely by domain specific constraints, the characteristics of the data available, and - probably most importantly, by the purpose and application for which the data is (f)used. Using similar arguments as in Dailey et al. (1996), data fusion generally leads to

- Increased confidence and accuracy and reduced ambiguity;
- Increased robustness: one sensor can contribute information where others are unavailable, inoperative, or ineffective;
- Enhanced spatial and temporal coverage: one sensor can work when or where another sensor cannot;
- and (more tentatively), decreased costs, because (a) a suite of ‘average’ sensors can achieve the same level of performance as a single, highly-reliable sensor and at a significantly lower cost, and (b) fewer sensors may be required to obtain a (for a particular application) *sufficient* picture of the system state.

With these arguments in mind, data fusion techniques provide an obvious solution for traffic state estimation. However, most approaches to traffic state estimation (e.g. Wang et al. (2006)) consider only a *single* source (i.e. minute aggregated or averaged flows and speeds from local inductive loop detectors), whereas of the studies that *do* consider data from various traffic sensors (e.g. Dailey et al. (1996); Kikuchi et al. (2000)) most are concerned

with limited traffic networks or corridors (e.g. single traffic links), and not as in the RENAISSANCE approach of Wang and Papageorgiou at comprehensive traffic surveillance and monitoring for entire freeway corridors or networks. As we will elaborate in the later chapters, the Kalman Filter (KF) approaches demonstrated in Wang et al. (2006); Van Lint and Hoogendoorn (2007); Herrera and Bayen (2007) do have other limitations, which relate to the spatio-temporal alignment of the available data.

1.2 Problem formulation

As shown in Figure 1.1, traffic state estimation plays a critical role in the traffic system. The performance of traffic management (including speed control, route guidance, etc) is highly dependent on the traffic state estimation. For this reason, we do need less-biased, solid estimation of basic traffic variables.

However, traffic data from different sensors are heterogeneous. They have different types of errors and don't provide time-space traffic measurements directly. Although now there exist some methods which can provide traffic state estimation by fusing some of traffic data, these methods involve quite a few assumptions and can neither fuse many more types of data nor lead to solid and reliable results. This will be shown and discussed later in Chapter 3.

The problem we face is: *In the state-of-the-art on traffic data fusion, there is no appropriate method which is able to provide solid traffic state estimation by fusing heterogeneous multi-types of data with limited assumptions.* This thesis will make effort in finding parsimonious approaches to fusing heterogeneous traffic data.

1.3 Research scope and objective

This research will focus on macroscopic traffic state estimation on a road stretch. The main traffic state variables in study are speed, density and traffic flow. The methods to achieve this estimation can be useful for network-wide traffic estimation, and can contribute to, for example, estimation of queue length in road intersection, OD (origin-destination) estimation, etc.

The objectives of this research is to reconstruct less-biased and solid traffic variables by fusing heterogeneous traffic data in a parsimonious way. We will provide a new approach and methods to give more reliable time-space mean traffic states. Meanwhile, many more types of traffic data can be fused while less assumptions are made.

1.4 Theoretical and practical contributions

The research in this book has a number of scientific, methodological and practical contributions. These contributions are summarized in this section.

1.4.1 Scientific contributions

This thesis provides scientists with a comprehensive and deep insight into traffic data fusion. A new traffic data fusion paradigm and a new fusion approach are given. The methods proposed in this thesis can help scientists to know more about traffic data and may contribute to new traffic theories and models. Also these methods and approaches may open a door to developing more advanced data fusion methods. The scientific contributions made in this thesis are listed as below:

- This thesis develops a new taxonomy to classify all traffic data fusion methods and it presents the state-of-the-art on traffic data fusion methods from the perspective of this taxonomy. Each of the data fusion methods has two major components: a core and a shell. The core represents the assumptions in traffic theory, which establishes the connections between different types of data, between data and estimated variables. The shell represents the assimilation techniques, particularly some statistical techniques, which may be able to combine models and data in statistically optimal ways.
- This thesis establishes a new paradigm that uses a data-data consistency approach to fuse different types of traffic data. This paradigm is parsimonious, which only needs very few assumptions but can fuse more types of data. Scientists can use this paradigm to develop many other simple but effective methods to fuse traffic data.
- This thesis proposes a new scheme to estimate optimal traffic speeds in OLS (ordinary least square) sense by using local speeds and travel times. Scientists can also use this method to easily reconstruct individual trajectories and therefore obtain more details about traffic characteristics.
- Travel times can be easily estimated from the ground-truth traffic speeds, but the inverse process is very difficult and complex since traffic changes over both time and space domains. This thesis theoretically shows that the inverse process is possible and that the exact trajectories can be reconstructed by only travel times under certain circumstances. As a consequence, it provides theoretical supports for scientists to develop new methods to estimate traffic speeds only by using travel times.

- A new iterative approach with dual-loop iterations is proposed to reconstruct individual trajectories by using only travel times. In this approach, a new stochastic model based on Brownian Motion is given to determine the confidence on different travel time records considering both time intervals and traffic conditions.
- It establishes a new stochastic model to fuse low-resolution positioning data and prior speed information. This model establishes a probabilistic relationship between traffic speeds and sampled traffic flow.
- It develops a new scheme to fuse traffic speeds, flow and travel times. In this scheme, only fundamental physical laws and traffic theory are used, but these three types of data of different semantics can be fused by simply using an iterative approach and a linear regression technique.

1.4.2 Methodological contributions

This thesis proposes new methods that provide both efficiency in computation cost and excellent performance in estimation accuracy and robustness. The methodological contributions made in this thesis are listed as below:

- This thesis develops a new iterative approach combined with a linear regression technique to fuse local speeds and travel times. With this methodology, the reconstructed trajectories can quickly converge in less than ten iterations.
- This thesis develops a new iterative approach to make the best use of travel times. An inner loop and an outer loop form the whole iterative approach. The inner loop and outer loop work together and enable the convergence of speed estimation within tens of iterations and as a result more accurate speed estimation can be made out of travel times only.
- It uses the Bayesian rule and order statistics to establish the probabilistic relationship between speeds and sample flow in a simple way.
- In the method that fuses speeds, travel times and flow, it combines both the iterative approach and linear regression techniques. This methodology provides high computation efficiency.

1.4.3 Practical contributions

The methods proposed in this thesis can satisfy many real life requirements. For example, the available data in real life world may only have low resolutions. FCD (Floating Car Data) data may have low polling rates or may have low positioning resolution, and the cameras for travel time may be far away from each other. Some data source has its particular attributes. For example, the speed measurements from loop detectors have considerable biases. This thesis aims to use these real-life data and tackle the practical challenges in them. The contributions are listed as below:

- It was difficult for previous methods to use travel times for speed estimation, or particular for the travel times from far spaced cameras. The proposed method ‘PIS-CIT’ (Piece-wise Inverse Speed Correction by using Individual Travel-time) is able to fuse local traffic speeds and travel times so that much-less biased traffic speed estimation can be achieved. This proposed method can easily use travel times of larger intervals and can reduce the error in local speed measurements by a few times.
- When travel times are the only data source available, the previous methods that use it for speed estimation lead to considerable bias. But our proposed method ‘TravRes’ in Chapter 6 is very useful to reconstruct accurate individual trajectories and therefore can achieve more accurate speed estimation. The accuracy can be improved by about two times.
- Mobile phone tracking data in cell levels were not used for high-resolution speed estimation, but they exist in large quantities. The proposed ‘FlowRes’ method in Chapter 7 is able to use this low-resolution positioning data to estimate traffic speeds and further give the magnitude on estimation error. The magnitude on estimation error can provide the user with the confidence and reliability on traffic speeds and travel times. This method can also easily be extended to be applicable in network-wide speed estimation.
- Fusing local speeds, flow and travel times in a simple framework is highly demanded in practical applications but was always a challenge. The proposed method ‘ITSF’ (Integrated algorithm for fusing Travel times, local Speed and Flow) in Chapter 8 is able to estimate both speeds and density by fusing local traffic speeds, flow and travel times through the data-data consistency approach. In comparison with the result from the single data source-biased loop speeds, this method can improve the accuracy by up to 10 times.

1.5 Thesis Outline

Below is the brief outline of this thesis. Chapter 1 is an introduction, which gives the background, research scope, research objectives and main contributions. Chapter 2 discusses the characteristics of traffic data. Chapter 3 presents the state-of-the-art on traffic data fusion. Chapter 4 proposes a data-data consistency approach. Chapter 5, 6, 7 and 8 proposes four data fusion algorithms for different purposes. Among them, Chapter 5, 6 mainly use physics law $speed * time = distance$; Chapter 7 uses physical law $density * speed = flow$; Chapter 8 uses both laws. Chapter 9 further discusses applications of the proposed methods and proposes a possible fusion framework. Chapter 10 concludes this thesis. Figure 1.2 illustrates the layout of this thesis.

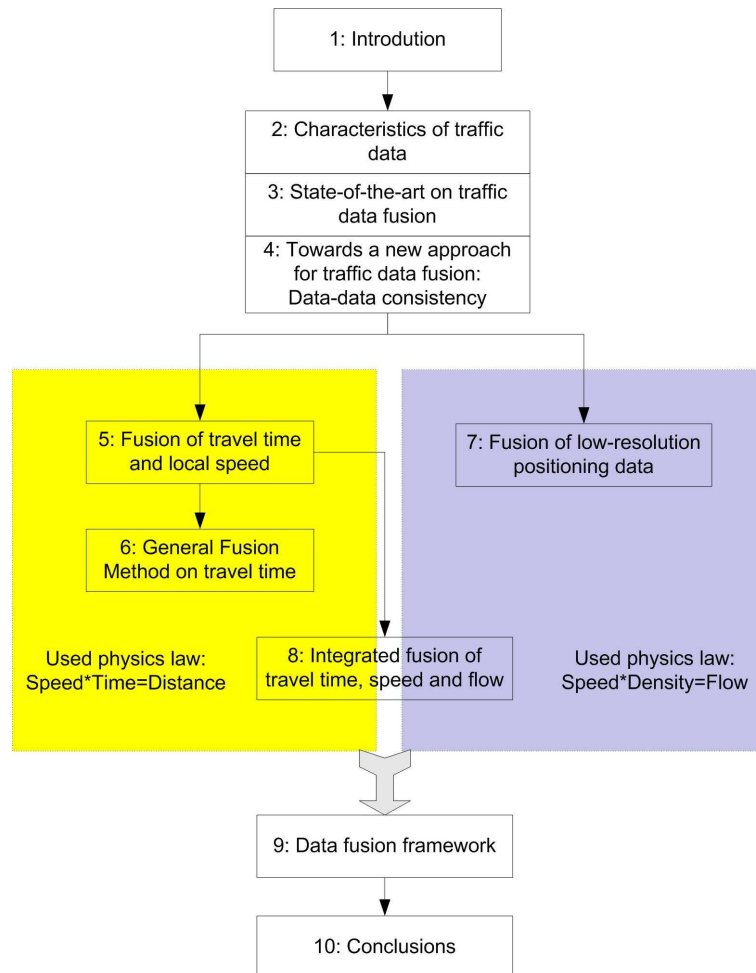


Figure 1.2: The Layout of this thesis

Chapter 2

Traffic data collection and importance of data fusion

2.1 Introduction

This chapter will give the definitions of traffic variables in this thesis, mainly traffic flow, speeds and density. It also introduces the mainly used traffic data collection techniques and data characteristics. We will establish the basic knowledge and concepts about traffic data and their characteristics. In addition, we will also find that data fusion techniques are indeed important in order to make use of the data.

Our focus is the macroscopic level, in which traffic is in analogy to fluid or gas, described as a continuum. First we give the definitions of the most important macroscopic variables, such as flow, speed, density, etc. We use Edie's definition (Edie (1963)) to define these variables. Following that, we will talk about the data collection techniques, and characteristics of different data sources. In the end, we will summarize and conclude this chapter.

2.2 Definitions of basic traffic variables

The basic macroscopic traffic variables used to describe a traffic state include flow q (veh/h), speed v (km/h) and vehicular density ρ (veh/km). Since traffic evolves over time-space region, the definitions of the variables normally depend on the observation approach.

2.2.1 generalized variables with Edie definition

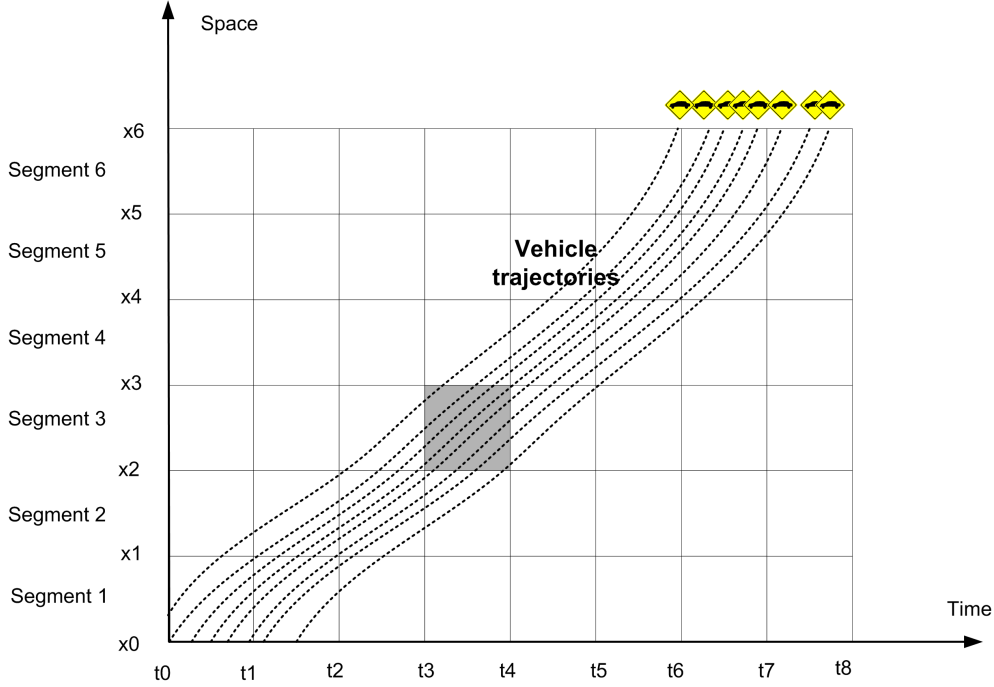


Figure 2.1: Trajectories in gridded time-space domain

Edie (1963) gives general definitions of these variables from perspective of vehicle trajectories. Figure 2.1 shows vehicle trajectories in a time-space domain. According to Edie's definition, the traffic characteristics in the shaded region in Figure 2.1 are given by

$$\text{Flow: } q(A_n) = \frac{d(A_n)}{|A_n|} \quad (2.1)$$

$$\text{Density: } \rho(A_n) = \frac{t(A_n)}{|A_n|} \quad (2.2)$$

$$\text{Speed: } v(A_n) = \frac{d(A_n)}{t(A_n)} \quad (2.3)$$

where A_n can actually represent an arbitrary time-space region. In our illustration, it is the shaded region $\{A_n : x \in (x_2, x_3) \quad t \in (t_3, t_4)\}$, $d(A_n) = \sum d_i$ is the total distance traveled by all vehicles in region A_n , $t(A_n) = \sum t_i$ is the total travel time spent by all these vehicles in region A_n , and $|A_n|$ is the area of region A_n . Based on such definitions, each quantity depicts traffic states over a certain time-space region, which make it

convenient and neat to represent traffic evolution over time and space. In particular, v is the so-called *time-space* mean speed, which is a primary input to compute travel times. Combining Equations 2.1, 2.2 and 2.3, these variable are related to one another by

$$q = \rho v \quad (2.4)$$

Traffic-flow characteristics such as flow, speed and density are of importance to traffic operations and management. Flow is a direct measure of traffic throughput, density is the most important variable for many of macroscopic traffic models, and (time-space mean) speed determine travel times. As Ni (2007) suggests, successful applications in Intelligent Transport System call for a solution that is able to determine these characteristics by using all kind of traffic sensors and the same time is able to preserve the basic relationship ($q = \rho v$).

2.2.2 Eulerian and Lagrangian measurements

Also local and instantaneous traffic variables can be defined according to Edie's definition. As already seen above, the generalized traffic variables are defined in a certain time-space region. If the time interval for this region becomes very small, the generalized variables are simplified to instantaneous variables. If the space interval for this region becomes very small, they are simplified to local variables.

In order to estimate the traffic variables, the traffic measurements are needed. These measurements can be categorized into Eulerian measurements and Lagrangian measurements. Eulerian measurements can be further classified into local, instant and time-space measurements. Lagrangian measurements are typically represented by vehicle trajectories. The categorization can be seen in Table 2.1. In the ensuing, we will define these traffic flow measurements.

Table 2.1: Taxonomy of traffic measurements

Measurement types	Sub-types	Examples
Eulerian measurements	time-space measurements	radar measurements
	local measurements	data from loop detectors
	instant measurements	traffic image
Lagrangian measurements	complete	vehicle trajectories
	incomplete	travel times from camera

Local Measurements. Local measurements are the measurements which measure traffic in a time-space region which space interval becomes extremely small. They refer point

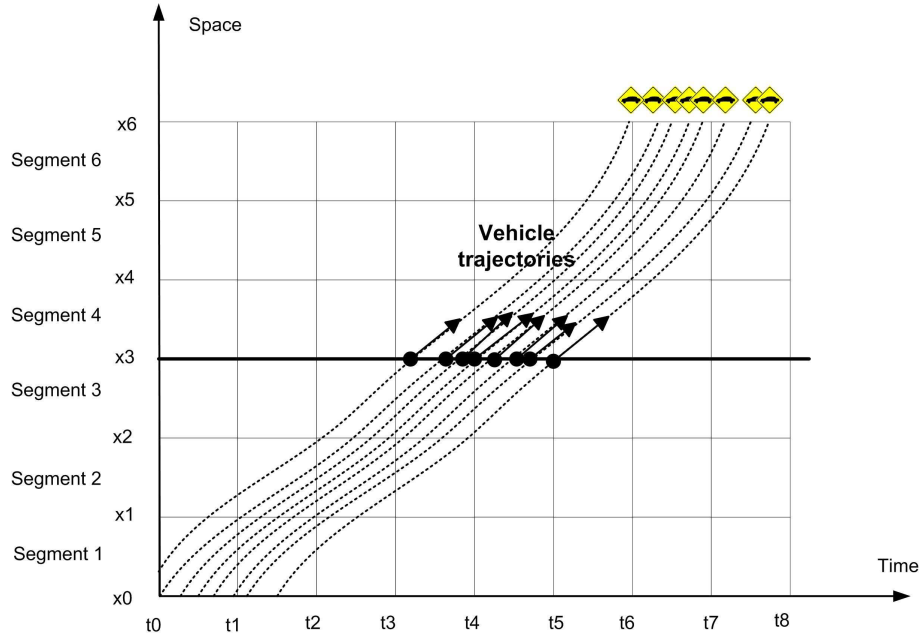


Figure 2.2: Illustration of local measurements

observations on a road. Local flows represent the number of vehicles n passing a certain point x during a period T .

$$q(x) = \frac{n}{T} \quad (2.5)$$

Local speed can be expressed in time-mean speed or harmonic speed and local time-mean speed is

$$v(x) = \frac{\sum v_i(x)}{n} \quad (2.6)$$

or local harmonic speed

$$v(x) = \frac{n}{\sum 1/v_i} \quad (2.7)$$

Harmonic speed is equal to time-space mean speed under conditions of homogeneous and stationary traffic. The local density during period T cannot be directly observed but only be derived from speeds and flow. The derived local density is

$$k(x) = \frac{q(x)}{v(x)} \quad (2.8)$$

Since there are two types of local speeds, there are two types of local density which may be quite different from each other.

Instantaneous Measurements. Instantaneous measurements are the measurements

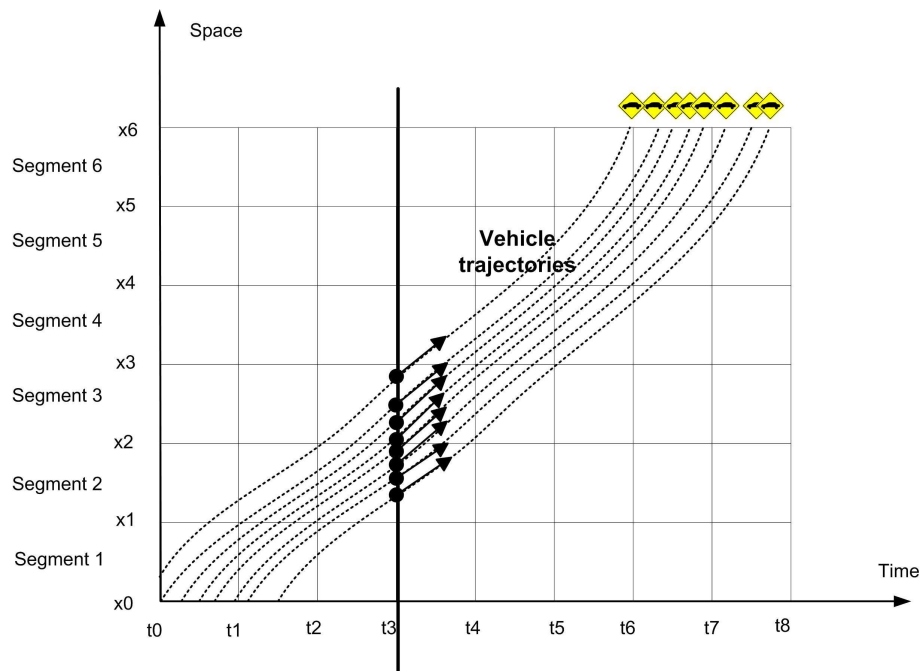


Figure 2.3: Illustration of instant measurements

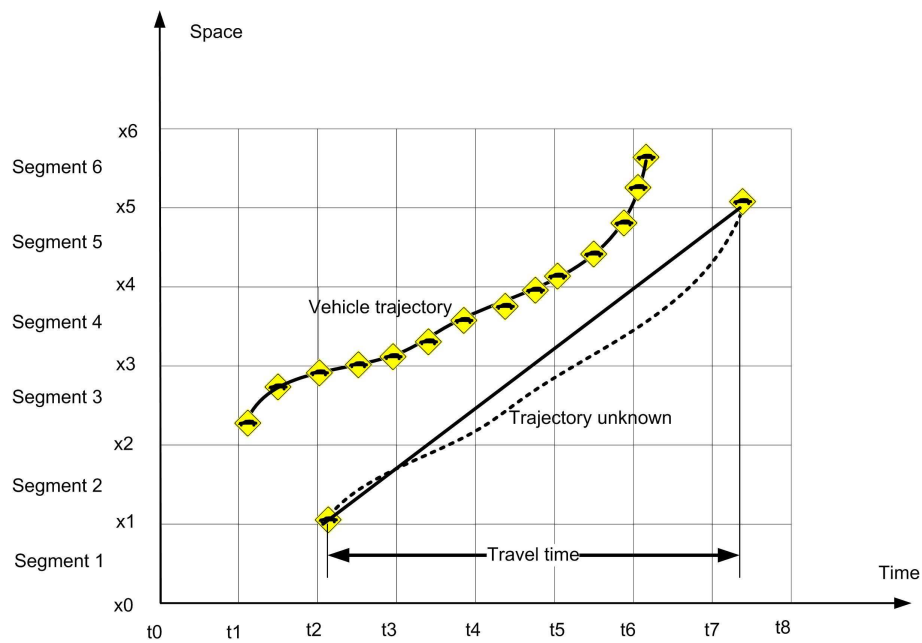


Figure 2.4: Illustration of Lagrangian measurements

which measure traffic condition in a time-space region which time-intervals become extremely small. They give traffic conditions over a road section of a certain length X at instant time t . They can be taken as a picture captured by a camera at time t and the traffic states are ‘frozen’ at this moment. Instantaneous density is defined as:

$$k(t) = \frac{n}{X} \quad (2.9)$$

The space-mean speed over this section at instant time t is

$$v(t) = \frac{\sum v_i(t)}{n} \quad (2.10)$$

It is worth to mention that v_i here refers to the speed of vehicle i on this road section at time t . For instantaneous variables, flow cannot be directly observed but only derived as

$$q(t) = k(t) * v(t) \quad (2.11)$$

In order to achieve the estimation of traffic over a complete time-space region, the time-space measurements are needed. However, due to the limitation of measurement techniques, only local measurements and instantaneous measurements can be available in majority of cases. Unfortunately, local measurements can only provide the traffic conditions at a certain point, and instantaneous can only provide traffic conditions at a certain moment. This makes a challenge in traffic state estimation.

Lagrangian Measurements. Lagrangian measurements are the measurements which measure an individual fluid parcel (e.g. an individual vehicle) as it moves through space and time. These measurements can plot all the positions of an individual parcel through time. Vehicle trajectories are typically Lagrangian measurements as seen in Figure 2.4. This kind of measurements can be obtained via GPS technology or any tracking devices. Different from Eulerian measurements, Lagrangian measurements reflect how a vehicle experiences traffic.

In many cases, however, we cannot get a complete vehicle trajectory. Instead we can only know the time spent for a vehicle to travel from one location to another. Such travel time records can also be regarded as Lagrangian measurements. Since corresponding trajectories can not be determined, travel times turn to be incomplete Lagrangian measurements.

2.2.3 Issues in computing traffic variables from available data sources

We want to estimate traffic states under Edie's definition, which are time-space mean variables. In many cases, however, we can not obtain time-space mean measurements to cover a complete time-space region. If we want to use local measurements or instant measurements to estimate the time-space variables, we have to extend the range of these measurements. For example, local measurements only provide the traffic states at a certain point. So we have to assume the traffic is homogeneous in the space region around this point. For instant measurements, we may have to assume stationary traffic during an interval around a certain time instant. The questions are: what is the consequence of making the above assumption? Does any error will involve when such assumptions are made.

Furthermore, for Lagrangian measurements, we cannot obtain trajectories for all vehicles and we can not even get trajectories but travel times instead. The question is: how do these measurements contribute to the estimation of time-space variables?

In the following section, we will look into specific types of traffic data or traffic measurements in real-life world. We will show that there is indeed a big gap between time-space variables and different types of traffic measurements and data, which cause a challenge for traffic data fusion.

2.3 Traffic data

2.3.1 Brief overview

In the last decades the amount of empirical data becoming available for both online and offline use has steeply increased, particularly in terms of the wide range of sensor technologies developed and applied to collect these data. Traffic sensors may range from inductive loop detectors, radar, microwave, ultrasonic sensors to infrared cameras and in-vehicle GPS/GSM receivers/transmitters ("floating car data"), to name a few. Data from these different sensors (cameras, induction loops, or in-car GPS/GSM devices) are typically characterized by different formats, semantics, temporal and spatial resolution and accuracy, availability as a function of location, time and circumstances. Figure 2.5 gives some examples of different sensors and their representation on a time-space plane.

In addition, each of the traffic observation are characterized by Van Lint (2004) as shown

below:

- data semantics (for example: space-mean speeds and time-mean speeds have different semantics, since both of them represents speeds but have different intrinsic meanings and levels)
- spatial level of aggregation (for example: distance between inductive loop or cameras)
- temporal level of aggregation (for example: 1 minute or 5 minutes aggregation)
- availability in terms of frequency (time) and scope (place, link, route)
- accuracy (expressed as a function of time, place and traffic conditions)
- technical aspects (for example, database format, communication protocols, etc)
- infrastructure bound or free (for example: roadside versus in-car GPS/GSM)
- ownership of data (for example: private or public)
- usage cost (for example: equipment cost, installation cost, maintenance cost, etc)

Although data from traffic sensors come in many forms and qualities, they can essentially be subdivided along two dimensions. The first relates to their spatio-temporal semantics, that is, do the data represent local traffic quantities (speed, time headway(s), etc) or do the data reflect quantities over space (journey speed, local speed, instantaneous speed, time-space mean speed, travel times, trajectories). The second relates to the degree of aggregation, where data may represent an aggregate or average over fixed time periods (e.g. 1 minute aggregate flows or averaged speeds), or a single event (vehicle passage, travel time, full trajectory). Table 2.2 presents an overview of this classification with a few examples. The main consequence for fusing these fundamentally different data is that these data need to be aligned over space and time such that we can employ mathematical and statistical models to average, filter and combine them into one consistent picture (space-time map) of the traffic conditions.

A comprehensive overview of traffic data collection systems can be found in Westerman (1995); Michalopoulos and Hourdakis (2001). Next, more attention is paid to the data types that concern this thesis.

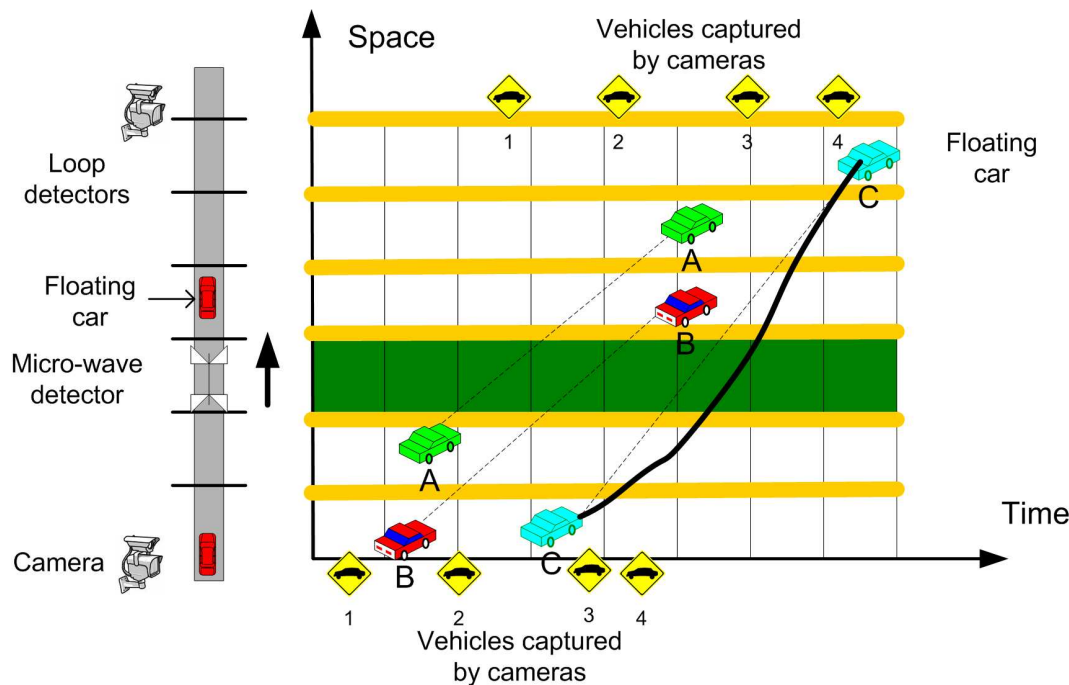


Figure 2.5: Examples of some traffic data and their sources

Table 2.2: Classification of data from traffic sensors with some examples.

	Event-based	Aggregate
Local	vehicle passage speed, low-resolution FCD, etc	loop flow loop speed, etc
Spatial	AVI travel time, journey speed, etc	time-space mean speed, mean travel time, etc

2.3.2 Loop data

The most common data available for traffic state estimation come from (dual) loop detectors. In the Netherlands for example, the main freeways have an inductive loop about every 500 meters. These loop detectors can provide speed measurements and flow measures at the exact locations where they are installed. These data are presented in aggregated values for a time period ranging from 30 seconds to 10 minutes. Although the speed

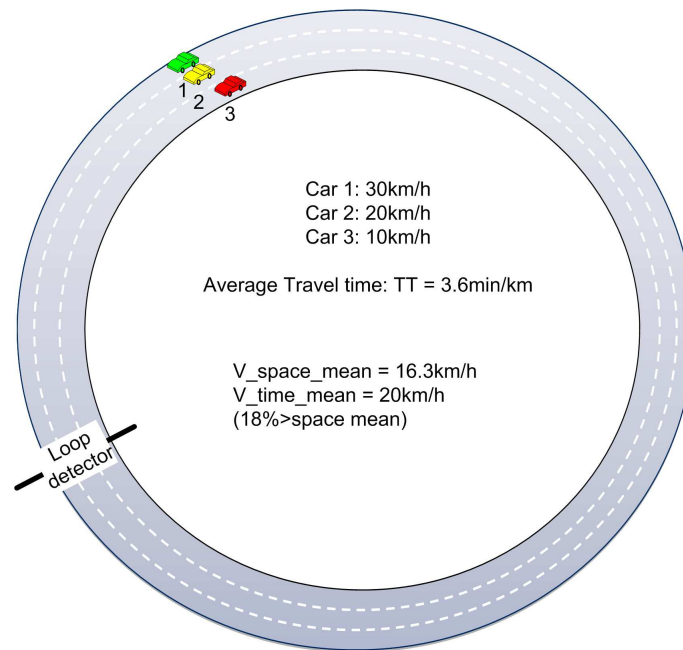


Figure 2.6: Illustration of vehicles traveling around a ring-road to explain the consequence when time-mean speeds are used.

measurements have errors within 5% for ordinary vehicles, the time-mean speeds stored in a collection system will make a considerable bias for density estimation and travel time estimation. Another type of speed measures are called time-space mean speeds. They represent the journey speeds with which vehicles cover a certain road stretch. If we want to get a correct estimate of travel time, the space mean speeds are required. It is also true for traffic density estimation. The following example shows the difference between time-mean speeds and space-mean ones.

Figure 2.6 shows a one-kilometer long ring road on which there are three cars running with constant speeds 10km/h, 20km/h and 30km/h respectively. Assuming that the three cars run on the ring road for only once, it can be deduced that the travel times for them to cover this road stretch correspond to $1\text{km}/10\text{km/h} = 6\text{min}$, $1\text{km}/20\text{km/h} = 3\text{min}$ and

1km/30km/h = 2min. The average travel time is $(6 + 3 + 2)/3 = 3.67min$. However, if one loop detector is placed on this road and aggregates the data for every 6 minutes, then the time-mean speed is 20km/h. As a result, the estimated travel time from time-mean speed turns out to be 3 minutes. In contrast, the space-mean speed will exactly lead to travel time estimates of 3.67 minutes.

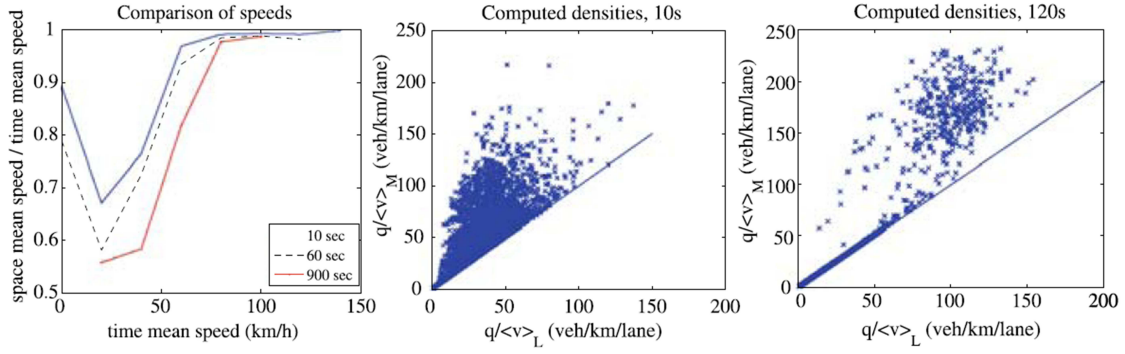


Figure 2.7: The impact of difference between time mean and harmonic mean speed: (left) average speed difference; (middle) difference in density, 10 seconds aggregation; (right) difference in density, 120 seconds aggregation (from: Knoop et al. (2007))

Let v depict the so-called *space-mean* speed v_S , i.e. the arithmetic average of the speeds of vehicles present on the road section r of interest (with length L_r) at some time instant t . With loop detectors along r , this speed can be approximated by the local *harmonic* mean speed v_M , that is

$$v_S \approx v_M = \langle v \rangle_M = \frac{L_r}{\frac{1}{N} \sum_i \frac{1}{v_i}} \quad (2.12)$$

The approximation in (2.12) is exact in the case that road stretch is very short and speeds are constant over the region. The local arithmetic (or simply *time*) mean speed $v_L = \langle v \rangle_L = \frac{1}{N} \sum v_i$ provides a biased approximation of the space mean, due to the fact that in a time sample *faster* observations are over represented. That this bias is significant, specifically under congested (low-speed) conditions has been demonstrated for example by Treiber and Helbing (2002) for estimating travel times (errors of over 30%), and by Knoop et al. (2007) for estimating densities, where the resulting errors can mount up to over 100% as shown in Figure 2.7. In this figure, $q/\langle v \rangle_M$ represents the density estimates derived from harmonic mean speed, and $q/\langle v \rangle_L$ represents the density estimates derived from time-mean speed. Since harmonic mean speed is the approximation of space-mean speed, so this figure also implicates the difference between time-mean speed and space-mean speed.

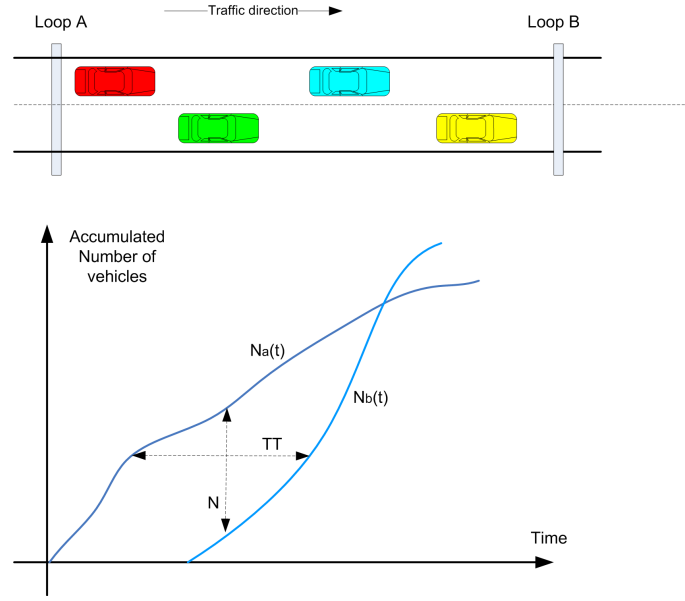


Figure 2.8: The accumulated traffic counts from loop detectors. The actual curve for accumulated counts may cross each other due to accumulated error.

Apart from speed measures, loop detectors are able to count the number of passing vehicles during a certain interval. The counts lead to estimates of flow q , and they are also quite reliable. Theoretically, the accumulated vehicle counts may tell travel time and vehicle density (or vehicle number) on a closed road section between two consecutive loop detectors as shown in Figure 2.8. In this figure, $N_a(t)$ and $N_b(t)$ are the accumulated vehicle counts from loop A and loop B respectively. Given the initial condition that no vehicle is on the road section, $N(t) = N_a(t) - N_b(t)$ is the number of vehicle on this road section. Another estimate $TT(t)$ under the condition that $N_a(t) = N_b(t + TT)$, can be taken as the average travel time from loop A to B. These estimates are reliable and accurate if loop detectors made no errors in counting vehicles.

However, correct estimates can not be obtained in reality, due to error accumulation. In the above example, the errors in $N_a(t)$ and $N_b(t)$ are accumulated with the time t . We made an empirical study of one-day data on a one-kilometer section of highway A13 in Netherlands. The data was from Regionlab-Delft which stores the traffic data in Zuid-Holland region. With these data, it was found, during the period from 6:00 to 20:00, the total number of inflow vehicles was 64823 counted by the upstream loop detector (loop A) on the section, and that the number of outflow vehicles was 71000 counted by the downstream loop detector (loop B). The difference between inflowing traffic and outflowing traffic is accumulated up to -6177. According to vehicle conservation law, one explanation is that there are thousand of vehicles within the one-kilometer section, which

is impossible in reality. The other explanation is that the loop detectors have considerable errors in accumulated counting. For this reason, the curve for accumulated vehicle counts may cross each other as illustrated in Figure 2.8.

2.3.3 Travel time and trajectory data

In this thesis, travel time is given a particular attention considering that more and more travel time information becomes available. Travel times can be measured by means of for example automated vehicle identification (AVI) systems, which identify vehicles at two consecutive locations A and B at time instants t_A and t_B and deduce the *realized* travel time afterwards with $TT_r = t_B - t_A$. AVI systems may employ cameras and license plate detection technology, or may match vehicles through induction footprints, tolling tags or otherwise. Methodologically, the most important characteristics of travel time are that:

- Travel time can only be measured for realized trips, i.e. after a vehicle has finished it. The so-called *actual* travel time TT_a of a vehicle departing at the current moment must hence be predicted per definition (Figure 2.9).
- Travel time (or its reciprocal average journey speed $u_r = L_r/TT_r$) is an average representation of the traffic conditions (e.g. the speed $v(t, x)$) a vehicle encountered during its trip. Figure 2.9 illustrates this by superimposing vehicle trajectories on a speed contour map. This implies that the relationship between this travel time and the underlying traffic conditions (the speed contour through which a vehicle has ‘traversed’ is $1 : N$. It is possible to estimate travel time from local speeds (Van Lint and Van der Zijpp (2003); Ni and Wang (2008)), but conversely, it is very difficult to estimate local speeds accurately from travel times, unless other sources of information are available.

By sampling data (location and/or speed) from instrumented vehicles (e.g. through GPS or GSM) at consecutive time instants, also vehicle trajectories can be measured. Clearly, when *all* vehicle trajectories are sampled at high frequencies, the traffic state (prevailing speeds, flows, densities, travel times, etc) can be completely deduced from these so-called *floating car data* (FCD). However, it is estimated at the end of 2009 that the penetration rate of real-time traffic information and GPS enabled vehicles which actually transmit their location and speed to their service provider is in the order of one percent or less of the total amount of vehicles driving on the Dutch freeways. Therefore, at penetration rates far below 100%, FCD at best provide a proxy for average speed on the road segments from which these data are collected.

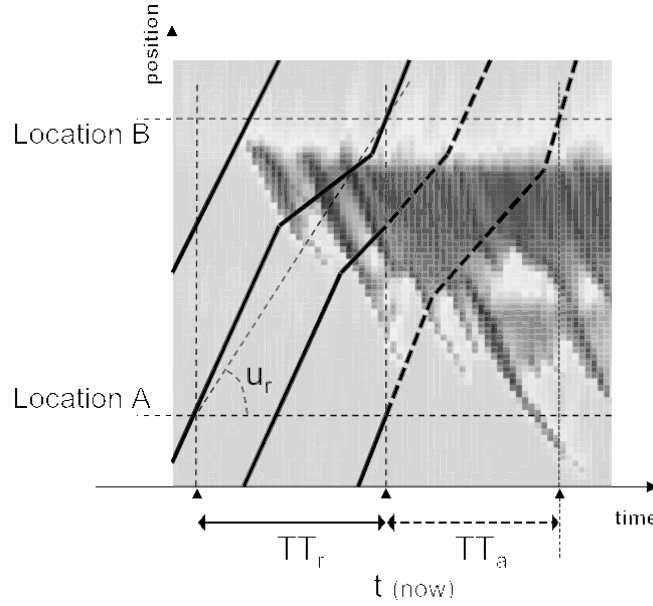


Figure 2.9: Relationship between vehicle trajectories (the thick solid and dotted lines), realized and actual travel time (TT_r and TT_a), average journey speed (u_r) and the underlying speeds (represented by a speed contour plot, where dark areas represent low speeds)

In addition, the availability of communication resources restricts the access to floating car data. FCD can hardly be used for traffic state estimation before they are sent to information center via certain communication tunnels and collected. The simple way is to use wireless networks. When in-car equipments send relevant information via wireless networks, a certain amount of communication resources have to be consumed. If trajectory data are required to be available, vehicles must report its location at a quite high polling rate (once a few seconds). In that case, more communication resources will be consumed due to more information transmission or more frequent communication. This may bring about high cost. In addition to high cost, there are also physical limits, since the frequency band-width for civil communication is quite limited and precious. For this reason, this thesis tends to focus on FCD of low polling rates (once for one minute).

Nonetheless, with the in-car ICT revolution, it is reasonable to assume that more floating car data will become available in the coming years. But for estimation of flows or densities (required for many traffic management purposes such as intersection control, ramp metering, but also for forecasting traffic information) other (local) data sources rather than travel time or trajectory samples are necessary.

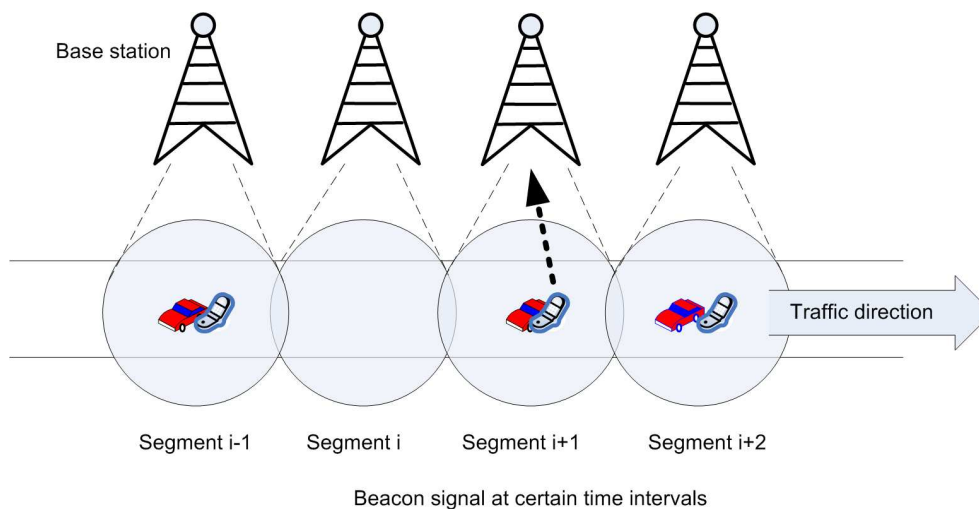


Figure 2.10: An example of low-resolution positioning data from wireless communication networks

2.3.4 Low-resolution positioning data

Probe vehicles with global positioning systems (GPS) can provide accurate positions which enable spatial-average speed estimation. However, some probe vehicles cannot provide accurate positions but can provide some location-specific information when and where they are located at the segment or cell level. These low-resolution positioning data with segment or cell level accuracy cannot provide the distance component that is necessary for traffic speed estimation, but they can be easily available in large quantities in wireless networks.

As shown in Figure 2.10, such kind of data don't provide exact geographical positions but point to a location area—a cell or a road segment. When a mobile phone in a vehicle sends beacon signals periodically, the cellular networks are able to trace the phone and record the cell where it is located. Although traffic speeds can not be directly estimated with these data, the traffic flow may be indicated by these data. So they are essentially *sampled flow*.

Assuming that beacon signal transmission can be one way and occurs at a frequency on the order of minutes, the communication is relatively simple and low cost. In addition, low-resolution positioning data are more widely available in terms of time and space, since devices as possible providers (e. g. mobile phones, laptops, iPhones, etc) are being widely used in communication networks and are becoming increasingly popular. It can be seen that the simplicity and wide availability of low-resolution positioning data may

have potential for traffic estimation in large networks.

2.3.5 Summary

The below table 2.3 gives a summary of data which are mainly used in this thesis.

Table 2.3: Summary of data characteristics

Data sources	Features
Speed measurements by loop	Loop detectors can provide the speed measurements only at certain points on a road (local speeds). They cannot provide the speed measurements over a road section. Normally loop detectors aggregate the speed measurements every 20 seconds or 1 minute, which leads to over-representation of high speed measurements. Therefore the speed measurements from loop detectors have structural deviation from the ground-truth speeds. The deviation is relatively bigger when the speed is lower.
Flow measurements by loop	Loop detectors can count the vehicles which pass and aggregate them every a certain period of time. In contrast to loop speed measurements, loop flow measurements have much less bias. But if these measurements are used to estimate the total flow during a long period of time, there are accumulated errors. For this reason, one can not apply vehicle conservation law on loop detectors for density estimation.
AVI data (travel time data)	AVI systems provide the travel times from location A to location B. The data may come from cameras that capture vehicle plates and make comparison, or from in-car GPS devices which may report its location at certain intervals. Less-biased journey speeds can be derived from these data. If a vehicle report its locations at short intervals, high-resolution trajectory can be constructed. But it will cost more communication resources. If it reports the location at long intervals, the data can not provide the details on traffic conditions but journey speed
low-resolution positioning data	Low-resolution positioning data may come from cell phone networks. The positioning data cannot provide the exact position but only tell in which area the vehicle is. Therefore it is hard to reconstruct accurate trajectory or obtain the journey speeds. But such data may be available in large quantities. It is quite beneficial if they can be used for traffic state estimation.

2.4 Conclusion

This chapter mainly gives basic concepts about traffic system from macroscopic perspectives. The major traffic variables on macroscopic level are speed, flow and density. Furthermore, we use Edie's definition to define these variables. Edie's definition can easily represents the traffic states in time-space domain and can establish clear and solid relationship between the variables. For these reasons, speeds, flow and density in terms of time-space mean are our focus and output from the estimators.

In general, there are Eulerian measurements and Lagrangian measurements for traffic variable estimation. However, not all the measurements for estimation are simply time-space ones. They could be local or instant measurements. So when we use these measurements for estimation of time-space variables, we often have to make some assumptions. For example, we may have to assume that the local speeds from loop data can represent the speeds over a large space. These assumptions, however, may lead to considerable errors.

In this chapter, we also have a close look at very common traffic measurements and traffic data. Loop data (speed and flow measures), travel times from AVI systems, low-resolution positioning data (e.g. cell-phone data) are the major data types that this thesis studies, for these types of data are the main data types that traffic data collection systems can provide or will provide in large quantities.

Seen from this chapter, each type of data can partially provides traffic states. However, the estimates may be unreliable, inaccurate or have limited time-space coverage. So we need data fusion techniques to improve this. The next chapter "state-of-the-art" will show how the already-existing data fusion methods use the above-mentioned traffic data to estimate traffic states. However, those fusion methods can fuse only certain types of data but under quite a few assumptions. The estimation results may not be desirable. For example, the bias in measurements may not be substantially removed. For this reason, this thesis proposes a new data fusion approach which is able to fuse more types of data, and only needs very limited assumptions. Better estimations of traffic states can be achieved.

Chapter 3

The state-of-the-art in traffic data fusion

The previous chapter gave the fundamental concepts about traffic data and traffic systems on macroscopic level. With this knowledge in mind, we begin to talk about traffic data fusion techniques. This chapter will focus on the state-of-the-art in traffic data fusion. It will firstly give a brief introduction to data fusion and the traffic data fusion. Then it will present some classic methods or algorithms which are able to fuse traffic data with different characteristics from different sources by using various traffic models and assimilation tools. Following that, we reveal some drawbacks of these methods and the challenges that are still left un-conquered. These challenges lead to the creation of a new approach.

3.1 Data fusion and traffic data fusion

3.1.1 Levels on data fusion

Varshney (1997) proposed a simple three-level model for data fusion. Each level has its particular function and purpose. Furthermore, the higher level data fusion is supported by the result from low level data fusion. Table 3.1 shows the main functions and often-used methods on each level.

The first-level data fusion is targeted at the raw data processing and estimates the basic states of an object. For example, the measurements from several radars can be fused to estimate the states of a flying aircraft, such as its speed, direction, location. The quality

Table 3.1: Three-level data fusion

Level	Purpose	Typically used methods
Level 1	Processing raw data	State estimation methods (Digital filters, LS estimation, EM methods, Kalman filters, Particle filters, etc)
Level 2	Deriving features and patterns	Classification/inference methods (statistical pattern recognition, Dempster-Shafer, Bayesian methods, Neural Networks, Correlation Measures, Fuzzy Set Theory, etc)
Level 3	Making decisions and detecting events	DSS (Decision Support System) and Expert System (Bayesian belief networks, Fuzzy/AI, etc)

of estimation is determined not only by the accuracy and number of measurements but also by the data fusion techniques. The mainly used methods on level one are varieties of digital filters, model-based filters (e.g. Kalman filters), simulation techniques (e.g. particle filters), etc. The results from this level are the foundation of the higher-level data fusion.

Level two is aimed to derive features and patterns from the previous state estimates. Following the previous example, when the speed, direction and location of an aircraft are available, further information can be deduced by fusing more extensive information. The deduced information may contain the flight destination, the type of the aircraft, flying mode (e.g. auto, manual). For this purpose, quite a few statistics and inference methods are used on level two. The common methods are Bayesian methods, Neural networks, regression models, fuzzy logic, etc. It can be seen that training and learning processes are involved in these methods.

Level three can be regarded as a decision level. Data fusion on the former two levels provide some ‘facts’ concerning the observed object. These ‘facts’ may trigger a certain decision or initiate a chain of events. For example, based on the derived (fused) information (flight pattern, speed, height, destination, type of aircraft, etc), military air traffic control may infer that the plane has been hijacked and that the hijackers are up to no good. On this level, more human effects are contained and many subjective evaluation and assessment are involved. The common approaches on this level are DSS (Decision Support System), Expert System, etc.

From level one to level three, the used data and the results of data fusion may change from exact figures to language specification, and the difficulty in data fusion increases. In addition, more and more human effects are involved, which leads to more uncertainties.

3.1.2 Traffic data fusion

Also in traffic data fusion, the three-level model provides a useful categorization. On the first level, the used data are collected from varieties of traffic sensors (e.g. loop detectors, floating cars, cameras, etc), and then are fused and translated into basic traffic information such as speed, density, flow, etc. On level two, traffic prediction or incident detection models can be established on the above information by using for example Neural Networks. As a result, some tasks such as traffic state prediction, incident detection can be done on this level. The results from the two of lower levels may lead to traffic decision and management on level three of data fusion. Figure 3.1 gives the flow chart of traffic data processing with different fusion levels. Table 3.1 gives the examples for different levels of data fusion.

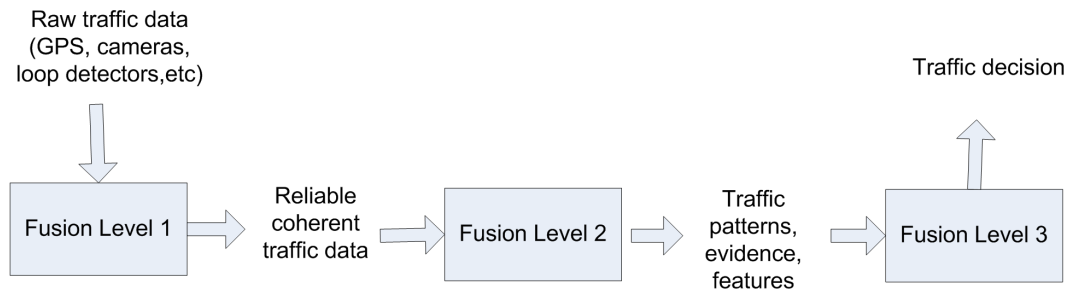


Figure 3.1: The flow chart of traffic data processing on different fusion levels

Table 3.2: Examples in three-level traffic data fusion

Level	Purpose	Typical examples in traffic
Level 1	Processing raw data	Basic state estimation (e.g. speed, flow, density)
Level 2	Deriving features and patterns	Queue length, Incident detection, State prediction
Level 3	Making decisions and detecting events	Network Management DSS tools

Components in traffic data fusion methods.

No matter what kind of a method is used in traffic data fusion, it consists of two main components, a ‘core’ and a ‘shell’, as shown in Figure 3.2.

The core represents the physical laws and assumptions in traffic theory. The physical laws, for example, can be the vehicle conservation law. The assumptions can be that the traffic is homogeneous in a certain time-space region. These laws and assumptions may lead to essential traffic models and theories, for example, first-order traffic models,

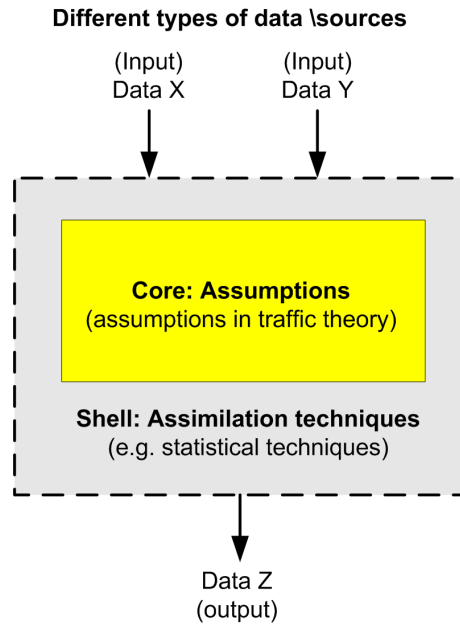


Figure 3.2: Components in level one-data fusion of raw data

fundamental diagrams, car-following models, lane-changing theory, etc. In essence, the core establishes the connections between data and data, and between data and estimated variables. In addition, physical laws are associated with assumptions though physical laws are always valid without assumption support. However, a certain physical law may not be used until certain assumptions are given. For example, the vehicle conservation law can be used for density estimation when flow measurements are assumed to be correct. However, due to the accumulated error from flow measurements, conservation law may not be used for density estimation for a long period.

The shell represents the assimilation techniques, particularly some statistical techniques, which may be able to combine models and data in statistically optimal ways. The shell ‘sticks’ to the core and needs the core to provide certain assumptions or particular models. The Kalman filter is a typical example of assimilation techniques. In order to achieve the optimal estimation, it needs the core to provide a linear model and needs the core to give the assumption of Gaussian distribution in model and measurement errors. Seen from this point of view, the shell cannot be simply separated from the core. They need to work together to accomplish data fusion.

For traffic data fusion, the total information in data will not change with the processing techniques. Better techniques may maximize the output information from the data, and present the information as true as possible. In the ‘core’ part, assumptions may distort

the true information during the fusion processing. Also more assumptions possibly mean more restrictions which prevent some types of data being included. For this reason, less assumptions are appreciated.

3.2 Assimilation techniques in the first level of traffic data fusion

This thesis only focuses on the first level of data fusion, and the fusion goal is to get more reliable and accurate traffic state estimates, particularly for speed, density and flow. The assimilation techniques for this goal are given below. Table 3.3 show the literature review of the main data fusion methods

3.2.1 Kalman filters and its variations

The most widely utilized data assimilation technique applied to traffic state estimation problems is the Kalman Filter (KF) and/or its many variations (extended and unscented KF). Kalman filter for data assimilation uses the fact that many analytical traffic models can be expressed in state-space form, that is

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \quad (3.1)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (3.2)$$

In (3.2) k depicts discrete time steps of duration $t_k - t_{k-1} = \Delta t$ seconds. Equation (3.1) depicts the process equation also known as state-transition equation, which describes the dynamics of state $\mathbf{x}_k$ (e.g. density and/or speed) as a function of $\mathbf{x}_{k-1}$ and external disturbances $\mathbf{u}_k$ (for example traffic demand at network boundaries) plus an error term $\mathbf{w}_k$, reflecting errors in the process model (e.g. model misspecification, process noises). Equation (3.2) depicts the observation equation also known as measurement equation h which relates the system state to measurements $\mathbf{y}_k$. The error term $\mathbf{v}_k$ depicts errors in either the measurement model h and/or the measures themselves. The fundamental diagram of traffic flow $q = Q^e(\rho)$, or $u = U^e(\rho)$, relating speed or flow to density, is a good example of such an measurement equation. q and u represent the flow and speed measurements from loop detectors, and ρ represents the density variables that need to be estimated.

	Used	data	Models		
	AVI data	Loop data	First-order model Second-order model Basic physic laws Fundamental diagram		
Gazis and Knapp (1971)		x	x		
Szeto and Gazis (1972)		x	x		
Kurkjian et al. (1980)		x	x		
Nahi and Trivedi (1973)		x	x		
Ghosh and Knapp (1978)		x	x x		
Chu et al. (2005)	x	x	x		
Herrera and Bayen (2007)	x	x		x	
Van Lint et al. (2008)		x		x	
Willsky et al. (1980)		x		x	
Cremer and Papageorgiou (1981)		x		x	
Kohan and Bortoff (1998)		x		x	
Meier and Wehlan (2001)		x		x	
Wang and Papageorgiou (2005)		x		x	
Wang et al. (2006)		x		x	
Wang et al. (2008)		x		x	
Wang et al. (2009)		x		x	
Cheng et al (2006)	x	x		x	
Cheng et al (2006)	x	x		x	
Mihaylova et al. (2007)		x		x	
Van Lint and Hoogendoorn (2009)	x	x	x		
Herrera and Bayen (2007)	x	x		x	
CLAUDEL and BAYEN (2008)	x	x		x	
CLAUDEL et al. (2009)	x	x		x	

Figure 3.3: Literature review: the state of the art in traffic data fusion.

If the above equations represent a linear dynamic system, f and h are the linear operators that can be expressed by matrices $\mathbf{F}_k$ and $\mathbf{H}_k$ respectively. As a result, the following equations can be derived:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \quad (3.3)$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (3.4)$$

where $\mathbf{w}_k$ is assumed to be drawn from a zero mean multivariate normal distribution with covariance $\mathbf{Q}_k$:

$$\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$$

and $\mathbf{v}_k$ is assumed to be a zero mean Gaussian white noise with covariance $\mathbf{R}_k$:

$$\mathbf{v}_k \sim N(0, \mathbf{R}_k)$$

The initial state, and the noise vectors at each step $\{\mathbf{x}_0, \mathbf{w}_1, \dots, \mathbf{w}_k, \mathbf{v}_1 \dots \mathbf{v}_k\}$ are all assumed to be mutually independent.

In what follows, let the notation $\hat{\mathbf{x}}_{n|m}$ represent the estimate of $\mathbf{x}$ at time n given observations up to, and including at time m . The state filter is represented by two variables: $\hat{\mathbf{x}}_{k|k}$ is the posteriori state estimate at time k given observations up to and including at time k ; $\hat{\mathbf{P}}_{k|k}$ is the posteriori error covariance matrix for the state estimate at time k given observations up to and including at time k . The initial conditions are given:

$$\hat{\mathbf{x}}_{0|0} = \hat{\mathbf{x}}_0, \hat{\mathbf{P}}_{0|0} = \hat{\mathbf{P}}_0 \quad (3.5)$$

With the initial conditions, a Kalman filter is iteratively executed in the two distinct steps:

1. state prediction

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \quad (3.6)$$

2. state correction

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \quad (3.7)$$

The so-called Kalman gain $\mathbf{K}_k$ in (3.7) is computed to make the Kalman filter an optimal

estimator in terms of least square.

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (3.8)$$

where $\mathbf{S}_k^{-1}$ is called innovation covariance or residual. It is worth to mention that the computation of the inverse of the matrix may be quite time-consuming. The detail can be seen Kalman (1960). It can be informally understood as

$$\mathbf{K}_k = \frac{\text{uncertainty process model}}{\text{uncertainty observation model \& data}} \times \text{sensitivity obs. model to state variables} \quad (3.9)$$

This implies that (a) the more uncertain the data are, the more weight is put on the model predictions and vice versa, and (b) that the KF adjusts $\mathbf{x}_k$ proportionally to the sensitivity of the observation model to changes in the state variables. For example, under free flow conditions the relationship between traffic density and speed is very weak, which would imply only small corrections in state variables ($\mathbf{x}_k$) even if the speeds measured by sensors ($\mathbf{y}_k$) differ largely from those predicted by the observation model ($\mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$). This intuitive structure can be easily explained to traffic operators and professionals using such state estimation tools. Moreover, the same process and observation model can be subsequently used for prediction and control purposes, given proper predictions of the boundary conditions (traffic demand, turn fractions and capacity constraints) and estimates of the model parameters are available.

Extended Kalman Filter. When the dynamic system is nonlinear, f and h cannot be expressed by matrices $\mathbf{F}_k$ and $\mathbf{H}_k$. However, the system can be linearized by computing a matrix of partial derivatives (the Jacobian) around the current estimate. The state transition and observation matrices become the following Jacobians:

$$\mathbf{F}_{k-1} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k} \quad (3.10)$$

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}} \quad (3.11)$$

Unlike the standard Kalman filter, the Extended Kalman filter is not an optimal estimator when the process model or observation model is not linear. If the initial estimates of the state or the process model is not correct, the filter may quickly diverge due to linearization.

Unscented Kalman filters. An improvement to the extended Kalman filter led to the development of the Unscented Kalman filter (UKF), which is also a nonlinear filter. In the UKF, the probability density is approximated by a nonlinear transformation of a random

variable, leading to more accurate results than the first-order Taylor expansion of the nonlinear functions in the EKF. The approximation utilizes a set of sample points, which guarantees accuracy with the posterior mean and covariance to the second-order for any nonlinearity. In addition, unlike the EKF, there is no need in the UKF to calculate the Jacobian. However, there is a demand for computing many sample points.

Details on KF algorithms and its variations can be found in many textbooks (e.g. Simon (2006), Speyer and Chung (2008)). There are a few remarks that can be made on their applicability of fusing semantically different traffic data. The key advantage of KF based state estimation approaches is that they provide a convenient and principled approach to recursively correct state estimates by balancing the errors (uncertainties) in the process and observation model and in the data.

Application of Kalman filters to traffic data fusion: Many data fusion methods for traffic state estimation take (Extended) Kalman filter as the data assimilation technique. They differ mainly in data input, data output, traffic models or assumptions.

Gazis and Knapp (1971) use time-series flow and speed from loop detectors to estimate traffic density. Basic physical laws are commonly used to approximate travel time on a road section, and then Kalman filters are applied to combine data and the model. Szeto and Gazis (1972) estimates traffic density between the two consecutive loops by fusing aggregated loop speeds and flow. The traffic model is based on the vehicle conservation law and speed-density relation. Nahi and Trivedi (1973) also uses loop flows and speeds as input data. This method contains a simpler traffic model which simply employed the conservation law, but it is able to estimate both density and speed. Ghosh and Knapp (1978) approximated the space speed over two consecutive loops by simply averaging speeds from the two. As a result, a linear state model can be achieved by exploiting the conservation law. Input being little different from the above, another contribution is reported in Kurkjian et al. (1980) managed to use loop flow and occupancy to estimate traffic density. The traffic models used in the above methods are first-order macroscopic models, and the majority of them do not consider any speed-density relation but only employ vehicle conservation law.

Since the end of 1970s, people began to use more advanced traffic models. Almost simultaneously, Willsky et al. (1980) and Cremer and Papageorgiou (1981) combined a second-order macroscopic traffic model and Kalman filter to estimate traffic states (speed, flow and density) by using loop speed and flow. Particularly, the traffic model used by the former is the Payne model. Following the similar method, Kohan and Bortoff (1998) proposed a nonlinear sliding mode observer when combining Kalman filter and a second-order macroscopic model. Also based on the Payne model, Meier and Wehlan (2001) proposed a new scheme called section-wise modeling of traffic flow which helped to approximate the boundary variables between the sections. Exploiting to the extent possible the above

approach, Wang and Papageorgiou (2005) proposed a general approach to the real-time estimation of the complete traffic state on freeway stretches, which was based on an extended Kalman filter and a second-order traffic model. In this method, the important but unknown model parameters such as free speed, critical density and exponent can be on-line estimated. The further study and applications were shown in their later publications as Wang et al. (2006), Wang et al. (2008), Wang et al. (2009).

In addition to loop data, some other types of data also can be fused by employing a Kalman filter and its variations. Chu et al. (2005) estimated traffic density and travel-time by fusing loop flow and probe car travel times over a section. In this method, the traffic inside a section is assumed to be homogeneous, and probe vehicles provide travel-times over the section that are used as measurements in Equation (3.2). The assimilation technique is adaptive Kalman filtering. Herrera and Bayen (2007) estimated density by fusing loop data and vehicle trajectory from mobile sensors. In their methods, a first-order traffic model is employed for process model. Loop flow is used as Eulerian measurement and the vehicle trajectory as Lagrangian measurement from which local density is computed.

Figure 3.4 gives an example to show the performance of Extended Kalman filter. It can be found that the results become smoother after applying EKF

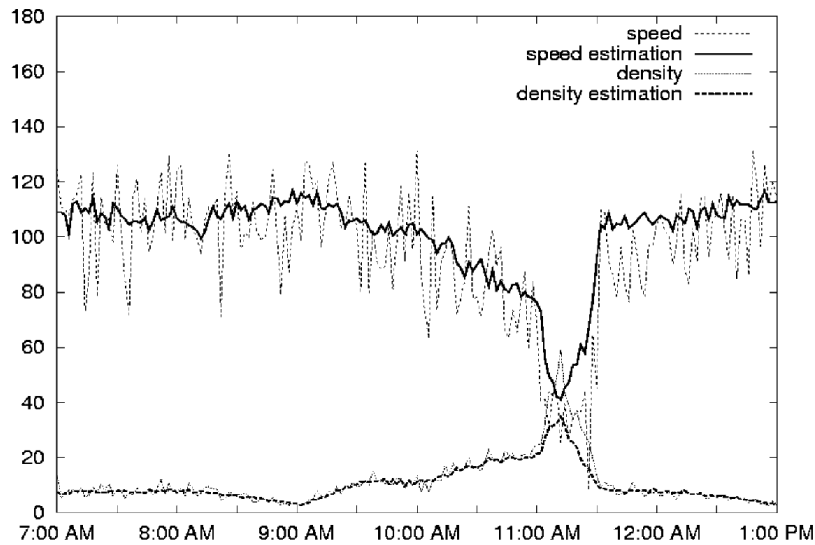


Figure 3.4: The estimation result by using an Extended Kalman filter (Adapted from: Wang and Papageorgiou (2005)).

Used Data: Loop speed and flow. Model: Second-order traffic model.

A comparison of an EKF and an UKF for traffic state estimation is investigated by

Hegyi et al. (2006) using simulated loop data. The result is shown in Table 3.3, in which J_ρ , J_v , J_{par} represent the root mean square relative errors on density, speed and model parameter estimation, respectively. This research reaches some conclusions as follows. Although the unscented Kalman filter has advantage that it propagates the state noise distribution with higher precision than the EKF, its performance was nearly equal (slightly better) to that of the extended Kalman filter. Also they find that fewer detectors result in larger state estimation errors, but have no effect on the parameter estimation error.

Table 3.3: The performance of an EKF and an UKF for different detector configuration for joint estimation. The result is from Hegyi et al. (2006)

filter type	flow loop locations	speed loop locations	J_ρ	J_v	J_{par}
EKF	1,2,3,4	1,2,3,4	0.054	0.055	0.035
UKF	1,2,3,4	1,2,3,4	0.049	0.051	0.042
EKF	1,2,3	1,2,3	0.071	0.080	0.034
UKF	1,2,3	1,2,3	0.066	0.076	0.041
EKF	2,3	2,3	0.112	0.101	0.039
UKF	2,3	2,3	0.114	0.110	0.041

3.2.2 Particle filters

Both EKF and UKF assume Gaussian distributions of the process of the noise in Equation 3.3, observation in Equation 3.4. These methods fail when the distributions are heavily skewed, bimodal or multimodal. In order to handle any arbitrary distribution, particle filters are proposed as an alternatives to the extended Kalman filter and Unscented Kalman filter when it comes to non-Gaussian distributions. Particle filters are simulation-based techniques, which are able to approach Bayesian optimal estimates with sufficient samples.

Sampling importance resampling (SIR) is a very commonly used technique and the original particle filtering algorithm proposed by Gordon et al. (1993). Like Kalman filters, Particle filters have also two phases: prediction from the previous state and correction by the current measurements. In the prediction stage, P particles of $x_k^{(L)}$ are sampled from $p(x_k|x_{k-1}^{(L)})$, where $p(x_k|x_{k-1}^{(L)})$ can be described by the process Equation 3.3. For each sample $x_k^{(L)}$, there is a confidence weight $w_k^{(L)}$. The weight can be simply updated by the measurement y_k as follows:

$$\hat{w}_k^{(L)} = w_{k-1}^{(L)} p(y_k|x_k^{(L)}) \quad (3.12)$$

where $p(y_k|x_k^{(L)})$ can be described by the observation Equation 3.4. Resampling is used

to avoid the problem of degeneracy of the algorithm, that is, avoiding the situation that all but one of the importance weights are close to zero. This algorithm approximates the filtering distribution $p(x_k|y_0, \dots, y_k)$ by a weighted set of the P particles.

$$x_k^{(L)} \sim p(x_k|y_0, \dots, y_k) \quad (3.13)$$

For state estimation, the expectation of process function $f(\cdot)$ is approximated as a weighted average

$$\int f(x_k)p(x_k|y_0, \dots, y_k)dx_k \approx \sum_{L=1}^P w^{(L)} f(x_k^{(L)}) \quad (3.14)$$

The state estimate is

$$\hat{x}_k = \sum_{L=1}^P w^{(L)} f(x_k^{(L)}) \quad (3.15)$$

Similar to Kalman filters, particle filters need a process model and an observation model. The process model can be based on various traffic models (e.g. first-order traffic model) and the observation model depends on various measurements (e.g. loop measurement). Within the framework of particle filters, Mihaylova et al. (2007) use loop speed and flow to estimate the traffic states (speed, flow and density). A second-order macroscopic traffic model is employed to establish process equations and observation equations. Cheng et al. (2006) also use particle filters to estimate traffic states from cell phone network data. In wireless communication networks, each base station is responsible for the communication service within a certain area known as cell. When a cell phone moves from one service cell to another service cell, the communication service for the cell phone will be hand over from one base station to another. The base station records the switching times so that travel time for a vehicle can be known. In this paper, such a hand-off technique is aimed to achieve the section-speed and traffic flow with known probe penetration rates given. Both a first-order traffic model and a second-order one are used, respectively for comparison. The estimated states are flow and speed.

There are some remarks on Particle filters. In Particle filters, the true posterior probability distribution can be well approximated only when there are enough particles. Therefore, if the assumptions for Kalman filters can be guaranteed, no Particle filters can outperform them. In addition, computational cost for particle filters is quite high compared to the Kalman filter and its variations.

Figure 3.5 gives an example to show the performance of a particle filter. It shows that the results become smoother when applying the particle filter.

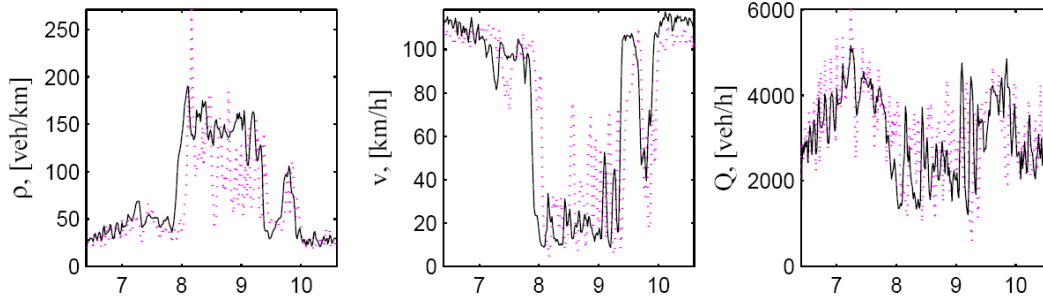


Figure 3.5: The estimation result by using Particle filters (Adapted from: Mihaylova et al. (2007)). Used Data: Loop speed and loop flow. Model: Second-order traffic model. Solid lines represent the estimates and dotted lines represent the measurements

3.2.3 Linear programming

Kalman filters or particle filters iteratively process the data by using current data and the data one time-step before the current step. One of biggest advantages of Kalman filters is that the computational and memory cost can be very low. With the development of the computer technology, large memory devices are widely available, so that a large amount of data can be processed in a batch. Thus we can use the techniques like Linear programming to input a lot of historical data to estimate the current states. Linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints. Linear programming can be employed when the problems can be expressed in canonical form:

$$\text{Minimize (Maximize)} \quad \mathbf{c}^T \mathbf{x} \quad (3.16)$$

$$\text{Subject to} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \quad (3.17)$$

When using Linear Programming (or LP in abbreviation) for traffic state estimation, the objective function is aimed at traffic states (e.g. max or min travel times), and the linear inequality constraints may be given by an appropriate combination of traffic models and data. In Claudel and Bayen (2008), Claudel et al. (2009), Linear Programming is used to estimate traffic density and travel time from loop flow and probe trajectory. The former one focuses on the theoretical part and latter one focuses on the application part. The solution of their LP model yields two objective values: minimal traffic density and maximal traffic density, that is the part $\mathbf{c}^T \mathbf{x}$, which further returns the maximal travel time and minimal travel time.

$$\text{Minimize density or Maximize density} \quad \mathbf{c}^T \mathbf{x} \quad (3.18)$$

They use the Moskowitz function to describe both vehicle trajectory and loop flow data. With the help of Moskowitz function, the vehicle trajectory and loop flow data can establish an inequality, that is:

$$\text{Trajectory data and loop data subject to } Ax \leq b \quad (3.19)$$

The Moskowitz function $M(t, x)$ (also known as cumulative number of vehicle function) represents the number of the last vehicle to pass an observer at location x before time t , and encodes the distribution of the vehicles on the highway at all times. The loop flow (loop counts from downstream and upstream) as well vehicle trajectory can be expressed by Moskowitz function as below:

- $M(t, x_{in})$: Inflow of vehicles at upstream loop
- $M(t, x_{out})$: Outflow of vehicles at downstream loop
- $\{(t, x_i(t)) | M(t, x_i(t)) = M_i\}$: The curve $(t, x_i(t))$ approximates the trajectory of vehicle labeled M_i .

In addition, the partial differential equation from conservation of vehicles

$$\frac{\partial \rho(t, x)}{\partial t} + \frac{\partial q(t, x)}{\partial x} = 0 \quad (3.20)$$

can be transformed by Moskowitz function to

$$\frac{\partial M(t, x)}{\partial t} + q\left(-\frac{\partial M(t, x)}{\partial x}\right) = 0 \quad (3.21)$$

It is worth to note that $q\left(-\frac{\partial M(t, x)}{\partial x}\right)$ is the function of $\frac{\partial M(t, x)}{\partial x}$. As a result, the Moskowitz function can link loop counts, vehicle trajectory and first-order traffic model all together. Also it is assumed that cars do not overtake each other. All of them (data, model and assumptions) finally constitute the linear inequality constraints in form of $Ax \leq b$ (see Claudel and Bayen (2008) for details). Taking the density Δ as the objective function $c^T x$, Linear Programming computes an associated possible range of density and further traces back to the possible range of travel times $[TT_{min}, TT_{max}]$ for all the vehicles.

Figure 3.6 gives an example to show the performance of Linear Programming applied in traffic data fusion.

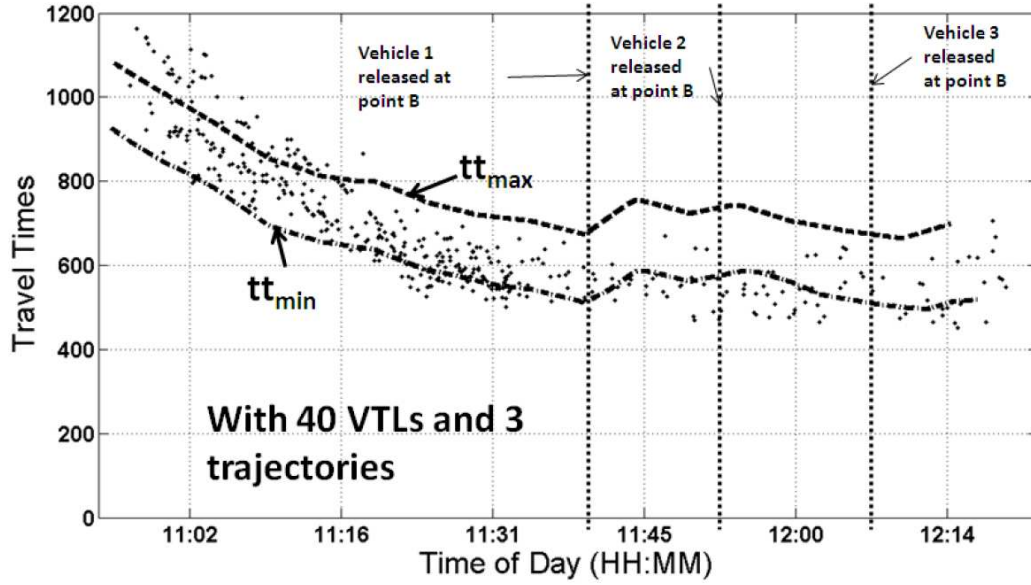


Figure 3.6: The estimated travel times by using Linear Programming (Claudel et al. (2009)). Used Data: Loop flow (counts) and trajectories. Model: first-order traffic model. Solid lines represent the estimates and dots represent the measurements

3.2.4 Treiber filter

Treiber filter is originally designed for processing single data source and reconstruct the spatio-temporal traffic map. It is proposed by Treiber and Helbing (2002) and is based on the spatio-temporal characteristics, that is, perturbations in traffic travel along the so-called characteristics (refer to Equation 3.25) with (approximately) constant characteristic speeds c_{cong} in congestion, and c_{free} under free flow conditions.

Single data source

The reconstructed quantity z at (t, x) is described as follows:

$$z(t, x) = w(t, x)z_{cong}(t, x) + (1 - w(t, x))z_{free}(t, x) \quad (3.22)$$

Equation (3.22) shows that the reconstruction involves a weighted combination using two reconstructions of the signal. The first assumes congested traffic operations (i.e. $z_{cong}(t, x)$) and the second free flow conditions (i.e. $z_{free}(t, x)$). To reconstruct $z(t, x)$ on the basis of data measured at some time and location (t_i, x_i) , the time and space dependent weights are computed as follows. First define the below functions:

$$\begin{aligned}\phi_{cong}(t, x) &\equiv \phi_0\left(t - \frac{x}{c_{cong}}, x\right), \text{ and} \\ \phi_{free}(t, x) &\equiv \phi_0\left(t - \frac{x}{c_{free}}, x\right)\end{aligned}\quad (3.23)$$

with

$$\phi_0(t, x) = \exp\left(-\frac{|x|}{\sigma} - \frac{|t|}{\tau}\right) \quad (3.24)$$

where σ and τ are parameters of the filter which describe the width and time window size of the “influence” region around (t_i, x_i) . The value of the weights of a data point (t_i, x_i) is given by

$$\begin{aligned}\beta_{cong}^i(t, x) &= \phi_{cong}(x_i - x, t_i - t), \text{ and} \\ \beta_{free}^i(t, x) &= \phi_{free}(x_i - x, t_i - t)\end{aligned}\quad (3.25)$$

The weights describe the importance of the measurement quantity z_i at the time-space point (t_i, x_i) for the value of the quantity z (to be estimated or reconstructed) at (t, x) . Loosely speaking, the weight is determined by the distance between the point (t_i, x_i) and (t, x) considering the speed at which information moves through the flow under free flow or congested conditions. To determine the value of the quantity $z(t, x)$ on the basis of the congested and the free flow filter, the weights are used as follows:

$$\begin{aligned}z_{cong}(t, x) &= \frac{\sum_i \beta_{cong}^i(t, x) z_i}{\beta_{cong}(t, x)}, \text{ and} \\ z_{free}(t, x) &= \frac{\sum_i \beta_{free}^i(t, x) z_i}{\beta_{free}(t, x)}\end{aligned}\quad (3.26)$$

Where the normalization factors are given by

$$\begin{aligned}\beta_{cong}(t, x) &= \sum_i \beta_{cong}^i(t, x), \text{ and} \\ \beta_{free}(t, x) &= \sum_i \beta_{free}^i(t, x)\end{aligned}\quad (3.27)$$

An important filter design choice is the weight factor $w(t, x)$ used in eq. (3.22). This factor describes whether the conditions in (t, x) are dictated by free flow conditions or by congested conditions or a combination of both. Treiber and Helbing (2002) propose to use speed data for this purpose and use the following expression for this weight factor:

$$\begin{aligned} w(t, x) &= \omega(z_{cong}(t, x), z_{free}(t, x)) \\ &= \frac{1}{2} \left[1 + \tanh \left(\frac{V_c - V^*(t, x)}{\Delta V} \right) \right] \end{aligned} \quad (3.28)$$

with:

$$V^*(t, x) = \min(V_{cong}(t, x), V_{free}(t, x)) \quad (3.29)$$

where $V_{cong}(t, x)$ and $V_{free}(t, x)$ are calculated with 3.26, V_c depicts a critical speed marking the transition from free to congested flow and ΔV a bandwidth around it. Note that the functions (3.28) and (3.29) are arbitrary filter design choices. Any crisp, smooth or even fuzzy function which is able to discriminate between free-flowing and congested traffic operations based on whatever data is available (density/occupancy, speed) would in principle do.

Multi data source

Van Lint and Hoogendoorn (2009) extend the Treiber filter, so that it can process the multi data source of the same type. Let $z^{(j)}(t, x)$ denote the considered traffic value as reconstructed from data source j . To fuse data from multiple data sources, we propose the following linear combination:

$$z(t, x) = \frac{\sum_j \alpha^{(j)}(t, x) \phi^{(j)}(t, x) z^{(j)}(t, x)}{\sum_j \alpha^{(j)}(t, x) \phi^{(j)}(t, x)} \quad (3.30)$$

where the second dynamic weight $\phi^{(j)}(t, x)$ is defined by:

$$\begin{aligned} \phi^{(j)}(t, x) &= w^{(j)}(t, x) \cdot \phi_{cong}^{(j)}(t, x) \\ &\quad + (1 - w^{(j)}(t, x)) \cdot \phi_{free}^{(j)}(t, x) \end{aligned} \quad (3.31)$$

The first (dynamic) weight factor $\alpha^{(j)}(t, x)$ in (3.30) can be interpreted as a dynamic indicator of the *reliability* of the data from source j at (t, x) and could for example be determined on the basis of a priori estimates of the measurement accuracy of data source j . For induction loops, where measurements become increasingly unreliable as speeds decrease, it makes sense that $\alpha^{(j)}(t, x)$ is proportional to speed. Although also location tracking equipment (e.g. GPS) is likely to make relative errors proportional to speeds, the reliability of such FCD measurements in terms of speeds would most probably still be

higher than that of induction loops.

The hyperbolic tangent function in equation (3.28) is used to calculate this weight, which reads

$$\alpha^{(j)}(t, x) = \frac{1}{\Theta_0^{(j)} [1 + \mu^{(j)}(1 - w^{(j)}(t, x))]} \quad (3.32)$$

In (3.32) $\Theta_0^{(j)}$ represents the standard deviation of the measurement error of data source j at low speeds (under congestion), and $[1 + \mu^{(j)}]\Theta_0^{(j)}$ the standard deviation under free-flowing conditions. For induction loops $\Theta_0^{(j)}$ is typically in the order of 4 km/h, and $\mu^{(j)}$ around $1\frac{1}{2}$ (yielding a standard deviation of around 10 km/h under free-flow conditions).

Figure 3.7 gives an example to show the performance of Treiber filter.

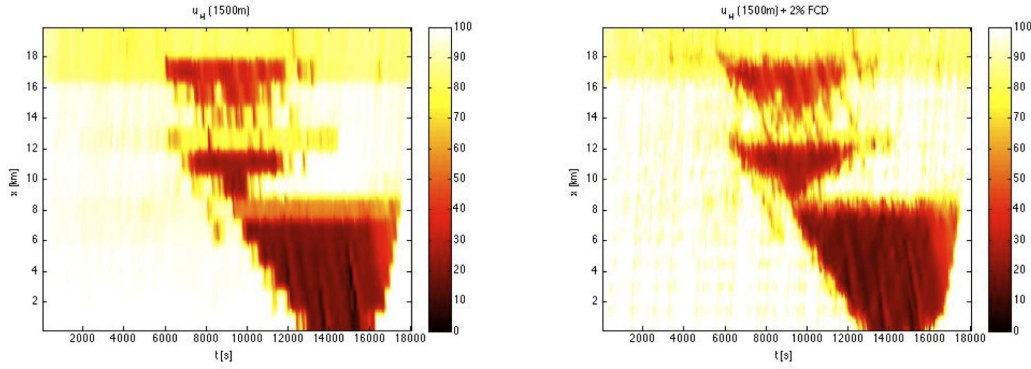


Figure 3.7: The estimated time space speed plots by using Treiber filter (Van Lint and Hoogendoorn (2009)). (left figure) Used data: loop speed only; (right figure) Used Data: speed measurements from loop detectors and floating cars. Model: Fundamental diagrams.

3.2.5 Nudging Technique

Nudging, also known as Newtonian relaxation or 4DDA, is particularly used in weather forecasting. In this data assimilation technique, model variables are driven (nuded) toward observations. A source term proportional to the difference between the predicted and observed state is included in the constitutive equation $f(z, x, t) = 0$ of the model (e.g. LWR Partial differential equation in traffic):

$$f(z, x, t) = \lambda(x, t) \cdot (z - z^o) \quad (3.33)$$

where x is the space variable, t is time, z is the state vector (e.g. density), and z^o is the observation measurements. The nudging factor $\lambda(x, t)$ vanishes away from the measurement location and after the measurement time. As a result, λ drives the solution towards the observations when the observation is made.

In Herrera and Bayen (2007), nudging technique is used to fuse loop flow and probe trajectories. In their method, $f(z, x, t) = 0$ is simply the LWR PDE:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q(\rho)}{\partial x} = 0 \quad (3.34)$$

where $\rho(x, t)$ represents the vehicle density at (x, t) and $q(\rho)$ is the fundamental diagram. The nudging technique adds a source term to the above dynamic model, leading to:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q(\rho)}{\partial x} = - \sum_{j=1}^J \lambda(x - x_j(t), t) \cdot (\rho(x_j(t), t) - \rho^o(x_j(t), t)) \quad (3.35)$$

The summation over the index j accounts for the J different vehicle trajectories that can combine the loop counts to provide measured density $\rho^o(x_j(t), t)$. A possible expression for the nudging factor can be found in Ishikawa et al. (1996):

$$\lambda(x, t) = \begin{cases} \frac{1}{T_a} \exp\left(-\frac{x^2}{X_{nudge}^2}\right) \exp\left(-\frac{t-t^o}{T_d}\right) & \text{if } x \leq \alpha X_{nudge} \text{ and } t \geq t^o \\ 0 & \text{otherwise} \end{cases} \quad (3.36)$$

where T_a is the timescale and determines the strength of the nudging factor, t^o is the observation time, and T_d and X_{nudge} reflect how the effect of any observation decrease over time and space. The coefficient α determines the zone of influence of the measurements. The nudging factor is zero outside the influence zone. In traffic context, the nudging term ‘adds’ the vehicles on the freeway when the model underestimates the density and ‘removes’ vehicles when the model overestimates the density.

Figure 3.8 gives an example to show the performance of Nudging method.

3.2.6 Summary

The summary of the above mentioned techniques is shown in Table 3.4

Figure 3.9 gives a summary of data input and data output by using the above assimilation techniques. It can found that quite a few combinations of different data are still missing.

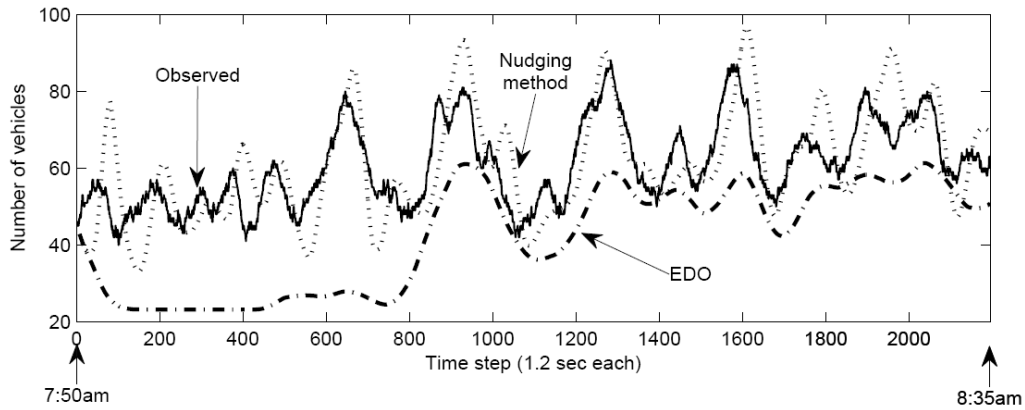


Figure 3.8: Traffic flow estimation by Nudging method (Herrera and Bayen (2007)). Used data: loop counts and trajectory; Model: First-order traffic model. Dotted lines represent the estimates when using both loop counts and trajectory, dashed lines represent the estimates when using only loop counts, and solid lines represent the measurements.

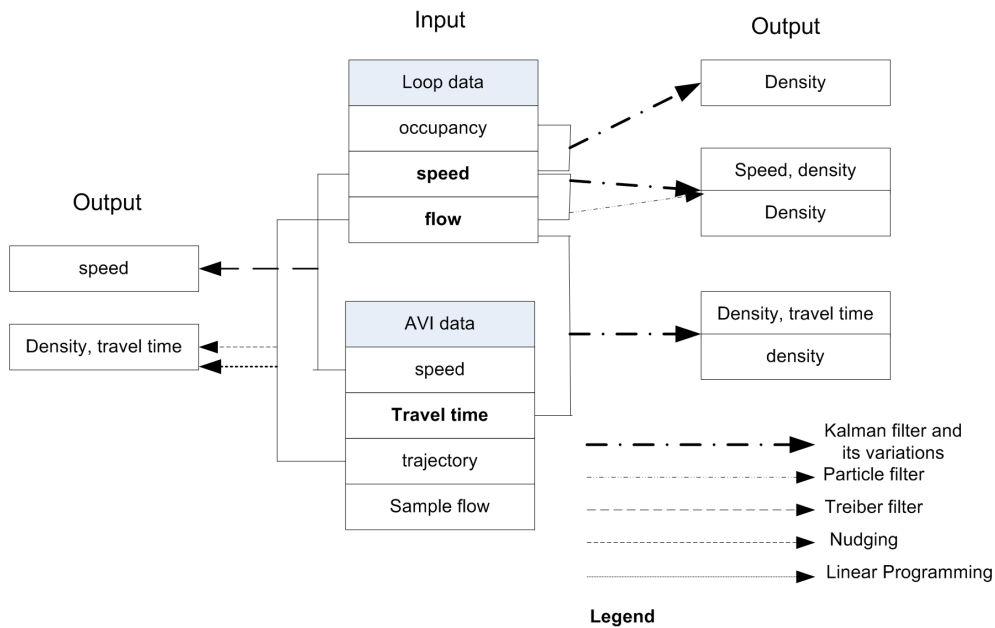


Figure 3.9: Input data and output data types in data fusion techniques of the state of the art

Table 3.4: Summary of varieties of assimilation techniques in traffic data fusion

Filter	Feature
KF	Description: Give the optimal estimation by iteratively processing data in linear and Gaussian models. Pros: The optimal solution can be achieved when models are linear and Gaussian. Computation requirements are less than of an EKF. Better than an EKF, UKF and PF in linear and Gaussian system. Cons: It cannot deal with a non-linear system
EKF	Description: This method is chosen when it comes to slightly nonlinear and Gaussian system with local approximation. It is an iterative processing technique Pros: The estimation result is quite good for slightly nonlinear system. Cons: EKF is difficult to tune, the Jacobian may be hard to derive, can only handle limited amount of nonlinearity.
UKF	Description: Based on values on a larger area; Can only be applied to models driven by Gaussian noises; A trade-off between EKF and PF; Pros: Functions don't need to be differentiable. Cons: More computations than EKF and the results is not better than EKF for slightly nonlinear system; Not a truly global approximation but based on trial points
PF	Description: An iterative processing technique like Kalman filters; can handle a highly nonlinear and non-Gaussian system Pros: No restrictions in model; Can be applied to non-Gaussian models, hierarchical models etc; Global approximation; Approaches the exact solution, when the number of samples goes to infinity. Cons: The result is not better than EKF or UKF if there are no enough number of particles; Computational requirements much higher than of the Unscented Kalman filters.
LP	Description: Process data in a large batch and achieve the optimal estimation by seeking the biggest or smallest value. The value is dependent on objective functions and subject to a linear inequality constraints. Pros: Have a simple form; Computation cost is low compared with Kalman filter and its variation. Can give a range of estimates (e.g. minimal and maximal travel time estimates). Cons: It needs a particular form, which is hard to obtain in many cases.
Nudging	Description: Work for systems (e.g. weather, traffic) which evolves over time space and can be described by differential equations. Combine model and data similar KF but no optimal estimation is guaranteed. Pros: Easy to understand and implement; Low computation cost is asked Cons: Not easy to find an appropriate nudging factor; Data used must be linked to the major variables in differential equations (e.g. density in first-order traffic model)
Treiber	Description: Treiber filter similar to image processing. Pros: Easy to understand and implement; Low computation cost. Cons: Can only fuse the data of the same type (e.g fusion of loop speed and probe car speed)

3.3 Discussion on further challenges

This section will look into the issues which have not been properly solved in the above-mentioned methods. The main problems consist in spatio-temporal alignment, parameter calibration and identification, computational costs, etc. As a result, it is hard to fuse more types of data, and the estimation results based on these methods have relative less accuracy and reliability. Considering the implementation of data fusion in large networks, less computation cost is also important.

3.3.1 The spatio-temporal alignment problem

There is, however, one major drawback of the above assimilation techniques, which relates to the spatial and temporal alignment of the data. Let us take Kalman filter for illustration of this problem. For every data source used, an (additional) observation equation (3.2) is required, which relates the data to the traffic state. This is not necessarily a problem, as long as the spatial and temporal discretization of the data (detector locations x_i and time periods of size ΔT) can be aligned with the discretization used in the model (road segments of length Δx and time periods of size Δt). For example, some spatial data can be transformed fairly easily into local measurements, such as sampled floating car data used in e.g. Herrera and Bayen (2007) and Van Lint and Hoogendoorn (2007). This, however, is not the case for e.g. travel time or journey (segment) speeds. These are available after trips are realized, that is, after a time period equal to the measured travel time (or traveled distance divided by the average journey speed). As a result, a (realized) travel time observation equation has the following general form

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-TT^{max}}) + \xi_t \quad (3.37)$$

where the output variable $\mathbf{y}_k$ now depicts (realized) travel time TT , and TT^{max} is the maximum observed travel time on the route of interest. In the observation equation, the observation function $h(\cdot)$ is needed in order to establish the relationship between the state variables and measurements TT . However, this function cannot be obtained until the correct trajectory is found. Figure 3.10 shows that one travel time may correspond numerous possible trajectories, so it is almost impossible to obtain the observation function $h(\cdot)$ only from this travel time. As a result, Kalman filter cannot assimilate the travel time in such a case. For this reason, in some research where travel time is used as measurements in Kalman filter framework, traffic on the study road has to be assumed to be homogeneous and stationary during the travel period. In Chu et al. (2005), the observation equation using

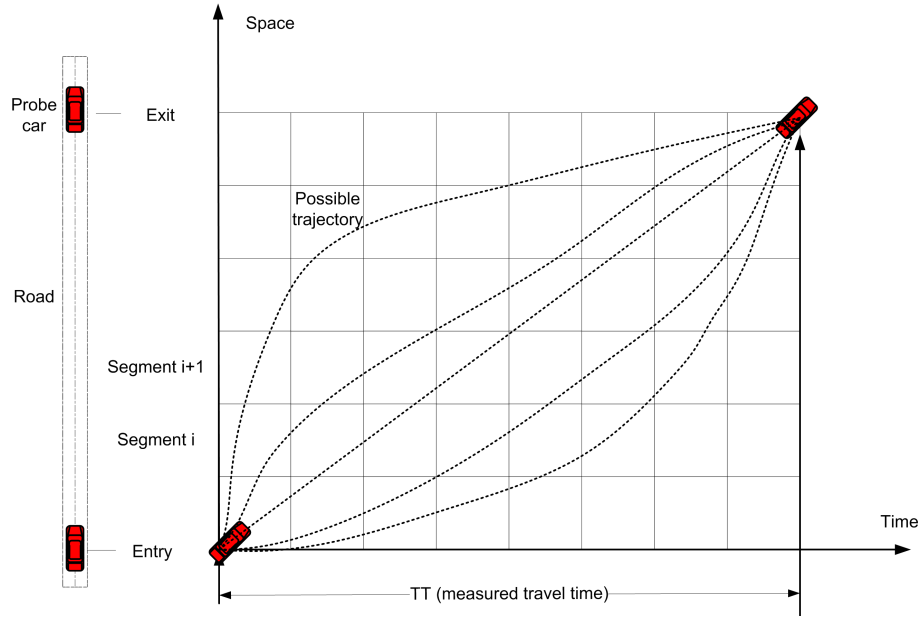


Figure 3.10: Numerous trajectories correspond one travel time record

travel time as measurement is given as:

$$TT(t) = \frac{\Delta x}{v(t)} = \frac{\Delta x \cdot \rho(t)}{q(t)} \quad (3.38)$$

where Δx is the length of the road segment, ρ is the density and q is the flow. This method can work only in some special cases (e.g. when the travel time over only one road segment is given), but if the travel times over a few road segments are given, this method does not work. Many other techniques mentioned above cannot assimilate these kinds of travel time either. In addition to travel time records, some other already-existing data sources have potential for traffic estimation but don't fit in any of these assimilation techniques. For this reason, some important combinations of different-typed data as input and output are missing, for example combining biased local speed data with travel times to estimate space speed, or even further, combining local speed, flow and travel time to estimate both space speed and density. In conclusion, when the data cannot be straightforwardly aligned, no conventional data fusion techniques can be employed.

3.3.2 Measurability and parameter identification

The main variables in traffic are density, speed and flow, which can be associated to one another by the equation $q = \rho v$. This relationship holds only when v is space mean speed. However, loop detectors fail to give space-mean speed measurements. They can only give speed measures at a certain point on a road. In any event, the space-mean speed cannot be measured by loop detectors. And in order to apply this equation on loop data, the time-mean speed measures from loop detectors have to be taken as mean-space speeds.

Apart from time-space mean speeds, traffic density cannot be measured, either. Theoretically, traffic density is easily available given initial conditions and traffic counts from loop detectors. But the accumulated errors in loop counts make the density estimates deviate far from the true results. In estimation of traffic density from loop data, fundamental diagrams have to be employed to bridge flow and density.

Majority of the above methods need to identify parameters in models, particularly parameters in fundamental diagrams. Fundamental diagrams roughly represent the relationship between the major traffic variables, and more roughly in congested traffic. To make things worse, the parameters in such a rough relationship are identified with data that are probably not reliable and accurate enough, e.g. speed measures from loop detectors. These methods have to be processed by data before they are applied on data, so the quality of these methods rely on the quality of data.

3.3.3 Computation cost

Thanks to the advanced computation algorithm and modern computers, the computation cost is not a big problem in an application of (extended) Kalman filters, if the the state vector and measurement vector have only a few hundred variables. But Kalman filters involve computation of matrix inverse as required in Equation 3.8, and the complexity of calculating the inverse of a matrix $N * N$ with the most effect and practical algorithm *Strassen* is $O(N^{2.807})$. Therefore, when Kalman filters are applied on a large traffic network with thousands of state and measurement variables, the computation cost becomes a problem. Particle filters ask for much more computation compared to Kalman filter, though they are better at processing non-linear functions and non-Gaussian probability. In Chen et al. (2004), it is found that 500 particles are needed in application of Particle filters so that the performance is as good as Kalman filters, but the computation cost is 200 times as that of a Kalman filter.

3.4 Conclusion

This chapter discusses and synthesizes the state-of-the-art in traffic data fusion. The data fusion methods are composed of two parts core and shell as we present. The core represents the assumption in traffic theory which is used to model the traffic system. The shell represents the assimilation techniques e.g. statistical techniques which are used to combine data and traffic model in an optimal way. The previously presented methods have deal with many data fusion issues, but still there are quite a few to be left unsolved. Still we need some methods which can fuse more types of data and output more reliable results. Next chapter will propose the idea of new approach “data-data consistency” which may help to solve this issue with less assumptions.

Chapter 4

Towards a new approach

This chapter will look at the traffic data fusion from a new angle, presenting the concept of a new approach called “data-data consistency”. It makes it possible to fuse more types of data but with limited assumptions. Based on this idea, the following chapters give quite a few methods (algorithms) which can effectively fuse data from different levels and output better results.

4.1 From data-model consistency to data-data consistency

The conventional data fusion techniques attempt to drive traffic models and data against each other. It means that the models are constantly calibrated to fit into the measurement data. The data are corrected through models so that they more fit into models. For state estimation, when the models are more reliable, more weight is put on models, otherwise more weight is placed on measured data. By putting certain weights on models and data respectively, the assimilation techniques make a balance between models and data. As a result, a better estimate is expected in appropriate combination of model and data. In the process of finding optimal estimates, these techniques can lead to consistency between models and data. In this thesis, these techniques can be simply categorized into so-called ‘Data-Model Consistency’ approach.

However, as seen in the previous chapter, the models that are used in classic data fusion methods need quite a number of assumptions. For example, we may have to assume that the models are unbiased and have only random errors so that assimilation techniques like Kalman filters can be used. In addition, probably we may have to assume that traffic

behaviors are the same when the same average traffic conditions are given. Based on this assumption, fundamental diagrams can be used. These methods put more attention on traffic models rather than traffic data itself. For model calibration, we will meet the issue of overfitting: the number of parameters in the model may be so large that although the model can be fitted nicely to the data, it does not generalize the data. For highly stochastic traffic system, traffic modelling will lead to large model errors: the traffic process is a complex process, which is very hard to model. Using a ‘wrong’ model may lead to model-data consistency, but not necessarily to a better estimate of the state. For example, the widely-used LWR model assumes stationarity, even if traffic is in a non-stationary situation. As presented in the previous chapter, such a data-model consistency approach have quite a few drawbacks and fail to fuse more types of data effectively. In order to tackle the above issues, we may need to shift our focus from the traffic model and a new approach has to be found.

Although traffic measurements come for many numbers of sensors, there are only very few traffic variables that need to be estimated (e.g. speed, flow, density, travel times, etc). Also these measurements can be simply categorized into speed, flow, density and travel time measurements. The relationship between one another is quite simple or can be made simple. The relationship between speed and travel times is strictly determined by simple physical laws. Based on Edie’s definition, we will have the simple equation $density = flow/speed$. We do not necessarily need those advanced traffic models. Why not just use basic physical laws and try to avoid using many assumptions? The main barrier lie in this fact that the relationship between measured data and traffic variables cannot always be expressed in explicit equations or ones in required forms. For example, the physical laws give $s = \int_0^{TT} v(t)dt$, where s depicts travel distance and TT is the travel time. Although this physics can simply deduce travel time if travel distance and speed are known, no explicit equations can be found if travel time TT is known measurement and $v(t)$ is estimated variable. Therefore, the new approach proposed in this thesis makes the use of such simple relationships and expresses them in simple forms of equations.

Data from different sources have their own characteristics. Loop detectors are able to measure traffic speeds at certain points and thus give the traffic details. But estimated local speeds derived from loop data are biased due to the over-representation of fast vehicles by loop detectors (This depends on the collection systems. At least, it is true in the Netherlands). Cameras or floating cars may be able to give the travel time, which is the aggregated result of traffic over certain time space region but it does not give the traffic detail. In contrast to the local speed estimates from loop detectors, travel time has unbiased traffic information and is able to capture the global profile of traffic. By comparison of these two data, it can be seen that each type of data and source has its particular strengthes and weaknesses, therefore the proposed approach is supposed to make the use of the strength in one type of data to compensate the weakness in another.

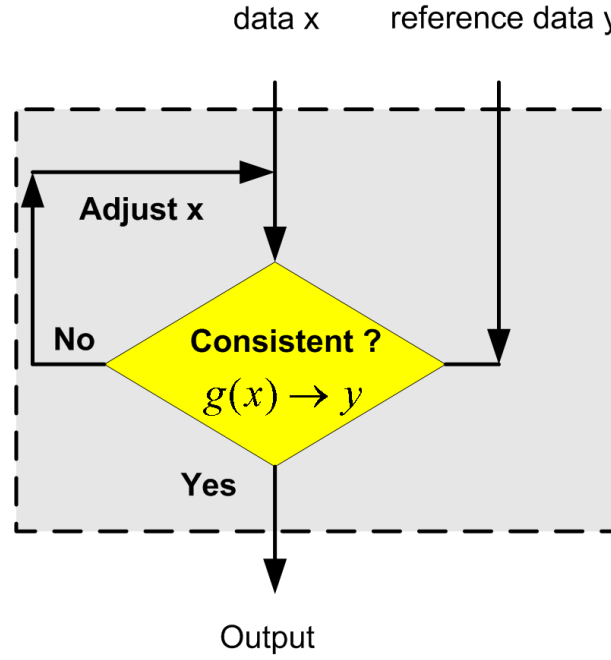


Figure 4.1: New approach: Data-Data Consistency approach.
 $g(\cdot)$ represents the relationship between data x and data y . It exists independent of data.

The proposed approach in this thesis is inspired by the above facts and it is called ‘Data-Data Consistency’ Approach. This approach still needs traffic models, but these models are simply based on some basic physical laws and very few assumptions. Unlike first-order or second-order traffic models, the used models don’t contains any parameters that need to be calibrated.

This approach can be illustrated by Figure 4.1. It takes one type of data x as prior information and the other y as reference information. The reference data y has a specific strength that data x is lacking. For illustration, let x represent local speeds from loop data and y travel time. Since travel time is unbiased information and local speed estimate is biased, an appropriate adjustment to x can be done so that the posterior estimates of local speeds are consistent with travel time information. In the adjustment, the local information that depict the traffic details is kept and it is also corrected to be unbiased by travel time. If necessary, this adjustment may be an iterative process. In the end, less biased estimates of local speeds are achieved.

4.2 Benefits of the new approach

There are quite a few benefits from this approach.

- Minimal assumptions are required. The models used in this approach are simply based on basic physical laws and fewer assumptions. The mainly-used physical laws are $distance = speed * time$ (it actually mean $\Delta s = v * \Delta t$), $density = flow / speed$ (based on Edie's definition) and vehicle conservation law. The measurement data are expected to satisfy several assumptions. For example, we assume that the travel time is unbiased and much more reliable than loop measurement. Or we may assume that the flow measurements are very reliable with less than 1% errors. Or we may assume that the measurements have Gaussian errors.
- No model parameters need to be estimated. As we know, traffic system is highly stochastic. For this reason, even on a given road segment, the model parameters will change due to all kinds of external impacts such as weather, composition of vehicles, time of day, etc. Even if these parameters can be estimated online, the quality of calibration is still determined by measurement data. Biased measurements lead to biased models, and biased models lead to biased estimation. As matter of fact, one important motivation or aim of this thesis is to provide more accurate traffic state and data for better model calibration but not vice versa.
- Less restrictions are placed. This approach seeks for consistency between different types of data. The so-called consistency can be expressed or defined in any ways as long as it makes sense. In addition, no particular forms of equation are required to formulate the relationship between data. But in some assimilation techniques like Kalman filter, for example the measurements are always expressed in the combination of variables. The state equation must be express in an incremental and iterative way. The linearization of nonlinear equations also brings about inconvenience.
- More data sources can be better used. Since there are no particular forms to formulate the relationship between the data. Any two types of data can be fused by this approach as long as there is a link between the two. Besides, each type of data can only cover one aspect of traffic, and have its own strength and weakness. This approach tends to combine the strength in data and compensate the weakness in data, so the better use of data can be achieved.
- Compatible with conventional data fusion techniques. Sometimes it is difficult to choose among different fusion techniques. It is hard to decide which one is better. When one method is chosen, then the rest have to give way. Data-Data Consistency approach is an addition to the conventional methods more than an alternative. This

approach aims to do some data fusion that is hardly done by other techniques. The result from this approach can be used to establish a more accurate model that for example Kalman filters need. Also the output estimates from the other techniques also can be further used in approach to do further fusion work.

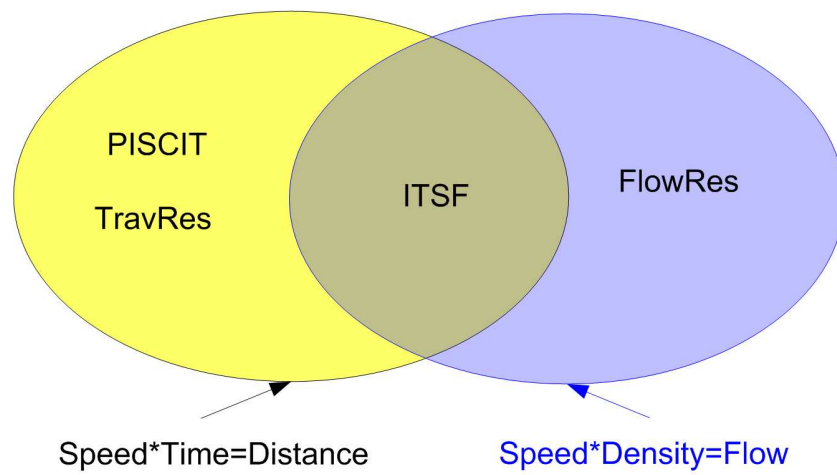
- More efficient implementation. It can be seen from Figure 4.1, that the whole fusion process is simple and thus easily understood. There is no high computation cost for things like the inverse of a high-dimensional matrix. This approach uses ‘feed-back’ strategy in which the data are iteratively adjusted or corrected. Such iteratively processing is just where computers are specialized in. As a result, less time cost for computation is required.

4.3 Methods based on this approach

Following this approach, a few methods are designed to do different data fusion work. So-called PISCIT is able to achieve much less-biased local speed estimation from biased local speed measurements and travel time. TravRes is aimed at the same goal when few loop data or none is available. FlowRes is a theoretical framework which can combine low-resolutioned positioning data from e.g wireless communication networks. So-called ITSF can give both density and speed estimated by fusing flow, local speed and travel times. All of the data used in the methods are very common or can be available in large quantities. The fusion of these data has not been well done in the previous work as seen in Chapter 3.

As Figure 4.2 shows, the advanced traffic models with a number of assumptions are not required but ‘speed * time = distance’. The third mainly uses the ‘model’ that is ‘speed * density = flow’. The last method is based on both of the models.

This chapter simply gives the idea behind the approach. Following that, the next few chapters will give the details of these methods, the implementation and the results.



different physical models in data-data consistency approach

Figure 4.2: The new methods are based on different physical models

Chapter 5

Trajectory reconstruction by using travel time and local speed

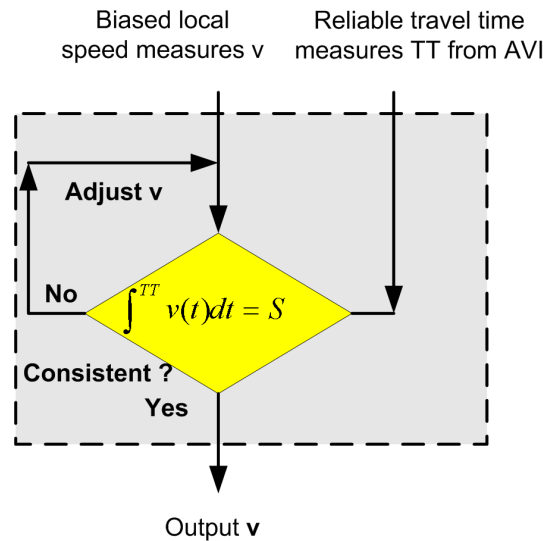


Figure 5.1: Illustration of how data-data consistency approach works for biased local speed measures and reliable travel time measures

As shown in Chapter 2, the local speed measures from loop detectors have considerable bias depending on the speeds. Such a bias may lead to an error up to 100% in density estimation, also lead to biased estimation of travel times. In contrast with loop speeds, travel times from AVI system are statistically unbiased. Further considering the intrinsic relations between time, space and speed, it is a good solution that the travel times are used to remove the bias in loop data.

However, as shown in Chapter 3, fusing travel times and local speeds is quite difficult and no appropriate methods have been proposed so far to solve this issue. The main challenge is that the different data have different semantics over space and time. A travel time measurement equals average journey speed (of a single vehicle) over a stretch of road during a variable time interval with a length equal to the travel time itself, whereas for example a local average speed gives an average over a fixed and predefined time interval, which has meaning only over a very small region in space (a cross section). This local average can not simply be added, subtracted or even compared to spatial traffic quantities. But a solution of this issue is of practical relevance in case of the (potentially) wide availability of loop data and travel time data.

Based on the idea of the data-data consistency given in Chapter 4, this chapter will give an algorithm which is able to reconstruct vehicle trajectories by fusing travel time and local speeds. The reconstructed trajectories are much less-biased and thus return less-biased time-space speeds. Figure 5.1 illustrates the main rationale of the data fusion algorithm we will describe in this chapter. The idea is that the local speed measures $v(t)$ over the time-space region are indirectly adjusted to become consistent with each other by applying an iterative algorithm. The consistency here means that the given travel times can be almost exactly derived from the update time-space speeds under the physics law $speed * time = distance$.

The main symbols used in this chapter are listed in Table 5.1.

5.1 Introduction

5.1.1 Analysis of fusing travel time and local traffic data

Data fusion of travel times (e.g. derived from stamping-time provided by cameras) and local traffic speeds (from e.g. loops) is not as easy as it may appear. Travel times and local traffic data have different levels as shown in Figure 5.2. Local traffic data shows the traffic information in the discretized cells. But travel times may represent the traffic information over a large area of time-space region when the polling rates are quite low. In contrast to other traffic information such as traffic flow, density and speed, travel times may be regarded as a kind of integral of its experienced traffic speeds over the travel-space region, which is mathematically represented by $TT(t) = f(x_{[t, t+TT]})$ where f can be determined by using the fact $speed * time = distance$.

From a mathematical perspective, travel time estimation based on traffic speed information represents the projection of high-dimensional data (speeds) into low-dimension

Table 5.1: Symbol list

L_j	Length of road segment j
$TT^{(k)}$	Measured travel time for individual vehicle on a specified road stretch (link)
$tt_j^{(k)}$	Sub-travel time for vehicle k to traverse segment j
$\hat{t}t_j^{(k)}$	Estimated sub-travel time for vehicle k to traverse segment j
$t_j^{(k)}$	Moment when vehicle k starts to travel at segment j
$\hat{t}_j^{(k)}$	Estimated moment when vehicle k starts to travel at segment j
$\hat{v}(i, j)$	Ground-truth traffic speed at segment j during period i
$\hat{v}^-(i, j)$	Prior estimated speed at segment j during period i (before fusing TT)
$\hat{v}(i, j)$	Posterior estimated speed at segment j during period i (after fusing TT)
$\hat{v}^{(k)}(i, j)$	Estimated speed for vehicle k at time-space cell (i, j) , where i and j indicate the time and location respectively
$\hat{s}^{(k)}(i, j)$	Estimated travelled distance by vehicle for vehicle k at time-space cell (i, j)
$\hat{n}(i, j)$	Number of reconstructed trajectories that traverse the time-space cell (i, j)

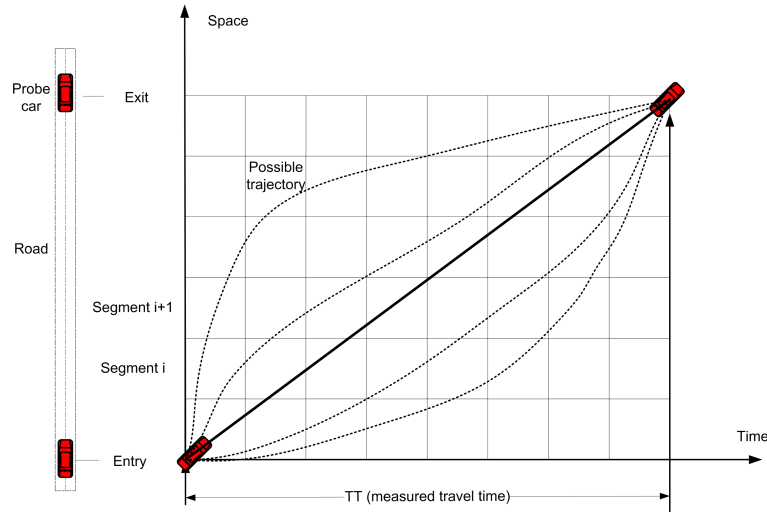


Figure 5.2: Travel time and local speed data have different levels.

space, shown in Graph A in Figure 5.3. This projection is straightforward and many ready-use algorithms have been proposed to this end such as the piece-wise linear speed-based (PLSB) trajectory algorithm proposed in Van Lint and Van der Zijpp (2003). In this algorithm imaginary vehicle trajectories are drawn through a space-time grid, based on a number of assumptions:

- A space-time grid of cells (i, j) is constructed in which loop detectors are located at each cell boundary j .
- Traffic conditions are assumed stationary within each cell (i, j) . Speeds at detectors are assumed to be averaged harmonically (to counter effect overestimation due to the over-representation of fast observations in local time samples). If this is not the case and only time averaged speeds are available these need to be corrected for this bias.
- The slope of each trajectory (i.e. the speed of the imaginary vehicle) in a cell (i, j) is considered a convex linear combination of the speeds measured during i at detectors j and $j + 1$.

The resulting travel times can be easily derived from the start and end times of these imaginary vehicle trajectories. Clearly, a number of strong assumptions are made in this heuristic method. First traffic conditions are assumed stationary (so constant over each period i), and secondly, speeds are assumed to change linearly from one detector location to the next. In case of for example passing shock waves due to congestion downstream it can be easily seen these assumptions do not hold, particularly if detectors are widely spread or data from large aggregation intervals is available. Moreover, loop data exhibits both structural and random errors (see e.g. Van Lint and Van der Zijpp (2003), Lindveld and Thijs (1999)). Their study shows that this leads to travel times which are at best in a 5% range around true realized travel times, but which become quickly more biased as fewer loops are available or traffic conditions in between loops are more heterogeneous and non-stationary. Recently more advanced travel time estimation techniques have been proposed which use spatiotemporal filtering as seen in Van Lint (2010) and Kesting and Treiber (2008).

The inverse process (from travel times to section speeds) is much more difficult, particularly when nothing but travel time information (from AVI systems) is available. For instance, trajectories A and B in Graph B in Fig 5.3, both result in exactly the same travel time. The inverse problem is hence undetermined, which implies that to solve it additional information (e.g. from induction loops) is required, which implies that to solve it additional information (e.g. from induction loops) is required.

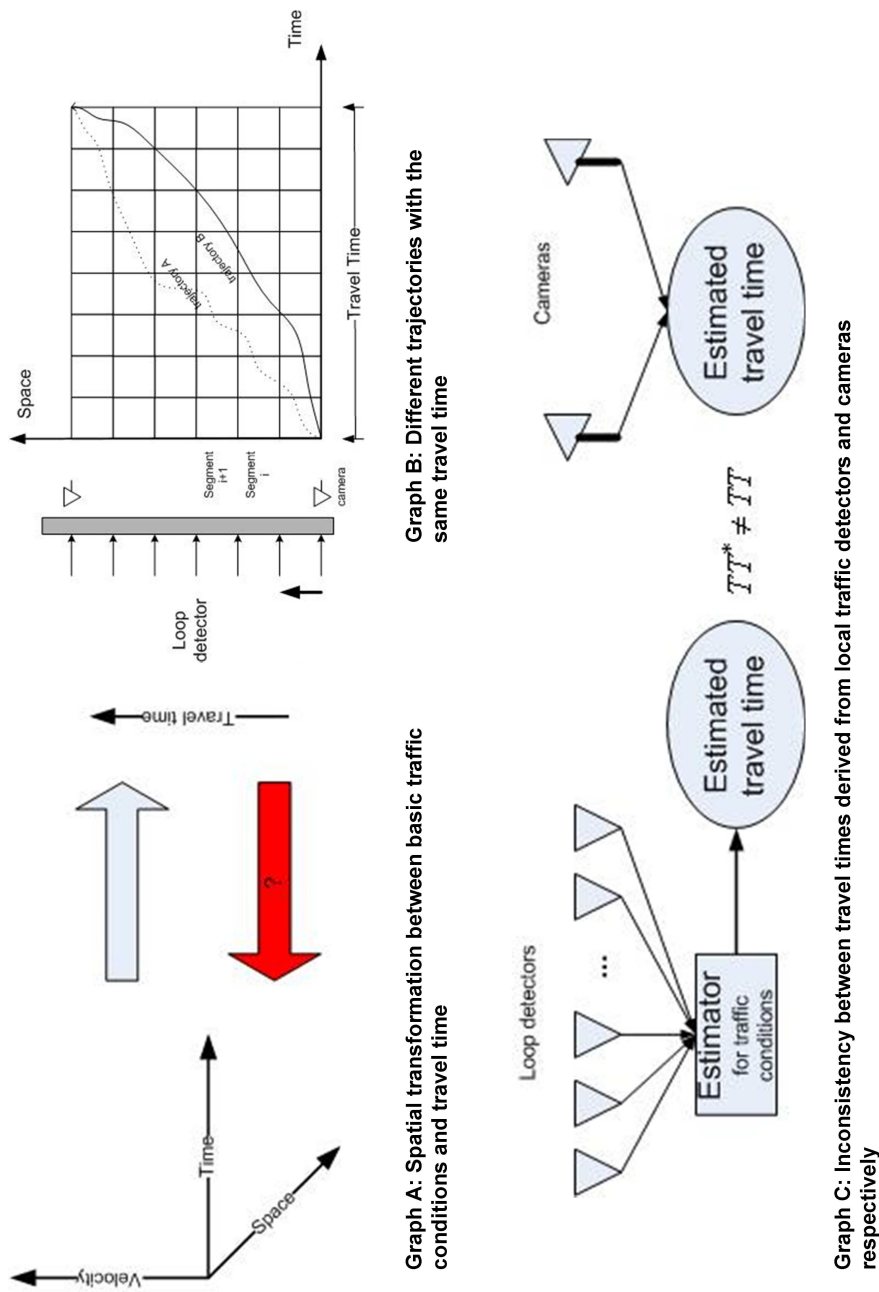


Figure 5.3: Data fusion of travel times and local traffic data

5.2 Methodology: PISCIT

In this section we present this two-step algorithm named PISCIT (Piece-wise Inverse Speed Correction by using Individual Travel-time) for fusing individual travel times with an initially estimated time-space speed contour plot. This initial time-space contour plot of speeds may be the result of a simple interpolation between consecutive detector measurements, or even the result of a model-based state estimator as in (Van Lint and Hoogendoorn (2007), Wang and Papageorgiou (2005)).

5.2.1 Framework of PISCIT

The main assumption underlying the PISCIT algorithm is that travel times measured with cameras (or other automated vehicle identification (AVI) systems) have errors which are substantially smaller than travel times estimated from local traffic data such as inductive loop-data. Particularly, travel time measurements are assumed to be unbiased. The errors in an estimated travel time induced from loop speeds are proportional to this travel time. In other words, longer travel time, more errors. Also we assume that traffic is homogeneous and stationary in each cell of time-space region. The physical law used in this method is as simple as $distance = speed * time$.

The algorithm (schematically outlined in Figure 5.4) consists of two steps:

Step 1 In the first step, approximate vehicle trajectories are reconstructed based on an initial time-space speed contour plot, and individual travel times. The ingredients for step one are

1. A initial (prior) time-space speeds (visualized as contour plots)
2. **individual** travel times

Step 2 In the second step, all the approximated trajectories from the first step are used to re-estimate (correct) the speeds from the initial (prior) speed contour plot. The result is a posteriori time-space speeds (visualized as contour plots), which fit best with all the estimated trajectories.

5.2.2 Step one: reconstruction of individual vehicle trajectories

The measured travel times provide (virtually error free) entry and exit times of the approximate vehicle trajectory. These entry and exit times (literally) provide the constraints

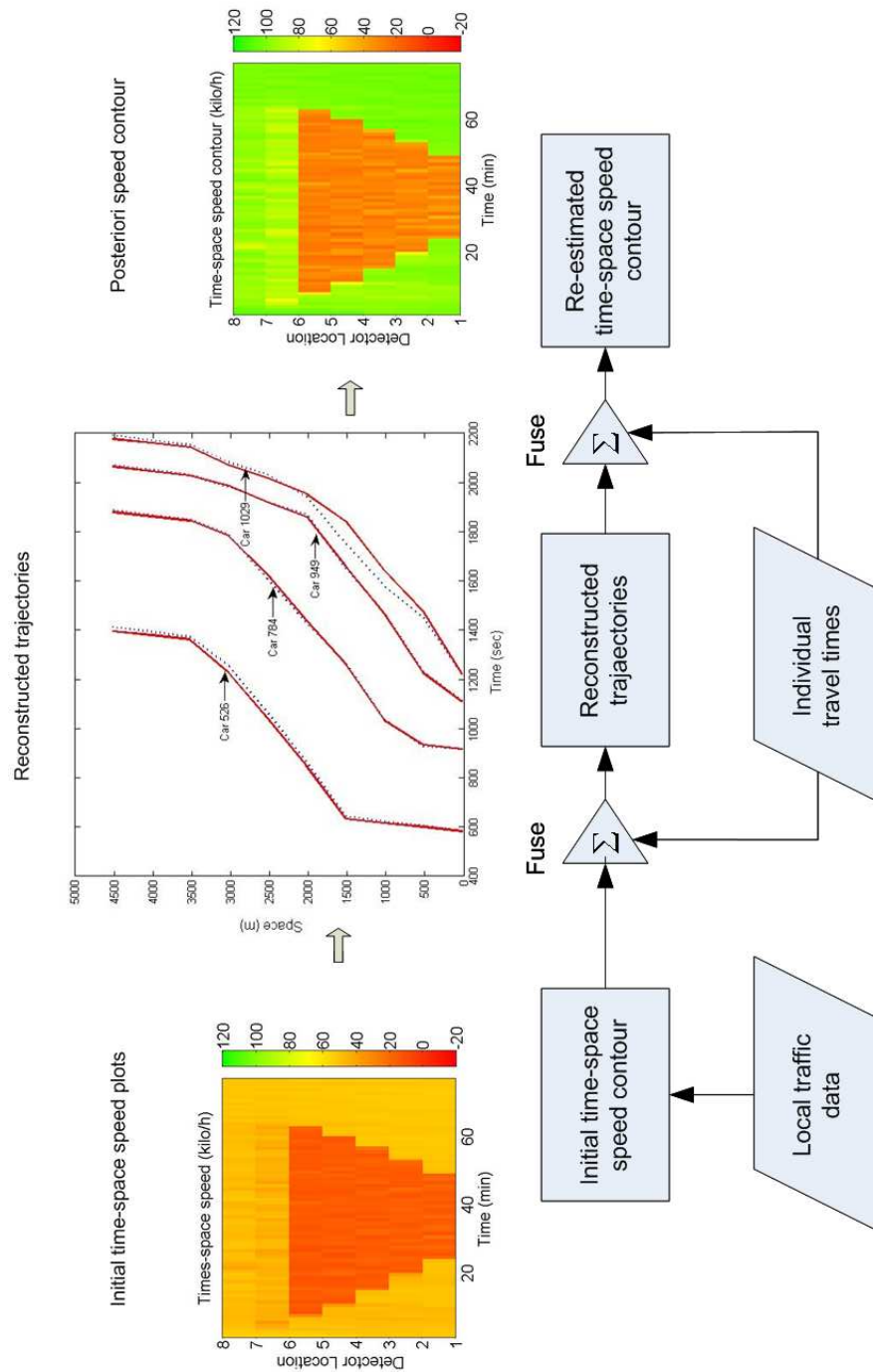


Figure 5.4: The framework of PISCIT

for each approximated vehicle trajectory over time. The loop data provide the information which determines the slope of each trajectory over space. Consequently there is inconsistency between two kinds of travel times. For example, there is inconsistency between AVI travel times and loop travel times. In order to compromise such inconsistency, loop travel time on each segment is to be proportionally adjusted. In other words, the total error is proportionally distributed to each segment if individual error characteristic on each segment is unknown. Particularly, this step is a heuristic process: starting from a straight line (with slope route length / measured travel time) an optimal vehicle trajectory is reconstructed on the basis of the additional data from loops.

Below we point out the relationships between the relevant variables:

$$\begin{aligned} t_{j+1}^{(k)} &= t_j^{(k)} + tt_j^{(k)} \\ \hat{t}_{j+1}^{(k)} &= \hat{t}_j^{(k)} + \hat{t}t_j^{(k)} \end{aligned}$$

The main idea behind the approach in step one is to use so-called proportion-multipliers to repeatedly correct sub-travel times at every segment (based on the prior speed contour plot) such that the sum of the sub-travel times satisfies the total travel time on the whole link (based on the measured travel times). As starting a point we assume that the vehicle is driving with a constant speed over segment j . This yields the following estimation $\hat{t}t_j^{(k)}$:

$$\begin{aligned} \hat{t}t_j^{(k)} &= TT^{(k)} * L_j / L \\ \text{where } L &= \sum_j L_j \end{aligned} \tag{5.1}$$

that is the initial estimate of a vehicle trajectory is the straight line shown in Graph A in Figure 5.5. Based on $\hat{t}t_j^{(k)}$ and the entry moment $t_j^{(k)}$, $\hat{t}_{j+1}^{(k)}$ can be calculated.

Next, we will make use of the prior-estimated time-space speed contour to correct the estimate of $\hat{t}t_j^{(k)}$. Since higher speeds imply less sub-travel time, the following iterative update rule is applied:

$$\hat{t}t_j^{(k)} \propto 1/\hat{v}^-([t_j^{(k)}, \hat{t}_{j+1}^{(k)}], j) \tag{5.2}$$

with constraint

$$TT^{(k)} = \sum_j \hat{t}t_j^{(k)} \tag{5.3}$$

In which $j = 1, 2, \dots, n$ and n is the (arbitrary) number of segments divided on the link.

When the Equation 5.2 and Equation 5.3 are iteratively executed, the reconstructed vehicle

trajectories approximate the true trajectory to some extent (see Graph A in Figure 5.5). The iterative execution will stop when the difference between previous values and current ones is smaller than some preset threshold (a small number).

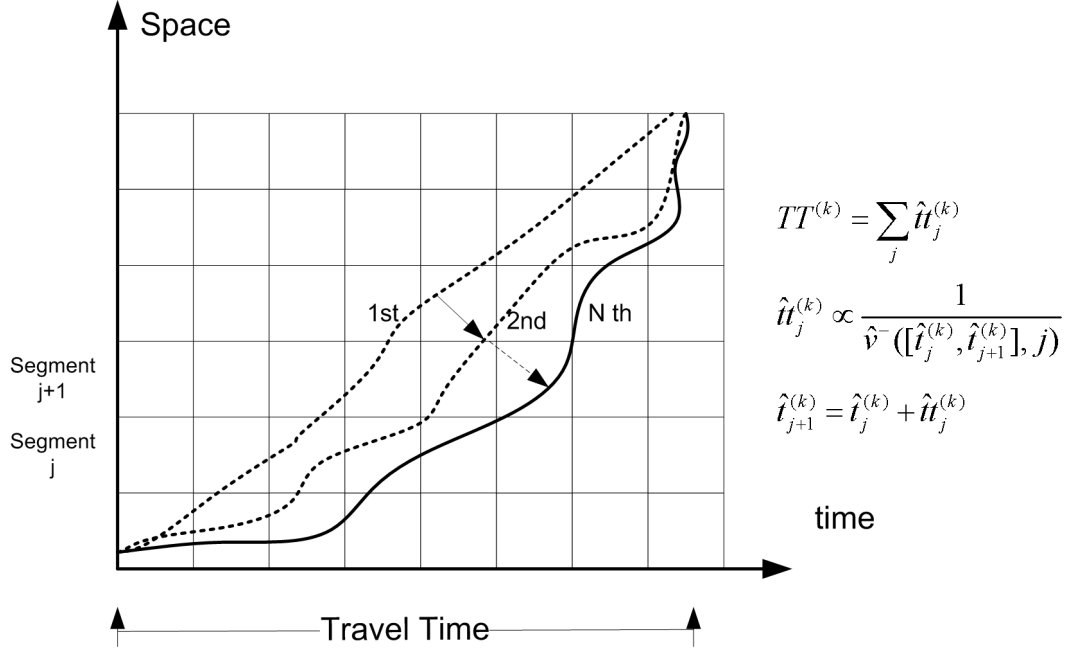


Figure 5.5: Recursive reconstructions of vehicle trajectories

5.2.3 Step two: speed re-estimation

In this section, the estimated vehicle trajectories obtained in the first step are used in turn to correct the speeds in the initial (prior) speed contour plot. To illustrate the idea, consider the case when only one trajectory passes through a particular region. Clearly, the posteriori speed in that region equals the slope of this part of the trajectory. In case many trajectories pass through the same region as seen in Graph A in Figure 5, it becomes impossible to satisfy all the trajectories, and a best fit mean speed is estimated. To tackle the problem, we introduce a simple and effective linear regression technique with constraints.

First of all, for each segment (i, j) we collect the estimated speeds $\hat{v}^{(k)}(i, j)$ and the approximated traverse lengths $\hat{s}^{(k)}(i, j)$ for each vehicle k which traversed region (i, j) , as illustrated in Graph A in Figure 5.6.

The average speed in each segment (i, j) can be derived by simply averaging these speeds $\hat{v}^{(k)}(i, j)$. However, in doing so, this results in a corrected (posteriori) speed contour map

which may result in travel times which no longer equal the measured travel times. To make consistency, we add the below constraint:

$$\sum_{(i,j) \in Traj(k)} \hat{s}^{(k)}(i,j)/v(i,j) = TT^{(k)} \quad \text{for } k = 1, 2, 3 \dots m \quad (5.4)$$

where m is the number of vehicles whose travel times are available, $v(i,j)$ denotes the variables of speeds over the time-space cell (i,j) and $Traj(k)$ is the set of cells in which vehicle k traverse during its trip. Now assume the following relationship between the speeds of each approximated vehicle trajectory (from step one) and the a posteriori speed contour map:

$$\left(\sum_k^{n(i,j)} 1/\hat{v}^{(k)}(i,j) \right) / n(i,j) = 1/v(i,j) + e(i,j) \quad (5.5)$$

where $n(i,j)$ is the number of reconstructed trajectories that traverse region (i,j) , and $e(i,j)$ is an estimation error which is assumed zero mean normally distributed.

In theory, it is possible to use equations (5.4) and (5.5) to give the optimal estimate of $1/v(i,j)$. To simplify computation, however, we propose to group these vehicle trajectories into subsets which share identical (i,j) regions along their route. For example, in Graph B in Figure 5.6, trajectory A and B belong to the same group while trajectory C belongs to a different one. Suppose that the classified trajectory groups are $TrajGroup(r)$ for $r = 1, 2, 3 \dots$, then equation (5.4) becomes:

$$\sum_{(i,j) \in TrajGroup(r)} \hat{l}^{(k)}(i,j)/v(i,j) = \sum_k^{TrajGroup(r)} TT^{(k)} \quad \text{for } r = 1, 2, 3 \dots \quad (5.6)$$

We now have the necessary ingredients to optimally estimate $1/v(i,j)$ (instead of $u(p,i)$ for mathematical purposes) for all regions (i,j) . To this end we cast the problem as a *Linear Model subject to Linear Restriction*, which is formulated below. First of all, consider a general linear regression equations:

$$\begin{aligned} Y &= X\beta + \varepsilon \\ \varepsilon &\sim N(0, \sigma^2 I) \end{aligned} \quad (5.7)$$

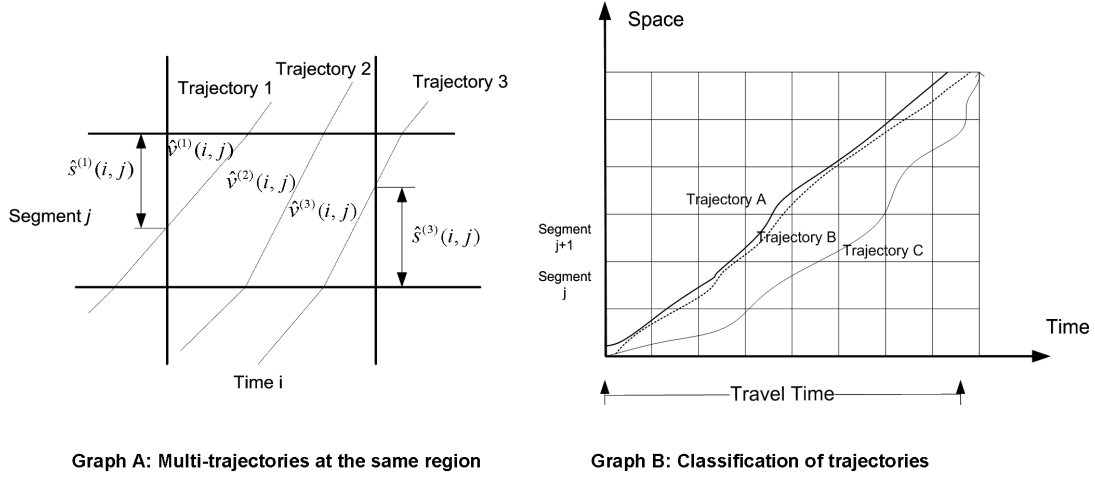


Figure 5.6: Illustration of individual vehicle trajectories

where Y is a $N \times 1$ vector (observations), X is a $N \times k$ matrix, β is $k \times 1$ (parameters to be estimated), and ε is an $N \times 1$ vector of Gaussian distributed zero-mean random variables, which reflects the random errors produced by the model. A general linear reduction (a linear restriction) in the parameter space from k to $k - m$ can be written as:

$$H'\beta = h \quad (5.8)$$

where H' is $m \times k$ matrix and known and the rank of H is m , and h is an $m \times 1$ vector (observation). Then the least square estimate of β under such a restriction as shown in Equation 5.8 or specifically Equation 5.6 is:

$$\hat{\beta} = \tilde{\beta} - (X'X)^{-1}H'(H(X'X)^{-1}H')^{-1}(H\tilde{\beta} - h) \quad (5.9)$$

where $\tilde{\beta} = (X'X)^{-1}X'Y$.

In order to apply Equation 5.8 to Equation 5.5, let

$$\beta = \text{Vec}[1/v(i, j)]_{i \times j} \quad (5.10)$$

where Vec is an operator to vectorize a matrix, that is to reshape a matrix such that it becomes a vector. Corresponding with β , Y may be easily produced with elements $\left(\sum_k^{n(i,j)} 1/\hat{v}^{(k)}(i, j) \right) / n(i, j)$. Similarly, H and h are easily determined according to Equation 5.6. Particularly, in this case X has a very simple form that reads $X = I$. Conse-

quently, the least square estimation $\hat{\beta}$ also has a simple form as:

$$\hat{\beta} = Y - H'(HH')^{-1}(HY - h) \quad (5.11)$$

where there is quite computation cost in the term $(HH')^{-1}$. After all trajectories are classified, the dimensions of HH' are greatly reduced. Finally $\hat{v}(i, j)$ can be obtained by:

$$\hat{u}(i, j) = \hat{\beta} \quad (5.12)$$

5.3 Validation

In this part, we will validate this algorithm by using synthetic data. In the first place, the synthetic ‘ground-truth’ data are generated by assuming the real loop data are true, and then the observed data are generated by tampering the ‘ground truth’ data. In the second part, the proposed algorithm is applied on the observed data and returns the estimated data. The performance of this algorithm is shown by comparing the ‘ground-truth’ and estimated results.

5.3.1 Experiment setup & data generation

First of all, a 9.5 kilometer stretch of 3-lane Highway A4 eastbound in Netherlands is considered (Graph (a) in Figure 8.7), where 18 loop detectors are placed spacing around 500 meters and aggregated traffic speed measures and counts every one minute.

- **Ground-truth speed** We assume the loop detectors give the ground-truth speed measures over certain segments. The resulting time-space speed contour plots (Figure 5.8) shows 5 hour traffic condition on this stretch from 6:00 A.M. till 11:00 A.M. on July 8th, 2008, during which congestions onset and dispersed twice.
- **Observed speeds** The observed speeds in each time-space cell are assumed by tampering the ground-truth speeds with the below assumption. This assumption is based on the empirical study in Knoop et al. (2007)

$$v^o = e^{1.1v^g(0.5-0.5v^g/120)} \quad (5.13)$$

where v^o is the observed speed and v^g is the ground-truth speed. With this assumption, the observed speed is 10% higher when ground-truth speed is 120km/h, and

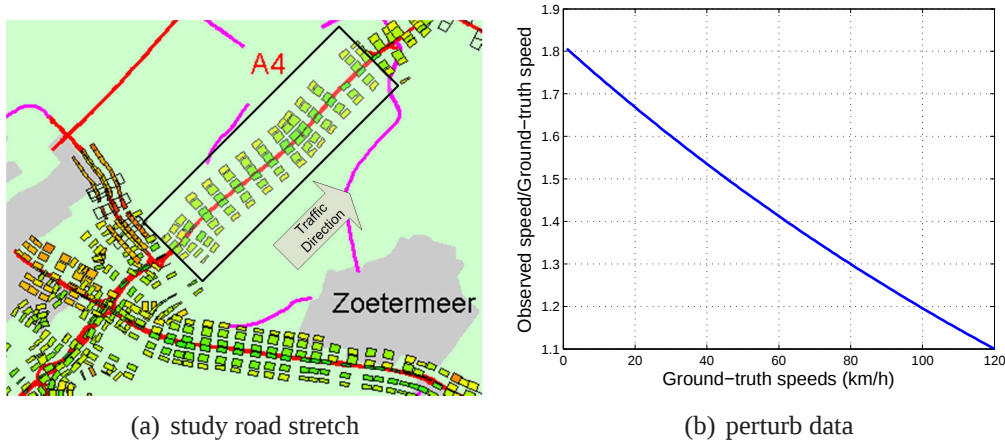


Figure 5.7: Illustration of the study road and how the ground-truth data are perturbed

70% higher at the speed of 20km/h. The resulting observed time-space speeds are shown in Graph (a) in Figure 5.9. The relationship between them is shown in Graph (b) in Figure 8.7.

- **Travel times** The travel times are generated by sampling the ‘ground-truth’ time-space speed plots. There are three virtual cameras placed at the entry, exit and middle of the whole road stretch. It is assumed that 10% of vehicles are captured by the cameras, giving the travel times from milepost 0km to 4.8km and others from 4.8km to 9.5km.

5.3.2 Results

We use mean absolute relative error (*MARE*) to evaluate the results. The definition of *MARE* is shown in Equation 8.16.

$$MARE = \frac{1}{M * N} \sum_i^M \sum_j^N \frac{|\hat{x}(i, j) - x(i, j)|}{x(i, j)} \quad (5.14)$$

$\hat{x}(i, j)$ represents the estimate and $x(i, j)$ represents the ground-truth quantity. The comparison of the results without and with using the algorithm can be seen in Figure 5.9.

- **Before** The observed speeds and density have large errors. *MARE* for the observed speeds is 33.4% in the given scenario.

Table 5.2: Comparison of MARE before and after using the proposed algorithm.

Measure type	MARE (before)	MARE (after)
MARE on speeds (global)	33.4%	4.8%
MARE on low speeds (<50kmph)	64.6%	10.8%
MARE on travel times	26%	3.5%

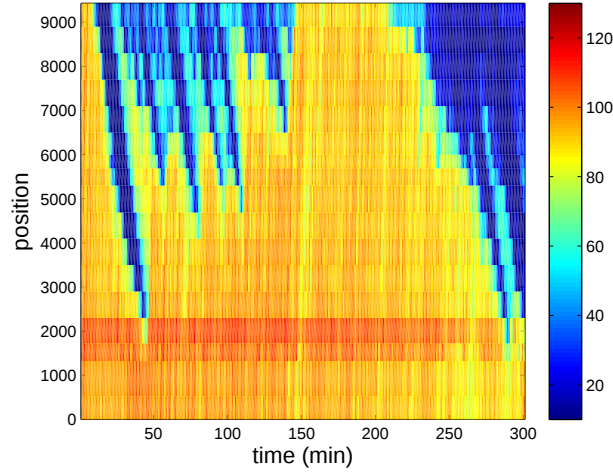


Figure 5.8: Ground-truth time-space speed plot

- **After** After the proposed algorithm is applied to fuse the observed speeds, travel times and flow, the above errors remarkably decrease. MARE for estimated speeds becomes 4.8%.

With time-space speed plots, travel times can be easily derived. Figure 5.12 makes comparison of travel time estimates between before and after using this algorithm. Before using it, the travel times based on observed speeds have mean absolute error of 202 seconds and MARE 26%. After using it, the travel times have a much smaller error of 32 seconds and MARE 3.5%. In Figure 8.14, the thick green line represent the ground-truth travel time, dark dashed line represents the results after using the algorithm and thin red line represents the travel time estimation based on the observed speeds. The former two lines almost overlap with each other.

Next, we study how the added floating car data influence the speed estimation when they are added into travel times from camera data. The travel time data from camera remain as above, that is 10% vehicles are captured by fixed cameras. The reporting rate of floating car data is 60 seconds. The penetration range from 0% through 10%. The impacts of

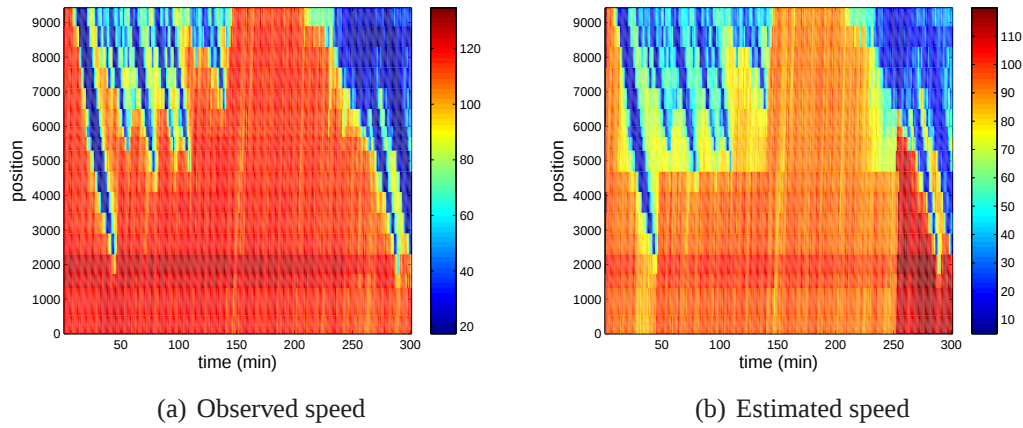


Figure 5.9: Comparison between observed speeds and estimated speeds after applying the proposed fusion algorithm

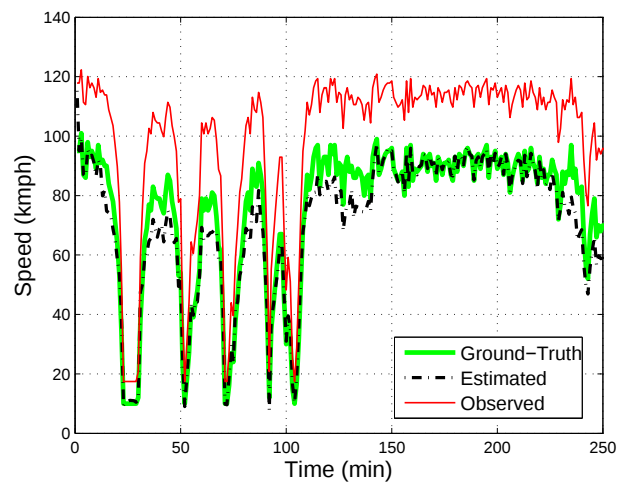


Figure 5.10: Comparison between ground-truth speeds, observed and estimated ones on road-segment 12 (around 5.5km milepost)

added floating car data can be seen in Table 5.3 and Figure 5.13. The results show that the added floating car data help to further reduce the estimation error. When the penetration rate change from 0% to 5%, MARE is considerably improved. But higher rates do not bring much improvements.

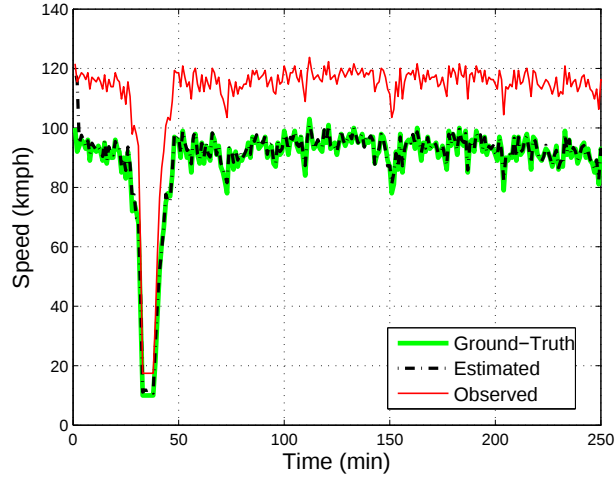


Figure 5.11: Comparison between ground-truth speeds, observed and estimated ones on road-segment 7 (around 3.8km milepost)

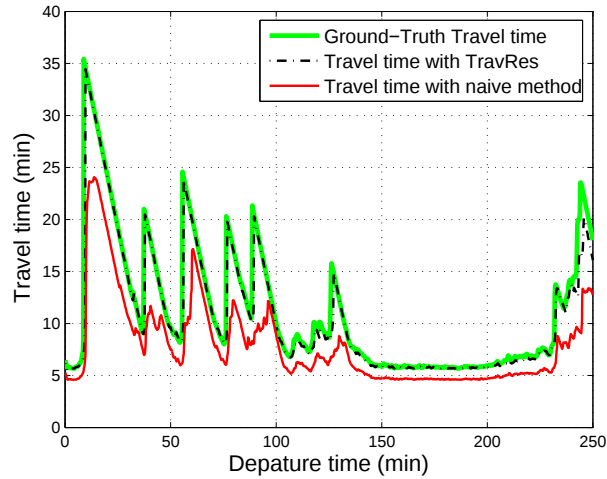


Figure 5.12: Compare travel times between before and after using this algorithm

5.4 Conclusion and recommendations

This chapter proposed a new algorithm (PISCIT) for fusing speeds from local detectors such as inductive loops with individual travel times measured by AVI systems. This algorithm is based on data-data consistency approach. In this algorithm, individual vehicle trajectories are reconstructed, consistent with the given travel times and proportionally

Table 5.3: The impacts of added floating car data. MARE without travel time information is 32.3% (global) and 64.6% (speeds lower 50kmph).

Penetration of FCD added	10%	5%	2.5%	1%	0.5%
MARE on speeds (global)	2.5%	2.65%	2.9%	3.5%	4.2%
MARE on low speeds (<50kmph)	5.6%	6.3%	7.7%	9.3%	10.2%

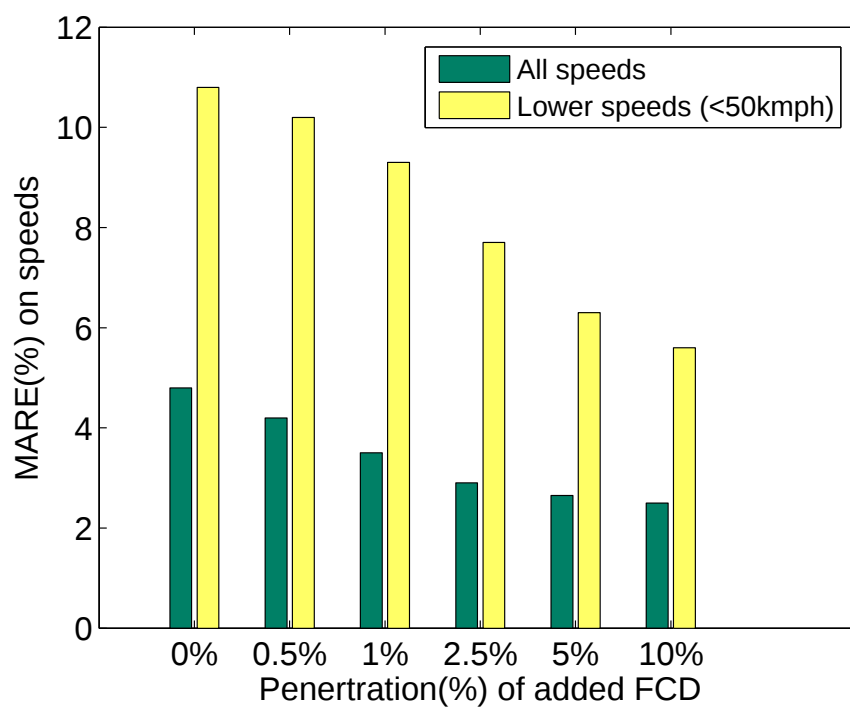


Figure 5.13: The impact of FCD added into camera data (camera data capture 10% vehicles).

consistent with the local speeds. Very few assumptions are needed: the traffic is assumed to be homogeneous in each time-space cell, and the travel times are assumed to be correct. We assume the proportionality of the errors in loop-estimated travel times. As matter of fact, when travel times are less reliable, we still can sample from distributions of given travel times and each sample has its own confidence. When the trajectories are combined, the confidence for each travel time sample can be taken into account.

The algorithm consists of two steps. In the first step vehicle trajectories are reconstructed by individual travel times in combination with an initial (prior) time-space speed plot (based on loop data). It is worth mention that the trajectory reconstruction technique in this part does not guarantee the convergence, though it seems that the reconstructed trajectory converges. In the second step a corrected (posterior) time-space speed plot is produced on the basis of these reconstructed vehicle trajectories. In this algorithm, the travel times are assumed to be reliable and correct, so they are used as constraints when a regression technique is employed to combine all reconstructed trajectories and output the time-space speeds.

On the basis of synthetic data driven by real-life loop data, we demonstrated PISCIT is able to successfully correct strongly biased prior speed measurements. The applications for PISCIT are manifold. First of all, in an offline context, PISCIT enables a simple but effective method to fuse data from local detectors and travel time data from AVI systems and hence improve the quality of archived datasets significantly. This is beneficial for a multitude of applications which depend on such historical data archives (e.g. simulation studies, performance analysis, policy evaluations, etc). In case the AVI data contains travel times over short distances, the algorithm might also have benefits for real-time ITS applications such as route guidance systems or ATIS.

However, this algorithm relies on prior speed information from loop detectors. When there is very little local speed information or no such information at all, PISCIT cannot work. To deal with this issue, the next chapter propose a more general algorithm which are able to fuse the travel times themselves or fuse them with little loop speed data. This approach is still based on data-data consistency.

Chapter 6

General approach for speed reconstruction by fusing travel times

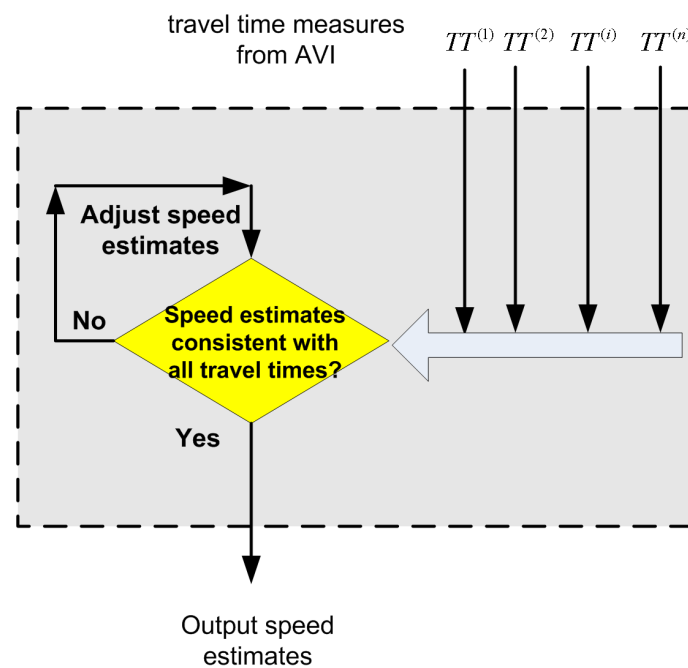


Figure 6.1: Illustration of internal consistency. The main function is to find speeds that are ‘best’ consistent with all travel time records from Floating Car Data

The previous chapter, Chapter 5 used a data-data consistency approach to remove much of the bias in local speed estimates by fusing travel times. In that method, the initial local speed measures from loop detectors are used as prior information, which are updated

Table 6.1: Symbol list

L_j	Length of road segment j
$t_j^{(k)}$	Time moment when probe vehicle k enters road segment j
$\hat{t}_j^{(k)}$	Estimated time moment when probe vehicle k enters road segment j
$tt_j^{(k)}$	Travel time of probe vehicle k on road segment j
$\hat{tt}_j^{(k)}$	Estimated travel time of probe vehicle k on road segment j
$tt_j^{(k)}(i, j)$	Estimated duration of vehicle k dwelling in cell (i, j)
$\hat{s}^{(k)}(i, j)$	Estimated traveled distance of vehicle k in cell (i, j)
$\hat{q}^-(i, j)$	Measured flow in cell (i, j) by loop detectors.
$\hat{v}^-(i, j)$	Measured (biased) speed in cell (i, j) by loop detectors.
$\hat{v}^{(k)}(i, j)$	‘Measured’ speed in cell (i, j) by trajectory k . Similarly for density $\hat{\rho}^{(k)}(i, j)$ and flow $\hat{q}^{(k)}(i, j)$
$\hat{v}(i, j)$	The final estimated speed in cell (i, j) by using travel times, speed and flow. Similarly for density $\hat{\rho}(i, j)$ and flow $\hat{q}(i, j)$

when the extra information of travel times are given. But in many cases, the loop detectors are sparsely installed or none of them is installed on a long road stretch. In order to estimate the traffic speeds in such a case, people may have to rely on floating car data (FCD). As shown in the following, a considerable bias in speed estimation is likely to be made out of these floating car data, especially when the FCD have low resolution in time horizon (lower polling rates).

To tackle this issue, this chapter proposes a new method called TravRes for accurate reconstruction of high-resolution (e.g. 500m*60sec) time-space speeds on the basis of sampled floating car data. This new method is superior to naive methods (classic methods), methods which reconstruct time-space speeds by simply dividing travel distance by travel time between consecutive FCD samples. Later on in the chapter, we will show the improvements by applying the method to a test case. It uses an iterative trajectory reconstruction technique as given in Chapter 5. Still based on data-data consistency approach, the ‘best’ estimate of speeds is found, such that the speed estimates are best consistent with all travel time measures, or in other words speed estimates and travel times meet the physics law $speed * time = distance$. This method is mainly checking the internal consistency, and only FCD are needed as input data. Figure 6.1 roughly shows the overall idea. Although only FCD is necessary for this method, this method can be extended such that other data source e.g. loop speeds can be fused. In methodology part, we will also show the extension of this method.

6.1 Introduction

6.1.1 Floating car data used for speed estimation

With the fast growth of in-vehicle ICT, the number of feasible applications based on in-car intelligent transportation system applications is rapidly growing as well. These include not only the (off- and online) collection of vehicle trajectories, speed, travel time (Laird (1996)) and even OD paths, but also incident detection and route guidance. Traffic monitoring on the basis of probe vehicle systems has many advantages over classic infrastructure based monitoring such as low cost per unit of data, continuous (over space) data collection, and the inherently non-intrusive nature of this type of monitoring (Turner et al. (1998)). The common use of global positioning systems (GPS) in mobile phones and/or in-vehicle navigation systems makes it easier and feasible to provide (more detailed) data of higher accuracy. In terms of traffic management, typical applications of floating car data include traffic state estimation (Herman (1984)), travel time estimation and prediction (Coifman (2002)). For the estimation and prediction of travel times, good traffic state estimation is the pre-condition, since traffic models for prediction are likely to take traffic states—density or speeds or both as input (Van Lint et al. (2002)). The accuracy of the source data for such applications is crucial. It is found, for instance, that errors in speed estimation of up to 20% may lead to errors in estimated density up to 100% (Stipdonk et al. (2008)). In this chapter, we focus on the accurate estimation of traffic speeds by using floating car data (also known as probe data or in brief FCD).

6.1.2 Challenge from floating car data

It is seemingly simple to derive traffic speeds from floating car data. Floating car data at least contain the probe vehicle's relative position on a road and timestamps. The commonly used method to derive the speed between consecutive probe vehicle reports is to divide traveled distance by travel time, resulting in time-space mean speed (Graph (a) in Figure 6.2). Such a method works well when floating car data have high polling rates (e.g. one report every one or a few seconds). But as shown in the following, this method brings about considerable errors when it comes to the estimation of high-resolution (e.g. 500m*60sec) time-space speeds from floating car data of low polling rate (30sec-90sec or more).

High resolution traffic speed plots may not be derived from FCD in a straightforward way. This is due to the fact that a probe vehicle reports its positions at a regular interval (e.g. 1 or 2 minutes), during which the vehicle may have already covered 1 or 2 kilometers

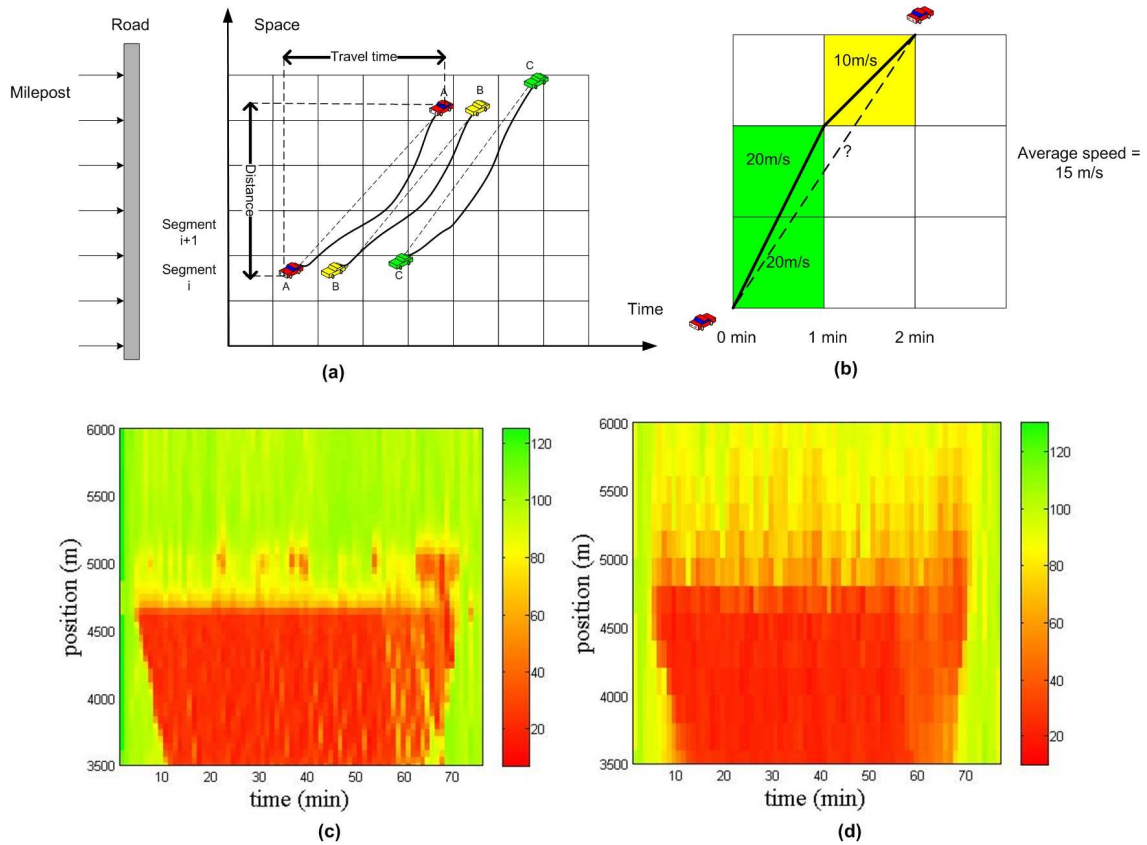


Figure 6.2: (a) Comparison of assumed vehicle trajectories with ground-truth ones.

(b) An example for estimated speed and ground-truth one.

(c) Ground-truth time space speed contour plots from simulated data in an example of Vissim simulation.

(d) Reconstructed time-space speed contour plots by naive method (classic method) on the basis simulated FCD in the Vissim simulation.

on a highway or passed several blocks in urban roads. Without any other assumptions or taking into other data sources, one can only derive the average journey speed a vehicle experiences during such polling intervals by dividing distance traveled with travel time. The underlying assumption is hence that the vehicle does not change its speed during the interval. Consequently, the trajectory of this vehicle is a straight line on a time-space map, obviously differing from its ground-truth trajectory as illustrated in Figure 6.2 (a). On the basis of such average journey speeds (straight vehicle trajectories) one can calculate an average speed in each time-space cell through which at least one probe vehicle ‘traversed’. We will refer to this method of calculating speeds from probe vehicle data as the naive method (classic method) from hereon in this thesis. This naive method causes considerably errors, particularly in low-speed regions. Figure 6.2 (b) gives a simple example, in which probe vehicle reports its location at an interval of 2 minutes and it travels at 20meters/sec in the first minute and 10 meters/sec in the second minute, resulting in 15 meters/sec on time-space average. This implies an 25% underestimation of speed in the first minute and a 50% overestimation in the second minute. What’s worse is that the estimation error “diffuses” into neighboring cells or regions. As a result, the reconstructed time-space speed plots from FCD are inaccurate as illustrated in the example in graph (c) and (d) in Figure 6.2. In this example, Vissim software was employed to give Ground-truth speeds as shown in Graph (c) and also generated virtual probe vehicle data with reporting interval of 2 minutes. When the FCD were processed with the naive method (classic method), the resulting speed plots Graph (d) displayed noticeable diffusion of speed estimation. Due to this diffusion, the speed estimation on the downstream road (from the mile point 4.8 km) has the average relative error of 37.8%.

In real-life and large scale application of FCD, it is likely that we may only get FCD of low polling rates rather than data of high polling rates due to restriction on communication cost. Therefore, it makes sense to find a way to use low polling rate data or low-resolution data. For this purpose, this chapter proposes a new method called TravRes to tackle the above-mentioned issue, borrowing the idea of trajectory reconstruction from PISCIT that is given in the previous chapter. In the next sections of this chapter, some theoretical analysis about speed reconstruction from FCD is given before the technique details on the method are fully presented. After presentation of the methodology, the method is validated on the basis of an experiment with simulated data.

6.2 Theoretical background

In this section, some quantitative analyses are given to reveal the relationship between floating car data (FCD) and cell speeds on time-space plots, finding that the ground-truth speeds can be exactly reconstructed by FCD under some assumptions. The first assump-

tion is that each cell of the time-space speed plot has homogeneous traffic conditions. This implies that the vehicle trajectories in a cell are parallel straight lines. The second assumption is that the given floating car data are consistent with the ground-truth time-space speed plots, which means the floating car data can either be exactly reconstructed through the time-space speed plots.

6.2.1 Analysis of simple cases

On the basis of simple example cases we will first demonstrate that it is possible to use low-resolution FCD to exactly reconstruct ground-truth speed in each time-space cell, if the above assumptions are made. Figure 6.3 (a) presents such a simple two-cell case. This graph shows two neighboring time-space cells referred to as x and y , in which two probe vehicles only pass the space-boundary of cells once without passing the time-boundary. Both the probe vehicles report their locations after a time interval, which are marked by black dots in the graph. Since the locations with timestamps are provided, the distance from their reporting locations to space-boundary of the cells is known, represented by $S1$, $S2$, $S3$ and $S4$ shown in Figure 6.3 (a). Assuming homogeneous speed on each cell, the relationship between time, distance and speeds can be established through the below equations:

$$S1/vx + S2/vy = \Delta t \quad (6.1)$$

$$S3/vx + S4/vy = \Delta t \quad (6.2)$$

It can be expressed with matrix as:

$$\begin{bmatrix} S1 & S2 \\ S3 & S4 \end{bmatrix} \begin{bmatrix} 1/vx \\ 1/vy \end{bmatrix} = \begin{bmatrix} \Delta t \\ \Delta t \end{bmatrix} \quad (6.3)$$

Looking into the equations, it is found that the unique solution can be achieved if the matrix

$$\begin{bmatrix} S1 & S2 \\ S3 & S4 \end{bmatrix} \quad (6.4)$$

has full rank. For FCD with the same polling rate, the unique and reasonable solution (positive speeds) can be achieved only if the two probe vehicles do not report the same locations. In addition, the unique solution has to equal to the ground-truth speeds. Since the ground-truth speeds and the above solution both satisfy this equation, if they were not

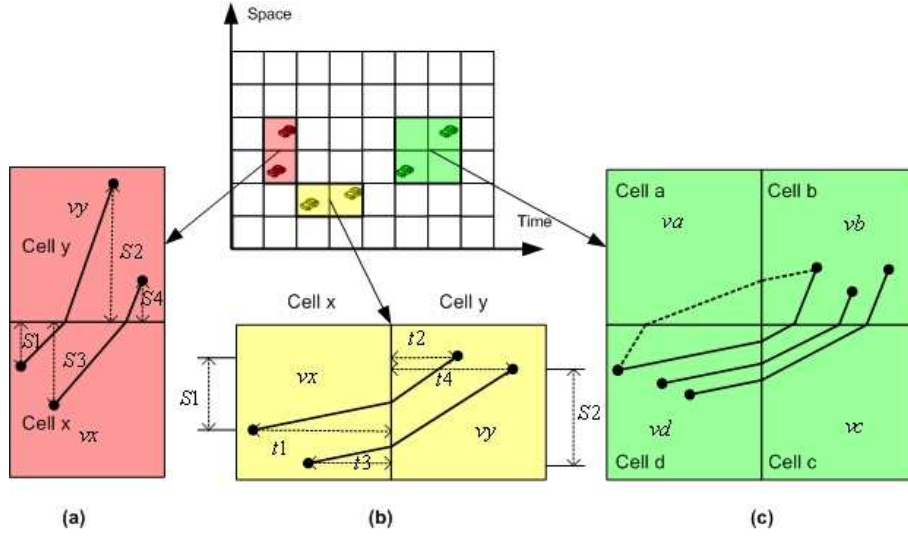


Figure 6.3: Illustrations of a few cases when probe-vehicle trajectories run through time-space cells.

equal, then solutions to this equation will not be unique.

Figure 6.3 (b) shows the case when the probe vehicles only pass the time-boundary once without pass the space-boundary. Similarly, the below equations can be established as

$$t1 \cdot vx + t2 \cdot vy = S1 \quad (6.5)$$

$$t3 \cdot vx + t4 \cdot vy = S2 \quad (6.6)$$

or

$$\begin{bmatrix} t1 & t2 \\ t3 & t4 \end{bmatrix} \begin{bmatrix} vx \\ vy \end{bmatrix} = \begin{bmatrix} S1 \\ S2 \end{bmatrix} \quad (6.7)$$

has full rank. The unique and reasonable solution can be achieved only if the two probe vehicles do not report their locations at the same time. Same with the first case, the unique solution give exactly the ground-truth speeds.

6.2.2 Analysis of more complicated cases

Figure 6.3 (c) illustrates a more complicated case in which probe vehicles run through three time-space cells. Assuming that the trajectories are known to run through cell d, c and b, it is not easy but still feasible to establish a similar linear equation in terms of $\mathbf{Ax} = \mathbf{b}$ as already shown in the first two cases, where vector $\mathbf{x}$ represents cell speeds or inverse speeds (1/speed). It can also be found that the ground-truth speeds on cell d, c and b can be exactly deduced, given enough FCD such that $\mathbf{A}$ has full rank. Further more, if the trajectory is incorrectly assumed to run through cell d, c and b instead, the conflict equation (no solution) will be established when enough FCD are given. In that case, the assumption has to be changed till no conflict equation occurs. In the light of the above, such a linear equation $\mathbf{Ax} = \mathbf{b}$ always exist for whatever complex cases. Furthermore, if enough FCD are given so that $\mathbf{A}$ has full rank, the unique and exact solution to cell speeds can be achieved. Even when the FCD are not fully consistent with ground-truth speeds (e.g. in real-life, some vehicles run faster while some other slower than ground-truth ones), one optimal solution to the possible conflict equation $\mathbf{Ax} = \mathbf{b}$ can be solved by a transformed equation such as $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$.

We therefore conclude that if enough FCD are available and if these are fully consistent with ground-truth speeds, these cell speeds can be estimated exactly under the assumption of homogeneous speed on each cell, no matter how small the cell size is chosen or how large the polling interval is. Furthermore, this estimation is unique and the only one which can be fully consistent with FCD.

6.2.3 Implied difficulties in complicated cases

Practically, however, when more cells and more floating car data (FCD) are considered, the establishment of such a linear equation becomes impossible. More generally put, the key difficulty lies in the fact that whereas sampled vehicle trajectories (i.e. floating car data) can be easily reconstructed from the time-space speeds, the inverse, deriving time space speeds from sampled vehicle trajectories becomes rapidly infeasible, even in case the assumptions introduced above are met. Particularly, it is hard to make correct assumptions about which cells are traversed through by the probe vehicles. In addition, the time cost for solving the linear equations outlined above exponentially rises with the dimensions, which may even make the real-life implementation impossible.

The new method proposed below does not establish such linear equations for solution. Instead, it uses a heuristic and iterative way to approximate the solution for cell-speed estimation. We call the new method TravRes since it enables more detailed *Trave* experience

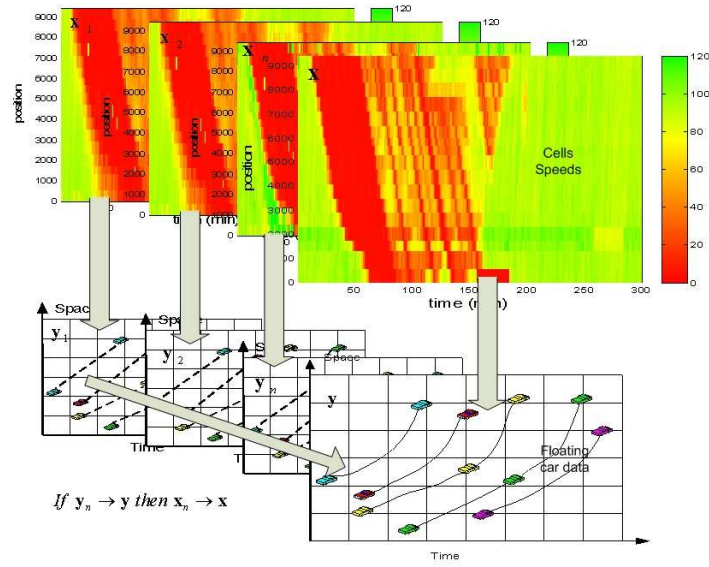


Figure 6.4: Evolution and convergence of time-space speeds plots and corresponding sets of FCD. This graph shows that when the sampled travel times approach the given travel time records, the estimated speed plots approach the ground-truth speed plots (under certain conditions).

Reverse from FCD, leading to high-resolution time-space speeds.

6.3 Methodology: TravRes

This section elaborates the method in detail. This method is based on some simple assumptions and physical laws. Homogeneous and stationary traffic conditions are assumed in each time-space cell. The travel times are assumed to be correct and consistent with ground-truth traffic speeds. The quantities of travel times are large enough so that no more than one speed result can satisfy all the travel times. The physical law is $distance = speed * time$.

Before we give the method detail, we first present the basic ideas in this method.

6.3.1 Basic ideas

The fundamental idea is shown in Figure 6.4. In this figure, there are a sequence of time-space speed plots. For each time-space speed graph, FCD can be reconstructed by sampling speed plots. However, under certain assumptions, there is only one time-space speeds which can exactly lead to the given FCD. Our goal is to find such time-space graph which can exactly reconstruct the given FCD.

For example, there is one piece of FCD, which shows that a vehicle is at location 0 at time 0 minute, and at location 1000 meters at time 1 minute. When we get a graph of estimated time-space plots, we can reconstruct its trajectory by sampling speed plots. If the reconstructed trajectory shows that the vehicle is indeed location 0 at time 0 minute and at location 1000 meters at time 1 minute, we call that the estimated speed plots are *consistent* with this piece of FCD. If the reconstructed trajectory shows that the vehicle is at location 900 meters at time 1 minute, we can update the estimated time-space speeds so that the resulting speeds are consistent with this piece of FCD.

It is very likely that many estimation results are consistent with this piece of FCD. But for a certain set of FCD, there is one and only one speed estimation which is consistent with this set of FCD. Such a set of FCD does exist if we assume that traffic is homogeneous and stationary in each time-space cell so that vehicle trajectories in each cell are parallel with one another. As shown in the cases in the previous section, one piece of FCD implicates one equation where time-space speeds are taken as variables. When more and more equations are established with more FCD, there is one solution to all the equation and this solution is ground-truth time-space speed. In other words, such a set of FCD can be sampled from the ground-truth time-space speeds, and there are enough of them so that there exists one and only one graph of time-space speeds which can exactly reconstruct all the FCD.

As a matter of fact, we are not going to establish a set of equations and solve the equations for traffic speed estimation. The previous section has already shown that it is very inefficient and difficult to establish equations and solve them when many FCD are taken. For this reason, we are taking a ‘feed-back’ strategy. Let us assume that there is only one estimation that is exactly consistent with all the given FCD. Firstly, we find arbitrary speed estimation. If this estimation is not consistent with the given FCD, we update the estimation and get new estimation which tends to have less inconsistency with the FCD. Iteratively update the estimation until the estimation is enough or exactly consistent with the FCD. The next problem is how to design such an algorithm to update the estimation so that less consistency will be achieved in each time. The answer will be given below.

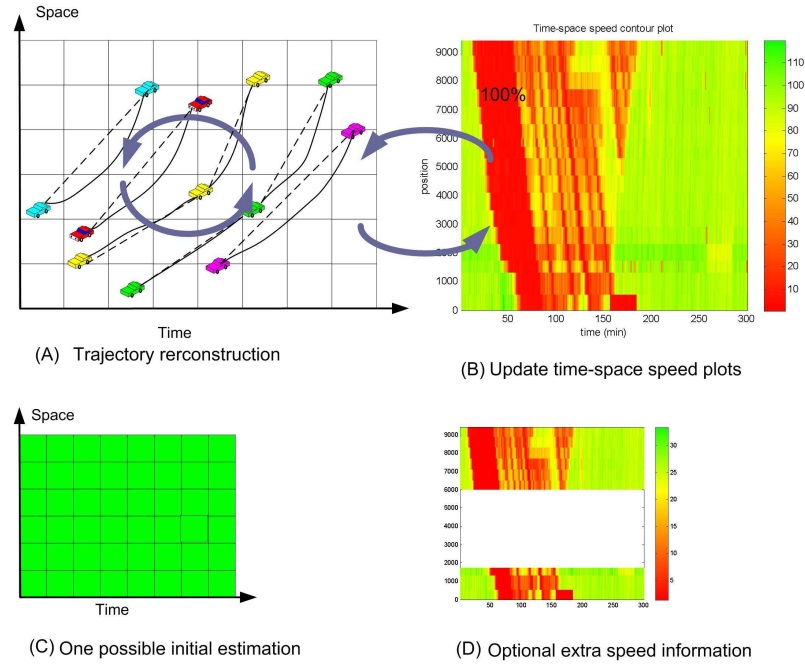


Figure 6.5: The whole framework of TravelRes

6.3.2 Methodology framework

This method consists of two iteration loops—inner loop and outer loop as shown in Figure 4. The inner loop accomplishes the reconstructions of trajectories and outer loop further reconstruct time-space speeds. With enough cycles (iterations), estimated speeds are supposed to be consistent with the given FCD to some extent.

- Figure 6.5 (A) represents the inner loop which accomplishes trajectory reconstruction, an algorithm adapted from PISCIT. The input is the given FCD and the previously estimated time-space speeds. For the initial estimation, we simply assume that the speeds over the whole time-space region are equal as shown in (C). As matter of fact, the initial estimation is not supposed to change the final result.
- Figure 6.5 (A) and (B) constitute the outer loop, in which time-space speed estimation is reconstructed by the reconstructed trajectories and the resulting speed estimation is to be taken as input for inner loop in the next cycle.
- Figure 6.5 (D) represents an optional input for the inner loop when there is additional traffic speed information available from other data sources or modeling. Different confidences are put on different time-space regions for different data sources.

For example, if correct speed information on some road segments is available, it can be used to replace the corresponding time-space region in graph (B) in each cycle.

The above iterative steps can be executed until the sampled FCD from the resulting speed estimation is close enough to the given FCD.

6.3.3 Inner loop: trajectory reconstruction

This part of the algorithm is able to reconstruct individual trajectories by combining the given travel times and previously observed (estimated) cell-speeds from loop detectors. Cameras or in-car GPS can provide the entry point for a vehicle, that is where and when the vehicle enters a road stretch. Also the exit point is given about where and when this vehicle leaves the road stretch. Any line which links the two points could be a trajectory for this vehicle. The algorithm presented below is able to find the most ‘likely’ trajectory with the help of the time-space speed information from loop detectors, even though there is considerable bias in these speed measures. The mechanism behind is quite simple. For a fixed road segment in a road stretch, longer travelled distance, more travel time; higher speed, less travel time;

In illustration, it is assumed a probe vehicle k entered road segment 1 at reporting time $\hat{t}_1^{(k)}$ and exited segment 6 at the next reporting time $\hat{t}_7^{(k)}$ (Refer to Figure 6.6). This trajectory reconstruction algorithm is made up of the steps below, the first four of which accomplish reconstruction on segment level while the last two on cell level. Table 8.1 lists the important symbols used below.

STEP 1: Get $\hat{t}_j^{(k)}$ and $t\hat{t}_j^{(k)}$ from this previously-estimated trajectory as shown in Graph (a) in Figure 6.6 (The initial trajectory can be assumed to be a straight line)

STEP 2: Based on the given time-space speeds (biased), (re)-calculate the average speed $\bar{v}([\hat{t}_j^{(k)}, \hat{t}_{j+1}^{(k)}], j)$ over segment j during the time between $\hat{t}_j^{(k)}$ and $\hat{t}_{j+1}^{(k)}$.

STEP 3: Update $\hat{t}_j^{(k)}$ and $t\hat{t}_j^{(k)}$ based on the average speed $\bar{v}([\hat{t}_j^{(k)}, \hat{t}_{j+1}^{(k)}], j)$. The updated $t\hat{t}_j^{(k)}$ can be obtained from the equations displayed below

$$t\hat{t}_j^{(k)} \propto \frac{L_j}{\bar{v}([\hat{t}_j^{(k)}, \hat{t}_{j+1}^{(k)}], j)} \quad (6.8)$$

$$\sum_j t\hat{t}_j^{(k)} = tt \quad (6.9)$$

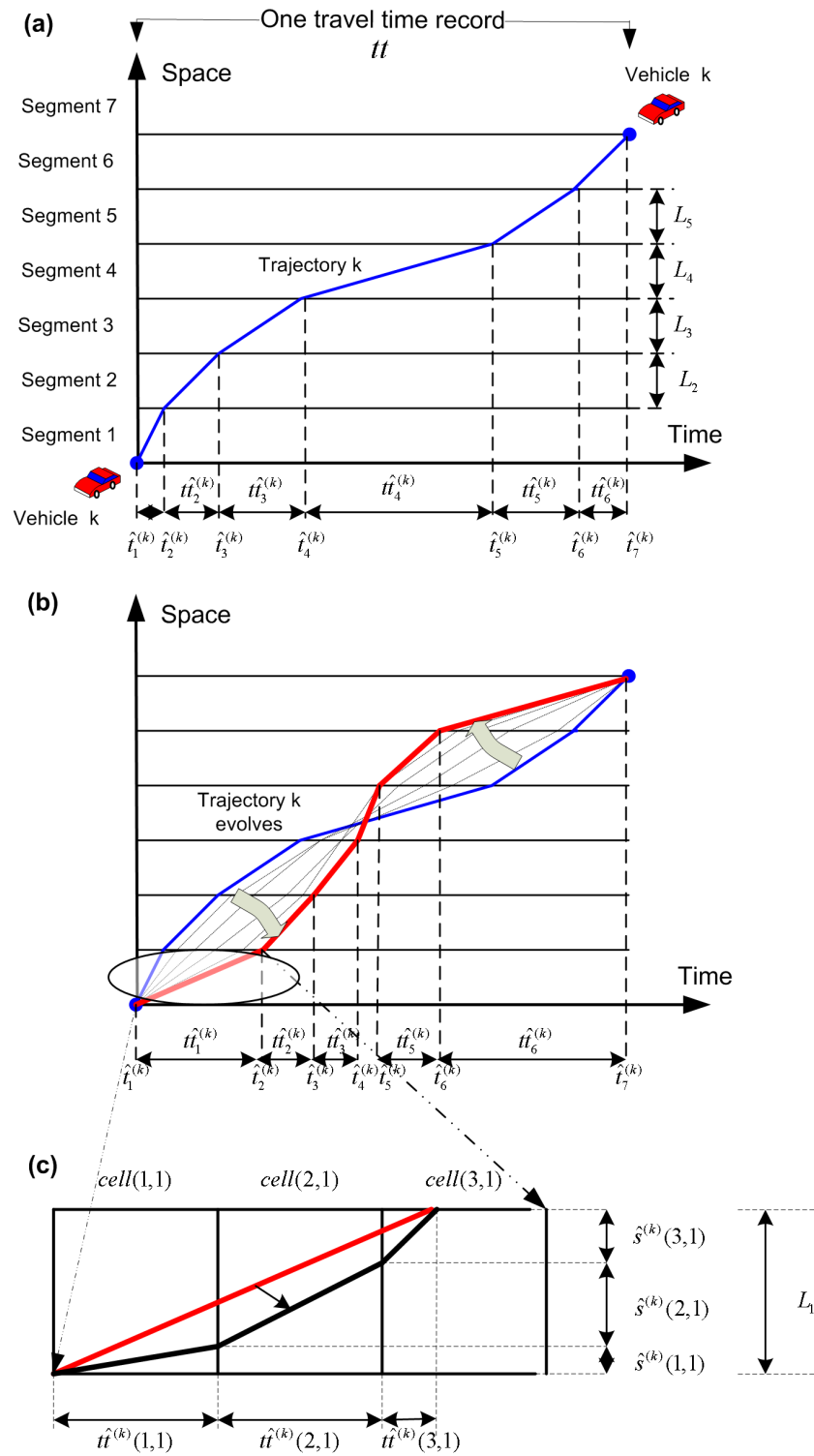


Figure 6.6: Illustration of trajectory reconstruction algorithm

,where tt is the given travel time for a vehicle over the whole stretch. After that, update $\hat{t}_j^{(k)}$ based on $tt_j^{(k)}$.

STEP 4: Repeat STEP 2 and STEP 3 until $\hat{t}_i^{(k)}$ and $tt_i^{(k)}$ converge to a specific extent. (Refer to Graph (b) in Figure 6.6)

STEP 5: Deduce $tt^{(k)}(i, j)$ from $tt_j^{(k)}$, $\hat{t}_j^{(k)}$ and cell size. (Refer to Graph (c) in Figure 6.6)

STEP 6: Deduce $\hat{s}^{(k)}(i, j)$ under the below equations. (Refer to Graph (c) in Figure 6.6)

$$\hat{s}^{(k)}(i, j) \propto tt^{(k)}(i, j) * \hat{v}^-(i, j) / \bar{v}([\hat{t}_j^{(k)}, \hat{t}_{j+1}^{(k)}], j) \quad (6.10)$$

$$\sum_i \hat{s}^{(k)}(i, j) = L_j \quad (6.11)$$

L_j is the length of segment j . The division of a whole stretch depends on the requirement for estimation resolution as well as the input FCD. Normally, we make the length of each segment about 500 meters. If we have a large number of the input FCD with high polling rate e.g. 30seconds, we can divide a road stretch into more smaller road-segments.

6.3.4 Outer loop: time-space speed reconstruction

In this section, cell speeds are estimated via the above reconstructed trajectories. With the reconstructed trajectories, $tt_i^{(k)}(i, j)$ and $\hat{s}_i^{(k)}(i, j)$ can be known, which are the travel time and travel distance for probe vehicle k on time-space cell (i, j) respectively. In the case that a cell has two or more trajectories pass through, the speed on the time-space cell (i, j) can be estimated as

$$\begin{aligned} \hat{v}(i, j) &= \frac{\sum_k w^{(k)}(i, j) \hat{v}^{(k)}(i, j)}{\sum_k w^{(k)}(i, j)} \\ &= \frac{\sum_k w^{(k)}(i, j) \hat{s}^{(k)}(i, j) / tt^{(k)}(i, j)}{\sum_k w^{(k)}(i, j)} \end{aligned} \quad (6.12)$$

where $w^{(k)}(i, j)$ is the weight put on the speed estimation $\hat{v}^{(k)}(i, j)$ from probe vehicle k in time-space cell (i, j) . One possible choice for weight is:

$$w^{(k)}(i, j) = \sqrt{\hat{s}^{(k)}(i, j) \hat{t}^{(k)}(i, j)} / \int_0^{tt/2} |\hat{v}^{(k)}(t)| dt \quad (6.13)$$

The reason for this choice is given below.

If a probe vehicle k leave a longer trajectory in a cell (i, j) , it is supposed to put more weight on $\hat{v}^{(k)}(i, j)$ when it is used for the speed estimation $\hat{v}(i, j)$. Considering both space length and time span, we assume

$$w^{(k)}(i, j) \propto \sqrt{\hat{s}^{(k)}(i, j) \hat{t}^{(k)}(i, j)} \quad (6.14)$$

The main reason to choose such an assumption is that the actual ‘length’ may lead to a dimensionality issue. The actual ‘length’ this trajectory left in the cell is $\sqrt{\hat{s}^{(k)}(i, j)^2 + \hat{t}^{(k)}(i, j)^2}$, however $\hat{s}^{(k)}(i, j)$ and $\hat{t}^{(k)}(i, j)$ have different units. Considering that inequality $ab \leq (a^2 + b^2)/2$, $\sqrt{\hat{s}^{(k)}(i, j) \hat{t}^{(k)}(i, j)}$ is simply used.

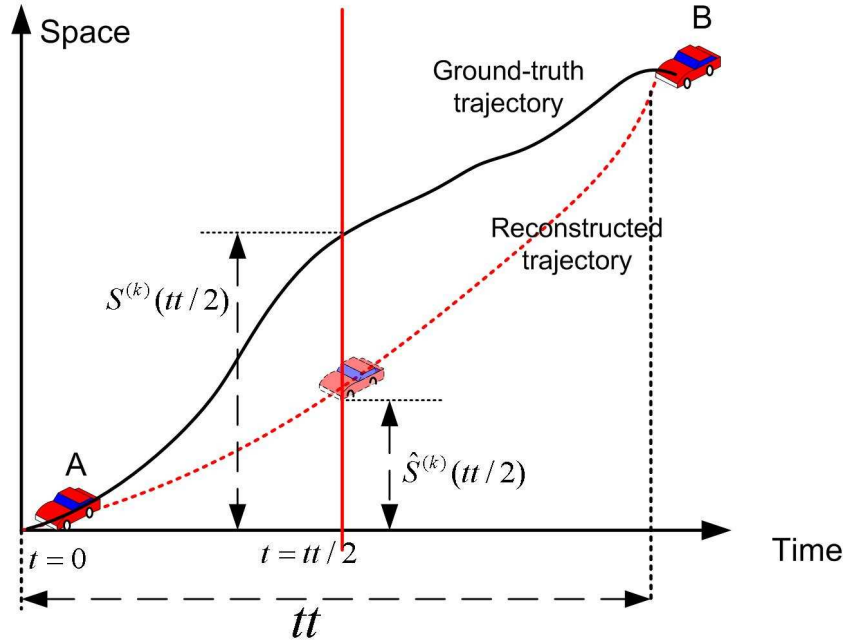


Figure 6.7: Comparison of the ground-truth trajectory and the reconstructed trajectory

When polling rates of probe vehicles are different or camera data (also providing travel times) are considered, different confidence is supposed to be placed on trajectories of

different polling rate and traffic experience. Compared with one minute polling rate, two minutes rate can lead to larger deviation of reconstructed trajectories from the ground-truth ones. More deviation, less confidence is put this trajectory. If we consider polling rate tt , we will also find

$$w^{(k)}(i, j) \propto 1 / \int_0^{tt/2} |\hat{v}^{(k)}(t)| dt \quad (6.15)$$

Now let us explain why this equation makes sense. Figure 6.7 shows that a probe vehicle reports its location at time 0 and report its next location at time $t = tt$. This piece of FCD can precisely tell its locations at time 0 and at time $t = tt$, so the reconstructed trajectory (dashed red) and the ground-truth one (black) share the same locations at the both ends. But during the polling interval, their trajectories are most likely to deviate from each other.

Since there is no deviation at the two ends, the biggest deviation is supposed to be between. For simplicity, we assume that the biggest deviation occurs at time $tt/2$. Let $\hat{S}^{(k)}(t)$ and $S^{(k)}(t)$ represent the estimated location and ground-truth one of probe vehicle k at time t respectively. An attempt is made to approximately quantify the difference between $\hat{S}^{(k)}(tt/2)$ and the exact value $S^{(k)}(tt/2)$. Given the ground-truth speed for this vehicle, it can easily found that

$$dS^{(k)}(t) = v^{(k)}(t)dt \quad (6.16)$$

or

$$S^{(k)}(t + \Delta t) = S^{(k)}(t) + v^{(k)}(t)\Delta t \quad (6.17)$$

for very small Δt . But for the estimated reconstructed trajectory, only $d\hat{S}^{(k)}(t) = v^{(k)}(t)dt + \eta$ can be established, in which possible errors on speed estimation and trajectory reconstruction are considered, and η represents the random displacement due to these errors. To better describe this random displacement, Brownian motion (Wiener Process) is introduced. In the standard Brownian motion, the random displacement during the period from t till $t + \Delta t$ is $W(t + \Delta t) - W(t)$, where $W(t)$ is the position of particle at time t and the displacement follows the normal distribution:

$$W(t + \Delta t) - W(t) \sim N(0, \Delta t) \quad (6.18)$$

Considering that the absolute error on the estimate of higher speed is larger while the relative error on the estimation of lower speed is larger, it makes sense to establish

$$\eta \approx \alpha \sqrt{v^{(k)}(t)}(W(t + \Delta t) - W(t)) \quad (6.19)$$

For very small Δt , it can be assumed that $\eta = \alpha \sqrt{v^{(k)}(t)} dW(t)$, leading to a stochastic differential equation

$$d\hat{S}^{(k)}(t) = v^{(k)}(t)dt + \alpha \sqrt{v^{(k)}(t)} dW(t) \quad (6.20)$$

where $W(t)$ is Brownian motion process with covariance $\sigma^2 = 1$ and α is a tunable factor with the dimension $sec * \sqrt{meter/sec}$. Actually, α would have been displayed in Equation 6.23 and Equation 6.24, but when α is assumed to be constant, it can be cancelled in Equation 6.12. Therefore we simply make $\alpha = 1$. Integrating Equation 6.20 from time 0 till time $tt/2$ when the biggest deviation may occur as shown in Figure 6.7, the following equation can be deduced.

$$\hat{S}^{(k)}(tt/2) = \int_0^{tt/2} v^{(k)}(t)dt + \int_0^{tt/2} \sqrt{v^{(k)}(t)} dW(t) \quad (6.21)$$

Further, the covariance of $\hat{S}^{(k)}(tt/2)$ can be deduced as below:

$$\begin{aligned} \text{cov}(\hat{S}^{(k)}(tt/2)) &= \text{cov}\left(\int_0^{tt/2} v^{(k)}(t)dt + \int_0^{tt/2} \sqrt{v^{(k)}(t)} dW(t)\right) \\ &= \text{cov}\left(\int_0^{tt/2} \sqrt{v^{(k)}(t)} dW(t)\right) \\ &= \int_0^{tt/2} |v^{(k)}(t)| dt \end{aligned} \quad (6.22)$$

Since $v(t)$ is unknown, we replace it with the estimation $\hat{v}^{(k)}(t)$ in Equation 6.22, leading to the below result:

$$\text{cov}(\hat{S}^{(k)}(tt/2)) \propto \int_0^{tt/2} |\hat{v}^{(k)}(t)| dt \quad (6.23)$$

Obviously, the bigger the deviation is, the less confidence is put on the trajectory. So we can also assume

$$w^{(k)}(i, j) \propto 1 / \int_0^{tt/2} |\hat{v}^{(k)}(t)| dt \quad (6.24)$$

Combined with Equation 6.14, the weight $w^{(k)}(i, j)$ now equals

$$w^{(k)}(i, j) = \sqrt{\hat{s}^{(k)}(i, j) \hat{tt}^{(k)}(i, j)} / \int_0^{tt/2} |\hat{v}^{(k)}(t)| dt \quad (6.25)$$

Now $\hat{s}^{(k)}(i, j)$, $\hat{tt}^{(k)}(i, j)$ and $w^{(k)}(i, j)$ become available, so the speed estimation in cell (i, j) can be achieved via Equation 6.12. Considering all the FCD, then time-space speeds

on the specific region can be reconstructed.

6.4 Method validation

In this section, three scenarios are set up to validate the method proposed in the previous section. They are aimed to show how TravRes outperforms the naive method (classic method), how probe vehicle polling rates and penetration rates impact the results, and how camera data and partial loop data can be added to further increases the accuracy of estimation. In scenario A, only FCD are used for speed estimation under different polling rates and penetrations. In scenario B, travel times from virtual cameras spaced 4,5 kilometers apart are used as additional data source. Such camera data can be taken as a special kind of FCD in processing since they share the same data format, they are processed the same way that FCD are done. In scenario C, loop data are added and taken as optional input as shown in Figure 6.5 (c). It is assumed that the loop detectors only covered part of road segments. The motivation of Scenarios C is to see if the external data can help FCD to give better speed estimation on the road where the loop detectors Do Not cover.

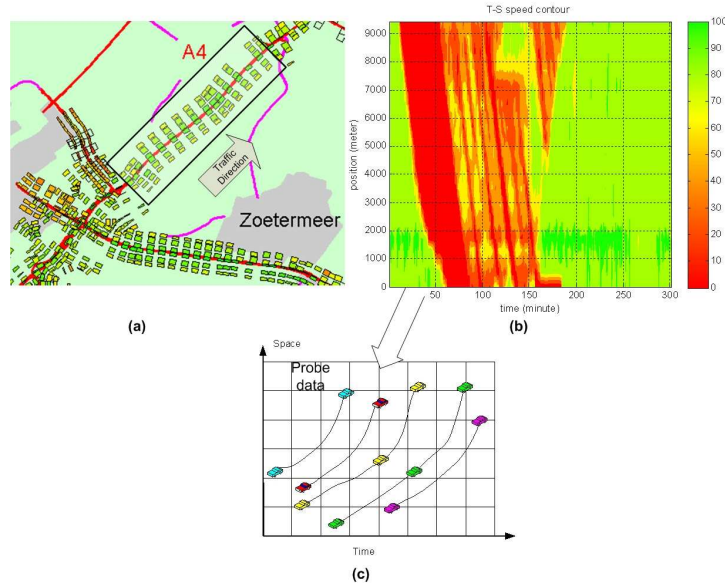


Figure 6.8: Assumed ground-truth time-space contour plots, generated through real-life data from 18 loop detectors spacing around 500 metres on a 9,5 km stretch of Highway A4 Eastbound in Netherlands.

In the three scenarios, the ultimate outputs are time-space speed plots with resolution

500m*60sec or so. So when FCD have high polling rate (e.g 30seconds), there is lower chance that the probe vehicles traverse multi time-space cells during one polling interval. When FCD have low polling rate (e.g. 120seconds), the probe vehicles are likely to traverse quite a few time-space cells during one polling interval.

6.4.1 Experiment setup and generation of virtual FCD

First of all, a 9.5 kilometer stretch of 3-lane Highway A4 eastbound in Netherlands is considered (Graph (a) in Figure 6.8), where 18 loop detectors are placed spacing around 500 meters and aggregated traffic speed measures and counts every one minute. The resulting time-space speed contour plots (Graph (b) in Fig. 6.8) from loop detector shows 5 hour traffic condition on this stretch from 6:00 A.M. till 11: A.M. on July 8th, 2008, during which congestions onset on the morning and dispersed on the afternoon. Next, the above real-life speeds are taken as ground-truth ones to generate virtual probe vehicles and subsequent virtual FCD (Figure 6.8 (c)). The probe vehicles were generated at the location 0 meter at random time during the 5 hours. With the help of time-space speeds, the ‘ground-truth’ trajectories were reconstructed. As a result, FCD were generated by sampling the time-space locations of the ‘true’ trajectories at a certain rate and meanwhile the resulting FCD were fully consistent with ‘ground-truth’ speeds. Penetrations of probe vehicles were given based on the fact that there were totally about 28500 vehicles passing through the first loop detectors during the 5 hours. Penetrations varies from 2% to 10% and polling rates varied from 30sec to 120sec.

6.4.2 Speed reconstruction by FCD only

The initial speed plots on the whole time-space region (refer to Figure 6.5 (b)) are assumed to have uniform speeds 20meter/sec and the generated FCD are put into the method (refer to Fig. 6.5 (a)). In this scenario, there are no other data source as input. With the above data as input, the inner cycle is executed iteratively 15 times and outer cycle only 9 times. As the later findings show that the error inconsistency decreased very rapidly, so only about 10 iterations are needed. In the speed reconstruction part, the Equation 6.12 and Equation 6.13 are employed. *MARE* (Mean Average Relative Error) is used to evaluate the performance of the naive method (classic method) and TravRes as shown below:

$$\text{MARE} = \frac{1}{M * N} \sum_i^M \sum_j^N \frac{|\hat{v}(i, j) - v(i, j)|}{v(i, j)} \quad (6.26)$$

where $\hat{v}(i, j)$ is the estimated speed in cell (i, j) and $v(i, j)$ is the ground-truth speed. Figure 6.9 and Fig. 6.10 compare the performances of the naive method (classic method) and TravRes with different penetrations and different polling rates. In calculating MARE, the speeds on some of cells failed to be estimated due to the limit of penetration, so only the cells where there are reconstructed trajectories passing through were taken into account. The penetration is based on the number of probe vehicles passing by the most upstream detector, that is about 28500 in number. This coverage approximately indicates the percentage of reconstructed trajectories covering the whole time-space regions based on the case of 5% penetration and 60 seconds polling rate (The coverage is similar with the case of 120 seconds polling rate). Both Fig. 6.9 and Fig. 6.10 show that the proposed method TravRes outperforms the naive method (classic method) that takes weighted average speed for estimation if multi trajectories are found in one cell. In addition, MARE significantly decreases with the polling rate rise from 2 min to 30 seconds.

Table 6.2: MARE of speed estimation over the whole time-space region by using Naive and TravRes methods respectively.

Penetration	Polling rate	MARE with naive	MARE with TravRes
10%	120sec	15.4%	7.4%
10%	60sec	9.1%	3.7%
10%	30sec	5.4%	2.7%
5%	120sec	16.8%	9.5%
5%	60sec	9.8%	3.9%
5%	30sec	5.7%	2.6%
2%	120sec	17.7%	12.3%
2%	60sec	11.4%	6.3%
2%	30sec	7.2%	3.1%

Table 6.3: MARE of speed estimation (speeds are lower than 50kmph) by using Naive and TravRes methods respectively.

Penetration	Polling rate	MARE with naive	MARE with TravRes
10%	120sec	38.9%	11.8%
10%	60sec	23.4%	7.0%
10%	30sec	13.8%	6.5%
5%	120sec	44%	17.3%
5%	60sec	24.9%	7.5%
5%	30sec	15%	6.6%
2%	120sec	50%	29%
2%	60sec	33.5%	14.2%
2%	30sec	20%	7.9%

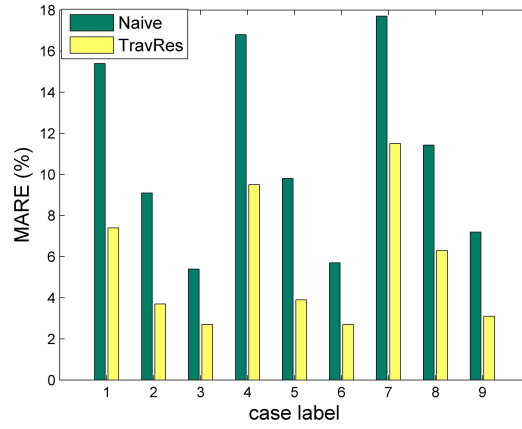


Figure 6.9: Global MAREs (%) in speed estimation by using the naive method and TravRes under different polling rates and penetration rates
 case 1,2,3: penetration rate 10% with polling rate 120sec, 60sec and 30sec respectively;
 case 4,5,6: penetration rate 5% with polling rate 120sec, 60sec and 30sec respectively;
 case 7,8,9: penetration rate 2% with polling rate 120sec, 60sec and 30sec respectively.

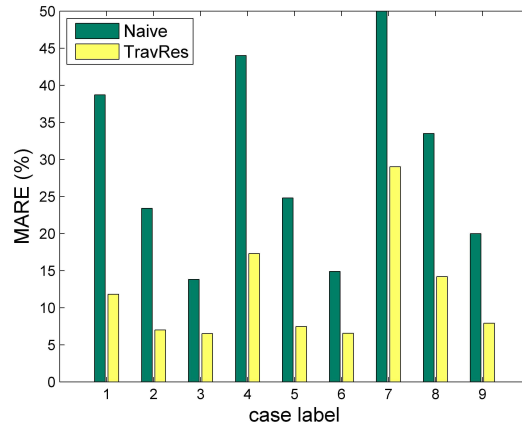


Figure 6.10: For time-space region with speeds smaller than 50km/h, this figure shows the MAREs (%) in speed estimation by using the naive method and TravRes under different polling rates and penetration rates
 case 1,2,3: penetration rate 10% with polling rate 120sec, 60sec and 30sec respectively;
 case 4,5,6: penetration rate 5% with polling rate 120sec, 60sec and 30sec respectively;
 case 7,8,9: penetration rate 2% with polling rate 120sec, 60sec and 30sec respectively.

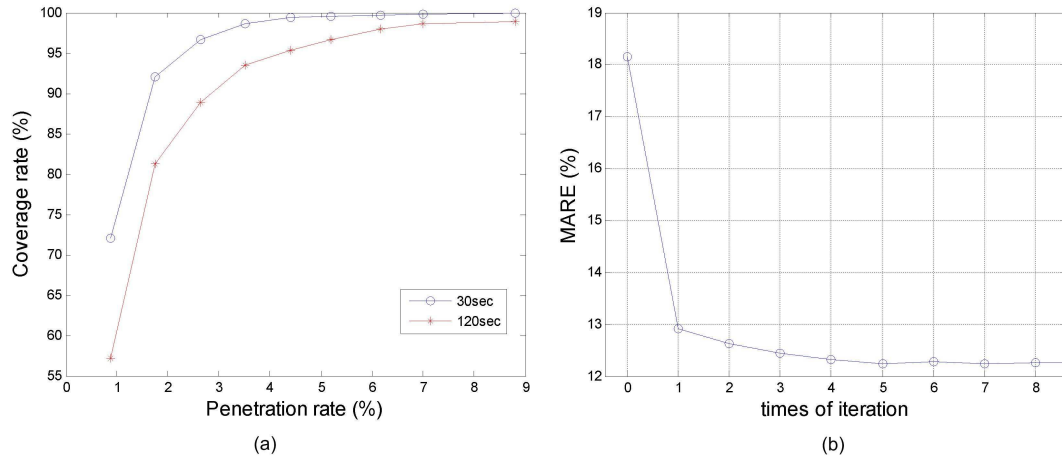


Figure 6.11: (a) Coverage rate: the percentage of time-space cells whose speeds can be estimated among all time-space cells;
(b) Decrease and Convergence of MARE with the iterations in the outer cycles.

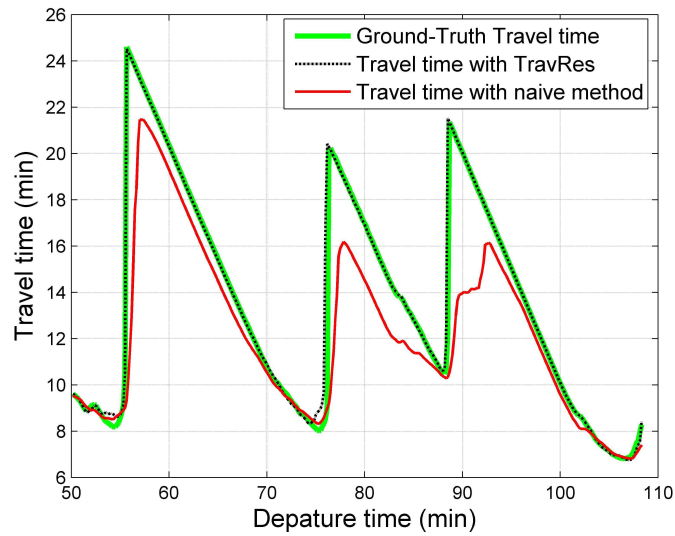


Figure 6.12: Comparison of estimated travel times with different methods

Graph (a) in Figure 6.11 shows the coverage rate of travel time records with different polling rates and penetration rates. It can be seen that when the penetration rate is more than 3%, we can estimate the majority (90%) time-space cells. Graph (b) in Fig. 6.11 shows an example of how fast the result converge with TravRes method. It can be seen MARE of estimated speeds has already leveled after the 5th estimation.

Also we can estimate travel times by sampling the reconstructed time-space speed plots. As shown in Figure 6.12, the red curve represents the estimated travel times by using naive-method-based speed plots, and the dashed curve represents the estimated travel times by using TravRes-based speed plots, and the bold green curve represents the ground-truth travel times. They refers to travel times over this 9 kilometer road segment during 50th minute (6:50 a.m.) till 110th minute (7:50 a.m.). It is found that the travel times from naive method (classic method) have a relative error of 8.6%, while the travel times from TravRes have a relative error of only 1.2%. This shows that TravRes is able to give much better travel time estimation.

In this validation, it is found that TravRes can significantly improve the accuracy of speed estimation compared with the naive method (classic method). Combination with camera data or loop data leads to better estimations. In experiments, it is also found that this method is time-cost efficient. For processing data with 10% penetration and 60 seconds polling rate for 5 hour durations, 10 times of outer cycles can be accomplished within 8.5 seconds. In other words, 21600 travel time records are processed within 8.5 seconds.

6.5 Conclusion and recommendations

This chapter proposed a new method to accurately reconstruct high-resolution time-space speeds from floating car data (FCD). This new method called TravRes, does this by iteratively reconstructing the (unobserved) probe vehicle trajectories between polling time instants, until the resulting time-space speed map is as consistent as possible with all probe vehicle reports. Like the method presented in the previous chapter, we can not guarantee that the reconstructed trajectories will converge in this iterative processing. The underlying rationale is simple: instead of assuming constant speeds between probe vehicle reports (which we refer to as the naive method or classic method) and deriving a time-space speed map by averaging these constant speeds, TravRes assumes and approximates probe vehicle trajectories which are consistent with all other probe vehicle reports. One of the main assumptions made is that speed is constant in (arbitrarily chosen) time-space regions. Compared to the naive method (classic method), the new method almost doubles the estimation accuracy, particularly in low-speed regions where such an increased accuracy is most valuable.

It is found that the polling interval (i.e. the time interval between probe vehicle reports) significantly influences estimation accuracy, and that polling intervals of 60sec can lead to twice better accuracy than 120sec polling intervals. It is, however, important to note that one should choose the size of the space-time grid cells in accordance with the polling interval, that is, each time-space cell had better cover a time period equal or larger than

the polling interval. The probe vehicle penetration rate (average percentage of equipped vehicles among the all vehicles) largely determines the spatio-temporal coverage, that is, the amount of time-space cells in which a speed can be estimated. From our results it appears that 3%-4% penetration can lead to both high coverage and accurate estimation.

By now, we already tackled the issue how to fuse travel times (low time-resolution travel time data in particular) by using data-data consistency approach. With this approach, very few assumptions are needed, but considerably improved speed estimations are achieved. However, there are still a lot left to be done. In addition to low resolution travel time data, there are also low-resolution positioning data which are widely available in wireless networks as mentioned in Chapter 2. Considering their wide availability, fusing these data will also be practical relevant. The next chapter will give a theoretical framework to fuse low-resolution positioning data. And this framework is still based on the idea of data-data consistency.

Chapter 7

Speed estimation by fusing low-resolution positioning data

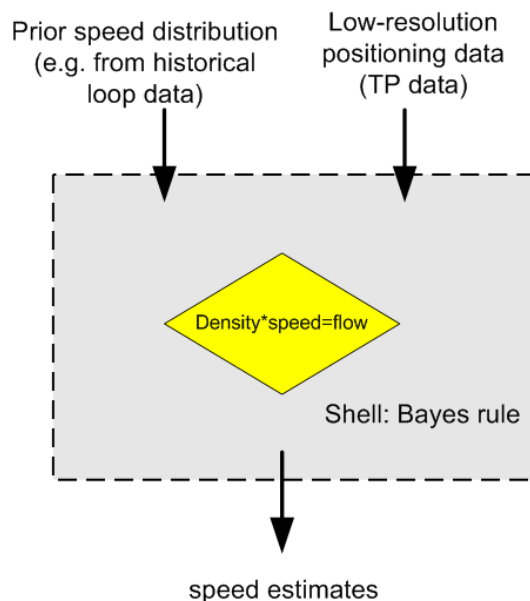


Figure 7.1: Illustration of the consistency rule in this chapter

Chapter 5 and 6 are concerned with low-resolution travel time data (low polling rates). This chapter will deal with another type of data, data which may not only have low time-resolution but also have quite low position-resolution. Such data cannot pinpoint the accurate positions of vehicles but can only give some location-specific information when and where the vehicles are located at the segment or cell level. For this reason, we will

refer to such data as topological position data or TP data.

In contrast to high-resolution positioning data, TP data cannot provide the distance component that is necessary for traffic speed estimation. However, considering the wide availability of TP data in the existing telecommunications network, there is still hope and benefits to make use of the data for traffic state estimation. The proposed method in this chapter is capable of fusing low resolution positioning data with other data sources, leading to more accurate and reliable speed estimation of relatively low bias.

This method is also based on data-data consistency approach. Instead of using $speed * time = distance$ law, we use $density * speed = flow$ under the assumption of homogeneous traffic in each discretized time-space cell. Two types of information are taken as input. One is prior distribution of traffic speeds from e.g. historical loop data, and the other is TP data, which actually contain information concerning flow and density. Bayes rule is used to update the prior speed distribution when TP data is given, such that posterior speed (distribution) with more reliability is given.

7.1 Introduction

We first look at high-resolution positioning data as GPS data. The emerging use of global positioning systems in mobile phones or as part of in-vehicle navigation systems makes it feasible to derive real-time traffic state information on the basis of these data. Since GPS-equipped vehicles or devices provide their geographic positions with time stamps, one can derive a moving vehicle's mean speed by dividing vehicle distance traveled by travel time. With more and more high-resolution probe vehicle data becoming available, accurate and reliable traffic state estimation can likely be achieved.

However, before state estimation from a very large sample size of GPS-based probe vehicles is fully achieved, some other data sources are needed. Lower resolution positioning data via cellular communications networks can be one of them. Therefore, the motivation of this chapter is to examine the potential benefits of fusing those low resolution data as extra information to achieve better traffic speed estimation.

7.1.1 Cellular networks and topology positioning data

Cellular phone location data is considered to be a low-resolution location data source. Cellular phones operate in a network made up of multiple radio cells, as shown in Figure 7.2. In each cell, a base station provides the communication service for phones. Base

stations are all interconnected so that an on-call phone can move from one cell to another without losing the connection. All cells are grouped into particular location areas (LA). Normally a cell hexagon ranges in size between 200 and 1000 meters in urban areas, and location areas range from 1 kilometer to several kilometers in size.

Previous research on wireless-network-based traffic monitoring and estimation (Smith et al. (2001)) has focused on positioning techniques (Smith (2007)) and the impact of probe penetration on the quality of these positioning techniques. Major positioning techniques include angle-of-arrival positioning, time-of-arrival positioning, time-difference-of-arrival positioning (Roos et al. (2002)), and the handoff approach (Zhao (2000)). These techniques obtain the geographical positions of probe vehicles so that the travel distance is available for traffic speed estimation. Previous field operational tests (Thiessenhusen et al. (2005)) assessed the accuracy of estimation with different methods and different probe penetration. In previous research, cellular data has been treated similarly to positioning data obtained from GPS.

This chapter describes a mechanism for tracking cellular phones at the cell level. Cell level data has low positioning accuracy, ranging between 100 meters and one kilometer, depending on the size of cell, not appropriate for travel distance estimation.

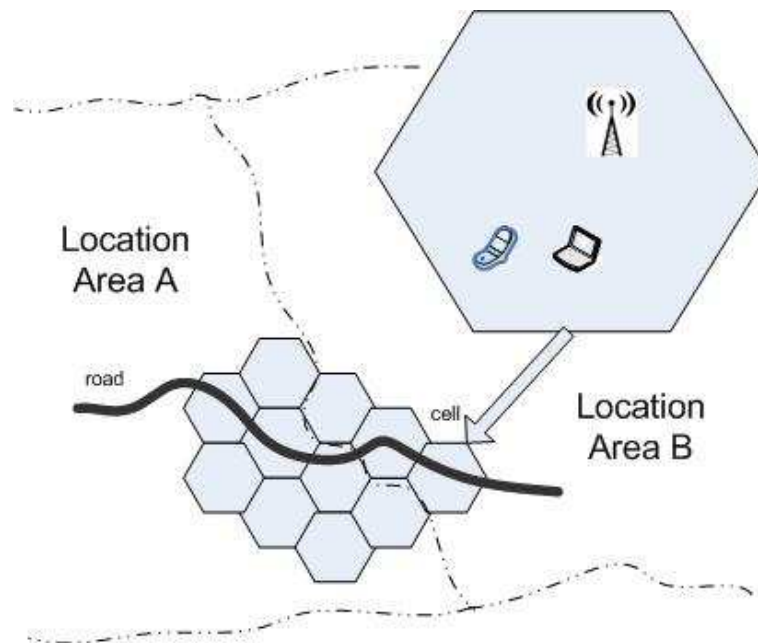


Figure 7.2: Example cellular network.

Technically, TP data are location data which don't provide exact geographical positions but point to a location area—a cell or a road segment. When a mobile phone in a vehicle

sends beacon signals periodically, the cellular networks are able to trace the phone and record the cell where it is located. Considering that beacon signal transmission is one way and occurs at a frequency on the order of minutes, the communication is relatively simple and low cost. In addition, TP data are more widely available in terms of time and space, since devices possibly providing TP data (e.g. mobile phones, laptops, iPhones, etc) are being widely used in communication networks and are becoming increasingly popular. In sum, the simplicity and wide availability of TP data may have potential for traffic estimation in large networks.

7.1.2 Challenge and objectives

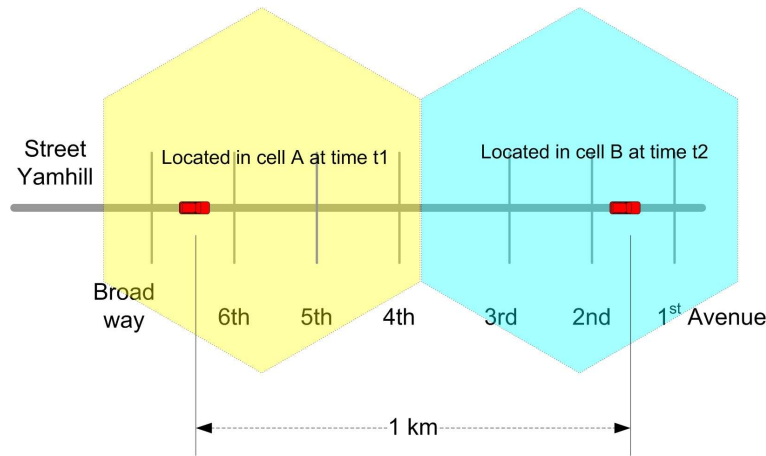


Figure 7.3: Illustration of location displacement of vehicles at different levels of resolution.

Before TP data can be used for traffic monitoring, a new method is needed for processing the data.

In order to satisfy the real-life application, below we list the objectives which this method is supposed to achieve.

- (a) Usage of TP data. For traffic monitoring with TP data, probe vehicle locations are grouped only according to their road segment or cell. As a result, the accuracy of positioning is only at the segment or cell level. So TP data can't be treated in the same way that GPS data are normally handled. In Figure 7.3, the vehicle could have provided accurate traffic estimation if it was known that it traversed 1 kilometer during the time interval between t_1 and t_2 . But with segment-level

accuracy, the vehicle can only report that it traversed the cell boundary and entered the next cell during this time period. So the proposed method must be able to handle this inaccuracy issue.

- (b) Error tolerance. The cell boundary is uncertain and varies with the signal power and signal distribution of the cellular network. Because the size of the cell is more or less uncertain, incorrect pre-matching of road segments may occur. Due to inaccurate positioning, probe vehicles may not be snapped into the correct cell especially when the vehicles are close to the boundaries. So the proposed method should be tolerant of errors and uncertainties.
- (c) Fusion with other data. TP data should not be the only data source for traffic estimation. It would be preferable for TP data to serve as added information, fused with other high-resolution and more reliable data sources, e.g. GPS data. The proposed method should include data fusion using low-resolution positioning data and high-resolution ones .
- (d) Magnitude of error in estimation. Traffic estimation is an initial step in the traffic management process. The estimated traffic states will be further used for e.g. travel time prediction. Traffic prediction not only needs the current state estimation but also needs the confidence on estimated results. For this purpose, the proposed method should also provide an error estimate, such as the variance.
- (e) Extension to network-wide estimation. A cell may not only cover a simple road segment but a portion of the road network. So the method should be easily extended to applications on network-wide traffic estimation with some simple modifications.

So we want the proposed method to possess several properties which together help to achieve the five objectives described above. The methodology and validation are presented next, followed by some conclusions and suggestions for further research.

7.2 Methodology

In the first place, we present the assumptions and physical laws used in this method. The main assumption is that the traffic is homogeneous and stationary in each time-space cell. TP data are assumed to be correct and reliable despite of the low-resolution. The prior distribution of speeds is assumed to be known. The physical law that is used is $density = flow/speed$.

This section is sub-divided in four parts:

- The first section uses an analogy to show the fundamental idea behind the method.
- The second section describes a stochastic process on vehicle counting which helps to set up a mathematical formula to fuse TP data with prior speed information from other data sources e.g. historical GPS data.
- The third section proposes an improved formula. It considers upstream traffic flow and better estimation of speed is expected.
- The last section uses order-statistics to analyze the theoretic variance of the speed estimation using this method.

7.2.1 Prototype and basic idea

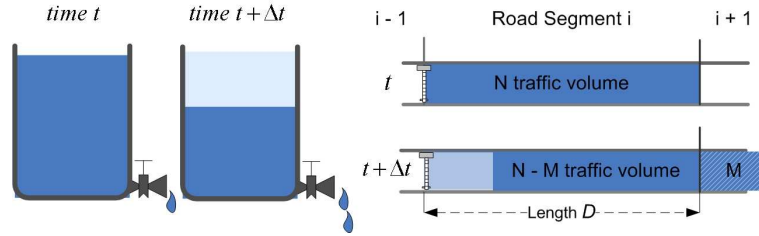


Figure 7.4: Model comparison of water container and roadway.

The initial inspiration for this method comes from a water-container prototype, in a similar fashion as first order traffic flow theory. Suppose there is a container with 1kg water, and after a valve is opened, the out-flowing rate is constantly 10g/seconds. As a result, it takes 100 seconds to drain the container after this valve is opened. Even when the liquid has non-uniform density in the container, the total draining time is still 100 seconds. Similarly, let us take traffic as water and a segment of roadway as the container. Suppose that there are N vehicles on the road segment of length D , and suppose that the upstream traffic is held so that there is no traffic flowing into the downstream segment (see Figure 7.4). If it takes time Δt to ‘drain’ *all* the traffic on this segment, the space mean speed can be estimated with $v \approx D/\Delta t$. Furthermore, if it is known that only M vehicles (out-flowing volume) enter the downstream segment, the speed can be estimated as

$$v \approx \frac{D \times M}{\Delta t \times N} \quad (7.1)$$

Equation 7.1 may still hold when vehicles are not uniformly distributed on the road segment, because the ratio M/N simply indicates the flow rate and largely determines the

traffic speed. For example, when congestion occurs, the upstream traffic is in free-flow and the downstream is in congestion. In this case, the ratio M/N becomes smaller and therefore the ‘effective’ speed on the whole segment will drop until the speed is approximately proportional to flow rate M/N .

However, if there is no way to measure the total number of vehicles and the outflow, this estimation would not be possible. As an alternative solution (see Figure 7.5), some indicators e.g. number number of particles or isotopes can be put in the container. These particles are assumed to be uniformly distributed in the water. With this addition, we need only measure the total amount of the indicators and their outflow amount so that the approximate estimate of the drop rate in the water level can be measured.

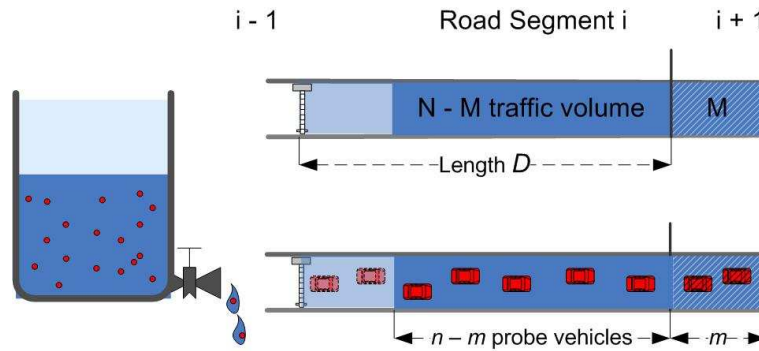


Figure 7.5: Illustration of all vehicles moving and probe vehicles moving

For traffic speed estimation, probe vehicles with segment level accuracy positioning can serve as such indicators. Suppose that there are n probe vehicles at time t and that m probe vehicles move to the next segment by $t + \Delta t$ (Refer to Figure 7.5). The pair (n, m) then becomes characteristic of the space mean speed on this segment during this time period.

Analogously to the above, this estimation reads

$$v \approx \frac{D * m}{\Delta t * n} \quad (7.2)$$

The question is, how good this estimate is, i.e., how much confidence we can have in this estimate. For example, the pairs $(n, m) = (100, 50)$ and $(n, m) = (10, 5)$ result in the same estimation, but the latter would be trusted less. Things become more complicated with prior knowledge of traffic speed (e.g. the speed distribution based on historical GPS data). In the next section, a more accurate estimation method will be given with (and without) prior knowledge of traffic speed combined.

7.2.2 Probe vehicle count process and speed re-estimation

This section will describe a mathematical formula in terms of $\hat{v} = f(v, m, n)$ as shown in Equation 7.11, where v depicts prior speed probabilistic information from another data source, m and n are the information from TP data, and $\hat{v}$ is the posterior speed estimation. As a result, this formula may serve in the data fusion of TP data and other data.

First, let us define a count process $\{\Psi(x), x \geq 0\}$, where x is the number of the counted vehicles (also traffic volume) and $\Psi(x)$ is the number of probe vehicles when x vehicles are counted. (Refer to Figure 7.6).

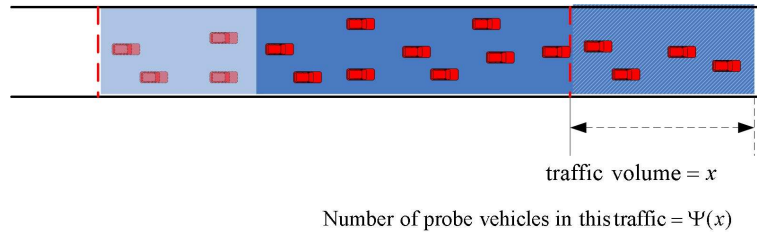


Figure 7.6: Illustration of a count process $\{\Psi(x), x \geq 0\}$

It is found that $\Psi(x)$ has the following four attributes:

- (1) $\Psi(0) = 0$: No vehicle, no probe vehicles.
- (2) $\Psi(x + y) - \Psi(x)$ is independent of $\Psi(x)$ for any $x \geq 0, y > 0$: Whether a vehicle is a probe vehicle has nothing to do with other vehicles.
- (3) $P(\Psi(x + h) - \Psi(x) = 1) = \lambda h + o(h)$ for small h , where λ is the proportionality factor associated with the percentage of probe vehicles; It is worth noting that this percentage is unknown.
- (4) $P(\Psi(x + h) - \Psi(x) \geq 2) = o(h)$: One vehicle implicates at most one probe vehicle.

As a result, $\{\Psi(x), x \geq 0\}$ is a Poisson process according to its definition (Medhi (2002)). It is worth to mention that $\Psi(x)$ has no business with the traffic condition e.g. congestion or free-flow. $\Psi(x)$ is a counting process, which shows how many probe vehicles are counted when counting vehicles of all kinds. For example, if all the vehicle are probe vehicles, it will read $\Psi(x) = x$. Taking another example, if no probe vehicle is found in counting the first 10 vehicles, we have $\Psi(10) = 0$.

To fuse the TP data with prior knowledge of the traffic speed distribution $P(v)$, a Bayesian rule will be employed to (re)-estimate the traffic speed with the pair (n, m) . As shown in Figure 7.5, according to the Bayesian rule, we have

$$\begin{aligned} P(v|\Psi(N) = n, \Psi(M) = m) &\propto \\ P(\Psi(M) = m|\Psi(N) = n, v) * P(v|\Psi(N) = n) \end{aligned} \quad (7.3)$$

where it is found that $P(v|\Psi(N) = n) = P(v)$ because $\Psi(N)$ is independent of v . It is straightforward to normalize it and get the exact probability. However, $P(\Psi(M) = m|\Psi(N) = n, v)$ is not easily deduced, so let us only focus on $P(\Psi(M) = m|\Psi(N) = n)$. In view of the conclusion that $\{\Psi(x), x \geq 0\}$ is a Poisson process, it can be deduced that

$$\begin{aligned} &P(\Psi(M) = m|\Psi(N) = n) \\ &= P(\Psi(M) = m, \Psi(N) = n)/P(\Psi(N) = n) \\ &= P(\Psi(M) = m, \Psi(N) - \Psi(M) = n - m)/P(\Psi(N) = n) \\ &= P(\Psi(M) = m) * P(\Psi(N) - \Psi(M) = n - m)/P(\Psi(N) = n) \\ &= \frac{(\lambda M)^m e^{-\lambda M}}{m!} * \frac{(\lambda(N - M))^{n-m} e^{-\lambda(N-M)}}{(n-m)!} * \frac{n!}{(\lambda N)^n e^{-\lambda N}} \\ &= \frac{n!}{m! * (n-m)!} \left(\frac{M}{N}\right)^m \left(1 - \frac{M}{N}\right)^{n-m} \end{aligned} \quad (7.4)$$

Thus, $P(\Psi(M) = m|\Psi(N) = n)$ becomes :

$$P(\Psi(M) = m|\Psi(N) = n) = \frac{n!}{m! * (n-m)!} \left(\frac{M}{N}\right)^m \left(1 - \frac{M}{N}\right)^{n-m} \quad (7.5)$$

It can be found that $P(\Psi(M) = m|\Psi(N) = n)$ already contained the speed information v as prior information. So when we assume $v \approx D * M/\Delta t * N$, we can rewrite the Equation 7.5 and get:

$$P(\Psi(M) = m|\Psi(N) = n) \approx \frac{n!}{m! * (n-m)!} \left(\frac{v\Delta t}{D}\right)^m \left(1 - \frac{v\Delta t}{D}\right)^{n-m} \quad (7.6)$$

It is found that Equation 7.6 includes an unknown parameter v . This unknown parameter is actually a random variable that needs to be estimated, which implicates that it is a given condition in this equation. So we can explicate this condition by further rewriting Equation ?? and get:

$$P(\Psi(M) = m | \Psi(N) = n, v) \approx \frac{n!}{m! * (n - m)!} \left(\frac{v\Delta t}{D} \right)^m \left(1 - \frac{v\Delta t}{D} \right)^{n-m} \quad (7.7)$$

Putting Equation 7.7 and $P(v | \Psi(N) = n) = P(v)$ into Equation 7.3, we finally obtain:

$$P(v | \Psi(N) = n, \Psi(M) = m) \propto \frac{n!}{m! * (n - m)!} \left(\frac{v\Delta t}{D} \right)^m \left(1 - \frac{v\Delta t}{D} \right)^{n-m} P(v) \quad (7.8)$$

where $m, n, v, P(v)$ and D are all known as mentioned above, while unknowns M, N and λ are not needed. An important benefit from this formula is that the percentage of probe vehicles is not needed.

For the estimate of traffic speed with prior knowledge of the speed distribution, we have

$$\begin{aligned} \hat{v} &= E(v | \Psi(N) = n, \Psi(M) = m) \\ &= \frac{1}{C} \int \frac{n!}{m! * (n - m)!} \left(\frac{v\Delta t}{D} \right)^m \left(1 - \frac{v\Delta t}{D} \right)^{n-m} P(v) v dv \end{aligned} \quad (7.9)$$

where

$$C = \int \frac{n!}{m! * (n - m)!} \left(\frac{v\Delta t}{D} \right)^m \left(1 - \frac{v\Delta t}{D} \right)^{n-m} P(v) dv \quad (7.10)$$

is a normalization factor. In particular, if there is no prior knowledge of the actual speed distribution, we simply assume $P(v)$ has a uniform distribution with regard to v . It is worth noting that the method is valid under the restriction

$$v\Delta t < D \quad (7.11)$$

This restriction implies that some of vehicles don't move to the next segment after one time interval.

7.2.3 Physical explanation and improved re-estimation

This section will present an improved formula shown in Equation 7.17 after giving a physical explanation. In Equation 7.8, $v\Delta t/D$ is the outflowing traffic (in terms of vehicle number) proportion and $1 - v\Delta t/D$ is the proportion of traffic that remains on this segment. Since speed v is the variable to be estimated, the two proportions are actually unknown. Thus, the pair (n, m) is needed to weigh the two proportions with regard to v . For a fixed v , a larger m means a larger outflow proportion, so it will be more heavily weighted. Furthermore, when considering the number of vehicles from the upstream

$$P(v | n, m) \propto \frac{n!}{m! * (n-m)!} \left(\frac{v\Delta t}{D} \right)^m \left(1 - \frac{v\Delta t}{D} \right)^{n-m} * P(v)$$

Number of probe vehicles in outflow traffic Number of probe vehicles in remain traffic
 Outflow traffic proportion Remaining traffic proportion

Figure 7.7: Physical explanation of Equation 7.8.

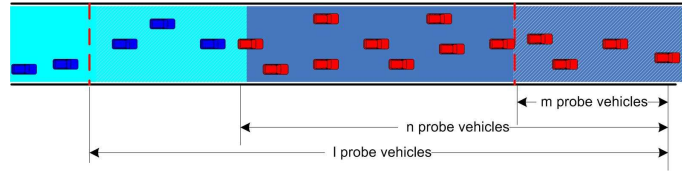


Figure 7.8: Probe vehicle in inflowing and outflowing traffic.

segment, more information can be added for estimation (Figure 7.8). We expect more accurate and reliable speed estimation can be achieved. Similarly, we consider three proportions: the outflow traffic proportion, the remaining traffic proportion and the inflow traffic proportion which are $\left(\frac{v\Delta t}{D+v\Delta t}\right)$, $\left(\frac{D-v\Delta t}{D+v\Delta t}\right)$ and $\left(\frac{D}{D+v\Delta t}\right)$ respectively. The corresponding weighting factors are m , $n - m$ and $l - n$. Analogously to the derivation of Equations 7.5, 7.7, 7.8, we can now write:

$$\begin{aligned}
 & P(\Psi(M) = m, \Psi(N) = n | \Psi(L) = l) \\
 = & \frac{P(\Psi(M) = m, \Psi(N) = n, \Psi(L) = l)}{P(\Psi(L) = l)} \\
 = & \frac{P(\Psi(M) = m, \Psi(N) - \Psi(M) = n - m, \Psi(L) - \Psi(N) = l - n)}{P(\Psi(L) = l)} \\
 = & \frac{P(\Psi(M) = m)P(\Psi(N) - \Psi(M) = n - m)P(\Psi(L) - \Psi(N) = l - n)}{P(\Psi(L) = l)} \\
 = & \frac{(\lambda M)^m e^{-\lambda M}}{m!} \frac{(\lambda(N - M))^{n-m} e^{-\lambda(N-M)}}{(n-m)!} * \frac{(\lambda(L - N))^{l-n} e^{-\lambda(L-N)}}{(l-n)!} \frac{l!}{(\lambda L)^l e^{-\lambda L}} \\
 = & \frac{l!}{m! * (n-m)! * (l-n)!} \left(\frac{M}{L} \right)^m \left(\frac{N-M}{L} \right)^{n-m} \left(1 - \frac{N}{L} \right)^{l-n}
 \end{aligned} \tag{7.12}$$

Moreover, we have the below equations as approximation:

$$\frac{M}{L} = \frac{v\Delta t}{D + v\Delta t} \quad (7.13)$$

$$\frac{N - M}{L} = \frac{D - v\Delta t}{D + v\Delta t} \quad (7.14)$$

$$\frac{N}{L} = \frac{D}{D + v\Delta t} \quad (7.15)$$

and

$$P(v|\Psi(N) = n, \Psi(M) = m, \Psi(L) = l) \propto P(\Psi(M) = m, \Psi(N) = n|\Psi(L) = l, v) * P(v|\Psi(N) = n) \quad (7.16)$$

Similar to Equation 7.11, the estimation then becomes:

$$\begin{aligned} \hat{v} &= E(v|n, m, l) = \frac{1}{C} \frac{l!}{m!(n-m)!(l-n)!} \\ &\quad * \int \left(\frac{v\Delta t}{D + v\Delta t} \right)^m \left(\frac{D - v\Delta t}{D + v\Delta t} \right)^{n-m} \left(\frac{D}{D + v\Delta t} \right)^{l-n} P(v) v dv \end{aligned} \quad (7.17)$$

where

$$\begin{aligned} C &= \frac{l!}{m!(n-m)!(l-n)!} \\ &\quad * \int \left(\frac{v\Delta t}{D + v\Delta t} \right)^m \left(\frac{D - v\Delta t}{D + v\Delta t} \right)^{n-m} \left(\frac{D}{D + v\Delta t} \right)^{l-n} P(v) dv \end{aligned} \quad (7.18)$$

7.2.4 Analysis of variance

This section presents an approximate variance analysis of the estimation results. Variance $var(\hat{v})$ is a critical indicator of estimation quality, which will help practitioners to make an optimal/economical configuration of number of probe vehicles and sampling time interval, and will also contribute to travel time prediction and traffic management since the variance is known.

Ideally $var(\hat{v})$ should be used to estimate the variance of speed estimation. Instead, we use a proxy, namely $var(m/n)$, which provides us with an analytical solution. According to the container-draining model, we have speed $v \approx D * M/\Delta t * N$, in which speed is determined by the ratio M/N . Further, m is the number of probe vehicles from M vehicles and n is from N , therefore the ratio m/n can indicate the ratio M/N and speed

v. Although $\text{var}(m/n)$ is a rough and approximate indicator of $\text{var}(\hat{v})$, it still enables us to give some general analysis and results, for example, ‘which parameters have what kind of effect on the variance’.

Seen as in Figure 7.8 and Equation 7.7, the proportion m/n plays a critical role in this method. It will lead to an accurate estimation if m/n is almost equal to M/N . However, all probe vehicles are uniformly distributed, resulting in an inconsistency between m/n and M/N . This analysis begins with a brief introduction of a typical order statistic. Suppose samples follow a uniform distribution over the range $[0, 1]$, and the samples are sorted in increasing order as $U_{(1)}, U_{(2)} \dots U_{(m)} \dots U_{(n)}$, where m is the p (th) sample quantile, that is $m = \lceil np \rceil$. With the knowledge of order statistics, $U_{(m)}$ is asymptotically normally distributed, that is

$$U_{(m)} \sim AN\left(p, \frac{p(1-p)}{n}\right) \quad (7.19)$$

In our method, we assume all vehicles are uniformly distributed on the road, so it makes sense to let $U_{(m)} = m/n$ and $p = M/N = v\Delta t/D$. As a result, we have

$$\frac{m}{n} \sim AN\left(\frac{v\Delta t}{D}, \frac{v\Delta t(D - v\Delta t)}{D^2 n}\right) \quad (7.20)$$

It can be further transformed into

$$\frac{m}{n} * \frac{D}{\Delta t} \sim AN\left(v, \frac{v(D - v\Delta t)}{\Delta t n}\right) \quad (7.21)$$

If we let

$$\sigma^* = \sqrt{\frac{v(D - v\Delta t)}{\Delta t n}} \quad (7.22)$$

then σ^* can be approximately taken as the standard deviation of the absolute error in speed estimation. This expression with respect to σ^* shows that the variance of the error in speed estimation can be diminished by making Δt as large as possible or by putting more probe vehicles in operation.

If the input flow of probe vehicles is counted as shown in Figure 7.8, the information used for estimation will double. Assuming that input flow and output flow play identical and independent roles in this estimation, both sources of information will upgrade Equation 7.22 into:

$$\sigma^{**} = \sqrt{\frac{v(D - v\Delta t)}{2\Delta t n}} \quad (7.23)$$

where σ^{**} is the standard deviation when taking into account both inflow and outflow.

7.3 Validation

This section illustrates how the method accomplishes the five objectives promised in the introduction.

- (1) First, the validation of the proposed approach will be shown by means of an experiment with synthetic data generated using a microscopic simulation program. In this section, the accomplishment of the objective (a) and (c) is shown.
- (2) Following that, the effect of the number of probe vehicles on estimation accuracy is analyzed, including an estimate of an economically optimal percentage of probe vehicles. In this section, the accomplishment of objective (d) is shown.
- (3) Third, the impact of different prior distribution forms on the posterior distribution is given, in which the accomplishment of objective (b) is shown.
- (4) The final section concerns the network-wide traffic estimation by an update approach so that Equation 7.17 fits in network cases, in which the accomplishment of objective (e) is shown

7.3.1 Validation by synthetic data

The simulation environment used for data generation is VISSIM 4.20. We will show the performance of this algorithm in a typical scenario, in which the ground-truth speeds are perturbed with both non-uniformly structural deviations and considerable random errors, leading to a very poor observation of speeds. When considering the TP data, estimated speeds are obtained by this method. The contrast between ground-truth speeds, observed and estimated (corrected) is shown.

Using VISSIM a 35.25 km section of freeway was coded (Figure 7.9), made up of 15 segments each 2.35 km long. At the end of this freeway, a speed limit control was established, resulting in a simulated incident. Every 1 minute, probe vehicles (10% of total vehicles) reported which segments they were positioned on. So we have $D = 2.35\text{km}$ and time interval $\Delta t = 1$ minute. VISSIM provides the exact position of each vehicle, but we reduced the position accuracy to the segment level. Since the segment boundaries are pre-determined, we know how many probe vehicles remain on a segment and how many flow in and out. That way, the information (m, l, n) is known on the specific segment.

In addition, we assumed observed speeds with non-uniformly structural deviations up to

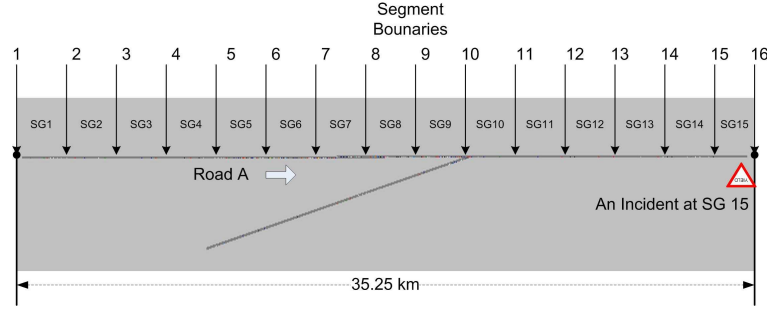


Figure 7.9: The layout of road infrastructure, segment and scenario settings in VISSIM.

80% plus random errors based with regard to ground-truth speeds:

$$\text{Structural Deviation rate}(\%) = \left(\frac{0.73 - 1.9}{110} * v + 1.85 \right) * 100\% \quad (7.24)$$

$$\text{Random Error rate}(\%) = N(0, 0.13^2) * 100\% \quad (7.25)$$

where v was a ground-truth speed and $N(0, 0.13^2)$ was a normal distribution with mean 0 and standard deviation 0.13. This way, the observed speed v^- was fabricated by applying the structural deviation rate and random error rate to the ground truth speed. In particular, the structural deviation rate was primarily dependent on actual speeds. When speeds were higher than 80 km/h, they were underestimated. When lower than 80 km/h, they were overestimated. Then, a uniform probability distribution with regard to v was assumed to be

$$P(v) = \frac{1}{1.5v^- - 0.5v^-} \quad (7.26)$$

where $P(*)$ is a probability density function. With more empirical analysis, one may assume that v follows other possible distributions such as normal or Poisson or with more accurate parameters. For example, GPS data provide accurate traffic speed information but they may have only sparse time-space coverage. We may make some inferences on these GPS data, deducing the most possible prior speed distribution.

Now that we have the information concerning (m, l, n) , prior speed distribution $P(v)$, segment length D and sampling time interval Δt , we apply Equation 7.17, leading to the posterior speed estimation on a specific road segment. When employing the above mentioned method on each segment, the space mean speeds on all segments of the whole corridor can be estimated.

Figure 7.10(b) shows the perturbed speed contour plots with bias and random errors. After applying this method, the bias is largely removed and many details become visible,

e.g. more detailed color-gradation. The contrast between Figure 7.10(a), Figure 7.10(b) and Figure 7.10(c) shows the good performance of this method in correcting errors. The average relative-error on the entire time space region sharply dropped from 41% to 10.5%.

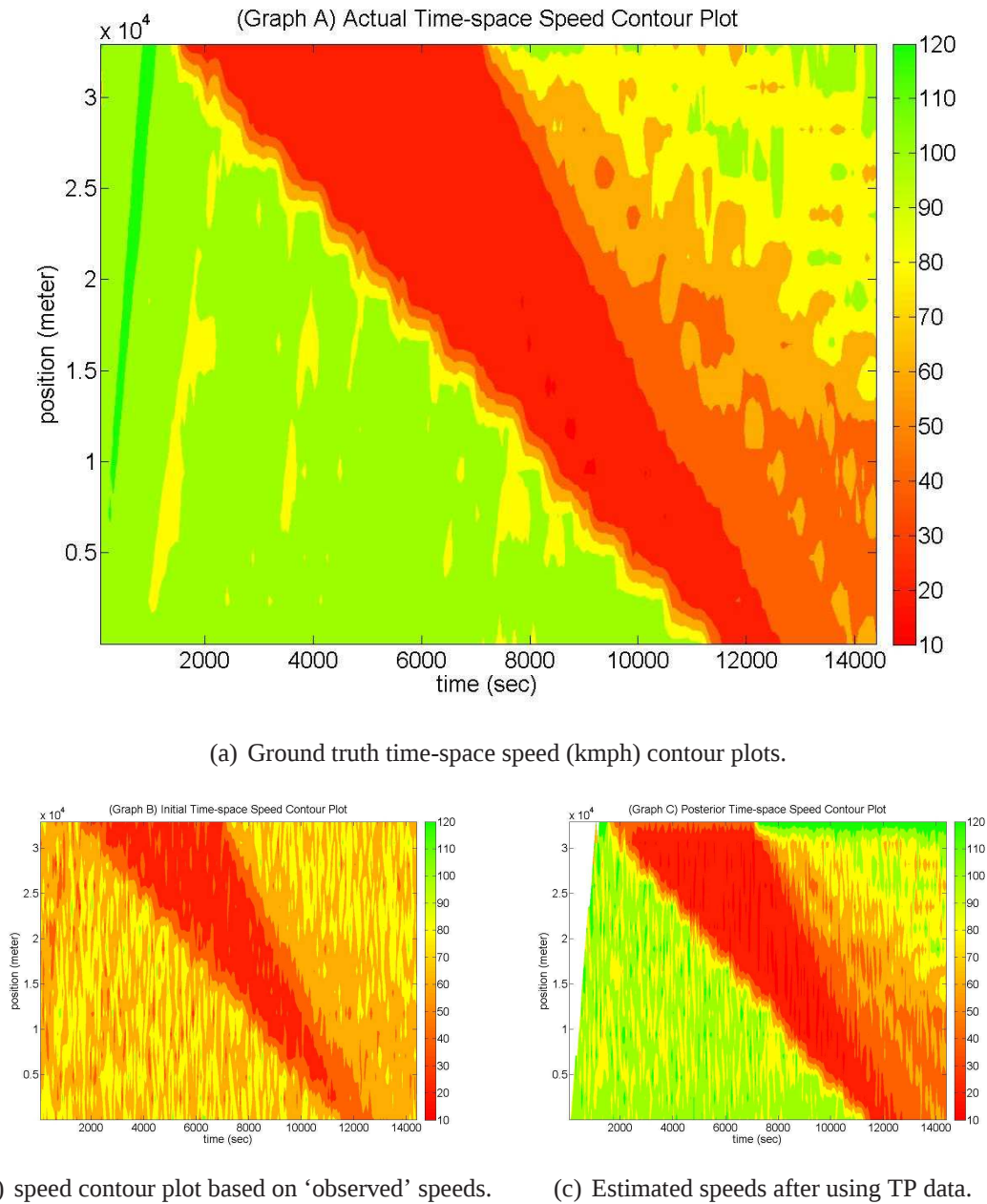


Figure 7.10: Time-space speed plot: comparison between ground-truth, observed and corrected speeds after applying this method

In Figure 7.11(a) and Figure 7.11(b), corrections of structural deviation are also noticeable regardless of the overestimation speed or underestimation. The random errors are also essentially diminished. After correcting the bias and diminishing the random errors, it would be possible to use other filtering techniques, e.g. moving average, Butterworth filtering or Kalman filtering to further improve the estimation results.

7.3.2 Impact of penetration of probe vehicles

Next, the analysis of the pairs (n, m) is performed with $D = 2.35\text{km}$, which partially validates the error estimation Equations 7.22 and 7.23 and further leads to an estimate of the economically optimal percentage of probe vehicles.

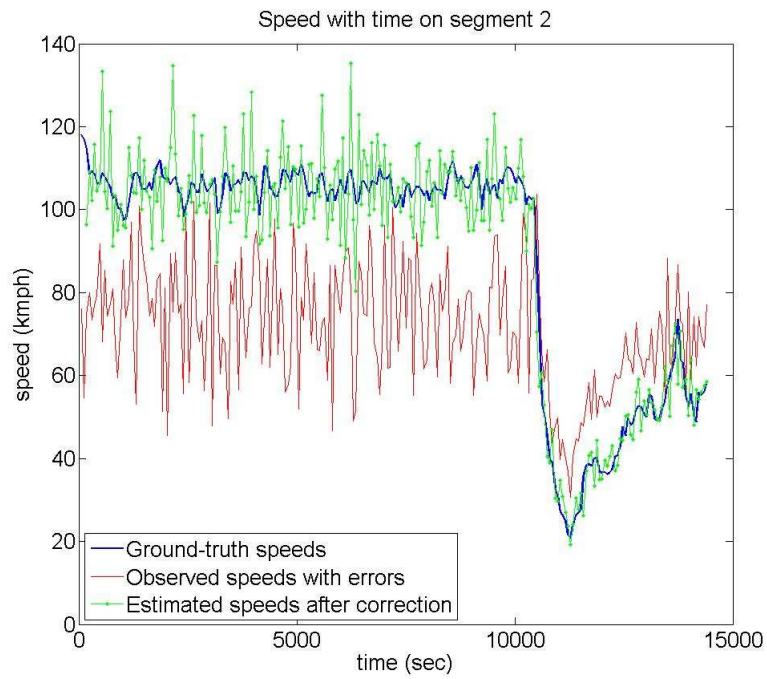
In this analysis, the prior probability distribution of speed is assumed to be uniform, ranging from 0 through 141 (kmph), and the proportional rate of n over m is fixed at 2 : 1. With an increasing number of probes, the posterior probability distribution evolves steadily (Refer to Figure 7.12(a)), resulting in smaller standard deviations. So the pair (n, m) plays a critical role in the accuracy and reliability of the posterior speed estimation.

In order to estimate how the number of probes affects the accuracy, an indicator needs to be defined. Since the posterior distributions approximately follow normal, pseudo standard deviations may be defined by σ as shown in Figure 7.12(b).

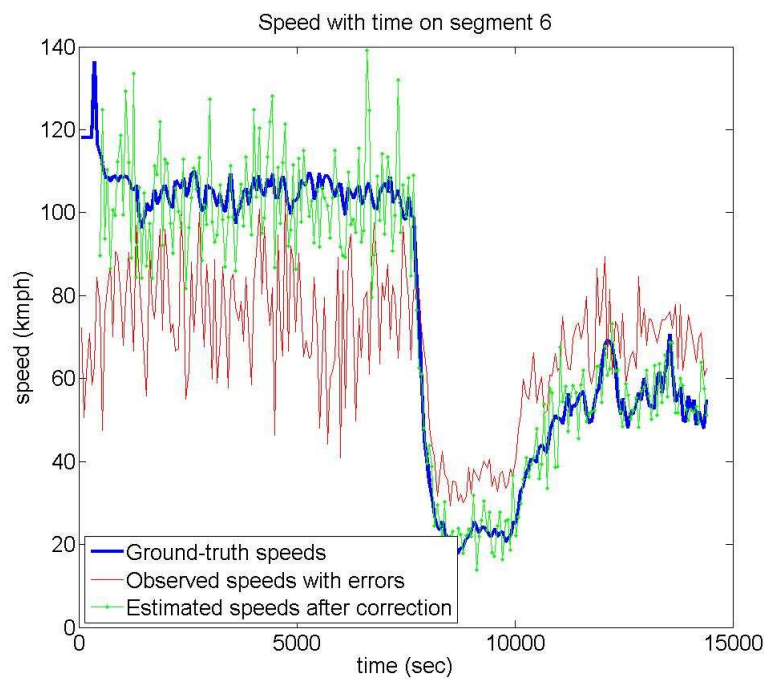
Although this defined σ is not an actual standard deviation, it is able to indicate a standard deviation. With this pseudo deviation, the relationship between σ and m under several n/m proportions is plotted in Figure 7.13(a). Under the $\log_{10}$ scale, it is found $\sigma \propto m^\alpha$ approximately, where $\alpha \approx -0.5$. We can see this relationship is fixed under different n/m proportions (Refer to the parallel lines in Figure 7.13(a)). Thus, for a fixed ratio n/m , we have $n \propto m$ and therefore $\sigma \propto n^\alpha$ can be approximately established.

In addition, some interesting results can be found in Figure 7.13(a). For a fixed m , when $n = 1.03m$ or $n = 8m$, σ is small. However, when $1.03m < n < 8m$, σ is relatively larger. Below is an explanation. When n/m becomes larger with fixed m , the smaller proportion of out-flowing vehicles can be observed. So the error should have been larger due to smaller proportions of out-flowing observation. In the meantime, when n/m becomes larger with fixed m , the total observations n will be larger. With more observations, the estimation error should drop. Forced by the both factors, we get the results as shown in Figure 7.13(b). In this figure, the theoretical curve is based on Equation 7.22. The ‘actual’ and theoretical curves do not overlap because the actual curve only uses the approximate σ as shown in Graph (B) in Figure 7.12.

Being an application, considering the cost of probe equipment, the above conclusion is

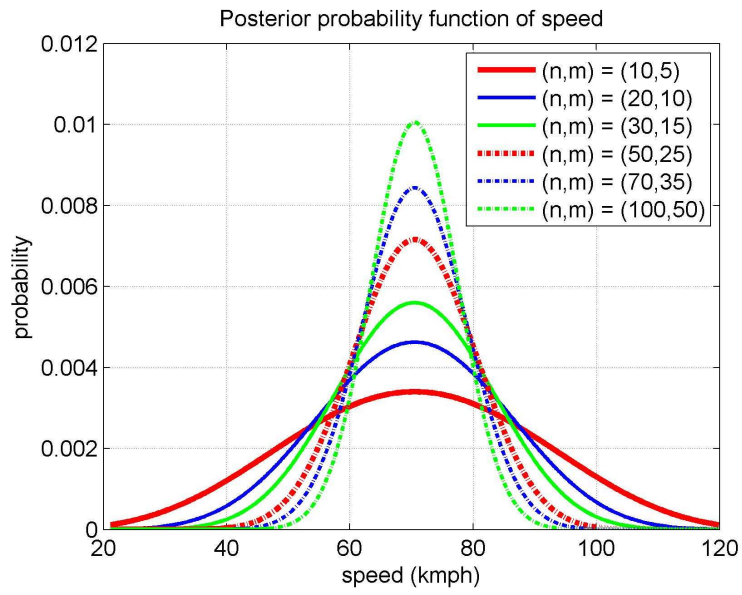


(a)

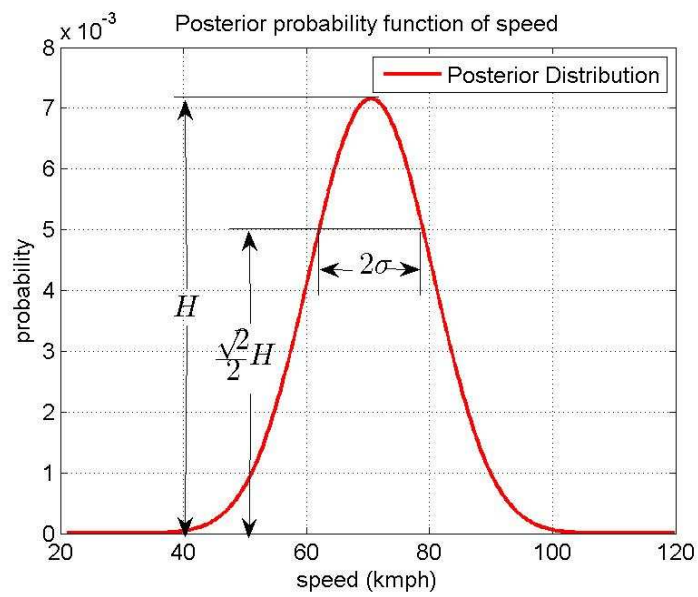


(b)

Figure 7.11: (a) Speed time series plots on segment 2. Blue curve represents ground-truth speeds.
 (b) Speed time series plots on segment 6. Blue curve represents ground-truth speeds.



(a)



(b)

Figure 7.12: (A) The influence of number of probe vehicles on posterior speed probability distribution under various (n, m) pairs.
 (B) Illustration of the defined deviation σ

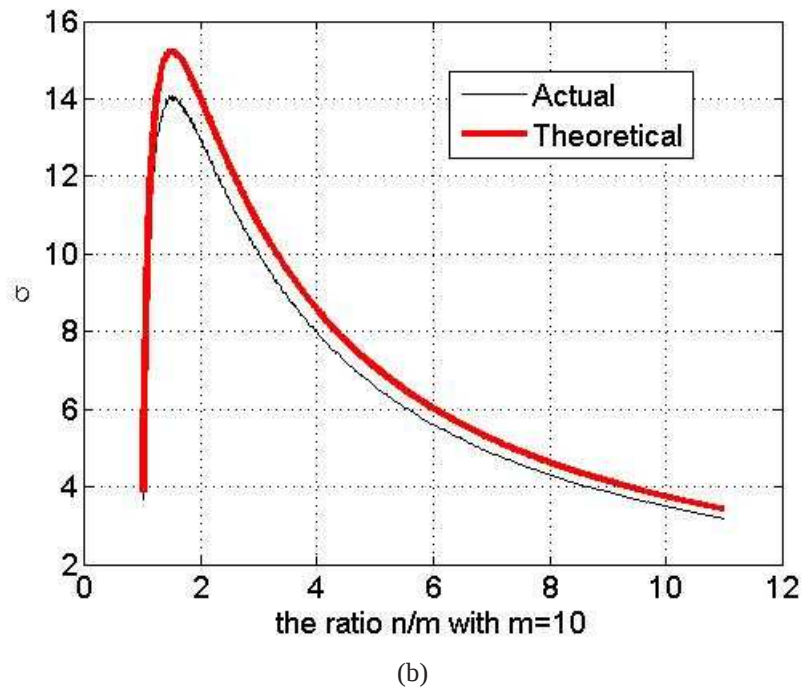
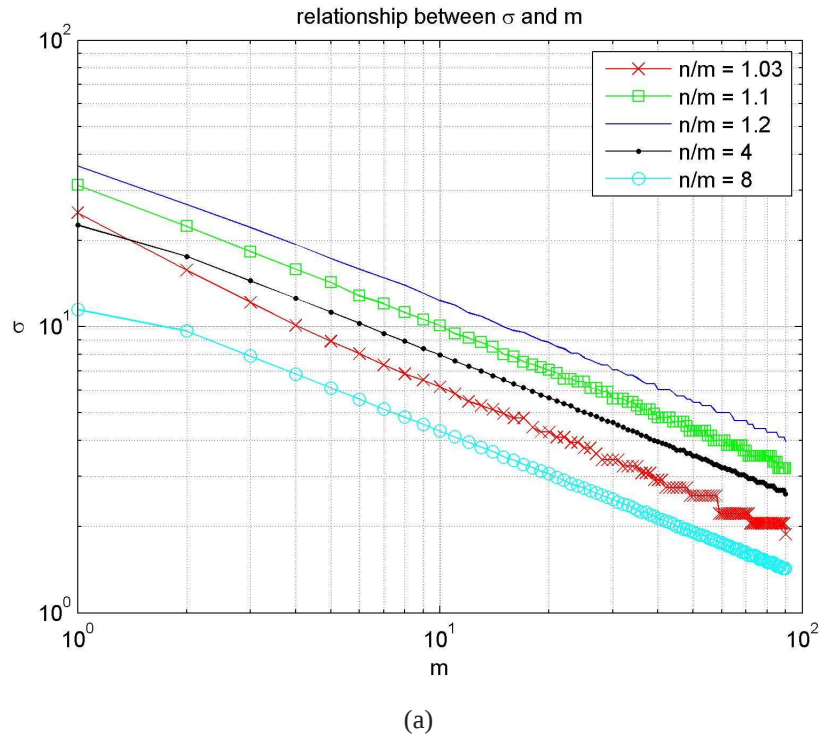


Figure 7.13: (a) The relation between the vehicle number and σ .
 (b) The relation between σ and the ratio n/m with $m=10$. The theoretical line is based on Equation 7.22.

beneficial for defining an optimal percentage of probe vehicles. For a general example, when n low-resolution positioning probe vehicles have already been available, the added benefit from installation of one additional probe is diminished as $B(n) = K[n^\alpha - (n+1)^\alpha]$, K is the multiplier factor for the trade off between diminished error and benefits in terms of money. If $B(n)$ is smaller than the unit cost, it is better to stop adding more probes. In this way, the optimal percentage of probe vehicles can be determined.

7.3.3 Impacts of different prior speed probability distributions

In traffic operations, it is difficult to estimate the form of the prior speed distribution, which may change with time or circumstances.

Here we assume three prior distributions of substantially different profiles-normal, triangular, and uniform (Refer to Figure 7.14(a)). Given the fixed pair $(n, m) = (30, 15)$ and $D = 2.35\text{km}$, the corresponding posterior distributions all look like normal (see Figure 7.14(b)). Table 7.1 shows estimates of the parameters of a normal distribution for each of the three of the three prior distributions.

Table 7.1: Given the different prior distributions, the posterior distributions fit into Normal distributions with Standard Error.

Prior distributions	Uniform	Triangle	Normal
$\hat{\mu}$	71.2 ± 0.1	71.0 ± 0.1	70.7 ± 0.1
$\hat{\sigma}$	13.0 ± 0.1	11.8 ± 0.1	11.0 ± 0.1

This result can be understood by looking at the expression

$$\frac{n!}{m! * (n - m)!} \left(\frac{v\Delta t}{D} \right)^m \left(1 - \frac{v\Delta t}{D} \right)^{n-m} \quad (7.27)$$

which serves an important role in our method. This is exactly the expression of a binomial distribution, which approximates to normal distribution with n approaching infinity. However, it is worth to mention that the posterior distribution may not exactly be normal or asymptotically normal due to the appended $P(v)$.

It can be seen that this result does not highly rely on the types of prior distributions. As mentioned above, the boundary of a cell is uncertain to some extent, which boils down to the uncertainty in prior information. This method is able to tackle the uncertainty for it is robust to prior information, always keeping an invariant distribution profile. When a vehicle is close to the boundary, it is hard to snap it to the correct cell. But it can be snapped into the correct cell with some probability. Since the method is embedded in a

framework of probability with invariant profiles, it becomes tolerant of the probabilistic errors.

7.3.4 Network-wide traffic speed estimation

For network-wide traffic estimation using TP data, the above method can be extended. As shown in Figure 7.15, a hypothetical road network is circled. Within this area, probe vehicles flow in and out, while others remain inside during Δt , leading to l, m, n which are required in Equation 7.17. As for the road length D , an equivalent value is needed for the network case. Considering the O-D pairs and corresponding traffic flow, such an equivalence is given as $D = \sum_{i,j} N_{ij} * D_{ij} / \sum_{i,j} N_{ij}$, where N_{ij} is the traffic flow on route $i \rightarrow j$, and D_{ij} is the length of this route. With an equivalent D available, Equation 7.17 can be used to estimate the average speed on the road network. As a result, this method is applicable for the network, particularly for urban network cases.

7.4 Conclusions and recommendations

In this chapter, we have proposed an algorithm for using segment-level topological positioning (TP) data for traffic speed estimation. In contrast with high accuracy position data from GPS, topological position data provide only lower resolution segment-level vehicle location information. For this reason, using TP data is not straight-forward like GPS data. However, as shown in this chapter TP data can be a valuable additional data source for traffic state estimation since they are widely available via the existing communication networks. Based on the proposed algorithm, TP data can be used alone or with other data sources, leading to more flexible traffic applications.

This algorithm corrects strongly biased prior speed measurements and reduces the impact of random errors. To run the algorithm, the percentage of probe vehicles is not required to be known, but a higher percentage contributes to the accuracy and reliability based on the finding $\sigma \approx n^\alpha$ ($\alpha \approx -0.5$). The algorithm has sufficient robustness to make the posterior distributions follow approximately normal under different profiles of prior distributions. Since the method is conceptually simple, it can be extended to fit in network-wide traffic speed estimation, which is the next step in this research. It is also notable that there are some technical barriers to be overcome when it comes to practical operations. For example, the vehicles/cell-phones that are not traveling on the road must be identified.

By now, we have proposed three algorithms (methods) to fuse loop speeds and floating

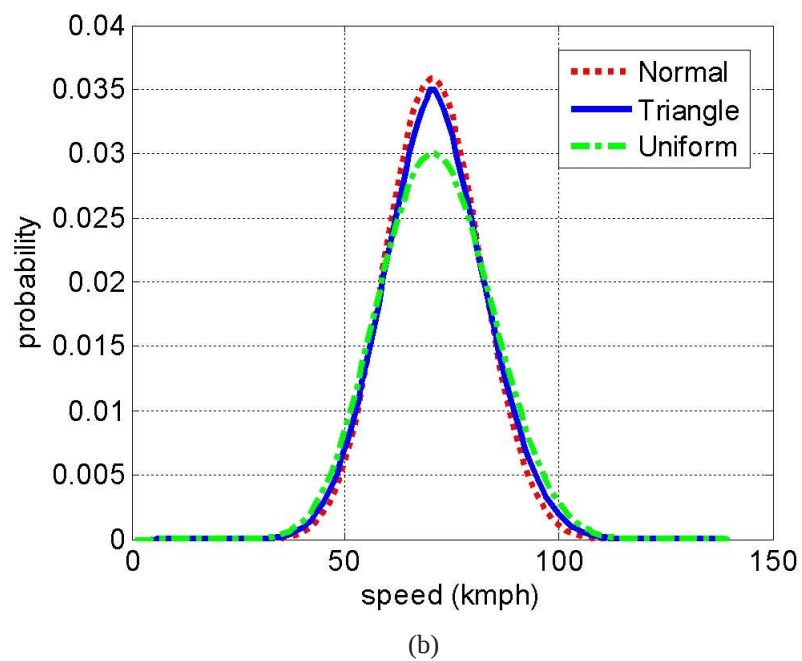
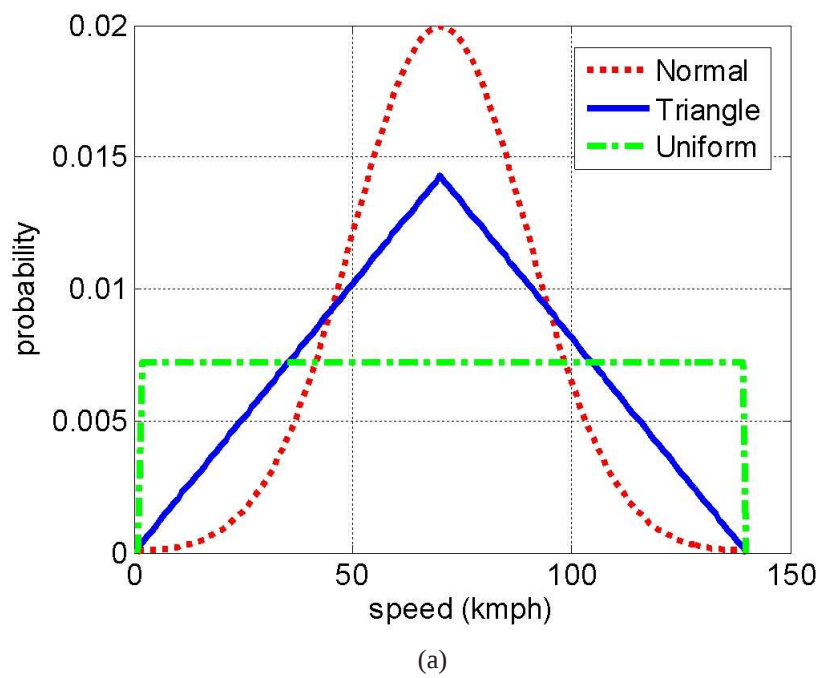


Figure 7.14: The influence of the forms of probability distribution. (a) Prior speed probability distribution ; (b) Posterior speed probability distribution.

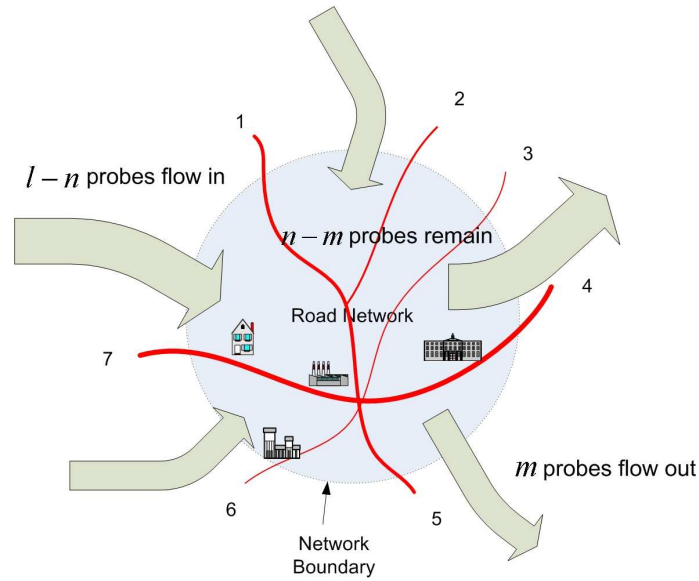


Figure 7.15: Road network and throughput of probe vehicles.

car data including travel times captured by cameras. Chapter 5 and 6 focus on the data which have low-resolution in time horizon. This chapter focused on the floating car data which shows the low-resolution in positioning. The output from these methods are speeds. But there is another type of very important information which has not been fused in our methods. This type of information is traffic flow by loop detectors. The next chapter will add the ingredient of flow observed at cross-sections and fuse it with loop speeds and travel times. As a result, the output not only contains the speeds but also the density, which greatly improves the applicability of the method to for instance short-term traffic predictions using traffic flow models.

Chapter 8

An integrated algorithm for fusing travel times, local speed and flow

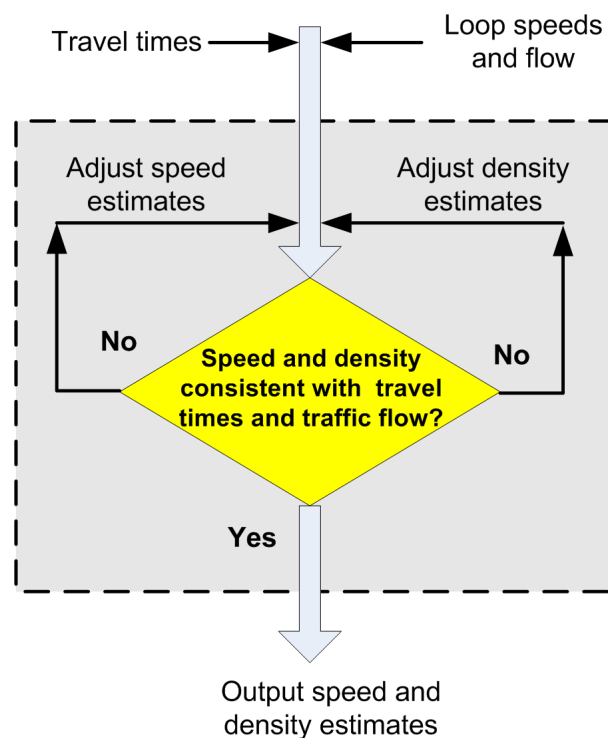


Figure 8.1: Illustration of Data-data consistency in this chapter

The previous chapters 5, 6 and 7 mainly focus on the estimation of traffic speeds by fusing local speeds (e.g. loop speeds) and AVI data (e.g. FCD). However, many forms of

detectors such as loop detectors may also provide traffic flow. Further more, the estimation of traffic density is also important in traffic management and operation. This chapter proposes an algorithm (method) to fuse loop speeds, loop flow and travel times from AVI system. This algorithm is able to provide estimates of both speeds and density.

The algorithm uses the trajectory reconstruction technique which has been presented in Chapter 5. Then it uses loop flow to estimate traffic density on each road segment. This algorithm is still based on data-data consistency approach. The major assumption is homogeneous traffic in each time-space cell, under which the physics law $speed * time = distance$ and $density * speed = flow$ can be simply used. Figure 8.1 shows this consistency approach in this chapter.

Table 8.1: Symbol list

L_j	Length of road segment j
$t_j^{(k)}$	Time moment when probe vehicle k enters road segment j
$\hat{t}_j^{(k)}$	Estimated time moment when probe vehicle k enters road segment j
$tt_j^{(k)}$	Travel time of probe vehicle k on road segment j
$\hat{tt}_j^{(k)}$	Estimated travel time of probe vehicle k on road segment j
$tt^{(k)}(i, j)$	Estimated duration of vehicle k dwelling in cell (i, j)
$\hat{s}^{(k)}(i, j)$	Estimated traveled distance of vehicle k in cell (i, j)
$\hat{q}^-(i, j)$	Measured flow in cell (i, j) by loop detectors.
$\hat{v}^-(i, j)$	Measured (biased) speed in cell (i, j) by loop detectors.
$\hat{v}^{(k)}(i, j)$	'Measured' speed in cell (i, j) by trajectory k . Similarly for density $\hat{\rho}^{(k)}(i, j)$ and flow $\hat{q}^{(k)}(i, j)$
$\hat{v}(i, j)$	The final estimated speed in cell (i, j) by using travel times, speed and flow. Similarly for density $\hat{\rho}(i, j)$ and flow $\hat{q}(i, j)$

8.1 Introduction

8.1.1 Basic relationships between traffic variables

Let's recall the characteristics of traffic data and traffic flow operations which have been presented in Chapter 3. There are three major variables in macroscopic traffic. They are density (ρ), speed (v) and flow (q). The evolution of these traffic states can be compactly and efficiently visualized by means of time-space contour plots. These plots are discrete representations of traffic states by using discretized time-space cells and color

(e.g. Figure 8.8). This way, the traffic states at the given time and location can be simply shown. In each cell, the traffic is assumed to be homogeneous and stationary. According to Edie's definition (Edie (1965)), these three traffic variables have such relationship between one another as shown in the below equation:

$$q = \rho v \quad (8.1)$$

where q , ρ and v are all time-space mean quantities.

In addition, there is a physics law in macroscopic traffic, called *Vehicle Conservation Law*: “the change in vehicle number on a road segment equals to the net difference between inflowing vehicle number and outflowing vehicle number”. It reads

$$\rho(i, j) = \rho(i - 1, j) + \frac{\Delta t}{L_j} (q(i, j - 1) - q(i, j)) \quad (8.2)$$

where $i = 1, 2, 3 \dots$ represents discrete time, j indicates the location, L_j is the length of the road segment j and Δt is the span of one discrete time.

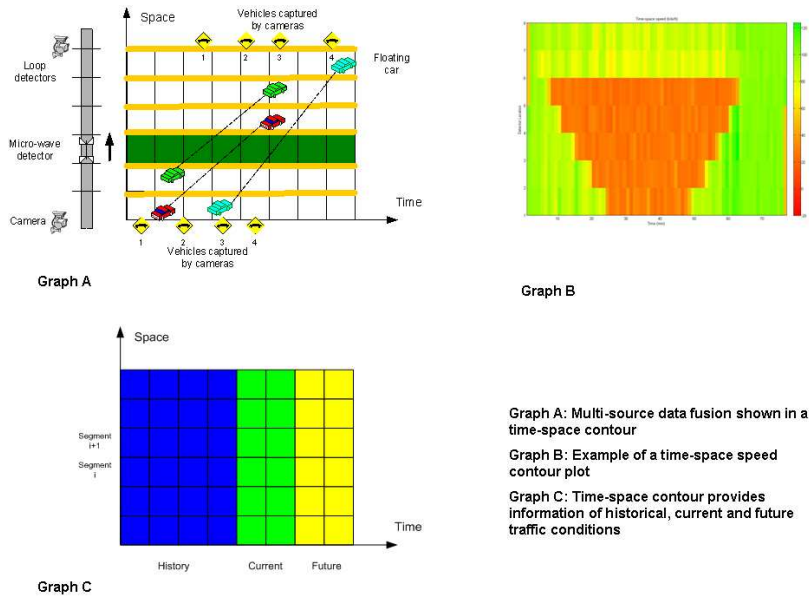


Figure 8.2: Time-space contour plot and its functions

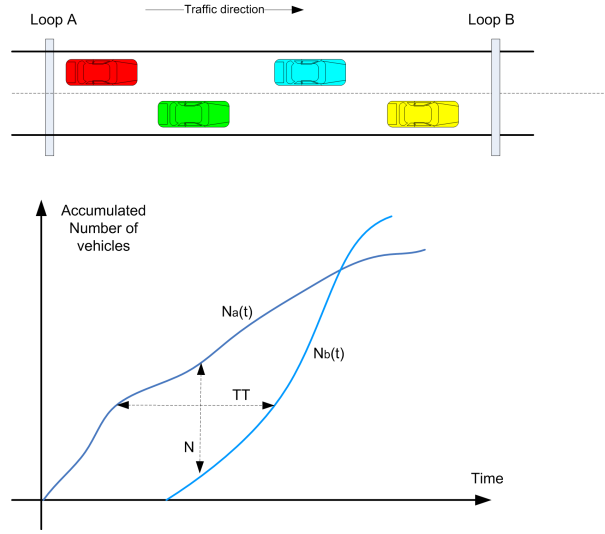


Figure 8.3: The accumulated traffic counts from loop detectors

8.1.2 Impact of traffic flow data

The most common data available for traffic state estimation come from (dual) loop detectors. As we already mentioned, loop speeds may probably be biased. This bias is significant, specifically under congested (low-speed) conditions has been demonstrated for example by Treiber and Helbing (2002) for estimating travel times (errors of over 30%), and by Knoop et al. (2007) for estimating densities, where the resulting errors can mount up to over 100%. It is true particularly for the data collection system in the Netherlands. In the previous chapters, we put a lot of effort on local speed measurements. Now we push forward and put another common data into account.

Apart from speed measurements, loop detectors and other forms of detectors may be able to count the number of vehicles that pass a location during a certain interval. These counts lead to estimates of flow q , the number of vehicles passing per unit time. Theoretically, the accumulated vehicle counts may tell travel time and vehicle density (or vehicle number) on a closed road section between two consecutive loop detectors as shown in Figure 8.3. In this figure, $N_a(t)$ and $N_b(t)$ are the accumulated vehicle counts from loop A and loop B respectively. Given the initial condition that no vehicle is on the road section, $N = N_a(t) - N_b(t)$ is the number of vehicle on this road section, and $TT(t)|N_a(t) = N_b(t + TT)$ can be taken as the travel time from loop A to B under the assumption of no overtaking. These estimates are reliable and accurate if loop detectors made no errors in counting vehicles.

However, correct estimates cannot be obtained in reality, due to error accumulation. In

the above example, the errors in $N_a(t)$ and $N_b(t)$ are accumulated with the time t . As a result, it may be found that the total number vehicle in a road segment is minus thousands in the end. We made an empirical study of one-day data on a one-kilometer section of highway A13 in Netherlands. The data was from Regionlab-Delft which stores the traffic data in Zuid-Holland region. With these data, it was found, during the period from 6:00 to 20:00, the total number of inflow vehicles was 64823 counted by the upstream loop detector (loop A) on the section, and that the number of outflow vehicles was 71000 counted by the downstream loop detector (loop B). The difference between inflowing traffic and outflowing traffic is accumulated up to -6177. According to vehicle conservation law, one explanation is that there are thousand of vehicles are within the one-kilometer section, which is impossible in reality. The other explanation is that the loop detectors have considerable errors in accumulated counting. For this reason, the curve for accumulated vehicle counts may cross each other as illustrated in Figure 8.3.

In this chapter, travel time is still an important ingredient. Travel time can be measured by means of for example automated vehicle identification (AVI) systems, which identify vehicles at two consecutive locations A and B at time instants t_A and t_B and deduce the *realized* travel time afterwards with $TT_r = t_B - t_A$. AVI systems may employ camera's and license plate detection technology, or may match vehicles through induction footprints, tolling tags or otherwise. It is worth to note that this paper uses individual travel times instead of aggregated or average travel times. In contrast to other traffic information such as traffic flow, density and speed, travel times may be regarded as a kind of causal aggregation of traffic history information over realized travel space. In addition, compared to loop data, travel times have an higher order of accuracy without structural bias. Although, travel times can be derived from time-space speed information, the reverse process is impossible. For this reason, it is quite a challenge to use travel times to estimate the local traffic states.

This chapter proposed a new algorithm to fuse these three ingredients (flow, biased speed and travel times) to achieve reliable and more accurate estimates of traffic density and speeds without using traffic models such as second-order traffic models, but only using the basic relationship between traffic variables as shown in Equation (8.1) and (8.2).

8.2 Methodology

In the first place, we present the assumptions and physical laws that are used in this method. The main assumption is still that traffic is homogeneous and stationary in each time-space cell. Travel time measurement are unbiased and very reliable in contrast to speed measurements. Traffic flow measurements are also very reliable but with very small

errors. These small errors are used to bring a serious defect in using a conservation law (see Chapter 2). The main physical laws are $distance = speed * time$, $density = flow/speed$ and vehicle conservation law.

8.2.1 Framework

The whole fusion algorithm consists of two parts. The first part fuses travel time and speed measures by loop detectors. The second part further fuses the flow measures. In the end, the density and speed over the whole time space region are achieved. (See Figure 8.4)

- In the first part, the vehicle trajectories are reconstructed on a time-space plot by combining individual travel times and the given speed measures. This part of algorithm stems from *PISCIT* algorithm Ou et al. (2008). The individual travel times are obtained from vehicle identification systems e.g. in-car GPS, cameras or other AVI devices. The given speeds are measured by loop detectors which cause measurement bias due to time-mean aggregation. The reconstructed trajectories are able to remove the bias effects to some extent by satisfying the given travel times as constraints.
- In the second part, the traffic density and speed in the time-space cells where the trajectories pass are deduced by simply using these trajectories. Next, the flow information is used to further deduce the density in the other time-space cells by employing *Vehicle Conservation Law*. Assuming that the traffic is homogeneous and stationary in each time-space cell, the traffic speed in the all time-space cells also becomes available by applying $v = q/\rho$ for each cell.

8.2.2 Fusion part one: trajectory reconstruction

This part of the algorithm is able to reconstruct individual trajectories by combining the given travel times and previously observed (estimated) cell-speeds from loop detectors. Many AVI devices such as cameras or in-car GPS can provide the entry point for a vehicle, that is where and when the vehicle enters a road stretch. Also the exit point is given about where and when this vehicle leaves the road stretch. Any line which links the two points could be a trajectory for this vehicle. The algorithm presented below is able to find the most ‘likely’ trajectory with the help of the time-space speed information from loop detectors, even though there is considerable bias in these speed measures. The mechanism behind is quite simple. For a fixed road segment in a road stretch, longer travelled distance, more travel time; higher speed, less travel time;

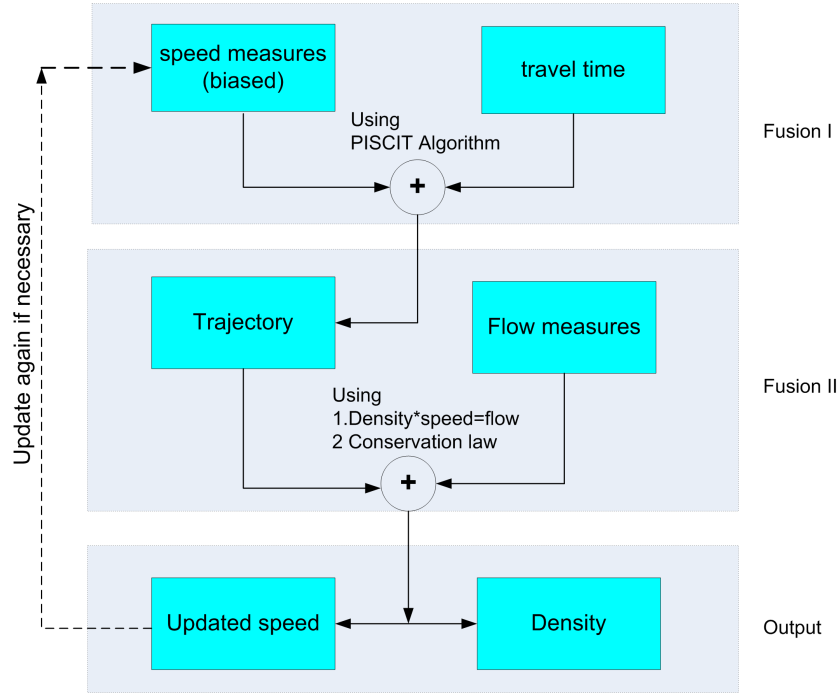


Figure 8.4: Fusion framework

For simple illustration, it is assumed a probe vehicle k entered road segment 1 at reporting time $\hat{t}_1^{(k)}$ and exited segment 6 at the next reporting time $\hat{t}_7^{(k)}$ (Refer to Figure 8.5). This trajectory reconstruction algorithm is made up of the below steps, the first four of which accomplish reconstruction on segment level while the last two on cell level. Table 8.1 lists the important symbols used below.

STEP 1: Get $\hat{t}_j^{(k)}$ and $\hat{t}_{j+1}^{(k)}$ from this previously-estimated trajectory as shown in Graph (a) in Figure 8.5 (The initial trajectory can be assumed to be a straight line)

STEP 2: Based on the given time-space speeds (biased), (re)-calculate the average speed $\bar{v}([\hat{t}_j^{(k)}, \hat{t}_{j+1}^{(k)}], j)$ over segment j during the time between $\hat{t}_j^{(k)}$ and $\hat{t}_{j+1}^{(k)}$.

STEP 3: Update $\hat{t}_j^{(k)}$ and $\hat{t}_{j+1}^{(k)}$ based on the average speed $\bar{v}([\hat{t}_j^{(k)}, \hat{t}_{j+1}^{(k)}], j)$. The updated $\hat{t}_j^{(k)}$ can be obtained under the below restrain equations

$$\hat{t}_j^{(k)} \propto \frac{L_j}{\bar{v}([\hat{t}_j^{(k)}, \hat{t}_{j+1}^{(k)}], j)} \quad (8.3)$$

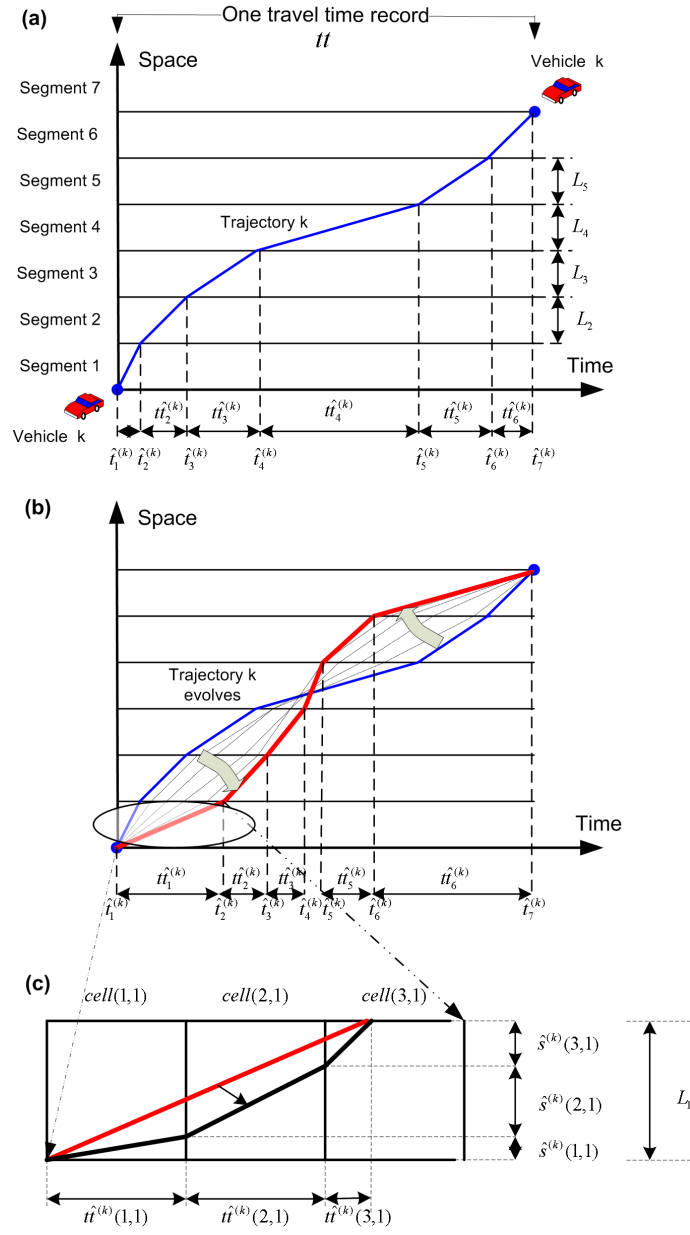


Figure 8.5: Illustration of PISCIT algorithm

$$\sum_j tt_j^{(k)} = tt \quad (8.4)$$

where tt is the given travel time for a vehicle over the whole stretch. After that, update $\hat{t}_j^{(k)}$ based on $tt_j^{(k)}$.

STEP 4: Repeat STEP 2 and STEP 3 until $\hat{t}_i^{(k)}$ and $tt_i^{(k)}$ converge to a specific extent or the difference between the results from two consecutive iterations falls into a given range. (Refer to Graph (b) in Figure 8.5)

STEP 5: Deduce $tt^{(k)}(i, j)$ from $tt_j^{(k)}$, $\hat{t}_j^{(k)}$ and cell size. (Refer to Graph (c) in Figure 8.5)

STEP 6: Deduce $\hat{s}^{(k)}(i, j)$ under the below equations. (Refer to Graph (c) in Figure 8.5)

$$\hat{s}^{(k)}(i, j) \propto tt^{(k)}(i, j) * \hat{v}^-(i, j) / \bar{v}([\hat{t}_j^{(k)}, \hat{t}_{j+1}^{(k)}], j) \quad (8.5)$$

$$\sum_i \hat{s}^{(k)}(i, j) = L_j \quad (8.6)$$

8.2.3 Fusion part two: speed and density reconstruction

In fusion part two, the reconstruction of trajectories returns the traffic speed and density where they pass through, and then traffic density and speed over the whole time space region are deduced by fusing flow information (Refer to Figure 8.6).

Assuming trajectory k passes the cell (i, j) , the measured (estimated) density and speed by the trajectory at this cell can be obtained with the below equations:

$$\hat{v}^{(k)}(i, j) = \frac{\hat{s}^{(k)}(i, j)}{tt^{(k)}(i, j)} \quad (8.7)$$

Assuming the traffic is homogeneous in each time-space cell, it reads

$$\hat{\rho}^{(k)}(i, j) = \frac{\hat{q}^-(i, j)}{\hat{v}^{(k)}(i, j)} \quad (8.8)$$

In order to distinguish the final estimation $\hat{v}(i, j)$, we simply call $\hat{v}^{(k)}(i, j)$ ‘measured’ speed by trajectory k , though it is actually deduced from the reconstructed trajectory. Similarly for density and flow.

For a fixed road segment j , *Vehicle Conservation Law* leads to below equations

$$0 = \rho(i-1, j) - \rho(i, j) + \frac{\Delta t}{L_j} (q(i, j-1) - q(i, j)) \quad (8.9)$$

where $i = 1, 2, 3 \dots$ represents discrete time.

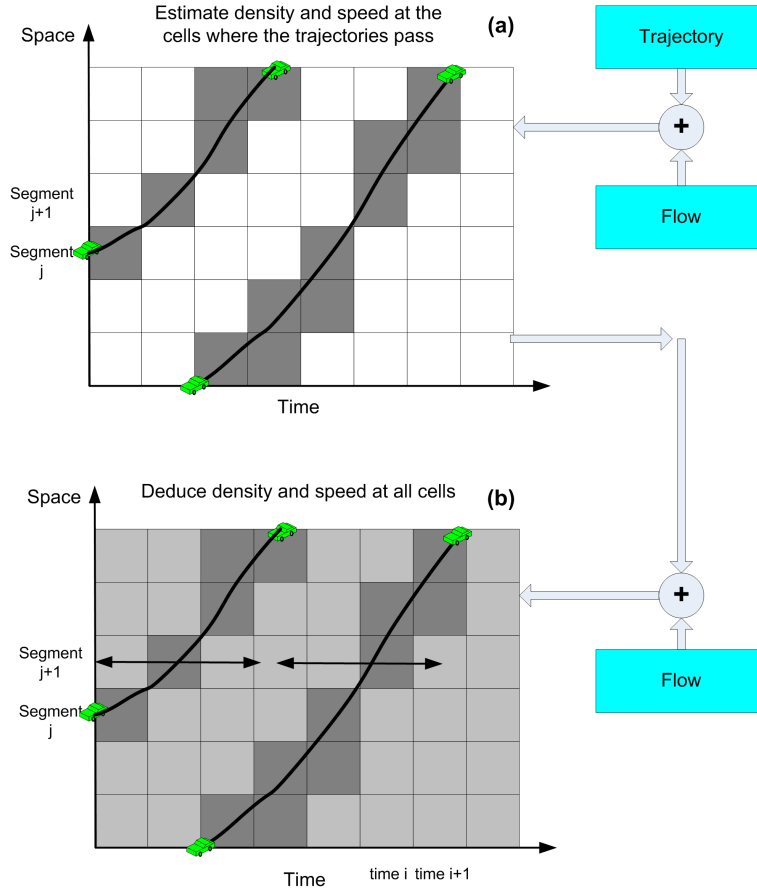


Figure 8.6: Fusion part two

The measured density by trajectories leads to measurement equations for density:

$$\hat{\rho}^{(k)}(i, j) = \rho(i, j) \quad i = 1, 2, 3... \quad (8.10)$$

The measured flow by loop detectors leads to measurement equations for flow:

$$\hat{q}^-(i, j) = q(i, j) \quad i = 1, 2, 3... \quad (8.11)$$

A regression model can be easily established by combining these three sets of formula (8.9), (8.10) and (8.11)

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad (8.12)$$

where $\mathbf{y}$ contains measures 0 , $\hat{\rho}^{(k)}(i, j)$ and $\hat{q}^-(i, j)$ for a fixed j , and $\mathbf{x}$ contains estimated states $\rho(i, j)$ and $q(i, j)$. The optimal estimate of $\mathbf{x}$ in terms of least square given errors

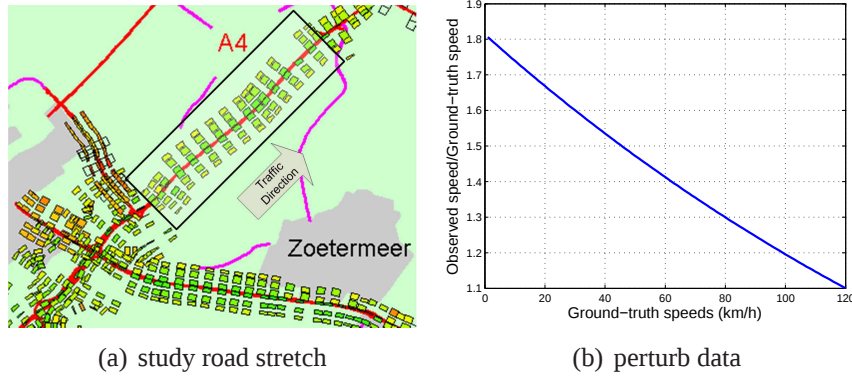


Figure 8.7: Illustration of the study road and how the ground-truth data are perturbed

with the independent and identical distribution is

$$\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{y} \quad (8.13)$$

where $\hat{\mathbf{x}}$ contains the optimal estimation $\hat{\rho}(i, j)$ and $\hat{q}(i, j)$. Further, the speed can be estimated as

$$\hat{v}(i, j) = \frac{\hat{q}(i, j)}{\hat{\rho}(i, j)} \quad (8.14)$$

Now we have density, flow and speed estimates over the whole time-space region.

8.3 Validation

In the first part, the synthetic ‘ground-truth’ data are generated by assuming the real loop data are true, and then the observed data are generated by tampering the ‘ground truth’ data. In the second part, the proposed algorithm is applied on the observed data and returns the estimated data. The performance of this algorithm is shown by comparing the ‘ground-truth’ and estimated results.

8.3.1 Experiment setup & data generation

First of all, a 9.5 kilometer stretch of 3-lane Highway A4 eastbound in Netherlands is considered (Graph (a) in Figure 8.7), where 18 loop detectors are placed spacing around 500 meters and aggregated traffic speed measures and counts every one minute.

- **Ground-truth speed** We assume the loop detectors give the ground-truth speed measures over certain segments. The resulting time-space speed contour plots (Figure 8.8) shows 5 hour traffic condition on this stretch from 6:00 A.M. till 11:00 A.M. on July 8th, 2008, during which congestions onset and dispersed twice.
- **Ground-truth density & flow** The ‘ground-truth’ density and flow are generated by using the ‘ground-truth’ speeds and loop counts as boundary condition. The generated data satisfy *Conservation Law* and the homogeneous condition $\rho v = q$.
- **Observed speeds** The observed speeds in each time-space cell are assumed by tampering the ground-truth speeds with the below assumption:

$$v^o = e^{1.1v^g(0.5-0.5v^g/120)} \quad (8.15)$$

It can lead to nonlinear bias in speed measurements as in real-life. In order to show the performance of our method, this nonlinear bias is made larger than realistic. In this equation, v^o is the observed speed and v^g is the ground-truth speed. With this assumption, the observed speed is 10% higher when ground-truth speed is 120km/h, and 70% higher at the speed of 20km/h. The resulting observed time-space speeds are shown in Graph (a) in Figure 8.9 The relationship between them is shown in Graph (b) in Figure 8.7.

- **Observed flow & density** The observed flow is assumed to almost equal to the ground-truth ones. But still we tamper the ground-truth flow measurement with only 1% relative errors (uniform distribution between $[-1\%, +1\%]$). The purpose is to keep the characteristics of accumulated errors in flow measurements as in real-life world. Due to these errors, we cannot estimate traffic states by only using flow measurements and conservation law. The observed density is actually estimated by using $\rho = q/v$ (Refer to Graph (a) in Figure 8.12).
- **Travel times** The travel times are generated by sampling the ‘ground-truth’ time-space speed plots. There are three virtual cameras placed at the entry, exit and middle of the whole road stretch. It is assumed that 10% of vehicles are captured by the cameras, giving the travel times from milepost 0km to 4.8km and others from 4.8km to 9.5 km.

8.3.2 Results

We use mean absolute relative error (*MARE*) to evaluate the results. The definition of *MARE* is shown in Equation 8.16.

$$MARE = \frac{1}{M * N} \sum_i^M \sum_j^N \frac{|\hat{x}(i, j) - x(i, j)|}{x(i, j)} \quad (8.16)$$

$\hat{x}(i, j)$ represents the estimate and $x(i, j)$ represents the ground-truth quantity. The comparison of the results before and after using the algorithm can be seen in Figure 8.9 and Figure 8.12.

Table 8.2: Comparison of MARE before and after using the proposed algorithm.

Measure type	MARE (before)	MARE (after)
MARE on speeds (global)	33.7%	3.3%
MARE on speeds (<50kmph)	63.4%	3.4%
MARE on density	24.1%	3.46%
MARE on travel times	26.9%	0.74%

- **Before** The observed speeds and density have large errors. MARE for the observed speeds is 33.7% in the given scenario. MARE for observed density is 24.12%.
- **After** After the proposed algorithm is applied to fuse the observed speeds, travel times and flow, the above errors remarkably decrease. MARE for estimated speeds becomes 3.3% and MARE for estimated density becomes 3.46%.

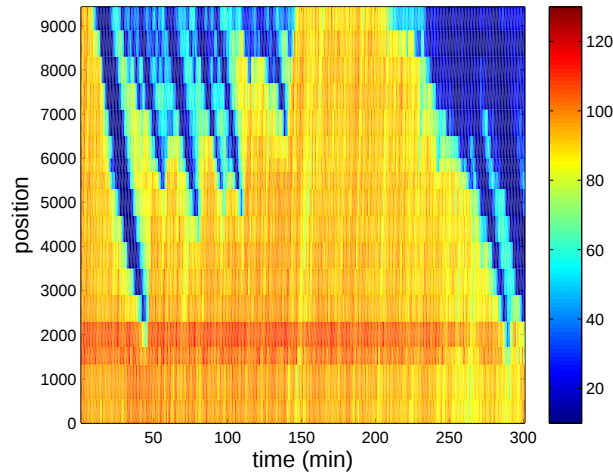
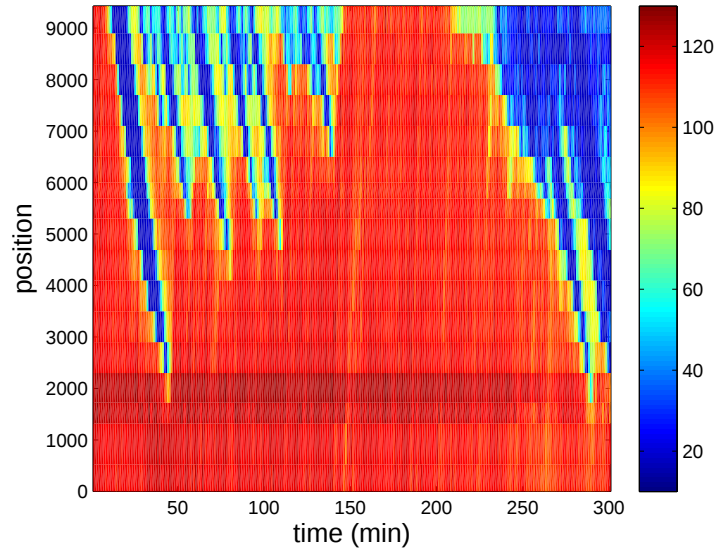
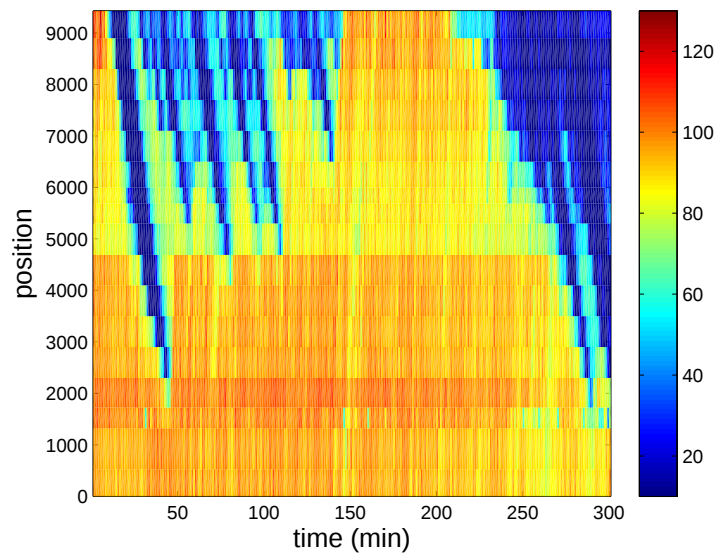


Figure 8.8: Ground-truth time-space speed plot

With time-space speed plots, travel times can be easily derived. Figure 8.14 makes comparison of travel time estimates between before and after using this algorithm. Before



(a) Observed speed



(b) Estimated speed

Figure 8.9: Comparison between observed speeds and estimated speeds after applying the proposed fusion algorithm

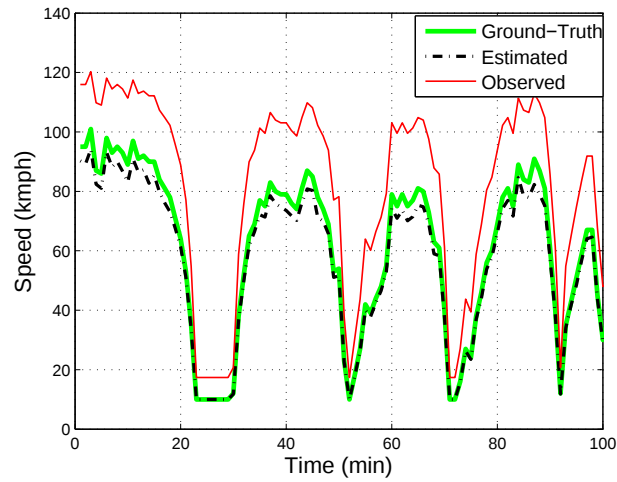


Figure 8.10: Comparison between ground-truth speeds, observed and estimated ones on road-segment 10 (around 5.5km milepost)

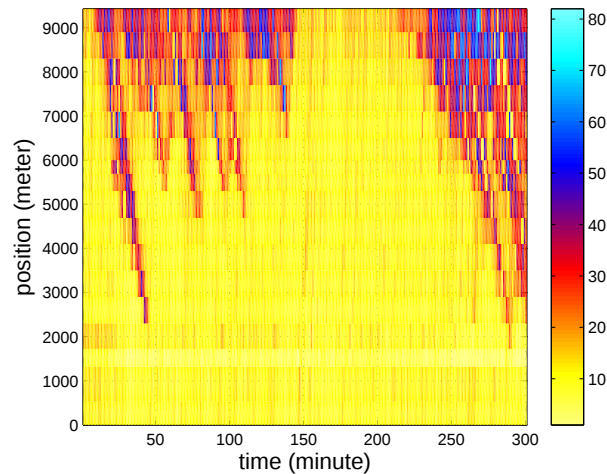
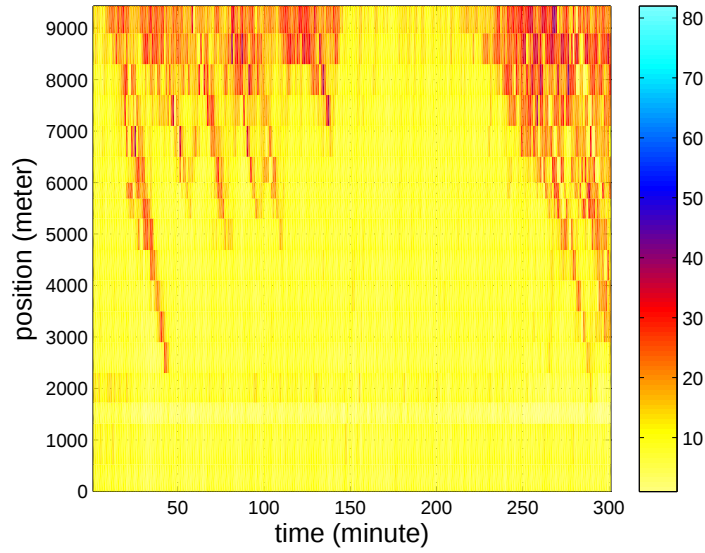
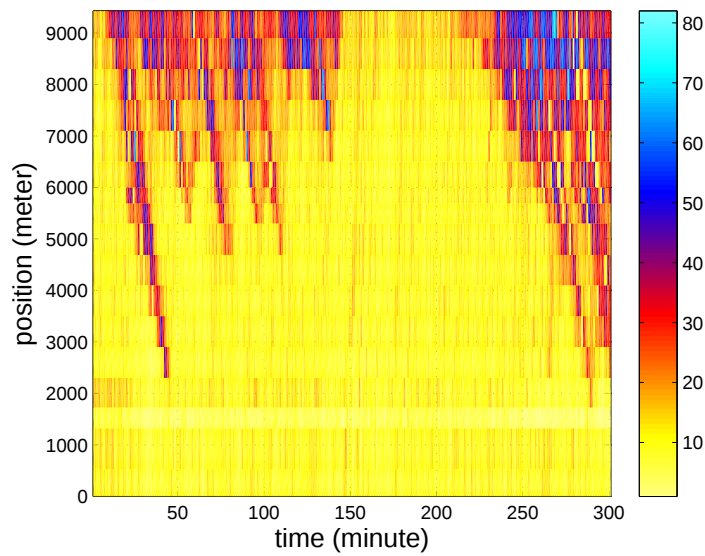


Figure 8.11: Ground-truth time-space density plot

using it, the travel times based on observed speeds have mean absolute error of 230.4 seconds and MARE 26.89%. After using it, the travel times have a much smaller error of 6.6 seconds and MARE 0.74%. In Figure 8.14, the thick green line represent the ground-truth travel time, dark dashed line represents the results after using the algorithm and thin red line represents the travel time estimation based on the observed speeds. The former two lines almost overlap with each other.



(a) Observed density



(b) Estimated density

Figure 8.12: Comparison between observed density and estimated density after applying the proposed fusion algorithm

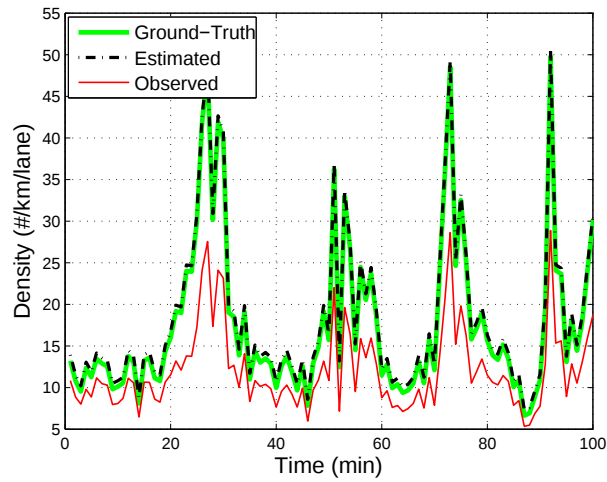


Figure 8.13: Comparison between ground-truth density, observed and estimated ones on road-segment 10 (around 5.5km milepost)

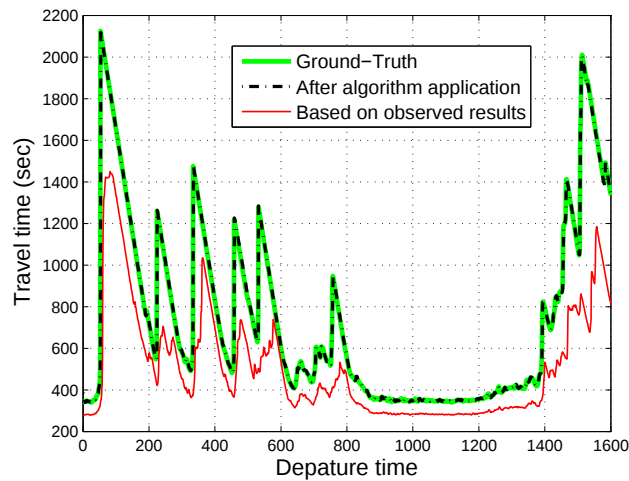


Figure 8.14: Compare travel times between before and after using this algorithm

8.4 Conclusion

The previous chapters 5, 6 and 7 mainly focus on the estimation of traffic speeds by fusing loop speeds and floating car data. But they cannot fuse loop flow and provide traffic density. To tackle this issue, this chapter proposed a new algorithm for fusing *speeds* and *flow* from local detectors with individual *travel times* measured by AVI systems. Travel times

from in-car GPS or cameras can provide average journey speeds over a few road segments but fails to provide the traffic details on each segment. Loop detectors can provide local traffic information, but there are biased errors in speed measures and the error in vehicle counts is accumulated over the time. The proposed algorithm exploits the strength of each type of data and avoids their weakness. In contrast to the often-used traffic fusion techniques presented in Chapter 3, this algorithm needs very few assumptions on traffic behavior but can fuse more types of data.

The validation shows this fusion algorithm can improve the estimation accuracy by up to 10 times. On the basis of synthetic ‘ground-truth’ data, we demonstrated how this algorithm is able to successfully correct strongly biased prior speed measurements. It is able to improve the estimation accuracy up to ten times, e.g decreasing MARE from 33% to 3.3%, decreasing errors in travel time estimation from 230 seconds to 6.6 seconds.

Chapter 9

Synthesis: a data fusion framework

In this thesis, four methods have been proposed to fuse multi-source traffic data. These methods are all based on the data-data consistency approach. For each method, there are separate validations and specific applications. This chapter will discuss issues regarding the applications of these four methods. We will discuss how these four methods are used in different conditions and how they can work with other already-existing methods.

9.1 Synthesis and comparison of data-data consistency methods

PISCIT, TravRes and ITSF all deal with travel times that come from camera data or floating car data. The previous chapters have already given the validations of these methods under almost the same experiment setup and data assumption. 18 loop detectors are installed on the about 10 kilometer road stretch on A4 in the Netherlands. The synthetic ‘observed’ speeds from loop detectors have relative errors of 32% over the whole region and errors of 64% over the region where the ground-truth speeds are lower than 50kmph. The virtually installed cameras are spaced 5 kilometers, which capture 10% vehicles. In PISCIT validation, the added floating cars have polling rate of 60 seconds. In the validations, the travel times and flow are not made to be biased or erroneous.

In this section, the main validation results are put together to compare their performance when different data combinations are used. Table 9.2 shows the result when travel times (camera data and FCD of polling rate 60 seconds) and local speeds are fused by PISCIT. Table 9.3 and 9.4 shows the performance when only travel times are fused by using TravRes. Table 9.5 shows the result when travel times (camera data), local speeds and

Table 9.1: Brief on the proposed algorithms in the book based on data-data consistency approach

Method name	function	input	output
ITSF	Fuse travel times, time-space speeds and flow all together. The output is time-space speed plots and density plots	T-S speeds; travel times; T-S flow	T-S speeds; T-S density
TravRes	Mainly fuse floating car data of low-polling rates. The output is time-space speed plots with high accuracy and little bias.	travel times	T-S speeds
PISCIT	Remove much of the bias in the initial time-space speed plots by using travel times from e.g. cameras. The bias-free time-space speed plots can be achieved after fusion	T-S speeds; travel times	T-S speeds
FlowRes	Fuse low-resolution positioning data with initially measured time-space speeds. The output is time-space speeds	T-S speeds; sampled flow; prior speed distribution	T-S speeds; posterior distribution and error variance

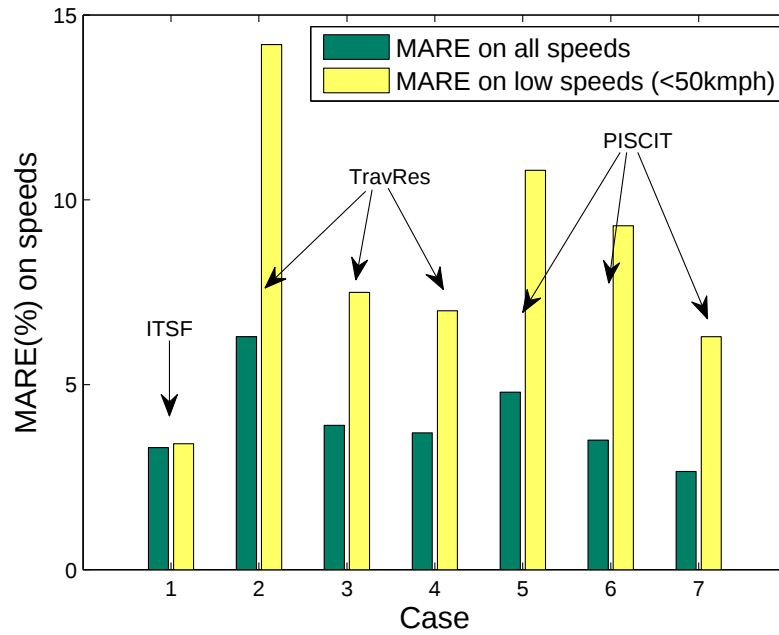


Figure 9.1: MARE on speed estimation under different cases.
 case 1: ITSF with 10% camera travel times + loop speeds + traffic flow
 case 2: TravRes with 2% FCD
 case 3: TravRes with 5% FCD
 case 4: TravRes with 10% FCD
 case 5: PISCIT with 10% camera data + loop speeds
 case 6: PISCIT with 10% camera data + 1% FCD + loop speeds
 case 7: PISCIT with 10% camera data + 5% FCD + loop speeds

flow measured by loop detectors.

Table 9.2: The result from PISICT.

MARE without travel time information is 32.3% (global) and 64.6% (speeds lower 50kmph).

Penetration of FCD added	10%	5%	2.5%	1%	0.5%	0%
MARE on speeds (global)	2.5%	2.65%	2.9%	3.5%	4.2%	4.8%
MARE on low speeds (<50kmph)	5.6%	6.3%	7.7%	9.3%	10.2%	10.8%

By using PISCIT, fusing FCD of penetration 1%, camera data (penetration 10%) and loop speeds leads to 3.5% relative error over the whole space region. The similar result of MARE 3.7% can be achieved by using TravRes to fuse 10% FCD. But for the estimation

*Table 9.3: The result from TravRes.
MARE of speed estimation over the whole time-space region.*

Penetration	Polling rate	MARE with naive	MARE with TravRes
10%	120sec	15.4%	7.4%
10%	60sec	9.1%	3.7%
10%	30sec	5.4%	2.7%
5%	120sec	16.8%	9.5%
5%	60sec	9.8%	3.9%
5%	30sec	5.7%	2.6%
2%	120sec	17.7%	12.3%
2%	60sec	11.4%	6.3%
2%	30sec	7.2%	3.1%

*Table 9.4: The result from TravRes.
MARE of low-speed estimation (speeds are lower than 50kmph).*

Penetration	Polling rate	MARE with naive	MARE with TravRes
10%	120sec	38.9%	11.8%
10%	60sec	23.4%	7.0%
10%	30sec	13.8%	6.5%
5%	120sec	44%	17.3%
5%	60sec	24.9%	7.5%
5%	30sec	15%	6.6%
2%	120sec	50%	29%
2%	60sec	33.5%	14.2%
2%	30sec	20%	7.9%

Table 9.5: The result from ITSF.

Measured type	MARE (before)	MARE (after)
MARE on speeds (global)	33.7%	3.3%
MARE on speeds (<50kmph)	63.4%	3.4%
MARE on density	24.1%	3.46%
MARE on travel times	26.9%	0.74%

of low-speeds (lower than 50kmph), TravRes outperforms PISCIT with the MARE 7.0% v.s. 9.3%. Even with FCD penetration of 5%, TravRes can outperform PISCIT with the MARE 7.5% v.s. 9.3% over the low speed region. When more FCD are used in PISCIT, PISCIT will achieve better results. If the FCD penetration rise to 5% in PISCIT, PISCIT can give MARE of 2.65% (global) and 6.3% (lower speeds) with FCD penetration 5%. The same result can be achieved by TravRes with FCD penetration 10% and polling rate 30 seconds.

Compared with PISCIT and TravRes, ITSF is able to fuse another type of data—traffic flow, which leads to further improvements in results. As seen in Table 9.5, MARE in low-speed region is only 3.4%, while PISCIT can only achieve MARE 5.6% in low-speed region by fusing FCD penetration 10%, camera data (10% penetration) and local speeds.

FlowRes is able to fuse low-resolution positioning data from wireless networks. When it works with other methods such as PISCIT, TravRes or ITSF, we have to deal with inconsistency issue. In wireless networks, base station may cover a road segment with length e.g. 3 kilometers. With TP data, FlowRes can only estimate the average speeds on this 3-kilometer road segment. But in PISCIT or other methods, a road stretch may be divided into segments with length about 500 meters. As a result, the road segment defined in different methods have different space boundary, which leads to space inconsistency. In addition to space inconsistency, there is time inconsistency. For example, the reporting time or rate from wireless networks may not be synchronized with the ones from loop detectors. As a result, we need to upgrade the low-resolution time-space cell to high-resolution time-space cell. Figure 9.2 show one time-space cell defined in FlowRes may partially or completely cover several time-space cells defined in other methods.

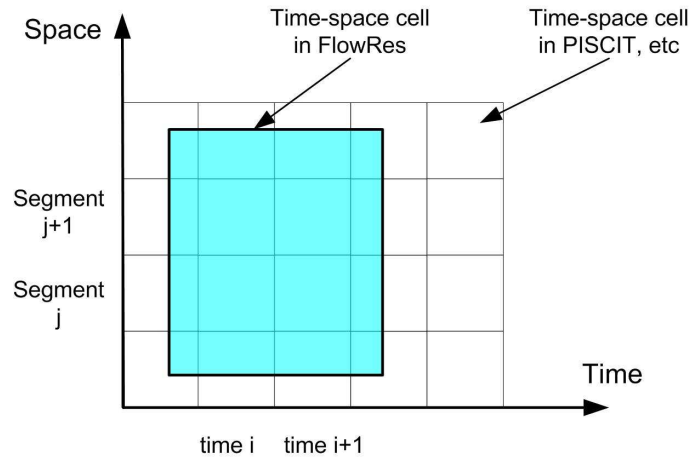


Figure 9.2: Time-space consistency when FlowRes works with other methods

This issue can be easily dealt with by establishing linear equations and then giving best

estimation with regression techniques.

$$\sum w(i, j)v(i, j) = v^{TP} + \epsilon \quad (9.1)$$

and equations

$$v(i, j) = v^o(i, j) + e(i, j) \quad (9.2)$$

for (i, j) in the shadowed region. v^{TP} is the average speed over the shadowed time-space region, which is given by FlowRes. $v^o(i, j)$ is the estimated speeds in the time-space region (i, j) , which is given by PISCIT, TravRes, etc. Since the shadowed region may only partially cover some region (i, j) , we append $w(i, j)$ as weight. Particularly, $w(i, j)$ satisfies $\sum w(i, j) = 1$ for (i, j) in the shadowed region. ϵ is the error involved in v^{TP} , and $e(i, j)$ represents the error in speed estimation from other methods. The best estimation $\hat{v}(i, j)$ can be achieved by using linear regression techniques.

9.2 Data fusion framework

As presented in Chapter 3, there are some other fusion methods that have been proposed, such as ‘EKF plus traffic model’, ‘Treiber filter plus traffic model’, etc. We don’t necessarily intend to replace these methods by our proposed methods. On the contrary, the proposed algorithms or methods can be well used to enrich the already-existing methods and work along with these methods.

One of important features in our methods is that they are able to provide much less-biased time-space speed and density plots. These plots can be used to better calibrate parameters (e.g. critical speed) in traffic models such as the fundamental diagram. These models and parameters play a critical role in e.g. Extended Kalman filters or Treiber filter. Recalling the attributes of Kalman filters, the noises in the equations are not supposed to have structural bias but only random errors. Therefore, the output from PISCIT or TravRes can be taken as ‘bias-free measurements’ for Kalman-filter-based methods. Furthermore, the method FlowRes provides not only speed estimation but also the error variance in this speed estimation. The error estimation is equally important as state estimation, which helps to determine the confidence on the estimation and can be taken as input for traffic prediction.

These ‘model-data consistency’ approaches can also contribute to the proposed methods

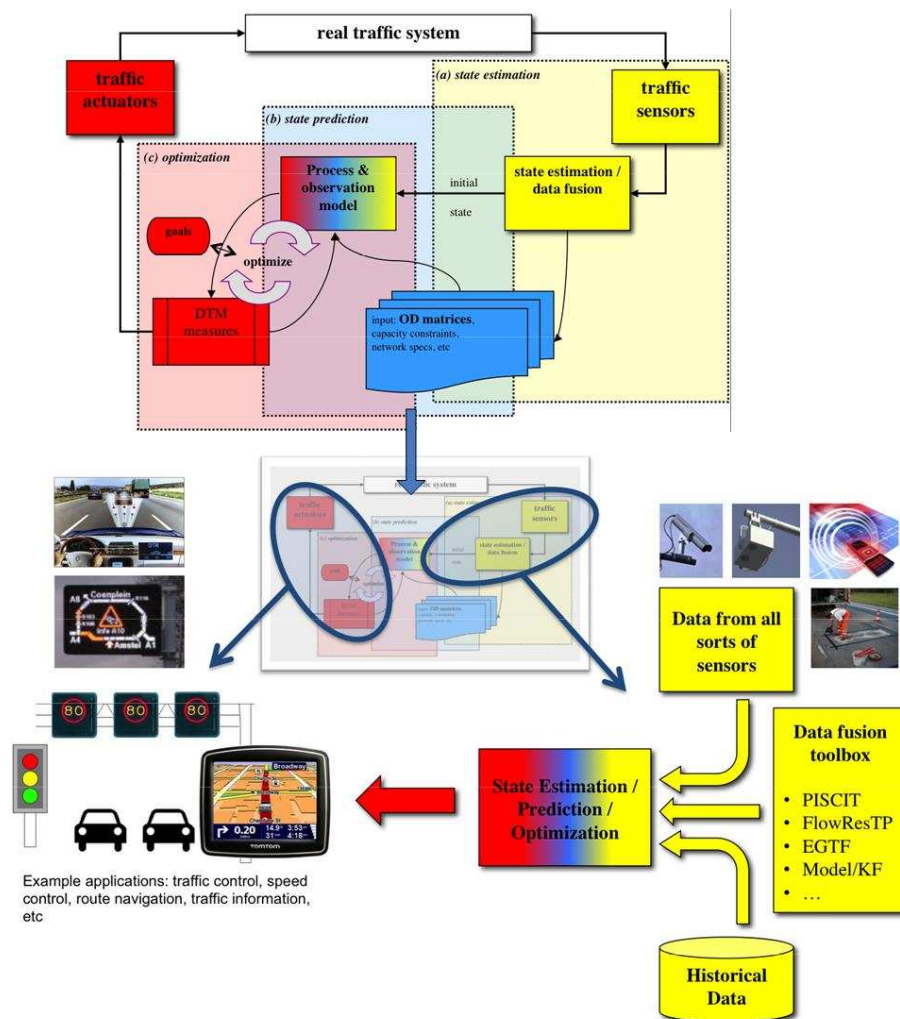


Figure 9.3: Application of fusion tools in the traffic system

in this thesis. The travel time data from cameras or floating car may not have enough time-space coverage for traffic estimation. Also the loop detectors may not be installed for about 500 meters as in South-Holland. In many countries, loop detectors on freeways may be sparsely installed. Due to lack of enough coverage of loop detectors, floating cars and cameras, the methods like PISCIT, TravRes or ITSF can not work well or even does not work at all. As a result, the reconstructed time-space plots for the proposed methods will have a lot of blank regions. In order to fill in these blank regions, we can use many model-based methods like Kalman-filter-based methods or Treiber-filter-based methods. For example, we can use Kalman-filter-based methods to reconstruct an initial time-space speed plots when loop detectors are sparsely installed. Then PISCIT method is used to remove the bias in the initial estimation. Following that, these less biased results can be re-used by e.g. Kalman-filter-based methods to re-estimate the traffic states.

In sum, all the methods are not exclusive to one another. On the contrary, they need to collaborate to achieve better estimation results. All these methods can make a tool box of traffic data fusion. As shown in Figure 9.3, they are able to process the current data as well historical data from different sensors. This tool box plays a very critical role in traffic state estimation, prediction and optimization.

9.3 Summary

This chapter mainly compare the applications of the proposed methods and further applications combined with already-existing methods. We only give a rough guideline to the applications of these methods in the different conditions and how they can work together for more advanced application. Although no real-life application has been given, the broad application potential of the proposed methods can be clearly seen.

Chapter 10

Conclusions and recommendations

Reliable and accurate estimation of traffic state variables from the available traffic data plays an important role in traffic management practice and science. State estimation on the basis of multiple data sources is a challenging task, since many of the variables of interest, such as space mean speed or traffic density, cannot be observed directly and must be deduced from the data which are available. Moreover, the available data from various sources (loop detectors, floating car data or automated vehicle identification systems) differ largely in terms of quality, reliability, availability and even spatiotemporal semantics, which renders state estimation on the basis of multiple data sources even more problematic.

In this thesis a new traffic data fusion paradigm based on data-data consistency is proposed and several example data fusion algorithms in line with this new paradigm are presented. In this final chapter we will highlight the main conclusions and provide recommendations for practical application as well as directions for future research.

10.1 Conclusions

10.1.1 Main conclusions

The main conclusion of this thesis is that the proposed ‘data-data consistency’ (DDC) paradigm works, and that it provides a parsimonious and robust framework for fusing data from different sources, even if these data have fundamentally different spatiotemporal semantics. Methods based on the DDC paradigm are characterized by the fact that one data source is used to constrain or correct the state variables deduced from a second data

source using as few additional assumptions (parameters) as possible. This new DDC paradigm for traffic state estimation and data fusion has two major advantages over classic data fusion paradigms.

In the first place we can conclude that DDC algorithms require making (much) less assumptions than classic state estimation and data fusion approaches such as recursive filtering methods (Kalman filtering, particle filtering, nudging techniques). These recursive approaches essentially use a model-data consistency paradigm, in which process and observation models are required to specify the relationship between state variables and between state variables and observations respectively. Both process and observation models are typically parameterized and make many implicit and explicit assumptions about the data. For example, the speed-density relationship used in recursive data assimilation techniques is a coarse and noisy observation model and largely underdetermined in free flow conditions. The DDC methods developed in this thesis use only simple parameter-free physical laws (e.g. travel time = distance / speed and flow = density x speed), and a minimum number of assumptions with respect to the data assimilation methods.

Secondly, the DDC approaches solve the spatio-temporal alignment problem of recursive techniques. This problem occurs when the available data sources have incompatible spatiotemporal semantics. The best example of semantically incompatible data involves travel times (from AVI systems or partial vehicle trajectories) and spot mean speeds (from local detectors). Classic data fusion approaches based on recursive filtering require observation equations to link the first to the latter, which in this case is impossible, since the relationship between travel times and spot speeds is underdetermined. The various algorithms presented in this thesis solve the problem by using the first data source (travel times) as a constraint to re-estimate the second (spot speeds). This effectively results in much less-biased and accurate spatiotemporal speeds and - as a direct consequence - also in improved travel time records without resorting to complicated observation models. Note that all DDC methods are designed to reconstruct average speeds, flows and/or densities on small spatiotemporal areas of length ΔX and period size ΔT , in the sense of Edie Definition (Edie (1965))

10.1.2 Conclusions related to the presented algorithms

In this thesis four DDC algorithms were designed and validated. The methods developed are

- PISCIT: a method for traffic speed reconstruction based on prior speeds and individual travel time

- TravRes: a method for traffic speed reconstruction based on travel time of low-polling rates.
- FlowRes: a method for traffic speed re-estimation based on low-resolution positioning FCD and prior speeds
- ITSF: a method for speed and density reconstruction based on traffic flow, speed and individual travel times.

Based on the research in this thesis, the following conclusions can be drawn. The PISCIT method fuses (possibly biased) spot mean speeds with individual travel time records and uses the second to constrain and correct the first. From our results we can conclude that it is effective in removing much of the bias in local speed measurements. The result is a much less-biased spatiotemporal map of average speeds. The TravRes method fuses travel time (or average speed) records from a percentage p probe vehicles which may provide these records at arbitrary polling rates T . The method also allows incorporation of spot speeds, which is used as prior information. Again the data are used as constraints for additional data such that no assumptions need to be made on the actual vehicle trajectories (e.g. that vehicles drive with constant speeds). From our results we conclude that the method results in much-less biased estimates of spatiotemporal mean speeds for penetration rates above 10%, in which polling rates may vary from 30 seconds to 120 seconds. Based on synthetic data, it is found that classic methods may lead to errors of 30% in speed estimations and error of 50% in lower speed (50kmph) estimation. TravRes doubles or even triples the estimation accuracy. The FlowRes method also provides a solution for optimally using data from probe vehicles but takes a slightly different approach and uses different data. FlowRes is able to fuse low-resolution positioning data (e.g. from mobile phones) to reconstruct spatiotemporal mean speeds. FlowRes takes low-resolution positioning data as sampled flow, and fuses the sampled flow with local speeds by using a Bayesian update rule. The validation shows that the method can decrease the relative error in speed estimation from 41% to 10.5%. The ITSF method finally builds on the same ideas as the PISCIT method, but further takes traffic flow as input. As a result, it can provide more accurate speed estimations than PISCIT, particularly for low-speed estimation. Additionally, ITSF provides a means to also estimate traffic density. As far as we know, this is the only viable reliable alternative for density estimation aside from the much more elaborate recursive filtering approaches (e.g. Kalman filtering, particle filtering) discussed earlier

10.2 Implications and recommendations

The methods based on the data data consistency paradigm are scientifically new and provide practical solutions for problems which cannot be addressed by existing methods. Below we list a number of implications and recommendations for both science and practice

- Since the data-data consistency approaches only make very few assumptions, we recommend scientists to use this approach in the early stages traffic theory and model development in case ground-truth data (in the Edie-sense) are not available.
- We also strongly recommend scientists from various disciplines (traffic theory, but also control and systems engineering) to consider data-data consistency methods as a viable alternative to well-known traditional approaches (e.g. Kalman-filter-based approach). We believe that in many cases it is possible to find appropriate consistency criteria and to find a data assimilation method to achieve this consistency. Additionally, one may use the results of such a DDC method in traditional recursive filtering.
- Although the problem of time averaging local traffic data has been acknowledged widely, still quite a few traffic data collection systems use time averaging, leading to biased speeds and density estimates. We strongly recommend methods based on data-data consistency to remove this bias, instead of using again parameterized methods for this purpose.
- The methods developed in this thesis may be used in combination with other methods. In this sense, data data consistency methods constitute a toolbox of data fusion algorithms, which can be readily used in both science and practice
- The four example algorithms (fusion tools) developed in this thesis provide a good starting point, in that they cover data fusion of the most commonly available data in practice to date. These four algorithms are effective in fusing low-resolution data, e.g. travel times, FCD of low polling rates and low-resolution positioning data.
 - When a large amount of local speeds and individual travel time records are available, the PISCIT method can be used to improve the local speed estimation.
 - In case partial probe vehicle records are available, the TravRes method is recommended.
 - When additionally also traffic flow information is available beside local speed and travel time, the ITSF method is recommended.

- When a large amount of low-resolution positioning data from wireless networks become available, we can use the FlowRes method.

10.3 Future research

In this final section we will provide some pointers for future research. First of all, more research is needed to assess the impact of errors in the available data. For example, this thesis assumes that all travel times are correct. In real-life, individual travel times may be error-free. Aside from measurement inaccuracies, the main cause of errors (particularly from AVI systems) relate to vehicles traveling over different paths, or making unobserved stops. Further research into automated methods for detecting and resolving these errors is needed. Similarly, more research is needed into resolving errors in traffic flows

- Impact of error in traffic flow. In ITSF method, we also use very reliable traffic flow data, in which relative errors are smaller than 1%. Indeed, loop detectors can give quite reliable flow measurement, but in some time-space region e.g. near ramps, the measurement may not be reliable. The future research should also consider the impact of error in flow measurements.
- Technology realization in FlowRes. In order to apply FlowRes, we have to trace floating car in cell level for every one minute. Also we have to distinguish vehicles from pedestrian, and we have to make sure on which road the vehicle is. All these requirements must be met before FlowRes can be used. In order to achieve this, there are still some techniques and equipments that need to be developed. One of solution is to invent a in-car dock for mobile phones. With the support of this dock, the mobile phone can send a beep signal to a base station every one minute.
- Convergence in trajectory reconstruction. PISCIT, TravRes and ITSF all involve the trajectory reconstruction algorithm. However, we cannot theoretically prove the trajectory will converge in this iteration algorithm. In other words, the convergence of trajectory cannot be guaranteed. The research on convergence turns to be a very important topic when data-data consistency approach is used.
- Design of optimal data composition. This thesis has showed that fusing more types of data normally can lead to better estimation result. Each type of data can partially contribute to the state estimation. Meanwhile there are costs for different types of data collection. Therefore, considering the trade-off between estimation accuracy and the cost, we should try to find the optimal data composition. That is a very important topic in practical applications.

- Design of fusion framework. Data-data consistency approach is not a replacement for other approaches or in conflict with them. The proposed methods based on this approach piece together different types of data in a simple way, while the previous methods normally use traffic models. e.g. first-order and second-order model. All these methods can be combined together and form a fusion framework which can intelligently choose appropriate methods to accomplish data fusion under different scenarios.
- Network application. This thesis mainly focuses on traffic state estimation on a single route. When it comes to the network-wide state estimation, the situation becomes more complex. For example, travel times from cameras may refer to several routes in this network. Therefore we have to infer which route a vehicle travel before the proposed methods can be used. Also route inference can be achieved by fusing data from different sources.

Bibliography

- Chen, T., Morris, J., and Martin, E. (2004). Particle filters for the estimation of a state space model. In *Proceedings of European Symposium on Computer Aided Process Engineering*.
- Cheng, P., Qiu, Z., and Ran, B. (2006). Particle filter based traffic state estimation using cell phone network data. In *Proceedings of the IEEE ITSC 2006*.
- Chu, L., Oh, J., and Recker, W. (2005). Adaptive kalman filter based freeway travel time estimation. In *84th TRB Annual Meeting, Washington*.
- Claudel, C. G. and Bayen, A. M. (2008). Guaranteed bounds for traffic flow parameters estimation using mixed lagrangian-eulerian sensing. In *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing, Allerton, IL*.
- Claudel, C. G., Hofleitner, A., Mignerey, N. D., and Bayen, A. M. (2009). Guaranteed bounds on highway travel times using probe and fixed data. In *Proceedings of the Transportation Research Board (TRB), 2009 Washington, DC, January 10-14, 2009*.
- Coifman, B. (2002). Estimating travel times and vehicle trajectories on freeways using dual loop detectors. *Transportation Research A*, 36:351–364.
- Cremer, M. and Papageorgiou, M. (1981). Parameter identification for a traffic flow model. *Automatica*, 17:837–843.
- Dailey, D. J., Harn, P., and Lin, P.-J. (1996). Its data fusion. Technical Report Research Project T9903, Task 9, Washington State Transportation Commission, Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration.
- Edie, L. (1965). Traffic stream measurements and definitions. In *Proceedings of the 2nd International Symposium on the Theory of Traffic Flow*.
- Edie, L. C. (1963). Discussion on traffic stream measurements and definitions. In *Proc. 2nd Int. Symp. Theory Traffic Flow*, pages 139–154.

- Gazis, D. and Knapp, C. (1971). On-line estimation of traffic densities from time-series of flow and speed. *Transportation Science*, 5:283–301.
- Ghosh, D. and Knapp, C. H. (1978). Estimation of traffic variables using a linear model of traffic flow. *Transportation Research*, 12(6):395 – 402. ISSN 0041-1647.
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEEE Proceedings on Radar and Signal Processing*.
- Hegyi, A., Girimonte, D., Babuska, R., and Schutter, B. (2006). A comparison of filter configurations for freeway traffic state estimation. In *Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference (ITSC 2006)*.
- Herman, R. (1984). Characterizing traffic conditions in urban areas. *Transportation Science*, 21:101–140.
- Herrera, J. C. and Bayen, A. M. (2007). Traffic flow reconstruction using mobile sensors and loop detector data. In *87th TRB Annual Meeting*.
- Ishikawa, Y., Awaji, T., and Akimoto, K. (1996). Succesive correction of the mean sea surface height by the simultaneous assimilation of drifting buoy and altimetric data. *Journal of Physical Oceanography*, 26:23812397.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–48.
- Kesting, A. and Treiber, M. (2008). Calculating travel times from reconstructed spatio-temporal traffic data. In *Proceedings of the 4th International Symposium Networks for Mobility. Stuttgart.*
- Kikuchi, S., Miljkovic, D., and van Zuylen, H. (2000). Examination of methods that adjust observed traffic volumes on a network. *Transport Research Record*, 1717:109–119.
- Knoop, V., Hoogendoorn, S. P., and van Zuylen, H. (2007). Empirical differences between time mean speed and space mean speed. In *Traffic and Granular Flow 07*.
- Kohan, R. and Bortoff, S. (1998). An observer for highway traffic systems. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference*.
- Kurkjian, A., Gershwin, S. B., Houpt, P. K., , Willsky, A. S., Chow, E. Y., and Greene, C. S. (1980). Estimation of roadway traffic density on freeways using presence detector data. *Transportation Science*, 14:232–261.

- Laird, D. (1996). Emerging issue in the use of gps for travel time data collection. In *National Traffic Data Acquisition Conference*.
- Lindveld, C. and Thijs, R. (1999). On-line travel time estimation using inductive loop data: The effect of instrumentation peculiarities on travel time estimation quality. In *Proceedings of the 6th ITS World Congress*.
- Linn, R. J. and Hall, D. L. (1991). A survey of multi-sensor data fusion systems. In *Proceedings of the SPIE - The International Society for Optical Engineering*, pages 13–29. Orlando, Florida.
- Medhi, J. (2002). *Stochastic Processes*. New Age International(P) Ltd.
- Meier, J. and Wehlan, H. (2001). Section-wise modeling of traffic flow and its application intraffic state estimation. In *Proc. 2001 IEEE Conference on ITS, Oakland, CA*.
- Michalopoulos, P. and Hourdakis, J. (2001). Review of non-intrusive advanced sensor devices for advanced traffic management systems and recent advances in video detection. *Proceedings of the Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering*, 215(14):345–355.
- Mihaylova, L., Boel, R., and Hegyi, A. (2007). Freeway traffic estimation within particle filtering framework. *Automatica*, 43(2):290 – 300. ISSN 0005-1098.
- Nahi, N. E. and Trivedi, A. N. (1973). Recursive estimation of traffic variables: Section density and average speed. *Transportation Science*, 7:269–286.
- Ni, D. (2007). Determining traffic-flow characteristics by definition for application in its. *IEEE Transactions on Intelligent Transportation Systems*, 8.
- Ni, D. and Wang, H. (2008). Trajectory reconstruction for travel time estimation. *Journal of Intelligent Transportation Systems*, 12(3):113–125.
- Ou, Q., Van Lint, J., and Hoogendoorn, S. (2008). Piecewise inverse speed correction by using individual travel times. *Transportation Research Record: Journal of the Transportation Research Board*, 2049(-1):92–102. 10.3141/2049-11.
- Piella, G. (2003). A general framework for multiresolution image fusion: from pixels to regions. *Information Fusion*, 4(4):259–280.
- Roos, T., Myllymaki, P., and Tirri, H. (2002). A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing*, 1:59–69.
- Simon, D. (2006). *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. John Wiley & Sons, New York.

- Smith, B., Pack, M., Lovell, D., and Sermons, M. (2001). Transportation management application of anonymous mobile call sampling. In *in the 80th Annual Meeting, Transportation Research Board*.
- Smith, M. (2007). Investigation of the performance of wireless location technology-based traffic monitoring system. *Transportation Engineering*, 113:157–165.
- Speyer, J. L. and Chung, W. H. (2008). *Stochastic processes, estimation, and control*. Philadelphia : Society for Industrial and Applied Mathematics.
- Stipdonk, H., Toorenborg, J., and Postema, M. (2008). Phase diagram distortion from traffic parameter averaging. In *European Transport Conference (ETC)*.
- Szeto, M. W. and Gazis, D. C. (1972). Application of kalman filtering to the surveillance and control of traffic systems. *TRANSPORTATION SCIENCE*, 6:419–439.
- Thiessenhusen, K., Schafer, R., and Lang, T. (2005). Traffic data from cell phones: A comparison with loops and probe vehicle data. Technical report, Institute of Transport Research, German Aerospace Center, Berlin.
- Treiber, M. and Helbing, D. (2002). Reconstructing the spatio-temporal traffic dynamics from stationary detector data. *Cooper@tive Tr@nsport@tion Dyn@mics*, 1:3.1–3.24.
- Turner, S., Benz, R., and Holdener, D. (1998). *Travel Time Data Collection HandBook*. Federal Highway Administration.
- Van Lint, J. and Hoogendoorn, S. P. (2007). The technical and economic benefits of data fusion for real-time monitoring of freeway traffic. In *World Congress of Transportation Research*. WCTRS, Berkely, CA, USA.
- Van Lint, J., Hoogendoorn, S. P., and van Zuylen, H. J. (2002). Freeway travel time prediction with state-space neural networks. In *the 81st Annual Meeting of Transportation Research Board*.
- Van Lint, J. W. C. (2004). *Reliable travel time prediction for freeways*. TRAIL Thesis series. TUD Technische Universiteit Delft, Delft.
- Van Lint, J. W. C. (2006). Reliable real-time framework for short-term freeway travel time prediction. *Journal of transportation engineering-asce*, 132(12):921–932.
- Van Lint, J. W. C. (2010). Empirical evaluation of new robust travel time estimation algorithms. In *Compendium of papers TRB 89th annual meeting, 2010 Washington DC*.

- Van Lint, J. W. C. and Hoogendoorn, S. P. (2009). A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways. *Computer Aided Civil and Infrastructure Engineering*, accepted for publication.
- Van Lint, J. W. C., Hoogendoorn, S. P., and Van Zuylen, H. J. (2005). Accurate travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5-6):347–369.
- Van Lint, J. W. C. and Van der Zijpp, N. J. (2003). Improving a travel time estimation algorithm by using dual loop detectors. *Transportation Research Record*, 1855:41–48.
- Varshney, P. K. (1997). Multisensor data fusion. *Electronics and Communications Engineering Journal*, (December):245–253.
- Wang, Y. and Papageorgiou, M. (2005). Real-time freeway traffic state estimation based on extended kalman filter: a general approach. *Transportation Research Part B: Methodological*, 39(2):141 – 167. ISSN 0191-2615.
- Wang, Y., Papageorgiou, M., and Messmer, A. (2006). A real time freeway network traffic surveillance tool. *IEEE Transactions on control systems technology*, 14(1).
- Wang, Y., Papageorgiou, M., and Messmer, A. (2008). Real-time freeway traffic state estimation based on extended kalman filter: Adaptive capabilities and real data testing. *Transportation Research Part A: Policy and Practice*, 42(10):1340 – 1358. ISSN 0965-8564.
- Wang, Y., Papageorgiou, M., Messmer, A., Coppola, P., Tzimitsi, A., and Nuzzolo, A. (2009). An adaptive freeway traffic state estimator. *Automatica*, 45(1):10 – 24. ISSN 0005-1098.
- Westerman, M. (1995). *Real-Time Traffic Data Collection for Transportation Telematics*. Delft University Press, Delft, The Netherlands.
- Willsky, A., Chow, E., Gershwin, S., Greene, C., Houpt, P., and Kurkjian, A. (1980). Dynamic model-based techniques for the detection of incidents on freeways. *IEEE Transactions on Automatic Control*, 25:347–360.
- Xiong, N. and Svensson, P. (2002). Multi-sensor management for information fusion: issues and approaches. *Information Fusion*, 3(2):163–186.
- Yager, R. R. (2004). A framework for multi-source data fusion. *Information Sciences*, 163(1-3):175–200.
- Zhao, Y. (2000). Mobile phone location determination and its impact on intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 1:55–62.

Summary

Reliable and accurate estimation of traffic states play an important role in traffic management and traffic theory development, which therefore has significant social and scientific relevance. Data for traffic estimation normally from different sources and have different types, characteristics, etc, so data fusion techniques are used. Traffic state estimation involves fusion of data from different sources. Therefore, data fusion techniques are used.

The subject of this thesis is about traffic data fusion and the main objective is to propose more efficient approach and algorithms to accomplish traffic data fusion.

Different types of data have different characteristics and accuracy, which causes challenges. Let us take data from loop detectors as an example. Loop detectors can provide the speed measures only at certain points on a road (local speeds). The speed measures from loop detectors have structural deviation from the ground-truth speeds based on Edie's definition. The deviation is relatively bigger when the speed is lower. This deviation can lead to the 100% error in density estimation. Similarly, some other types of data, e.g. floating car data, camera data, etc, have their particular characteristics. It is a challenge to fuse all kinds of data with different characteristics, semantics, resolution, accuracy and reliability.

Although the previous methods have already solved quite a few traffic data fusion issues, yet there are quite a few challenges left. For example, due to spatio-temporal alignment problem, Kalman filter, the most commonly used assimilation techniques for data fusion can not be well used to fuse travel times and local data. Majority of these methods need model calibration that is made through biased data e.g. biased loop speeds, so these methods are not effective in removing the structural bias in data. In sum, previous data fusion techniques normally involves quite a few assumptions, but they may not fuse many types of data or give reliable results.

In order to fuse more types of data and give more accurate results, we propose a new approach. This approach is called 'Data-Data Consistency' Approach. It still needs traffic models, but these models are simply based on some basic physical laws and very few

assumptions.

Based on this data-data consistency paradigm, we develop four methods for traffic data fusion. The first proposed algorithm is called PISCIT which is able to fuse traffic speeds from local detectors such as inductive loops with individual travel times measured by AVI systems. The second is called TravRes which is able to accurately reconstruct high-resolution time-space speeds from floating car data (FCD). It achieves this by iteratively reconstructing the (unobserved) probe vehicle trajectories between polling time instants, until the resulting time-space speed map is consistent (enough) with all probe vehicle reports. The above-mentioned two algorithms are concerned with the low-resolution travel time data (low polling rates). The third is called FlowRes. It deals with another type of data, data which may not only have low time-resolution but also have quite low position-resolution. Such data cannot pinpoint the accurate positions of vehicles but can only give some location-specific information when and where the vehicles are located at the segment or cell level. This algorithm corrects strongly biased prior speed measurements and reduces the impact of random errors. It can be easily extended to fit in network-wide traffic speed estimation. The fourth algorithm is called ITSF, which is able to fuse traffic flow, local speeds and travel times all together. It uses extra data source: traffic flow. As a result, more accurate and reliable estimation is achieved compared to the first two algorithms.

Rather than these four methods, more methods and algorithms can be developed by following data-data consistency paradigm. Furthermore, we propose a data fusion framework, in which all proposed methods can work with other already-existing methods so that better estimation can be achieved.

About the Author

Qing Ou was born in Zunyi, China on June 30th, 1980. From 1999 to 2006, he studied in information school at University of Science and Technology of China. He mainly studied computer science and system control during this period and got BSc and MSc degree respectively in 2003 and 2006. He temporarily worked as an engineer in *Kodak* company to develop a medical image system, and worked in *Freeway* company to develop in-car VCD players from 2003 to 2005. He also took a research role in both *Nonlinear Center* and *Lab of Networking and Control* from 2003 to 2006.

In September 2006, Qing Ou moved to the Netherlands to start his PhD at the Department of Transport and Planning, Delft University of Technology and TRAIL Research School. His research project is Traffic Data Fusion, which is part of ICIS projects. ICIS is short for Interactive Collaborative Information System, one of nice ICT research projects funded by the BSIK program of the Dutch government. The major objective of this data fusion project is to fuse heterogeneous traffic data from different sensors into consistent, accurate and reliable traffic information. This research, conducted under the supervision of Professor Serge Hoogendoorn and Dr. Hans van Lint, resulted in a new traffic data fusion approach, followed by a number of publications in proceedings of several international conferences, for instance TRR, TRB, etc. He finished his PhD project in September in 2010 and finish his defense in May 2011.

TRAIL Thesis Series

A series of The Netherlands TRAIL Research School for theses on transport, infrastructure and logistics.

Qing, O., *using Heterogeneous Traffic Data: Parsimonious Approaches using Data-Data Consistency*, T2011/5, May 2011, TRAIL Thesis Series, the Netherlands

Walta, L., *Getting ADAS on the Road: actors' interactions in Advanced Driver Assistance Systems deployment*, T2011/4, April 2011, TRAIL Thesis Series, the Netherlands

Lin, S., *Efficient Model Predictive Control for Large-Scale Urban Traffic Networks*, T2011/3, April 2011, TRAIL Thesis Series, the Netherlands

Oort, N. van, *Service Reliability and Urban Public Transport Design*, T2011/2, April 2011, TRAIL Thesis Series, the Netherlands

Mahmod, M.K.M., *Using Co-Operative Vehicle-Infrastructure Systems to Reduce Traffic Emissions and Improve Air Quality at Signalized Urban Intersections*, T2011/1, March 2011, TRAIL Thesis Series, the Netherlands

Corman, F., *Real-Time Railway Traffic Management: dispatching in complex, large and busy railway networks*, T2010/14, December 2010, TRAIL Thesis Series, the Netherlands

Kwakkel, J., *The Treatment of Uncertainty in Airport Strategic Planning*, T2010/13, December 2010, TRAIL Thesis Series, the Netherlands

Pang, Y., *Intelligent Belt Conveyor Monitoring and Control*, T2010/12, December 2010, TRAIL Thesis Series, the Netherlands

Kim, N.S., *Intermodal Freight Transport on the Right Track? Environmental and economic performances and their trade-off*, T2010/11, December 2010, TRAIL Thesis Series, the Netherlands

Snelder, M., *Designing Robust Road Networks: a general design method applied to the Netherlands*, T2010/10, December 2010, TRAIL Thesis Series, the Netherlands

Hinsbergen, C.P.IJ. van, *Bayesian Data Assimilation for Improved Modeling of Road Traffic*, T2010/9, November 2010, TRAIL Thesis Series, the Netherlands

Zuurbier, F.S., *Intelligent Route Guidance*, T2010/8, November 2010, TRAIL Thesis Series, the Netherlands

Larco Martinelli, J.A., *Incorporating Worker-Specific Factors in Operations Management Models*, T2010/7, November 2010, TRAIL Thesis Series, the Netherlands

Ham, J.C. van, *Zeehavenontwikkeling in Nederland: naar een beter beleidsvormingsproces*, T2010/6, August 2010, TRAIL Thesis Series, the Netherlands

Boer, E. de, *School Concentration and School Travel*, T2010/5, June 2010, TRAIL Thesis Series, the Netherlands

Berg, M. van den, *Integrated Control of Mixed Traffic Networks using Model Predictive Control*, T2010/4, April 2010, TRAIL Thesis Series, the Netherlands

Top, J. van den, *Modelling Risk Control Measures in Railways*, T2010/3, April 2010, TRAIL Thesis Series, the Netherlands

Craen, S. de, *The X-factor: a longitudinal study of calibration in young novice drivers*, T2010/2, March 2010, TRAIL Thesis Series, the Netherlands

Tarau, A.N., *Model-based Control for Postal Automation and Baggage Handling*, T2010/1, January 2010, TRAIL Thesis Series, the Netherlands

Knoop, V.L., *Road Incidents and Network Dynamics: effects on driving behaviour and traffic congestion*, T2009/13, December 2009, TRAIL Thesis Series, the Netherlands

Baskar, L.D., *Traffic Control and Management with Intelligent Vehicle Highway Systems*, T2009/12, November 2009, TRAIL Thesis Series, the Netherlands

Konings, J.W., *Intermodal Barge Transport: network design, nodes and competitiveness*, T2009/11, November 2009, TRAIL Thesis Series, the Netherlands

Kusumaningtyas, I., *Mind Your Step: Exploring aspects in the application of long accelerating moving walkways*, T2009/10, October 2009, TRAIL Thesis Series, the Netherlands

Gong, Y., *Stochastic Modelling and Analysis of Warehouse Operations*, T2009/9, September 2009, TRAIL Thesis Series, the Netherlands

Eddia, S., *Transport Policy Implementation and Outcomes: the case of Yaounde in the 1990s*, T2009/8, September 2009, TRAIL Thesis Series, the Netherlands