Real-Time Traffic State Prediction: Modeling and Applications

by

Hao Chen

Dissertation submitted to the faculty of Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Civil Engineering

Hesham A. Rakha (Chair)
Kathleen L. Hancock
Feng Guo
Bryan J. Katz

May 5th, 2014

Blacksburg, VA

Keywords: Traffic state prediction, travel time prediction, instantaneous and experienced travel times, Van Aerde model, macroscopic traffic model, computer vision and pattern recognition, dynamic template matching, particle filter, agent-based model.

Real-Time Traffic State Prediction: Modeling and Applications

Abstract

Travel-time information is essential in Advanced Traveler Information Systems (ATISs) and Advanced Traffic Management Systems (ATMSs). A key component of these systems is the prediction of the spatiotemporal evolution of roadway traffic state and travel time. From the perspective of travelers, such information can result in better traveler route choice and departure time decisions. From the transportation agency perspective, such data provide enhanced information with which to better manage and control the transportation system to reduce congestion, enhance safety, and reduce the carbon footprint of the transportation system.

The objective of the research presented in this dissertation is to develop a framework that includes three major categories of methodologies to predict the spatiotemporal evolution of the traffic state. The proposed methodologies include macroscopic traffic modeling, computer vision and recursive probabilistic algorithms. Each developed method attempts to predict traffic state, including roadway travel times, for different prediction horizons. In total, the developed multi-tool framework produces traffic state prediction algorithms ranging from short – (0~5 minutes) to medium-term (1~4 hours) considering departure times up to an hour into the future.

The dissertation first develops a particle filter approach for use in short-term traffic state prediction. The flow continuity equation is combined with the Van Aerde fundamental diagram to derive a time series model that can accurately describe the spatiotemporal evolution of traffic state. The developed model is applied within a particle filter approach to provide multi-step traffic state prediction. The testing of the algorithm on a simulated section of I-66 demonstrates that the proposed algorithm can accurately predict the propagation of shockwaves up to five minutes into the future. The developed algorithm is further improved by incorporating on- and off-ramp effects and more realistic boundary conditions. Furthermore, the case study demonstrates that the improved algorithm produces a 50 percent reduction in the prediction error compared to the classic LWR density formulation. Considering the fact that the prediction accuracy deteriorates significantly for longer prediction horizons, historical data are integrated and considered in the measurement update in the developed particle filter approach to extend the prediction horizon up to half an hour into the future.

The dissertation then develops a travel time prediction framework using pattern recognition techniques to match historical data with real-time traffic conditions. The Euclidean distance is initially used as the measure of similarity between current and historical traffic patterns. This method is further improved using a dynamic template matching technique developed as part of this research effort. Unlike previous approaches, which use fixed template sizes, the proposed method uses a dynamic template size that is

updated each time interval based on the spatiotemporal shape of the congestion upstream of a bottleneck. In addition, the computational cost is reduced using a Fast Fourier

Transform instead of a Euclidean distance measure. Subsequently, the historical candidates that are similar to the current conditions are used to predict the experienced travel times. Test results demonstrate that the proposed dynamic template matching method produces significantly better and more stable prediction results for prediction horizons up to 30 minutes into the future for a two hour trip (prediction horizon of two and a half hours) compared to other state-of-the-practice and state-of-the-art methods.

Finally, the dissertation develops recursive probabilistic approaches including particle filtering and agent-based modeling methods to predict travel times further into the future. Given the challenges in defining the particle filter time update process, the proposed particle filtering algorithm selects particles from a historical dataset and propagates particles using data trends of past experiences as opposed to using a state-transition model. A partial resampling strategy is then developed to address the degeneracy problem in the particle filtering process. INRIX probe data along I-64 and I-264 from Richmond to Virginia Beach are used to test the proposed algorithm. The results demonstrate that the particle filtering approach produces less than a 10 percent prediction error for trip departures up to one hour into the future for a two hour trip. Furthermore, the dissertation develops an agent-based modeling approach to predict travel times using real-time and historical spatiotemporal traffic data. At the microscopic level, each agent represents an expert in the decision making system, which predicts the travel time for each time interval according to past experiences from a historical dataset. A set of agent interactions are developed to preserve agents that correspond to traffic patterns similar to the real-time measurements and replace invalid agents or agents with negligible weights with new agents. Consequently, the aggregation of each agent's recommendation (predicted travel time with associated weight) provides a macroscopic level of output – predicted travel time distribution. The case study demonstrated that the agent-based model produces less than a 9 percent prediction error for prediction horizons up to one hour into the future.

Acknowledgements

First of all, I would like to express my heartfelt gratitude to my advisor, Dr. Hesham Rakha, for his support and mentoring during the course of this research effort. His research attitude and successful career have strongly influenced me, and his critical comments and suggestions enabled me to make my Ph.D. research and this dissertation better than I could have achieved. When there were moments that I doubted my research, Dr. Rakha always supported me and gave me the confidence I needed to move forward. I also want to extend my thanks to my advisory committee members – Dr. Kathleen Hancock, Dr. Feng Guo and Dr. Bryan Katz for their support through this dissertation.

I would like to express my thanks to many others who contributed to this dissertation and my Ph.D. studies. During my PhD research, I had the pleasure to work with wonderful colleagues in the Center of Sustainable Mobility at the Virginia Tech Transportation Institute – Dr. Ihab El-Shawarby, Dr. Jianhe Du, Dr. Shereef Sadek, Dr. Sangjun Park and Dr. Hao Yang. I also want to thank the students working in the same group – Ahmed Amer, Aly Tawfik, Ismail Zohdy, Raj Kishore Kamalanathsharma, Mohammed Elhenawy, John Sangster, Arash Jahangiri, Karim Fadhloun, Huan Li, Andy Edwardes, Boon Tech Ong and many others. It was a great pleasure to work with all of them. I also extend my thanks to all the project and funding sponsors for the work I was involved in during my Ph.D. studies including the Mid-Atlantic Universities Transportation Center (MAUTC), the Virginia Department of Transportation, the Federal Highway Administration, Leidos, and the Connected Vehicle Infrastructure University Transportation Center (CVI-UTC).

Finally, I would like to thank my wife, Shuwen for her understanding and encouragement over the duration of my studies at Virginia Tech. I also would like to thank my parents for their endless love and support throughout my life.

Table of Contents

List of Figures

List of Tables

Chapter 1

1 Introduction

The first chapter provides the scope of this dissertation followed by an introduction of the related problems in traffic prediction. Thereafter, a comprehensive framework including a multi-tool traffic prediction method is presented, with the summary of corresponding contributions to each methodology. Finally, the layout of the dissertation is provided.

1.1 Background

Congestion has proven to be a serious problem across urban areas in the United States. In 2007, it cost highway users 4.2 billion extra hours of sitting in traffic and an extra 2.8 billion gallons of fuel. This all translated into an additional $87.2 billion in congestion costs for road users in 2007, which represented a 50% increase in congestion costs compared to data from the previous decade. Even though the recent economic downturn is said to have marginally eased the congestion problem nationwide, new evidence shows an uptrend of traffic and, consequently, congestion is back [1].

Tackling congestion (both recurrent and non-recurrent) has proven to be a challenge for highway agencies. Adding capacity in response to congestion is becoming less of an option for these agencies due to a combination of financial, environmental, and social issues. Therefore, the main focus has been on improving the performance of existing facilities through continuous monitoring and dissemination of traffic information. The minimum that can be accomplished is to inform the public or, specifically, the potential users of what they should expect on the roadways before and during their trips. Additionally, this information can be applied to provide alternatives to users so that they may make informed decisions about their trips. This is the essence of Advanced Traveler Information System (ATIS) applications such as 511 that have been implemented nationwide. In many states relevant traffic information is also posted on variable message signs (VMS) that are strategically positioned along the highways. However, the effectiveness of such efforts and the accuracy and usefulness of information they provide have been questioned [2].

There are numerous reasons for such deficiencies in providing accurate and timely traffic data to the general public. Apart from sensor errors, communication line failures, and use of naïve estimation methods, the fact that posted travel times are typically based on instantaneous realized travel times (as opposed to predictions) should be counted as the main contributor to the distrust among users and agency officials alike. To remedy this situation, we propose that traffic data from different sources be considered and every attempt be made to complement stream data from point sensors with more accurate data that might become available at certain points in time every once in a while. In this context, the accuracy, update rate, and intrinsic value of different traffic data sources should be considered. Low-cost stationary traffic sensors are most prevalent but have greater measurement errors. Conversely, probes are essentially able to provide an accurate trajectory (time and location) of the vehicle as it passes through a road segment. In this dissertation, both of measurement data from loop detectors and probe vehicles are used as the input for traffic state or travel time prediction algorithms.

1.2 Problem Statement

Various traffic sensing technologies have been used to collect traffic data for use in computing travel times, including station-based traffic state measuring devices (loop

detector, video camera, remote traffic microwave sensor, etc.), and point-to-point travel time collection (license plate recognition systems, automatic vehicle identification systems, mobile, Bluetooth, probe vehicle, etc.). Loop detector data are used as the main data source to develop traffic state prediction algorithms. Private companies such as INRIX integrate different sources of measured data to provide section-based traffic state data (speed, average travel time), which are used as the main data source in our study to develop travel time prediction algorithms. The benefit of using section-based traffic state data is that travel time can be easily calculated from traffic state data. More importantly, the section-based data provide the flexibility for scalable applications on traffic networks.

By providing section-based traffic state data, there are two approaches to compute travel times reflecting the trip experience [3, 4]. Experienced travel time (also called dynamic travel time in this research) is the actual, realized travel time that a vehicle could experience during a trip. If a vehicle leaves its origin at the current time, the roadway speed will not only change over space but also over time during the entire trip. Consequently, experienced travel time can be obtained by using a prediction algorithm to compute the speed evolution in future time steps. Instantaneous travel time is the second approach to compute travel times without the consideration of speed evolution over time. It is usually computed using the current speed along the entire roadway; in other words, the speed field is assumed to remain constant in the future time intervals. The instantaneous travel time is close to the experienced travel time when the roadway speed does not change significantly over time during the trip. However, this approach may deviate substantially from the actual, experienced travel time under transient states during which congestion is forming or dissipating during a trip [5].

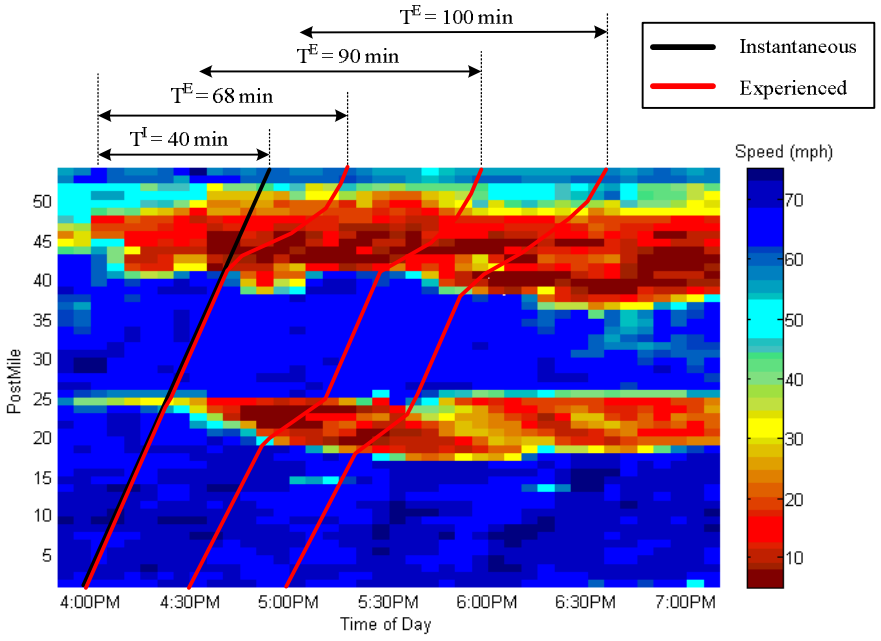


Figure 1.1: Spatiotemporal traffic speed map and trip trajectories on I-66 on June 22, 2013.

Previous research has demonstrated that prediction accuracy typically deteriorates quickly with the increase in the prediction horizon [6]. In order to demonstrate the discrepancy between instantaneous and experienced travel times, especially the errors of using instantaneous information for multi-step prediction of experienced travel time, spatiotemporal traffic speed data provided by INRIX is presented on Figure 1.1. The traffic data is collected along I-64 from Richmond to Norfolk during the afternoon peak hours on June 22, 2013. The trip trajectories are plotted on the contour speed map. According to the black trajectory, the instantaneous travel time is calculated as 40 minutes for a departure at 4:00 pm. Although the traffic on the selected route is uncongested at 4:00 PM, two bottlenecks are rapidly forming. Consequently, the instantaneous travel time on 4:00 PM underestimates the experienced travel time by 28 minutes, 50 minutes and 60 minutes for the prediction horizons of 0 minutes, 30 minutes and 60 minutes, respectively. These errors demonstrate that the instantaneous travel time may not be a good predictor of experienced travel time, especially for multi-step prediction.

During the past decades, many studies have been conducted attempting to predict travel times. According to the manner of modeling, these methods can be classified into time series models including Kalman filter models [7], auto-regressive integrated moving average (ARIMA) models [8] and data-driven methods, such as neural networks [9], support vector regression (SVR) [10] and k-nearest neighbor (*k*-NN) [11, 12] models. These techniques are implemented through direct and indirect procedures to predict travel times using different types of state variables. Travel time is directly used as the state variable in model-based or data-driven methods to predict travel times. Indirect procedures are performed using other variables (such as traffic speed, density, flow, occupancy , etc.) as the state variable to predict the future traffic state, and then future travel times are computed using some transition matrix from traffic status to travel times. This paper attempts to predict experienced travel times for departures at current or future time intervals. For real-time applications, instantaneous travel time can be obtained each time interval by summing the section travel times along the entire roadway. But experienced travel time can only be obtained on the completion of the trip, because the temporal and spatial evolution of speed should be considered. In this case, the experienced travel time for the previous time interval usually is not available for predicting travel times in the next interval, especially for long trips. Consequently, many of the state-of-practice methods cannot work well in predicting experienced travel times [13].

Other than real-time information, historical data provide a pool of past experienced traffic patterns for travel time prediction. Artificial neural networks (ANNs) are widely used to generate a predictor using a large training data set. However, prediction accuracy deteriorates rapidly considering multi-step predictions for ANNs methods [14, 15]. A recurrent neural network approach was developed considering spatiotemporal input dynamics for travel time prediction. Different input variables including traffic volume, occupancy, speed and their combination were used as input and the test results demonstrated that the combination of the three variables produced the best prediction accuracy. However, the prediction error for incident-free test data increased from 5.5% to 15.2% for a prediction horizon of 5 minutes to 25 minutes [16]. In addition, there are

several other deficiencies for ANNs, such as high computation costs associated with the training process, lack of flexibility in dealing with non-recurrent traffic patterns, and difficulty in implementation on large-scale traffic networks or different sites.

This research aims to develop a comprehensive framework to solve the problem of traffic state and travel time prediction for real-time applications. Considering the state-of-art in related research topics, this dissertation is expected to answer the following questions:

1) How to enhance a macroscopic traffic model using the Van Aerde fundamental diagram instead of the Greenshields model?

2) How to use the developed macroscopic traffic model with a Bayesian filtering framework for short-term traffic state prediction?

3) How to incorporate historical data in the developed traffic state prediction approach to extend the prediction horizon and eventually predict dynamic travel times?

4) How to develop algorithms to predict experienced travel times using real-time and historical traffic data?

5) How to predict experienced travel time for the trip departures on future time intervals?

6) How to find the optimum departure time based using a multi-step travel time prediction algorithm?

7) What are the performances of the proposed prediction algorithms using both simulated and field data, and then how to further enhance the prediction algorithms?

1.3 Research Methodology

In order to fulfill the objective of this research, a framework of the research methodology is proposed in Figure 1.2. The framework entails the use of multiple tools to predict traffic state and travel time for real-time applications. Generally, three major categories are included in the framework:

1) A macroscopic traffic modeling approach,

2) A computer vision and pattern recognition approach, and

3) A recursive probabilistic approach.

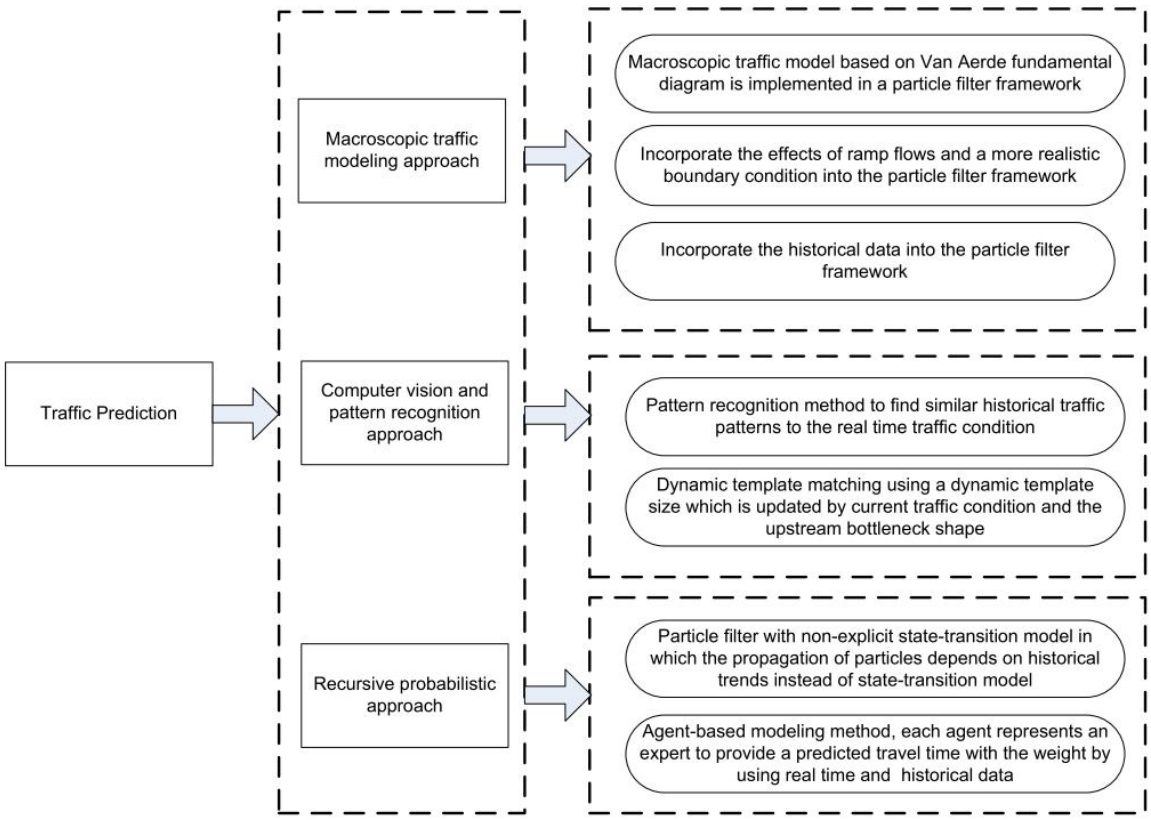


Figure 1.2: A framework of research methodologies.

1.4 Research Contributions

Considering the state-of-the-art in the related research topics, the main contributions of the proposed algorithms in this dissertation are listed below.

1) A comprehensive framework for real-time traffic prediction

   - This dissertation develops a unique comprehensive framework including four major categories of methodologies for real-time traffic state prediction.

2) Macroscopic traffic modeling

   - In this research, the traditional Lighthill-Whitham-Richards (LWR) flow continuity equation is combined with the Van Aerde traffic stream model to generate a new partial differential equation (PDE) named the Van Aerde flow continuity model. The numerical discretization of this model is used to generate a time series equation that characterizes the temporal and spatial relationship of traffic speed data, which can be used in various applications of traffic prediction.

- Compared to a data-driven statistics approach, the proposed time series equation derived from macroscopic traffic models accurately describes the temporal and spatial dynamics of traffic flow behavior along a freeway based on physical principles.

- Incorporating the adjustment of on- and off-ramp flows into the proposed traffic model, thus the model is more consistent to the real-world roadway situation and can provide a more accurate relationship of traffic speed over adjacent time intervals.

3) Traffic state prediction

- Develops a particle filter approach with the Van Aerde flow continuity model for multi-step traffic state prediction. The spatial and temporal propagation of shockwaves can be accurately predicted on the I-66 test site.

- The proposed Van Aerde flow continuity model is proven to be a non-conservative LWR equation. However, the proposed macroscopic traffic model still produces higher prediction accuracy in the same particle filter framework in comparison to the classic density formulation of the LWR equation.

4) Travel time prediction

- Feeds historical data into the proposed particle filter framework to extend the traffic state prediction horizon. Using these data, trip trajectory and corresponding dynamic travel times can be calculated.

- Develops a novel travel time prediction algorithm based on computer vision and pattern recognition techniques by matching historical data with real-time conditions. This work provides a framework to use spatiotemporal traffic data to predict dynamic travel times. More advanced template matching or pattern recognition techniques can be used within the proposed framework to improve prediction efficiency and accuracy.

- Develops a predictive travel time algorithm using dynamic template matching, in which the dynamic template size is updated each time interval based on the spatiotemporal shape of the congestion upstream of the bottleneck. In addition, the computational cost is reduced using a Fast Fourier Transform (FFT) instead of Euclidean distance.

- Develops a new particle filter approach for multi-step look-ahead travel time prediction using real-time and historical traffic data. Given the challenges in defining the particle filter time update dynamic model, the proposed method selects each particle from historical data and the prorogation of particles is accomplished using the data trends within the historical data sequences instead of a dynamic model.

- Develops a new agent-based modeling approach for travel time prediction. At the microscopic level, each agent represents an expert in the decision making system that predicts the travel time for each time interval according to past experiences from a historical dataset.

- The proposed travel time prediction methods are not sensitive to probe data. Other data sources, such as loop detector, blue tooth and cell phone, etc. can also be considered as the input data as long as the speed matrix along temporal and spatial can be estimated.

1.5 Dissertation Layout

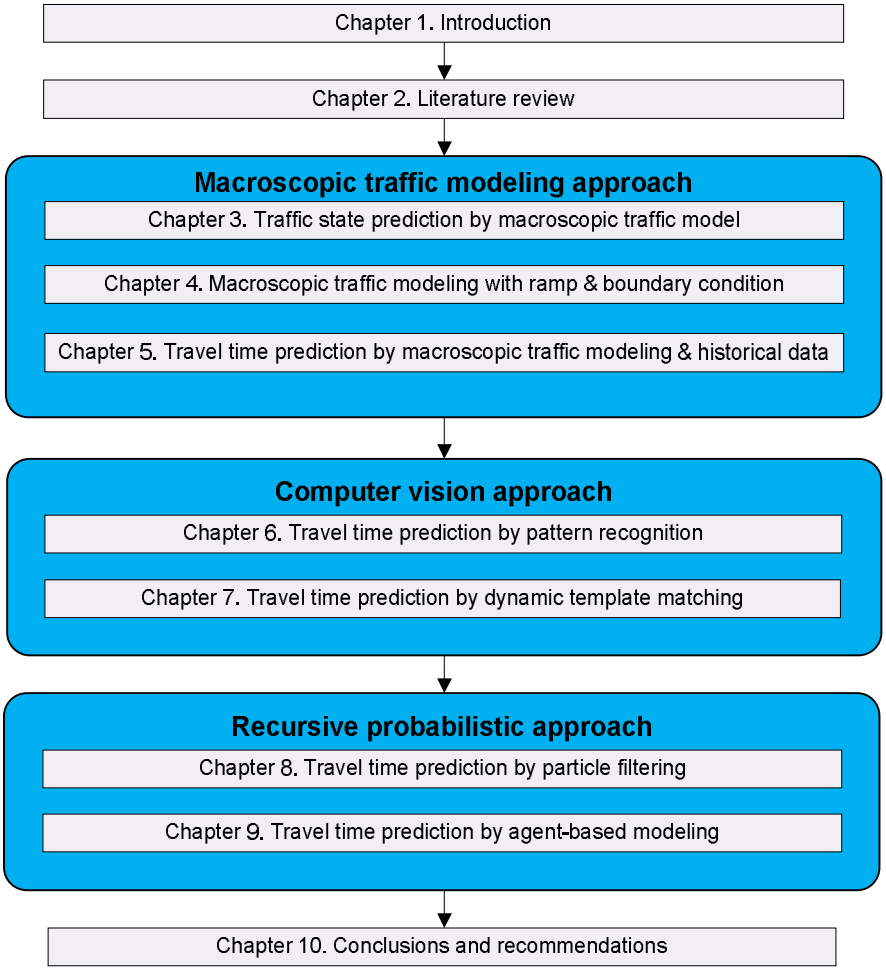


Figure 1.3: Layout of the dissertation.

The research efforts in this dissertation attempt to solve the aforementioned problems to predict traffic state and travel time with respect to theoretical models and practical

applications. The layout of the dissertation is presented in Figure 1.3, and includes ten chapters. The brief descriptions of each of these chapters are given below:

Chapter 1: This chapter gives an introduction to the dissertation topics along with the research methodologies and an overall proposal layout.

Chapter 2: A literature review is presented in this chapter to summaries the related research topics of traffic prediction. Firstly, the state-of-art traffic state estimation and prediction methods are discussed, followed by an overview of existing travel time prediction methods with the corresponding problems.

Chapter 3: The research in this chapter is co-authored with Hesham A. Rakha and Shereef A. Sadek, and was presented at 14th International IEEE Conference on Intelligent Transportation Systems, Washington, D.C. This chapter develops a multi-step traffic state prediction algorithm using spot speed measurements. The traditional Lighthill-Whitham-Richards (LWR) flow continuity equation is combined with the Van Aerde traffic stream model to generate a new partial differential equation (PDE). The developed PDE is implemented in a particle filter framework to predict traffic state.

Chapter 4: This chapter is co-authored with Hesham A. Rakha, Shereef A. Sadek, and Bryan J. Katz, and was presented at 91st Transportation Research Board Annual Meeting, Washington D.C. This chapter improves the speed formulation of chapter 3 by incorporating ramp flows and enhancing the boundary conditions. The numerical solution and near-term prediction accuracy (5-minute prediction) of the new speed formulation is compared with the conservative density formulation derived from LWR.

Chapter 5: This chapter develops a novel approach to construct vehicle trajectories using real-time and historical traffic data in the previous proposed particle filter framework. Co-authored with Hesham A. Rakha, this work was presented at 92nd Annual Meeting Transportation Research Board, Washington D.C.

Chapter 6: The research in this chapter is co-authored with Hesham A. Rakha and Catherine C. McGhee (VDOT) and was presented at 20th World Congress on Intelligent Transportation Systems, Tokyo, Japan. This chapter develops a travel time prediction algorithm based on computer vision and pattern recognition techniques by matching historical data with real-time traffic conditions. Moreover, a data reduction procedure is developed to transfer the raw INRIX probe data into a uniform format which can be used in the proposed travel time prediction algorithm.

Chapter 7: This chapter develops a predictive travel time algorithm using dynamic template matching to identify similar spatiotemporal trends in a historical dataset with regarding to the real-time traffic conditions. Unlike previous approaches, which use fixed template sizes, the proposed method uses a dynamic template size that is updated each time interval based on the spatiotemporal shape of the congestion upstream of the bottleneck. In addition, the computational cost is reduced using a Fast Fourier Transform instead of Euclidean distance. This work is co-authored with Hesham A. Rakha and was submitted for publication in IEEE Transactions on Intelligent Transportation Systems.

Chapter 8: This chapter is co-authored with Hesham A. Rakha, and was accepted for publication in Transportation Research Part C: Emerging Technologies. The research in this chapter develops a particle filter approach for the real-time short to medium-term travel time prediction using real-time and historical data. Given the challenges in defining the particle filter time update process, the proposed algorithm selects particles from a historical database and propagates particles using historical data sequences as opposed to using a state-transition model.

Chapter 9: This chapter develops an agent-based modeling approach to predict experienced travel times using real-time and historical spatiotemporal traffic data. At the microscopic level, each agent represents an expert in the decision making system, which predicts the travel time for each time interval according to past experiences from a historical dataset. The work in this chapter is co-authored with Hesham A. Rakha, and was presented at 93rd Annual Meeting Transportation Research Board, Washington D.C.

Chapter 10: This chapter summaries the main finding of the dissertation. Furthermore, the recommendations for various traffic prediction problems in previous chapters are provided in terms of theoretical improvements and practical applications.

The main contents of this dissertation have been published or are still under review in the listed journals and/or conferences as indicated below.

Journal Publications

1. **Hao Chen** and Hesham A. Rakha (2014), "Real-time Travel Time Prediction by Particle Filtering with non-Explicit State-Transition Model," *Transportation Research Part C: Emerging Technologies*. In press.

2. Mohammed Elhenawy, **Hao Chen** and Hesham Rakha (2014), "Dynamic Travel Time Prediction using Data Clustering and Genetic Programming," *Transportation Research Part C: Emerging Technologies*. Volume 42, pp. 82-98.

3. **Hao Chen**, Hesham A. Rakha, and Shereef A. Sadek (2014), "Real-time Freeway Traffic State Prediction using Particle Filtering," submitted to *Journal of Intelligent Transportation Systems*. (under review).

4. **Hao Chen** and Hesham A. Rakha (2014), "Multi-step Prediction of Experienced Travel Times using Agent-based Model," submitted to *Transportation Research Part C: Emerging Technologies*. (under review).

5. **Hao Chen** and Hesham A. Rakha (2014), "Predicting Freeway Travel Times using Dynamic Template Matching," submitted to *IEEE Transactions on Intelligent Transportation Systems*. (under review).

Conference Publications

6. **Hao Chen**, Hesham A. Rakha, and Shereef A. Sadek (2011), "Real-time Freeway Traffic State Prediction: A Particle Filter Approach," in *14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, pp. 626-631.

7. **Hao Chen**, Hesham A. Rakha, Shereef A. Sadek, and Bryan J. Katz (2012), "A Particle Filter Approach for Real-time Freeway Traffic State Prediction," in *91st Transportation Research Board Annual Meeting*, Washington D.C.

8. **Hao Chen** and Hesham A. Rakha (2012), "Prediction of Dynamic Freeway Travel Times based on Vehicle Trajectory Construction," in *15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, pp. 576 - 581.

9. **Hao Chen** and Hesham Rakha (2013), "Data-Driven Particle Filter for Travel Time Prediction," in *92nd Annual Meeting Transportation Research Board*, Washington D.C.

10. **Hao Chen** and Hesham Rakha (2013), "Forecasting Freeway Dynamic Travel Times by Constructing Trip Trajectories," in *92nd Annual Meeting Transportation Research Board*, Washington D.C.

11. **Hao Chen**, Hesham A. Rakha, and Catherine C. McGhee (2013), "Dynamic Travel Time Prediction using Pattern Recognition," in *20th World Congress on Intelligent Transportation Systems*, Tokyo, Japan, **Best Scientific Paper Award**.

12. **Hao Chen**, Hesham A. Rakha (2013), "An Agent-based Modeling Approach to Predict Experienced Travel Times," in *Conference on Agent-based Modeling in Transportation Planning and Operations*, Blacksburg, VA.

13. Mohammed Elhenawy, Hesham A. Rakha, and **Hao Chen** (2013), "An Automated Statistically-principled Bottleneck Identification Algorithm (ASBIA)," in *16th International IEEE Conference on Intelligent Transportation Systems*, The Hague, Netherlands. pp. 1846-1851.

14. **Hao Chen** and Hesham Rakha (2014), "Data-driven Particle Filter for Multi-step Look-ahead Travel Time Prediction," in *93rd Annual Meeting Transportation Research Board*, Washington D.C.

15. **Hao Chen** and Hesham Rakha (2014), "An Agent-based Modeling Approach to Predict Experienced Travel Times," in *93rd Annual Meeting Transportation Research Board*, Washington D.C.

16. Mohammed Elhenawy, **Hao Chen** and Hesham Rakha (2014), "Dynamic Travel Time Prediction using Genetic Programming," in *93rd Annual Meeting Transportation Research Board*, Washington D.C.

17. Mohammed Elhenawy, **Hao Chen** and Hesham Rakha (2014), "Random Forest Travel Time Prediction Algorithm using Spatiotemporal Speed Measurements," in *21st World Congress on Intelligent Transportation Systems*, Detroit, USA.

Technical Reports and Other Publications

18. Hesham A. Rakha, **Hao Chen**, Ali Haghani and Kaveh Farokhi Sadabadi (2013), "Assessment of Data Quality Needs for Use in Transportation Applications," Report No. MAUTC-2011-01, [Sponsored by: U.S. Department of Transportation, University Transportation Centers Program].

19. Hesham A. Rakha and **Hao Chen** (2013), "Prediction of Dynamic Travel Times: The Richmond to Virginia Beach Case Study," [Sponsored by: Virginia Department of Transportation].

20. **Hao Chen** (2013), "Real-time Freeway Travel Time Prediction by Matching Traffic Patterns," **1st Place Award in 19th Annual ITS Virginia Student Paper Contest**.

21. **Hao Chen** (2013), "Rural Traveler Information System: Travel Time Prediction based on Template Matching," **3rd Place Award in the 4th Annual National Rural ITS Student Paper Competition**.

Chapter 2

2 Literature Review

A literature review is presented in this chapter to summaries the related research topics of traffic prediction. Firstly, the state-of-art traffic state estimation and prediction methods are discussed, followed by an overview of existing travel time prediction methods with the corresponding problems.

2.1 Traffic state estimation and prediction

Traffic state estimation and prediction are important for traffic surveillance and control. Travel time can be estimated or predicted afterward based on traffic state values [6, 17].

Since traffic states are usually not measured everywhere and measurement errors exist, traffic state estimation is necessary when dealing with local and noisy sensing data [18, 19]. Alternatively, in the case of traffic state prediction, current traffic measurement data are used to forecast future traffic flow variables. Recently, the implementation of various traffic macroscopic models within recursive Bayesian filter approaches has been widely used for both traffic state estimation and prediction problems [18-24]. For each time interval, both the time update (prediction) and measurement update (estimation) processes are included in this framework. The sequence of the two processes within a single time interval categorizes the problem as data estimation or prediction. Once new measurement data are available, they are used to adjust the prior predicted value and obtain the estimation. Conversely, prediction is calculated by implementing the estimated value in the time update equation.

A framework with different combinations of macroscopic traffic models and Bayesian filtering technologies has been used to estimate or predict traffic state variables in recent years [18-24]. There are two main advantages to this framework. Within the time update process, the relationship of traffic parameters across adjacent time intervals is accurately characterized by macroscopic traffic models. Apart from this, the recursive framework ensures that traffic state data are efficiently calculated using only data from previous states, not the entire history [25].

To construct this recursive Bayesian filter framework, a time series equation is needed to predict future traffic variables (i.e., flow [q], space mean speed [u], and density [k]) using current measured data. A macroscopic traffic system can be used to track the temporal and spatial dynamic traffic flow behavior along a freeway by constructing a time series traffic variable update equation [18]. Computing the traffic stream flow as the product of the traffic stream space mean speed and density reduces the problem to two independent variables that characterize the traffic stream state. If we assume that a fundamental diagram exists, then there is a unique relationship between traffic stream density and speed. Consequently, a time series equation used to predict a single traffic stream variable (the numerical solution of a first-order partial differential equation [PDE]) is required to estimate the three traffic stream variables. This is classified as a first-order macroscopic traffic model. Alternatively, if the fundamental diagram is not strictly enforced, a second equation is needed for the second traffic flow variable. The model is considered a second-order macroscopic traffic model by predicting two traffic variables (the numerical solution of two first-order PDEs).

Instead of a data-driven statistical approach, the time series equation derived from macroscopic traffic models has the advantage of describing the temporal and spatial dynamics of traffic flow behavior along a freeway based on physical principles. It has strong robustness and is easy to implement at various freeway locations. For instance, a

modeling algorithm proposed in [21] uses sending and receiving functions to represent the traffic perturbation behavior of propagating forward and backward. Both traffic flow and speed are used in the state variable to construct a second-order macroscopic model. A similar second-order macroscopic model was proposed in [23] using the concept of handoff, which is a mechanism that transfers an ongoing call from one cell to another when a cell phone user moves through the coverage area of a cellular system. Sending/receiving functions represent the vehicles that intend to leave/enter a segment. A more popular approach with which to derive time series equations is the traditional Lighthill-Whitham-Richards (LWR) model, which ensures that vehicle conservation is maintained on the road. For instance, the conservation equation in [22] is directly derived from LWR, and the change of section traffic flow for each time interval is constrained by the freeway section flow supply and demand. Both freeway density and flow volume are the state variables and are measured using loop detectors to estimate traffic state and then predict travel time. The conservation equation is also included in a second-order macroscopic traffic flow model and is used in the prediction process computation for traffic state estimation in [18, 19]. Loop detector data are also used for measuring the state variables of traffic flow and speed. Within the application of using speed data from mobile devices to estimate the freeway traffic state, a velocity cell transmission model (CTM-v) is derived from the LWR by replacing the traffic flow and density with traffic speed based on Greenshields' fundamental diagram [20]. Follow-up research demonstrates that the solution of the new PDE is equivalent to the LWR PDE under a quadratic flux function [24].

After obtaining the time series equation, a recursive Bayesian approach is used to incorporate the measurement data to update state variables from the time series equation. The classical KF is the easiest way to incorporate the error between state prediction and measurement data for estimation purposes. However, the classical KF works ideally only for linear systems with Gaussian noise. Since most of the derived time series equations are characterized by nonlinear behavior, an extended Kalman filter (EKF) is widely used for traffic state estimation [18, 19]. EKF is a revised classical KF with the calculation of Jacobian expression. This method has the same advantage as the classical method of propagating the error covariance matrix but can deal with a nonlinear system using Tyler estimation. However, it is difficult to compute the Jacobian expression for many nonlinear time series equations. By overcoming the defect of Jacobian computation and producing ideal accuracy for a nonlinear estimation, Ensemble Kalman filter (EnKF) enables the use of nonlinear evolution equation while exploiting the linear observation equation. EnKF uses Monte Carlo integrations to maintain the nonlinearity of error statistics. It has the same feature as KF that propagates errors by Kalman gain [20] and provides the average estimation or prediction output. However, it cannot deal with the model of nonlinear measurement equation and cannot output reliability state information. Regarding these problems, a powerful approach named a particle filter is implemented, which is applicable for any nonlinear system of equations and has no requirement for the distribution of the system noise [21]. A particle filter provides another benefit in that it delivers the estimation and prediction results as a distribution instead of just one value [25].

Although many studies have been conducted using different combinations of macroscopic traffic models and filtering techniques, these approaches suffer from a number of deficiencies. First, simple traffic stream models may not accurately replicate field-observed traffic stream characteristics. For instance, the Greenshields model does not provide sufficient degrees of freedom to replicate field observations [26]. As a result, the traffic state time series equation derived from simple traffic stream models usually results in lower estimation or prediction accuracy. On the other hand, a complex model (e.g., second-order macroscopic model) has many parameters; thus, the calibration of the model becomes a challenge [18, 19, 21, 23]. Third, although deriving the time series equation from the LWR based on vehicle conservation law is a promising approach, no underlying traffic stream models other than the Greenshields model have been tested or evaluated. Furthermore, although these methods can be used for traffic state prediction in terms of the features of recursive Bayesian filters, few prediction results are presented or compared. Since predicting future traffic state data has many realistic applications for ramp metering, incident detection, and travel information broadcasting [22], it is necessary to conduct research about traffic state prediction by using macroscopic traffic model with Bayesian filters.

2.2 Travel time prediction

Travel time prediction is an essential part of an Advanced Traffic Management System (ATMS) and Advanced Traveler Information System (ATIS). The Federal Highway Administration (FHWA) encourages all Traffic Management Centers (TMCs) to post travel time and incident information, which not only provides useful information to motorists but also assists them in planning their route choices. This planning can cause a small number of drivers to divert away from the congested highway, thus providing critical additional capacity and assisting in the management of congestion [27].

Traffic state in the near future usually cannot provide enough information to cover the entire trip, especially for long trips. For instance, in the case of a 100-mile trip, departures at the current time would still be traveling 1 hour in the future even under free-flow traffic conditions. For this case, the traffic state for the following 1 hour or more should be predicted in order to compute dynamic travel times. An alternative to solving this problem is to use historical data. The historical data set provides a pool of past experienced traffic patterns which can be useful to extend the perdition horizon.

During past decades, many studies have been conducted to predict travel times. Some of the reviews of different methods can be found in earlier publications [9, 11, 28, 29]. According to the manner of modeling, those methods can be classified into time series models, including: Kalman filter [7, 30], Auto-Regressive Integrated Moving Average (ARIMA) models [8, 30, 31] and data-driven methods, such as neural networks [9, 32], support vector regression (SVR) [10, 33] and k nearest neighbor (*k*-NN) [11, 12, 34] models. These techniques are implemented through direct and indirect procedures to predict travel times using different types of state variables. Travel time is directly used as the state variable in model-based or data-driven methods to predict travel times. Indirect

procedures are performed by using other variables (such as traffic speed, density, flow, occupancy, etc.) as the state variable to predict traffic status, and then future travel time can be calculated based on the transition to predicted traffic status.

Time series models construct the time series relationship of travel time or traffic state, and then current and/or past traffic data are used in the constructed models to predict travel times in the next time interval [35]. Kalman filter is a popular method for data estimation and tracking, in which time update and measurement update processes are included. A time series equation is used to predict state variables and then state values are corrected according to the new measurement data. The main advantage of KF is that the recursive framework ensures that traffic data is efficiently updated using only data from previous states and not the entire history [6]. Kalman filter methods were proposed to predict travel times using Global Positioning System (GPS) information and probe vehicle data [30, 36]. The state transient parameter in the time series equation is defined from average historical data to calculate future travel times. A similar idea was used in the Bayesian dynamic linear model for real-time short-term travel time prediction [7]. The system noise can be adjusted for unforeseen events (incidents, accidents, or bad weather) and integrated into the recursive Bayesian filter framework to quantify random variations on travel times. The experiment results based on loop detector data from a segment of I-66 demonstrates that the proposed method produces higher prediction accuracy under both recurrent and non-recurrent traffic conditions. However, in these methods a problem exists in that the travel time in the previous time interval is needed to calculate the future travel time. For real-time applications, the travel time is usually greater than the time interval step size. Hence, the actual travel time from the previous time interval is not available to apply in the algorithms used to predict travel times for the next time interval.

A seasonal ARIMA model was proposed to quantify the seasonal recurrent pattern of traffic conditions (occupancy) [8, 31]. Moreover, an embedded adaptive Kalman filter was developed in order to update the occupancy estimate in real-time using new traffic volume measurements. Consequently, multi-step look-ahead occupancy information is estimated to obtain a data matrix representing the temporal-spatial traffic conditions for the future trip. Since travel time cannot be directly computed through traffic conditions (occupancy), future traffic speed can be calculated using occupancy data by assuming an average vehicle length and using a constant conversion factor known as the g-factor in the literature. Consequently, dynamic freeway corridor travel times are predicted with the consideration of traffic state evolution along the corridor. However, this approach may be difficult to implement since the described recurrent pattern of traffic conditions may not be found everywhere.

Data-driven methods usually predict travel times using a large amount of historical traffic data. Time series models are not specified in the data-driven methods, considering the complex randomness of traffic systems. Neural networks can be trained from historical data to discover hidden dependencies which can be used for predicting future states. A state-space neural network (SSNN) method was proposed to predict freeway travel times for missing data [9]. The missing data problem was tackled by simple imputation schemes, such as exponential forecasts and spatial interpolation. Travel time was the

direct state variable used for the training process and the experiment results demonstrated that the SSNN methods produced accurate travel time predictions on inductive loop detector data. Supported vector machine (SVM) is a successor to artificial neural network (ANN); it has greater generalization ability and is superior to the empirical risk minimization principle as adopted in ANN [10]. The application of SVM to time series forecasting is called support vector regression (SVR). The SVR predictor was demonstrated to perform well for travel time prediction. The point-to-point travel time is usually used as the input to ANNs and SVRs. However, both methods require long training processes and are nontransferable to other sites [11].

The *k*-NN method can be used to find several candidate sequences from historical data by matching current with short past data sequences. Travel time and occupancy sequences were used to predict dynamic travel times using the *k*-NN method with combined data from vehicle detectors and automatic toll collection systems [11]. The occupancy was used since travel time sequence was collected for the recent past time intervals. The results from the case study demonstrated the improvement of prediction accuracy by combining two types of sequences for the matching process. Moreover, a *k*-NN method was proposed by selecting candidates through the Euclidean distance and data trend measures to predict freeway travel times under different weather conditions [12]. Unlike ANNs and SVRs, *k*-NN methods are easy to implement at different sites without data training required.

2.3 Summary

Considering the above mentioned problems on the state-of-art traffic state and travel time prediction studies, this dissertation aims to develop a comprehensive framework including various traffic prediction algorithms. The performance of traffic state and travel time prediction can be improved by the proposed algorithms in terms of different application environments. The research gap of using macroscopic traffic modeling and Bayesian filters for short-term multi-step traffic state prediction is fulfilled by chapter 3 and 4. Based on the proposed particle filter traffic estimation and prediction framework, a further application to predict dynamic travel time by constructing trip trajectories with real-time and historical traffic measurements is investigated in chapter 5. Moreover, the methodologies used in other application fields are modified to solve the problems of travel time prediction. Specifically, pattern recognition and dynamic template matching are used in chapter 8 and 9, respectively, to predict travel time by the similar historical traffic patterns to the real-time traffic condition. Furthermore, chapter 6 proposes a data-driven particle filter to replace dynamic model by data trend sequences in historical dataset. Lastly, an agent-based modeling approach is developed in chapter 7 to fulfill the gap of agent based method and travel time prediction.

Chapter 3

3 Traffic State Prediction based on Macroscopic Traffic Modeling

This chapter is an edited version of: Hao Chen, Hesham A. Rakha, and Shereef A. Sadek (2011), "Real-time Freeway Traffic State Prediction: A Particle Filter Approach," in *14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, pp. 626-631.

The research presented in this chapter develops a multi-step traffic state prediction algorithm using spot speed measurements. The traditional Lighthill-Whitham-Richards (LWR) flow continuity equation is combined with the Van Aerde traffic stream model to generate a new partial differential equation (PDE) named the Van Aerde flow continuity model. The numerical solution of the PDE is obtained using the Godunov discretization scheme to generate a time series equation that characterizes the temporal and spatial relationship of traffic speed data. Because of the strong nonlinearity of the discretized speed update equation, a robust particle filter is applied to conduct a muti-step speed prediction using speed measurements. The prediction accuracy of the proposed approach is compared to the state-of-the-art Ensemble Kalman filter with the Greenshields traffic stream model using simulated loop detector data from Interstate 66. The results demonstrate that the proposed particle filter approach in combination with the discretized Van Aerde flow continuity model produces the lowest prediction error of 4.3 km/h for a five-minute prediction horizon, and accurately predicts the spatial and temporal propagation of shockwaves.

3.1 Introduction

Traffic state estimation and prediction problems are significantly important for freeway traffic surveillance and control. The purpose of estimation is to use limited available measured traffic data to estimate missing observations for the same time interval [18]. Alternatively, in the case of traffic state prediction, traffic parameters are adjusted using current measurement data and then used to predict future traffic states. Recently, research has focused on the use of macroscopic traffic flow models with recursive Bayesian filters for traffic state estimation and prediction [18, 20-22]. In this framework, there is a time update process (prediction) and measurement update process (estimation). The sequence of implementing the two processes distinguishes between data estimation and prediction problems. Once the current measured data are available they are used to adjust the predicted value in the traffic state estimation step. On the other hand, prediction is obtained if estimated values are implemented in the time update equation. There are two main advantages in this framework. Within the time update process, the relationship of traffic parameters over adjacent time intervals is accurately characterized by macroscopic traffic stream models. In addition, the recursive framework ensures traffic state data are efficiently calculated only using data from previous states, not the entire history [25].

In order to construct this recursive Bayesian filter framework, a time series equation is needed to predict future traffic variables (flow (q), space mean speed (u), and density (k)) using current measured data. A macroscopic traffic system can be used to track the temporal and spatial dynamic traffic flow behavior along a freeway by constructing a time series traffic variable update equation [18]. Computing the traffic stream flow as the product of the traffic stream space mean speed and density, reduces the problem to two independent variables that characterize the traffic stream state. If we assume that a fundamental diagram exists there is a unique relationship between traffic stream density and speed. Consequently, a time series equation to predict a single traffic stream variable (the numerical solution of a first-order PDE) is required to estimate the three traffic stream variables. This is classified as a first-order macroscopic traffic model. Alternatively, if the fundamental diagram is not strictly enforced, a second equation is needed for the second traffic flow variable and the model is considered a second-order macroscopic traffic model by predicting two traffic variables (the numerical solution of two first-order PDEs). Different Bayesian filtering methods, such as a Kalman filter, Extended Kalman filter, Unscented Kalman filter or particle filter, can be used to update the predicted values, depending on whether the time series equation is linear or non-linear and how to incorporate measured data.

Different combinations of traffic models and Bayesian filtering techniques have been used to estimate or predict traffic state variables in recent years [18, 20-24]. A first-order partial differential equation (PDE) was discretized by following Greenshields traffic stream model to derive a speed time series equation [20]. Subsequently, an Ensemble Kalman filter was implemented with the Cell Transmutation Model - velocity (CTM-V) to reduce computational time for estimating speed using GPS data. Follow-up research demonstrated that the solution of the new PDE is equivalent to the LWR PDE under a quadratic flux function [24]. The numerical solution of a second-order macroscopic

traffic flow model was used to predict speed and flow variables. An Extended Kalman filter was applied for the nonlinear time update equation to estimate the three traffic stream variables using loop detector data [18]. By following the fundamental diagram between flow and density, a first-order macroscopic model was used to estimate and predict traffic conditions. A particle filter was found to be the best approach because of the nonlinear traffic variable update feature [22]. Similar reasoning was considered in 23] which implemented a particle filter with different nonlinear traffic state time series equations for data estimation.

These methods suffer from a number of deficiencies. First, simple traffic stream models may not replicate field observed traffic stream characteristics accurately. For instance, the Greenshields model does not provide sufficient degrees of freedom to replicate field observations [37]. As a result, the traffic state time series equation derived from simple traffic stream models usually results in lower estimation or prediction accuracy. Alternatively, a complex model (e.g. second-order macroscopic model) has many parameters and thus the calibration of the model becomes a challenge [19, 21]. Furthermore, although these methods can be used for traffic state prediction in terms of the feature of recursive Bayesian filters, very few prediction results were presented or compared. Since predicting future traffic state data has many realistic applications for ramp metering control, incident detection and travel information broadcasting [22], it is necessary to conduct research on traffic state prediction (e.g. predicting a shockwave that will propagate past a specific location after 5 minutes).

The method proposed in this chapter can efficiently overcome the above problems. A first-order macroscopic traffic stream model is proposed to predict speed estimates using the Van Aerde traffic stream model, which has been demonstrated to fit field data more accurately compared to the Greenshields, Greenberg and Pipes traffic stream models [37]. Consequently, the Van Aerde traffic stream model is used in conjunction with the LWR flow continuity equation in non-conservative form to derive a new Van Aerde flow continuity model. Godunov's scheme is used to discretize the proposed model to obtain a speed update equation to model the relationship between traffic stream speeds between neighboring freeway sections over adjacent time intervals. Since the speed time series equation has a strong nonlinear relationship, a particle filter is considered for data filtering. In order to enhance the weight updating procedure in the particle filter framework, the average error is eliminated in the likelihood calculation for different freeway sections. Considering the feature of multi-step state prediction, a new particle filter approach is proposed using speed measurements from freeway sections. Only four basic traffic stream parameters (free-flow speed (u_f), speed-at-capacity (u_c), capacity (q_c) and jam density (k_j)) are required in the proposed method and thus it is easy to calibrate. The loop detector data from the simulated Interstate 66 freeway network are used to test the different approaches.

The remainder of this chapter is organized as follows. The derivation for Van Aerde flow continuity model and the corresponding numerical discretized solution are introduced first. Thereafter, the proposed particle filter prediction approach is described. Subsequently, a demonstration of the field data testing and simulation test case are

presented. This is followed by a comparison of the four prediction frameworks. Finally, the main conclusions of the study and recommendations for future research are presented.

3.2 Model description

A discretized Van Aerde flow continuity model is proposed to describe the speed evolution over space and time along a homogenous freeway section. This assumes that the four basic traffic stream parameters (u_f, u_c, q_c, and k_j) are identical for different freeway segments, as will be described in the model derivation section.

3.2.1 Van Aerde flow continuity model

The flow continuity equation proposed in [24, 38] models highway traffic flow using one-dimensional kinematic wave motion theory. The continuity equation ensures that vehicle conservation is maintained and is written as

$$\frac{\partial k}{\partial t}+\frac{\partial q}{\partial x}=0 \tag{3.1}$$

where q and k are the traffic stream flow rate and density, respectively, at any instant t and location x along the roadway. Equation (3.1) has two independent variables k and q. Lighthill and Whitham [39] and Richards [40] independently hypothesized a relationship between traffic stream speed and density – the fundamental diagram, in which traffic speed is only a function of density ($u=u(k)$). Consequently, traffic flow can be represented as the flux function of k ($q=q(k)=ku(k)$). Equation (3.1) can then be written as

$$\frac{\partial k}{\partial t}+c\frac{\partial k}{\partial x}=0 \tag{3.2}$$

where c is the derivative of q with respect to k and represents the kinematic wave speed. This relationship is a function of only the traffic density, and thus can be solved by discretizing the solution over space and time considering initial and boundary conditions. However, density is not measured directly in the field. Since traffic stream speed can be measured using different technologies (loop detectors, remote traffic microwave sensors, Bluetooth detectors, license tag readers or probe vehicles) and density can also be represented as a function of speed only, a similar governing equation can be obtained for speed. Consequently, the evolution of traffic speed over space and time can be calculated by the numerical solution of the derived equation. For this purpose, a practical invertible function between traffic stream speed and density is needed.

In this section, the relationship between traffic density and speed follows the Van Aerde traffic stream model because of its comprehensiveness, simplicity and ease of calibration. The model was proposed by Van Aerde [41] and Van Aerde and Rakha [42]. The Van-Aerde model is expressed as

$$k = \left(c_1 + \frac{c_2}{u_f - u} + c_3 u \right)^{-1} \tag{3.3}$$

Here u is the traffic stream space mean speed (km/h), u_f is the facility free-flow speed (km/h), c_1 is a fixed distance headway constant (km), c_2 is a variable headway constant (km^2/h), and c_3 is a variable distance headway constant (h). This combination provides a functional form with four degrees of freedom by allowing the speed-at-capacity (u_c) to differ from the free-flow speed (u_f), as is the case in most models including the Pipes, Gipps, and Newell's simplified model, or half the free-flow speed, as is the case with the Greenshields model. By considering three boundary conditions, we can compute the model constants as

$$c_1 = \frac{u_f}{k_j u_c^2}\left(2u_c - u_f\right); \quad c_2 = \frac{u_f}{k_j u_c^2}\left(u_f - u_c\right)^2; \quad c_3 = \frac{1}{q_c} - \frac{u_f}{k_j u_c^2}. \tag{3.4}$$

The derivatives of flow and density with respect to time and space can be computed for the Van Aerde model, as demonstrated in Equation (3.5) and Equation (3.6). The flow continuity equation can then be cast as Equation (3.7).

$$\frac{\partial k}{\partial t} = \frac{\partial}{\partial t}\left(\left(c_1 + \frac{c_2}{u_f - u} + c_3 u \right)^{-1} \right) \tag{3.5}$$

$$\frac{\partial q}{\partial t} = \frac{\partial}{\partial t}\left(u\left(c_1 + \frac{c_2}{u_f - u} + c_3 u \right)^{-1} \right) \tag{3.6}$$

$$\frac{\partial u}{\partial t} + \frac{2c_2 u - c_2 u_f - c_1 (u_f - u)^2}{c_2 + c_3 (u_f - u)^2} \cdot \frac{\partial u}{\partial x} = 0 \tag{3.7}$$

In order to solve the above partial differential equation, Equation (3.7) should be transformed to a one-dimensional scalar conservation law of the form shown below.

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0 \tag{3.8}$$

Such that the velocity flux function satisfies the following relation.

$$\frac{dq}{dk} = \frac{dF}{du} \tag{3.9}$$

Integrating the above relation yields the following expression for F,

$$F = f(u) = -\frac{c_1}{c_3}u + \frac{c_2}{c_3}\ln\left[c_3\left(u_f - u\right)^2 + c_2\right] - \sqrt{\frac{c_2}{c_3}}\left(\frac{c_1}{c_3} + u_f\right)\arctan\left[\sqrt{\frac{c_3}{c_2}}\left(u_f - u\right)\right] \quad (3.10)$$

The above Equation (3.10) is the proposed Van Aerde flow continuity model. According to this equation, the partial differential of speed u over time plus the partial differential of the flux F over space is equal to zero. Here, the flux F is a function of speed, $f(u)$. Speed is the only variable in this flux function and other coefficients can be calculated using the four basic traffic stream parameters as was demonstrated earlier in Equation (3.4). Here, the proposed PDE of Equation (3.8) is non-conservative for the roadway density and isn't equivalent to the LWR PDE of Equation (3.1).

Given the fact that the above mentioned formulation assumes a differentiable flux function, q, a shock fitting algorithm can be implemented to resolve sharp discontinuities. Such schemes are not implemented in the current work, however, the error from the proposed PDE will be most significant in the case of very light traffic approaching stopped vehicles.

3.2.2 Numerical discretization

Since the Van Aerde flow continuity equation is a continuous PDE over time and space, it cannot be directly used in a Bayesian filter framework. Godunov's scheme, which is a conservative numerical scheme for solving PDEs, can be implemented here to discretize the original continuous equation [43]. In this method, the numerical solution is calculated in a piecewise constant manner to represent the conservation variable over a spatial grid at each time step.

A cell-centered finite volume approach is used to discretize (8) where all flow variables are stored at cell centers and the flux function F is approximated at the cell interface. A freeway stretch of length L, can be discretized into N control volumes of equal length ($\Delta x=L/N$). In this way, u^n_i or $u(x_i,t_n)$ represents the speed of the traffic stream on i^{th} control volume at time $n\Delta t$. Equation (8) is integrated over each volume to obtain the finite volume form of the governing equation and using the divergence theorem yields Equation (3.11).

$$\frac{\partial}{\partial t}\left(u\Delta x\right) + F_{i+1/2} - F_{i-1/2} = 0 \quad (3.11)$$

Using the Godunov method, u is assumed piecewise constant within the i^{th} grid cell from location $x_{i-1/2}$ to $x_{i+1/2}$. Here, the solution of each Riemann problem under the conservation law is self-similar, which means the value of u for the Riemann problem is constant at the interface of location $x_{i+1/2}$ over the time interval t_n to t_{n+1}. If the flux only depends on u, then F is also constant on the interface of adjacent cells. Consequently, $F(x_{i-1/2},t)$ on the boundary of $(i-1)^{th}$ and i^{th} cell is constant within time interval t_n to t_{n+1}, and the constant flux only depends on the value of u on the interface [38]. Thereafter, Equation (3.11)can be transformed to

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n - F_{i-1/2}^n\right) \tag{3.12}$$

Under the scalar conservation law, the flux function has the following Riemann solution [44].

$$F_{i+1/2}^n = \begin{cases} \min\limits_{u_i^n \le u \le u_{i+1}^n} f(u) & if\ u_i^n \le u_{i+1}^n \\ \max\limits_{u_{i+1}^n \le u \le u_i^n} f(u) & if\ u_i^n > u_{i+1}^n \end{cases} \tag{3.13}$$

Here, *f(u)* is a convex function for the region of zero velocity to free-flow speed, as illustrated in Figure 3.1. The value of the extreme point is computed as u_0 in Equation (3.14), by setting the derivative of the flux function equal to zero. This extreme value equals the speed-at-capacity as demonstrated in Equation (3.14). This is because the speed-at-capacity corresponds to the speed at the maximum flow, where the traffic stream flow represents the flux considering density as the state variable.

$$u_0 = u_f + \frac{c_2}{c_1} - \frac{c_2}{c_1}\sqrt{\frac{c_1}{c_2}u_f + 1} = u_c \tag{3.14}$$

Knowing the function *f(u)* and the extreme point of u_0, the solution of the flux function can be simplified to Equation (3.15).

$$F_{i+1/2}^n = \begin{cases} f(u_{i+1}^n) & if\ u_i^n \le u_{i+1}^n \le u_0 \\ f(u_0) & if\ u_i^n \le u_0 \le u_{i+1}^n \\ f(u_i^n) & if\ u_0 \le u_i^n \le u_{i+1}^n \\ \max\left(f(u_i^n), f(u_{i+1}^n)\right) & if\ u_i^n \ge u_{i+1}^n \end{cases} \tag{3.15}$$

In order to have a unique solution to Equation (8) boundary conditions must be specified [45]. In this study, non-reflective boundary conditions are applied as

$$\left.\frac{\partial u}{\partial x}\right|_{x=0} = 0 \quad \text{and} \quad \left.\frac{\partial u}{\partial x}\right|_{x=x_{\max}} = 0 \tag{3.16}$$

In order to maintain numerical stability, the Courant number must satisfy the following condition:

$$\left|\frac{\Delta t}{\Delta x}\max\left(f'(u)\right)\right| \le 1 \tag{3.17}$$

After implementing the Godunov scheme for the Van Aerde flow continuity model, numerical solution of the PDE is obtained to characterize the spatio-temporal evolution of traffic flow velocity that can be used in the following sections.

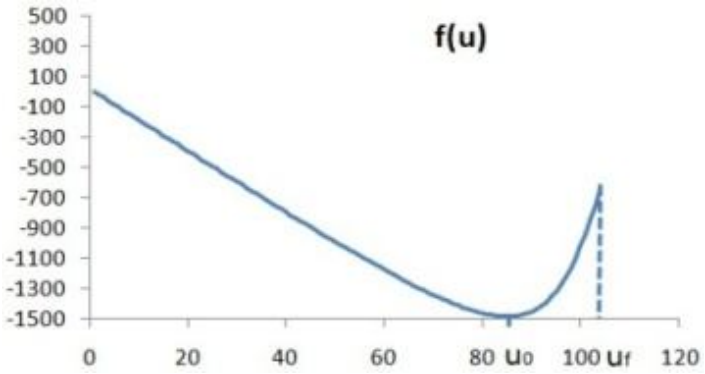


Figure 3.1: The Curve of Flux Function.

(u_f=104 km/h, u_c=85 km/h, q_c=2400 veh/h, k_j=150 veh/km)

3.2.3 State update model

A discrete time series equation can be written that uses the estimated traffic stream speed. The time update process in the recursive Bayesian filter framework can be cast as,

$$x^{n+1} = G\left(x^n\right) + \varphi^n \tag{3.18}$$

where $x^n=[u_1^n, u_2^n, ..., u_N^n]^T$ represents all speed state variables in the traffic network at the n^{th} time interval; the process noise is defined by $\varphi^n=[\varphi_1^n, \varphi_2^n, ..., \varphi_N^n]^T$; and the nonlinear function $G(x)$ is described in Equation (3.12).

The freeway speed measurements are needed to adjust the predicted value and then state estimation data can be obtained. As mentioned, many traffic data measurement devices can be used to measure traffic stream speeds. Loop detector data are used in this study. Assume loop detectors are installed on the boundaries of adjacent freeway sections, the measurement update process can be written as follows,

$$y^n = H^n\left(x^n\right) + \gamma^n \tag{3.19}$$

where y^n includes all available observation speed data from loop detectors at time t_n; H^n is a diagonal matrix, in which the Eigen value is one if the corresponding freeway section has observation data at the n^{th} time interval, otherwise it is zero; and γ represents the measurement noise vector.

3.3 Particle filter approach for state prediction

Bayesian filters represent a general probabilistic approach to estimate an unknown probability density function (pdf) recursively over time, by using a mathematical time

series model and incoming measurement data. The conditional density $p(x^{n+1}|y^{1:n})$ is recursively updated according to Equation (3.20) and (3.21) as below [21].

$$p\left(x^{n+1} \mid y^{1:n}\right) = \int p\left(x^{n+1} \mid x^{n}\right) p\left(x^{n} \mid y^{1:n}\right) dx^{n} \tag{3.20}$$

$$p\left(x^{n} \mid y^{1:n}\right) = \frac{p\left(y^{n} \mid x^{n}\right) p\left(x^{n} \mid y^{1:n-1}\right)}{p\left(y^{n} \mid y^{1:n-1}\right)} \tag{3.21}$$

Where $p(x^n|x^{n-1})$ is the probability of system evolution, given by the time update process of Equation (3.18) and $p(y^n|x^n)$ is defined by the measurement update process of Equation (3.19). However, the analytical solution of $p(x^n|y^{1:n})$ is very hard to directly compute [46]. The approach of particle filter can be used here to estimate the posterior pdf of $p(x^n|y^{1:n})$, which is represented by a set of random samples with corresponding weights. When the number of samples is large enough to approach infinity, these particles approximate the equivalent representation of the posterior pdf [46]. Suppose $x^n\{x^n_{(j)}, w^n_{(j)}\}$ denotes a collection of m particles, in which $x^n_{(j)}$ is the state value and $w^n_{(j)}$ is the corresponding weight of the j^{th} particle at time t_n. The summation of all weights is unity. The posterior pdf can be approximated as Equation (3.22), and the weights update is defined in Equation (3.23).

$$p(x^n \mid y^{1:n}) \approx \sum_{j=1}^{m} w^n_{(j)} \delta(x^n - x^n_{(j)}) \tag{3.22}$$

$$w^n_{(j)} \propto w^{n-1}_{(j)} \cdot \frac{p(y^n \mid x^n_{(j)}) p(x^n_{(j)} \mid x^{n-1}_{(j)})}{\pi(x^n_{(j)} \mid x^{n-1}_{(j)}, y^n)} \tag{3.23}$$

Where $\pi(x^n_{(j)}|x^{n-1}_{(j)}, y^n)$ is the importance density, which is a known pdf chosen to generate the particles. It is often convenient to set the importance density to be the same as the prior pdf $p(x^n_{(j)}|x^{n-1}_{(j)})$, so weight updating in Equation (3.23) is simplified to Equation (3.24) [47]. Following this approximation, the original integral calculation of Equation (3.20) is transformed to an easier formulation of computing the summation of particles with corresponding weights.

$$w^n_{(j)} \propto w^{n-1}_{(j)} \cdot p(y^n \mid x^n_{(j)}) \tag{3.24}$$

After the arrival of new measurement data, weights are updated by considering the importance of corresponding particles. Within the likelihood calculation, the smaller error between prediction and measurement data results in assigning a particle a larger weight. For the application of traffic state prediction in this paper, the state variable is an $n \times 1$ vector containing the speed values of n freeway sections. However, the average errors for different sections may not be of similar magnitude. In this case, the likelihood calculation using an uneven level of speed errors will result in incorrect weight updates. For solving this problem, the average error vector, which is calculated based on all particles of each

measurement vector, is eliminated in the likelihood calculation. Consequently, the particles of each section are adjusted and then can be weighted within the same level.

The degeneracy phenomenon is another serious problem for weight updating, which results in all but one particle having negligible weights after several iterations. Re-sampling is a good method to deal with this problem as described in [47]. At this time, the value of each particle and the corresponding updated weights can be applied to the time update equation (3.18) multiple times to predict state values in the future.

In the application of this paper, Δt, Δt_m represent the time interval of state update and measurement update respectively. Δt_p denotes the prediction time span, which means the forecast time length for future traffic state. Those time variables have the relationship that $\Delta t_m/\Delta t=n_m$ and $\Delta t_p/\Delta t_m=n_p$. Normally Δt is the least one to ensure the accuracy of state propagate in Equation (3.18). Measurement time step is decided by on-field device, usually 1 minute for loop detector. The prediction time span is much longer than the other two values, since long term traffic state prediction has lots of benefits for the application of ramp metering or traveler information broadcasting. To summarize, the implementation of the proposed particle filter prediction approach is described as follows. Here, n and k represent the time index of state and measurement update index, respectively.

1) Initialization:
   Set $n=0$, $k=1$;
   For $j=1,2,\cdots,m$, generate m samples $x_{(j)}^0$ with the weights $w_{(j)}^0$ according to the initial distribution $p(x^0)$.
2) Time update:
   $$x_{(j)}^{n+1} = G\left(x_{(j)}^n\right) + \varphi_{(j)}^n$$
   Run the above time update calculation for n_m times.
3) Measurement update:
   Measurement data is obtained at time $k\times\Delta t_m$;
   At this time, $x_{(j)}^k = x_{(j)}^{\mathrm{nm}}$;
   a) Calculate average measurement error:
   $$e_{mean}^k = \frac{1}{m}\sum_{j=1}^{m}\left[y_{(j)}^k - H^k\left(x_{(j)}^k\right)\right]$$
   b) Weight update:
   Update the weights by likelihood function p_e [48]:
   $$w_{(j)}^k = w_{(j)}^{k-1} p_e\left[y_{(j)}^k - H^k\left(x_{(j)}^k\right) - e_{mean}^k\right]$$
   Normalize the weights.
   c) Resampling:
   Only resample the particles for the condition:
   $$S_e = \left(\sum_j \left(w_{(j)}^k\right)^2\right)^{-1} < S_{th},$$
   where S_{th} is equal to $m/2$.

The resampling method of [25] is used to avoid the significant particle degeneracy.

4) Prediction output:
Run step 2 for n_p times, so $n= n_m \times (n_p+1)$ right now;
Output the prediction result for time $(k+n_p) \times \Delta t_m$:
$$x\_pre = \sum_{j=1}^{m} x_{(j)}^{n} \cdot w_{(j)}^{k}$$

5) Set n=0, $x_{(j)}^{0} = x_{(j)}^{\mathrm{nm}}$, increase k and then return to step 2.

3.4 Model testing

3.4.1 Simulation setup

The INTEGRATION microscopic traffic simulation software is used to simulate the freeway traffic state data. A detailed description of the software is provided in [49].

The freeway under study is a stretch of Interstate 66 as shown in Figure 4.5, which starts from Manassas (Virginia) west towards the direction of Vienna east, between Exit 47 and Exit 52 of I66. The stretch has the total length of 8 km (5 miles), and contains 10 sections with the distance interval of 0.8 km (half mile). Eleven loop detectors are used to collect freeway speed data and they are located on the boundaries of each section as shown in Figure 4.5. Since I-66 is a main highway entering Washington, D.C. from west, the traffic volume of study area is usually heavy from 8:00 am to 3:30 pm. Also, since a lot of traffic leaves I-66 by Exit 52, a bottleneck is usually formed on the last one third part of the stretch, which is between 8th and 10th loop detector. The O-D demands of traffic network used in INTEGRATION simulation has been calibrated based on the previous studies by QueensOD software [50]. The time dependent O-D demands from 6:00 am to 9:00 am on May 3rd, 2002 is used for model testing. After calibrating the model using field data, basic traffic parameters u_f , u_c, q_c and k_j are 104 km/h, 85 km/h, 2400 veh/hour and 150 veh/km/lane, respectively. For the testing, the state update time interval Δt was set at 10 seconds and the measurement interval Δt_m was set at 60 seconds. Prediction rates of traffic speed for five minutes in the future (prediction time span Δt_p) are compared using four Bayesian filter approaches.

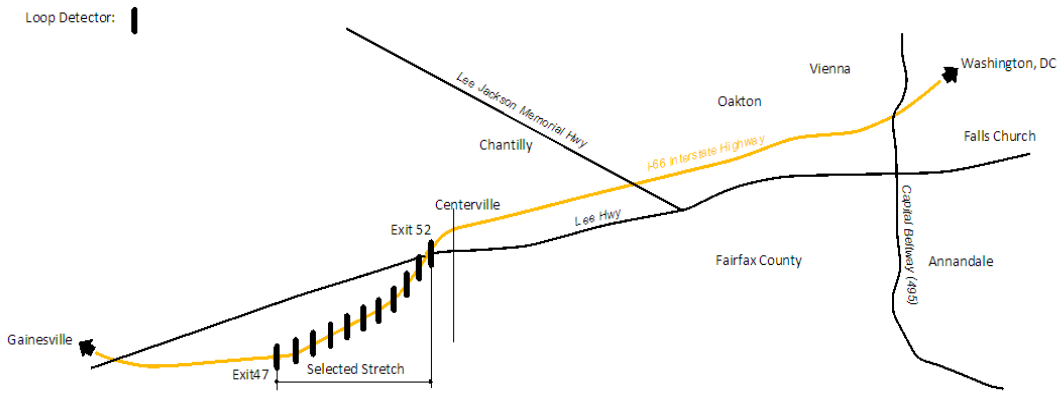


Figure 3.2: Sample I66 network configuration.

3.4.2 Prediction schemes and performance indices

In order to provide a better evaluation of the proposed prediction approach of the Van Aerde flow continuity model and particle filter, the Greenshields based CTM-V model and Ensemble Kalman filter [51] are used to generate four combinations of prediction approaches, and compared to identify the best prediction results. The same free flow speed 104 km/h is used in the CTM-V model. In all approaches a total of 100 particles using Gaussian noise are used for the time and measurement update. These alternative prediction approaches are presented below:

1) *Approach "GS+EnKF"*: Greenshields based CTM-V model and Ensemble Kalman filter

2) *Approach "VA+EnKF"*: Van Aerde flow continuity model and Ensemble Kalman filter

3) *Approach "GS+PF"*: Greenshields based CTM-V model and Particle filter

4) *Approach "VA+PF"* (Proposed): Van Aerde flow continuity model and Particle filter

Both relative and absolute prediction errors are calculated to evaluate the prediction accuracy of different approaches. The relative error is the Mean Absolute Percentage Error (MAPE) as Equation (3.25), and the corresponding absolute error is presented by Mean Absolute Deviation (MAD) as Equation (3.26).

$$MAPE = \frac{100}{KN}\sum_{k=1}^{K}\sum_{i=1}^{N}\frac{\left|y_i^k - \hat{y}_i^k\right|}{y_i^k} \tag{3.25}$$

$$MAD = \frac{1}{KN}\sum_{k=1}^{K}\sum_{i=1}^{N}\left|y_i^k - \hat{y}_i^k\right| \tag{3.26}$$

Where K is the total number of measurement update intervals; N is the number of sections in the traffic stretch; $y^k{}_i$ denotes the measurement average speed and $\hat{y}_i^k$ denotes the predicted speed obtained by Bayesian filter prediction approach at the k^{th} time interval on the i^{th} freeway section.

3.4.3 Test results

The five-minute freeway speed prediction results for the four approaches are summarized in Table 3.1. The proposed approach has the lowest absolute and relative errors of 4.34 km/h and 19.28%, respectively. At freeway section 8, shown in Figure 3.3(a), speed values after passage of the shock wave are close to zero (vehicles are almost stopped), hence small deviations in the speed estimates result in significant relative errors and thus the relatively high MAPE values. According to the plot in Figure 3.3(a), the predicted speed curve over time shows that "VA+PF" approach can produce a very accurate representation of the field data, on the other hand, a significant error is observed for congested traffic conditions using the "GS+PF" approach. By comparing graph (c) and (d) to (b) in Figure 3.3,we can clearly see the predicted shockwave in (c) moves from location 7.5 km at 7:00 am to location 6.0 km at 8:30 am, which is consistent with the field observations. Another interesting fact from Table 1 is that the use of the Van Aerde functional form provides larger improvements compared to the use of the particle filter. The combination of the Van Aerde and particle filter approach results in minimum prediction error.

Table 3.1: Five minutes speed prediction results by four approaches.

Approaches	Five minutes speed prediction	
	MAD(km/h)	MAPE(%)
GS+EnKF	9.36	65.13
VA+EnKF	6.42	31.47
GS+PF	7.02	53.41
VA+PF(proposed)	4.34	19.28

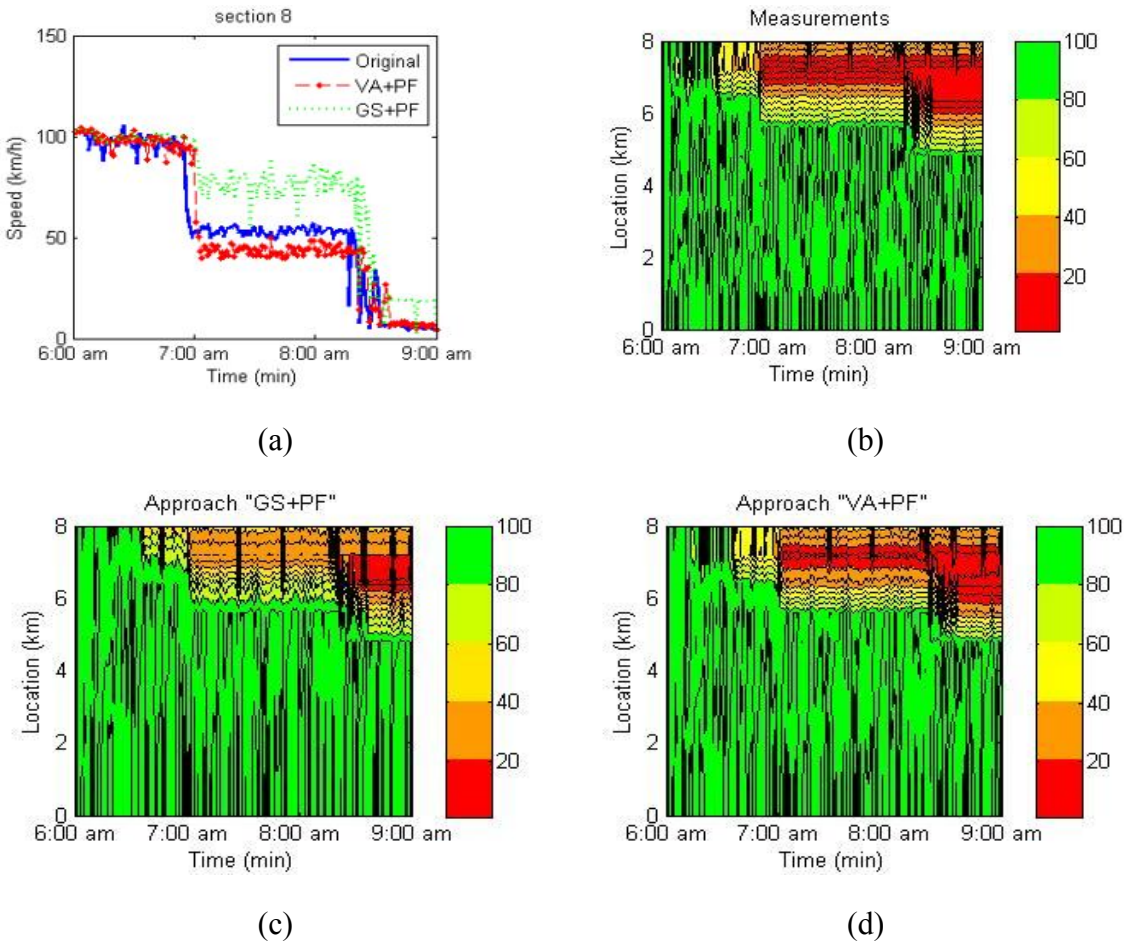


Figure 3.3: Prediction results by different approaches; (a) Temporal speed variation for freeway section 8; (b) Time-space speed data by loop detector; (c) Predicted time-space speed contour by "GS+PF"; (d) Predicted time-space speed contour by "VA+PF".

3.5 Conclusions

This chapter develops a particle filter approach that predicts the traffic stream state using a new time series speed equation. A more realistic traffic stream model - Van Aerde, is used to derive a new Van Aerde flow continuity model via a non-conservative LWR equation. The corresponding numerical solution is obtained using a cell-centered finite volume Godunov scheme. The speed time-evolution equation is used in the proposed particle filter approach together with the available measured speed data to conduct a multi-step freeway speed prediction. Using simulated data from I-66, model test results show that the proposed approach produces the least error and accurately predicts the propagation of shockwaves. In future research, the advantage of particle filter, in which each particle has a prediction value and an associated weight, will be used to predict traffic state reliability information. Implementation of a shock fitting scheme will be investigated to overcome the fact that a non-conservative form of LWR equation is used

to derive the Van-Aerde continuity model. In addition, historical measurement data will be combined with the proposed approach to conduct long-term data prediction (e.g. 30 min.).

Chapter 4

4 Macroscopic Traffic Modeling with Ramp Adjustment and Realistic Boundary Conditions

This chapter is an edited version of: Hao Chen, Hesham A. Rakha, Shereef A. Sadek, and Bryan J. Katz (2012), "A Particle Filter Approach for Real-time Freeway Traffic State Prediction," in *91st Transportation Research Board Annual Meeting*, Washington D.C.

The research presented in this chapter improves the speed formulation of chapter 3 by incorporating ramp flows and enhancing the boundary conditions. The numerical solution and near-term prediction accuracy (5-minute prediction) of the new speed formulation is compared with the conservative density formulation derived from LWR. Although the proposed speed formulation is non-conservative and not equivalent to the solution of LWR under the same initial and boundary conditions, it produces significant enhancements in the traffic state predictions. Specifically, the prediction error using simulated I-66 data is in the range of 3.0 to 4.5 km/h for a 5-minute prediction horizon. This error is approximately half the prediction error of the LWR formulation. Similarly, the traffic stream density prediction error is approximately half that of the LWR formulation.

4.1 Introduction

In the previous chapter, a particle filter approach for solving the problems in state-of-art traffic state prediction is proposed [17]. This approach combines a more realistic traffic stream model – Van Aerde model with the traditional LWR flow continuity equation to derive a new time series equation to describe the spatial and temporal relationship of traffic speed on a freeway stretch. After implementing the speed equation in the particle filter framework, a multi-step prediction approach was proposed, tested and was demonstrated to produce good traffic state predictions. However, there are several problems with this approach. Firstly, it is a non-conservative form of LWR, hence not equivalent to LWR in the presence of discontinuities. Consequently, there is a need to investigate whether the use of this complex non-conservation equation is better than the use of the density formulation (conservative LWR formulation) when such discontinuities exist. Secondly, ramp flows were not considered in the previous formulation and thus need to be integrated into the framework.

The method proposed in this chapter can efficiently overcome the above problems. The effect of ramps within the freeway stretch is considered to improve the previous Van Aerde flow continuity model. After applying the same discretization, a time series traffic speed formulation is derived and implemented in a particle filter approach to predict future traffic state variables. The numerical solution and data prediction accuracy of the new speed formulation is compared with a density formulation directly derived from LWR. Although the proposed speed formulation is non-conservative and not equivalent to the solution of LWR under the same initial and boundary conditions, it produces much higher traffic state prediction accuracies using the particle filter framework on simulated I-66 data for different days.

The outline of the chapter is as follows. The model derivation and the corresponding numerical discretization are described in the next section. Following this, the framework of particle filter for data prediction is designed. Subsequently, the comparison of numerical solution for the proposed speed formulation and a density formulation is presented. This is followed by the application of the traffic speed prediction by two approaches on the simulated I-66 data. Finally, the Summary and the future research recommendations are provided in the last section.

4.2 Model description

The more realistic traffic stream model – Van Aerde model is chosen since it provides four degrees of freedom by allowing the speed-at-capacity to differ from the free-flow speed , as is the case in most models including the Pipes, Gipps, and Newell's simplified model, or half the free-flow speed, as is the case with the Greenshields model [37]. In this paper, the Van Aerde model is used to provide a unique relationship between traffic speed and density. By implementing this relationship to the partial differential equation of the traditional LWR, a new PDE which includes traffic speed as the only variable is derived [17]. However, the traffic flow on ramps was not considered in the previous research. According to conservation law, the change of ramp flow for the system should

be treated as the source term in the PDE. In this section, a detailed description of the derivation of the new model is presented.

4.2.1 Van Aerde flow continuity model with ramp adjustment

The flow continuity equation proposed in [24, 38]indicates that the continuous characteristics of highway traffic flow can be represented by one-dimensional wave motion theory. When the freeway section has no ramps, the formulation can be represented using the traditional LWR equation. However, if the freeway includes on- and off-ramps, according to the conservation law, the change of ramp flow should be represented as the source term on the hand right side of the PDE as Equation (4.1) [52].

$$\frac{\partial k}{\partial t}+\frac{\partial q}{\partial x}=\frac{\partial (r-s)}{\partial x}. \tag{4.1}$$

Here q and k are the traffic stream flow rate and density states, respectively, at any instant t and location x along the roadway; r is the on-ramp inflow; and s is the off-ramp outflow. In order to simplify the equation, the ramp flows are modeled as constant fractions of the mainstream flow volumes [18]. Assume the ratio flow rates between on-ramp flow and off-ramp flow to the mainstream flow are α and β, respectively, consequently Equation (4.1) can be modified as

$$\frac{\partial k}{\partial t}+\frac{\partial q\left(1-\alpha+\beta\right)}{\partial x}=0. \tag{4.2}$$

In this paper, the relationship between traffic stream density (k) and speed (u) follows the Van Aerde traffic stream model because of its comprehensiveness, simplicity and ease of calibration [37]. The model has four degrees of freedom and is formulated as

$$k=\left(c_1+\frac{c_2}{u_f-u}+c_3 u\right)^{-1}. \tag{4.3}$$

$$c_1=\frac{u_f}{k_j u_c^2}\left(2u_c-u_f\right);\quad c_2=\frac{u_f}{k_j u_c^2}\left(u_f-u_c\right)^2;\quad c_3=\frac{1}{q_c}-\frac{u_f}{k_j u_c^2}. \tag{4.4}$$

Here the free-flow speed (u_f), speed-at-capacity (u_c), capacity (q_c) and jam density (k_j) are four basic traffic stream parameters that require calibration. The c_1, c_2 and c_3 parameters can be formulated as a combination of these four parameters using Equation (4.4).

After using the same approach to replace traffic density k and flow by speed u using the Van Aerde traffic stream model as described in [17], the new partial differential equation with the consideration of ramp flow can be cast as

$$\frac{\partial u}{\partial t}+\left(1-\alpha+\beta\right)\frac{2c_2u-c_2u_f-c_1(u_f-u)^2}{c_2+c_3(u_f-u)^2}\cdot\frac{\partial u}{\partial x}=0\,. \tag{4.5}$$

In order to solve the above partial differential equation, Equation (4.5) should be transformed to a one-dimensional scalar conservation law of the form shown in Equation (4.6), under the condition that the velocity flux function (F) satisfies the relation of Equation (4.7).

$$\frac{\partial u}{\partial t}+\frac{\partial F}{\partial x}=0\,. \tag{4.6}$$

$$\frac{dq}{dk}=\frac{dF}{du}\,. \tag{4.7}$$

Integrating the above relation yields the following expression for F,

$$F=f(u)=\left(1-\alpha+\beta\right)\left\{-\frac{c_1}{c_3}u+\frac{c_2}{c_3}\ln\left[c_3\left(u_f-u\right)^2+c_2\right]-\sqrt{\frac{c_2}{c_3}}\left(\frac{c_1}{c_3}+u_f\right)\arctan\left[\sqrt{\frac{c_3}{c_2}}\left(u_f-u\right)\right]\right\}\,. \tag{4.8}$$

Equation (4.6) is the proposed Van Aerde flow continuity model. According to this equation, the partial differential of speed u over time plus the partial differential of the flux F over space is equal to zero. Here, the flux F is a function of speed, *f(u)*. Speed is the only variable in this flux function and other coefficients can be calculated using the four basic traffic stream parameters as was demonstrated earlier in Equation (4.4). Since the above mentioned formulation assumes a differentiable flux function, the proposed speed formulation is non-conservative and not equivalent to the LWR equation. When the freeway traffic experiences a sudden change in speed, this assumption cannot be satisfied and a moving shockwave or rarefaction will be generated. Consequently, the proposed speed formulation will predict the propagation shockwaves and/or rarefaction waves different from the LWR formulation. The differences will be presented and discussed in the model testing section later on.

4.2.2 Numerical discretization with realistic boundary condition

In order to use the proposed traffic speed formulation within a Bayesian filtering framework, a Godunov scheme is implemented to discretize the original equation over time and space [43]. The discretization process is almost identical to that described in an earlier publication [17]. The only difference is that the flux in this paper is multiplied by a coefficient *(1-α-β)* as presented in Equation (4.8), which accounts for the effects of on-ramp and off-ramp flows.

A freeway stretch of length L, can be discretized into N control volumes of equal length ($\Delta x=L/N$). In this way, u^n_i or $u(x_i,t_n)$ represents the speed of the traffic stream on i^{th} control

volume at time $n\Delta t$. Equation (4.8) is integrated over each volume to obtain the discrete form of the governing equation and using the divergence theorem yields

$$\frac{\partial}{\partial t}\left(u\Delta x\right)+F_{i+1/2}-F_{i-1/2}=0\,. \tag{4.9}$$

Using the Godunov method, u is assumed piecewise constant within the i^{th} grid cell from location $x_{i-1/2}$ to $x_{i+1/2}$. Here, the solution of each Riemann problem under the conservation law is self-similar, which means the value of u for the Riemann problem is constant at the interface of location $x_{i+1/2}$ over the time interval t_n to t_{n+1}. If the flux only depends on u, then F is also constant on the interface of adjacent cells. Consequently, $F(x_{i-1/2},t)$ on the boundary of $(i-1)^{th}$ and i^{th} cell is constant within the time interval t_n to t_{n+1}, and the constant flux only depends on the value of u at the interface [38]. Thereafter, Equation (4.9) can be transformed to Equation (4.10). Under the scalar conservation law, the flux function is evaluated using Equation (4.11).

$$u_i^{n+1}=u_i^n-\frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n-F_{i-1/2}^n\right). \tag{4.10}$$

$$F_{i+1/2}^n=\begin{cases} f(u_{i+1}^n) & if\ \ u_i^n\le u_{i+1}^n\le u_c \\ f(u_c) & if\ \ u_i^n\le u_c\le u_{i+1}^n \\ f(u_i^n) & if\ \ u_c\le u_i^n\le u_{i+1}^n \\ \max\left(f(u_i^n),f(u_{i+1}^n)\right) & if\ \ u_i^n\ge u_{i+1}^n \end{cases}. \tag{4.11}$$

In order to obtain a unique solution to Equation (8) boundary conditions must be specified. A different boundary condition is used in this paper with the advantage of considering the characteristics may be entering or leaving the domain at both the left and right boundaries, instead of simple non-reflect boundary condition from [17]. The direction of the characteristic should be calculated in order to set the correct boundary flux value. Assume c represents the derivative of q with respect to k ($c=dq/dk$), which is the characteristics speed. In this paper we enhance the boundary conditions in **[17]** by expressing them as formulated in Equation (4.12).

$$\begin{aligned} &\text{at } x=0 \text{ (left-hand boundary)} \\ &\qquad \text{if } c(u)>0 \quad u_{-1/2}^n=u_0^n \\ &\qquad \text{if } c(u)\le 0 \quad u_{N+1/2}^n=u\left(1,t_n\right) \\ &\text{at } x=L \text{ (right-hand boundary)} \\ &\qquad \text{if } c(u)>0 \quad u_{N+1/2}^n=u_N^n \\ &\qquad \text{if } c(u)\le 0 \quad u_{N+1/2}^n=u\left(L,t_n\right) \end{aligned}. \tag{4.12}$$

In order to maintain numerical stability, the Courant number must satisfy the following condition:

$$\left|\frac{\Delta t}{\Delta x}\max\left(f'(u)\right)\right| \le 1. \tag{4.13}$$

4.2.3 State update model

The numerical solution of the proposed Van Aerde flow continuity model is obtained to model the spatiotemporal evolution of traffic flow velocity that can be used in the time update process in the recursive Bayesian filtering framework. Given the freeway velocity of all sections on the traffic network at time $n\Delta t$, the network velocity at time $(n+1)\Delta t$ is constructed as

$$x^{n+1} = G\left(x^n\right) + \varphi^n. \tag{4.14}$$

Where, $x^n=[u_1^n, u_2^n, ..., u_N^n]^T$ represents all speed state variables in the traffic network at the n^{th} time interval; the prediction error is defined by $\varphi^n=[\varphi_1^n, \varphi_2^n, ..., \varphi_N^n]^T$; and the nonlinear function $G(x)$ is described in Equation (4.10).

Once the freeway speed measurement data are available, they are used to adjust the predicted traffic state for the same time interval, in order to adjust the error and prepare for the prediction in the next iteration. Traffic stream speed can be measured using different technologies such as loop detectors, remote traffic microwave sensors, Bluetooth detectors, license tag readers and probe vehicles. Loop detector data are used in this study. Assume detectors are installed on the boundaries of adjacent freeway sections, the measurement update process can be written as

$$y^n = H^n\left(x^n\right) + \gamma^n. \tag{4.15}$$

Here y^n includes all available observation speed data from detectors at time t_n; H^n is a diagonal matrix, in which the eigen value is one if the corresponding freeway section has observation data at the n^{th} time interval, otherwise it is zero; γ represents the measurement error vector.

4.3 Particle filter approach for state prediction

The general introduction of particle filtering approach has been provided in chapter 3.3. To summarize, the proposed particle filter prediction approach is demonstrated in Figure 4.1, in which n and k represent the time index of state and measurement update, respectively.

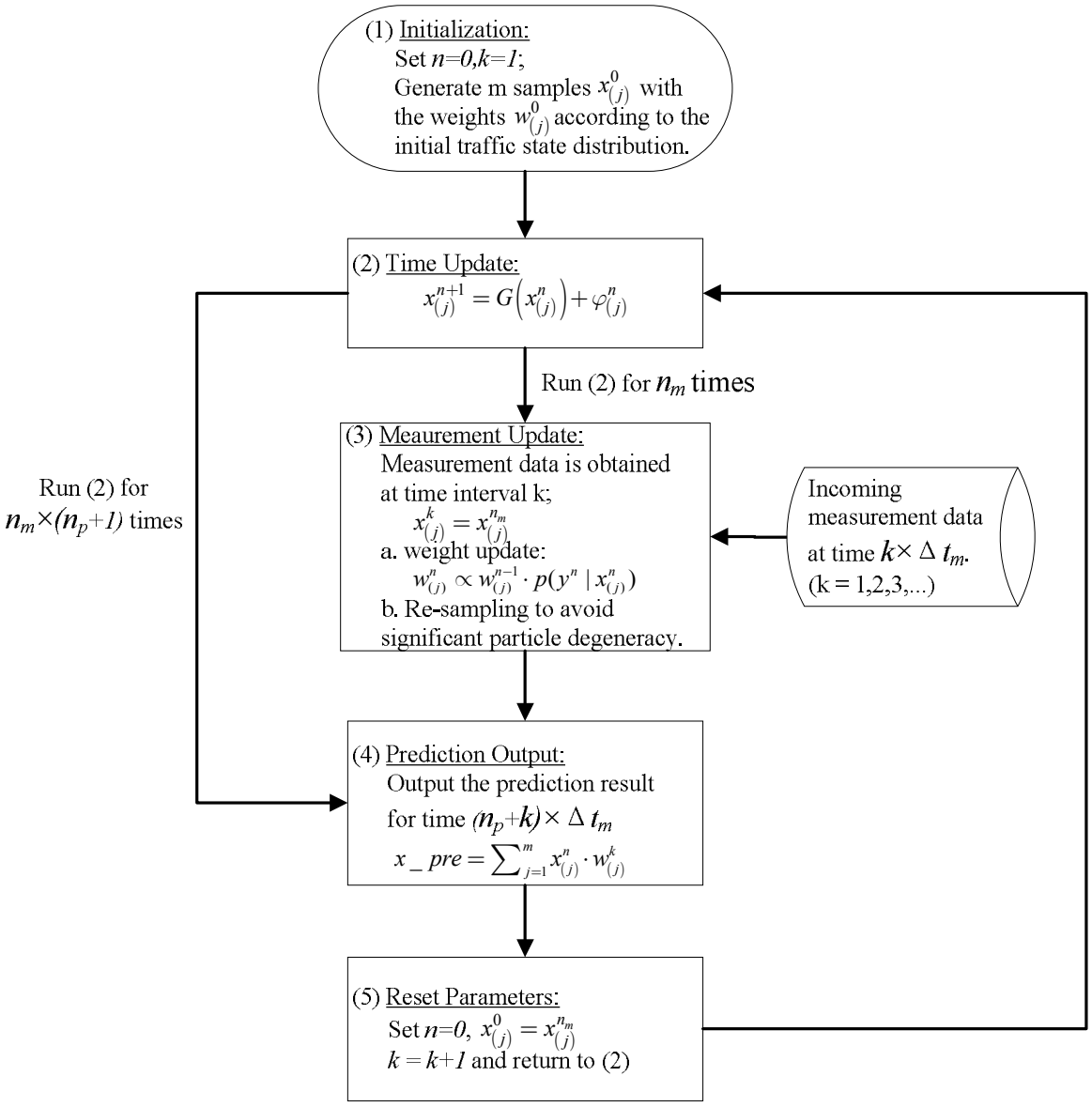


Figure 4.1: The flow chart of proposed particle filter approach for traffic state prediction.

4.4 Model testing

In this section, results from the proposed speed formulation are compared with those obtained from the LWR conservative density formulation. Two test cases are considered; the first is a moving shockwave and the second is a moving bottleneck. We are testing the formulation for these cases given that the conservative assumption is violated in these cases. Next, the convergence property of the particle filter approach is tested to prove the accuracy of filtering estimation to the measurement. Finally, both formulations are

implemented with the particle filter approach to test the accuracy of the traffic state prediction on simulated Interstate 66 data.

4.4.1 Comparison of numerical solution

The numerical solution of the proposed speed formulation is given by Equation (4.10) and (4.11),while the numerical solution of the LWR conservative density formulation is given by Equations (4.16) and (4.17).

$$k_i^{n+1} = k_i^n - \frac{\Delta t}{\Delta x}\left(Q_{i+1/2}^n - Q_{i-1/2}^n\right). \tag{4.16}$$

$$Q_{i+1/2}^n = \begin{cases} q(k_{i+1}^n) & if\ k_c \le k_{i+1}^n \le k_i^n \\ q(k_c) & if\ k_{i+1}^n \le k_c \le k_i^n \\ q(k_i^n) & if\ k_{i+1}^n \le k_i^n \le k_c \\ \min\left(q(k_i^n), q(k_{i+1}^n)\right) & if\ k_i^n \le k_{i+1}^n \end{cases}. \tag{4.17}$$

Here k_c is the density at capacity; Q is the flux, or traffic flow at the boundary. The function $q(k)$ represents the traffic flow according to the Van Aerde fundamental diagram when the roadway density is k. If ramps are included in the network, the same treatment of flux, which is traffic flow in the density formulation, multiplied by coefficient $1\text{-}\alpha+\beta$ will be implemented based on Equation (4.2). In our model testing, both the input and output data are traffic speed. Consequently, when the density formulation is implemented, the speed data should be firstly transferred to the corresponding traffic stream density according to the Van Aerde model of Equation (4.3). After computing the traffic stream using Equation (4.16), the predicted density will be transferred back to the speed data as the output. Since the speed formulation assumes a differentiable flux, two cases with the sudden drop and increase of freeway speed are simulated and the calculations of future speed using the two formulations are compared.

Case I: Shockwave Moving Backward

The purpose of this test case is the simulation of the propagation of a shockwave induced by an incident that blocks the entire roadway 5km away from the upstream boundary of the domain. The total length of the roadway is 8km, evenly divided into 40 sections. The state update interval is 5 seconds. The u_f, u_c, q_c and k_j are 104 km/h, 85 km/h, 2400 veh/hour and 150 veh/km, respectively. These parameters are typical for a freeway section. The initial traffic speed condition is shown as Figure 4.2(a), the speed is 104 km/h before the location of 5km, then jumps to zero after this location. This triggers a shockwave moving backward on the roadway. Figure 4.2(a)-(c) shows that the propagation of the shockwave computed by the speed formulation is slower than that for the density formulation, and the difference between the two speed curves increases as the simulation continues from 250s to 1000s. Consequently, the error in the speed formulation increases as time passes.

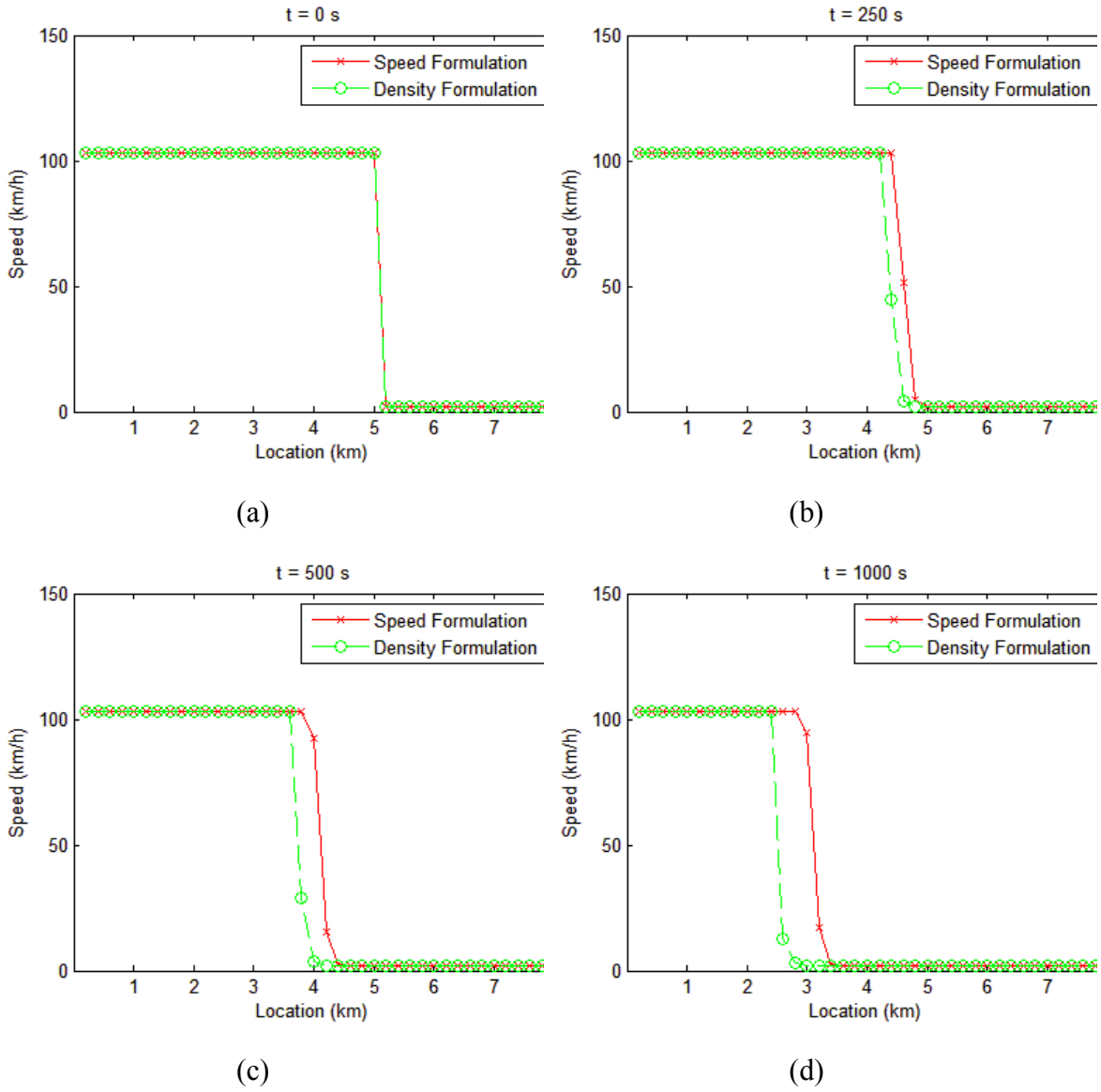


Figure 4.2: Numerical solutions for shockwave moving backward.

Case II: Moving Bottleneck

In this test case the traffic and roadway parameters are identical to those in case I. The initial traffic speed condition, shown in Figure 4.3 (a) starts with a constant traffic stream speed of 104 km/h that decreases to 45 km/h from 1.0 to 2.5 km(at the bottleneck) and then gradually increases to 104 km/h at 4.0 km following a half-cosine function shape. Upstream of the minimum speed location a forward moving shockwave is developed. On the other hand, downstream of the minimum speed location the flow expands continuously. According to Figure 4.3 (b)-(c), the predicted speed profile using the speed formulation still propagates at a slower speed downstream. The difference between the

two predicted curves from the two formulations also increases as the simulation progresses from 50 s to 250 s.

These two cases demonstrate that the speed formulation produces errors in the modeling of the spatio-temporal evolution of traffic when a discontinuity is introduced into the traffic stream.

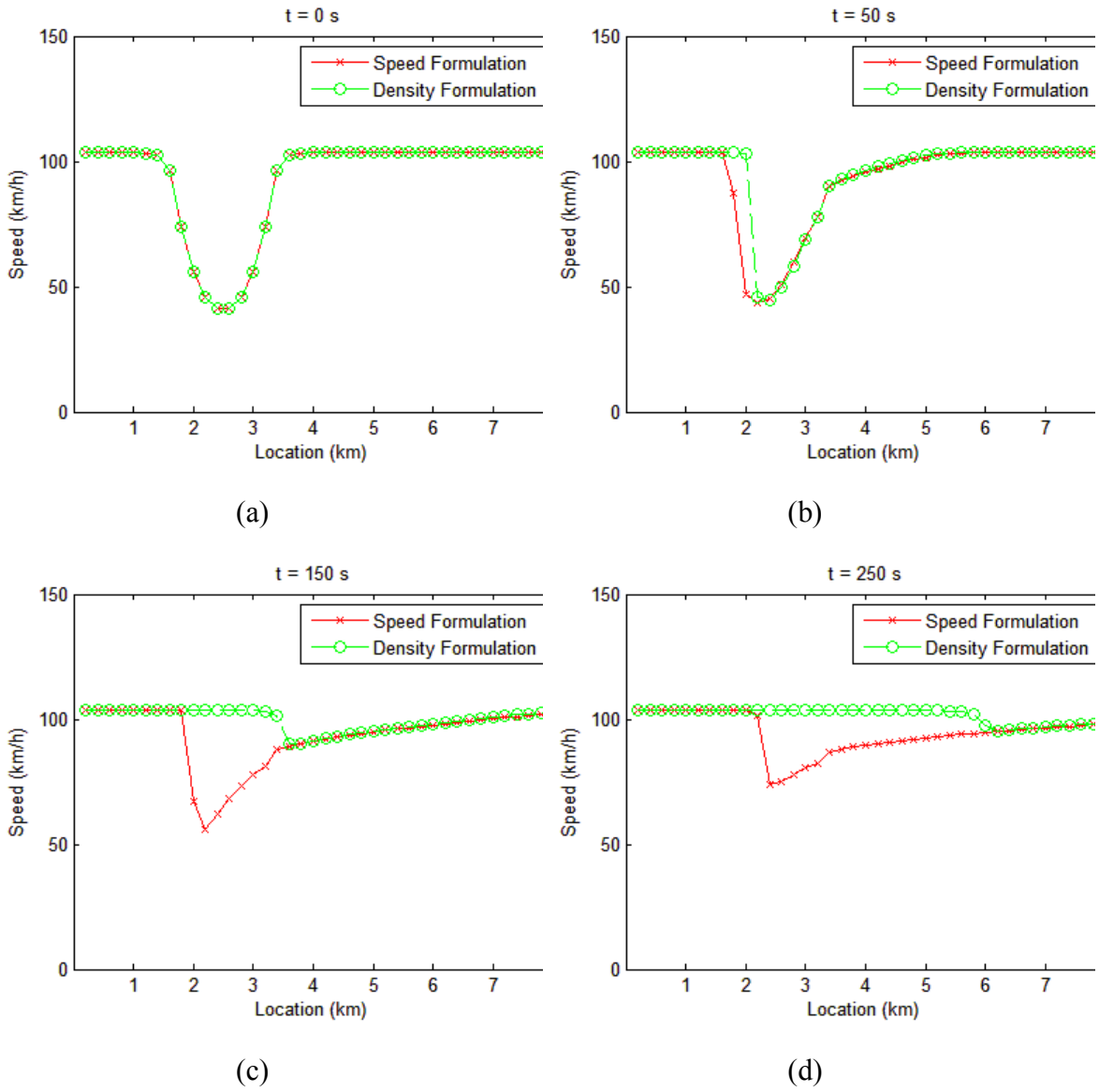


Figure 4.3: Numerical solutions for rarefaction waves

4.4.2 Test of convergence property

A simple corridor is used to test the convergence property of the particle filter. The link between location 1km to 1.5km is one-lane freeway, and all the rests are two-lane

freeway. Three loop detectors are located on the two boundaries and the center of the corridor. The incoming traffic flows are presented in Figure 4.5(a) as well. The basic parameters u_f, u_c, q_c, and k_j are 104km/h, 85km/h, 2400veh/km/lane and 150veh/km/lane respectively. Both of time update and measurement equations include Gaussian noises of variance 2. The traffic speed on every 0.1km intervals are estimated based on the speed formulation and particle filter, so there are three actual detectors and seven virtual detectors. Both of time and measurement update intervals are 5 seconds. Figure 4.5(b) presents the comparison between estimation value on the location 0.8km from virtual detector and the measurement data. It shows the estimate value on virtual detector converges to the measurement data very well. The total average absolute estimation error is 2.89 km/h.

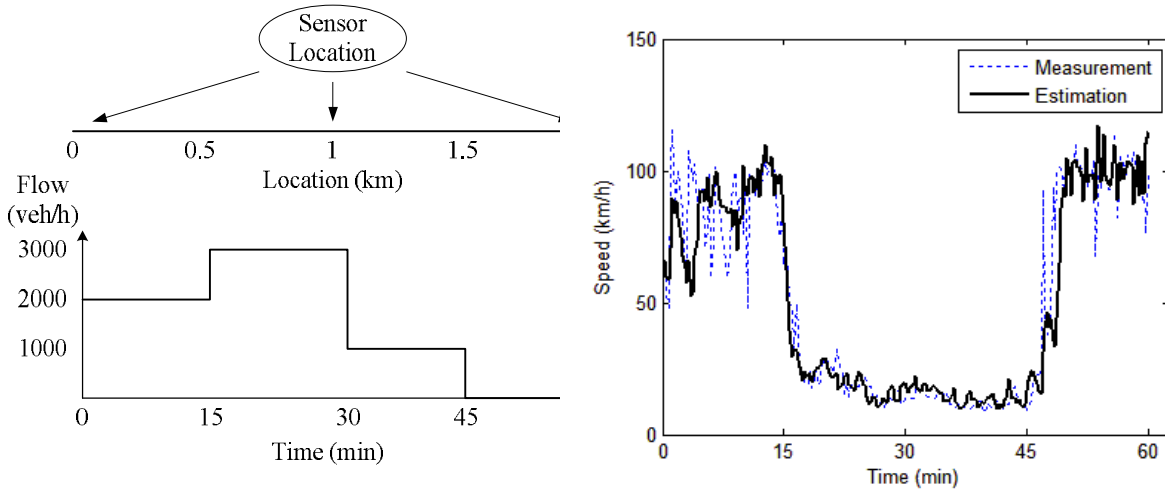


Figure 4.4: Convergence testing on a simple corridor case; (a) corridor layout and incoming traffic flows; (b) traffic speed estimation results on location 0.8km.

4.4.3 Implementation of state prediction on I-66

Base on the two simple examples that were presented earlier we concluded that the proposed speed formulation does produce errors when discontinuities are present. The next step is to test the two formulations using a realistic network that has on- and off-ramps in which a shockwave propagates. We combine the two formulations with the particle filter to predict the traffic state into the future.

Simulation Setup

A microscopic traffic simulation software, INTEGRATION, which was developed over the past two decades [49, 53-55]is used to simulate the freeway traffic state data. The INTEGRATION model is a trip-based microscopic traffic assignment, simulation and optimization model. The INTEGRATION model updates vehicle speeds every deci-second based on a user-specified, steady-state speed-spacing relationship and the speed differential between the subject vehicle and the vehicle immediately ahead of it. In order

to ensure realistic vehicle accelerations, the model uses a vehicle dynamics model that estimates the maximum and typical vehicle acceleration levels. Specifically, the model utilizes a variable power vehicle dynamics model to estimate the vehicle's tractive force that implicitly accounts for gear-shifting on vehicle acceleration. This model is described in more detail in the literature [56, 57].The model allows detailed analyses of lane-changing movements and shock wave propagations. It also permits considerable flexibility in representing spatial and temporal variations in traffic conditions. In addition to estimating vehicle stops and delays [58-60], the model can also estimate the fuel consumed by individual vehicles and the emissions [61], [62-64]. Finally, the model also estimates the expected number of vehicle crashes using a time-based crash prediction model [65].

The INTEGRATION model has not only been validated against standard traffic flow theory [37, 59, 60, 66] but also has been utilized for the evaluation of real-life applications [67-69]. For example, the INTEGRATION software has been demonstrated to model traffic dispersion through two procedures: a) introduction of randomness in driver/vehicle speeds or b) the use of stochastic car-following models. In the case of a stochastic car-following model, each driver has a unique random realization of a roadway specific steady-state car-following model with an average that is consistent with the mean driver-behavior. Several studies [60, 70, 71] showed that INTEGRATION's outputs are consistent with both observed field data and fundamental traffic flow theory. In addition, a study involved a basic validation of the traffic dispersion module using two sample field datasets [72]. The first dataset was gathered by Denney [73] in Houston, Texas and the second dataset was gathered by Castle and Bonniville [74] in Kuwait City, Kuwait. The Houston data contained the observed average flow profiles at an upstream signal and at a checkpoint 300 m (990 ft) downstream on a 3-lane arterial roadway. The observed average speed and standard deviation of speeds were reported as 48.3 km/h (44 ft/s) and 5.9 km/h (5.4 ft/s), respectively, which results in an observed speed coefficient of variation (CVobs) of 12.3%. Given that vehicle interactions may reduce vehicle speed variability, a slightly higher value of CV was input to the simulation software (CVin = 15%). The network was modeled as a 300-meter three-lane link, with a free-flow speed of 50 km/h, a speed-at-capacity of 40 km/h, a saturation flow of 1800 veh/h/lane, and jam density of 100 veh/km/lane. In the case of the Houston dataset, the simulated average speed and standard deviation were 48.2 km/h and 6.1 km/h (CVout=12.7%), which is very similar to the field-observed parameters. Consequently, the results demonstrated a high degree of consistency between simulated and field observed data in terms of aggregate trip measures (trip mean and variance) and in the progression of vehicles within platoons.

The simulated freeway is a stretch of Interstate 66 as shown in Figure 4.5(a), which starts from Manassas (Virginia) west towards the direction of Vienna east, between Exit 47and Exit 52 of I66.The stretch is8 kilometers (5 miles) long, and is divided into 10 sections with a distance interval of 0.8 kilometers (half mile). Loop detectors are installed in the simulation to measure traffic speed at different locations. In addition, loop detectors are located at the boundaries of each section as shown in Figure 4.5(a). The O-D demands of the traffic network used in INTEGRATION simulation were calibrated based on a previous study using the QueensOD software [75].

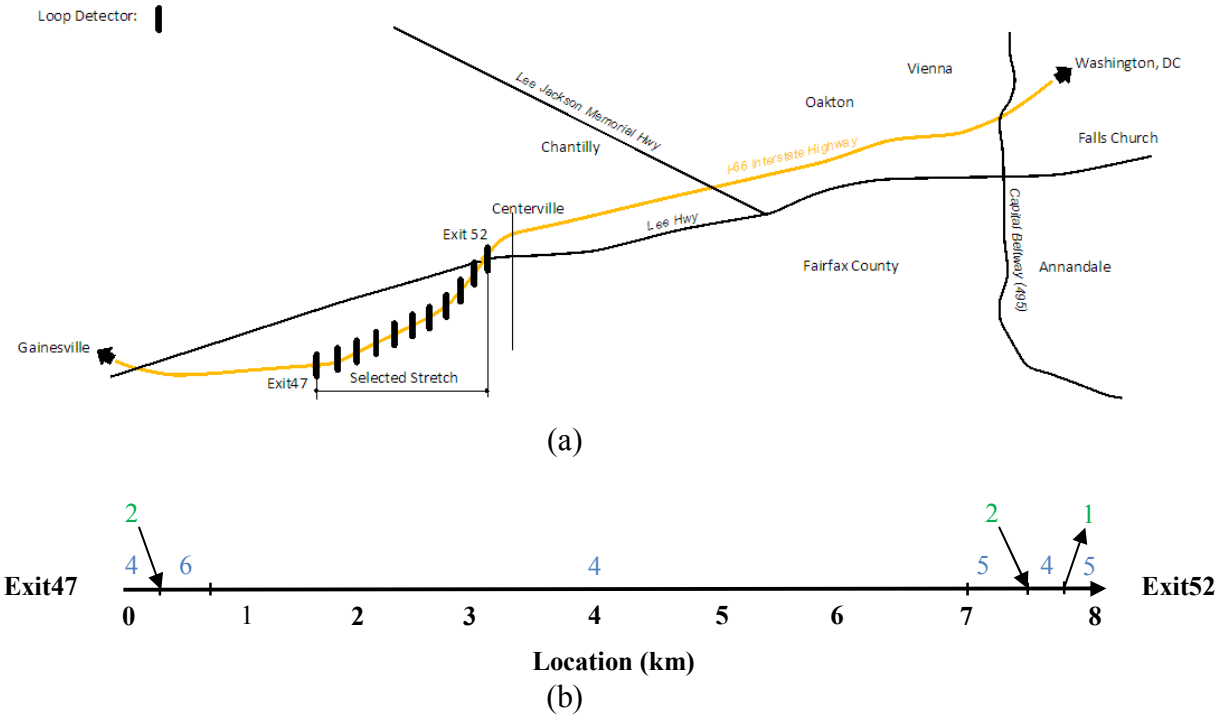


Figure 4.5:Selected freeway stretch on I66. (a) sample I66 freeway network configuration; (b) layout of the selected freeway stretch on I66.

The layout of the selected freeway stretch with loop detectors on I-66 is shown in Figure 4.5 (b). Location represents the distance from Exit 47, so the location of Exit 52 is 8 kilometers. The blue numbers 4, 5 and 6 above the freeway stretch indicate the number of lanes. A change in the number of lanes occurs on the first and last section of the freeway. In addition, there is a two-lane on-ramp located on the first section; the last section includes a two-lane on-ramp and a one-lane off-ramp. Since I-66 is a main highway entering Washington, D.C. from the west, the traffic volume of the study area is usually heavy from 8:00 am to 3:30 pm. Also, since a significant volume of traffic leaves I-66 at Exit 52, a bottleneck is usually formed from the last section of the selected stretch and then propagates backwards. The time dependent O-D demands calibrated using the real-world traffic data from 6:00 am to 9:00 am on May 3rd, 7th and23rd, of 2002 are used for model testing.

Since the proposed speed formulation is non-conservative and not equivalent to the density formulation, the prediction results using the two formulations in the particle filter are compared for these three days. All parameters used in the particle filter are presented on Table 1.Different particle sizes were tested (e.g. 50, 200 and 300), the size of 100 was chosen because it provided a high computational speed with minimum loss in accuracy. Specifically, the computation speed was approximately 1 s to predict the traffic states 5 minutes into the future. Gaussian time update and measurement noises in the particle filter were used for the two approaches. Further testing of different noise distributions are recommended for future research. Since the ramp flows are considered, the coefficients α

and β are set to 0.2 and 0.0 for the first section. For the last section, α and β are set at 0.2 and 0.3, respectively. These values were derived from the field measurements. In order to ensure that the speed propagation is calculated accurately, the state update time interval Δt was set at 15 seconds. The frequency of collecting loop detector data Δt_m was 60 seconds (same polling interval of the field data), and the prediction time span Δt_p was set at five minutes. Various prediction intervals were tested and a 5-minute prediction horizon was found to be the maximum prediction period within an acceptable level of error (speed prediction error less than 5 km/h).

Both relative and absolute prediction errors (MAPE and MAD) are calculated in this paper. It should be noted that the α and β terms were computed using the average conditions based on the field data measurements. The u_f, u_c, q_c, and k_j parameters were calibrated using the field loop detector data. The prediction error (φ) was assumed to be normally distributed with a variance of 2 based on typical values reported in the literature [21]. Conversely, the measurement error (γ) was assumed to be normally distrusted with a variance of 2 given that is the typical margin of error in loop detector speed measurements. However, we do recommend that future work be conducted to quantify the impact of the prediction and measurement error on the performance of the proposed algorithm. A summary of the various parameters is provided below.

$$\alpha_1 = 0.2, \alpha_2 = 0.2; \beta_1 = 0, \beta_2 = 0.3;$$
$$u_f = 104 \text{ km/h}, \; u_c = 85 \text{ km/h}; \; q_c = 2400 \text{ veh/h/lane}, \; k_j = 150 \text{ veh/km/lane};$$
$$\Delta t = 15\text{s}, \Delta t_m = 60\text{s}, \Delta t_p = 300\text{s};$$
$$\varphi_1^n, \; \varphi_2^n, \; \ldots, \; \varphi_N^n \sim N(0,2);$$
$$\gamma_1^n, \; \gamma_2^n, \; \ldots, \; \gamma_N^n \sim N(0,2);$$

Test Results

Here, we use the "speed formulation" to refer to the proposed particle filter using the speed formulation in the time update process, and "density formulation" refers to the same approach but replacing the speed formulation for the density formulation in the particle filter framework. The computation time for one prediction (five minute prediction span) varies between 1 to 2 seconds using MATLAB 2011. The five minute freeway speed prediction results using the two approaches are presented in Table 1. The proposed approach using the speed formulation has lower absolute and relative errors in comparison to the density formulation for all three days. The average absolute and relative errors using the speed formulation are 3.98 km/h and 15.96 percent, respectively. Alternatively, the average error for the density formulation is 10.06 km/h and 34.55 percent. The traffic speed after passage of the shockwave is close to zero (vehicles are almost stopped), hence a little oscillation of speed results in high relative errors. This is the reason that the MAPE values for the two approaches are relatively high. Based on the Van Aerde traffic stream model, the prediction value of density can be calculated using speed, then the accuracy of the density prediction results can be obtained by comparing to density data estimates obtained from the loop detectors (k=q/v). The proposed approach

has an average absolute prediction error of 5.23 veh/km/lane and average relative error of 18.55%, compare to an error of 9.02 veh/km/lane and 30.38% using the density formulation. Hence, the proposed approach predicts the traffic stream density at a higher accuracy.

Table 4.1: Performances for speed and density prediction by two approaches.

Testing Date	Approach	Speed		Density	
		MAD (km/h)	MAPE (%)	MAD (veh/km/lane)	MAPE (%)
May 3 2002 (Friday)	Density Formulation	9.35	32.01	8.24	29.84
	Speed Formulation (Proposed)	3.31	14.24	4.68	16.92
May 7 2002 (Tuesday)	Density Formulation	9.56	32.94	9.14	31.67
	Speed Formulation (Proposed)	4.21	17.32	5.17	20.31
May 20 2002 (Monday)	Density Formulation	11.28	38.69	9.67	29.64
	Speed Formulation (Proposed)	4.43	16.32	5.83	18.41

The spatio-temporal variation in the simulated and predicted traffic stream speed results for May 23rd is illustrated in Figure 4.6. The color bar represents the speed ranges where blue represents free-flow speed and red represents low speeds. According to the plot in Figure 4.6(a), the predicted speed over time shows that the speed formulation can produce a very accurate spatial-temporal match to the measurements; on the other hand, the density formulation produces large errors. By comparing graph (c) and (d) to (b) in Figure 3,we can clearly see that the predicted shockwave in (c) passes through location 7.5 km at 7:00 am to location 6 km at 8:30 am, which is consistent with the field measurements. According to the plot in Figure 4.6(a), the speed drops from free-flow speed to congestion before 7:00 am, and then the speed increases after 8:30 am. The predicted speed profile using the speed formulation offers a better fit to the simulated data. Alternatively, the speed profile produced using the density formulation has significant oscillations. In addition, the predicted speed is over-estimated for the congested condition and underestimated when the speed is around 50 km/h. According to Figure 4.6(b), the shockwave happens at 6:30 am around location 7km and then propagates to location 5km one hour later. The congestion is gradually released after 8:10 a.m. The predicted speed contour in Figure 4.6(d) is consistent with the simulated data. Hence, the proposed approach appears to provide a better match to the simulated I-66 data, although as was demonstrated earlier the speed formulation is not consistent with the LWR theory. One reason why the speed formulation provides a better match to the data is that there are less fluctuations in the speed measurements. In addition, density estimates are obtained from measured speed and flow. Thus although the formulation is mathematically not necessarily correct the reduced error in the speed measurements results in a better prediction relative to the density formulation.

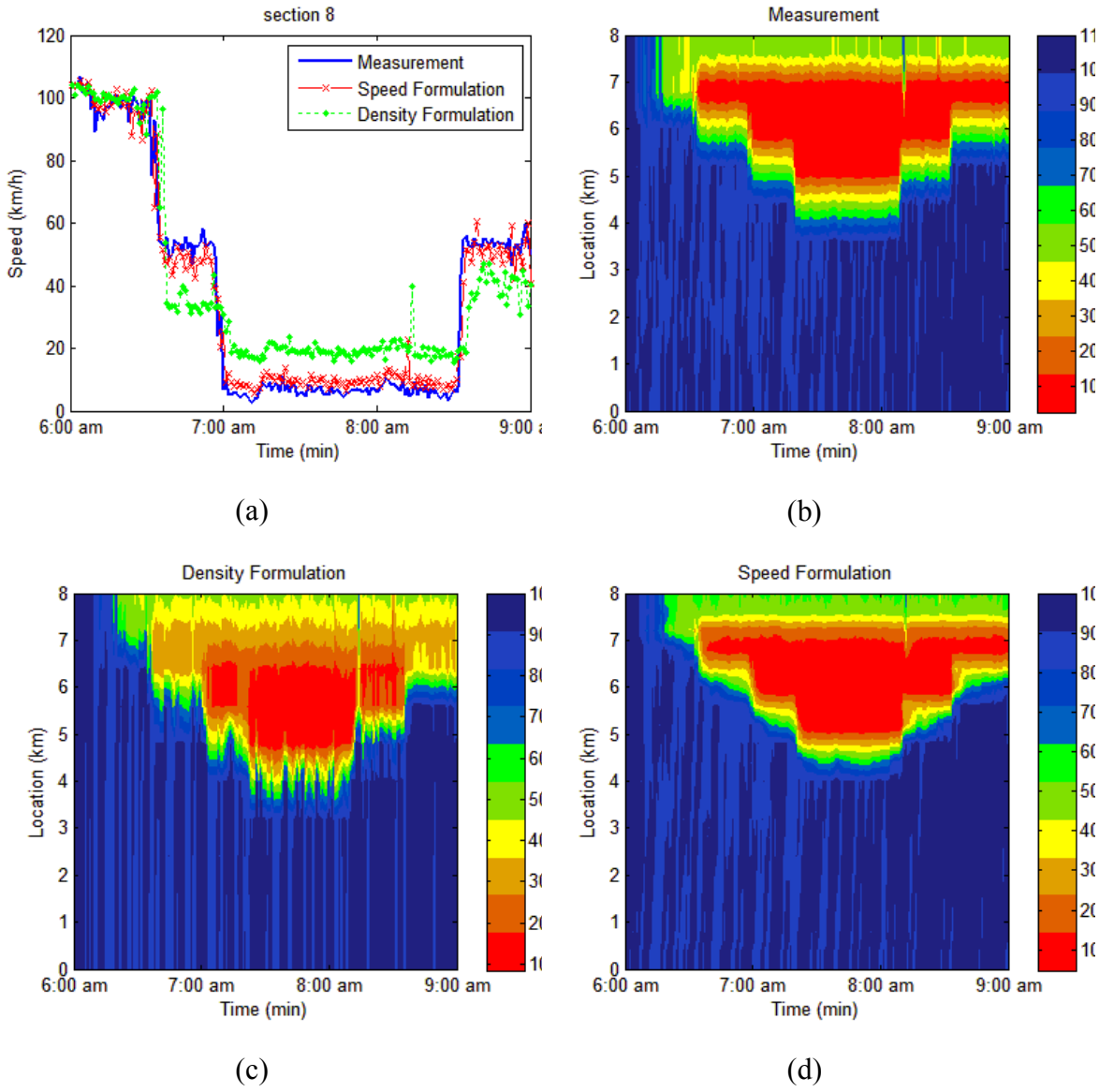


Figure 4.6: Prediction results on May 23 2002. (a) temporal speed variation for section 8; (b) time-space speed data based on measurements; (c) predicted time-space speed contour using density formulation; (d) predicted time-space speed contour using speed formulation.

4.5 Conclusions

The research presented in this chapter develops a particle filter approach to predict traffic state evolution using a new time series speed formulation and measured speed data. With the consideration of ramp flow and incorporating a more realistic traffic stream model – the Van Aerde model – a new Van Aerde flow continuity model is derived via a non-conservative LWR equation. The corresponding numerical solution is obtained using a

cell-centered finite volume Godunov scheme. The speed time evolution formulation is used in the proposed particle filter approach together with the available measured speed data to conduct multi-step freeway speed predictions. In comparison to the density formulation, the proposed non-conservative speed formulation is demonstrated to be not equivalent to the LWR equation. However, the application of the formulation for the prediction of traffic states along I-66 is demonstrated to produce better predictions when compared to the LWR density formulation. This better prediction is attributed to the direct use of speed measurements in the prediction process as opposed to estimating density from speed measurements and the associated errors with these conversions. Consequently, the use of the speed formulation appears to be better from a practical standpoint even though from a mathematical standpoint it has its flaws.

As with any effort additional research is required. We recommend that additional simulations be conducted to quantify the sensitivity of the prediction algorithm to different input parameters. For example, the sensitivity of the results to the use of a non-Gaussian distribution for the prediction and measurement error terms, varying the prediction and measurement error variance, and the percentage of on-ramp and off-ramp flows. Further research is also required to quantify test the algorithm using a combination of loop detector and probe data.

Chapter 5

5 Travel Time Prediction using Macroscopic Traffic Modeling and Historical Data

This chapter is an edited version of: Hao Chen and Hesham A. Rakha (2013), "Forecasting Freeway Dynamic Travel Times by Constructing Trip Trajectories," in *92nd Annual Meeting Transportation Research Board*, Washington D.C.

The chapter develops a novel approach to construct vehicle trajectories using real-time and historical traffic data to predict dynamic travel times. The approach combines real-time and historical data within a particle filter framework to dynamically predict future traffic state maps. The predicted travel trajectory is then constructed using the velocity spatiotemporal map. Based on the nature of particle filters, the variance of each speed grid traversed during the trip can be calculated and then used to compute the travel time variance. The proposed approach is tested using simulated data along a section of I-66. The prediction results demonstrate that the proposed method produces a prediction error half that of the other state-of-the-art methods with a mean absolute deviation of 1.30 minutes and a mean absolute percentage error of 6 percent.

5.1 Introduction

Recently several studies have focused on travel time prediction [3, 7, 9, 31, 35, 36]. Generally there are two categories of previous methods in terms of state variable: direct and indirect methods. In the first category of methods travel time is directly used as the state variable. Different modeling approaches are used to describe the time series travel time relationship, including: time-series models, filtering techniques, and artificial neural network (ANN) techniques. Statistical modeling use current and previous travel time data to develop time series models for predicting the travel time in the next time interval [35]. Some other methods use current speed measurements to compute travel times and apply a Bayesian filter to predict future travel times [7]. Similarly, a Kalman filter approach can be applied by characterizing the temporal relationship of travel time using historical data [36]. In addition, ANNs can also be employed to train the model using historical data [9]. This first category of modeling platforms suffers from a major drawback and that is the travel time in the previous time interval is needed to calculate future travel times. For real-time applications, travel time is typically greater than the prediction interval. Hence, the actual travel time from the previous time interval is not available to make predictions in the following time interval. Alternatively, since it is difficult to construct a physical or mathematical model to describe the temporal relationship of travel time, several attempts have been made to use other variables (speed, density, flow) as inputs to predict travel time, which falls into the second category of prediction methods. For instance, the ARMA model is used in [31] to predict traffic flow and occupancy. Future speed is calculated to predict dynamic freeway corridor travel times with the consideration of traffic evolution along the corridor. However, this approach may be difficult to implement since the described recurrent pattern of traffic conditions may not be found everywhere.

Using traffic speed data, there are two approaches to compute travel times depending on the trip experience [3, 4]. Dynamic travel time is the actual realized travel time that a vehicle would experience during a trip. If a vehicle leaves its origin at the current time, the roadway speed will not only change across space but also across time during the entire trip. Consequently, dynamic travel time can be obtained by using a prediction algorithm to compute the speed evolution in future time steps. Instantaneous travel time is the other approach available to compute travel times without the consideration of speed evolution across time. It is usually computed using the current speed along the roadway; the speed field is assumed to remain constant in time. The instantaneous travel time is close to the dynamic travel time when the roadway speed does not change significantly across time during the trip. However, this approach may deviate substantially from the actual, experienced travel time under transient states during which congestion is forming or dissipating during a trip.

Considering the difficulties associated with formulating a travel time mathematical function and the lag in historical travel time data, the approach presented in this chapter focuses on modeling speed evolution to construct spatiotemporal traffic state maps for use in predicting dynamic travel times. The current and historical measured traffic speed data are used as inputs to particle filter frameworks to predict the traffic state map. The

map describes the spatiotemporal roadway traffic status along the trip. Afterward, the travel trajectory can be constructed using the velocity map to predict dynamic travel times. Moreover, not only can the travel time be predicted but also the entire trip trajectory, the travel time variance of each grid traversed during the trip can be aggregated to compute the travel time variance based on the nature of a particle filter. Simulated I-66 data are used to test three travel time prediction methods and the results demonstrate that the proposed approach is able to provide the best travel time prediction output.

The remainder of this chapter is organized as follows. The first section defines the prediction objective in this study. Subsequently, the travel time prediction methodologies used in the paper are described including the traffic state estimation and trajectory construction approach. The following section describes the model testing based on simulated I66 data. Subsequently, a comparison of the performance of the three prediction models is described. The summary findings, conclusions, and recommendations for future research are provided in the last section.

5.2 Problem definition

In this paper, a freeway stretch of length L can be divided into N sections of length Δx_i, $i=1,..., N$. Traffic stream speed can be measured using various sensing technologies, such as loop detectors, remote traffic microwave sensors, Bluetooth detectors, probe vehicles, etc. Assume the average speed along a freeway section is measured by detectors over time interval Δt. The n^{th} time interval represents time t_n ($t_n=n\times\Delta t$), and u^n_i or $u(x_i,t_n)$ represents the speed of the traffic stream on the i^{th} freeway section at time t_n. The measurement data $x^n=[u_1^n, u_2^n, ..., u_N^n]^T$ describes the traffic state along the whole sections at time t_n. Other than the real-time measuring data, the historical measured freeway speed information is also provided for each time interval, which represents the history traffic pattern on the freeway stretch. Considering the field application of broadcasting travel time information on Dynamic Message Signs (DMSs), a prediction horizon of zero is chosen for the sake of simplicity. Longer prediction windows can also be conducted using the proposed framework, however a detailed demonstration is beyond the scope of this paper. Consequently, when a driver approaches a DMS at current time t_n, the future travel time that departures from time t_n is predicted and provided to the driver by the proposed method.

5.3 Methodology

5.3.1 Proposed framework for travel time prediction

The main idea of the proposed travel time prediction approach is illustrated in Figure 5.1. The current time is t_n, and the future travel time is obtained by constructing the trajectory depicted as the dotted line from location x_0 to x_N. There are two frameworks for speed

estimation in this approach. Since traffic states usually cannot be measured everywhere (the interested freeway stretch may have a poor coverage of loop detectors) and measurement errors exist from various sensing devices (bias measurements or missing data), traffic state estimation is needed to deal with local and noisy sensor data [18, 19]. Consequently, the first estimation framework attempts to deal with missing data and measurement errors. The second framework combines current and historical traffic data to estimate the spatiotemporal traffic evolution starting from t_n and ending with the trip completion time (denoted by t_d). The travel trajectory can be constructed using the estimated speed from the second estimation process as depicted by the highlighted cells in the figure. Hence, future travel time is t_d-t_n. In this paper, both estimation frameworks use the same particle filtering technique, but the measurement data are different for each estimation framework. The first estimation uses the incoming speed measurements from loop detectors, and the second estimation uses the historical speed data as the measurement update. The details of the particle filter estimation are presented in the next section.

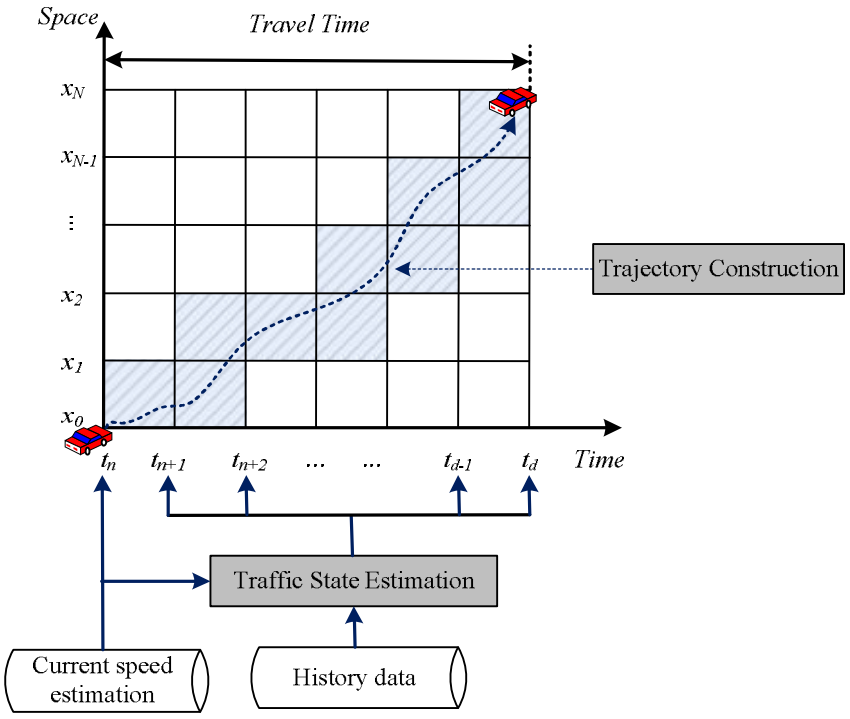


Figure 5.1: Illustration of proposed travel time prediction approach.

5.3.2 Traffic state estimation

In this chapter, the particle filter approach proposed from our previous work [6, 76] is used for the purpose of traffic state estimation. In order to model freeway state evolution, a new time series traffic speed formulation, which describes the spatial and temporal relationship of traffic speed on the freeway stretch, was generated by combining a more realistic traffic stream model - Van Aerde model with the traditional Lighthill-Whitham-Richards (LWR) flow continuity equation. Data estimation can be conducted by

implementing the speed formulation in the particle filter framework. Before the detailed introduction of how to use the particle filter approach in this paper, the two basic processes within the general particle filter framework are presented below.

State Update Process:

$$x^{n+1} = G\left(x^n\right) + \varphi^n . \tag{5.1}$$

Measurement Update Process:

$$y^n = H^n\left(x^n\right) + \gamma^n . \tag{5.2}$$

where $x^n=[u_1^n, u_2^n, ..., u_N^n]^T$ represents all speed state variables in the traffic network at the n^{th} time interval; $G(x)$ denotes the state evolution function; the process noise is defined by $\varphi^n=[\varphi_1^n, \varphi_2^n, ..., \varphi_N^n]^T$; y^n includes all available measurement speed data from detectors at time t_n; H^n is a diagonal matrix, in which the eigenvalue is one if the corresponding freeway section has measurement data at the n^{th} time interval, otherwise it is zero; γ represents the measurement noise vector.

Given the freeway velocity of all sections on the traffic network at time $n\times\Delta t$, the network velocity at time $(n+1)\times\Delta t$ is constructed using Equation (5.1). Once the freeway speed measurement data are available, they are used to adjust the predicted traffic state for the same time interval, in order to converge the error and prepare for the prediction in the next iteration. A discretized Van Aerde flow continuity model formulated for the traffic stream speed [6] was derived to characterize the spatiotemporal evolution of traffic using Equation (5.1) in a particle filter framework. The speed formulation is represented as Equation (5.3). The boundary flux is computed as Equation (5.4) for different conditions by the direction of speed propagation characteristic. The flux F is a function of speed defined by Equation (5.5). Speed is the only variable in this flux function and other coefficients can be calculated using the four basic traffic stream parameters as demonstrated in Equation (5.6).

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n - F_{i-1/2}^n\right). \tag{5.3}$$

$$F_{i+1/2}^n = \begin{cases} f(u_{i+1}^n) & if\ u_i^n \le u_{i+1}^n \le u_0 \\ f(u_0) & if\ u_i^n \le u_0 \le u_{i+1}^n \\ f(u_i^n) & if\ u_0 \le u_i^n \le u_{i+1}^n \\ \max\left(f(u_i^n), f(u_{i+1}^n)\right) & if\ u_i^n \ge u_{i+1}^n \end{cases}. \tag{5.4}$$

$$F = f(u) = \left(1-\alpha+\beta\right)\left\{-\frac{c_1}{c_3}u + \frac{c_2}{c_3}\ln\left[c_3\left(u_f-u\right)^2 + c_2\right] - \sqrt{\frac{c_2}{c_3}}\left(\frac{c_1}{c_3}+u_f\right)\arctan\left[\sqrt{\frac{c_3}{c_2}}\left(u_f-u\right)\right]\right\}. \tag{5.5}$$

$$c_1 = \frac{u_f}{k_j u_c^2}\left(2u_c - u_f\right); \quad c_2 = \frac{u_f}{k_j u_c^2}\left(u_f - u_c\right)^2; \quad c_3 = \frac{1}{q_c} - \frac{u_f}{k_j u_c^2}. \tag{5.6}$$

Here c_1 is a fixed distance headway constant (km), c_2 is a variable headway constant (km^2/h), and c_3 is a variable distance headway constant (h). This combination provides a functional form with four degrees of freedom by allowing the speed-at-capacity (u_c) to differ from the free-flow speed (u_f) [26]. The four basic traffic stream parameters are free-flow speed (u_f), speed-at-capacity (u_c), capacity (q_c) and jam density (k_j). The α and β parameters represent the ratio of on-ramp flow and off-ramp flows to the mainstream flow, respectively.

In order to have a unique solution to Equation (5.4), boundary conditions must be specified [45]. Considering the characteristics may be entering or leaving the domain on the right-hand boundary. The direction of characteristic should be calculated in order to set the correct boundary flux value. Assume c represents the derivative of q with respect to k ($c=dq/dk$), which is the characteristic. The boundary conditions are expressed in Equation (5.7). The Courant-Friedrichs-Lewy (CFL) condition must be satisfied as described in Equation (5.8) in order to maintain numerical stability [43].

$$\begin{aligned}
&\text{at } x = 0 \text{ (left-hand boundary)} \\
&\qquad \text{if } c(u) > 0 \;\; u_{-1/2}^n = u_0^n \\
&\qquad \text{if } c(u) \le 0 \;\; u_{N+1/2}^n = u\left(1, t_n\right) \\
&\text{at } x = L \text{ (right-hand boundary)} \\
&\qquad \text{if } c(u) > 0 \;\; u_{N+1/2}^n = u_N^n \\
&\qquad \text{if } c(u) \le 0 \;\; u_{N+1/2}^n = u\left(L, t_n\right)
\end{aligned} . \tag{5.7}$$

$$\left| \frac{\Delta t}{\Delta x} \max\left(f'(u)\right) \right| \le 1. \tag{5.8}$$

The framework of particle filter estimation is illustrated in Figure 5.2. Detailed descriptions are presented in [6, 76]. There are several advantages of using the proposed particle filter approach. Firstly, the more realistic Van Aerde traffic stream model is used instead of the Greenshields model, which is usually used in other studies. The Van Aerde model provides a functional form with four degrees of freedom by allowing the speed-at-capacity (u_c) to differ from the free-flow speed (u_f), as is the case in most models including the Pipes, Gipps, and Newell's simplified model, or half the free-flow speed, as is the case with the Greenshields model [76]. Secondly, the traffic stream speed is the only state variable in the traffic evolution model, therefore speed measurement data can be used directly to correct the model prediction error. Thirdly, according to the nature of a particle filter, each particle is associated with a weight. Consequently, the travel time for each component cell (the shaded cell in Figure 5.1) is a distribution rather than a single value. The mean and variance of the total travel time can be computed afterwards. For simplicity, the correlation between the component cells is assumed to be zero (i.e.

they are independent). More advanced approaches to calculate the trip mean and variance of travel time can be found in [77].

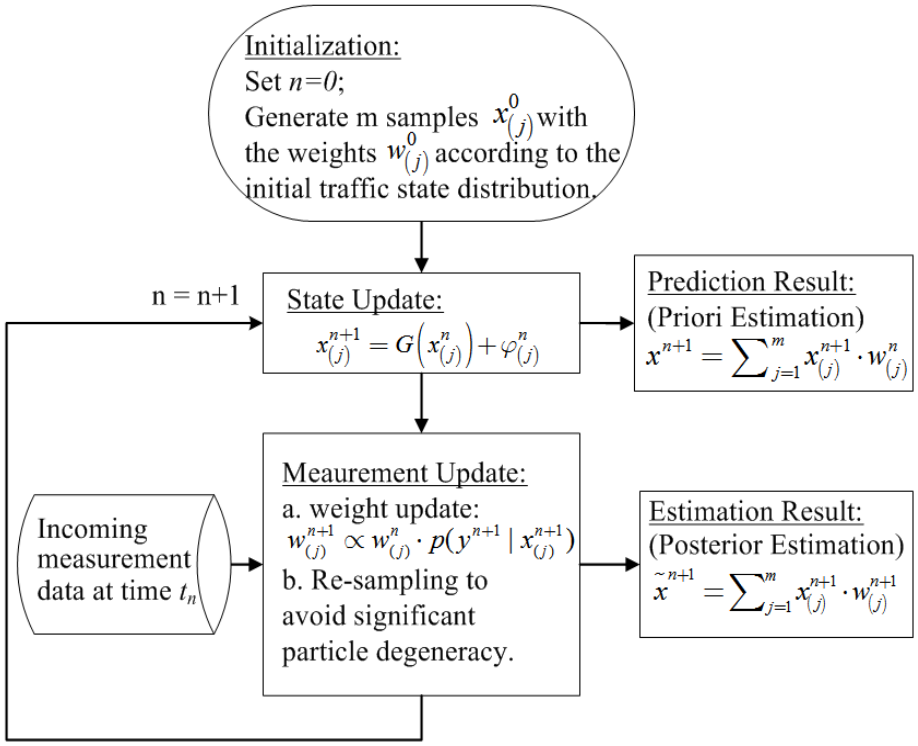


Figure 5.2: Particle filter traffic state estimation framework.

5.4 Trajectory construction

In the process to predict the travel time starting from the current time t_n, the incoming measurement data is replaced by the historical data. The estimation result represents the spatiotemporal traffic state map from time t_n to t_d for trajectory construction. A simple example of travel time calculation based on trajectory construction is demonstrated in Figure 5.3. The roadway is divided into four sections using a space interval Δx, and the time interval Δt. In the numerical application of this paper, the traffic state is assumed to be homogenous within each cell. The average speed of the red dotted cell ($i=2$, $n=3$) is $u(x_2,t_3)$. The traffic speed along the freeway stretch can be estimated using the particle filter approach. Consequently, the trajectory slope, which represents the speed in each cell, is a constant value within the cell. In this way, once the vehicle enters a new cell, the trajectory within this cell can be drawn as the straight blue line in Figure 5.3 using the cell speed as the slope. Finally, the future travel time can be calculated when the trip reaches the downstream boundary of the last freeway section.

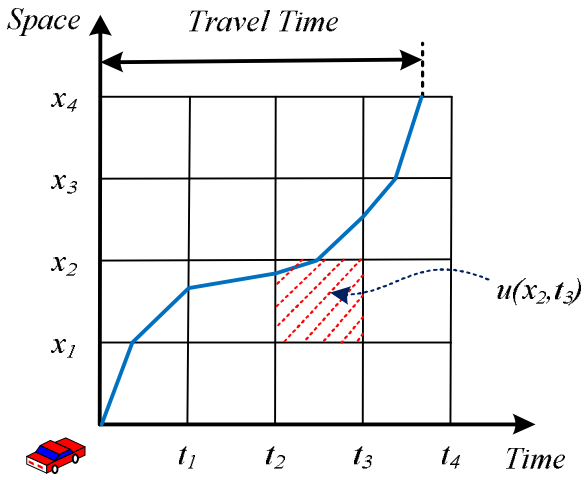


Figure 5.3: Illustration of travel time calculation.

However, there is a computational problem associated with this procedure. Since the trip destination time is unknown before the completion of the trip, the required number of estimated speed cells is not known a priori. In order to minimize the computational time for traffic state estimation, an intelligent computational procedure is proposed to detect if the trip trajectory leaves the current time interval to determine if speed estimates are needed for the following interval. The traffic state estimation process is only needed when the trip reaches the right boundary (each time interval) of the current cell, which means it will enter the cell of the next time interval. So the spatiotemporal traffic pattern map grows towards the time axis only when the trip traverses the boundary of each time interval. The flow chart of the trip trajectory construction for travel time prediction is presented in Figure 5.4. The beginning time of trajectory construction is t_n. The initialization step attempts to setup the initial traffic state vector for the first time interval for travel time prediction, the speed vector is obtained from the first traffic state estimation framework. Since the trip begins from the upstream boundary of the first freeway section, it will travel across the first cell. The trajectory slope is used to identify if the trajectory will move vertically to the cell in the following section in the same time interval, or if it will move horizontally to the cell within the same section in the next time interval. For the former situation, traffic state estimation is not needed. One more state estimation is needed for the latter situation, because the time step of data space moves to $n+1$. Cell and trajectory locations are updated respectively to compute the accumulated travel time and distance. Travel time predictions will be obtained when the vehicle arrives at its destination. Otherwise the iteration will continue to proceed until the trip trajectory passes the time interval boundary. This process can efficiently save computational time. A case study will be presented in the next section to evaluate the proposed approach to predict travel times.

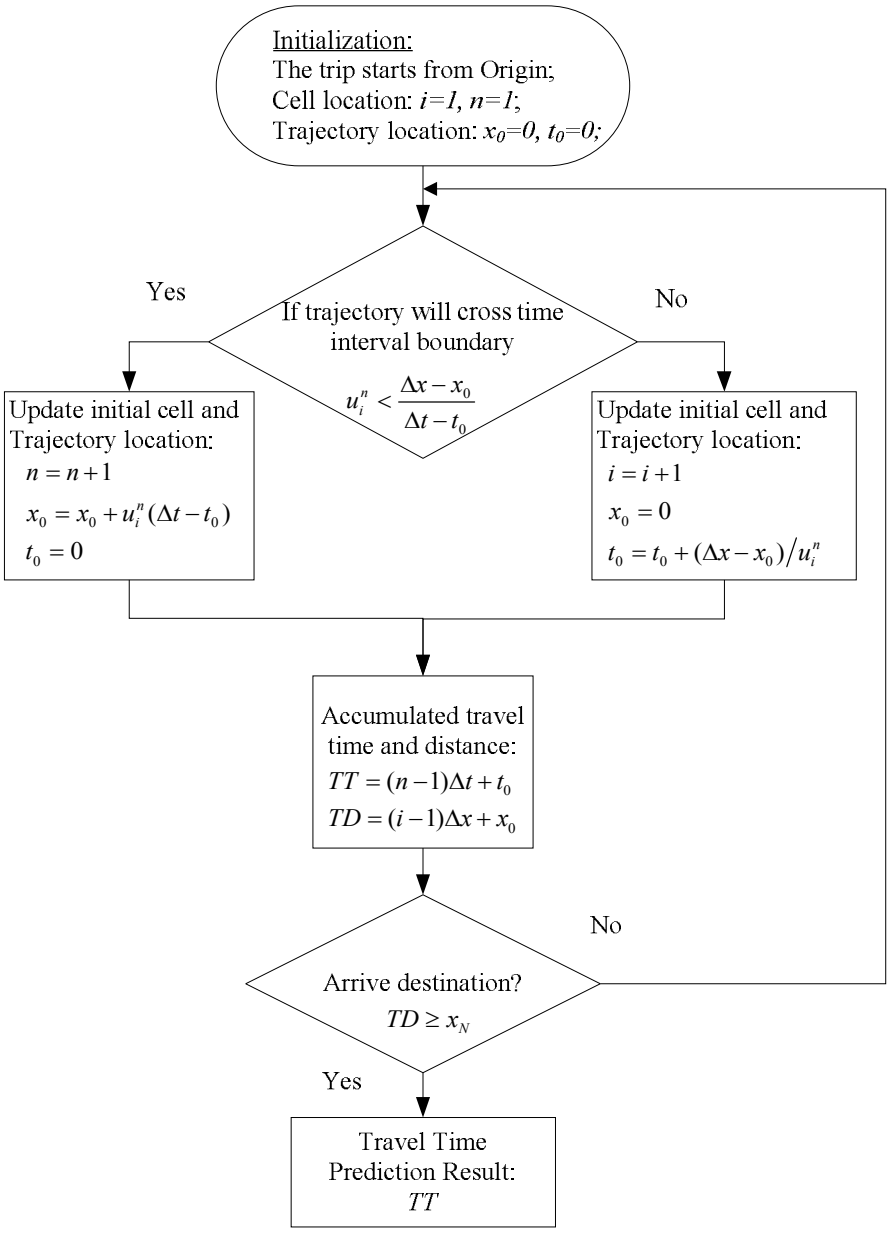


Figure 5.4: The flow chart of trajectory construction.

5.5 Case study

The performance of the proposed dynamic travel time prediction approach is investigated by the case study in this section. The description of test data is introduced and followed by the comparison between the proposed approach and other two methods for travel time prediction.

5.5.1 Simulation setup

A microscopic traffic simulation software, INTEGRATION, which was developed over the past two decades[49, 53-55]is used to simulate the freeway traffic state data. The INTEGRATION model is a trip-based microscopic traffic assignment, simulation and optimization model. The INTEGRATION model updates vehicle speeds every deci-second based on a user-specified, steady-state speed-spacing relationship and the speed differential between the subject vehicle and the vehicle immediately ahead of it. In order to ensure realistic vehicle accelerations, the model uses a vehicle dynamics model that estimates the maximum and typical vehicle acceleration levels. Specifically, the model utilizes a variable power vehicle dynamics model to estimate the vehicle's tractive force that implicitly accounts for gear-shifting on vehicle acceleration. This model is described in more detail in the literature[56, 57].The model allows detailed analyses of lane-changing movements and shock wave propagations. It also permits considerable flexibility in representing spatial and temporal variations in traffic conditions. In addition to estimating vehicle stops and delays[58-60], the model can also estimate the fuel consumed by individual vehicles and the emissions[61], [62-64]. Finally, the model also estimates the expected number of vehicle crashes using a time-based crash prediction model [65].

The INTEGRATION model has not only been validated against standard traffic flow theory [37, 59, 60, 66] but also has been utilized for the evaluation of real-life applications [67-69]. For example, the INTEGRATION software has been demonstrated to model traffic dispersion through two procedures: a) introduction of randomness in driver/vehicle speeds or b) the use of stochastic car-following models. In the case of a stochastic car-following model, each driver has a unique random realization of a roadway specific steady-state car-following model with an average that is consistent with the mean driver-behavior. Several studies [60, 70, 71] showed that INTEGRATION's outputs are consistent with both observed field data and fundamental traffic flow theory. In addition, a study involved a basic validation of the traffic dispersion module using two sample field datasets [72]. The first dataset was gathered by Denney [73] in Houston, Texas and the second dataset was gathered by Castle and Bonniville[74] in Kuwait City, Kuwait. The Houston data contained the observed average flow profiles at an upstream signal and at a checkpoint 300 m (990 ft) downstream on a 3-lane arterial roadway. The observed average speed and standard deviation of speeds were reported as 48.3 km/h (44 ft/s) and 5.9 km/h (5.4 ft/s), respectively, which results in an observed speed coefficient of variation (CVobs) of 12.3%. Given that vehicle interactions may reduce vehicle speed variability, a slightly higher value of CV was input to the simulation software (CVin = 15%). The network was modeled as a 300-meter three-lane link, with a free-flow speed of 50 km/h, a speed-at-capacity of 40 km/h, a saturation flow of 1800 veh/h/lane, and jam density of 100 veh/km/lane. In the case of the Houston dataset, the simulated average speed and standard deviation were 48.2 km/h and 6.1 km/h (CVout=12.7%), which is very similar to the field-observed parameters. Consequently, the results demonstrated a high degree of consistency between simulated and field observed data in terms of aggregate trip measures (trip mean and variance) and in the progression of vehicles within platoons.

The study freeway is a stretch of Interstate 66 as shown on Figure 4.5. The stretch starts from Manassas (Virginia) west towards the direction of Vienna east, between Exit 47 and Exit 62 of I-66. The total length of the freeway stretch of interest is 24 kilometres (15 miles) and 30 loop detectors are located every 0.8 kilometre (0.5 mile) in this freeway stretch to collect speed information by every one minute. Since I66 is the only main highway driving from the west end to Washington, D.C., the traffic volume of study stretch is usually heavy from west to east direction during morning peak hours. Also, since the traffic leaving I66 by Exit 52 is heavy, a shockwave is usually formed since Exit 52 and then moves backward, which is between 8th and 10th loop detector. The O-D demands of traffic network used in INTEGRATION simulation has been calibrated based on the previous studies by QUEENSOD software [44]. The time dependent O-D demands calibrated by the real world traffic data from 5:00 am to 12:00 pm on May and June 2002, is used for model testing. Since traffic congestion usually happens during morning peak hours on weekdays, weekend data is not considered in this paper. In this session, ten weekday traffic data on May are used as training data set to obtain the average freeway section speed and three weekday simulation traffic data on June are employed as testing data set to evaluate the prediction accuracy. In order to get the observed travel time (ground truth) on the freeway stretch, all the vehicles passing the origin and destination of the stretch are recorded from simulation and individual travel time are aggregated by five minutes interval as the ground truth data.

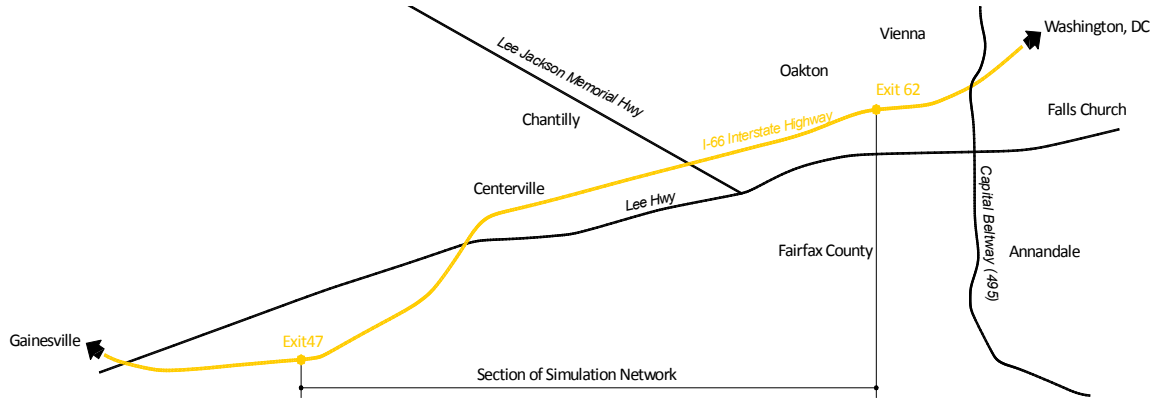


Figure 5.5: Sample I-66 network configuration.

The selected freeway stretch includes six on-ramp and seven off-ramp. The ramp flows are simplified as the ratios to main flow in the particle filter estimation framework using the average conditions based on the field data measurements. The u_f, u_c, q_c, and k_j parameters were calibrated using the field observation data as 104 km/h, 85 km/h, 2400 veh/h/lane and 150 veh/h/lane respectively. The prediction error (φ) was assumed to be normally distributed with a variance of 2 in the particle filter estimation. Conversely, the measurement error (γ) was assumed to be normally distrusted with a variance of 2 given that is the typical margin of error in loop detector speed measurements. The measurement data are obtained by one minute interval to update the speed estimation data on the

stretch. The estimated section speed are combined with history data to construct the future trip trajectory every five minutes, which is also the travel time prediction interval.

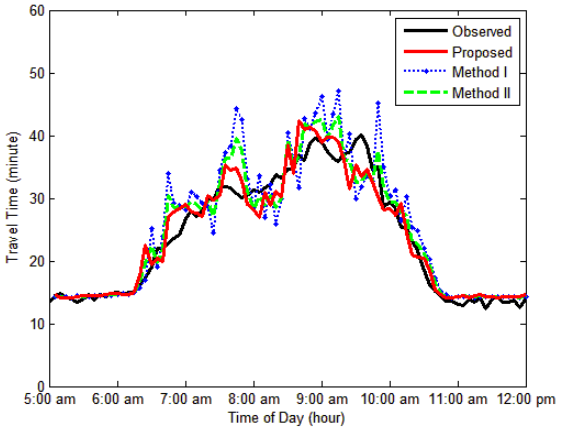
5.5.2 Comparison approaches and performance indices

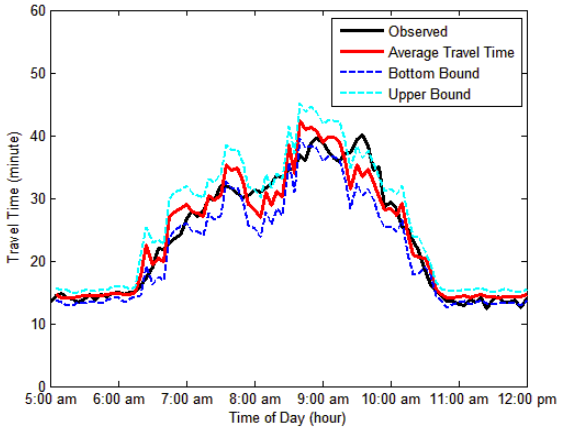
In order to better evaluate the performance of the proposed approach to predict travel time, two alternative methods are tested. Method *I* only calculates the instantaneous travel time, and thus does not consider the temporal change in freeway speed for the future trip. In other words, the current section estimated speeds are assumed to be unchanged for the entire trip (i.e. the trajectory is assumed to be a vertical line). Consequently, the predicted travel time is the aggregation of the section travel times at instant t_n. Alternatively, freeway section speeds are assumed to change in Method *II*, and the temporal change of speed is characterized only by historical data.

Both relative and absolute indices are used in this paper to measure the effectiveness of the three travel time prediction methods. The absolute error is presented by the Mean Absolute Deviation (MAD) and the relative error is presented by Mean Absolute Percentage Error (MAPE).

5.5.3 Test results

The travel time prediction results using the three methods on Jun 6 2002 are presented on Figure 5.6. Specifically, Figure 5.6(a) shows the comparison of predicted travel time using the three methods. The average travel time and its variance can be predicted using the proposed method, the average prediction result is represented by the red curve and has the best fit to the observed travel time data. The upper and bottom boundaries of the travel time information using the proposed method can be calculated according to a 95% confidence interval as shown on Figure 5.6(b). By using this predicted travel time variance information instead of the average value, the uncertainty in travel time prediction can be characterized to better track the fluctuation in future travel time. The upper and bottom boundaries cover the observed travel time curve very well.



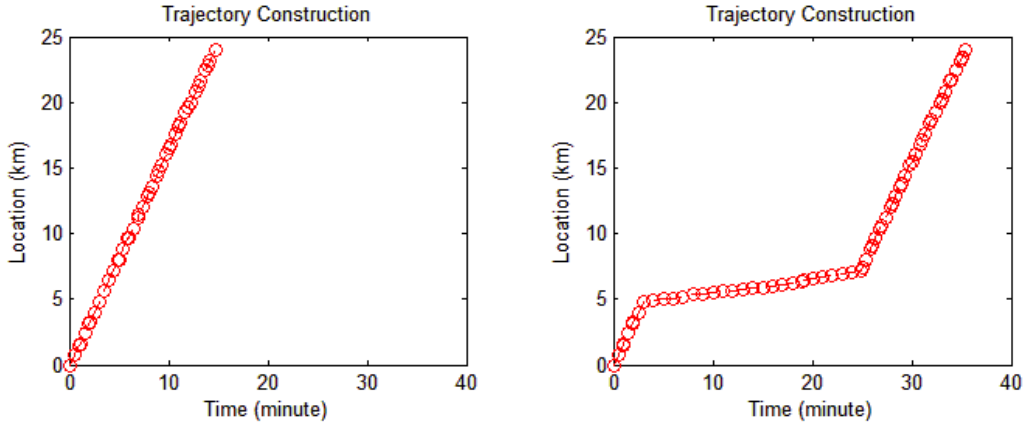


(a) (b)

Figure 5.6: Travel time prediction results on Jun 6 2002; a) predicted travel time by three methods; b) predicted average travel time and 95% confidence interval by proposed method.

Samples of predicted trip trajectories using the proposed approach are presented on Figure 5.7. The red circle represents the intersection point of trip trajectory and every cell boundary. Based on the proposed trajectory construction method, the number of iterations to estimate the spatiotemporal speed map is calculated until the end of the trip. Figure 5.7(a) demonstrates the uncongested trip trajectory, and the travel time is predicted using 15 minutes of historical data. Figure 5.7(b) demonstrates the congested trip trajectory, and the travel time is predicted using 35 minutes of historical data.

The travel time prediction errors for June 2002 using the three methods are presented on Table 1. Based on the statistics of absolute and relative prediction errors, the proposed method produces the best prediction accuracy. The other two methods produce approximately twice the prediction error compared to the proposed method. The results by Method *II* produce less errors compared to Method *I,* as demonstrated in Table 1, since recurrent congestion exists in the testing data set, which is consistent with the historical data. It should be mentioned that the proposed method in this paper works well for recurrent congestion, since we only use the average history speed data similar to the testing day. Analysing historical data in a more intelligent way in order to find candidate similar data is recommended for the future research.



(a) (b)

Figure 5.7: Sample of predicted travel trajectories by proposed approach.

Table 5.1: Comparison of prediction results by three methods.

Test Date	Prediction Methods	MAD (min.)	MAPE

June 3 2002	Proposed	1.29	6.11
	Method I	2.87	10.6
	Method II	2.25	8.42
June 6 2002	Proposed	1.32	6.45
	Method I	2.96	11.4
	Method II	2.89	10.85
June 18 2002	Proposed	1.41	7.52
	Method I	3.16	11.74
	Method II	3.24	12.05

5.6 Conclusions

This research presented in this chapter develops a dynamic travel time prediction approach based on trip trajectory construction. Two traffic state estimation frameworks are used based on a particle filter approach. The first framework estimates traffic speed from noisy measurement data. The second framework combines the estimated speed from the first framework with historical traffic information to estimate future temporal spatial speed evolution for the duration of the trip. Trip trajectory is constructed using the speed map to predict travel times. In order to minimize the computational time, an intelligent computation process is proposed to estimate the speed map in the next time interval only when the trajectory reaches the time boundary. Simulated data on a section of I-66 is used to test the proposed method. The prediction results demonstrate that the proposed method produces a prediction error half that of the other state-of-the-art methods with a mean absolute deviation of 1.30 minutes and a mean absolute percentage error of 6 percent.

Chapter 6

6 Travel Time Prediction using Pattern Recognition

This chapter is an edited version of: Hao Chen, Hesham A. Rakha, and Catherine C. McGhee (2013), "Dynamic Travel Time Prediction using Pattern Recognition," in *20th World Congress on Intelligent Transportation Systems*, Tokyo, Japan, and Hesham A. Rakha and Hao Chen (2013), "Prediction of Dynamic Travel Times: The Richmond to Virginia Beach Case Study," Project Report, [Sponsored by: Virginia Department of Transportation]

The research efforts in this chapter develop a travel time prediction algorithm using pattern recognition techniques to match historical data with real-time traffic conditions. Moreover, a data reduction procedure is developed to transfer the raw INRIX probe data into a uniform format which can be used in the proposed travel time prediction algorithm. The probe data from I-64 and I-264 between Richmond to Virginia Beach for the past three years are collected for testing proposed algorithms. INRIX data for the selected 37-mile freeway stretch (Newport News to Virginia Beach) are used to test the proposed algorithm in case study 1 and 2. The testing results indicate that the proposed algorithm outperforms the other three methods including using instantaneous measurements, using a Kalman filter, and using the k nearest neighbor method. Moreover, the case study 3 conducted on the entire 95-mile freeway stretch from Richmond to Virginia Beach demonstrates the superiority of the proposed algorithm over the instantaneous approach that is currently used by Virginia Department of Transportation. The proposed prediction method reduces the prediction error by approximately 50 percent compared to the current instantaneous method, especially at the shoulders of the peak periods.

6.1 Data reduction and analysis

The proposed approach used in this study is a data-driven method, yet it outperforms the previous methods by fully utilizing the relationship between traffic states and travel times. Moreover, unlike previous studies using travel time sequences as input, the proposed method uses temporal-spatial traffic data to match traffic patterns between real-time and historical data. Many advanced pattern matching techniques can be implemented in the proposed approach to find similar historical traffic patterns more efficiently and accurately, and obtain better travel time prediction results.

6.1.1 Data reduction of INRIX probe data

In order to use the collected traffic data in the proposed algorithm, data reduction is required to transfer the raw measurements into the required input format. In general, the spatiotemporal traffic state matrix is the main input component. Here, the data reduction of INRIX probe data is described as an example illustration. Similar procedures are required for other data sources (e.g. loop detector data).

The INRIX data is collected for each roadway segment at different time intervals. Each roadway segment represents a TMC station, and the geographic information of TMC station is also provided. The average speed for each TMC station can be used to derive spatiotemporal a traffic state matrix. However, the raw INRIX data includes several problems, such as geographically inconsistent sections, irregular time intervals of data collection, and missing data. Considering these problems, the data reduction process is illustrated as Figure 6.1.

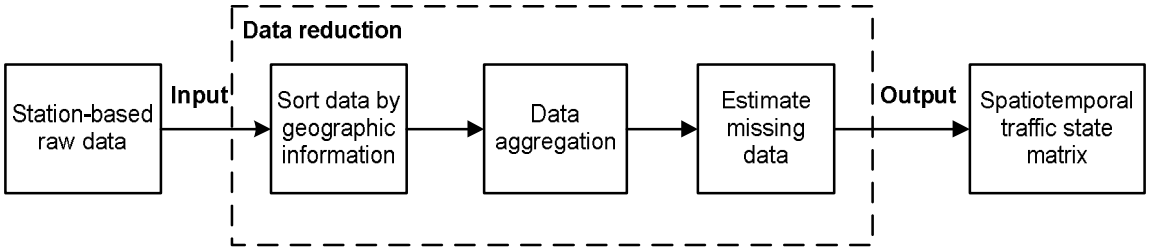


Figure 6.1: Data reduction of INRIX probe data.

According to the geographic information of each TMC station, the raw data are sorted along the roadway direction (e.g. eastbound or westbound). An examination should be adopted to check for any overlapping or inconsistent stations along the roadway. Subsequently, the speed data are aggregated temporally (e.g. 5 minutes) according to the resolution requirement of the algorithm. In this way, the raw data are aggregated to construct a daily data spatiotemporal matrix. It should be noted that missing data is a typical problem. Consequently, data imputation methods should be used to construct missing data using neighboring (spatial and temporal) cell data. Eventually, the daily

spatiotemporal traffic state matrix can be generated to construct travel dataset for case study.

Acquire INRIX Data

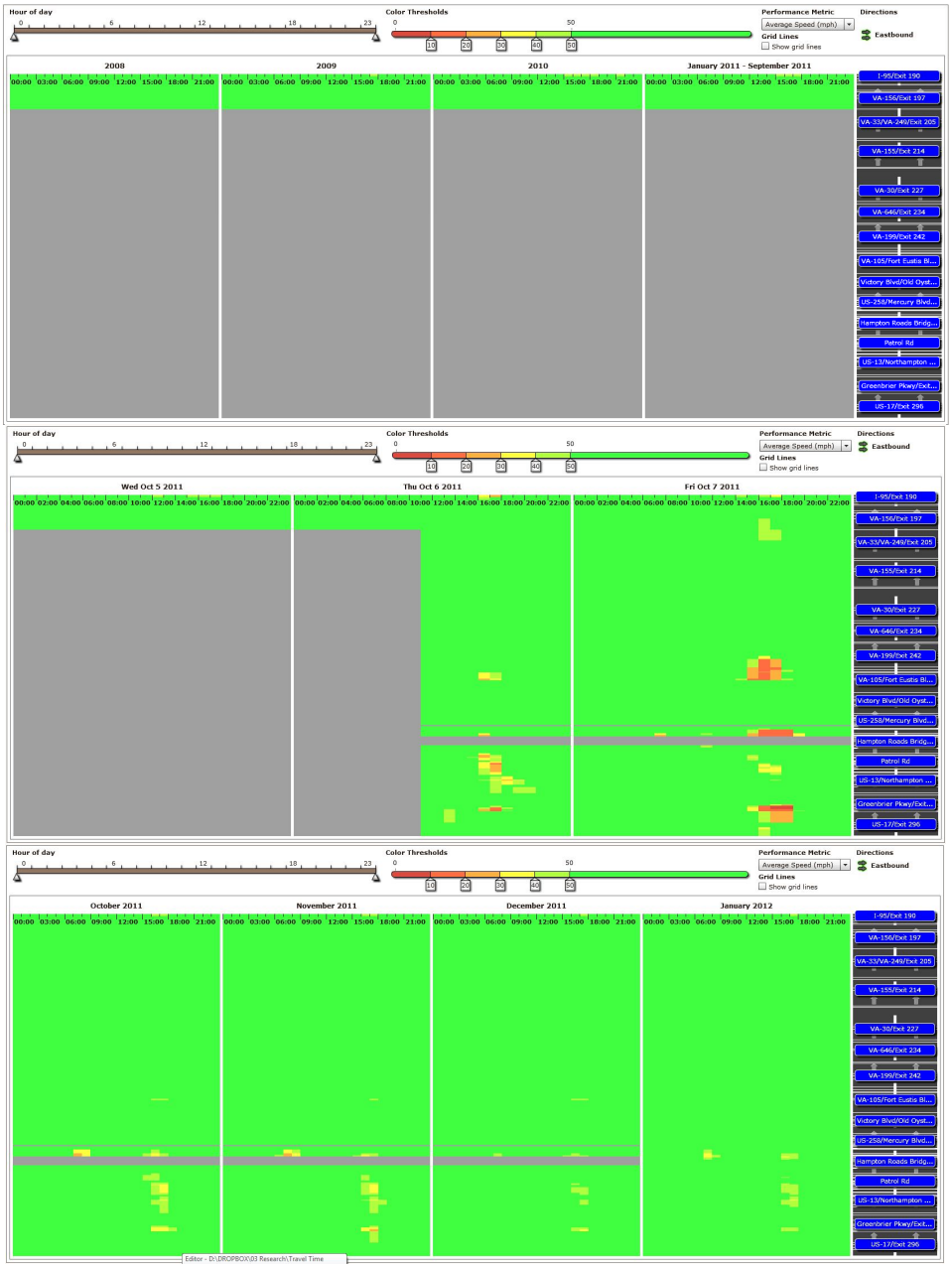
Data reduction for the available INRIX data is an important task to be accomplished during this study. The acquired data set will be used to construct the travel database, and to develop and test the travel time prediction algorithm. After signing the Date Use Agreement (DUA) with RITIS and sending the request for INRIX data, two packages of probe data on I-64 and I-264 from the time period of October 2008 to November 2012 are obtained, as illustrated in Figure 6.2 (a) and (b), respectively.

(a) INRIX data for I-64

(b) INRIX Data for I-264

Figure 6.2. INRIX data for I-64 and I-264.

According to Figure 6.3, the aggregated speed data on the RITIS website demonstrate the data coverage information for I-64. The gray color indicates that the corresponding area has no data at all, and the green/yellow/red color indicates that traffic measurement is available. During the time period before October 7 2011, only the first 11 freeway sections include traffic data measurement. Those 11 sections are located east of I-295, which is out of the range of the study area. Therefore, the INRIX data for those sections cannot be used for this study. The speed data from other sections are available starting from 12:00 p.m. on October 6, 2011. However, the Hampton Roads Bridge-Tunnel (HRBT) still had no measurement until the end of January 2012. Finally, the full coverage data can be observed from the end of January 2012 through November 2012.



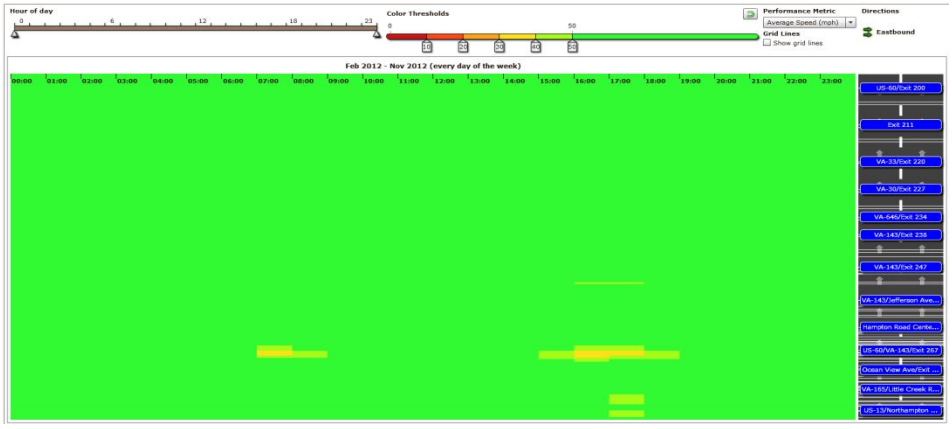


Figure 6.3. Aggregated INRIX data for I-64.

The same procedure of data analysis for I-264 is also conducted using the service from the RITIS website, and the display of aggregated INRIX data is presented in Figure 6.4. Similar to I-64, no measurement data were collected on I-264 before October 6, 2011. Afterward, the speed data measurements have almost full coverage on I-264 from October 6, 2011 to November 2012.

Figure 6.4. Aggregated INRIX data for I-264.

As illustrated in Figure 6.3 and Figure 6.4, the ideal coverage of traffic measurements along I-64 and I-264 from Richmond to Virginia Beach starts on October 7, 2011, and ends on November 30, 2012. Considering that serious congestion mainly occurs along I-64 during that time, the summer holiday season (June, July, and August) is the main focus of this study. The summer 2012 data are not available when the proposed algorithm is tested; however, a third data package collected by INRIX and covering January to December of 2010 was added to this study. Therefore, the summer 2010 data can be used in this study to validate the algorithms of travel time prediction.

The INRIX data from October 7, 2011, to November 30, 2012 are used to construct the first data set along I-64 and I-264. The 2010 INRIX data are used to construct the second data set with the same freeway stretch from Richmond to Virginia Beach. It should be noted that the data resolution and coverage are different between two data sets. The first data set has a fine resolution over the spatial and temporal domains as the data collection time interval is usually around 1 minute. Comparatively, the second data set includes a coarse coverage with 5 minutes of data representation and missing data during early morning/late night and weekends. The detailed information of data set representation will be described in the following sections.

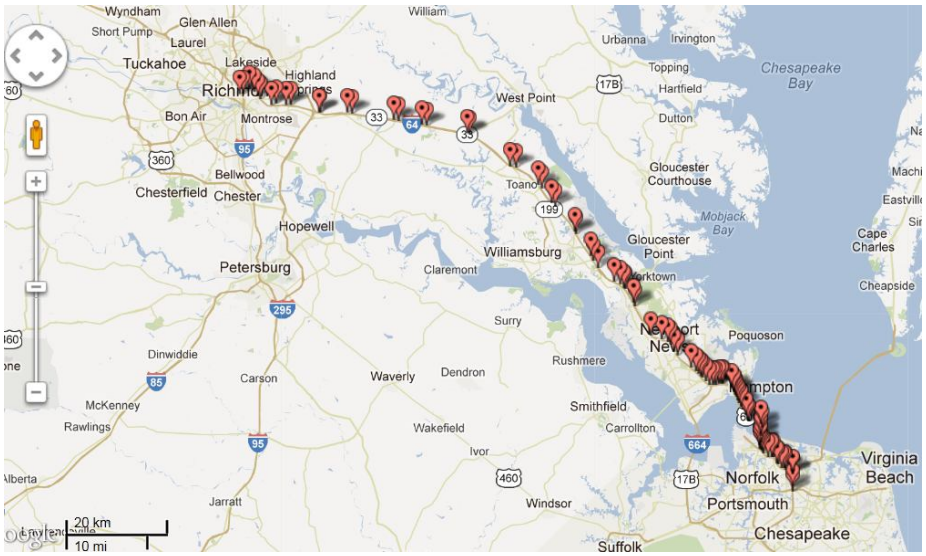
Problems with Raw Data

Since the size of the raw data is very large (e.g., freeway sections on I-64 from October 2011 to January 2012 include 3.33 GB raw data), Microsoft Excel or Access cannot open the data directly. SAS and MATLAB are used to filter the raw data to obtain the spatiotemporal average speed data of each individual day. The raw data are presented in Figure 6.5. Each row is generated by "tmc_code" and "measurement_tstamp." The speed information is used during this study to represent the traffic state of the corresponding roadway section at each time interval. The geographical information for each "tmc_code," which corresponds to a freeway section, is defined in a separate file. However, there is no information given to define the spatial relationship of each section. Consequently, the first step of data reduction is to sort all sections from west to east along

I-64, as shown in Figure 6.6. The same procedure for sorting sections is adopted for I-264.

tmc_code 1	measurement_tstamp 2	speed 3	average_speed 4	reference_speed 5	travel_time_minu... 6	confidence_score 7	cvalue 8
110+04706	2011-10-06 11:58:46.0	58	60	63	1.821	30	100
110P04702	2011-10-06 11:58:46.0	59	59	62	0.707	30	100
110+04703	2011-10-06 11:58:46.0	59	60	63	0.326	30	100
110+04710	2011-10-06 11:58:46.0	60	59	62	0.976	30	100
110+04709	2011-10-06 11:58:46.0	65	61	63	0.689	30	100
110P04703	2011-10-06 11:58:46.0	60	61	64	0.432	30	100
110P04705	2011-10-06 11:58:46.0	62	60	63	0.532	30	100
110+04704	2011-10-06 11:58:46.0	62	59	64	1.009	30	100
110P04708	2011-10-06 11:58:46.0	66	61	64	0.345	30	99

Figure 6.5. INRIX raw data.



(a)

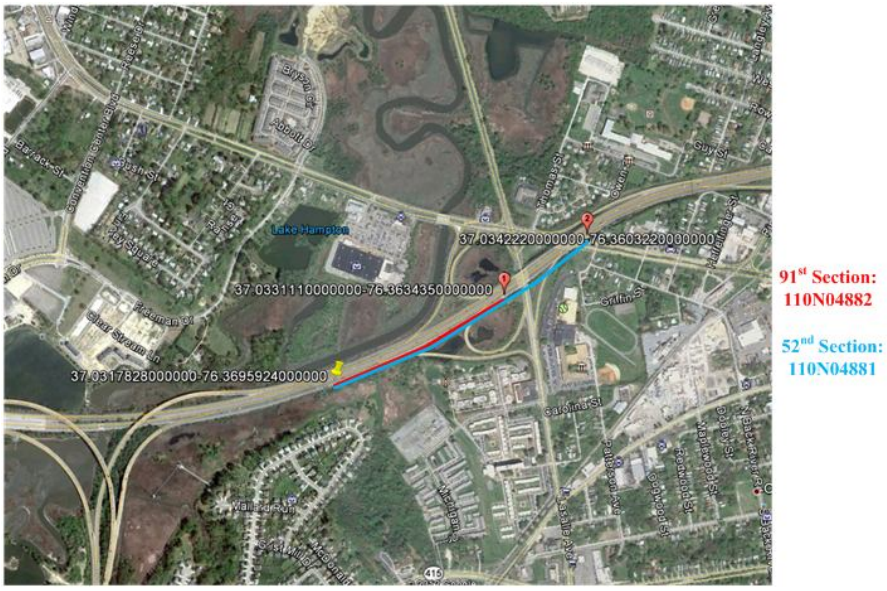
tmc	road	direction	intersection	state	county	zip	start_latitude	start_longitude	end_latitude	end_longitude	miles	Sort
110N04908	I-64	EASTBOUND	I-95/Exit 190	VA	RICHMON	23219	37.550763	-77.437978	37.55102	-77.428253	0.6342	1
110-04907	I-64	EASTBOUND	US-360/Exit 192	VA	RICHMON	23223	37.55102	-77.428253	37.558574	-77.415292	0.98349	2
110N04907	I-64	EASTBOUND	US-360/Exit 192	VA	RICHMON	23223	37.558574	-77.415292	37.5541808	-77.4050662	0.6411	3
110-04906	I-64	EASTBOUND	VA-33/Nine Mile Rd/Exit 193	VA	HENRICO	23223	37.5541808	-77.4050662	37.546013	-77.39553	0.78825	4
110N04906	I-64	EASTBOUND	VA-33/Nine Mile Rd/Exit 193	VA	HENRICO	23223	37.546013	-77.39553	37.54181	-77.390658	0.39496	5
110-04905	I-64	EASTBOUND	Laburnum Ave/Exit 195	VA	HENRICO	23231	37.54181	-77.390658	37.5305	-77.365128	1.73563	6
110N04905	I-64	EASTBOUND	Laburnum Ave/Exit 195	VA	HENRICO	23231	37.5305	-77.365128	37.532425	-77.356335	0.50433	7
110-04904	I-64	EASTBOUND	VA-156/Exit 197	VA	HENRICO	23150	37.532425	-77.356335	37.530097	-77.334342	1.21993	8
...												
110N04863	I-64	EASTBOUND	US-13/Northampton Blvd/Exit 282	VA	NORFOLK	23502	36.874567	-76.197334	36.870974	-76.195536	0.26813	87
110-04862	I-64	EASTBOUND	I-264/Exit 284	VA	NORFOLK	23502	36.870974	-76.195536	36.850876	-76.196226	1.38933	88
110N04862	I-64	EASTBOUND	I-264/Exit 284	VA	NORFOLK	23502	36.850876	-76.196226	36.841362	-76.196724	0.6567	89
110-04861	I-64	EASTBOUND	Indian River Rd/Exit 286	VA	VIRGINIA I	23464	36.841362	-76.196724	36.812615	-76.194894	1.98606	90
110N04882	I-64	EASTBOUND	VA-167/Lasalle Ave/Exit 265	VA	HAMPTON	23669	37.0317828	-76.3695924	37.033111	-76.363435	0.35345	91
110-04864	I-64	EASTBOUND	VA-165/Military Hwy/Exit 281	VA	NORFOLK	23502	36.882194	-76.21591	36.88215	-76.215767	0.00851	92

(b)

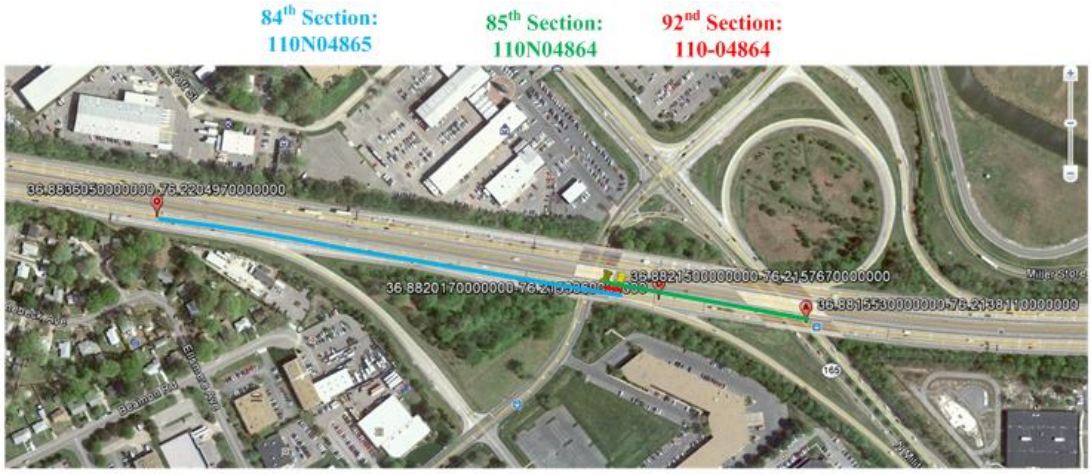
Figure 6.6. Sorted freeway sections along I-64.

A total of 92 sections are included in the mentioned I-64 data. However, two sections are not geographically consistent, as shown in Figure 6.7. The 91st section is overlaid by the

52nd section, and the 92nd section is overlaid by the 84th section. However, the 85th section shares a boundary with the 92nd section, not with the 84th section. Two overlaid sections are highlighted in yellow in Figure 6.6 (b). Both cases occur on the freeway ramps, and the research team believes that the traffic data for overlaid sections are collected to calculate the travel time leaving or coming from ramps on I-64. These cases will be used to adjust travel time computation accordingly to meet future needs.



(a)



(b)

Figure 6.7. Geographically inconsistent sample sections (source: Google map).

The raw data set is loaded as Coordinated Universal Time (UTC); therefore, an adjustment is needed to change the time zone to Eastern Standard Time (EST). Specifically, the UTC from January 1 to March 13 and November 7 to December 31

during 2010 entails subtracting 5 hours, while the remainder of the 2010 data entails subtracting 4 hours. The raw data of 2011 and 2012 are adjusted accordingly. The probe data collected for irregular time intervals present another problem for data analysis. As illustrated in Figure 6.8, the measurement time interval varies between 59 seconds to 6 minutes. The irregular time interval will increase difficulty of data aggregation and data estimation during this study. Since most raw data collected after October 2011 are measured in a 1-minute interval, this part of the data is aggregated by calculating the average speed every 1 minute. The 2010 raw data are aggregated every 5 minutes.

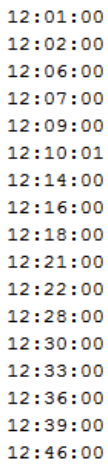


Figure 6.8. Sample irregular time interval of raw data.

When INRIX created freeway segments to collect traffic data, the fact that I-64 and US-60 run concurrently for some distance was not considered. As a result, the collected traffic data for I-64 tunnel segments were listed under US-60 until January 20, 2012. Therefore, the research team requested (through RITIS) all past INRIX data for the US-60 highway in order to obtain the tunnel traffic data. The tunnel section was marked by "TMC 110+14251" in the US-60 data set with a section length of 3.716 miles. The available data we can obtain about this section are also collected between October 7, 2011 and January 20, 2012. This data set will be used to fill the gap of the tunnel segment on I-64 for the first data set. It should be pointed out that the tunnel segment on I-64 was separated by five sections after January 20, 2012, instead of just one section. Therefore, the collected traffic data for the tunnel area provide more detailed information after January 20, 2012.

Unlike the above information – where the I-64 tunnel segment is represented by one section between October 7, 2011 to January 20, 2012, and five sections afterward – the tunnel in the 2010 INRIX data set includes three sections corresponding to TMC identification numbers "110+14251," "110P14251," and "110+14252." The fact of different spatial segment compositions and various data collection time intervals will be considered accordingly to construct the travel database and to develop a prediction algorithm.

6.1.2 Travel database construction

After solving the aforementioned problems of raw INRIX data, a spatiotemporal traffic speed map can be generated for each individual day in order to construct the travel database. To serve the purpose of travel time prediction, the travel database should include full daily coverage of spatiotemporal speed information. Such information will be used in the prediction algorithm to obtain similar traffic patterns between the current day and historical data and to forecast future traffic information. After constructing the daily spatiotemporal speed map on the INRIX data set, the problem of missing data is discovered to be a serious problem. The missing data problem varies with different data sets. The existence of missing data and the corresponding solutions are presented in this section. A selected 37-mile freeway stretch from Newport News to Virginia Beach is used to demonstrate the missing data problem. The same freeway stretch is used on the first case study to evaluate the proposed algorithm.

Considering the different data resolutions of spatial section composition and various data collection time intervals, two travel data sets are constructed as presented below.

Travel Database 1: Based on INRIX data from October 7, 2011, to November 30, 2012; represented by 96 sections across the space axis and a 1-minute time interval across the time axis.

Travel Database 2: Based on 2010 INRIX data; represented by 90 sections across the space axis and a 5-minute time interval across the time axis.

Generate Daily Spatiotemporal Speed Map for Travel Database 1

The freeway stretch used during this study includes sections along I-64 and I-264 from I-295 (east of Richmond) to I-264 (Virginia Beach). The entire 95-mile freeway is divided into 96 sections as shown in Figure 6.9. The section location is represented by mile posts starting from the first section in Richmond.

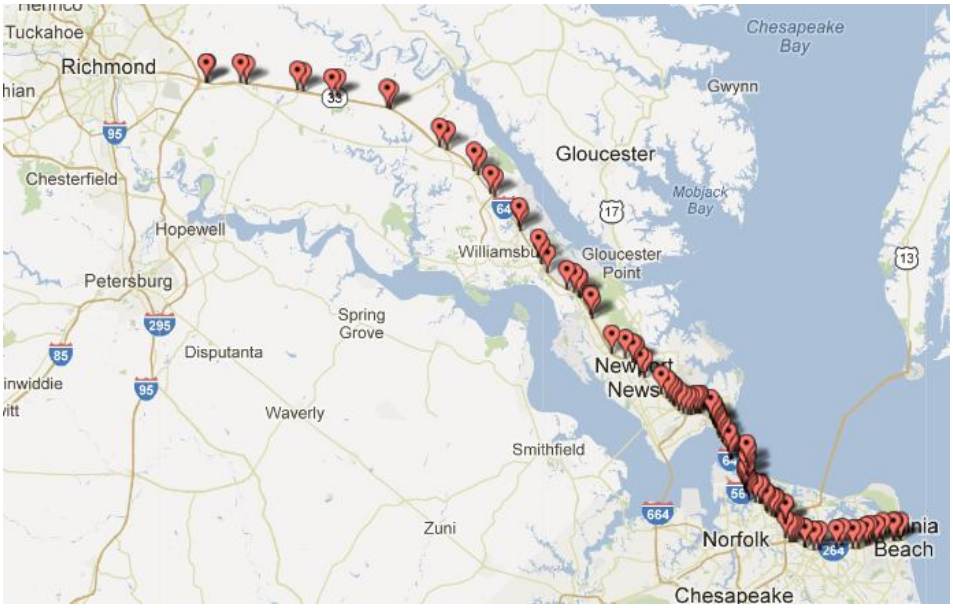


Figure 6.9. Freeway stretch for Travel Database 1.

Combining the data for I-64, I-264, and US-60, the speed data set along the entire freeway stretch is available from October 7, 2011, to November 30, 2012. After aggregating the speed data across space (section length) and time (1 minute), the daily temporal-spatial speed map is obtained. The speed map samples of typical weekday and weekend travel for Travel Database 1 are presented in Figure 6.10. The blue color represents free-flow speed, and the red color represents traffic congestion. It should be mentioned that the tunnel segment was covered by only one measurement for each time interval before January 20, 2012. This segment of data was filled by one section from US-60 as described in the Acquire INRIX Data section. For this case, the typical weekdays are represented as Figure 6.10 (a), (b), and (c). The typical weekend is represented as Figure 6.10 (d). Here we observe that the problem of missing data across the time space (between 1 to 20 minutes) occurs occasionally. The white strip areas of Figure 6.10 (a) and (c) demonstrate the missing data. The same approach of using neighboring data to estimate the missing measurement area is used for Travel Databases 1 and 2. Since the problem of missing data is more serious in Travel Database 2, detailed information of the estimation performance will be presented in the next section. Conversely, the tunnel segment was covered by five sections (five measurements for each time interval) since January 20, 2012. In this case, the typical weekday and weekend traffic maps are represented by Figure 6.10 (e) and (f), respectively. Moreover, the tunnel area includes more dynamic information compared with the blurred areas corresponding to the tunnel traffic status presented in Figure 6.10 (a) through (d).

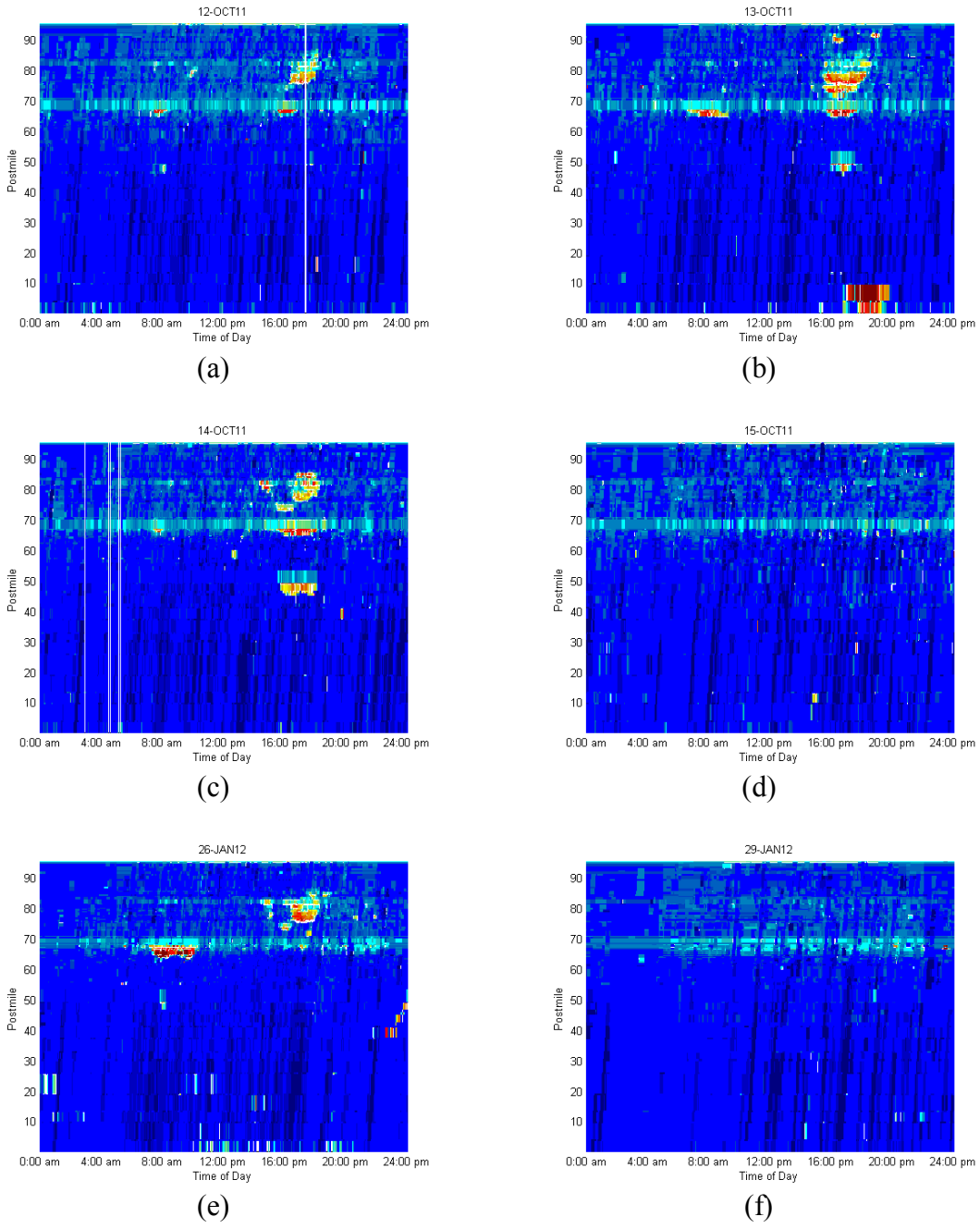


Figure 6.10. Samples of daily spatiotemporal speed map for Travel Database 1.

Generate Daily Spatiotemporal Speed Map for Travel Database 2

The same freeway stretch along I-64 and I-264 from I-295 (east of Richmond) to I-264 (Virginia Beach) is used to construct Travel Database 2. The entire freeway is divided by 90 sections instead of the 96 sections in Travel Database 1, as shown in Figure 6.11. The same mile post representation as with Travel Database 1 is used to maintain consistency.

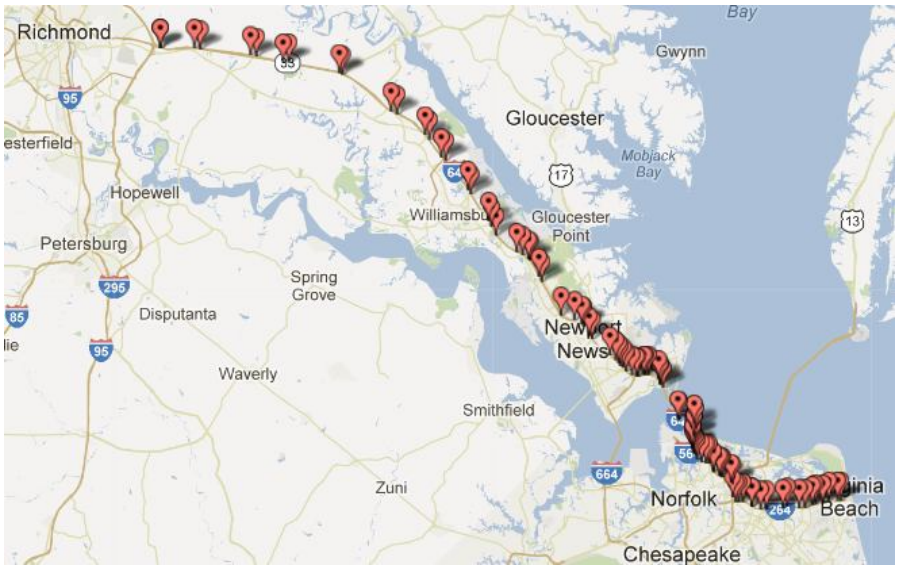


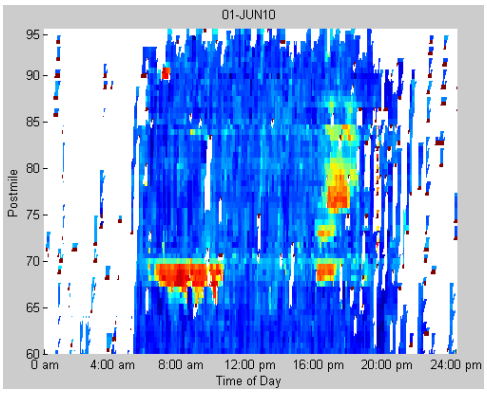
Figure 6.11. Freeway stretch for Travel Database 2.

The historical INRIX data for 2010 are analyzed using the same approach as for the 2011 data set except the time interval for the 2010 historical data is 5 minutes. We observe that traffic during weekends is usually uncongested in the 2011 data set. However, this differs for the summer season in the 2010 INRIX data, during which significant congestion is observed to occur during some weekends. Consequently, it is necessary to construct Travel Database 2 using 2010 INRIX data, especially if travel time predictions during the summer season should be investigated. To better illustrate the detailed traffic status for the 2010 INRIX data, a 37-mile freeway stretch is selected for Travel Database 2 that includes most of the congested areas along the entire freeway stretch. The selected freeway stretch is located between Newport News and Virginia Beach from the 33rd section to the 90th section. The same stretch is also used in the first case study for algorithm testing.

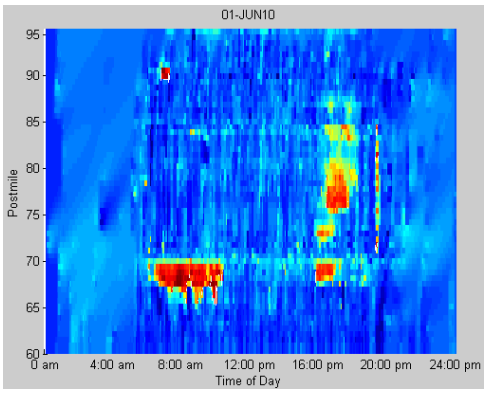
The data samples for typical weekday and weekend traffic occurring during June 2010 are presented in Figure 6.12. The figure illustrates a significant amount of missing data, especially for June 5 and June 6, 2010 (Saturday and Sunday). It appears from inspection of the data that the weekends involve more missing data than is the case for the weekdays, which may pose a problem, especially when making travel time predictions for weekend days. According to the speed map of Figure 6.12 (a) and (c), most missing data (white area) for a typical weekday occur between 21:00 p.m. and 5:00 a.m. (i.e., during the night and early morning hours). Normally there is low traffic volume during this time period, and free-flow speed could be assumed. However, sometimes the missing data also occur around a congested area (e.g., the speed map of Figure 6.12 e and g). Consequently, free-flow speed cannot be simply assumed for all missing data.

As demonstrated in the Literature Review section, various traffic data estimation algorithms are developed depending on the data resource. Since ramp traffic data are not available, greater errors will be introduced to macroscopic traffic models for estimating

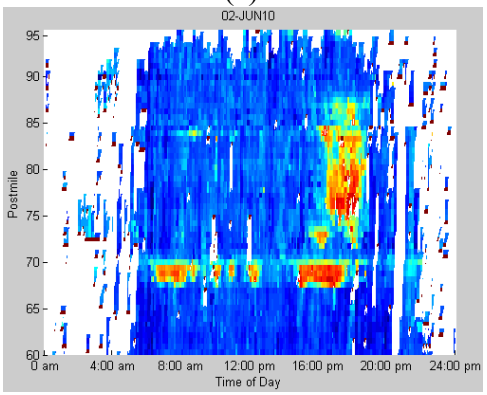
missing data. Alternatively, a statistical approach is employed here that uses temporal and spatial speed values around missing data. The average value of eight neighboring cells is used to estimate the missing speed data. Advanced approaches such as using kernel regression over spatial and temporal coordinates can be considered in the future. The samples of estimated speed maps for typical weekday and weekend traffic in June 2010 are presented in the right-hand columns of Figure 6.12.


01-JUN10
Postmile
95
90
85
80
75
70
65
60
0 am
4:00 am
8:00 am
12:00 pm
16:00 pm
20:00 pm
24:00 pm
Time of Day

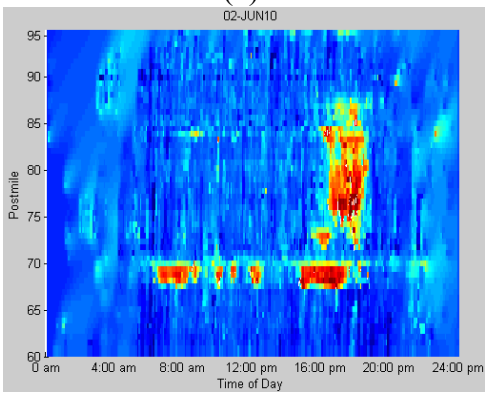
(a)


01-JUN10
Postmile
95
90
85
80
75
70
65
60
0 am
4:00 am
8:00 am
12:00 pm
16:00 pm
20:00 pm
24:00 pm
Time of Day

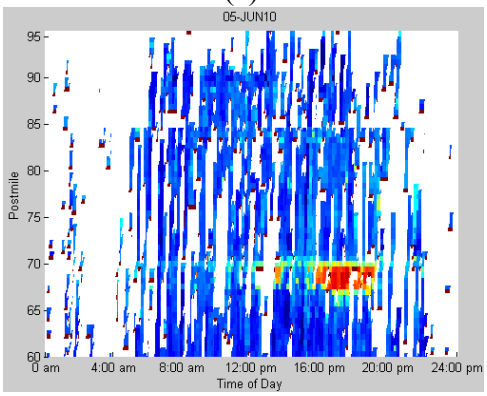
(b)


02-JUN10
Postmile
95
90
85
80
75
70
65
60
0 am
4:00 am
8:00 am
12:00 pm
16:00 pm
20:00 pm
24:00 pm
Time of Day

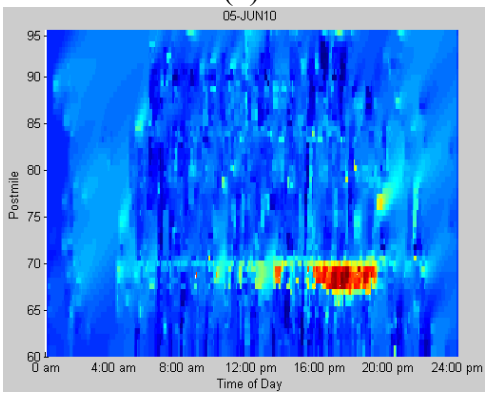
(c)


02-JUN10
Postmile
95
90
85
80
75
70
65
60
0 am
4:00 am
8:00 am
12:00 pm
16:00 pm
20:00 pm
24:00 pm
Time of Day

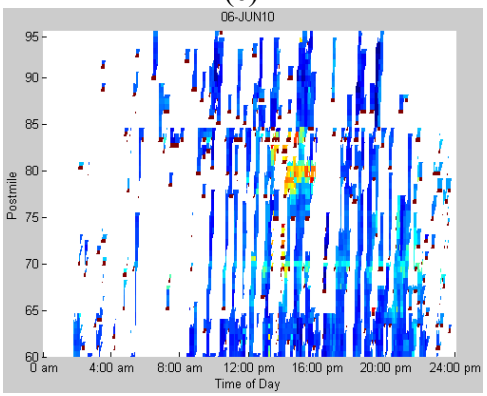
(d)


05-JUN10
Postmile
95
90
85
80
75
70
65
60
0 am
4:00 am
8:00 am
12:00 pm
16:00 pm
20:00 pm
24:00 pm
Time of Day

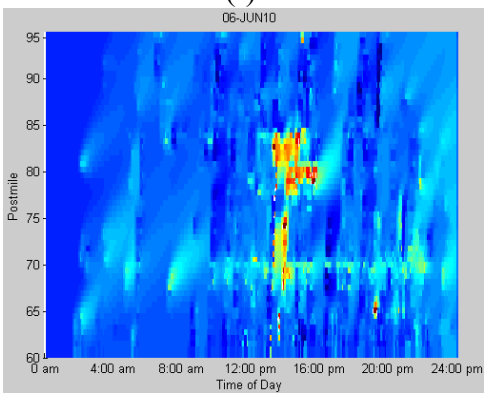
(e)


05-JUN10
Postmile
95
90
85
80
75
70
65
60
0 am
4:00 am
8:00 am
12:00 pm
16:00 pm
20:00 pm
24:00 pm
Time of Day

(f)


06-JUN10
Postmile
95
90
85
80
75
70
65
60
0 am
4:00 am
8:00 am
12:00 pm
16:00 pm
20:00 pm
24:00 pm
Time of Day

(g)


06-JUN10
Postmile
95
90
85
80
75
70
65
60
0 am
4:00 am
8:00 am
12:00 pm
16:00 pm
20:00 pm
24:00 pm
Time of Day

(h)

Figure 6.12. Samples of daily spatiotemporal speed map for Travel Database 2.

6.2 Algorithm development

6.2.1 The framework of dynamic travel time prediction

The proposed algorithm comprises three stages: identify current traffic status, select similar traffic patterns from historical data, and predict travel times. The framework of the three stages is demonstrated in Figure 6.13. The current traffic status is initially selected to represent the traffic status of all freeway sections from short-past to the current time interval. The traffic status in this case is a matrix across temporal and spatial axes. Thereafter, the historical traffic speed data with the same dimension to current traffic status is selected as a candidate. Based on the dissimilarity to the current speed matrix, several candidates are extracted to represent the historical recurrent traffic patterns that are similar to the current status. Finally, the subsequent dynamic travel times of those candidates are aggregated to represent the travel time distributions in the future.

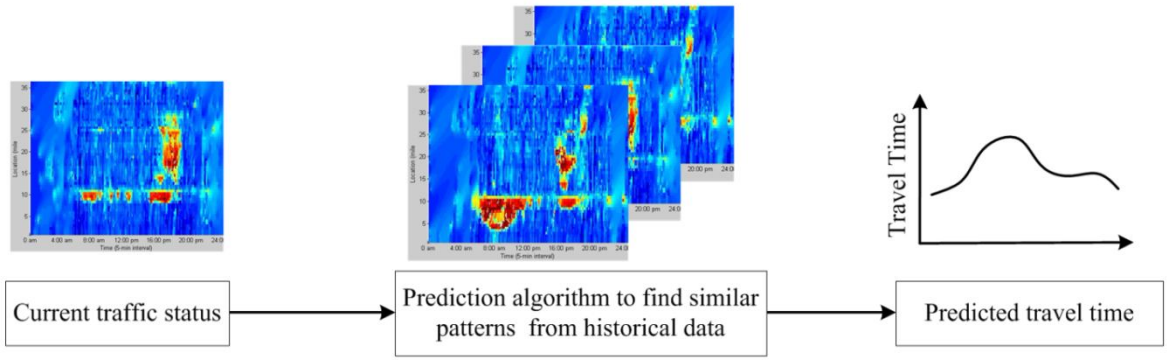


Figure 6.13. The framework of travel time prediction by pattern recognition.

Matching traffic patterns

A candidate selection scheme is proposed to select temporal-spatial traffic state candidates from a historical data set by matching with the real-time traffic state. Suppose c denotes the current time; the current traffic state $[c-L+1, c-L+2, \ldots, c]$ and the matching temporal-spatial traffic data $[t-L+1, t-L+2, \ldots, t]$ from a historical data set are denoted by tail time c and t, respectively. Here, L is the data length across time intervals to be matched. It should be noted that the traffic data of each time interval is a vector that covers all spatial sections (N sections) of the freeway stretch; therefore, the traffic data for L time intervals is a matrix with dimension L by N. Various template matching methods can be used to define the dissimilarity between the current traffic status and historical data, such as the Euclidean distance [78-81], data trends [12, 82], image pattern recognition [83, 84], neural networks [32, 85], etc. In this study, the average Euclidean distance between the current temporal-spatial traffic data and each data matrix with the

same dimension from the historical data set is calculated using Equation (6.1) to represent a dissimilarity measure. Other advanced methods can be adopted to increase the matching speed and accuracy and are being considered as part of future research efforts.

$$d(c,h) = \left| M(c,L) - M(h,L) \right| \Big/ \left(L \times N \right). \tag{6.1}$$

where *M(c,L)* and *M(h,L)* represent the traffic data of the current and historical time intervals, respectively; and *d(c,h)* is the average Euclidean distance between the traffic speed matrix data of different time intervals.

A small dissimilarity measure indicates that the matching historical data are similar to the current traffic pattern. Consequently, several candidates are selected according to the ascending order of the dissimilarity measure. Here, the maximum number of candidates is denoted by K, and the minimum acceptable dissimilarity is defined by d_{MIN}. The set of candidates H_c is selected as

$$\begin{aligned} H_c \quad &= \quad \{h_1, h_2, \cdots, h_{K'},\} \\ where \quad & \quad h_1 = \arg\min d(c,h) \\ & \quad d(c,h_i) \le d(c,h_{i+1}) \\ & \quad K' = \max\{i \mid i \le K, d(c,h_i) \le d_{MIN}\} \\ & \quad \left| h_i - h_j \right| \le \varepsilon, \qquad i \ne j \end{aligned} \quad . \tag{6.2}$$

where h_i is the selected candidate from the historical data set; K' denotes the resulting number of the selected candidates; and ε is used to avoid selecting adjacent candidates from the same day in the history data. The selected candidates represent the best matching to the current traffic status and will be used to calculate future travel times.

Dynamic travel time prediction

The future dynamic travel times on the current day can be calculated based on the selected historical candidates. Considering the stochastic nature of a traffic system, the travel time prediction problem can be recognized as a time series prediction for nonlinear dynamic (chaotic) systems [86, 87]. The future traffic state for the current day can be predicted by the subsequent traffic state of each candidate from the historical data set. The linear combination of each candidate's subsequent traffic state is used to predict the future traffic status, and the corresponding weight is defined as the inverse of the dissimilarity measure of each candidate. The prediction traffic state starting from time interval $c+p$ is obtained as

$$M(c+p) = \sum_{i=1}^{K'} w(h_i) \cdot M(h_i + p) \quad . \tag{6.3}$$

$$w(h_i) = \frac{d(c,h_i)^{-1}}{\sum_{i=1}^{K'} d(c,h_i)^{-1}}. \tag{6.4}$$

where $M(h_i+p)$ represents the p steps ahead subsequent traffic state for i^{th} candidate; and $w(h_i)$ denotes the weight of i^{th} candidate data.

The next step is to calculate the dynamic travel time based on the subsequent traffic state of each candidate. Dynamic travel time is the actual, realized travel time that a vehicle could experience during a trip. If a vehicle leaves a trip origin at the current time, the roadway speed will not only change across space but also across time during the entire trip. Therefore, the traffic state evolution over space and time is considered in our approach, as shown in Figure 6.14 to compute dynamic travel times. The speed values of the shaded cells are used to compute dynamic travel times. In this paper, the traffic state is assumed to be homogenous within each cell. Therefore, the trajectory slope, which represents the traffic speed, is a constant value in each cell. Assume that the trip starts from time interval t_n. In this way, once the vehicle enters a new cell, the trajectory within this cell can be drawn as the straight dotted line in Figure 6.14, with the slope value equal to the traffic stream speed. Finally, the dynamic travel time can be calculated when the trajectory reaches the downstream boundary of the last freeway section (destination).

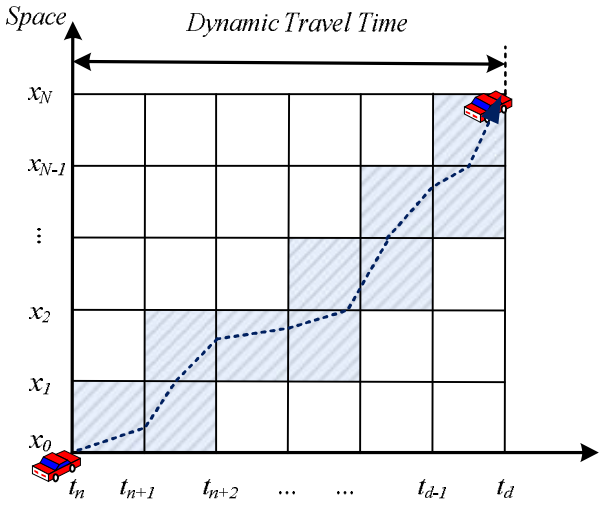


Figure 6.14. Illustration of dynamic travel time.

The procedure for computing dynamic travel times is shown in Figure 6.14. Consequently, the dynamic travel time of each subsequent candidate can be obtained and the corresponding weight (recurrent probability) is defined by the dissimilarity measure of Equation (6.5). Finally, the travel time distribution of the future trip can be represented as

$$TT(c+p) = \{TT(h_i+p), w(h_i) \mid i = 1, \cdots, K'\}. \tag{6.5}$$

where $TT(c+p)$ represents the dynamic travel time starting from time interval $c+p$; and $TT(h_i+p)$ denotes the subsequent travel time of i^{th} candidate, according to the calculation as shown in Figure 5.4. The travel time prediction result can also be calculated as the average value using

$$\overline{TT}(c+p) = \sum_{i=1}^{K'} w(h_i) \cdot TT(h_i + p) \quad (6.6)$$

An illustration of how to use current traffic status to find historical candidates and then predict travel time is shown in Figure 6.15. The testing day is August 4, 2010 and the current time is 16:00 p.m. The current traffic status is represented by a matrix between the first and second vertical lines along all the sections. The template matching algorithm is implemented to find three historical data sets (May 28, 2010, August 2, 2010 and August 3, 2010) with a similar traffic pattern to the current traffic status. The image strips between the first and second lines on all four days include very similar congestion patterns (represented by red color) around the 50th section. Consequently, the future spatiotemporal traffic states from the selected candidates are used to represent the future traffic status of current day. This deduction can be supported by the fact that the image strips between the second and third lines on four days are very similar.

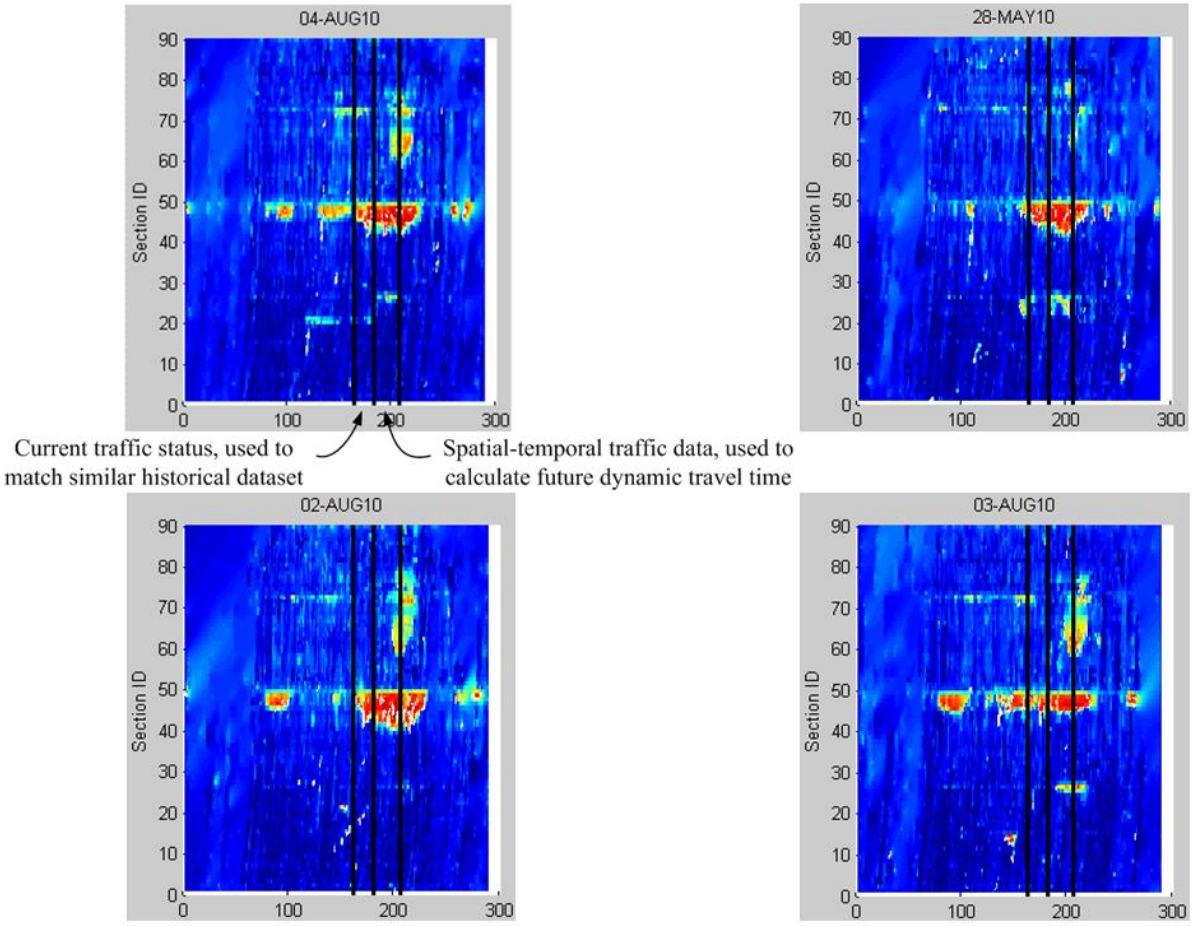


Figure 6.15. Samples of current traffic status and selected candidates.

6.2.2 Revised algorithm

Based on the first case study of using the proposed approach to predict travel time, several improvements have been adopted to our algorithm to improve the prediction accuracy. The revised prediction algorithm will be tested on the second and third case studies, and the detailed information of the revised algorithm is presented below:

1) The value of L is correlated to the current instantaneous travel time instead of using a fixed value for every time interval. For instance, a small value of L is used during uncongested time intervals and a large value is used for congested time intervals. The criterion is computed based on the value of the instantaneous travel time.

2) The selected candidate number K' is also a variable correlated to the instantaneous travel time. During uncongested conditions, K' is set to be a large value in order to obtain a smoothed aggregation value from historical candidates. During time periods of high congestion, a smaller number of candidates is used since there are limited high congested data in our existing database. This may not be a problem in the future when the historical data set increases.

3) During the process of template matching for each day in the historical data set, the data slice of the best fit (least matching cost) is selected. Moreover, the searching range is constrained to a fixed 2-hour window around the current time c. Therefore, only one data slice is selected for each day in the historical data set and, finally, K' candidates are selected from all data slices.

4) During the calculation of dissimilarity, a weight parameter is introduced to utilize the length of each section to scale the corresponding dissimilarity value. This is based on the fact that the same level of congestion on a long section compared to a short section should have more effect on the total dissimilarity calculation.

6.3 Algorithm test

6.3.1 Test environment

This section aims to investigate the performance of the proposed travel time prediction algorithm. In total, three case studies are conducted and the proposed algorithm is compared with different methods using the second travel database from 2010. The first travel database from October 2011 to November 2012 is expected to fuse with the 2010 data set to conduct more extensive tests in the future study. Since heavy traffic volume is usually observed along I-64 and I-264 heading to Virginia Beach during the summer season and on weekends, efficient and accurate travel time prediction can be helpful to travelers in planning their trips and reducing traffic congestion around the area.

Considering that most of the congestion areas between Richmond and Virginia Beach are located before the Hampton Roads Bridge-Tunnel or along I-264, a 37-mile freeway stretch is selected as the test site for the first two case studies before the extensive testing on the entire 95-mile freeway. The 37-mile freeway stretch selected is from Newport News to Virginia Beach along I-64 and I-264 and includes 59 sections as shown in Figure 6.16. Eventually, the third case study is conducted on the entire 95-mile freeway from Richmond to Virginia Beach. The detailed information of experiment setups and discoveries are provided as below.

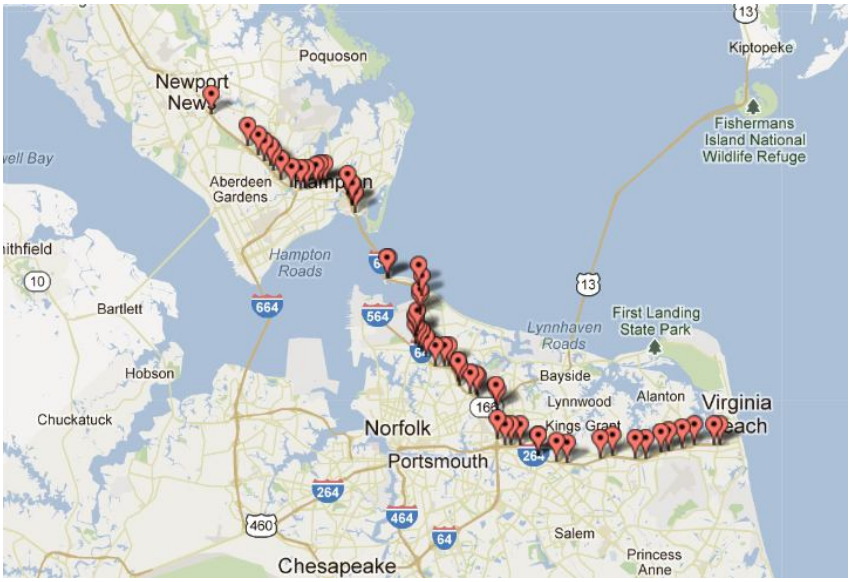


Figure 6.16. Selected 37-mile freeway stretch for case studies 1 and 2.

6.3.2 Case study 1

Because traffic congestion for the selected freeway stretch is significant during the summer holiday season and on weekends, the evaluation of the travel-time prediction algorithm focuses on traffic data from June to August of 2010 in this case study. Traffic data from June and July are used for the training data set and the August data are used for the testing data set. The dynamic travel time is calculated every 5 minutes using the daily spatiotemporal traffic speed map, as shown in Figure 6.14, which serves as the ground truth data. The prediction span p equals zero for this test, which indicates that the future trip starting from the current time is the prediction output. The average travel time is predicted using Equation (6.6).

Different parameters are tested to identify the best combination to minimize the prediction error. The range of L, which represents the data length across the time axis (look ahead time duration), is between 10 to 60 minutes at 10-minute intervals. H is another parameter representing the shift distance across the time space when searching for a traffic map slice from the historical data set (look back duration). The size of H should not be too small. Otherwise, many overlapping map slices may be extracted for comparison to the current traffic map, and the computation time would be significant.

Conversely, detailed information may be ignored if the value of *H* is too large. Therefore, the domain of the *H* value is also tested from 10 to 60 minutes at 10-minute increments.

The relative and absolute errors (MAPE and MAD) calculated by the proposed method across various parameters are presented in Table 6.1 and Table 6.2, respectively. Both the minimum relative error of 5.96% and the minimum absolute error of 2.96 min are obtained assuming that L = 20 minutes and H = 40 minutes. According to the tables, prediction errors are comparatively stable values of 6% and 3 min when L is less than 40 minutes. The change of the *H* value seems to have little impact on the average prediction accuracy under this situation. The optimum values of parameters can be used as a reference for applications on different sites.

Table 6.1: Relative errors by proposed travel time prediction method.

MAPE (%)		Time Interval of H (min.)					
		10	20	30	40	50	60
Time Interval of L (min)	10	6.09	5.98	6.13	5.98	6.00	6.03
	20	6.07	6.01	6.14	5.96	5.99	6.05
	30	6.17	6.05	6.14	5.99	5.97	5.98
	40	6.24	6.12	6.14	6.10	6.06	6.02
	50	6.27	6.15	6.20	6.15	6.21	6.12
	60	6.37	6.32	6.31	6.25	6.33	6.20

Table 6.2. Absolute errors by proposed travel time prediction method.

MAD (min.)		Time Interval of H (min.)					
		10	20	30	40	50	60
Time Interval of L (min)	10	3.02	2.98	3.05	2.99	2.99	3.00
	20	3.05	3.00	3.06	2.96	3.00	3.02
	30	3.11	3.04	3.08	3.01	3.00	3.00
	40	3.15	3.08	3.08	3.07	3.04	3.03
	50	3.17	3.09	3.11	3.10	3.12	3.09
	60	3.22	3.19	3.18	3.14	3.19	3.14

To better evaluate the proposed method in this case study, a traditional *k*-NN algorithm [12, 34] is tested to predict travel time by applying the same training and testing data sets. However, instantaneous travel time is used in the *k*-NN method instead of dynamic travel times as is used in the literature. Assuming the purpose is to predict the travel time starts from time interval *t*, the traditional *k*-NN method uses the travel time sequence between recent past *t-L* and time interval *t-1* to find a similar data sequence in the historical data set. However, the dynamic travel time of the recent past travel time sequence may not be available since the trip has not been completed (the travel time is around 38 min under free-flow speed for the selected 37-mile freeway stretch). Therefore, instantaneous travel

times between time interval *t-L* and *t-1* are used in the *k*-NN method to predict travel time in the next time interval *t*.

Table 6.3: Relative errors by *k*-NN method.

MAPE (%)		Time Interval of H (min.)					
		10	20	30	40	50	60
Time Interval of L (min)	10	6.80	6.68	6.85	6.68	6.70	6.74
	20	6.78	6.71	6.86	6.85	6.69	6.76
	30	6.61	6.59	6.61	6.69	6.66	6.68
	40	6.97	6.84	6.86	6.81	6.77	6.73
	50	7.01	6.87	6.93	6.87	6.94	6.83
	60	7.11	7.06	7.05	6.98	7.07	6.92

Table 6.4: Absolute errors by *k*-NN method.

MAD (min.)		Time Interval of H (min.)					
		10	20	30	40	50	60
Time Interval of L (min)	10	3.51	3.53	3.55	3.51	3.53	3.52
	20	3.52	3.49	3.51	3.50	3.53	3.56
	30	3.52	3.48	3.47	3.51	3.56	3.54
	40	3.56	3.58	3.54	3.60	3.59	3.64
	50	3.59	3.61	3.58	3.67	3.64	3.68
	60	3.67	3.64	3.64	3.68	3.71	3.73

The same 10 closest candidates are selected using the Euler distance to calculate the average travel time for the future trip. The relative and absolute errors calculated by the proposed method across various parameters are presented in Table 6.3 and Table 6.4, respectively. The optimum parameter of *L*, which represents the domain of continuous time included in the traffic map slice, is 30 minutes; the corresponding minimum relative and absolute prediction errors are 6.59% and 3.47 min, respectively. Therefore, the average performance of the proposed method includes fewer errors compared to the traditional *k*-NN method for our database. The main difference between the two methods is that the travel time sequence is used to obtain similar traffic patterns from historical data in the *k*-NN method, while the traffic status across the spatial and temporal axes are used in the proposed method. The spatiotemporal traffic status provides more travel information given that it accounts for the spatial variation in the information. Therefore, such information serves a better pattern-matching result from the historical data and results in a more accurate travel time prediction performance. Moreover, the instantaneous travel time predicted by the *k*-NN method may deviate substantially from the dynamic travel time under transient states during the trip. Based on the testing results, we observed that the predicted travel time using the *k*-NN method is usually

underestimated when congestion is forming and is overestimated when congestion is dissipating.

A comparison of the two methods for a typical weekday (i.e., August 2, 2010) is presented in Figure 6.17. The typical weekday traffic occurring on the selected 37-mile freeway stretch usually includes two peak hours during the morning and afternoon peaks. The traffic congestion is especially serious during afternoon peak hours. The ground truth curve in Figure 6.17 indicates that the travel time during this period could be more than two times (78 minutes) the travel time occurring during a free-flow period (38 minutes). The red curve obtained from the proposed method is a better fit to the ground truth data for congested and uncongested time periods. However, the blue curve obtained by the *k*-NN method underestimates the actual travel time during congested afternoon periods and overestimates the actual travel time as the peak ends around 18:00 p.m. Consequently, the proposed method produces more accurate travel time prediction results compared to the *k*-NN method for the subject day.

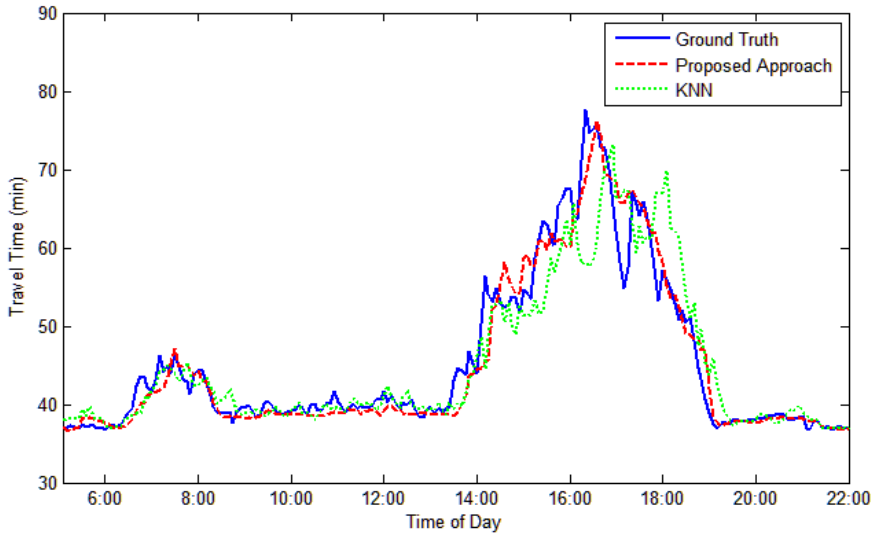


Figure 6.17: Comparison of prediction results for a typical weekday (August 2, 2010).

Another comparison of the two methods for typical weekend traffic occurring on August 7, 2010, is presented in Figure 6.18. Unlike typical weekday traffic, light traffic congestion occurs during the weekend that lasts for an extended time as many travelers go to Virginia Beach for that time period. Although the prediction accuracy is almost the same for this day when using the two methods, the green curve calculated by the traditional *k*-NN approach also indicates that the deviation from ground truth data happens under transient states during which congestion is forming or dissipating.

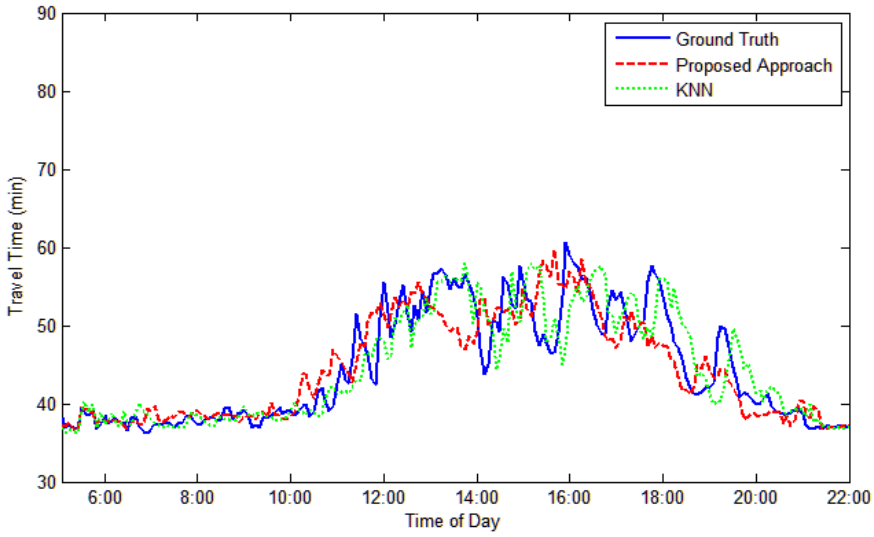


Figure 6.18: Comparison of prediction results for a typical weekend (August 7, 2010).

6.3.3 Case study 2

Based on the results of case study 1, several improvements are adopted to improve the prediction accuracy of the proposed method. Consequently, a revised travel time prediction algorithm is proposed with the detailed revisions that have been described earlier. The selected 37-mile freeway stretch from Newport News to Virginia Beach is also used to investigate the performance of the revised prediction algorithm. Since serious congestion usually happens during the summer season on I-64 and I-264 heading to Virginia Beach, the traffic data on August 2010 is used as the testing data set. All the previous traffic data from April to July 2010 are used as the historical data set.

In order to better evaluate the performance of the proposed approach, three other methods from the previous studies are also tested on the same data set. Descriptions of all prediction methods are presented below.

1) Method 1: summation of the real-time traffic data across all the segments of freeway stretch.

2) Method 2: Kalman filter approach using the travel times from previous time intervals to define the transition function.

3) Method 3: k-nearest-neighbor method to predict the travel time by the selected similar travel time sequence from the historical data set.

4) Method 4: the proposed revised travel time prediction algorithm.

As with the previous case study, both absolute (mean absolute errors [MAE]) and relative (MAPE) errors are used to measure the prediction accuracy. The prediction results of the

four methods are presented in Table 6.5. The prediction results are calculated for different days of a week (from Monday to Sunday), and the proposed method has the fewest prediction errors, except for Wednesday and Sunday. The average errors demonstrate that the proposed method is the best travel time prediction method. The best method for each row is highlighted by the blue color.

Table 6.5: Prediction results of four methods.

	Method1		Method2		Method3		Method4	
	MAE(min)	MAPE(%)	MAE(min)	MAPE(%)	MAE(min)	MAPE(%)	MAE(min)	MAPE(%)
Monday	2.07	4.57	2.37	5.25	1.99	4.44	1.68	3.75
Tuesday	2.14	4.86	2.47	5.60	2.17	4.91	2.02	4.48
Wednesday	2.70	5.77	3.06	6.53	2.63	5.64	2.70	5.76
Thursday	2.68	5.84	2.99	6.58	2.71	5.90	2.31	4.87
Friday	2.90	5.94	3.41	6.99	2.80	5.75	2.64	5.45
Saturday	2.64	5.69	3.23	6.94	2.61	5.62	2.45	5.22
Sunday	1.29	3.19	1.51	3.74	1.27	3.15	1.44	3.55
Average	2.34	5.12	2.72	5.95	2.31	5.06	2.18	4.73

Considering the real application, the accurate travel time prediction during congested time periods is more important to the traveler, since the traveler can change the trip schedule accordingly to avoid getting stuck in a traffic jam. On the other hand, the travel time prediction during uncongested time periods attracts less concern since it does not have much of an effect on the traveler's trip. In order to investigate the prediction results for congested and uncongested time periods, the absolute and relative errors are aggregated by every 2-hour interval, as presented in Table 6.6.

Table 6.6: Prediction results of four methods for different time periods.

		Method1		Method2		Method3		Method4	
		MAE(min)	MAPE(%)	MAE(min)	MAPE(%)	MAE(min)	MAPE(%)	MAE(min)	MAPE(%)
Monday	6 am - 8 am	1.63	3.77	1.95	4.55	1.58	3.61	1.91	4.42
	8 am - 10 am	1.10	2.73	1.42	3.54	1.15	2.86	0.91	2.28
	10 am - 12 pm	0.98	2.52	1.27	3.26	0.96	2.47	0.96	2.45
	12 pm - 14 pm	1.07	2.68	1.31	3.31	1.08	2.71	1.23	3.05
	14 pm - 16 pm	1.89	4.10	2.05	4.50	1.90	4.08	1.49	3.25
	16 pm - 18 pm	4.78	9.04	5.33	9.97	4.40	8.43	3.66	6.95
	18 pm - 20 pm	3.03	7.17	3.24	7.63	2.89	6.92	1.62	3.87
Tuesday	6 am - 8 am	1.78	4.15	2.16	5.07	1.78	4.13	1.64	3.89
	8 am - 10 am	1.32	3.36	1.57	3.97	1.41	3.56	1.18	2.97
	10 am - 12 pm	1.49	3.60	1.90	4.64	1.44	3.50	1.52	3.59
	12 pm - 14 pm	1.52	3.66	1.88	4.49	1.46	3.52	1.58	3.70
	14 pm - 16 pm	2.58	5.54	2.88	6.25	2.58	5.53	2.76	5.94
	16 pm - 18 pm	4.35	8.61	4.93	9.64	4.49	8.90	4.32	8.23
	18 pm - 20 pm	1.98	5.13	1.97	5.12	2.01	5.23	1.16	3.04
Wednesday	6 am - 8 am	1.46	3.57	1.81	4.44	1.41	3.46	1.51	3.75
	8 am - 10 am	2.07	4.60	2.19	4.96	2.06	4.59	2.20	4.90
	10 am - 12 pm	3.52	7.29	4.11	8.46	3.54	7.29	3.43	6.75
	12 pm - 14 pm	2.29	4.96	3.08	6.63	1.98	4.32	2.51	5.20
	14 pm - 16 pm	3.01	6.17	3.14	6.43	3.08	6.34	3.06	6.35
	16 pm - 18 pm	4.49	8.65	5.00	9.51	4.31	8.33	4.49	8.98
	18 pm - 20 pm	2.05	5.17	2.10	5.31	2.05	5.15	1.70	4.38
Thursday	6 am - 8 am	2.81	5.32	2.91	5.73	2.82	5.34	2.93	5.51
	8 am - 10 am	3.40	8.01	3.79	8.78	3.40	8.07	2.28	4.79
	10 am - 12 pm	1.11	2.82	1.56	3.96	1.14	2.88	1.07	2.68
	12 pm - 14 pm	1.42	3.53	1.60	3.97	1.44	3.59	1.70	4.17
	14 pm - 16 pm	2.42	5.33	2.65	5.99	2.50	5.49	2.71	5.90
	16 pm - 18 pm	4.44	8.27	5.02	9.34	4.60	8.56	3.57	6.48
	18 pm - 20 pm	3.14	7.62	3.37	8.26	3.03	7.38	1.88	4.56
Friday	6 am - 8 am	1.64	4.02	1.87	4.66	1.54	3.80	1.60	3.92
	8 am - 10 am	1.63	4.04	2.00	4.96	1.60	3.97	1.47	3.60
	10 am - 12 pm	1.57	3.76	1.85	4.43	1.56	3.74	1.83	4.38
	12 pm - 14 pm	2.66	5.75	3.18	6.89	2.66	5.79	3.40	7.13
	14 pm - 16 pm	4.57	8.54	5.32	9.89	4.18	7.84	3.85	6.96
	16 pm - 18 pm	4.61	7.78	5.73	9.80	4.35	7.30	4.21	6.86
	18 pm - 20 pm	3.58	7.71	3.89	8.30	3.67	7.85	3.23	6.67
Saturday	6 am - 8 am	1.03	2.72	1.14	3.00	1.00	2.63	0.72	1.90
	8 am - 10 am	0.90	2.36	1.14	2.96	0.94	2.44	0.88	2.28
	10 am - 12 pm	1.66	3.93	2.04	4.90	1.58	3.75	1.56	3.71
	12 pm - 14 pm	2.94	6.19	4.00	8.40	2.86	6.02	2.69	5.70
	14 pm - 16 pm	3.53	7.10	4.29	8.67	3.46	6.96	3.39	7.02
	16 pm - 18 pm	4.14	7.98	5.03	9.66	4.11	7.89	4.71	8.91
	18 pm - 20 pm	4.31	9.56	4.96	10.96	4.34	9.66	3.18	7.04
Sunday	6 am - 8 am	0.67	1.78	0.78	2.06	0.70	1.86	0.76	1.99
	8 am - 10 am	0.72	1.91	0.79	2.08	0.72	1.90	0.88	2.33
	10 am - 12 pm	0.94	2.44	1.11	2.89	0.92	2.41	0.81	2.11
	12 pm - 14 pm	1.06	2.66	1.34	3.38	1.02	2.56	1.12	2.74
	14 pm - 16 pm	1.65	3.98	1.91	4.64	1.55	3.77	2.12	5.11
	16 pm - 18 pm	1.29	3.25	1.59	4.03	1.25	3.17	1.59	4.06
	18 pm - 20 pm	2.69	6.29	3.05	7.08	2.74	6.40	2.83	6.50

According to the results in Table 6.6, the proposed method has the fewest prediction errors for most of the congested time periods, especially for evening peak hours (14 p.m. - 20 p.m.). For the uncongested time period, or the cases in which the proposed method does not work well, method 1 and method 3 produce the fewest errors. The detailed comparisons of prediction accuracy by different methods for every time period are demonstrated below.

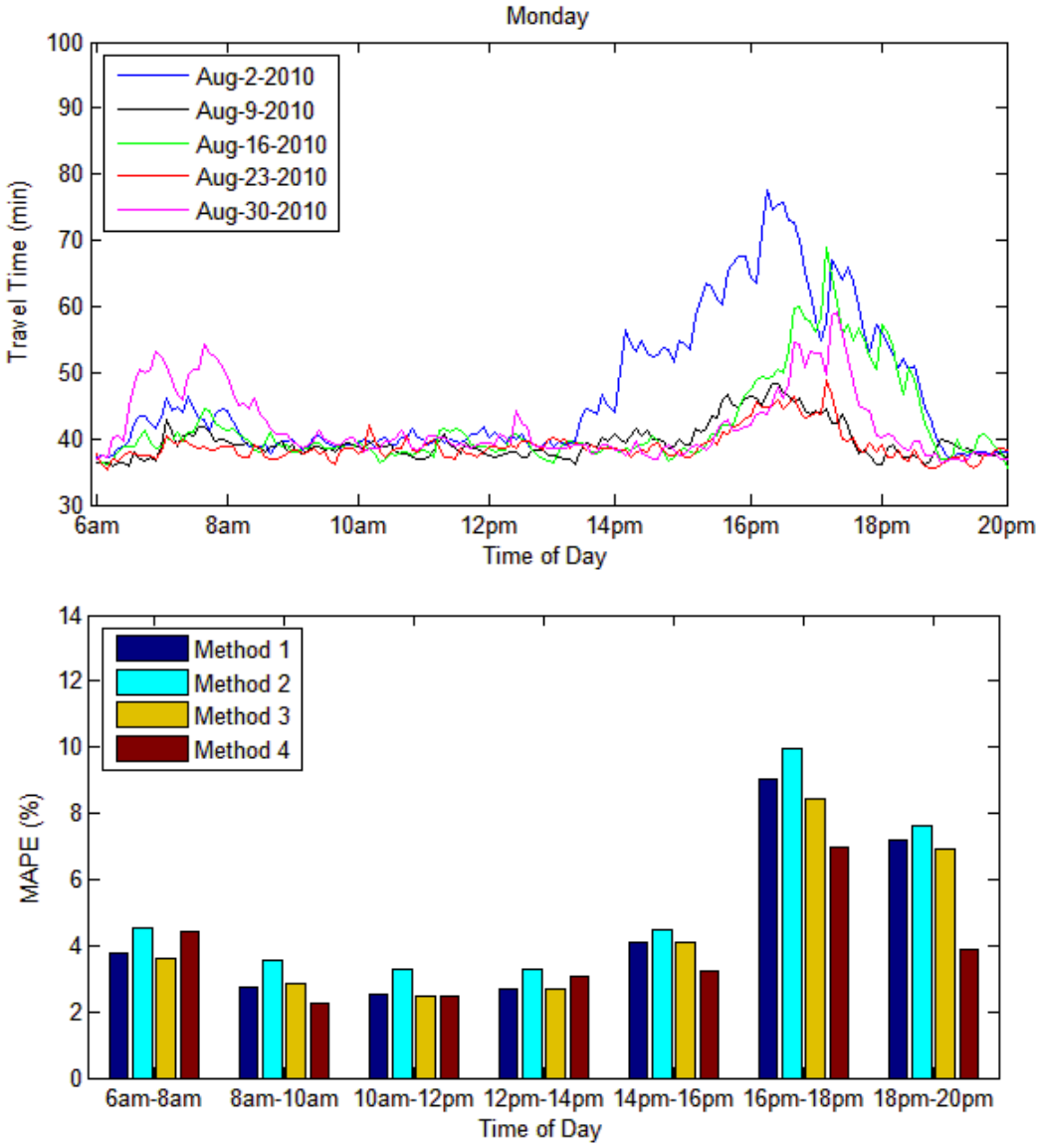


Figure 6.19: The ground truth travel time data and the comparison of four methods for every time period on Monday.

According to the travel time curves on Monday, there is mitigated congestion during morning peak hours (6 a.m. - 9 a.m.) and heavy congestion during afternoon peak hours (14 p.m. - 20 p.m.). The proposed method produces the fewest errors for the afternoon peak hours.

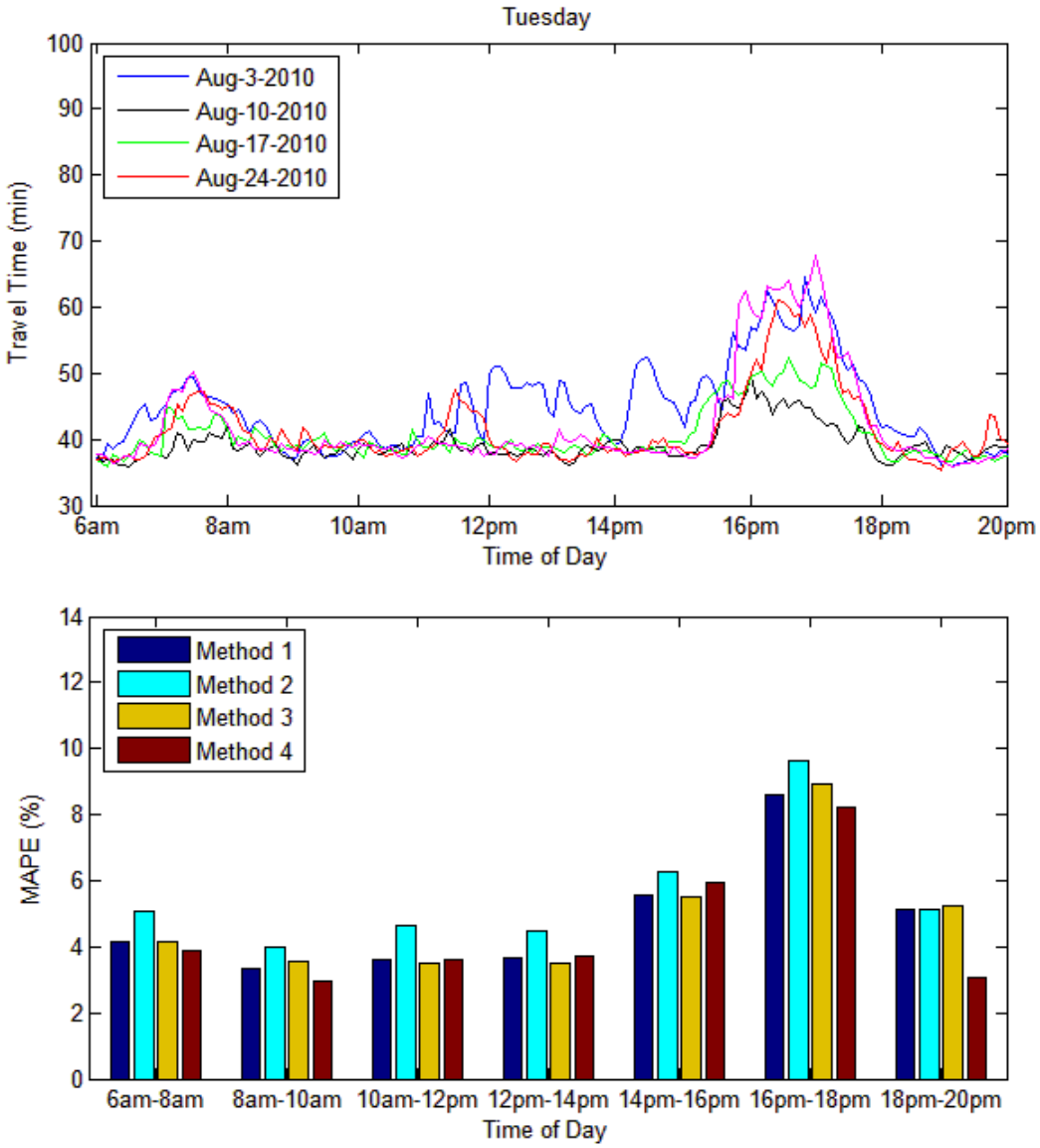


Figure 6.20: The ground truth travel time data and the comparison of four methods for every time period on Tuesday.

According to the travel time curves on Tuesday, there is mitigated congestion during morning peak hours (6 a.m. - 9 a.m.) and heavy congestion during afternoon peak hours (16 p.m. - 20 p.m.). The proposed method produces the least errors for both the morning and afternoon peak hours.

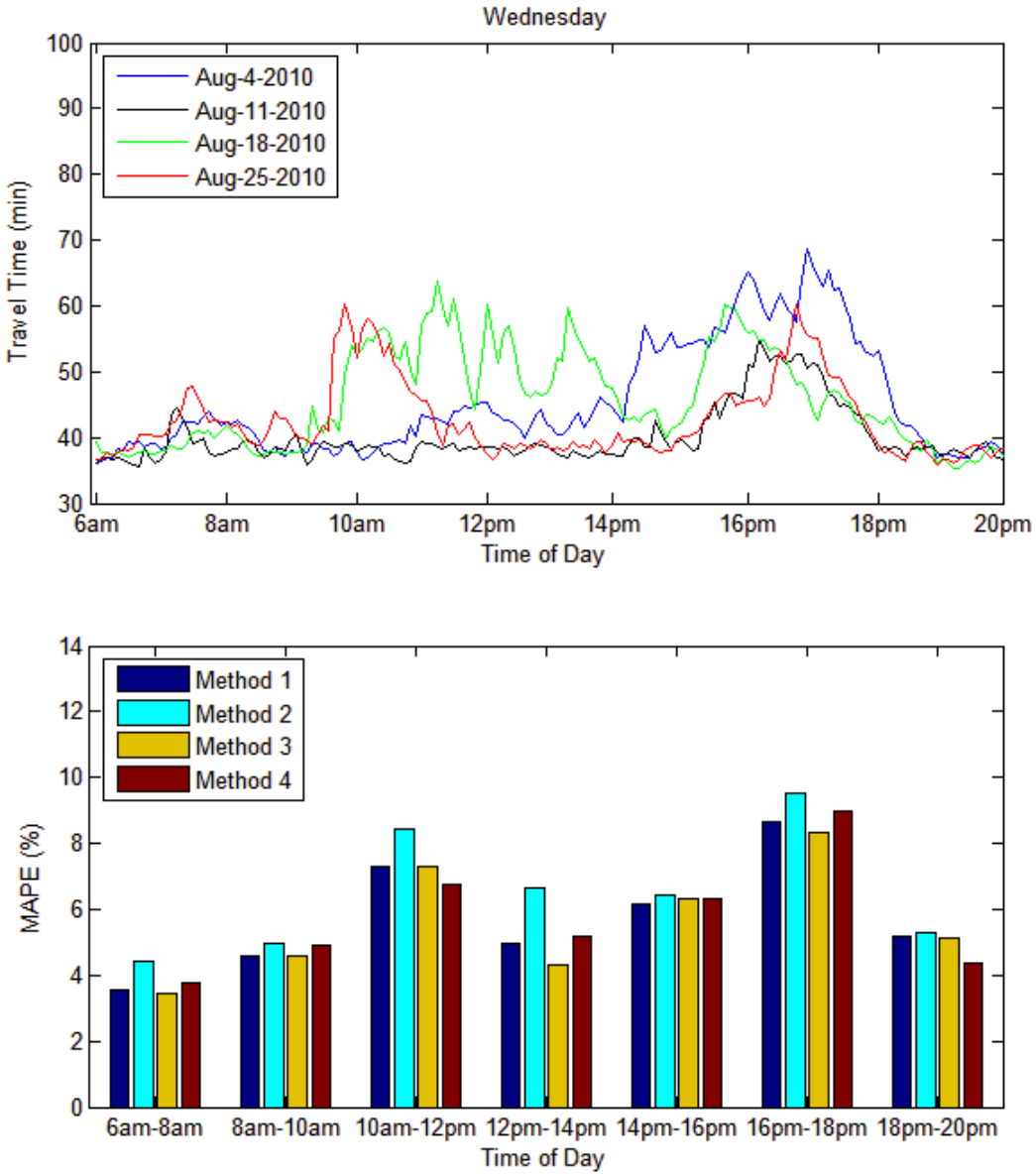


Figure 6.21: The ground truth travel time data and the comparison of four methods for every time period on Wednesday.

According to the travel time curves on Wednesday, there is mitigated congestion during morning peak hours (6 a.m. - 9 a.m.) and heavy congestion during afternoon peak hours (14 p.m. - 19 p.m.). The proposed method does not work well for both morning and afternoon peak hours, since there is abnormal congestion on August 18 (heavy congestion around 10a.m.) and August 25 (heavy congestion between 10 a.m. to 14 p.m.). Such abnormal congestion increases the difficulty of finding similar traffic patterns from the historical data set; therefore, the proposed method cannot produce the fewest prediction errors on Wednesday. This problem may be solved if such abnormal traffic data are included on our historical database.

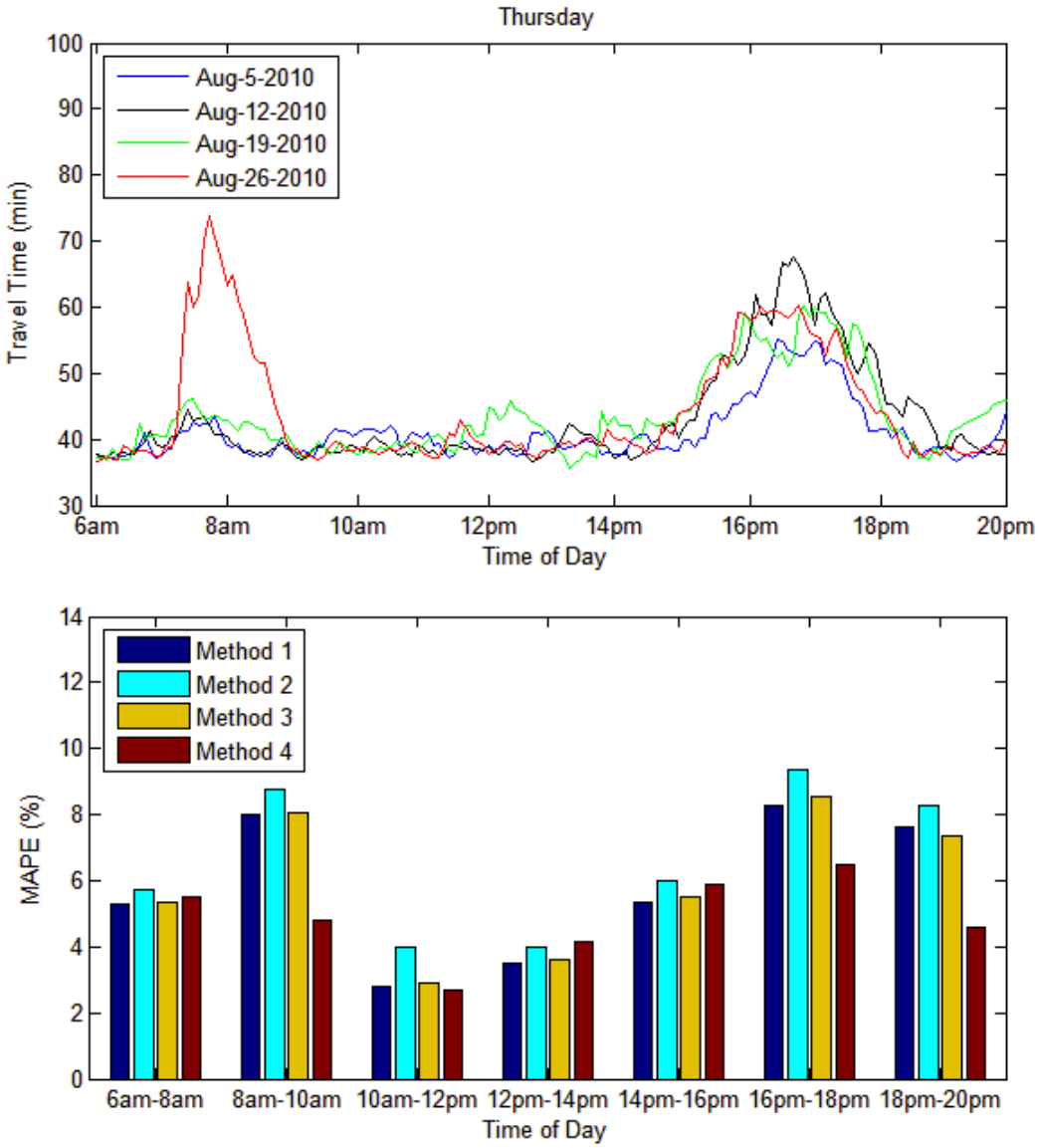


Figure 6.22: The ground truth travel time data and the comparison of four methods for every time period on Thursday.

According to the travel time curves on Thursday, there is mitigated congestion during morning peak hours (6 a.m. - 9 a.m.) and heavy congestion during afternoon peak hours (15 p.m. - 20 p.m.). The proposed method produces the fewest errors for both the morning and afternoon peak hours. The results from 6 a.m. - 8 a.m. and 14 p.m. - 16 p.m. demonstrate that the proposed method does not work very well when the congestion is forming; however, it works very well when congestion is dispatching, as compared to other methods.

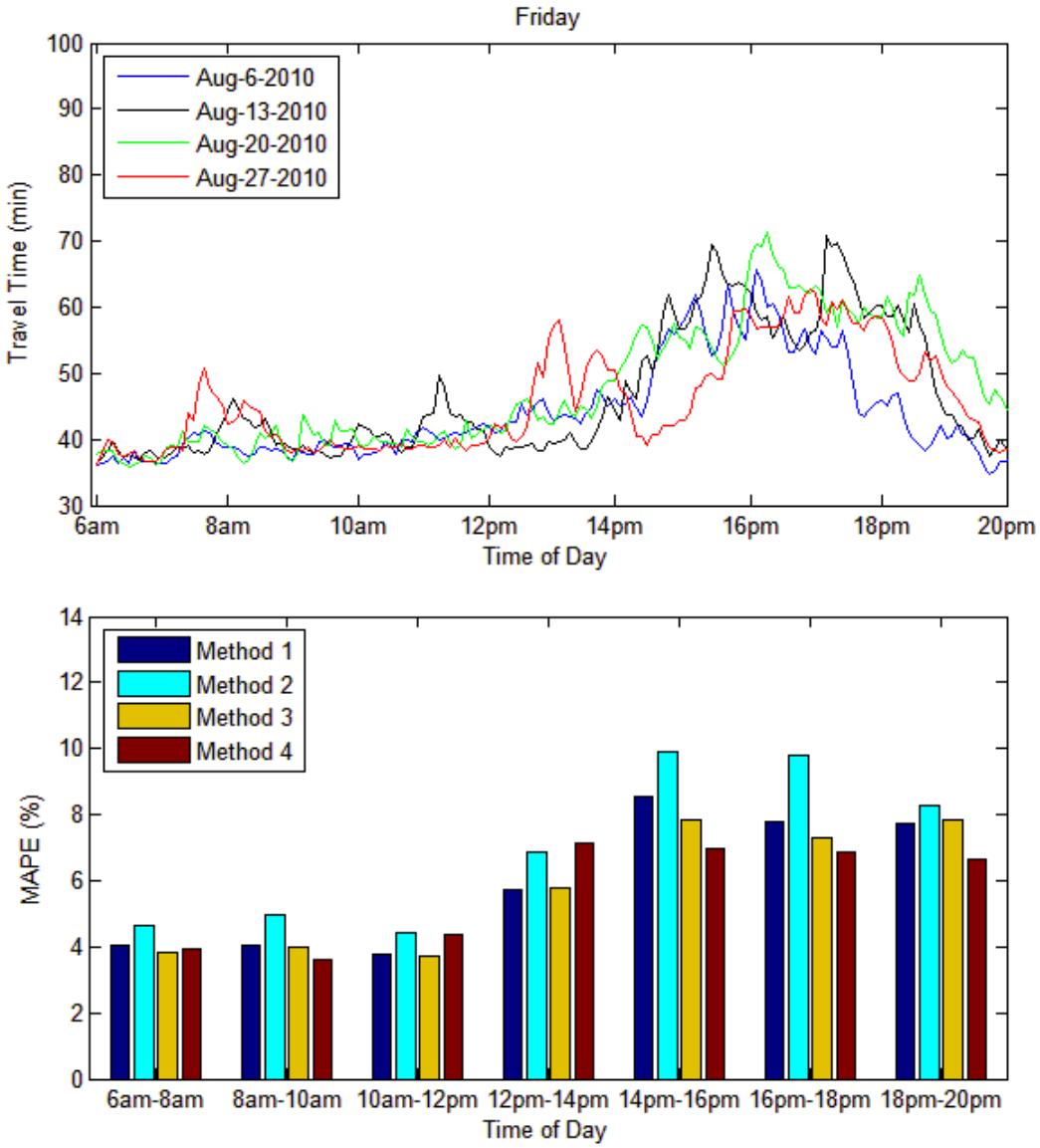


Figure 6.23: Method error relative to ground truth for Friday.

According to the travel time curves on Friday, there is mitigated congestion during morning peak hours (7 a.m. - 9 a.m.) and heavy congestion during afternoon peak hours (14 p.m. - 20 p.m.). The proposed method produces the fewest errors for both the latter half of the morning peak period and the afternoon peak hours.

According to the travel time curves on Saturday, the congestion occurs for a longer time from 11 a.m. to 20 p.m. than it does on weekdays. The proposed method works well, except the time period of 16 p.m. - 18 p.m.

According to the travel time curves on Sunday, congestion happens irregularly during the afternoon and evening. The proposed method does not work very well for such scenarios.

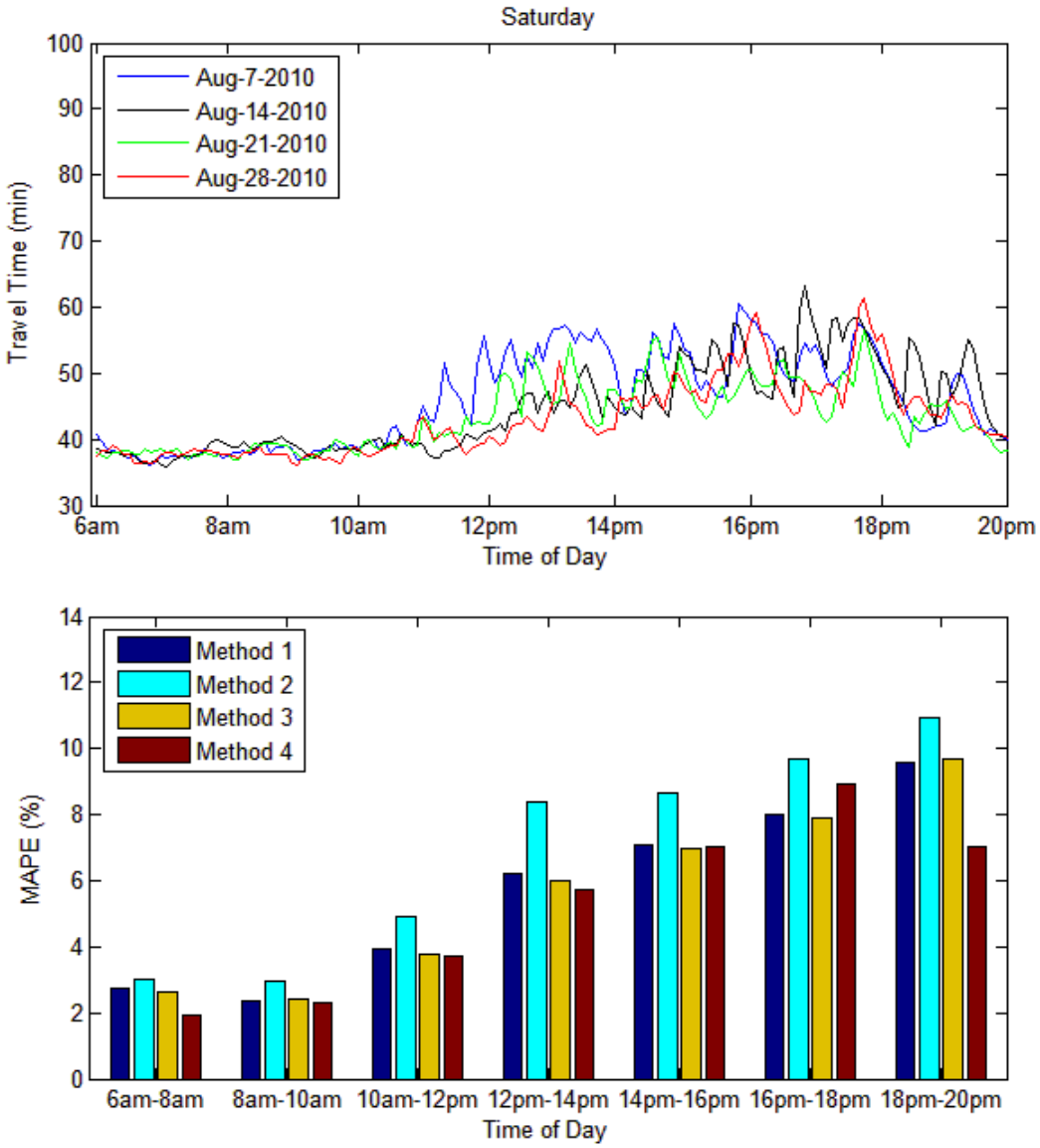


Figure 6.24: Method error relative to ground truth for Saturday.

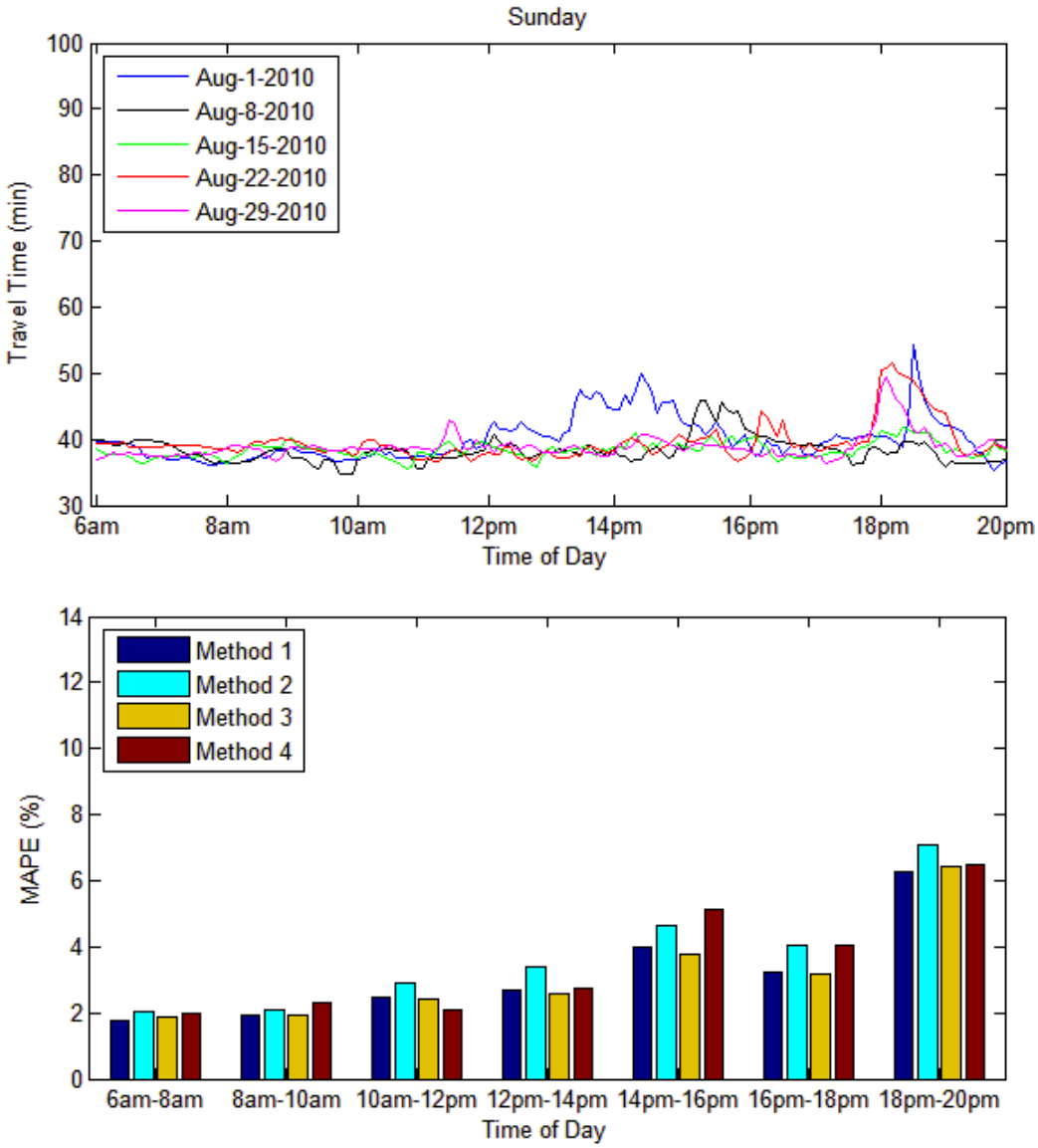


Figure 6.25: Method error relative to ground truth for Sunday.

6.3.4 Case study 3

The entire 95-mile freeway stretch from Richmond to Virginia Beach is used in this case study to test the revised prediction algorithm. The same testing data set of 31 days in August 2010 as in the previous case studies is also used. All the previous traffic data since April 1, 2010 are used as the historical data set. For instance, if the current testing day is August 21, 2010, the historical data set is constituted by the 142 days from April 1, 2010 to August 20, 2010.

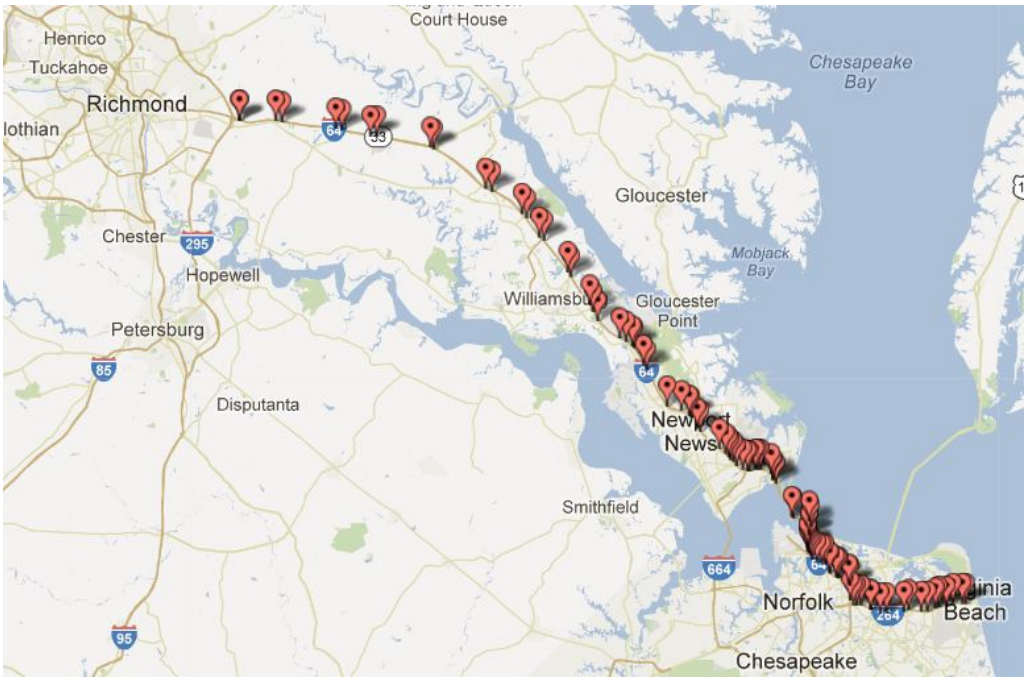


Figure 6.26: Freeway stretch from Richmond to Virginia Beach along I-64 and I-264.

In order to evaluate the performance of the proposed algorithm, the predicted travel time is compared with the instantaneous travel time as currently used by the Virginia Department of Transportation. For an actual trip, the experienced travel time is the actual travel time for a vehicle to travel from its origin to its destination. The roadway speed will not only change across space but also across time during the trip. Comparatively, the instantaneous travel time is the summation of real-time section travel times without the consideration of speed evolution across time. Many traffic agencies use instantaneous travel time to represent the future travel time information and provide this to drivers, which is valid if the current traffic status remains constant until the completion of the trip. However, instantaneous travel time may deviate substantially from the experienced travel time under transient states during which congestion is forming or dissipating.

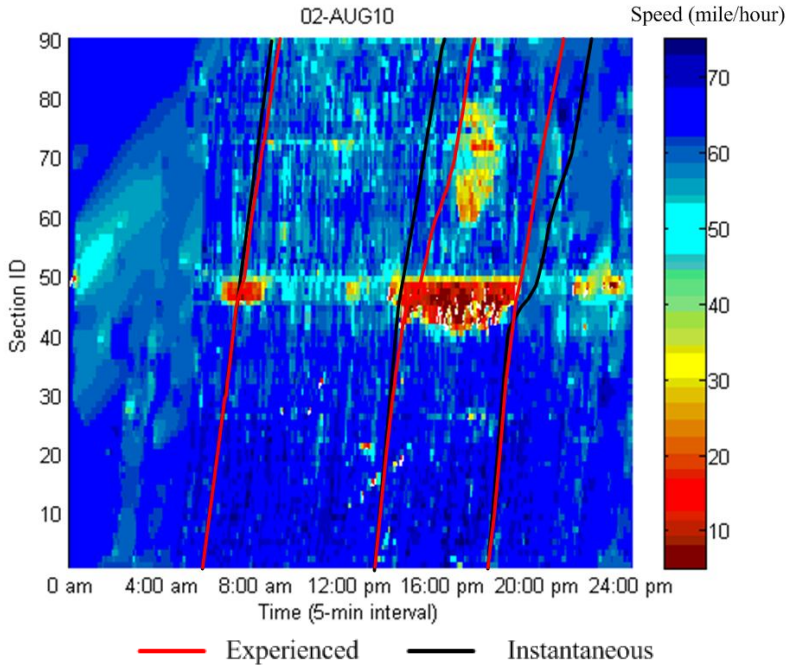


Figure 6.27: Spatiotemporal traffic state map and trip trajectories.

The temporal-spatial traffic state map is obtained from raw INRIX data on August 2, 2010 as presented in Figure 6.27. The differences between the instantaneous and experienced travel times are demonstrated by plotting the vehicle trajectories using both approaches. For a trip starting at 6:00 a.m., the instantaneous travel time is computed as 90 minutes (using the instantaneous speed values along all roadway segments at 6:00 a.m.) and the trip trajectory is depicted by the left black line. However, the corresponding experienced travel time (100 minutes) from the left red trajectory is 10 minutes longer than the instantaneous travel time, since the trajectory experiences the morning congestion (6:30 - 8:00 a.m.) before the tunnel along I-64. Large differences between instantaneous and experienced travel times occur during the afternoon peak hours because of heavy congestion. The instantaneous travel time starting at 14:00 p.m. is calculated as 92 minutes from the middle black trajectory, during which no congestion is included. But the experienced trajectory represented by the middle red curve is 117 minutes, since the trip encounters the tunnel congestion at around 14:30 p.m. and the low speed sections along I-264 at around 15:30 p.m. Therefore, the instantaneous travel time underestimates the experienced travel time by 25 minutes. The last demonstration is the opposite situation: the instantaneous travel time at 19:30 p.m. is 123 minutes, since the traffic is highly congested at two locations (tunnel and I-264). However, the experienced trajectory represented by the right red curve is 95 minutes and encounters almost no congestion. The instantaneous travel time overestimates the experienced travel time by 28 minutes in this case. To sum up, the above trajectories demonstrate that the instantaneous travel time calculated using the real-time data is not a good approach for predicting the experienced travel time, especially during congested periods.

The proposed approach is used to predict the temporal-spatial traffic states using historical data to construct vehicle trajectories and estimate travel times. The actual experienced travel time is calculated on the test data set as the ground truth. In order to effectively compare the instantaneous travel time results with the proposed approach, the congestion periods are extracted to calculate the prediction errors. The congestion status can be defined if the speed is less than 80% of the free flow speed. Therefore, the congested travel time can be identified if the travel time is higher than 1.25 times the free flow speed travel time. Generally, up to four congestion periods may be extracted during a day by aggregating the adjacent time intervals of congested status, which are morning, noon, afternoon, and evening congested periods. Considering the 31 days in August 2010, totally 57 congested periods are identified. The MAEs are calculated to compare the prediction accuracies between the predicted travel times and ground truth values during congested periods. By sorting the MAEs of instantaneous travel time and plotting the corresponding errors from two methods, the differences in prediction performance can be demonstrated in Figure 6.28. The average MAE by instantaneous travel time is 9.2 minutes and the average MAE from the proposed method is reduced to almost half error of 4.8 minutes. Among the 57 total congested periods, the proposed method produces many fewer errors (averaging less than 40%) than does the instantaneous travel time assumption for 49 periods. The samples of congested periods for August 4, 2010 and August 27, 2010 are highlighted in Figure 6.28. The proposed method outperforms instantaneous travel time during all congested periods for these two days.

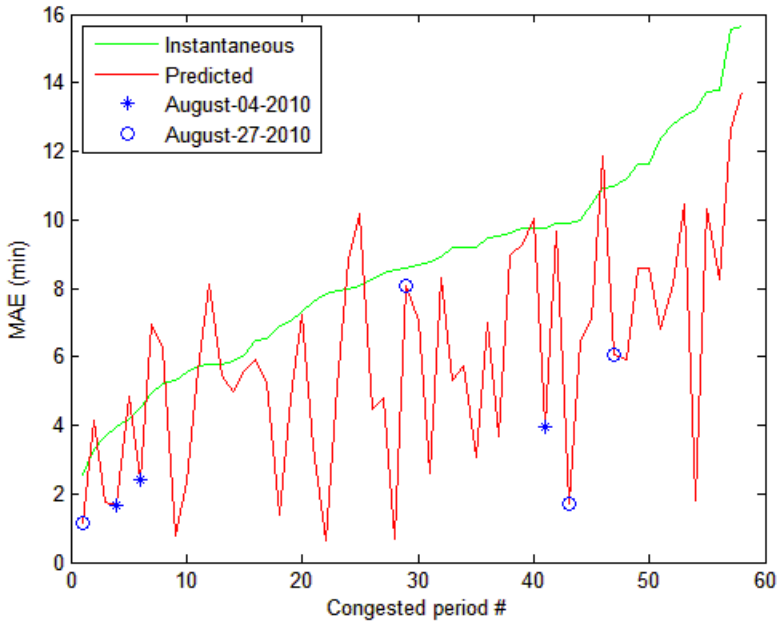


Figure 6.28: Average MAE of congested periods by two methods.

In order to investigate the maximum deviation between prediction results and ground truth data, the maximum MAE for using two methods for different testing days is selected and presented in Figure 6.29. The range of maximum MAE by instantaneous

method is 11.7 - 44.5 minutes. The range of maximum MAE by the proposed method is 8.5 - 26.2 minutes, which is much lower than for the instantaneous method.

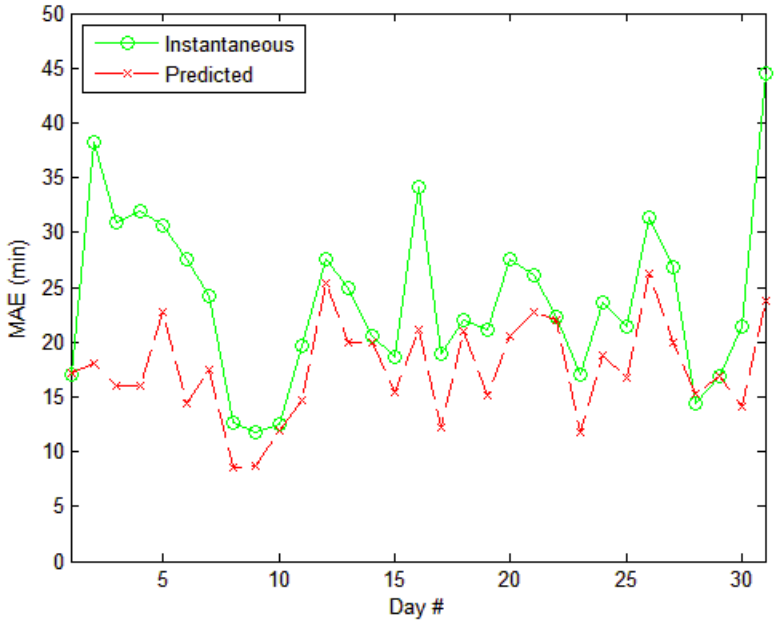


Figure 6.29: Maximum MAE by two methods for August 2010.

The travel time curves by the proposed prediction method, the instantaneous travel time, and the ground truth data for a typical weekday – August 4, 2010 –are presented in Figure 6.30 (a). The instantaneous travel time experiences some time lag to the ground truth data, especially during the time that congestion is forming or dissipating. Specifically, the instantaneous travel time highly underestimates the ground truth value when congestion is forming, and overestimates the ground truth travel time when congestion is dissipating. Comparatively, the proposed method improves the prediction performance when congestion is forming but still has some lag. The prediction performance during the congestion dissipating period is highly improved by the proposed method, since the propagation of shockwave can be predicted according to the historical trend from selected candidates. For instance, comparing the prediction errors by the proposed and instantaneous approach on August 4, 2010, the maximum reductions for congestion forming and dissipating periods are 12 minutes (from 13.4 to 1.4 minutes) at 14:30 p.m. and 25 minutes (from 25.8 to 0.8 minutes) at 17:40 p.m., respectively. The other benefit of the proposed approach is that travel time distribution can also be predicted other than as a deterministic value. The 95% and 5% confidence intervals of the predicted travel times are calculated as the upper and bottom boundaries as shown in Figure 6.30 (b). The green shadow area between boundaries covers most of the ground truth data curve, which demonstrates that the proposed approach provides very good accuracy to predict travel time reliability.

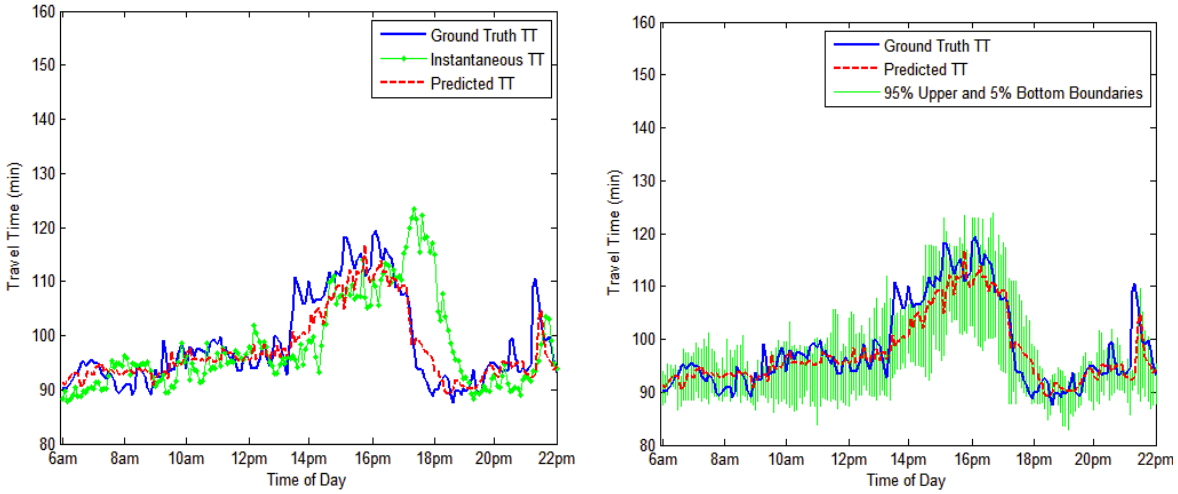


Figure 6.30: Travel time prediction results on August 04, 2010; (a) comparison between the proposed approach and instantaneous travel time; (b) the upper and bottom boundaries of proposed approach.

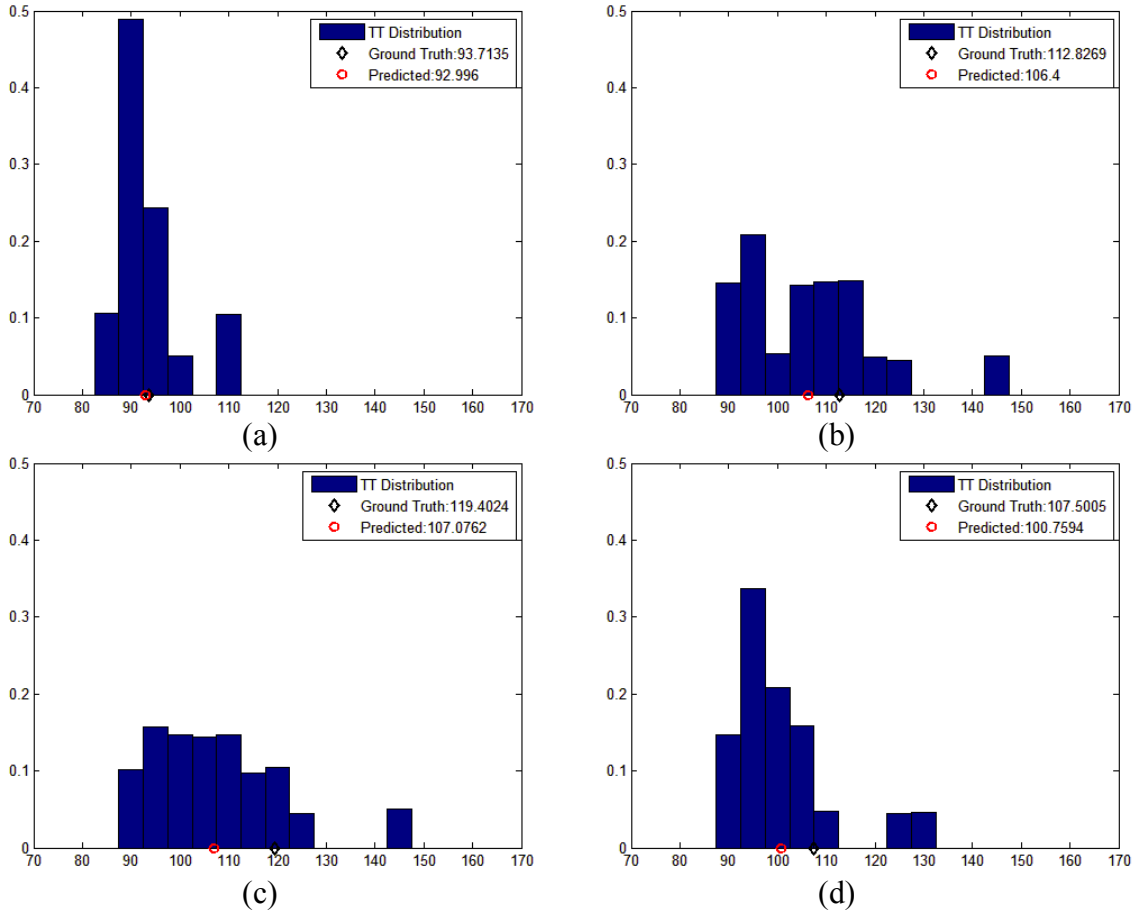
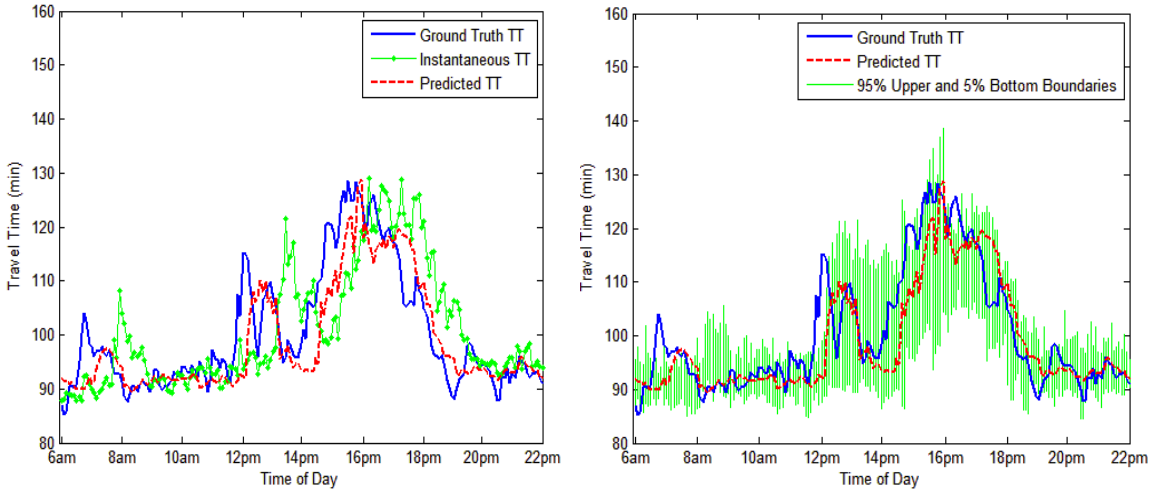


Figure 6.31: Predicted travel time distribution on August 04, 2010; (a) 9 a.m. (b) 15 p.m. (c) 16 p.m. (d) 17 p.m.

Besides calculating the upper and bottom boundaries of travel time reliability, the travel time distribution can also be predicted by the proposed algorithm. Each selected candidate traffic status is corresponding to a predicted travel time and an associated weight value, which has been described in Equation (6.5). The weight value represents the dissimilarity between current and historical traffic patterns and can be used to calculate the probability of the associated travel time prediction result. Consequently, the predicted travel time distribution can be calculated for each time interval, as presented in Figure 6.31. The mean of travel time distribution is denoted by red dots and compared with the ground truth travel time value to demonstrate the high prediction accuracy. Rather than computing the mean value, other statistical representations can also be calculated as the prediction output. For instance, computing the 80th percentile in Figure 6.31 can be another option and maybe has less error to ground truth data. More importantly, the previous work of the research team to model travel time distribution can also be considered to improve prediction accuracy in the future study.

Similarly, the proposed approach outperforms the instantaneous approach on August 27, 2010, as shown in Figure 6.32. The maximum reductions of prediction errors by the proposed approach are 12.2 minutes (from 15 to 2.8 minutes) at 12:55 p.m. and 17.4 minutes (from 17.7 to 0.3 minutes) at 17:55 p.m. during congestion forming and dissipating periods, respectively.



(a) (b)

Figure 6.32: Travel time prediction results on August 27, 2010; (a) comparison between the proposed approach and instantaneous travel times; (b) the upper and lower boundaries of proposed approach.

6.4 Conclusions

The research efforts in this chapter develop a travel time prediction algorithm based on computer vision and pattern recognition techniques by matching historical data with real-time traffic conditions. Besides, a data reduction procedure is developed to transfer the raw INRIX probe data into a uniform format which can be used in the proposed travel time prediction algorithm. The probe data from I-64 and I-264 between Richmond to Virginia Beach for the past three years are collected for testing proposed algorithms. INRIX data for the selected 37-mile freeway stretch (Newport News to Virginia Beach) are used to test the proposed algorithm in case study 1 and 2. The testing results indicate that the proposed algorithm outperforms the other three methods including using instantaneous measurements, using a Kalman filter, and using the k-nearest-neighbor method. Moreover, the case study 3 conducted on the entire 95-mile freeway stretch from Richmond to Virginia Beach demonstrates the superiority of the proposed algorithm over the instantaneous approach that is currently used by Virginia Department of Transportation. The proposed prediction method reduces the prediction error by approximately 50 percent compared to the current instantaneous method, especially at the shoulders of the peak periods.

The traffic state and travel time prediction in this dissertation are localized application based on individual route. The applications on network-level are recommended for future research. The predicted travel times on different corridors can be used to develop control strategies for route choice recommendations or area congestion reduction. A key input to these approaches is conducting research on how drivers respond to the provision of real-time information and how they switch their routes of travel depending on the information provided to them.

Chapter 7

7 Travel Time Prediction using Dynamic Template Matching

This chapter is an edited version of: Hao Chen, Hesham A. Rakha (2014), "Predicting Freeway Travel Times using Dynamic Template Matching," submitted to *IEEE Transaction on Intelligent Transportation Systems*.

This chapter develops a predictive travel time algorithm using dynamic template matching to identify similar spatiotemporal trends in a historical dataset for use in prediction purposes. Unlike previous approaches, which use fixed template sizes, the proposed method uses a dynamic template size that is updated each time interval based on the spatiotemporal shape of the congestion upstream of the bottleneck. In addition, the computational cost is reduced using a Fast Fourier Transform instead of Euclidean distance. Subsequently, the historical candidates that are similar to the current conditions are used to predict the experienced travel times. The proposed method is tested on a freeway stretch along I-64 using five-minute aggregated probe data provided by INRIX. The study demonstrates that the proposed method produces significantly better and more stable prediction results for prediction horizons up to 30 minutes into the future compared to instantaneous and fixed template matching methods. Furthermore, a comparison of the fixed-template and dynamic-template methods indicates that the dynamic template enhances the prediction accuracy at the shoulders of the congested periods. Finally, the proposed dynamic template matching approach has the flexibility of using an incremental historical dataset, which is demonstrated to further improve the prediction accuracy over the use of a fixed historical dataset.

7.1 Introduction

This chapter is an extended work from the developed framework of travel time prediction by pattern recognition on the previous chapter. In order to accurately match traffic patterns between real-time and historical data, one of the most frequently used pattern recognition techniques - template matching is revised and applied in this paper. Template matching is usually used in computer vision problems for finding small parts of an image (matrix) which match a template image matrix [88]. Comparing to other pattern recognition techniques, for instance image feature selection [83, 84] and artificial neural networks [9, 32, 89, 90], template matching is a simple but powerful algorithm and very suitable to deal with online pattern recognition problems since the offline training process is not needed. More importantly, the traffic flow fundamentals including congestion identification and bottleneck propagation are also used in the proposed approach to enhance the performance of template matching.

This study develops a dynamic template matching method to predict experienced travel times over multiple prediction horizons. Instead of using a fixed template size, as is done in other studies, the template size is dynamically updated each time interval based on the spatiotemporal shape of the congestion formed upstream of the bottleneck. In addition, a Fast Fourier Transform (FFT) is used to reduce the computation cost in the template matching process. The selected historical candidates that are similar to current conditions are used to predict the experienced travel times. A freeway stretch on I-64 is selected to test the proposed algorithm using five-minute aggregated traffic data provided by INRIX. The travel time prediction results demonstrate that the proposed method produces higher prediction accuracies compared to instantaneous travel times and fixed template matching methods.

The remainder of this chapter is organized as follows. A literature review of previous travel time prediction methods is provided. Subsequently, the proposed methodology of using current and historical traffic status to predict experienced travel times is presented. This is followed by a description of the test data for the case study and the comparison results of using proposed approach for prediction. The last section provides the summary conclusions of this study and some research recommendations for future research.

7.2 Methodology

The same three-stage processes as presented in Figure 6.13 are used in the proposed algorithm: identify current traffic status, match similar traffic patterns from historical and predict travel times. The current traffic status is initially selected to represent the traffic status of all freeway sections from short-past to the current time interval. The traffic status in this case is a matrix across temporal and spatial axes. Thereafter, the historical traffic speed data with the same dimension to current traffic status is selected as a candidate. Based on the similarity to the current speed matrix, several candidates are extracted to represent the historical recurrent traffic patterns that are similar to the current

status. Finally, the subsequence experienced travel times of those candidates are aggregated to represent the travel time distributions in the future.

Suppose T is the current day and C denotes the current time interval. Considering the application of matching traffic patterns in this paper, the current traffic status $x(C,T)$ in the first stage of the proposed framework is the template matrix for time period $[T-L+1, T-L+2, \dots, T]$, and the matrix $x(h_i)$ representing the spatial-temporal traffic status on each historical day h_i is the image to be matched with template. Here, L represents the width of template across time intervals. Apparently the width of historical matrix $x(h_i)$ is greater than template of real-time traffic status $x(C,T)$, given that the template only includes the traffic data in a short time period and the historical matrix covers the entire time periods on that day. If t is a selected time index in historical day h_i, a matrix $x(h_i, t)$ for time period $[t-L+1, t-L+2, \dots, t]$ can be matched with template. It should be noted that the traffic status for each time interval is a vector that covers the entire roadway segments (totally N segments), therefore the template is a matrix with dimension N by L. Given that the value of N is constant for a selected roadway stretch, the question is how to define the value of L in order to produce the best template matching result.

Different from the previous studies to match traffic pattern by a fixed window width, a dynamic template, which is updated in real-time according to the identified congestion and bottleneck shape, is proposed in this paper to optimize the template size. Moreover, instead of finding the minimum Euclidean distance, a fast Fourier transform (FFT) - based method is used in the second stage of the proposed framework in order to save the computation time of template matching. Eventually within the final stage, the selected historical candidates by template matching can be aggregated to provide the multiple-step prediction of travel times on the current day. The details of methodologies for the three stages of proposed framework are presented as below.

7.2.1 Updating dynamic template

In this paper, the dynamic template is updated in real-time according to the traffic flow fundamentals (e.g. congestion and bottleneck identifications). The identified congested roadway segments are used to track the change in traffic conditions, so that the template size is dynamically computed to reflect the activation and propagation of shockwaves. Thereafter, a more accurately template matching and travel time prediction can be achieved based on the propagation of shockwaves in the future time intervals. The bottleneck identification technique is part of our proposed algorithm to update dynamic template size. Considering the simple assumption and easy implementation, a mixture model congestion identification algorithm which is developed by Center for Sustainable Mobility at the Virginia Tech Transportation institute is used in this paper [91]. However, it should be noted that the proposed dynamic template matching method is not constrained to a specific technique, other similar approaches which deal with online congestion or bottleneck identification can also be considered as an alternative.

The mixture model congestion identification algorithm assumes the traffic speed across roadway road segments have an underlying fundamental diagram trend with randomness associated with the data. The variability of speed is substantial in congested traffic

condition. Due to this random nature of traffic speed, stochastic models are the best choice for modeling the distribution of speed. Stochastic models have been proven to be really good tools in travel time reliability modeling [92, 93]. Assume the traffic condition is either congested or uncongested, a mixture distributions where each component corresponds to a specific traffic condition is used to model multistate traffic condition. This algorithm does not require any pre-defined parameters making it easier to implement compared to the state-of-the-art Chen's algorithm [94]. By using the traffic speed measurement over spatiotemporal domain, this algorithm fits two lognormal distributions as demonstrated in Equation (7.1).

$$f\left(u \mid \lambda, \mu_1, \mu_2, \sigma_1, \sigma_2,\right) = \lambda \frac{1}{\sqrt{2\pi} u \sigma_1} e^{\frac{(\ln u - \mu_1)^2}{2\sigma_1^2}} + (1-\lambda) \frac{1}{\sqrt{2\pi} u \sigma_2} e^{\frac{(\ln u - \mu_2)^2}{2\sigma_2^2}} . \qquad (7.1)$$

where (μ_1,σ_1) and (μ_2,σ_2) are the mean and standard deviation of the first and second component distributions and λ is the mixture parameter. Thereafter, the threshold to identify the traffic condition can be calculated by locating the 0.001 quantile of the fitted uncongested distribution. Lastly, in order to filter noise of in the results, a morphological operation is used to remove the spatiotemporal local uncongested regions identified in the congested area. The current criterion is set at four cells.

The congestion identification algorithm produces a spatiotemporal binary matrix of traffic state, in which the value of zero and one represent the uncongested and congested conditions, respectively. For real-time application, the online congestion identification is conducted for every time interval. A default value of template size is assumed if the real-time traffic status is uncongested. Here, the default value is selected based on the optimum window width for fixed template matching approach. The details of how to find the default template size will be demonstrated on the case study later on. Once the current traffic is identified within a bottleneck, then the corresponding template width is computed as the time difference between bottleneck activation and current time interval. In this way, the dynamic traffic information of bottleneck activation and propagation are included in the template window. It should be pointed out that multiple bottlenecks of different locations are considered as correlated to each other. So the congested traffic conditions along the entire roadway stretch are integrated together to come up the strategy of finding dynamic template. Specifically, a projection map is conducted by the summation of binary matrix along vertical direction. Therefore, the activation of each bottleneck can be identified from the projection map, and then the dynamic template can be updated by the time intervals from current congested time back to the bottleneck activation point. In this way, the dynamic template keeps updating in response to the real-time traffic condition and bottleneck propagation.

An example of calculating the dynamic template width is demonstrated in Figure 7.1by using the probe data along I-64 between Richmond and Hampton Roads on June 19, 2012 from 5:00 AM to 22:00 PM. The contour of speed measurement over spatial and temporal is presented in Figure 7.1 (a) by different scale of colors from blue to red. The binary matrix is obtained as shown in Figure 7.1 (b) using the described mixture distribution algorithm. Thereafter, the horizontal projection of the binary matrix is

computed in Figure 7.1 (c). The dynamic template is updated using the horizontal projection map. For instance, assume the current time is 8:00 AM, then the corresponding template width is calculated as 17 time intervals since the current bottleneck begins on 6:35 AM. For the scenario on 18:00 PM, the same procedure can be used to calculate the template width as 43 time intervals since the associated bottleneck starts at 14:25 PM. Therefore, the width of dynamic template is calculated for each time interval in Figure 7.1 (d).

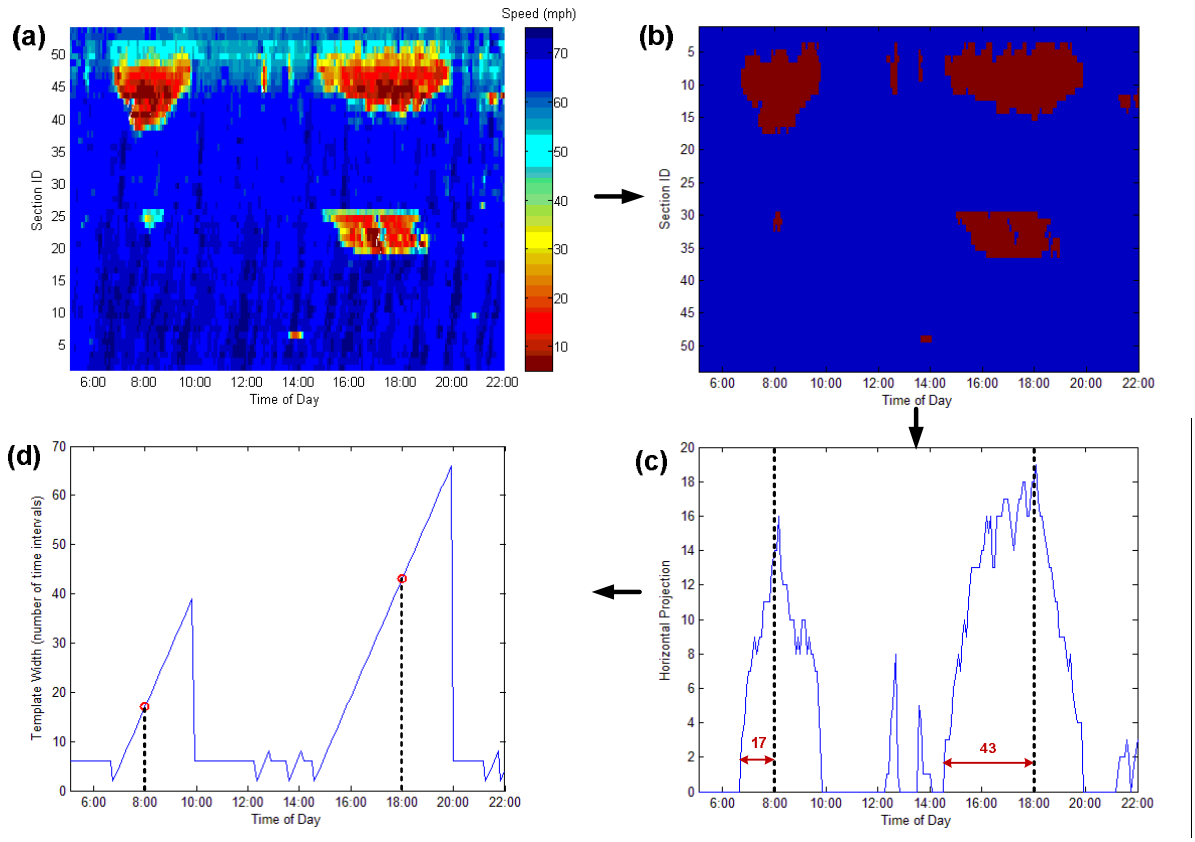


Figure 7.1: Calculate dynamic template width by congestion identification; (a) speed contour; (b) congestion identification result; (c) horizontal projection of congestion map; (d) dynamic template width.

7.2.2 Matching traffic patterns

In this section, template matching technique is conducted to match the template representing real-time traffic pattern with historical dataset in order to select the most similar candidates. Euclidean distance is the most widely used criterion to calculate the similarity of template matching result [88]. It should be noted that matching current and historical traffic data by Euclidean distance has also been frequently used to predict travel time or traffic flow in recent years [11, 95, 96]. In these studies, the average Euclidean

distance between the current traffic data and the data with same dimension from historical days is calculated to represent a similarity measure by Equation (7.2).

$$d\left(C,h_i\right)=\sum\left|x\left(C,T\right)-x\left(h_i,t\right)\right|. \tag{7.2}$$

where *M(c,L)* and *M(h,L)* represent the traffic data of the current and historical time intervals, respectively; *d(c,h)* is the summation of absolute error between the template and matching matrix in each cell.

However, the main disadvantage of the above method is the high computational cost [88]. It should be noted that the proposed template matching algorithm will be used on a large historical dataset, since a large set of different traffic patterns is helpful to produce higher prediction accuracy. In order to fill the requirement of real-time computation, a fast template matching approach is needed to avoid the delay in the system.

The idea of using the convolution theorem was proposed to solve this problem and proved to be an efficiently computation alternative and very easy to implement [88, 97]. In this paper, the FFT-based convolution method is used for fast matching template of traffic pattern. In this approach, the previous objective of finding the least Euclidean distance is replaced by searching the maximum cross correlation represented by Equation (7.3). Mathematically, the convolution theorem states that the Fourier transform of a convolution is the pointwise product of Fourier transforms. Therefore, the convolution between the current traffic pattern and target matrix of the historical day can be calculated as Equation (7.4).

$$\mathrm{conv}\left(C,h_i\right)=\sum x\left(C,T\right)\cdot x\left(h_i,t\right). \tag{7.3}$$

$$\mathcal{F}^{-1}\left\{\mathcal{F}\left\{x\left(C,T\right)\right\}\cdot\mathcal{F}\left\{x\left(h_i\right)\right\}\right\}. \tag{7.4}$$

where F is the Fourier transform, and F^{-1} is the reverse function of Fourier transform; $x(h_i)$ represents the entire measurements of traffic status over spatial and temporal for i^{th} historical day. In the previous studies of using Euclidean distance to match template with one historical day, the template matching process needs multiple iterations by shifting the template window along the time period of the entire day. However, only one computation by Equation (7.4) is enough to complete the template matching in the FFT-based convolution method. The output of Equation (7.4) is a set of similarities represented the matching result between template and the data matrix in historical day h_i. Therefore, the best matching can be located on time interval t_i, which corresponds to the maximum similarity denoted by $d_{max}(C,h_i)$.

The calculation of Equation (7.4) is iteratively conducted between the real-time template and each historical day. Therefore, several candidates are selected according to the descending order of the similarity measure. Suppose the maximum number of candidates is denoted by K, the set of candidates H_c is selected as

$$
\begin{aligned}
H_c \quad &= \quad \left\{\hat{h}_1, \hat{h}_2, \cdots, \hat{h}_K\right\} \\
\textit{where} \quad & h_i : \left[t_i, d_{\max}\left(C, h_i\right)\right] = \max \mathcal{F}^{-1}\left\{\mathcal{F}\left\{x\left(C,T\right)\right\} \cdot \mathcal{F}\left\{x\left(h_i\right)\right\}\right\} \\
& \hat{h}_1 = \arg\max d_{\max}\left(C, h_i\right) \\
& d_{\max}(C, \hat{h}_i) \le d_{\max}(C, \hat{h}_{i+1})
\end{aligned}
\tag{7.5}
$$

where $\hat{h}_i$ is the i^{th} selected candidate from historical dataset. The selected candidates represent the best matching to the current traffic status and will be used to calculate future travel times.

7.2.3 Travel time prediction

The future experienced travel times on the current day can be calculated based on the selected historical candidates. Considering the stochastic nature of a traffic system, the travel time prediction problem can be recognized as a time series prediction for nonlinear dynamic (chaotic) systems [86, 87]. The future traffic state for the current day can be predicted by the linear combination of subsequent traffic state of each candidate from the historical dataset, and the corresponding weight is defined as the normalized similarity measure. The predicted traffic state starting from time interval $c+p$ is obtained as

$$M(C, T+p) = \sum_{i=1}^{K} w(\hat{h}_i) \cdot M(\hat{h}_i, t_i + p). \tag{7.6}$$

$$w(\hat{h}_i) = \frac{d_{\max}(C, \hat{h}_i)}{\sum_{i=1}^{K} d_{\max}(C, \hat{h}_i)}. \tag{7.7}$$

where $M(h_i+p)$ represents the subsequent traffic state for i^{th} candidate starting from departure time t_i+p till the end of trip; and $w(h_i)$ denotes the weight of i^{th} candidate data.

Other than the predicted traffic state, the experienced travel time can also be predicted based on the subsequent experienced travel time of each candidate. Experienced travel time is the actual, realized travel time that a vehicle could experience during a trip. If a vehicle leaves a trip origin at the current time, the roadway speed will not only change across space but also across time during the entire trip. Therefore, the traffic state evolution over space and time is considered in our approach to calculate experienced travel times. The speed values of shaded cells are used to compute experienced travel times. In this paper, the traffic state is assumed to be homogenous within each cell. Therefore the trajectory slope, which represents the traffic speed, is a constant value in each cell. Assume the trip starts from time interval t_n. In this way, once the vehicle enters a new cell, the trajectory within this cell can be drawn with the slope value equal to the traffic stream speed. Finally, the experienced travel time can be calculated when the trajectory reaches the downstream boundary of the last roadway section (destination). In

this way, the subsequent experienced travel time of each candidate can be obtained and the corresponding weight (recurrent probability) is defined by the similarity measure of Equation (7.7). Therefore, the travel time distribution of the future trip departures on $T+p$ can be represented as

$$tt^{\exp}(C,T+p)=\left\{tt^{\exp}(\hat{h}_i,t_i+p),w(\hat{h}_i)\,|\,i=1,\cdots,K\right\}. \tag{7.8}$$

where $TT(c+p)$ represents the experienced travel time starting from time interval $c+p$; and $TT(h_i+p)$ denotes the subsequent travel time of i^{th} candidate according to the calculation illustrated in Figure 6.14. The travel time prediction result can also be calculated as the average value using Equation (7.9).

$$\overline{tt}^{\exp}(C,T+p)=\sum_{i=1}^{K}w(\hat{h}_i)\cdot tt^{\exp}(\hat{h}_i,t_i+p). \tag{7.9}$$

7.3 Case study

7.3.1 Test environment

The case study is conducted based on privately developed INRIX traffic data, which is mainly collected by GPS equipped vehicles. The collected probe data is supplemented by traditional road sensors, as well as mobile devices and other sources [98]. Since heavy traffic volumes are usually observed along I-64 heading to Virginia Beach during summer seasons and weekends, efficient and accurate travel time prediction can be helpful to travelers in planning their trips and reducing traffic congestion around the area. The INRIX data on the main segments along I-64 are used to construct the travel database in our study, which covers the major congested areas on I-64 from Richmond to Virginia Beach. The layout of the selected freeway stretch is presented in Figure 7.2, which includes 54 segments with the total length 67 miles. The average freeway segment length is 1.2 miles and the length of individual segment is unevenly distributed ranging from 0.1 to 6.4 miles. The average speed or travel time for each roadway segment are provided in the raw data, which are collected by every one-minute interval. In order to reduce the stochastic noises and measurement errors in the raw data, the raw speed information of each segment is aggregated by five-minute. Therefore, the traffic speed over spatial (upstream to downstream) and temporal (from 0:00 AM to 23:55 PM) can be obtained for each day, which is a data matrix with dimension 54 by 288. It should be noted that the full coverage of historical traffic speed data is required in this study. However, the problem of missing data is very common in the field measurements and thus must be addressed. Many traffic state estimation methods were proposed in order to obtain full coverage traffic state data by solving the mentioned problem [18, 19]. In the following sections, the traffic status is the full coverage traffic data after the process of data estimation. A detailed description of state estimation methods is beyond the scope of this paper and thus is not discussed further in this paper.

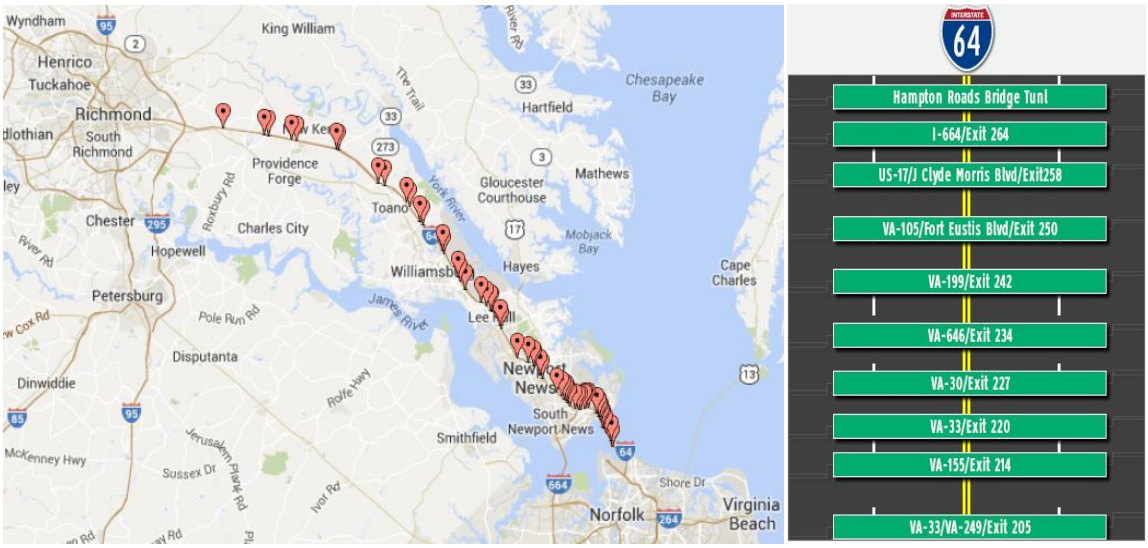


Figure 7.2: Layout of the selected freeway stretch.

Given the daily speed matrix, the instantaneous and experienced travel times can be calculated afterward. The instantaneous travel time is the summation of travel times for each roadway segment on the same time interval, in which the speed is constant over time. However, the experienced travel time is computed by considering the speed evolution across time. In this case, the speed profiles are piecewise constant values and the trip trajectory is a combination of diagonal curves over time and space [99]. In the following tests, time period between 5:00 AM to 22:00 PM is considered as the test period for each day since most of the congestions are covered in this period. Moreover, the ground truth travel time is represented by the experienced travel time, and the predicted travel times by different predictors are compared with the ground truth data to evaluate the prediction performance. Considering that the selected freeway stretch has heavy congestion during summer holiday season due to the high volume of traffic heading towards Virginia Beach, the traffic data from April to July, 2012 are used as the historical dataset and the subsequent traffic data in August and September, 2012 are employed as the test dataset to evaluate predictors.

Four prediction methods are tested the on the selected freeway stretch. Firstly, the real-time speed on each freeway segment is assumed to be constant on the future trip in the naïve idea to predict travel time. Therefore, the instantaneous travel time represents the easiest predictor to compare with other complicated methods. In order to explore the benefit of using dynamic template, a template matching by fixed window size is also included in this study and denoted as Method 1, in which a fixed template width is used as opposing to the dynamic template width in the proposed method. For the purpose of simplicity, the proposed dynamic template matching algorithm is denoted as Method 2. It should be pointed out that the pool of historical data for Method 1 and 2 only include the traffic data from April to July, 2012, which means the same historical dataset is used for different test days. However, the historical dataset keep updating to include all the previous days before the test day in Method 3, in which the dynamic template matching is used by an incremental historical dataset. For instance, the traffic data from April 1st to

August 31st, 2012 are used as the historical dataset if the test day is September 1st, 2012. By comparing Method 2 and 3, we can explore the benefit of using incremental historical dataset for field implementation. It should be noted that the proposed dynamic template matching method has the flexibility to use the updated historical dataset, comparing to other methods only use constant historical data pool, e.g. artificial neural networks [9, 32, 89, 90] and SVR [10, 33]. Considering the real world application, this is very important given that the characteristics of driver or traffic flow may change over time. For instance, an additional truck lane is added on the existing freeway and then the historical traffic data before this change may not provide a good pool of past traffic experiences to predict future traffic patterns.

Both relative and absolute prediction errors are used to evaluate the performance of predictors. The absolute error is denoted by the mean absolute error (MAE) using Equation (7.10), which represents the average absolute deviations between the predicted and the ground truth values. The corresponding relative error is represented by the mean absolute percentage error (MAPE) of Equation (7.11), which denotes the absolute proportional deviations between the predicted and the ground truth values.

$$MAE = \frac{1}{I \times J} \sum_{j=1}^{J} \sum_{i=1}^{I} \left| y_i^j - \hat{y}_i^j \right| . \tag{7.10}$$

$$MAPE = \frac{100}{I \times J} \sum_{j=1}^{J} \sum_{i=1}^{I} \frac{\left| y_i^j - \hat{y}_i^j \right|}{y_i^j} . \tag{7.11}$$

where J is the total number of days; I is the total number of time intervals in one day (i.e., 204 intervals occurring every five minutes between 5:00 am and 22:00 pm); and y_i^j and $\hat{y}_i^j$ denote the ground truth and the predicted value, respectively, of the experienced travel time for the i^{th} time interval on the j^{th} day in the test dataset.

7.3.2 Calculate the threshold for congestion identification

In order to use the proposed dynamic template matching method, the threshold in the mixture model approach to identify congestion needs to be calculated firstly for the historical dataset. The daily speed matrices from April to July, 2012 are used to generate the histogram of speed values. Here, the histogram of speed value is normalized so that the fitted distribution can be plotted on top of it. Fitting the histogram by the mixture log-normal distribution as Equation (7.1), the parameters λ, μ_1, σ_1, μ_2, σ_2 are estimated as 24.2, 11.0, 65.1, 4.9. The two distribution curves are shown in Figure 7.3 by red color to represent uncongested traffic and green color to represent congested traffic. Therefore, the threshold is selected by the speed value corresponding to 0.001 percentile of the uncongested traffic distribution, which is computed as 47.8 mph according to the cumulative distribution function (cdf) of its fitted distribution.

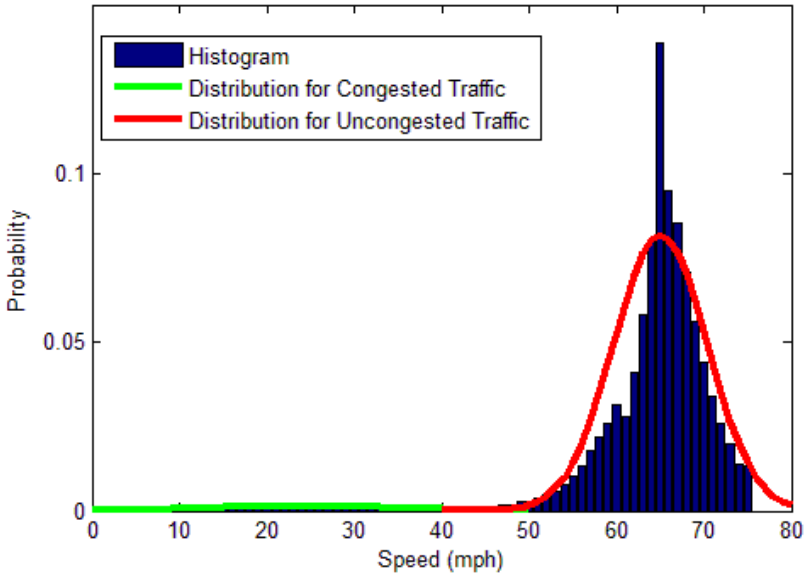


Figure 7.3: Calculate the threshold for congestion identification.

7.3.3 Find the optimum window width for fixed template method

As aforementioned, a default value of template width is assumed in the dynamic template matching approach if the real-time traffic status is uncongested. The default template width is selected in this study by finding the optimum window width for fixed template method. Given that a fixed window width L is used to match the real-time traffic pattern with historical data and eventually the selected K number of candidates are used to predict travel times, the template matching with fixed window size is essentially a k nearest neighbor (k-NN) method in the previous studies [95, 96, 100, 101]. These studies demonstrate the performance of k-NN approach depends on the parameters of candidate number K and window width L. Therefore, the variation of prediction accuracy by using the fixed template method with different K and L values is investigated, which is used as the criterion to select parameters for both fixed and dynamic template matching methods.

The test result on the selected I-64 dataset indicates the range of K between 4 and 14 produces a prediction result with very little variation for the fixed-window template matching method. This phenomenon that number of candidate within a certain range has very little impact on the matching result has also been observed in the similar work in [95]. Hence, the value of K is assumed to be constantly as 10 in the following tests for both fixed template matching and the proposed dynamic template matching methods.

The impacts of using template width between 5 to 50 minutes for various prediction horizons are presented in Figure 7.4. Generally the minimum MAPE can be obtained when template width of 30 minutes is used for various prediction horizon between 0 to 30 minutes, the only exception is that the MAPE associated with template width of 35 minutes is a slightly less than the template width of 30 minutes for prediction

horizon of 20 minutes. Hence, the template width of 30 minutes is selected as the optimum window width for fixed template method, and it's also used as the default template width in the dynamic template matching approach when the real-time traffic condition is uncongested.

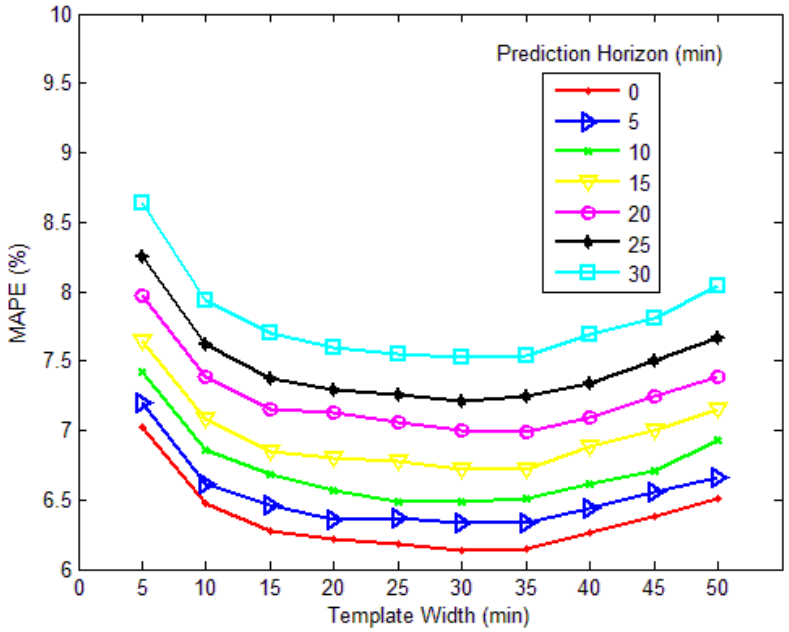


Figure 7.4: Impacts of template width on prediction accuracy

7.3.4 Test results

Table 7.1 presents the MAE and MAPE values by using four methods for prediction horizon between 0 to 30 minutes. The instantaneous travel time produces the worst prediction results especially for long prediction horizons, in which the MAPEs increase from 10.81% to 16.19% for prediction horizon of 0 to 30 minutes. Compared to instantaneous travel time, the prediction accuracy by using Method II is greatly improved, and the range of relative errors is between 6.13% to 7.52% for departure time up to 30 minutes in the future. Moreover, the dynamic template in Method II further improves the prediction performance of template matching method as opposing to a fixed template size in Method I. The MAPEs is ranging between 5.56% to 6.63% when prediction horizon increases from 0 to 30 minutes. Instead of the constant historical dataset in Method II, the incremental historical dataset in Method III is helpful to the proposed dynamic template matching method to produce less prediction errors, in which the MAPEs increase from 5.21% to 6.28% for prediction horizon of 0 to 30 minutes. This improvement is reasonable considering that the incremental historical dataset has more traffic patterns especially from the recent past days, which is beneficial to produce more accurate template matching result.

Table 7.1: Prediction results by four methods for prediction horizon between 0 to 30 min.

		Prediction Horizon (min)						
		0	5	10	15	20	25	30
Instantaneous	MAE (min)	8.33	8.89	9.46	9.83	10.37	10.81	11.43
	MAPE (%)	10.81	11.64	12.42	13.15	13.63	14.74	16.19
Method 1	MAE (min)	4.62	4.74	4.91	5.15	5.29	5.45	5.68
	MAPE (%)	6.13	6.31	6.48	6.72	6.97	7.21	7.52
Method 2	MAE (min)	4.23	4.34	4.45	4.53	4.67	4.85	5.03
	MAPE (%)	5.56	5.64	5.81	6.02	6.19	6.37	6.63
Method 3	MAE (min)	3.92	3.98	4.06	4.24	4.38	4.48	4.71
	MAPE (%)	5.21	5.29	5.42	5.61	5.73	5.96	6.28

Method 1: fixed template matching

Method 2: dynamic template matching

Method 3: dynamic template matching with incremental historical dataset

Rather than evaluating the average performance of each predictor, the prediction accuracies for different traffic conditions are also investigated in this study. Given that it's very easy to make predictions for uncongested traffic condition, the prediction accuracy under congested time periods is more valuable since travelers would need the predicted travel time to assist them under congested traffic. Therefore, the congested periods during August 2012 are selected to test the performance by using the fixed and dynamic template matching methods as opposing to instantaneous travel time.

The congested traffic condition can be defined as the speed under 80% of the free flow speed, which is usually used a typical value for the speed at capacity on freeways. Therefore, the congested period at least 30 minutes long is identified in this study when the corresponding travel times are higher than 1.25 times of free flow travel time. Generally, up to four congestion periods can be extracted during a day, which are morning, noon, afternoon and evening congested periods. Considering 31 days in August 2012, a total of 50 congested periods can be identified. The relative error of MAPE is calculated to assess the prediction accuracies between the predicted travel times and ground truth values during congested periods. It should be noted the index of congestion period is obtained by sorting the corresponding MAPE produced by the instantaneous travel time method in an ascending order. Figure 7.5 demonstrates the MAPE values using instantaneous travel time, Method 1 and Method 2 on each congested period for prediction horizon of 15 minutes. Out of the total 50 congestion periods, the template matching methods (Method 1 and 2) produce fewer errors than instantaneous method for 43 periods. For the left side of congestion periods (index from 1 to 22), the dynamic template in Method 2 doesn't provide an apparently improvement of prediction performance compared to the fixed template in Method 1. However, Method 2 produces fewer errors than Method 1 for the right side of congestion periods (index from 23 to 50). Given that the right side of congestion periods corresponds to the days with serious traffic

congestion, dynamic template works better than fixed template for highly congested traffic conditions.

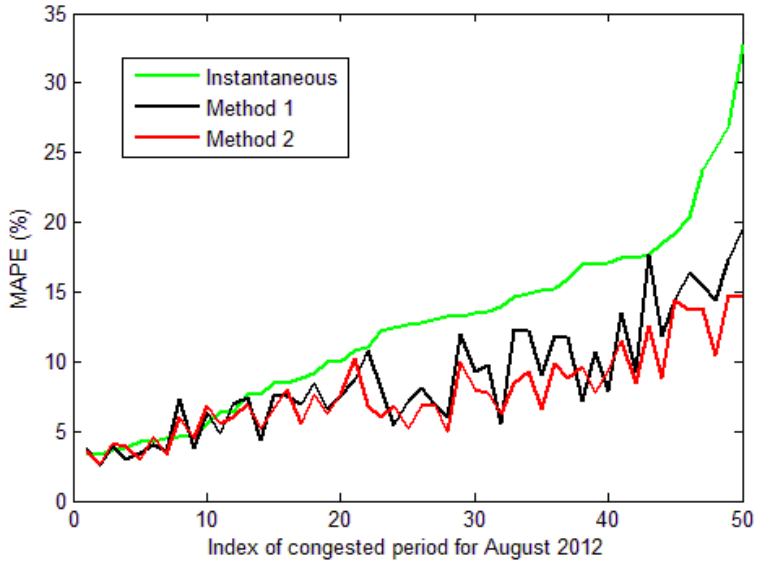


Figure 7.5: MAPE of congested periods by three methods for 15 min prediction horizon.

Figure 7.6 demonstrates the different results by using fixed and dynamic template matching methods on a sample day. The contour of speed matrix over space and time on August 18, 2012 is used as the test data. Assume the current time is 16:00 pm, the ground truth travel time when departures on 16:15 pm (prediction horizon of 15 minutes) can be calculated as 123 minutes by drawing the green trip trajectory on the speed contour. By using fixed window matching method, a template width ranging from 15:30 pm to 16:00 pm is extracted to match with historical dataset and the best matching result is located between 18:00 pm to 18:30 pm on July 19, 2012. Hence, the predicted travel time by fixed template matching method is 67 minutes, which underestimates the ground truth value by 56 minutes (MAPE of 45.5%). Alternatively, a much wider template width ranging between 11:30 am and 16:00 pm is used in the dynamic template matching method in order to cover the activation time of bottleneck. A similar traffic pattern with severe bottleneck to the dynamic template is selected between 12:00 pm to 18:30 pm on May 25, 2012, and the predicted travel time of 125 minutes only overestimates the ground truth value by 2 minutes (MAPE of 1.6%). This example clearly demonstrates that the dynamic template helps to find historical candidates with similar traffic pattern to the test day during congested traffic condition.

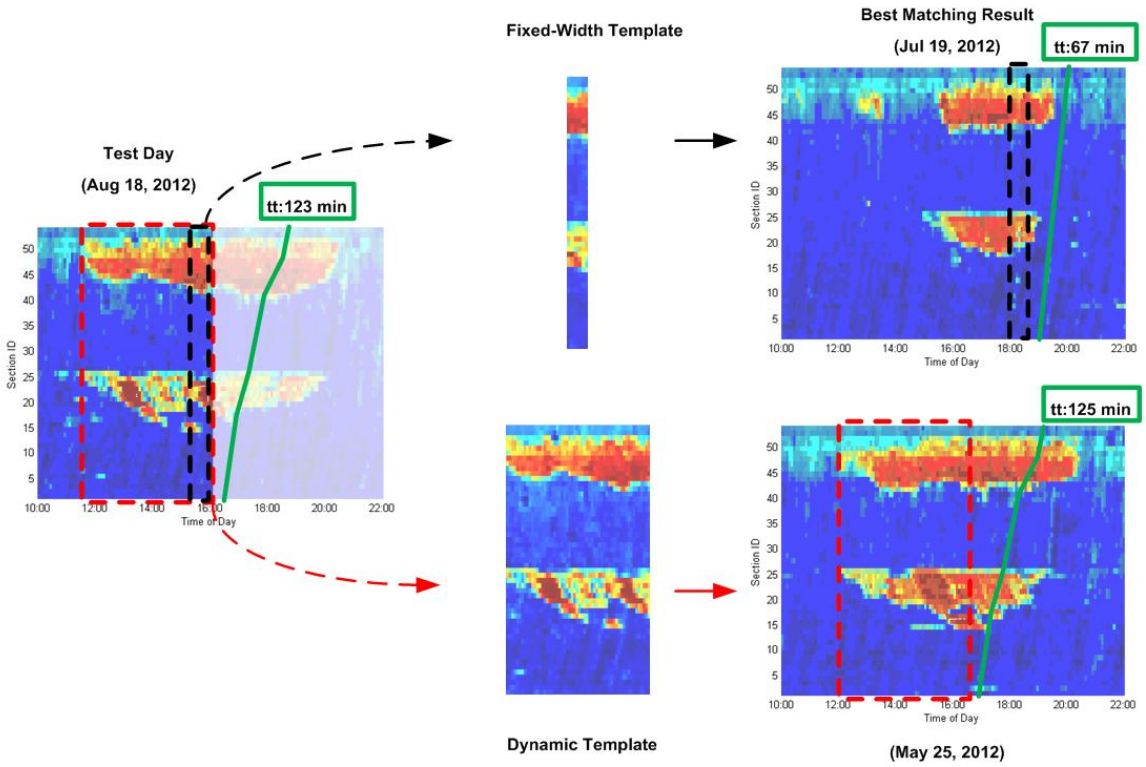


Figure 7.6: Illustration of fixed template width method vs. dynamic template matching.

The travel time curves produced by instantaneous travel time, fixed template matching, dynamic template matching method for a prediction horizon of 15 minutes, are compared with the ground truth data for a typical weekday on August 15, 2012 and a typical weekend on August 18, 2012 in Figure 7.7. The similar trends can be found on the results of two sample days. The instantaneous travel time experiences a temporal lag to the ground truth data, especially at the shoulders of the peak. Specifically, the instantaneous travel time highly underestimates the ground truth values when congestion is forming, and overestimates the ground truth travel time when congestion is dissipating. Comparatively, the fixed template method improves this problem and the predicted travel times have closer fit to the ground truth curve. It should be noted that Method 1 and 2 produce the same prediction results during uncongested traffic condition, even during the time period of congestion forming, since the dynamic template uses the same window width to the fixed template in these conditions. However, the proposed dynamic template matching method produces further improvements under congestion sustaining and dissipating periods, especially for the days with long congestion periods. The reason lies on the fact that the fundamental traffic flow theories help to find the bottleneck activation time and then the similar days with long congestion periods or severe bottlenecks can be accurately selected to improve the prediction accuracy.

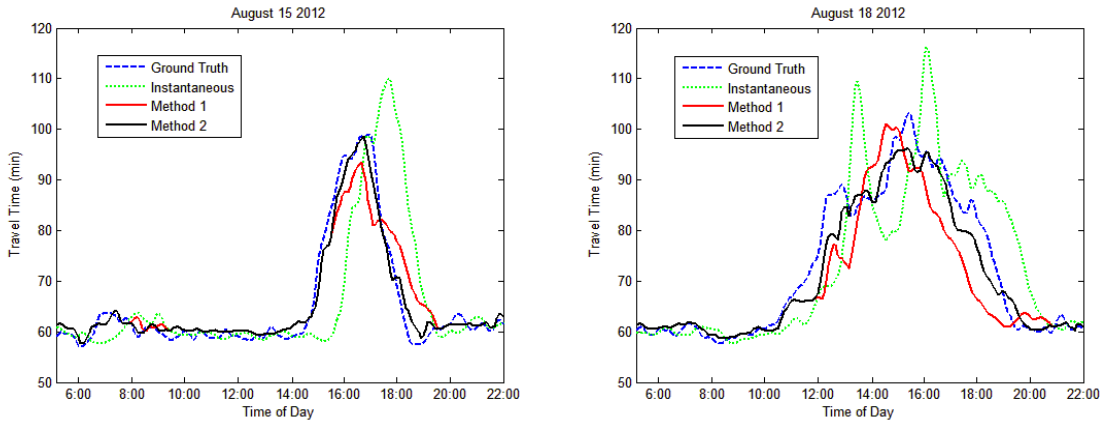


Figure 7.7: Travel time prediction results by three methods for 15 min prediction horizon and the confidence intervals by dynamic template matching on (a) August 15, 2012 (Wednesday); (b) August 18, 2012 (Saturday).

7.4 Conclusions

This study develops a travel time prediction algorithm by matching traffic patterns from historical data to current real-time conditions. Instead of using a fixed template size, the proposed method uses a dynamic template which is updated on each time interval according to the real-time traffic condition and bottleneck shape. A fast Fourier transform based method is used in the template matching to speed up the computation process for real-time application. The selected similar traffic patterns are then used to predict experienced travel times for departures from the current time or from future time intervals. A freeway stretch from Richmond to Hampton is selected as the test site to evaluate the prediction accuracy of the proposed algorithm. The section-based INRIX data along the selected freeway is used to evaluate the performance of different predictors.

The case study demonstrates that the proposed method produces much higher and very stable prediction results for prediction horizon of 0 to 30 minutes, compared to instantaneous travel time and fixed template matching method. The comparison between fixed template size and dynamic template matching methods indicates that the dynamic template improves the prediction accuracy during congestion sustaining and dissipating periods. Furthermore, the proposed dynamic template matching approach has the flexibility of using an incremental historical dataset, which is proved to further improve the prediction accuracy instead of using the constant historical dataset.

The proposed algorithm employed during this study provides a framework to use template matching technique to correlate real-time and historical traffic data to predict experienced travel times. More advanced pattern recognition techniques can be considered to enhance the prediction accuracy or save the computation cost for future research.

Chapter 8

8 Travel Time Prediction using Particle Filtering

This chapter is an edited version of: Hao Chen and Hesham A. Rakha (2014), "Real-time Travel Time Prediction by Particle Filtering with non-Explicit State-Transition Model," *Transportation Research Part C: Emerging Technologies*. In press

The research presented in this chapter develops a particle filter approach for the real-time short to medium-term travel time prediction using real-time and historical data. Given the challenges in defining the particle filter time update process, the proposed algorithm selects particles from a historical database and propagates particles using historical data sequences as opposed to using a state-transition model. A partial resampling strategy is then developed to address the degeneracy problem by replacing invalid or low weighted particles with historical data that provide similar data sequences to real-time traffic measurements. As a result, each particle generates a predicted travel time with a corresponding weight that represents the level of confidence in the prediction. Consequently, the prediction can produce a distribution of travel times by aggregating all weighted particles. A 95-mile freeway stretch from Richmond to Virginia Beach along I-64 and I-264 is used to test the proposed algorithm. Both the absolute and relative prediction errors using the leave-one-out cross validation concept demonstrate that the proposed method produces the least deviation from ground truth travel times, compared to instantaneous travel times, two Kalman filter algorithms and a K nearest neighbor (k-NN) method. Moreover, the maximum prediction error for the proposed method is the least of all the algorithms and maintains a stable performance for all test days. The confidence boundaries of the predicted travel times demonstrate that the proposed approach provides good accuracy in predicting travel time reliability. Lastly, the fast computation time and online processing ensure the method can be used in real-time applications.

8.1 Introduction

Many studies have been conducted to estimate or predict traffic states (e.g. flow, speed and density) [17, 18, 21, 51, 102-104] and travel times [7, 13, 30, 105-107]. Among the existing methods, the use of macroscopic traffic models within a Bayesian filtering framework has gained popularity in recent years to address real-time estimation and prediction problems [6, 7, 17, 18, 51, 103].

There are two main advantages of using the combination of macroscopic traffic models with recursive Bayesian filters. First, within the time update process, the relationship between traffic state parameters over adjacent time intervals, also known as the state-transition model, is accurately characterized using macroscopic traffic models. Compared to previous studies that construct a relationship between predicted and previous traffic states (state-transition model) using data-driven methods [8, 108] or simply based on neighboring variables [30, 106], macroscopic traffic models provide an analytical solution to model the state-transition function that is explainable through physical principles of traffic stream movement and thus is not affected by noises in data-driven methods. More importantly, macroscopic traffic models can respond to non-recurrent or sudden changes in traffic conditions. For example, the model parameters including roadway capacity and free-flow-speed can be adjusted in response to a traffic incident or inclement weather conditions. The other advantage lies in the fact that both the measurement update (estimation) and time update (prediction) processes are included in the Bayesian filtering framework each time interval. The sequence of the two processes within a single time interval categorizes the problem as either data estimation or prediction. Once a new measurement is available, it is used to adjust the prior predicted value and obtain the estimation. Conversely, prediction is calculated by implementing the estimated value in the time update equation (state-transition model) [6]. Consequently, the Bayesian filtering framework can deal with both estimation and prediction problems.

For real-time travel time prediction problems, the state-transition models in previous studies are usually simply defined by the time series trends from near-past or historical travel times. In this way, the nonlinear transition function between adjacent travel times is divided into discrete linear functions. Consequently, Kalman filter based travel time prediction algorithms are proposed [30, 36, 106, 107]. Although those methods demonstrate better performance compared to other naive methods using real-time or historical data, the assumption of Gaussian noise in the Kalman filter may not always be consistent with field data [104]. Moreover, it should be noted that such Kalman filter methods are essentially based on the simplified linear state-transition function. However, the travel times of neighboring time intervals have a strong nonlinear relationship considering the nonlinear traffic stream behavior. It should be mentioned that Extended Kalman Filter (EKF) methods can deal with nonlinear problems using Tayler estimation and thus are widely used for traffic state estimation [18, 19]. EKF is a revised classical Kalman filter with the calculation of the Jacobian expression. However, sometimes it is very difficult to compute the Jacobian expression for nonlinear state-transition models. To overcome this problem, the Ensemble Kalman filter (EnKF) is proposed which enables the use of a fully nonlinear evolution equation while exploiting the linear

observation equation [51]. However, it cannot deal with the problem of nonlinear measurement.

Compared with previous mentioned filtering methods, the particle filter is a sequential Monte Carlo method with the advantages of addressing strong nonlinear dynamic problems and without the assumption of noise distributions [46]. Numerous studies have demonstrated the advantages of the use of particle filters over Kalman filters for various applications with nonlinear state-transition models [46, 47]. In the field of transportation, an Unscented Particle filter (UPF) was tested and demonstrated to outperform Kalman filter methods for traffic state estimation [104], and a particle filter method was demonstrated to produce half the prediction error for traffic speed estimation when compared to an EnKF algorithm [17]. Considering the aforementioned nonlinear traffic behavior, particle filters provide a better Bayesian filtering solution for travel time prediction.

However, another key problem still exists with the use of particle filter techniques to accurately model the travel time state transition function. Recently, several revised particle filter approaches have been developed to predict state values in other domains without specified state-transition models. For example, a variant of particle filtering algorithms is proposed to track the eye location of tropical cyclones using historical data. An explicit state update is not required in the approach since the prior distribution is predicted using historical trends [81]. In addition, a memory-based particle filter is proposed for tracking abrupt face changes under occlusions. This method can handle nonlinear, time-variant and non-Markov dynamics, which employs a random sampling from the history to generate prior distributions [80, 109]. Moreover, a chaotic modeling of nonlinear dynamical systems is proposed for prediction problems without the underlying dynamic models. Given an initial condition, the predictions of state variables are accomplished using kernel regression [86]. Consequently, the proposed approach uses these concepts for predicting experienced travel times.

Unlike previous travel time prediction studies, in this paper we develop multi-step predictions that estimate future travel time departures up to one hour later. Previous research has demonstrated that prediction accuracy typically deteriorates quickly using Artificial Neural Networks (ANNs) for multi-step predictions [14, 15]. Recently, a novel travel time prediction method was developed by considering temporal-spatial input dynamics in recurrent Neural Networks [16]. The test results using five different ANNs under various scenarios with or without incident data demonstrated the errors for 5-step-ahead prediction were nearly twice the errors for 1-step-ahead prediction. Moreover, off-line data training is needed in the ANN method, which makes it difficult to use ANNs to predict non-recurrent conditions or transfer to other locations. The proposed method can efficiently address these issues and produce higher accuracy for multi-step prediction.

In conclusion, the research presented here develops a particle filter approach for travel time prediction using real-time and historical data. Unlike previous studies that require an underlying physical model in modeling the state-transition function between predicted and previous travel times, the proposed particle filter uses historical trends to model the state-transition trend. The identified invalid or low weighted particles are removed and

replaced in order to overcome the degeneracy problem in particle filters. Using the particle weights, the travel time reliability can be predicted by aggregating all the particles for each time interval. Probe data from Richmond to Virginia Beach are used to test and evaluate the performance of different prediction methods. The results indicate that the proposed method produces the least prediction error compared to the instantaneous method, two Kalman filters and a *k*-NN method for multi-step-ahead travel time prediction. A sensitivity analysis is also conducted to explore the impacts of different model parameters on the prediction accuracy.

The remainder of this chapter is organized as follows. Background information about Bayesian filtering and particle filtering techniques are provided in the next section. Subsequently, the framework of the proposed particle filter approach that does not use an explicit state-transition model is presented. This is followed by a description of the test data for the case study and a comparison of results using different prediction methods. The last section provides the conclusions of this chapter and future research recommendations.

8.2 Background

The new approach is developed from the concept of the Bayesian filter under the situation that only historical data are available instead of using an explicit state-transition model. Consequently, the theoretical background of the Bayesian filter and the general representation of a particle filter are initially introduced in this section.

When considering the problem of state tracking, the propagation of the state sequence using a state-transition model and the system update using measurement data are given by

$$x_t = f_t\left(x_{t-1}, \varphi_{t-1}\right) \tag{8.1}$$

$$z_t = h_t\left(x_t, \gamma_t\right) \tag{8.2}$$

where x_t and z_t represent the state variable and the data measurement at time interval t, respectively; φ_t and γ_t are time update and measurement update noises. Bayesian filters represent a general probabilistic approach to estimate the posterior probability density function (pdf) of a target state variable x_t at each discrete time interval t, using given past measurement data $z_{1:t}: \{z_1, z_2, \ldots, z_t\}$. Specifically, the conditional density $p(x^{n+1}|y^{1:n})$ is recursively updated according to Equation (8.3) and Equation (8.4) as shown below [110].

$$p\left(x_t \mid z_{t-1}\right) = \int p\left(x_t \mid x_{t-1}\right) p\left(x_{t-1} \mid z_{t-1}\right) dx_{t-1} \tag{8.3}$$

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})} \tag{8.4}$$

where $p(x_t|x_{t-1})$ is the probability of system evolution, given by the time update process of Equation (8.1); and $p(z_t|x_t)$ is a likelihood function defined by the measurement update process of Equation (8.2). However, the analytical solution of $p(x_t|z_{1:t})$ is difficult to calculate directly. In a particle filter approach, the posterior pdf of $p(x_t|z_{1:t})$ is represented by a set of random samples with corresponding weights. When the number of samples is large enough to approach infinity, these particles approximate the equivalent representation of the posterior pdf [46]. Suppose $x_t: \{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$ denotes a collection of particles, in which $x_t^{(i)}$ is the state value and $w_t^{(i)}$ is the corresponding weight of the i^{th} particle at time t. The posterior pdf can be approximated using Equation (8.5), and the weights updated using Equation (8.6).

$$p(x_t \mid z_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}) \tag{8.5}$$

$$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot \frac{p(z_t \mid x_t^{(i)})\, p(x_t^{(i)} \mid x_{t-1}^{(i)})}{q(x_t^{(i)} \mid x_{t-1}^{(i)}, z_t)} \tag{8.6}$$

Where $q(x_t^{(i)}|x_{t-1}^{(i)}, z_t)$ is the importance density, which is a known pdf chosen to generate the particles. If the importance density is chosen to be the same as the prior pdf $p(x_t^{(i)}|x_{t-1}^{(i)})$, then the weight update in Equation (8.6) is simplified to Equation (8.7) [46, 111]. Following this approximation, the original integral calculation of Equation (8.3) is transformed to an easier formulation of calculating the summation of particles with corresponding weights.

$$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot p(z_t \mid x_t^{(i)}) = w_{t-1}^{(i)} \cdot p_{e_t}\left(z_t - h_t\left(x_t^{(i)}\right)\right) \tag{8.7}$$

After the arrival of new measurement data, weights are updated considering the importance of corresponding particles. According to the calculation of likelihood $p(z_t|x_t^{(i)})$, the smaller error between a prediction and a measurement data results in the larger weight is assigned to the corresponding particle. In this way, the particle filter comprises recursive propagation of the weights and updates the state variable when new measurements are obtained. However, a common problem exists during weight updating in particle filter approaches, namely: the degeneracy problem. In this problem, the variance of weights can only increase over time, which results in all but one particle having negligible weights after several iterations. Although it is impossible to avoid the degeneracy problem, previous studies introduced resampling as an efficient alternative to reduce the effects of degeneracy [47]. The basic idea of resampling is to eliminate particles with small weights and to concentrate on particles with large weights. The sampling importance resampling (SIR) particle filter is described in Table 8.1, which is derived from the sequential importance sampling algorithm by choosing the importance density to be the transitional prior and by performing the resampling step at every time

interval. This approach has the advantage that the importance density can be easily updated and the important weights are easily evaluated [46]. The concepts of the Bayesian filter and SIR particle filter are the bases to construct the proposed algorithm under the condition that the state-transition model $p(x_t|x_{t-1})$ is not explicitly given.

Table 8.1: Sampling importance resampling particle filter.

$$\left[\left\{x_t^{(i)}\right\}_{i=1}^{N}\right] = SIR\left[\left\{x_{t-1}^{(i)}\right\}_{i=1}^{N}, z_t\right]$$

$-$*Initialize particles*

$$\text{Draw } x_0^i \sim p(x_0),\ i \in [1:N]$$

Step 1: *Time update*

$$\text{Draw } x_t^i \sim p\left(x_t \mid x_{t-1}^i\right), i \in [1:N]$$

Step 2: *Measurement update*

$$w_t^{(i)} \propto p\left(z_t \mid x_t^{(i)}\right) = p_{e_t}\left(z_t - x_t^{(i)}\right), i \in [1:N]$$

Step 3: *Resampling*

Initialize the cumulative density function: $c^{(1)} = w_t^{(1)}$

For $i = 2:N$

Construct the cumulative density function: $c^{(i)} = c^{(i-1)} + w_t^{(i)}$

End For

Let $i = 1$, *draw a starting point*: $u_1 \sim U(0, N^{-1})$

For $j = 1:N$

$u^{(j)} = u^{(1)} + N^{-1}(j-1)$

$i = i + 1, when\ u^{(j)} > c^{(i)}$

Assign sample: $x_t^{(j)} = x_t^{(i)}$

End For

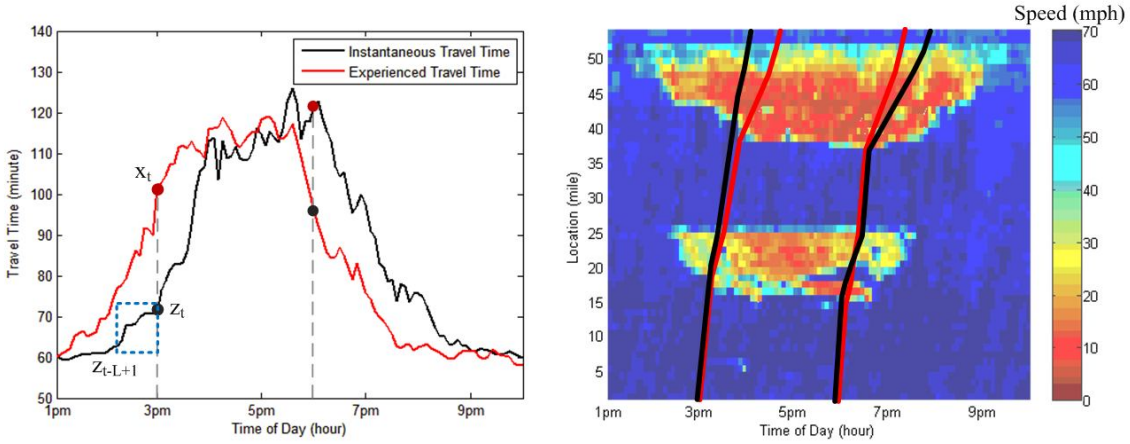
8.3 Methodology

The implementation of the traditional Bayesian filter approach is typically challenging in travel time prediction, since the state-transition model, which characterizes the relationship between predicted and previous travel times is difficult to quantify. Consequently, a data-driven method based on historical data sampling is proposed in this paper to address this problem. The definition of the problem is presented first in this section, followed by a description of the proposed solution and the related problems.

8.3.1 Definitions and Denotations

The methodology in this paper attempts to predict experienced travel times using real-time and historical data. The experienced travel time is the actual, realized travel time that a vehicle could experience during a trip. Comparatively, the instantaneous travel time is the summation of section travel times at the same time interval. The instantaneous travel time is close to the experienced travel time when the roadway speed does not change significantly across time space during the entire trip, e.g. free-flow conditions. Nevertheless, instantaneous travel times may deviate substantially from the experienced travel time under transient states during which congestion is forming or dissipating during the trip [5].

Various traffic sensing technologies have been used to collect traffic data for use in computing travel times, including point to point travel time collection (license plate recognition systems, automatic vehicle identification systems, mobile, Bluetooth, probe vehicle, etc.) and station based traffic state measuring devices (loop detector, video camera, remote traffic microwave sensor, etc.). Private companies such as INRIX integrate different sources of measured data to provide section-based average speed or travel time, which can be used to construct traffic speed matrix over spatial and temporal and thus is used in this paper. The benefit of using temporal-spatial speed data is that travel time can be easily calculated afterward [99]. More importantly, such data provides the flexibility for scalable applications on traffic networks.



(a) (b)

Figure 8.1: Representation of field data for May 19 2012 on I-66; (a) instantaneous and experienced travel times; (b) traffic speed contour.

In order to demonstrate the research problem, a contour plot of spatiotemporal traffic speed and the corresponding instantaneous and experienced travel times are illustrated on Figure 8.1. The speed contour is the five-minute aggregated INRIX probe data along I-66 between Richmond to Hampton Roads on May 19, 2012 from 1 PM to 10 PM, and traffic speed is represented by different colors from red (congested) to blue (uncongested) in

Figure 8.1 (b). Based on the spatiotemporal speed data, trip trajectories can be plotted to calculate instantaneous and experienced travel times, which are denoted by black and red curves, respectively. For trip departure time of 3 PM, the instantaneous travel time (72 minutes) underestimates the experienced travel time (102 minutes) by 30 minutes. Conversely, the instantaneous travel time (123 minutes) on 6 PM overestimates the corresponding experienced travel time (96 minutes) by 27 minutes. The above examples of two trips demonstrate the discrepancy between instantaneous and experienced travel times. In order to develop a travel time prediction approach for real-time applications, the experienced travel time is required, since the instantaneous travel time is not a good indicator of the actual travel time, especially at the shoulders of the peak period. The experienced travel time cannot be measured until a traveler completes their trip and thus the experienced travel time departure for the previous time step (e.g. five minutes earlier) may not be available at the current time, especially for long trips.

Considering the above analysis, the state-transition and measurement update formulations defined in Equation (8.1) and (8.2) are used in the proposed particle filter. It should be noted that both the state and measurement update equations are nonlinear functions in our application. The experienced travel time at time t is defined as the state variable x_t, and the state-transition function f_t or $p(x_t|x_{t-1})$ represents the nonlinear relationship from x_{t-1} to x_t. The instantaneous travel time sequence from short-past $t-L+1$ to current time t is defined as the measurement variable z_t,, which also represents the traffic pattern at current time t and is highlighted by the blue rectangular box in Figure 8.1 (a). Here, L denotes the length of the data sequence.

Since it is difficult to find an analytical solution for the state-transition model, historical data are used here to provide the pool of past information including traffic trends of instantaneous and experienced travel time sequences, which can be used to replace the state transition model. In the proposed method, assume historical data is denoted by Ω, the state variable x_t is approximated by a set of particles $\{x_t^{(i)}\}_{i=1}^{N}$ and each particle $x_t^{(i)}$ corresponds to the experienced travel time $\Omega_{exp}(d_t^{(i)},j_t^{(i)})$ at time $j^i{}_t$ on the historical day $d^i{}_t$. Moreover, each particle $x_t^{(i)}$ is also associated with traffic pattern $y_t^{(i)}$ represented by the tail value $\Omega_{inst}(d_t^{(i)},j_t^{(i)})$, which is used to match with the real-time traffic pattern and calculate the particle weight. Specifically, the real-time traffic pattern is used to update particles and calculate the weight of each particle based on the dissimilarity of two traffic patterns. The reason for using travel time sequences instead of a single values to represent traffic patterns is that more dynamic information are included in the data sequence and potentially can improve the accuracy of matching traffic patterns between real-time measurements and historical data. Here it should be noted that the nonlinear function h_t in the measurement update equation, which captures the relationship between experienced travel time and traffic pattern (instantaneous travel time sequence), can also be used to describe the correlation between $x_t^{(i)}$ and $y_t^{(i)}$.

8.3.2 Particle filtering with non-explicit state-transition model

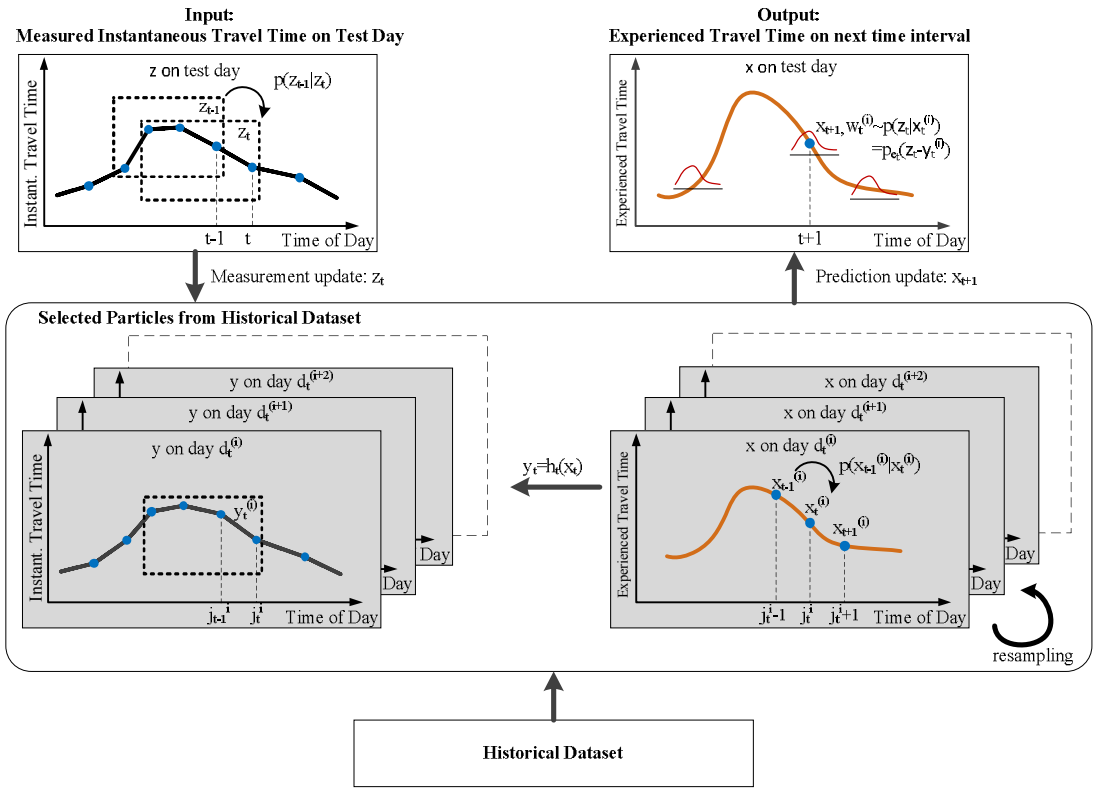


Figure 8.2: Demonstration of the proposed particle filter approach.

A graphical representation of the proposed approach of non-explicit state-transition particle filter (NSPF) is demonstrated in Figure 8.2. The input data are the measured instantaneous travel times for each time interval, the update of measurement data from z_{t-1} to z_t is conducted by shifting the data sequence window one time step forward. Each particle can be recognized as a data sequence of instantaneous travel times and a data sequence of experienced travel times on the same historical day. The time update of the particle filter from $x_{t-1}^{(i)}$ to $x_t^{(i)}$ is accomplished by shifting one step ahead along the data sequence of experienced travel time. For each particle, the corresponding traffic pattern $y_t^{(i)}$ can be derived according to the relationship with $x_t^{(i)}$ represented by $y_t=h_t(x_t)$. At the same time, the associated weight $w_t^{(i)}$ can be calculated as the likelihood $p(z_t|x_t^{(i)})$, which can be accomplished by comparing the dissimilarity between real-time and historical traffic pattern as $p_{e_t}(z_t - y_t^{(i)})$. In this study, the likelihood function p_{e_t} is chosen as a normal distribution $N(0,1)$ [111]. Consequently, the distribution of experienced travel time on the next time interval $t+1$ can be predicted as $\{x_{t+1}^{(i)}, w_t^{(i)}\}_{i=1}^{N}$. For multi-step prediction with prediction horizon $t+p$, the propagation along experienced travel time sequence on historical day can be iteratively conducted but the same weight updated by the current measurement is maintained for each particle. So the experienced travel time on $t+p$ can be predicted as $\{x_{t+p}^{(i)}, w_t^{(i)}\}_{i=1}^{N}$.

Considering the application of multi-step travel time prediction, the similar structure as SIR particle filter method is revised to develop the proposed algorithm, which includes the steps of initialization, time update process, measurement update process, resampling

and travel time prediction. The rest of this section describes the details of each step in the proposed particle filter method and the pseudo code of these steps is presented in Table 8.2.

Since there is no explicit state-transition model in the proposed method, the initial travel time values and the corresponding traffic patterns should be assigned to all the particles. For each particle $x_0^{(i)}$, the initialization process is accomplished by randomly selecting the day index d_0^i and the time index $j_0^{(i)}$ on that day from the historical data set. Therefore, the corresponding experienced travel time and traffic pattern are $\Omega_{exp}(d_0^{(i)},j_0^{(i)})$ *and* $\Omega_{inst}(d_0^{(i)},j_0^{(i)})$ respectively.

Comparing to the SIR particle filter described in Table 8.1, each particle in the proposed algorithm propagates along a historical experienced travel time sequence as opposed to using a state-transition model. For a particle $x_{t-1}^{(i)}$ at time *t-1*, the time update $p(x_t|x_{t-1})$ is conducted by shifting the time index $j_{t-1}^{(i)}$ one step ahead and keeping the same day index. A follow up process attempts to identify valid particles that provide a sufficient time interval buffer considering the prediction horizon *p*. Consequently, the invalid particles are selected if the corresponding experienced travel time sequences cannot provide prediction output by shifting the time index by *p*. Here, the last time interval of the historical experienced travel time sequence on day $d_t^{(i)}$ is denoted by $H_{d_t^{(i)}}$. Unlike the time update process using a continuous state-transition model, the prediction in the proposed algorithm considers the boundary of data sequence for each historical day. Although the end of one day is connected with the beginning of the following day, the option of moving to the subsequent data sequence is replaced by the alterative of resampling. The reason to use this alternative lies in the consideration that the travel time sequence for the next day may not be available in the historical data set. In this way, the valid particles with respect to prediction horizon *p* are identified as collection Ψ_t for the t^{th} time interval.

The measurement update process attempts to calculate the weights of all valid particles in Ψ_t. For each valid particle $x_t^{(i)}$, the weight is calculated by the likelihood function p_{e_t} with input of corresponding traffic pattern $y_t^{(i)}$ and real-time traffic pattern z_t. Thereafter, all the valid particles are sorted according to the associated weight values in descending order. The top N_{th} particles are maintained and the remaining particles are resampled in the next step. In this way, all the particles are divided into two groups. The first group includes the particles with large weights. The second group includes the invalid particles that cannot provide prediction values (exceed data sequence boundary) or the particles with negligible weights. The second group of particles will be re-selected in the next process so that new particles with similar traffic patterns to the current time interval can be selected.

It should be noted that the problem of degeneracy in the SIR particle filter also exists in the proposed method. A resampling procedure is proposed to tackle this problem. The traditional threshold-based resampling strategies include residual, stratified and systematic resampling. These methods are used to eliminate samples with low importance weights and multiply samples with high importance weights [46]. The procedure proposed in this study is used for the same purpose. To save on computation time, a

partial resampling method is developed instead of the threshold based resampling strategies [112], as will be discussed in further detail.

Table 8.2: Multi-step travel time prediction by proposed particle filter approach (NSPF).

$$\left[\left\{x_t^{(i)}\right\}_{i=1}^{N}\right] = NSPF\left[\left\{x_{t-1}^{(i)}\right\}_{i=1}^{N}, z_t, \Omega\right]$$

– *Initialize particles* $x_0 : \left\{x_0^{(i)} \mid x_0^{(i)} = \Omega_{\exp}\left(d_0^{(i)}, j_0^{(i)}\right), i \in [1:N]\right\}$

$d_0^{(i)}$ = *randomly select a day from* $[1,2,...,D]$, $j_0^{(i)}$ = *randomly select a time index at day* $d_0^{(i)}$, $i \in [1:N]$

Step 1: *Time update*

Propagate the particles by drawing $x_t^{(i)} \sim p\left(x_t^{(i)} \mid x_{t-1}^{(i)}\right)$

$d_t^{(i)} = d_{t-1}^{(i)}, j_t^i = j_{t-1}^i + 1, \ i \in [1:N]$

Identify valid particles with respect to prediction horizon p

$\Psi_t = \left\{i \mid j_t^{(i)} \le H_{d_t^{(i)}} - p, i \in [1:N]\right\}$

Step 2: *Measurement update*

$w_t^{(i)} \propto p\left(z_t \mid x_t^{(i)}\right) = p_{e_t}\left(z_t - y_t^{(i)}\right), i \in \Psi_t$

Select N_{th} *number of particles with least weight values*

For $j = 1 : N_{th}$

$x_t^{(j)} = x_t^{(i)}, w_t^{(j)} = w_t^{(i)}$, *when* $i = \underset{i \in \Psi_t}{\arg\max}\, w_t^{(i)}, \Psi_t = \Psi_t - \{i\}$

End For

Step 3: *Resampling*

For $j = N_{th} + 1 : N$

Calculate the probability of selecting each historical day λ_t^n

$\lambda_t^n = p_{e_t}\left(z_t - \Omega_{inst}\left(n, \sigma_t^n\right)\right)$, *when* $\sigma_t^n = \underset{k \in [L, H_n - p]}{\arg\min}\left(z_t - \Omega_{inst}(n,k)\right), \ n \in [1:D]$

$d_t^{(j)}$ = *randomly select a day from* $[1,2,...,D]$ *according to the probability* $[\lambda_t^1, \lambda_t^2, ..., \lambda_t^D]$

$x_t^{(j)} = \Omega_{\exp}\left(d_t^{(j)}, \sigma_t^{d_t^{(j)}}\right), w_t^{(j)} = \lambda_t^{d_t^{(j)}}$

End For

Step 4: *Prediction*

Draw $x_{t+p}^{(i)} \sim p\left(x_{t+p}^{(i)} \mid x_t^{(i)}\right), i \in [1:N]$

$$\overline{x_{t+p}} = \sum_{i=1}^{N} w_t^{(i)} \cdot x_{t+p}^{(i)} \Big/ \sum_{i=1}^{N} w_t^{(i)}$$

In the resampling process, the remaining N-N_{th} particles in the second group of the measurement update process are resampled from the historical data set. During the resampling process only the historical data that have traffic patterns similar to the real-time measurements are selected to increase the efficiency of particle propagation. Consequently, a process is developed to calculate the maximum similarity between the traffic patterns from each historical day and the current day/time interval. The maximum similarity of each historical day is transferred to a probability using a likelihood function.

In this way, the day with closer similarity to the current traffic pattern has a larger probability to be selected in the resampling process.

In the proposed algorithm, the multi-step prediction is conducted by iteratively shifting the time step along the corresponding experienced travel time sequence of each particle. At the same time, each particle maintains the same weight until a new measurement is obtained. Consequently, the aggregated results from all the particles can provide the travel time distribution prediction instead of a single expected value. Moreover, the average prediction result can also be calculated as the weighted average travel time of each particle.

8.4 Case study

To test the performance of the proposed travel time predictor, an empirical study is conducted in this section. The test environment is introduced, and then different prediction methods are implemented on the same test data set. Finally, the testing results and discussion are presented.

8.4.1 Test environment setup

A freeway stretch from Richmond to Virginia Beach (95 miles long) connected by I-64 and I-264 is selected as the test site in this study. This test site usually experiences high traffic volumes and serious congestion during the summer season, since Virginia Beach is a famous resort location, and the selected freeway stretch serves as the main route heading to the beaches. Consequently, efficient and accurate travel time prediction is needed for travelers in planning their trips and reducing traffic congestion around the The evaluation of travel time prediction on the test site is conducted based on probe data from INRIX. The data provided by INRIX are mainly collected by global positioning system (GPS)-equipped vehicles and supplemented with traditional road sensor data, as well as mobile devices and other sources [98]. The probe data on the test site covers 96 freeway segments with a total length of 95 miles. The average segment length is 0.65 miles long, and the length of each segment is unevenly divided in the raw data from 0.1 to 6.36 miles. The location of the study site and the deployment of segments are presented in Figure 8.3. The raw data provide the average speed for each segment and are collected at one-minute intervals. In this study, the raw data are aggregated at five minute intervals to reduce the noise in calculating the travel times. The aggregated daily traffic data between 2 and 8 p.m. from May 16, 2012 to September 15, 2012 are considered in this study to test the algorithm since the most congested periods are observed during this time frame. Consequently, the prediction performances using different methods are investigated during the peak periods. For each selected day, the instantaneous travel times are calculated for each time interval by assuming the segment speed does not change over time. Given the length of each section of roadway and the corresponding average speed for each time interval, the instantaneous travel time is calculated based on the aggregation of segment travel times at a specific time interval. Conversely, the experienced travel time is calculated from the ground truth data by considering the change of segment speed

over time. In other words the speed profiles are piecewise constant speed values and the trip trajectory is a combination of diagonal curves over time and spaces [99].

In total, the travel time data for 123 days are included in this study. The test is conducted using the leave-one-out cross-validation method to provide a consistent validation test. During testing, the different methods are run on each day and the remaining days (122 days) serve as a historical data set. Finally, the average performance across the 123 test days is used to compare the prediction accuracy of the different methods. Leave-one-out cross-validation is a classic model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. Considering the fact that we only have four months of data on the selected freeway stretch, leave-one-out is an ideal option to quantify the prediction accuracy on this limited data. This method has also been used in many application fields including traffic prediction problems [113].

Several parameters in the proposed particle filter approach are pre-defined for the test. The data sequence length is chosen to be 6 periods (30 minutes), which entails the use of instantaneous travel times over half an hour as the input data sequence. The number of particles N is selected to be 200. The resampling threshold is set at 80% of the total number of particles, so N_{th} is 160. The impact of various parameters on the algorithm performance is quantified later in this section.

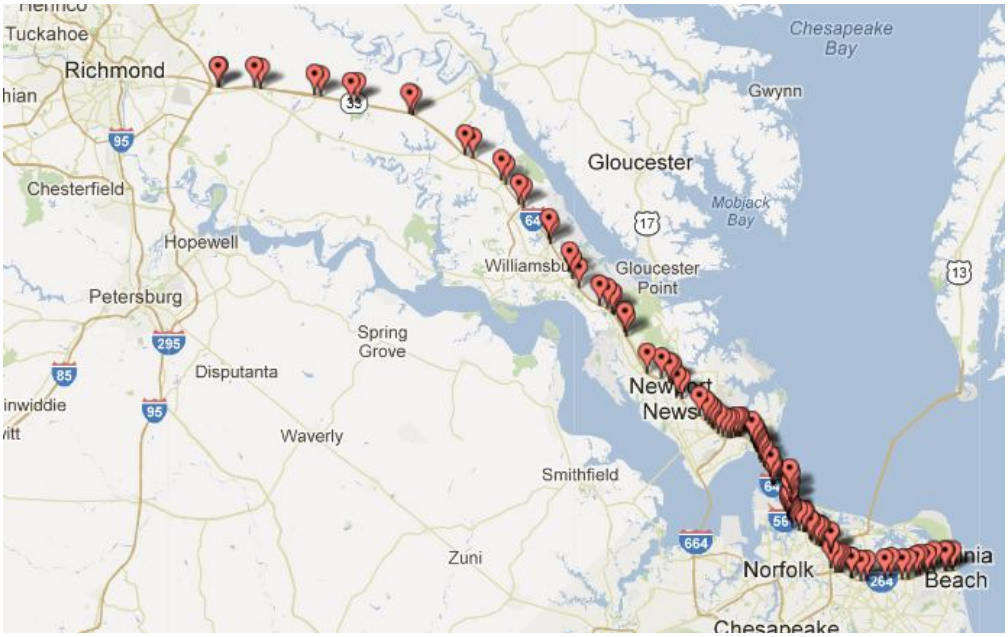


Figure 8.3: The study site from Richmond to Virginia Beach (source: Google map).

8.4.2 Comparison of algorithms and performance indices

To better evaluate the performance of the proposed predictor, several different methods are also considered on the same data set. The selected methods include state-of-the-practice instantaneous travel times, two types of state-of-the-art Kalman filter methods and a *k*-NN method. A detailed description of these methods is provided in this section.

The instantaneous travel time method is the easiest alternative to predict future travel times by assuming the current traffic speed along all the segments remains constant until the completion of the trip. This method is currently used by the Virginia Department of Transportation (VDOT) to display travel time information on variable message signs. Consequently, instantaneous travel times are considered the state-of-practice and used to quantify the tradeoff between simplicity and prediction accuracy.

As mentioned in the literature review, Kalman filters have been widely used in previous studies for real-time travel time predictions [30, 36, 106, 107]. In these studies, the problem of travel time prediction is modeled as a linear system as

$$\begin{cases} x_t = \Phi_{t-1} \cdot x_{t-1} + u_{t-1} \\ z_t = x_t + v_t \end{cases} \tag{8.8}$$

where state variable x_t is the predicted travel time at time t, and Φ_{t-1} is the state transition function to propagate travel times from time *t-1* to time t; z_t denotes the travel time measurement; u_{t-1} and v_t are system noises and are assumed to follow the standard normal distribution $N(0,1)$ in this study. Consequently, the travel time in the previous time interval is needed to calculate the predicted travel time. As far as the experienced travel time is concerned, Kalman filter methods cannot be used in real-time applications because previous experienced travel times are delayed significantly when the trip durations are long. Here, both of state and measurement variables are experienced travel times and we assume that the aforementioned problem can be ignored so that Kalman filter methods can be tested and compared with other predictors on the same data set.

According to previous studies using Kalman filter methods, the state-transition function is the key element that requires specification. Consequently, two types of Kalman filter methods are used in this study using two different methods to define the state-transition function. The transition function Φ_{t-1} in the first Kalman filter method (KF1) is defined as the ratio of measurement values from time interval *t-1* and *t-2* using Equation (8.9). This alternative is based on the assumption that traffic trends in the short past time periods will continue to propagate into the near future. Conversely, the state-transition function of the second Kalman filter method (KF2) is calculated using Equation (8.10) as the average measurement at time t and *t-1* using aggregated historical data on the same day-of-the-week as the testing day. This definition assumes that data trends from the same time intervals on historical days are consistent with the current day. For multi-step prediction considering a prediction horizon $t+p$, the state-transition function maintains a constant value of Φ_t and the time update equation is iteratively used for p times to calculate the prediction output x_{t+p}. It should be noted that the estimation value corrected by measurement z_t is used as the prediction output for a prediction horizon of zero.

$$KF1: \Phi_{t-1} = z_{t-1} / z_{t-2} \tag{8.9}$$

$$KF2: \Phi_{t-1} = \overline{z_t} / \overline{z_{t-1}} \tag{8.10}$$

The *k*-NN is another effective method, which is widely used to predict travel times for real-time applications [12, 34]. In order to conduct an objective comparison between the *k*-NN and the proposed method, the same instantaneous travel time sequence input z_t across L time intervals is used in the *k*-NN method. Thereafter, m numbers of similar data sequences with tail time $\{h_1,h_2,...,h_m\}$ can be selected as the candidates from the historical data set Ω as given in Equation (8.11). For each candidate with index i, a weight w_i is calculated using the average Euclidean distance from data sequences for the current time z_t and historical time h_i. Moreover, the corresponding experienced travel time departure at h_{i+p} on the selected candidate day i can be obtained from the historical data set. Consequently, the experienced travel time x_{t+p} on the current day can be predicted as the weighted average of the travel times from all candidates using Equation (8.12).

$$\begin{aligned} &H_c = \{h_1, h_2, \cdots, h_m\} \\ &For\ i = 1:m \\ &\quad h_i = \arg\min_{h \in \Omega} |z_t - h|, \Omega = \Omega - h_i \\ &End\ For \end{aligned} \tag{8.11}$$

$$\begin{aligned} w_i &= |z_t - h_i| \Big/ \sum_{i=1}^{m} |z_t - h_i| \\ x_{t+p} &= \sum_{i=1}^{m} x_{h_i+p} \cdot w_i \end{aligned} \tag{8.12}$$

Different combinations of parameters were tested and the optimum set of parameters was identified as $L = 5$ and $m = 20$, which corresponds to the least prediction error [12]. These parameters are used to test the *k*-NN method using the same data set and to serve as a comparison with other methods. It should be noted that the *k*-NN method is different from the proposed approach even if the number of candidates in the *k*-NN method is equal to the number of particles. The reason lies in the fact that there is no data propagation process in the *k*-NN algorithm, and all candidates are blindly selected from each time interval based on its similarity measure (shortest Euclidean distance) to the current travel time sequence.

To assess the different methods, the performance criteria are specified using both the absolute and relative prediction errors. The Mean Absolute Error (MAE) is the average absolute difference between the predicted travel time and ground truth using Equation (8.13). The corresponding Mean Absolute Percentage Error (MAPE) is the average absolute percentage change between the predicted and the true values relative to the true value as demonstrated in Equation (8.14).

$$MAE = \left(\sum \left| TT - \widehat{TT} \right| \right) \Big/ N_{TT} \tag{8.13}$$

$$MAPE = \left(\sum \left| TT - \widehat{TT} \right| \Big/ \widehat{TT} \right) \Big/ N_{TT} \tag{8.14}$$

where TT is the predicted travel time, $\widehat{TT}$ denotes the ground truth value of experienced travel time; and N_{TT} is the total number of predicted travel times.

8.4.3 Test results

The average absolute and relative prediction errors produced by the five methods are summarized in Table 8.3. As demonstrated in the table, the least prediction errors are produced by the proposed non-explicit state transition particle filter approach. Among all the methods, KF1 provides the worst performance, with a significant degradation in performance with an increase in the prediction horizon. The precondition of this method is that the short past travel time relationship continues into the near future. However, this assumption is less valid as the prediction horizon increases. KF2 produces marginally higher prediction errors compared to the instantaneous method for prediction horizons between 0 to 20 minutes and then produces slightly lower errors for longer prediction horizons. Such results demonstrate that the simple average values of measured travel times on the same day of week from previous weeks do not capture the change in experienced travel times. The *k*-NN method outperforms the instantaneous travel time method. The absolute error produced by the *k*-NN predictor increases from 10.69 to 14.79 minutes (a 38% increase) when the prediction horizon increases from 0 to 60 minutes. The results demonstrate that the average absolute error for the proposed predictor only increases from 8.26 to 10.54 minutes (an increase of 27%) when predicting the experienced travel time for departures from the current time to one hour later. Alternatively, the prediction error produced by the instantaneous travel time method increases at a much higher rate from 11.54 to 19.25 minutes, which is an increase of 67%. In conclusion the results demonstrate that the proposed algorithm outperforms the state-of-the-practice and state-of-the-art methods, especially for longer prediction horizons.

Table 8.3: Prediction results by different methods for various prediction horizons.

		Prediction Horizon (min)						
		0	10	20	30	40	50	60
Instantaneous	MAE (min)	11.54	12.97	14.37	15.74	17.04	18.21	19.25
	MAPE (%)	10.67	12.09	13.44	14.79	16.08	17.23	18.25
KF1	MAE (min)	12.45	14.16	16.34	19.01	21.61	24.09	26.40
	MAPE (%)	11.48	13.10	15.18	17.78	20.36	22.86	25.14
KF2	MAE (min)	12.22	13.29	14.39	15.53	16.45	17.18	17.89
	MAPE (%)	11.32	12.38	13.45	14.58	15.49	16.24	16.94
k-NN	MAE (min)	10.69	11.46	12.20	12.92	13.64	14.26	14.79
	MAPE (%)	9.38	10.09	10.75	11.38	12.01	12.52	12.97
NSPF	MAE (min)	8.26	8.81	9.27	9.65	10.03	10.30	10.54
	MAPE (%)	7.32	7.80	8.19	8.53	8.85	9.08	9.29

The relative absolute errors associated with the five methods are presented in Figure 8.4. The figure clearly demonstrates a significant degradation in the prediction accuracy with time for the KF1, KF2, and instantaneous methods. Alternatively, the *k*-NN and the proposed methods produce consistent errors over the one hour prediction horizon. Specifically, the relative absolute error produced by the *k*-NN method increases from 9.38% to 12.97% (a 38% increase), which is greater than the error associated with the NSPF method (a 27% increase). It should be noted that the relative errors produced by the proposed NSPF approach are always below 10% for the different prediction horizons within one hour, which indicates the prediction performance of the proposed method is much more reliable compared to the other four methods.

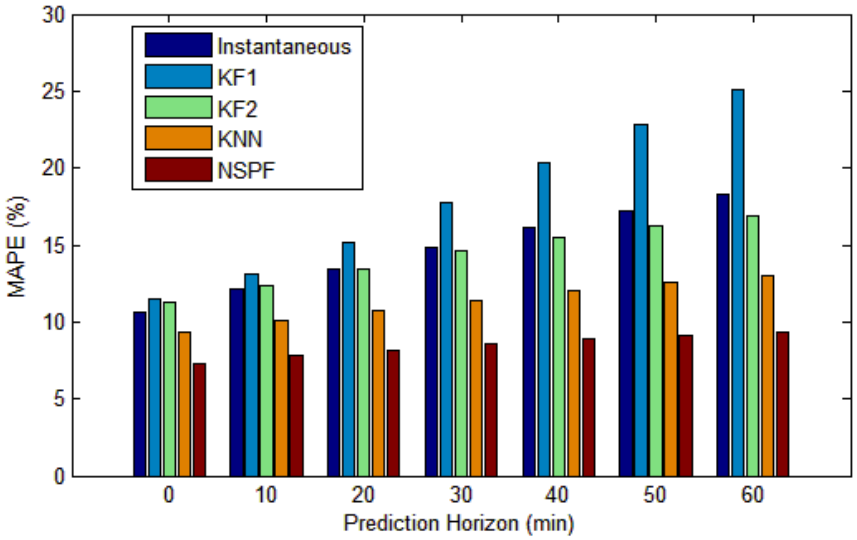


Figure 8.4: MAPE by different methods for various prediction horizons.

In order to investigate the maximum deviation between prediction results and ground truth data, the maximum MAPE produced by the five methods on different testing days in June 2012 are selected and presented in Figure 8.5. These results are for a prediction horizon of 30 minutes. The worst result is generated by the KF1 method (i.e., maximum MAPE ranging from 29.3% to 120%), followed by the KF2 and instantaneous methods (i.e., maximum MAPEs ranging from 20.9% to 84.3% and 13.6% to 81.5%, The maximum MAPE produced by the *k*-NN method varied between 20.4% and 43.2%, which is significantly better than the previous three methods. More importantly, the best performance corresponds to the proposed NSPF method with a maximum MAPE ranging from 13.5% and 38.2%. The results clearly demonstrate that the proposed predictor still produces the best performance even under the worst case conditions.

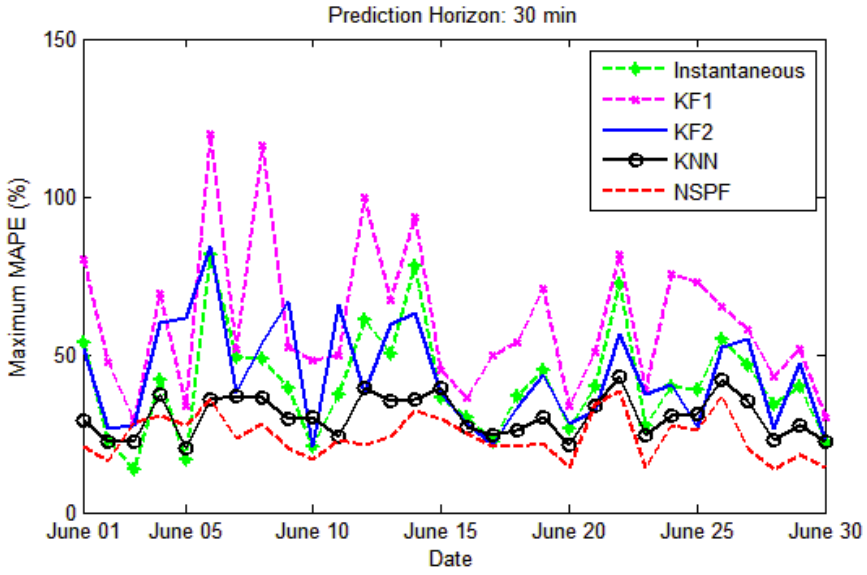


Figure 8.5: Maximum MAPE by different methods on June 2012.

Considering the prediction errors produced by the two Kalman filter methods are similar or worse than the instantaneous travel time estimates, these methods will not be considered in any further evaluations. The travel time curves for the remaining three methods are compared with the ground truth data for June 21, 23 and 29, as illustrated in Figure 8.6. These days include different traffic conditions on a weekday, weekend and Friday afternoon, respectively. Both the instantaneous and the *k*-NN method predictions experience a temporal lag relative to the ground truth data, especially during the formation and dissipation of congestion. Specifically, the instantaneous travel time method significantly underestimates the ground truth when congestion is forming, and overestimates the ground truth travel time when congestion is dissipating. Comparatively, the proposed method improves the prediction performance when congestion is forming and dissipating but still lags in some instances. For example, the red curve generated by the proposed method lags and oscillates between 2 to 4 p.m. on June 21, 2012. Similarly, the proposed method lags during the congestion period around 3 p.m. on June 29, 2012.

The NSPF not only predicts the expected travel time but can predict the travel time distribution. The 95% and 5% confidence intervals of the predicted travel times are calculated as the upper and lower boundaries in Figure 8.6. The gray shadow area between the boundaries covers most of the ground truth data temporal variation, which demonstrates that the proposed approach provides a good accuracy to predict travel time reliability. It should be noted that the researchers did not specify whether the historical data set comprised weekday or weekend data. This is one of the advantages of the proposed method. That is, if the test day is a weekend, the similar traffic pattern in historical weekends is automatically selected as a particle associated with larger weights, which contributes more significantly to the prediction.

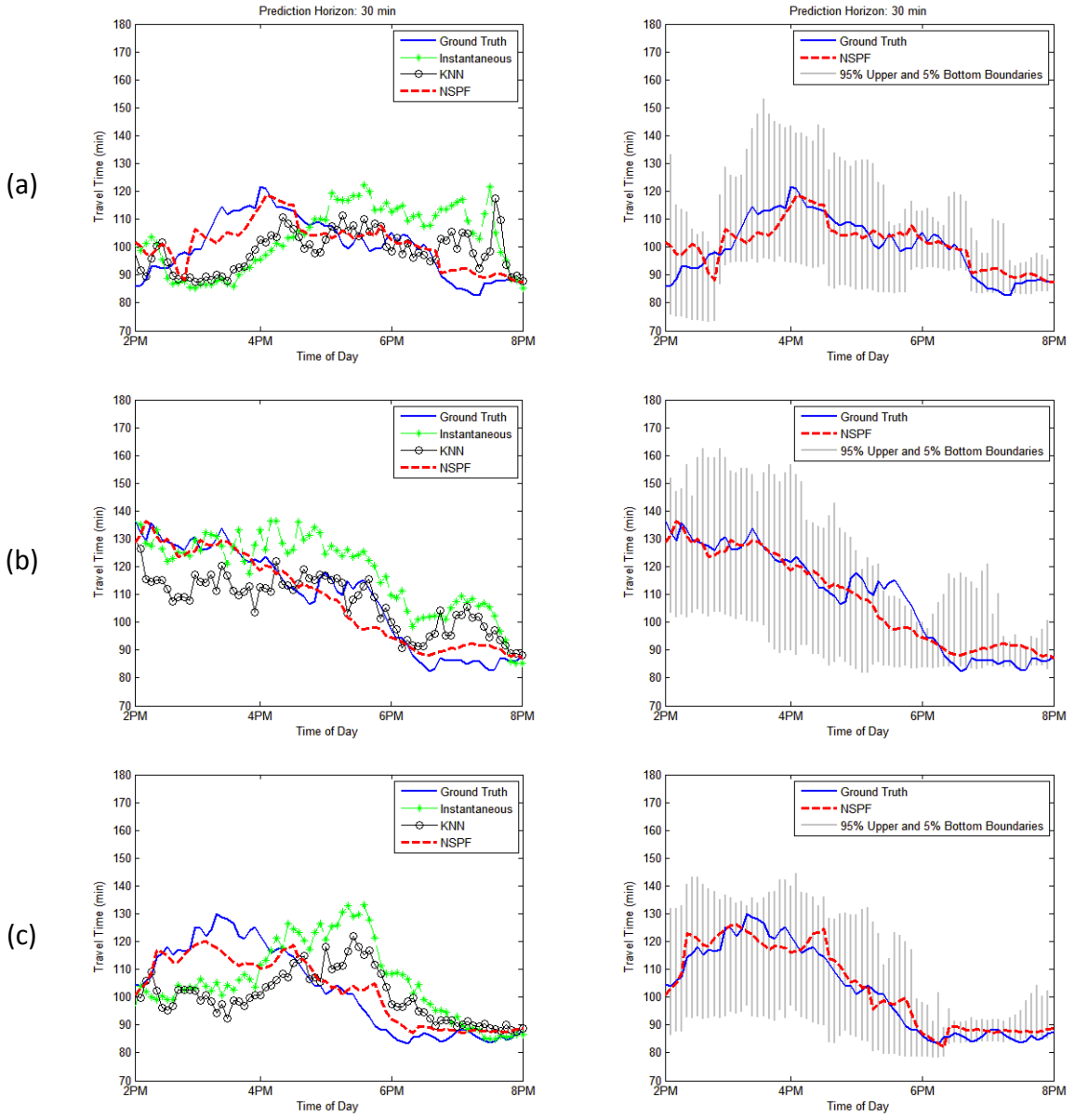


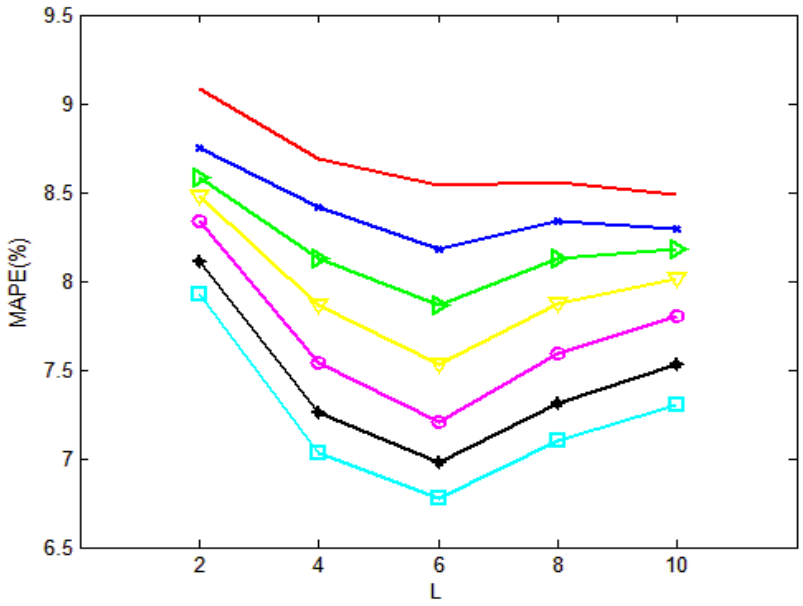
Figure 8.6: Travel time prediction results by three methods and the NSPF confidence boundaries on (a) June 21, 2012 (Thursday); (b) June 23, 2012 (Saturday); (c) June 29, 2012 (Friday).

Another advantage of the proposed NSPF approach is the fast computation time. This computational efficiency allows the model to be implemented in real-time in a TMC. The testing of the NSPF travel time predictor was performed on a personal computer with Intel dual core CPU, 2.40 GHz and 4GB of random-access memory within the MATLAB 2012b environment. Under the scenario of setting L=6, N_{th}=160 and N=200, the total average computation time for one day between 2 and 8 p.m. was 5.76 seconds. Consequently, the calculation of a single prediction only requires 0.08 seconds. Clearly, the computational performance of the proposed predictor meets the requirements for real-

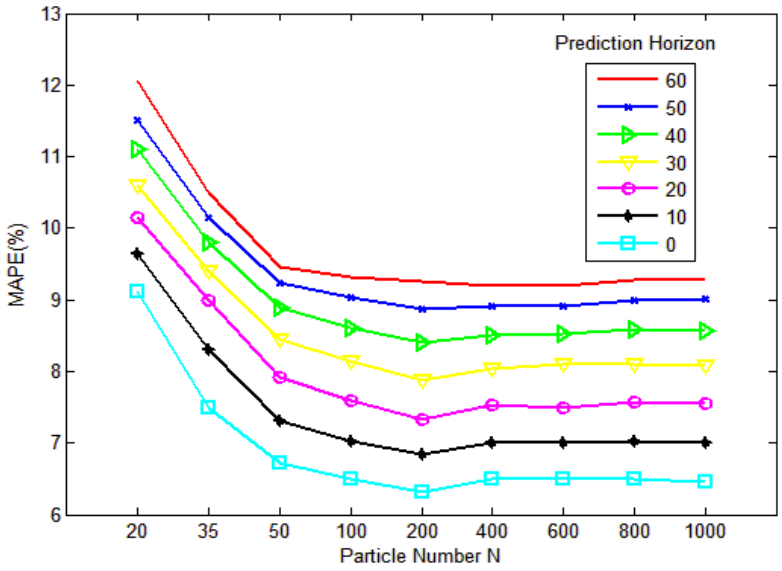
time applications. It's also worth noting that the INRIX probe data have a national coverage on most of the roadways in the United States and no off-line training is needed. Therefore, the proposed method can be implemented at any location with a similar data set.

8.4.4 Sensitivity analysis

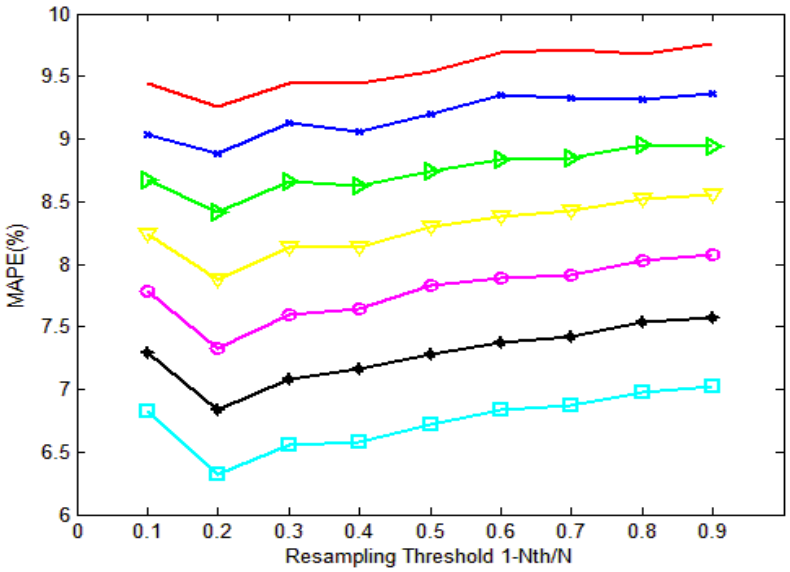
A sensitivity analysis is conducted to quantify the impact of the three parameters: L and N, and resampling ratio threshold $1\text{-}N_{th}/N$ on the prediction accuracy of the proposed method. The parameter L determines the matching window width in the particle propagation and re-selection processes, so a larger value of L results in a wider matching window and vice versa. Here, the values vary from 2 to 10 and are used to calculate the average MAPE for different prediction horizons, as presented in Figure 8.7 (a). The figure demonstrates that the minimum prediction error is obtained when L equals 6. Consequently, the best matching window is a half-hour of traffic speed data along all the freeway segments. Moreover, different particle numbers are also investigated to calculate the relative errors, as shown in Figure 8.7 (b). Generally, the prediction error decreases with larger particle numbers. However, the error reaches the minimum value when the particle number is 200, and then the prediction error increases slightly with a particle number greater than 200. Consequently, a particle number of 200 was used in this case study. Lastly, different values of resampling thresholds from 10% to 90% are tested, and the corresponding prediction errors with different prediction horizons are presented in Figure 8.7 (c). The optimum resampling threshold is reached when 20% of the total particles are resampled during each time interval. The same analysis can be conducted on different sites or roadway compositions to find the optimum model parameters.



(a)



(b)



(c)

Figure 8.7: Sensitivity analysis.

The prediction results in the case study demonstrate that the proposed NSPF method can produce highly accurate and reliable multi-step-ahead predicted travel times, based upon the comparison with the four state-of-the-art and state-of-the-practice methods under various traffic conditions.

8.5 Conclusions

The research presented in this chapter develops a new particle filter approach for the real-time application of multi-step travel time prediction using real-time and historical data set. Unlike previous studies that require an underlying physical model in modeling the state-transition function between predicted and previous travel times, the proposed particle filter uses historical trends to model the state-transition trend. A partial resampling strategy is then developed to address the degeneracy problem by replacing invalid or low weighted particles with historical data that provide similar data sequences to real-time traffic measurements. In this way, each particle can generate a travel time prediction value and a corresponding weight reflected by the similarity of the traffic patterns between each particle and the real-time traffic measurement. Consequently, the prediction can produce a distribution of travel times by aggregating all weighted particles.
The probe data on the selected freeway stretch from Richmond to Virginia Beach along I-64 and I-264 are used to investigate the performance of different prediction approaches. Considering the fact that only four months of data are used on the selected freeway stretch, the leave-one-out cross validation method is an ideal option to quantify the prediction accuracy on this limited data. The MAE and MAPE of prediction results demonstrate the proposed method produces the least deviation from ground truth travel times, compared to instantaneous travel time, two Kalman filter algorithms and *k*-NN method. Besides, the maximum daily prediction errors on June 2012 indicate the proposed NSPF method outperforms other methods by maintaining a stable performance for all test days. Moreover, the proposed approach provides good accuracy in predicting travel time reliability. Lastly, the fast computation speed and online processing ensure the proposed NSPF can be used in real-time applications.

The proposed predictor has only been used to predict freeway travel times. Nevertheless, the essential proposed particle filter method does not require certain types of data sources and can be applied for nonlinear data tracking problems in other application fields. The proposed approach is also flexible in addressing data prediction problems in other application fields and can potentially produce a comparatively high accuracy if enough historical data are provided. The implementation of the proposed predictor into arterial travel time prediction will be considered in the future. Moreover, rather than using a data sequence as the input, future research will consider the use of spatiotemporal traffic information (e.g. speed matrix) to predict travel times.

Chapter 9

9 Travel Time Prediction using Agent-based Modeling

This chapter is an edited version of: Hao Chen, Hesham A. Rakha (2014), "An Agent-based Modeling Approach to Predict Experienced Travel Times," in *93rd Transportation Research Board Annual Meeting*, Washington D.C.

The research presented in this chapter develops an agent-based modeling approach to predict experienced travel times using real-time and historical spatiotemporal traffic data. At the microscopic level, each agent represents an expert in the decision making system, which predicts the travel time for each time interval according to past experiences from a historical dataset. A set of agent interactions are developed to preserve agents that correspond to traffic patterns similar to the real-time measurements and replace invalid agents or agents with negligible weights with new agents. Consequently, the aggregation of each agent's recommendation (predicted travel time with associated weight) provides a macroscopic level of output – predicted travel time distribution. Probe vehicle data from a 95-mile freeway stretch along I-64 and I-264 was used to test the proposed method. The results show that the agent-based modeling approach produces the least prediction error compared to other state-of-practice and state-of-art methods (instantaneous travel time and k nearest neighbor), and maintains less than a 9% prediction error for trip departures up to 60 minutes into the future for a two-hour trip. Moreover, the confidence boundaries of the predicted travel times demonstrate that the proposed approach also provides high accuracy in predicting travel time reliability. Finally, no offline training is required for the proposed approach making it easily transferrable to other locations and adaptable to changes in traffic conditions.

9.1 Introduction

This chapter aims to develop a robust method for multi-step prediction of experienced travel times, yet is easily transferable to other sites and without data training. Considering the above mentioned problems, an agent-based modeling approach is developed in this chapter to predict experienced travel times using historical data. Agent-based modeling has been widely used for problems of decision making and complex social systems [114]. The advantages of this approach lie in the feature that each agent can behave as an individual expert decision system; so that every agent has the ability to analyze data input and produce its own decision output by constructing rules. More importantly, different groups of agents can cooperate to model complex social systems. In the past decades, agent-based modeling has been successfully applied to various transportation problems, because of the flexibility and computational advantages of modeling complex transportation systems [115]. However, the agent-based concept has not been used in travel time prediction. Although the direct application in predicting travel times has not been developed, several similar applications have already been attempted to deal with data prediction problems in other application fields. For instance, a group of individual cooperating agents were developed to simulate different components of the stock trading process and tested to provide accurate prediction for stock buying/selling decisions [116]. Similar approaches have been developed to predict the evolution of market share for electric vehicles [117] and the price change of the US wholesale power market [118]. In addition, a set of guidelines using agent-based models for data forecasting problems are developed in [119], and the related problems of building a predictor using an agent-based model for different categories of forecasting problems are discussed.

In this research, an agent-based modeling approach is developed to predict experienced travel times using real-time and historical spatiotemporal traffic data. At the microscopic level, each agent acts as an expert and a set of agent interactions are developed to produce a recommendation of future experienced travel times with a measurement of recommendation confidence. Consequently, the aggregation of each agent's recommendation (predicted travel time with associated weight) provides a macroscopic level of output – predicted travel time distribution. The 2012 INRIX probe data from Richmond to Virginia Beach along I-64 and I-264 are used to evaluate the performance of the proposed method. The results show that the agent-based modeling approach produces the least prediction error compared to other state-of-practice and state-of-art methods (instantaneous travel time and k nearest neighbor), and maintains less than a 9 percent prediction error for trip departures from the current time up to 60 minutes later.

The remainder of this chapter is organized as follows. The framework of the proposed agent-based modeling method is provided together with descriptions of the microscopic agent interaction rules. This is followed by an implementation to a selected test site and the proposed approach is compared with two predictors to estimate experienced travel times considering different prediction horizons (0~60 minutes). The last section includes the summary conclusions of the proposed method and recommendations for future research.

9.2 Agent-based model

The concept of correlating real-time and historical traffic measurement data by agent based model to predict travel time and the details of agent's interaction are described in this section.

For this research, we assume that the traffic speed data for each time interval keeps updating along all the roadway segments from trip origin to destination. In this way, the daily traffic measurement data can be represented as a matrix, in which each cell is an average speed for the corresponding time interval and roadway segment. Here, different colors are used to describe the speed value. Specifically, the dark blue denotes free flow speed and the bright red corresponds to congestion. Consequently, the traffic data matrix can be demonstrated as a colorful map. At the same time, the experienced travel time can be computed by providing the temporal-spatial traffic map [120]. Therefore, a traffic speed map and an experienced travel time curve are included for each day.

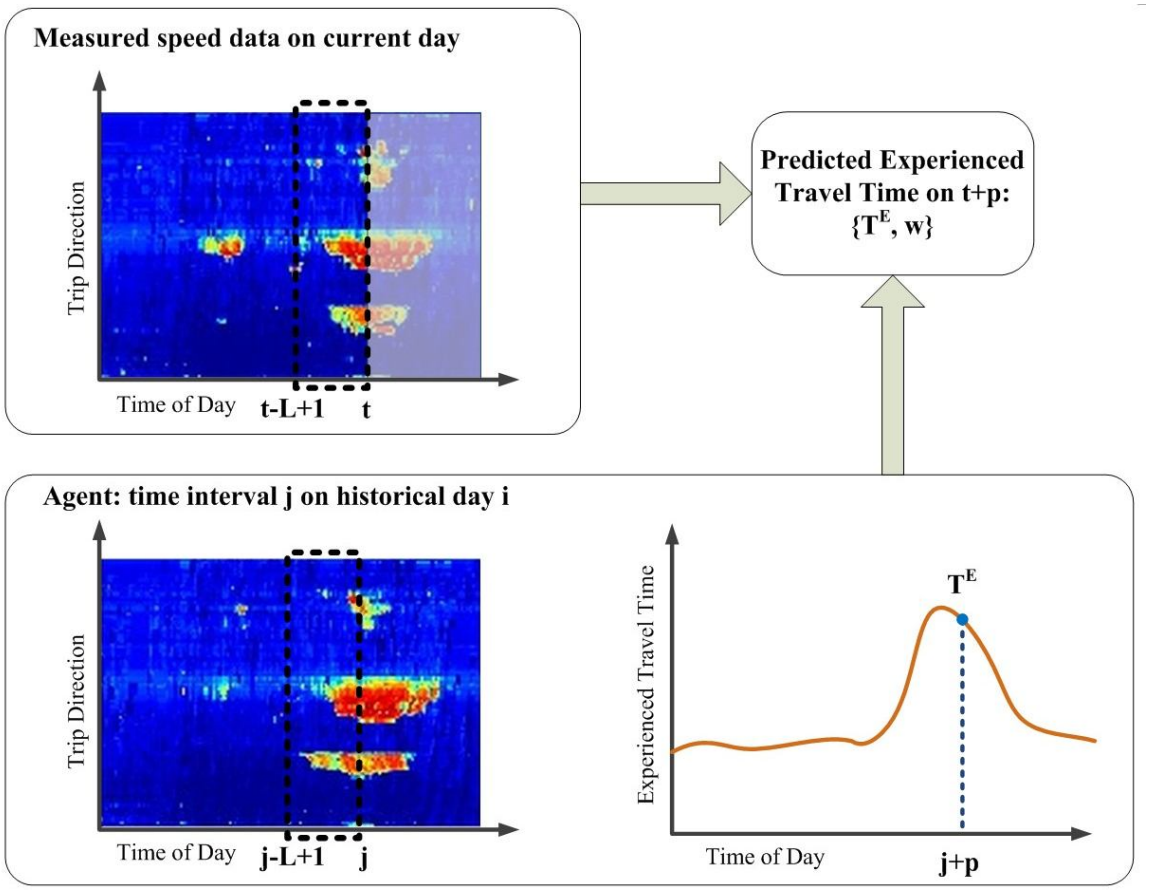


Figure 9.1: The illustration of agent-based modeling approach.

An illustration of the agent based modeling approach is demonstrated on Figure 9.1. Each agent corresponds to a specific time interval on a historical day. In this example, *i* and *j*

are the index of day and time interval of the sample agent. Assume the current traffic pattern for the testing day is denoted by the speed matrix from time *t-L+1* to *t* across all the segments, thus the agent is used to provide a prediction of experienced travel time on *t+p*. The prediction result includes a value of travel time T^E and a corresponding weight value *w*. The former value is obtained by finding the experienced travel time on time interval *j+p* on historical day *i*. And the latter value is calculated by comparing the dissimilarity between two matrixes from current day and historical day, represented by dotted rectangle windows on Figure 9.1. The details of the framework of proposed agent based model approach and the interactions between agents are described as below.

9.2.1 The framework of agent-based model

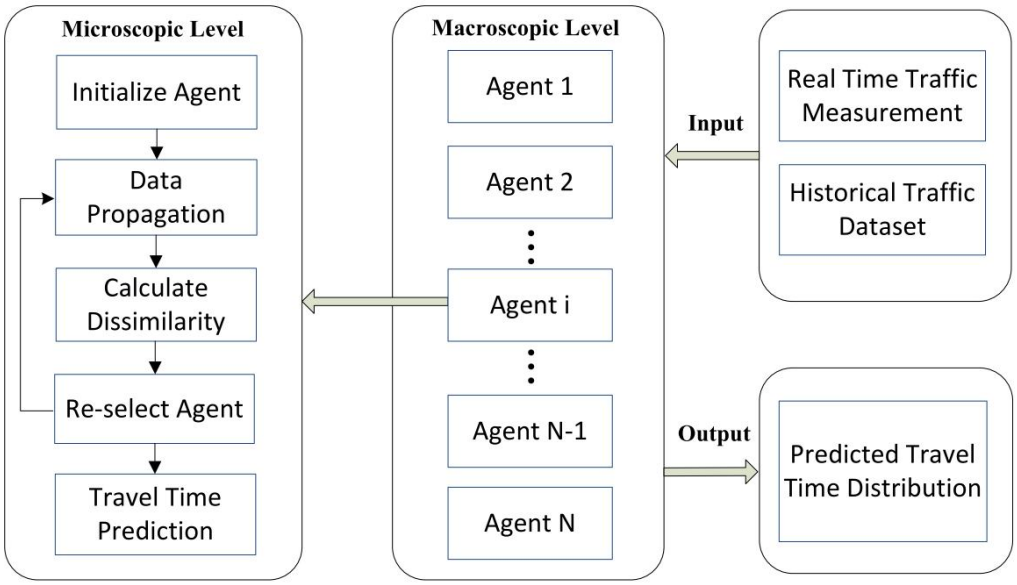


Figure 9.2: The framework of agent-based model.

For a traditional expert decision-making system, each expert makes a recommendation based on their own experience of the target problem and the aggregation of all recommendations is used to derive the final decision [121]. The proposed agent-based model adopts the same logic in order to predict travel times using real-time and historical traffic data. Each agent represents an expert, who is responsible for providing travel time predictions for each time interval. Eventually the weighted average travel time from all the agents represents the predicted travel time. The framework of the proposed agent-based model is presented in Figure 9.2. At the microscopic level, each agent interacts individually according to the real-time and historical traffic status. Different interaction rules are constructed in order to simulate the process of choosing and updating individual experts according to their performance (similarity to real-time traffic information) for each time interval. The aggregation of each agent's recommendation (predicted travel time with associated weight) provides a macroscopic level of output – the predicted travel time distribution. The definitions of variables and the details of agent interaction rules are described in the next section.

9.2.2 Agent interaction rules

Assume the current time is t and the total number of segments is N_{seg}, the real-time measurement data $u_{N_{seg}\times L}$ is the speed matrix from short past t-L+1 to t along all the freeway segments. Here, the speed matrix is denoted by the tail time as variable z_t, which also represents the real-time traffic status and keeps updating on every time interval. The purpose of this paper is to utilize the real-time traffic status and historical data to conduct a data mining process to predict the experienced travel time T^E_{t+p} for departures at time $t+p$. In this paper, each agent represents an expert who can give a prediction value based on the experience of a specific historical day. Therefore, the i^{th} agent, denoted by $x_t^{(i)}$, corresponds to a day index $d_t^{(i)}$ from the historical dataset Ω and a time index $j_t^{(i)}$ on that day. Consequently, the corresponding speed matrix from time interval d_t^i-L+1 to d_t^i along the N_{seg} segments can be obtained as $\Omega(d_t^i, j_t^i)$. The difference between two speed matrices from real-time measurement z_t and historical experience $\Omega(d_t^i, j_t^i)$ corresponding to the i^{th} agent can be used to calculate the confidence level of this agent, denoted by weight $w_t^{(i)}$. In addition, the experienced travel time can be calculated using the historical dataset, so the i^{th} agent can produce a recommendation of travel time $T^E(d_t^i, j_t^i+p)$ for departures at time j_t^i+p on historical day d_t^i. Finally, the prediction output is denoted by the integration of each agent's recommendation with the corresponding travel time $T^E(d_t^i, j_t^i+p)$ and weight $w_t^{(i)}$.

Agent Initialization

Initially each agent is given an initial correspondence on the historical dataset. The index of day $d_0^{(i)}$ is randomly selected from the historical dataset Ω (total D days) and then the index of the time interval $j_0^{(i)}$ is randomly selected for that day. Consequently, the initial agent set x_0 can be represented as

$$x_0 : \left\{ x_0^{(i)} \mid x_0^{(i)} = \Omega\left(d_0^{(i)}, j_0^{(i)}\right), i \in [1:N] \right\} \tag{9.1}$$

Data Propagation

In order to match with the new incoming measurement data, the corresponding speed matrix for each agent needs to propagate along the temporal dimension. This process is conducted by keeping the same day index and increasing the time index by an additional time interval as

$$d_t^{(i)} = d_{t-1}^{(i)},\ j_t^{(i)} = j_{t-1}^{(i)} + 1,\ i \in [1:N] \tag{9.2}$$

In the proposed algorithm, each daily traffic datum is considered as a separate data set from the adjacent days, even though the ending of one day is followed by the beginning of the next day. The reason to use each daily traffic data separately instead of continuously lies on two considerations. Firstly, the adjacent day's traffic data may not available on the historical dataset. Secondly, the measured traffic data may not have the full coverage of 24 hours. For instance, it's possible that the traffic data is only measured during day time or peak hours. Therefore, a process to identify the valid agent is

developed to examine if the data propagation reaches the data boundary of the same day. Here, the last time interval of the historical traffic speed matrix on day $d_t^{(i)}$ is denoted by $H_{d_t^{(i)}}$. Considering the prediction horizon p, the collection of the valid agent is identified as

$$\Psi_t = \left\{ i \mid j_t^{(i)} \le H_{d_t^{(i)}} - p, i \in [1:N] \right\} \tag{9.3}$$

Calculating Dissimilarity Measure

This process computes the weight of each valid agent and then identifies the top N_{th} number of agents associated with the large weight values. The average absolute error between the speed matrices for current and historical times is computed using Equation (9.4) to compute a dissimilarity measure $s_t^{(i)}$ between the current traffic status z_t and each valid agent $x_t^{(i)}$. A small dissimilarity measure indicates that the data matrices are more similar to each other. Here, a likelihood function which follows a Gaussian distribution $N(\mu,\sigma^2)$ is used to transfer the value of dissimilarity into weight $w_t^{(i)}$ as shown in Equation (9.5).

$$s_t^{(i)} = \left| z_t - x_t^{(i)} \right| \Big/ \left(L \times N_{seg} \right), \; i \in \Psi_t \tag{9.4}$$

$$w_t^{(i)} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{\left(s_t^{(i)} - \mu \right)}{2\sigma^2} \right\}, \; i \in \Psi_t \tag{9.5}$$

Thereafter, the value of the weight for each valid agent is sorted in descending order and the top N_{th} number of agents with large weight values are preserved to use in the next iteration. This process is described in Equation (9.6), in which the index of preserved agents is denoted by j and $j = 1{:}N_{th}$. In this way, the agents are divided into two groups. The first group includes the agents with large weights. The remainder agents represent the invalid agents that cannot provide prediction values (exceed data boundary) or the agents with negligible weights. This second group of agents will be re-selected in the next process so that new agents with similar traffic status to the current time interval are selected.

$$x_t^{(j)} = x_t^{(i)}, w_t^{(j)} = w_t^{(i)}, \; when \;\; i = \underset{i \in \Psi_t}{\arg\max}\, w_t^{(i)}, \Psi_t = \Psi_t - \{i\} \tag{9.6}$$

Re-selecting Agents

Considering the top N_{th} number of valid agents with large weights are maintained but the rest $N\text{-}N_{th}$ number of agents are disregarded. A re-selection algorithm is developed here to fill the gap with agents associated with similar traffic patterns to the current traffic speed matrix. Here, each historical day can be selected to represent the new agent. Consequently, the probability to select each historical should be calculated. Firstly, the index of time interval which corresponds to the traffic speed matrix with minimum dissimilarity to the current traffic status is computed as σ^n_t for each historical day n by Equation (9.7). Thereafter, the dissimilarity between current traffic status z_t and the

historical speed matrix $\Omega(n, \sigma^n_t)$ is calculated and then the same likelihood function as Equation (9.5) is used to obtain the selection probability λ^n_t of historical day n using Equation (9.8).

$$\sigma_t^n = \underset{k\in[L,H_n-p]}{\arg\min}\left(z_t - \Omega\left(n,k\right)\right),\ n\in\left[1:D\right] \tag{9.7}$$

$$\lambda_t^n = p_{e_t}\left(z_t - \Omega\left(n,\sigma_t^n\right)\right),\ n\in\left[1:D\right] \tag{9.8}$$

After the above calculation, the remainder N-N_{th} number of agents (the index is denoted by $j = N_{th}$+1:N) can be re-selected according to the probability of $\lambda^n_{t,}$, which represents the re-select probability of historical day n under the condition of current time t. The corresponding traffic speed matrix and weight can be located according to Equation (9.9).

$$\begin{aligned} d_t^{(j)} &= \left[\, n \middle| P\left(n\middle|t\right) = \lambda_t^n \,\right],\ n\in\left[1:D\right] \\ x_t^{(j)} &= \Omega\left(d_t^{(j)}, \sigma_t^{d_t^{(j)}}\right),\ w_t^{(j)} = \lambda_t^{d_t^{(j)}} \end{aligned} \tag{9.9}$$

Travel Time Prediction

Finally, the total N number of agents are located for time interval t. Since each agent corresponds to a historical day with a certain time index, the i^{th} agent can produce a recommendation of travel time $T^E(d_t^{(i)}, j_t^{(i)}+p)$ which departures on time $j_t^{(i)}+p$ at historical day $d_t^{(i)}$. Finally, the predicted travel time reliability is denoted by the aggregation of each agent's recommendation with the corresponding travel time $T^E(d_t^i, j_t^i+p)$ and weight $w_t^{(i)}$ as Equation (9.10). And the average predicted value is calculated by the weighted average travel time by Equation (9.11).

$$T_{t+p}^E = \left\{ T^E\left(d_t^{(i)}, j_{t+p}^{(i)}\right), w_t^{(i)} \right\},\ i\in\left[1:N\right] \tag{9.10}$$

$$\overline{T_{t+p}^E} = \sum_{i=1}^{N} w_t^{(i)} \cdot T^E\left(d_t^{(i)}, j_{t+p}^{(i)}\right) \Big/ \sum_{i=1}^{N} w_t^{(i)},\ i\in\left[1:N\right] \tag{9.11}$$

9.3 Case study

9.3.1 Test environment

A freeway stretch from Richmond to Virginia Beach (95 miles long) connected by I-64 and I-264 is selected as the test site in this study. This test site usually experiences high traffic volumes and serious congestion during the summer season, since Virginia Beach is a famous resort place and the selected freeway stretch serves as the main route heading to the beaches. The evaluation of travel time prediction on the test site is conducted based on probe data from INRIX. The data provided by INRIX are mainly collected by GPS-

equipped vehicles and supplemented with traditional road sensor data, as well as mobile devices and other sources [98]. The probe data on the test site covers 96 freeway segments with a total length of 95 miles. The average segment length is 0.65 miles long, and the length of each segment is unevenly divided in the raw data from 0.1 to 6.36 miles. The location of the study site and the deployment of segments are presented as Figure 9.3. The raw data provides the average speed for each segment and is collected at one-minute intervals. In this study, the raw data are aggregated at 5-minute intervals in order to reduce the noise in calculating the travel times. The aggregated traffic data from May 16, 2012 to September 15, 2012 and the corresponding afternoon time periods between 2:00 PM and 8:00 PM for each day are considered in this study to test the algorithm, since most congested periods are observed during this time frame. For each selected day, the instantaneous travel times are calculated by the summation of travel time of each segment for every time interval. On the other hand, the experienced travel time is calculated from the spatiotemporal speed data by considering the change of segment speed over time. In other words the speed profiles are piecewise constant speed values and the trip trajectory is a combination of diagonal curves over time and space [120].

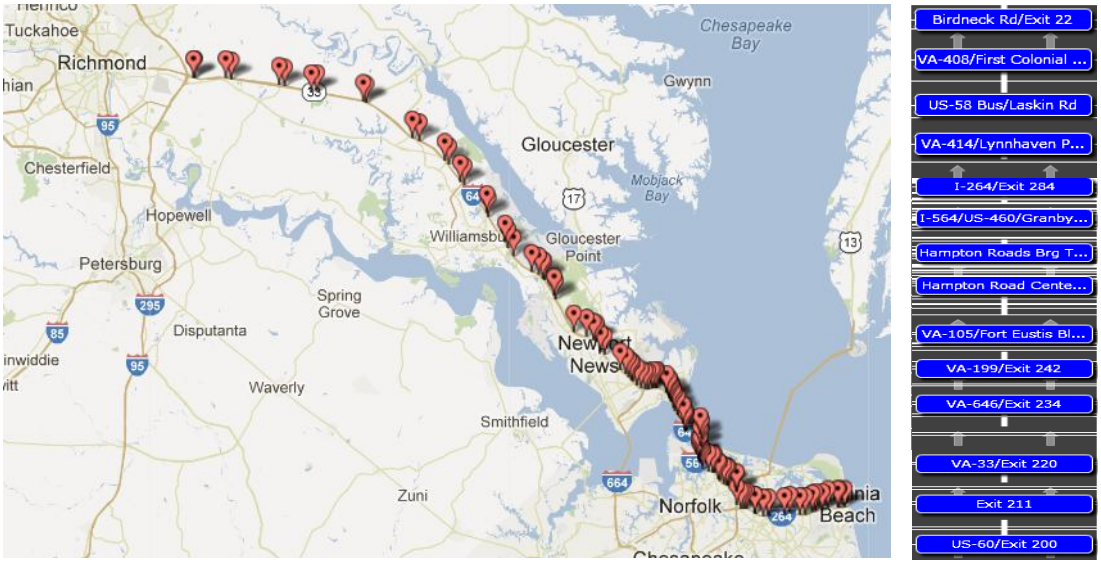


Figure 9.3: The selected freeway stretch from Richmond to Virginia Beach.

In total, the travel time data for 123 days are included in this study. The test is conducted using the leave-one-out cross-validation method in order to provide a consistent validation test. In the testing, the different methods are run on each day and the remaining days (122 days) serve as a historical dataset. Finally, the average performance over the 123 test days is used to compare the prediction accuracy of different methods. Several parameters in the proposed method are pre-defined for the test. The width of the matching window to measure the dissimilarity between historical and real-time traffic status is set at 6 time intervals (30 minutes), which entails the traffic speed matrix along all freeway segments over half an hour being used as an input variable. The agent number N is 100. Besides, 80% of the total agent number is used as agent re-select threshold, so

N_{th} is 80. Lastly, the likelihood function to calculate the agent weight factor follows a Gaussian distribution *N(0,2)*.

9.3.2 Comparison of methods and performance indices

In order to better evaluate the performance of the proposed predictor, several different methods are also tested on the same dataset. The instantaneous travel time method is the simplest alternative to predict future travel times by assuming the current traffic speed along all the segments will remain constant until the completion of the trip as Equation (9.12). This method is currently used by the Virginia Department of Transportation (VDOT) to display travel time information on variable message signs. Consequently, instantaneous travel time is considered as the state-of-practice method and used to investigate the trade off between simplicity and prediction accuracy. On the uncongested day, this approach works well since both of the current and future traffic are free flow condition. However, this approach produces unreliable prediction result during congested traffic condition.

$$T_{t+p}^{E} = T_{t}^{I} \tag{9.12}$$

The K-nearest neighbor is a widely used state-of-art method for real-time travel time prediction problems [12, 34]. In order to have a fair comparison to the proposed approach, the input variable for *k*-NN is the spatiotemporal traffic speed matrix from short past to current time along all the freeway segments, and the prediction output is experienced travel time which departures at current or future time intervals according to the prediction horizon. Consequently, if the current time is *t*, the input data for *k*-NN method is the same traffic speed matrix $u_{Nseg \times L}$, denoted by the tail time z_t. Thereafter, *m* numbers of similar data matrix with tail time $\{h_1,h_2,...,h_m\}$ can be selected as the candidates from the historical dataset Ω as given in Equation (9.11). For each candidate h_i, a weight factor w_i can be calculated using the normalized matching error. Consequently, the predicted travel time T^{E}_{t+p} can be calculated by the weighted average of the travel times from all the candidates as indicated in Equation (9.12).

$$\begin{aligned} &H_c = \{h_1, h_2, \cdots, h_m\} \\ &For\ i = 1:m \\ &\quad h_i = \arg\min_{h \in \Omega} |z_c - h|, \Omega = \Omega - h_i \\ &End\ For \end{aligned} \tag{9.13}$$

$$\begin{aligned} w_i &= |z_c - h_i| \Big/ \sum_{i=1}^{m} |z_c - h_i| \\ T_{t+p}^{E} &= \sum_{i=1}^{m} T_{h_i+p}^{E} \cdot w_i \end{aligned} \tag{9.14}$$

Different combinations of parameters were tested and the optimum parameters were selected as *L = 6* and *m = 20* to compare with other methods. It worth to point out that

the *k*-NN method is different from the proposed approach even if the candidate number of *k*-NN is equal to the number of particles. The reason lies in the fact that there is no data propagation process in the *k*-NN algorithm, and all candidates are blindly selected at each time interval with the highest similarity (lowest matching error) to the current traffic status.

To assess the different methods, the performance criteria are specified using both the absolute and relative prediction errors as demonstrated in chapter 8.

9.3.3 Test results

The prediction results by three methods are summarized in Table 9.1. The least prediction errors are highlighted on the table by using the proposed agent-based modeling (ABM) method. The relative absolute errors by three methods are presented in Figure 9.4. The figure clearly demonstrates a significant degradation in the prediction accuracy with large prediction horizon for the instantaneous methods. Alternatively, the MAPE by *k*-NN method increases from 9.21% to 13.13% (42% increase), which is higher that associated with the proposed method from 6.78% to 8.54% (25% increase). It's worth pointing out that the relative errors produced by proposed ABM approach is always below 9% for the different prediction horizons within one hour, which indicates the prediction performance of proposed method is much more reliable compared to the other methods.

Table 9.1: Prediction results by different methods.

		Prediction Horizon (min)						
		0	10	20	30	40	50	60
Instantaneous	MAE (min)	11.54	12.97	14.37	15.74	17.04	18.21	19.25
	MAPE (%)	10.67	12.09	13.44	14.79	16.08	17.23	18.25
k-NN	MAE (min)	10.43	11.13	12.07	12.74	13.58	14.21	14.97
	MAPE (%)	9.21	9.98	10.64	11.26	11.94	12.59	13.13
ABM	MAE (min)	7.70	8.05	8.37	8.58	8.84	9.09	9.40
	MAPE (%)	6.78	7.14	7.51	7.77	7.99	8.24	8.54

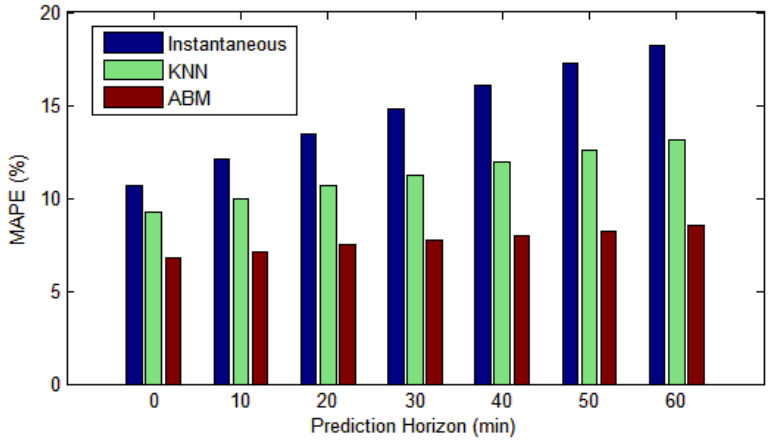


Figure 9.4: MAPE by different prediction horizons.

In order to investigate the maximum deviation between prediction results and ground truth data, the maximum MAPE produced by three methods on different testing days during June 2012 are presented on Figure 9.5. The demonstrated results are for a prediction horizon of 30 minutes. The fluctuation of the green curve by using instantaneous travel time comes from the fact that instantaneous method works well during weekend (no congestion days) and it produces large error during weekday (especially for highly congested days).The instantaneous method outperforms than other two methods during 3rd and 5th of June 2012, since there are no congestion during these days. However, the instantaneous method still produces the worst result with the maximum prediction error ranging between 13.6% and 81.5%. The maximum MAPE produced by the *k*-NN method varied between 20.1% and 41.8%, which is significantly better than the instantaneous method. More importantly, the best performance corresponds to the proposed agent-based model method with a maximum error ranging between 13.2% and 37.6%. The results clearly demonstrate that the proposed predictor still produces the best performance even under the worst case of producing maximum relative errors.

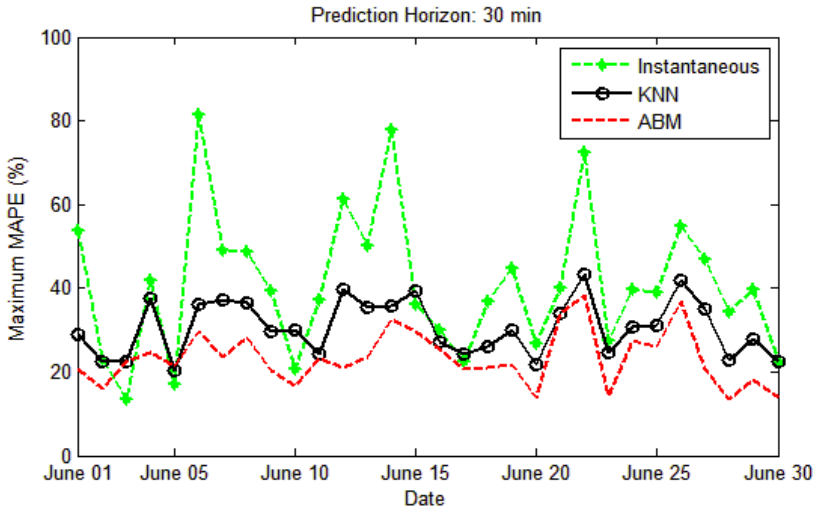
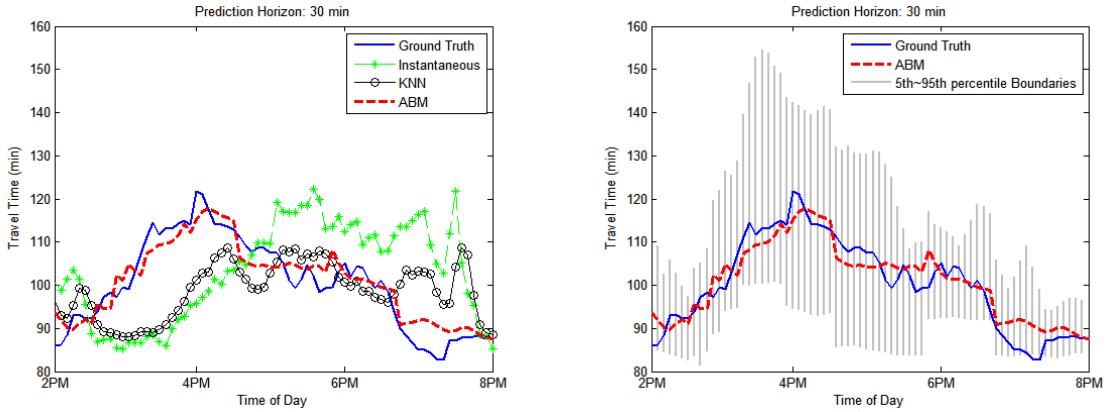


Figure 9.5: Maximum MAPE by different methods on June 2012.

The travel time curves by the three methods are compared with the ground truth data for two sample days, as illustrated in Figure 9.6. Both the instantaneous and the *k*-NN method experience some time lag relative to the ground truth data, especially during the onset and dissipation of congestion. Specifically, the instantaneous travel time highly underestimates the ground truth when congestion is forming, and overestimates the ground truth travel time when congestion is dissipating. Comparatively, the proposed method improves the prediction performance during these conditions. Moreover, the proposed method also provides travel time reliability. The 5th and 95th percentile of the predicted travel times are selected as the upper and lower boundaries in Figure 9.6. The gray shadow area between the boundaries covers most of the ground truth data temporal variation, which demonstrates that the proposed approach provides very good accuracy to predict travel time reliability.

(a)



(b)

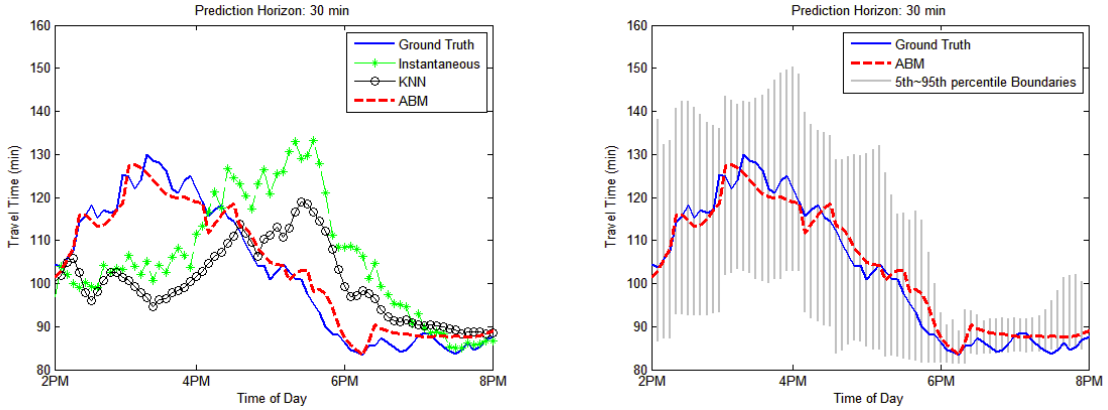
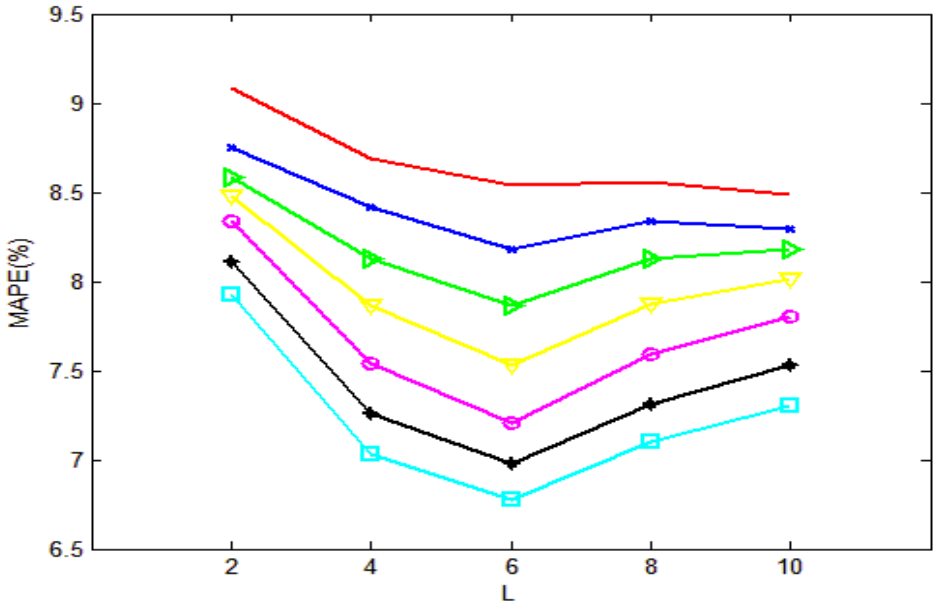


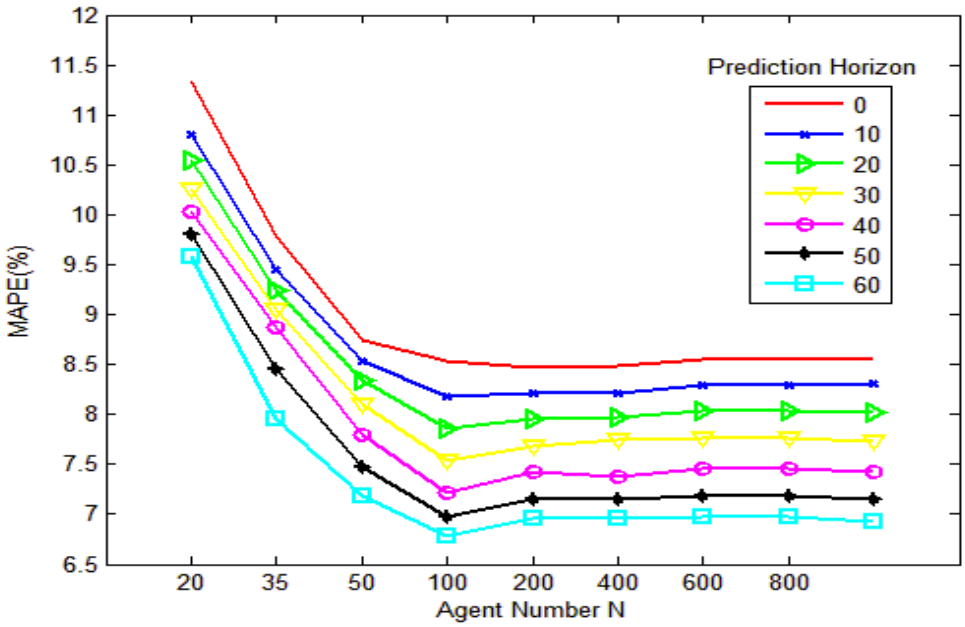
Figure 9.6: Travel time prediction results by three methods and the confidence boundaries by proposed method on (a) June 21, 2012 (Thursday); (b) June 29, 2012 (Friday).

9.3.4 Sensitivity analysis

A sensitivity analysis is conducted to investigate the impact of parameter L and agent number N on the prediction accuracy of the proposed agent-based method. The parameter L determines the matching window width in the agent propagation and re-selection processes, so a larger value of L results in a wider matching window and vice versa. the value varies from 2 to 10 are used to calculate the average MAPE of different prediction horizon as presented on Figure 9.7 (a). It demonstrates that the minimum prediction error is obtained when L equals to 6. Therefore, the best matching window is half hour's traffic speed data along all the freeway segments. On the other hand, different agent numbers are also investigated to calculate the relative errors as shown on Figure 9.7 (b). Generally, the prediction error decreases with larger agent number. However, the error reaches the minimum value when agent number is 100, and then prediction error slightly increases with agent number greater than 100. So the agent number 100 is the best value in our case study. The same analysis can be conducted on different site or roadway composition to find the optimum model parameters.



(a)



(b)

Figure 9.7: Sensitivity analysis.

9.4 Conclusion

The research presented in this chapter develops an agent-based modeling approach to predict dynamic travel times using real-time and historical spatiotemporal traffic data. At the microscopic level, each agent represents an expert in the decision making system that predicts the travel time for each time interval according to past experiences from a historical dataset. A set of agent interactions are developed to preserve agents that correspond to traffic patterns similar to the real-time measurements, and replace invalid agents or the agents with negligible weights with new agents. Consequently, the aggregation of each agent's recommendation (predicted travel time with associated weight) provides a macroscopic level of output – predicted travel time distribution. Probe vehicle data on a 95-mile freeway stretch along I-64 and I-264 in 2012 was used to test the proposed method. The results show that the agent-based modeling approach produces the least prediction error compared to other state-of-practice and state-of-art methods (instantaneous travel time and k nearest neighbour), and maintains less than a 9% prediction error for trip departures up to 60 minutes later on a two hour trip. Moreover, the confidence boundaries of the predicted travel times indicate that the proposed approach also provides high accuracy in predicting travel time reliability. In addition, no offline training is required to the proposed approach making it easily transferrable to other locations.

Although probe data are used as input to the model in this case study, the proposed method is flexible and can use any sensor data to obtain spatiotemporal traffic state maps. These data sources could be loop detector, blue tooth or cell phone measurements.

Considering the proposed method provides more than 90% accuracy in predicting travel times with departures up to 60 minutes into the future, the model can be extended to make a recommendation on the optimum departure time in addition to providing the expected travel time.

Chapter 10

10 Conclusions and Recommendations

This chapter summaries the main finding of the dissertation. Furthermore, the recommendations for various traffic prediction problems in previous chapters are provided in terms of theoretical improvements and practical applications.

10.1 Main Findings and Conclusions

This dissertation attempted to develop a comprehensive framework to solve the problem of traffic state and travel time prediction for real-time applications. Considering the state-of-art in the related research topics, the proposed algorithms in each chapter have the following findings.

A particle filter approach is developed in chapter 3 that predicts the traffic stream state using a new time series speed equation. A more realistic traffic stream model - Van Aerde, is used to derive a new Van Aerde flow continuity model via a non-conservative LWR equation. The corresponding numerical solution is obtained using a cell-centered finite volume Godunov scheme. The speed time-evolution equation is used in the proposed particle filter approach together with the available measured speed data to conduct a multi-step freeway speed prediction. Using simulated data from I-66, model test results show that the proposed approach produces the least error and accurately predicts the propagation of shockwaves.

Instead of the simplified assumption of homogenous roadway traffic, the research presented in chapter 4 improves the previous developed particle filter approach by considering ramp flows and realistic boundary conditions. In comparison to the density formulation, the proposed non-conservative speed formulation is demonstrated to be not equivalent to the LWR equation. However, the application of the formulation for the prediction of traffic states along I-66 is demonstrated to produce better predictions when compared to the LWR density formulation. The improved prediction accuracy is attributed to the direct use of speed measurements in the prediction process as opposed to estimating density from speed measurements and the associated errors with these conversions. Consequently, the use of the speed formulation appears to be better from a practical standpoint even though from a mathematical standpoint it has its flaws.

Based on the above developed particle filter framework, chapter 5 develops a dynamic travel time prediction approach based on trip trajectory construction from real-time and historical traffic measurements. Two traffic state estimation frameworks are used based on a particle filter approach. The first framework estimates traffic speed from noisy measurement data. The second framework combines the estimated speed from the first framework with historical traffic information to estimate future temporal spatial speed evolution for the duration of the trip. Trip trajectory is constructed using the speed map to predict travel times. In order to minimize the computational time, an intelligent computation process is proposed to estimate the speed map in the next time interval only when the trajectory reaches the time boundary. Simulated data on a section of I-66 is used to test the proposed method. The prediction results demonstrate that the proposed method produces a prediction error half that of the other state-of-the-art methods with a mean absolute deviation of 1.30 minutes and a mean absolute percentage error of 6 percent.

The research efforts in chapter 6 develop a travel time prediction algorithm based on computer vision and pattern recognition techniques by matching historical data with real-

time traffic conditions. Besides, a data reduction procedure is developed to transfer the raw INRIX probe data into a uniform format which can be used in the proposed travel time prediction algorithm. The probe data from I-64 and I-264 between Richmond to Virginia Beach for the past three years are collected for testing proposed algorithms. INRIX data for the selected 37-mile freeway stretch (Newport News to Virginia Beach) are used to test the proposed algorithm in case study 1 and 2. The testing results indicate that the proposed algorithm outperforms the other three methods including using instantaneous measurements, using a Kalman filter, and using the k nearest neighbor method. Moreover, the case study 3 conducted on the entire 95-mile freeway stretch from Richmond to Virginia Beach demonstrates the superiority of the proposed algorithm over the instantaneous approach that is currently used by Virginia Department of Transportation. The proposed prediction method reduces the prediction error by approximately 50 percent compared to the current instantaneous method, especially at the shoulders of the peak periods.

The framework of travel time prediction using pattern recognition is further developed in chapter 7 by a dynamic template matching algorithm to enhance the performance of matching traffic patterns from historical data to current real-time conditions. Instead of using a fixed template size, the proposed method uses a dynamic template which is updated on each time interval according to the real-time traffic condition and bottleneck shape. A fast Fourier transform based method is used in the template matching to speed up the computation process for real-time application. The selected similar traffic patterns are then used to predict experienced travel times for departures from the current time or from future time intervals. A freeway stretch from Richmond to Hampton is selected as the test site to evaluate the prediction accuracy of the proposed algorithm. The section-based INRIX data along the selected freeway is used to evaluate the performance of different predictors. The case study demonstrates that the proposed method produces much higher and very stable prediction results for prediction horizon of 0 to 30 minutes, compared to instantaneous travel time and fixed template matching method. The comparison between fixed template size and dynamic template matching methods indicates that the dynamic template improves the prediction accuracy during congestion sustaining and dissipating periods. Furthermore, the proposed dynamic template matching approach has the flexibility of using an incremental historical dataset, which is proved to further improve the prediction accuracy instead of using the constant historical dataset.

The research presented in chapter 8 develops a non-explicit state-transition model particle filter approach (NSPF) for real-time medium-term travel time prediction (2 to 4 hour predictions) using historical data. Given the challenges associated with defining the particle filter dynamic model, each particle in the proposed approach is selected from historical data to model the system evolution using historical trends instead of using a dynamic model. A partial resampling strategy is then proposed to deal with the degeneracy problem by replacing low weighted particles with historical data that are similar to the real-time field measurements. In this way, each particle can generate a travel time prediction value and a corresponding weight reflected by the likelihood between each particle and real-time field data sequence. Consequently, the algorithm

produces a travel time distribution as opposed to a single estimate. The DDPF was tested on a 95-mile freeway segment from Richmond to Virginia Beach along I-64 and I-264 using INRIX probe data from May 16 to September 15, 2012. Both of absolute and relative errors were computed considering a prediction horizon ranging from 0 to 60 minutes for an approximately 120 minute trip (i.e. predicting up to 3 hours into the future). The NSPF algorithm was then compared to actual experienced travel times and the performance of the algorithm was compared to instantaneous travel time estimates (state-of-practice), two state-of-the-art Kalman filter approaches, and a state-of-the-art *k*-NN approach. The maximum prediction error on June 2012 demonstrates that the proposed NSPF method produces the least deviation from ground truth and maintains a stable performance on different days. The confidence boundaries of the predicted travel times indicate that the proposed approach provides very good prediction accuracy of the expected and travel time reliability. Lastly, the fast computation time and online processing time ensure the method can be easily used in real-time applications.

An agent-based modeling approach is developed in chapter 9 to predict dynamic travel times using real-time and historical spatiotemporal traffic data. At the microscopic level, each agent represents an expert in the decision making system that predicts the travel time for each time interval according to past experiences from a historical dataset. A set of agent interactions are developed to preserve agents that correspond to traffic patterns similar to the real-time measurements, and replace invalid agents or the agents with negligible weights with new agents. Consequently, the aggregation of each agent's recommendation (predicted travel time with associated weight) provides a macroscopic level of output – predicted travel time distribution. Probe vehicle data on a 95-mile freeway stretch along I-64 and I-264 in 2012 was used to test the proposed method. The results show that the agent-based modeling approach produces the least prediction error compared to other state-of-practice and state-of-art methods (instantaneous travel time and k nearest neighbour), and maintains less than a 9% prediction error for trip departures up to 60 minutes later on a two hour trip. Moreover, the confidence boundaries of the predicted travel times indicate that the proposed approach also provides high accuracy in predicting travel time reliability. In addition, no offline training is required to the proposed approach making it easily transferrable to other locations.

10.2 Recommendations for Future Research

Some possible research directions and recommendations are provided in this section with respective to theoretical and/or practical improvements of the proposed algorithms.

Regarding to the proposed particle filter traffic state prediction algorithm, the distribution (reliability) of traffic state can be further predicted since each particle corresponds to a value of state variable and an associated weight. Besides, implementation of a shock fitting scheme will be investigated to overcome the fact that a non-conservative form of LWR equation is used to derive the Van-Aerde continuity model [122]. Moreover, we recommend that additional simulations be conducted to quantify the sensitivity of the prediction algorithm to different input parameters. For example, the sensitivity of the

results to the use of a non-Gaussian distribution for the prediction and measurement error terms, varying the prediction and measurement error variance, and the percentage of on-ramp and off-ramp flows.

The proposed pattern recognition algorithm employed during this study provides a framework to use spatiotemporal traffic data to predict dynamic travel times. More advanced template matching or pattern recognition techniques should be considered and tested within the proposed algorithm to identify similar traffic patterns more efficiently and accurately. Besides, the historical dataset used in the dissertation don't include weather, incident, or special event information. If such information can be used in the historical data set, the prediction accuracy is anticipated to be improved since a refined data set can be used to find similar traffic states more accurately. For instance, a subset of the database can be used for rainy conditions as opposed to using the entire data set. The development of such data will require the development of some clustering techniques to identify unique traffic state clusters. Furthermore, implementing the proposed algorithm on different corridors is also recommended to test the transferability of the approach.

Both of the NSPF and agent-based modeling approach can be further tested using different sources of data, rather than the INRIX probe data as was done in this dissertation. For instance, data measurements from loop detector, blue tooth or cell phones could be used. Besides, different data formation including sequence or matrix data can be tested by two approaches. Meanwhile, the trade-off between two approaches can be examined with respective to methodology complexity and prediction accuracy. Considering the proposed method provides more than 90% accuracy in predicting travel times with departures up to 60 minutes into the future, the model can be extended to make a recommendation on the optimum departure time in addition to providing the expected travel times.

The traffic state and travel time prediction in this dissertation is localized based on individual routes. The extension of the application to a network-level is recommended for future research. The predicted travel times on different corridors can be used to develop control strategies for route choice recommendations or area congestion reduction. A key input to these approaches is conducting research on how drivers respond to the provision of real-time information and how they switch their routes of travel depending on the information provided to them.

References

[1] David Schrank and Tim Lomax, "2007 Urban Mobility Report," Texas Transportation Institute 2007.

[2] S. Peeta and Jr J. L. Ramos, "Driver response to variable message signs-based traffic information," in *IEEE Proceedings of Intelligent Transport Systems*, 2006, pp. 2-10.

[3] Huizhao Tu, "Monitoring Travel Time Reliability on Freeways," Ph.D., Department of Transport and Planning, Technische Universiteit Delft, 2008.

[4] Pierre-Emmanuel Mazaré, Olli-Pekka Tossavainen, Alexandre Bayen, and Daniel B Work, "Trade-offs between Inductive Loops and GPS Vehicles for Travel Time Estimation: A Mobile Century Case Study," in *Transportation Research Board 91st Annual Meeting*, Washington, D.C., 2012.

[5] Hao Chen and Hesham A. Rakha, "Prediction of Dynamic Freeway Travel Times based on Vehicle Trajectory Construction," in *15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, 2012, pp. 576 - 581.

[6] Hao Chen, Hesham A. Rakha, Shereef A. Sadek, and Bryan J. Katz, "A Particle Filter Approach for Real-time Freeway Traffic State Prediction," in *Transportation Research Board 91st Annual Meeting*, Washington D.C., 2012.

[7] Xiang Fei, Chung-Cheng Lu, and Ke Liu, "A Bayesian Dynamic Linear Model Approach for Real-time Short-term Freeway Travel Time Prediction," *Transportation Research Part C: Emerging Technologies,* vol. 19, pp. 1306-1318, 2011.

[8] Jingxin Xia, Mei Chen, and Wei Huang, "A Multistep Corridor Travel-Time Prediction Method Using Presence-Type Vehicle Detector Data," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations,* vol. 15, pp. 104-113, 2011.

[9] J.W.C. van Lint, S.P. Hoogendoorn, and H.J. van Zuylen, "Accurate Freeway Travel Time Prediction with State-space Neural Networks Under Missing Data," *Transportation Research Part C: Emerging Technologies,* vol. 13, pp. 347-369, 2005.

[10] Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee, "Travel-time Prediction with Support Vector Regression," *IEEE Transactions on Intelligent Transportation Systems,* vol. 5, pp. 276-281, 2004.

[11] Jiwon Myung, Dong-Kyu Kim, Seung-Young Kho, and Chang-Ho Park, "Travel Time Prediction Using k Nearest Neighbor Method with Combined Data from Vehicle Detector System and Automatic Toll Collection System," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2256, pp. 51-59, 2011.

[12] Wenxin Qiao, Ali Haghani, and Masoud Hamedi, "Short Term Travel Time Prediction Considering the Weather Impact," in *Transportation Research Board 91st Annual Meeting*, Washington D.C., 2012.

[13] Mehmet Yildirimoglu and Nikolas Geroliminis, "Experienced Travel Time Prediction for Congested Freeways," *Transportation Research Part B: Methodological,* vol. 53, pp. 45-63, 2013.
[14] Alexander G Parlos, Omar T Rais, and Amir F Atiya, "Multi-step-ahead Prediction using Dynamic Recurrent Neural Networks," *Neural Networks,* vol. 13, pp. 765-786, 2000.
[15] R Boné and M Crucianu, "Multi-step-ahead Prediction with Neural Networks: A Review," *9emes rencontres internationales: Approches Connexionnistes en Sciences,* vol. 2, pp. 97-106, 2002.
[16] Xiaosi Zeng and Yunlong Zhang, "Development of Recurrent Neural Network Considering Temporal-Spatial Input Dynamics for Freeway Travel Time Modeling," *Computer-Aided Civil and Infrastructure Engineering,* 2013.
[17] Hao Chen, Hesham A. Rakha, and Shereef A. Sadek, "Real-time Freeway Traffic State Prediction: A Particle Filter Approach," in *14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, 2011, pp. 626-631.
[18] Yibing Wang and Markos Papageorgiou, "Real-Time Freeway Traffic State Estimation Based on Extended Kalman Filter: A General Approach," *Transportation Research Part B,* vol. 39, pp. 141-167, 2005.
[19] Yibing Wang, Markos Papageorgiou, and Albert Messmer, "Real-Time Freeway Traffic State Estimation Based on Extended Kalman Filter: Adaptive Capabilities and Real Data Testing," *Transportation Research Part A,* vol. 42, pp. 1340-1358, 2008.
[20] Daniel B. Work*., et al.*, "An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices," presented at the 47th IEEE Conference on Decision and Control, 2008.
[21] Lyudmila Mihaylova, René Boel, and Andreas Hegyi, "Freeway Traffic Estimation within Particle Filtering Framework," *Automatica,* vol. 43, pp. 290-300, 2007.
[22] Jacques Sau, Nour-Eddin El Faouzi, Anis Ben Aissa, and Olivier de Mouzon, *Particle Filter-Based Real-Time Estimation and Prediction of Traffic Conditions*. Chania, Greece, 2007.
[23] Peng Cheng, Zhijun Qiu, and Bin Ran, "Traffic Estimation Based on Particle Filtering with Stochastic State Reconstruction Using Mobile Network Data," in *Transportation Research Board Annual Meeting*, 2006.
[24] Daniel B. Work*, et al.*, "A Traffic Model for Velocity Data Assimilation," *Applied Mathematics Research Express,* pp. 1-35, 2010.
[25] Branko Ristic, and Sanjeev Arulampalam, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA, 2004.
[26] Hesham Rakha and Brent Crowther, "Comparison of Greenshields, Pipes, and Van Aerde Car-Following and Traffic Stream Models," *Transportation Research Record,* vol. 1802, 2002.
[27] Jimmy Chu. (2011). *Travel Time Messages on Dynamic Message Signs*. Available: http://ops.fhwa.dot.gov/travelinfo/dms/signs.htm
[28] Lili Du, Srinivas Peeta, and Yong Hoon Kim, "An Adaptive Information Fusion Model to Predict the Short-term Link Travel Time Distribution in Dynamic

Traffic Networks," *Transportation Research Part B: Methodological,* vol. 46, pp. 235-252, 2012.

[29] Eleni I. Vlahogianni, John C. Golias, and Matthew G. Karlaftis, "Short-term Traffic Forecasting: Overview of Objectives and Methods," *Transport Reviews,* vol. 24, pp. 533-557, 2004.

[30] Jiann-Shiou Yang, "Travel Time Prediction Using the GPS Test Vehicle and Kalman Filtering Techniques," in *Proceedings of the 2005 American Control Conference*, 2005, pp. 2128-2133.

[31] Jingxin Xia and Mei Chen, "Dynamic Freeway Corridor Travel Time Prediction Using Single Inductive Loop Detector Data," in *Transportation Research Board 88th Annual Meeting*, Washington D.C., 2009.

[32] C.P.I.J. van Hinsbergen, A. Hegyi, J.W.C. van Lint, and H.J. van Zuylen, "Bayesian Neural Networks for the Prediction of Stochastic Travel Times in Urban Networks," *IET Intelligent Transport Systems,* vol. 5, pp. 259-265, 2011.

[33] Lelitha Vanajakshi and Laurence R. Rilett, "Support Vector Machine Technique for the Short Term Prediction of Travel Time," in *IEEE Intelligent Vehicles Symposium*, Turkey, 2007, pp. 600-605.

[34] Brenda I. Bustillos and Yi-Chang Chiu, "Real-Time Freeway-Experienced Travel Time Prediction Using N-Curve and k Nearest Neighbor Methods," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2243, pp. 127-137, 2011.

[35] Menglong Yang, Yiguang Liu, and Zhisheng You, "The Reliability of Travel Time Forecasting," *IEEE Trans. Intell. Transport. Syst.,* vol. 11, pp. 162-171, 2010.

[36] Chumchoke Nanthawichit, Takashi Nakatsuji, and Hironori Suzuki, "Application of Probe-vehicle Data for Real-time Traffic-state Estimation and Short-term Travel-time Prediction on a Freeway," *Transportation Research Record: Journal of the Transportation Research Board,* pp. 49-59, 2003.

[37] Hesham Rakha and Brent Crowther, "Comparison of Greenshields, Pipes, and Van Aerde car-following and traffic stream models," *Transportation Research Record. ,* pp. 248-262 2002.

[38] Vincent Guinot, *Godunov-type Schemes: An Introduction for Engineers*: Elsevier, 2003.

[39] M.J. Lighthill and G.B. Witham, " On Kinematic Waves. I: Flood Movement in Long Rivers, II. A Theory of Traffic Flow on Long Crowded Roads," *Proceedings of the Royal Society of London,* vol. A229, pp. 281-345, 1955.

[40] P.I. Richards, "Shock waves on the highway," *Operations Research,* vol. 4, pp. 42-51, 1956.

[41] M. Van Aerde, "Single regime speed-flow-density relationship for congested and uncongested highways," *Presented at the 74th TRB Annual Conference, Washington DC, Paper No. 950802.,* 1995.

[42] Michel Van Aerde and Hesham Rakha, "Multivariate calibration of single regime speed-flow-density relationships," in *Proceedings of the 6th 1995 Vehicle Navigation and Information Systems Conference*, Seattle, WA, USA, 1995, pp. 334-341.

[43] S. K. Godunov, "A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations," *Math. Sbornik,* vol. 47, pp. 271-306, 1959.
[44] Randall J. LeVeque, *Numerical Methods for Conservation Laws*: Birkhäuser Basel, 1992.
[45] P. Le Floch, "Explicit formula for scalar non-linear conservation laws with boundary condition," *Mathematical Methods in the Applied Sciences,* vol. 10, pp. 265-287, 1988.
[46] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA, 2004.
[47] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing,* vol. 50, pp. 174-188, 2002.
[48] Sanjeev Arulampalam M., Maskell Simon, Gordon Neil, and Clapp Tim, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing,* vol. 50, pp. 174-188, 2002.
[49] M. Van Aerde and H. Rakha, *INTEGRATION © Release 2.30 for Windows: User's Guide –Volume I: Fundamental Model Features*. Blacksburg: M. Van Aerde & Assoc., Ltd., 2007.
[50] Sangjun Park, Hesham Rakha, and Feng Guo, "Calibration Issues for Multistate Model of Travel Time Reliability," *Transportation Research Record,* pp. 74-84, 2010.
[51] Daniel B. Work*, et al.*, "An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices," presented at the 47th IEEE Conference on Decision and Control, 2008.
[52] Markos Papageorgiou, Jean-Marc Blosseville, and Habib Hadj-Salem, "Modelling and real-time control of traffic flow on the southern part of Boulevard Peripherique in Paris: Part I: Modelling," *Transportation Research Part A,* vol. 24, pp. 345-359, 1990.
[53] M. Van Aerde and H. Rakha, *INTEGRATION © Release 2.30 for Windows: User's Guide –Volume II: Advanced Model Features*. Blacksburg, 2007.
[54] M. Van Aerde and S. Yagar, "Dynamic Integrated Freeway/Traffic Signal Networks: A Routeing-Based Modelling Approach," *Transportation Research,* vol. 22A(6), pp. 445-453, 1988.
[55] M. Van Aerde and S. Yagar, "Dynamic Integrated Freeway/Traffic Signal Networks: Problems and Proposed Solutions," *Transportation Research,* vol. 22A(6), pp. 435-443, 1988.
[56] H. Rakha*, et al.*, "Vehicle dynamics model for predicting maximum truck acceleration levels," *Journal of Transportation Engineering,* vol. 127, pp. 418-425, 2001.
[57] H. Rakha and I. Lucic, "Variable power vehicle dynamics model for estimating maximum truck acceleration levels," *Journal of Transportation Engineering,* vol. 128, pp. 412-419, 2002.
[58] H. Rakha*, et al.*, "Traffic signal coordination across jurisdictional boundaries: Field evaluation of efficiency, energy, environmental, and safety impacts," *Transportation Research Record,* vol. 1727, pp. 42-51, 2000.

[59] Hesham Rakha, Youn-Soo Kang, and Francois Dion, "Estimating vehicle stops at undersaturated and oversaturated fixed-time signalized intersections," *Transportation Research Record,* vol. 1776, pp. 128-137, 2001.

[60] Francois Dion, Hesham Rakha, and Youn-Soo Kang, "Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections," *Transportation Research Part B-Methodological,* vol. 38, pp. 99-122, 2004.

[61] H. Rakha, M. Van Aerde, K. Ahn, and A. A. Trani, "Requirements for evaluating traffic signal control impacts on energy and emissions based on instantaneous speed and acceleration measurements," *Transportation Research Record,* vol. 1738, pp. 56-67, 2000.

[62] Hesham Rakha, Kyoungho Ahn, and Antonio Trani, "Development of VT-Micro model for estimating hot stabilized light duty vehicle and truck emissions," *Transportation Research, Part D: Transport & Environment,* vol. 9, pp. 49-74, 2004.

[63] Kyoungho Ahn, Hesham Rakha, Antonio Trani, and Michel Van Aerde, "Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels," *Journal of Transportation Engineering,* vol. 128, pp. 182-190, 2001.

[64] Hesham Rakha and Kyoungho Ahn, "Integration modeling framework for estimating mobile source emissions," *Journal of Transportation Engineering,* vol. 130, pp. 183-193, 2004.

[65] A. Avgoustis, H . Rakha, and M. Van Aerde, "Framework for estimating network-wide safety Impacts of intelligent transportation systems," in *Intelligent Transportation Systems Safety and Security Conference* Miami, 2004.

[66] H. Rakha and B. Crowther, "Comparison and calibration of FRESIM and INTEGRATION steady-state car-following behavior," *Transportation Research,* vol. 37A, pp. 1-27, 2003.

[67] H. Rakha, "An Evaluation of the Benefits of User and System Optimised Route Guidance Strategies," M.Sc., Civil Engineering, Queen's University, Kingston, 1990.

[68] Hesham Rakha*, et al.*, "Evaluating alternative truck management strategies along interstate 81," *Transportation Research Record,* vol. 1925, pp. 76-86, 2005.

[69] H. Rakha, M. Van Aerde, L. Bloomberg, and X. Huang, "Construction and calibration of a large-scale microsimulation model of the Salt Lake area," *Transportation Research Record,* vol. 1644, pp. 93-102, 1998.

[70] B. Hellinga, H. Rakha, and M. Van Aerde, "Assessing the Potential of Using Traffic Simulation Model Results for Evaluating Automatic Incident Detection Algorithms." " in *Intelligent Transportation Systems Safety and Security Conference*, Miami, FL, USA, 2004.

[71] Hesham Rakha and Yihua Zhang, "INTEGRATION 2.30 framework for modeling lane-changing behavior in weaving sections," *Transportation Research Record.,* pp. 140-149, 2004.

[72] H. Rakha and M. Farzaneh, "Issues and solutions to macroscopic traffic dispersion modeling," *Journal of Transportation Engineering,* vol. 132, pp. 555-564, 2006.

[73] R. W. Denney, "Traffic Platoon Dispersion Modeling," *Journal of Transportation Engineering,* vol. 115, pp. 193-207, 1989.
[74] D. E. Castle and Bonniville J. W., "Platoon Dispersion over Long Road Links," *Transportation Research Record,* vol. 1021, pp. 36-44, 1985.
[75] M. Van Aerde, B.R. Hellinga, and G. MacKinnon, "QUEENSOD: A Method for Estimating Time Varying Origin-Destination Demands For Freeway Corridors/Networks," presented at the 72nd Annual Meeting of the Transportation Research Board, Washington D.C., 1993.
[76] Hao Chen, Hesham A. Rakha, and Shereef Sadek, "Real-time Freeway Traffic State Prediction: A Particle Filter Approach," in *14th International IEEE Conference on Intelligent Transportation Systems*, 2011.
[77] Hesham Rakha, Ihab El-Shawarby, and Mazen Arafeh, "Trip Travel-Time Reliability: Issues and Proposed Solutions," *Journal of Intelligent Transportation Systems,* vol. 14, pp. 232-250, 2010.
[78] Kazuhiro Otsuka, Tsutomu Horikoshi, Satoshi Suzuki, and Haruhiko Kojima, "Memory-Based Forecasting for Weather Image Patterns," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 2000, pp. 330-336.
[79] Kazuhiro Otsuka, Tsutomu Horikoshi, Haruhiko Kojima, and Satoshi Suzuki, "Image Sequence Retrieval for Forecasting Weather Radar Echo Pattern," *IEICE TRANSACTIONS on Information and Systems,* vol. E83-D, pp. 1458-1465, 2000.
[80] Dan Mikami, Kazuhiro Otsuka, and Junji Yamato, " Memory-based Particle Filter for Face Pose Tracking Robust under Complex Dynamics," in *Computer Vision and Pattern Recognition*, 2009.
[81] Anand Panangadan and Ashit Talukder, "A Variant of Particle Filtering Using Historic Datasets for Tracking Complex Geospatial Phenomena," presented at the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, California, USA, 2010.
[82] Jinsoo You and Tschangho John Kim, "Development and evaluation of a hybrid travel time forecasting model," *Transportation Research Part C: Emerging Technologies,* vol. 8, pp. 231-256, 2000.
[83] M.A. Turk and A.P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition*, 1991, pp. 586-591.
[84] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 2037-2041, 2006.
[85] J.W.C. van Lint, S.P. Hoogendoorn, and H.J. van Zuylen, "Accurate freeway traveltimeprediction with state-space neuralnetworks under missing data," *Transportation Research Part C: Emerging Technologies,* vol. 13, pp. 347-369, 2005.
[86] Arslan Basharat and Mubarak Shah, "Time Series Prediction by Chaotic Modeling of Nonlinear Dynamical Systems," in *12th International Conference on Computer Vision*, 2009, pp. 1941-1948.
[87] T. Ikeguchi and K. Aihara, "Prediction of Chaotic Time Series with Noise," *IEICE Transaction on Fundamentals,* vol. E78, pp. 1291-1298, 1995.

[88] Roberto Brunelli, *Template matching techniques in computer vision: theory and practice*: John Wiley & Sons, 2009.
[89] Xiaosi Zeng and Yunlong Zhang, "Development of Recurrent Neural Network Considering Temporal-Spatial Input Dynamics for Freeway Travel Time Modeling," *Computer-Aided Civil and Infrastructure Engineering,* vol. 28, pp. 359-371, 2013.
[90] Yunlong Zhang and Hancheng Ge, "Freeway Travel Time Prediction Using Takagi–Sugeno–Kang Fuzzy Neural Network," *Computer-Aided Civil and Infrastructure Engineering,* 2013.
[91] Mohammed Elhenawy and Hesham Rakha, "Title," unpublished|.
[92] Feng Guo, Hesham Rakha, and Sangjun Park, "Multistate Model for Travel Time Reliability," *Transportation Research Record: Journal of the Transportation Research Board* 2010.
[93] Feng Guo, Qing Li, and Hesham Rakha, "Multistate Travel Time Reliability Models with Skewed Component Distributions," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2315, pp. 47-53, 12/01/ 2012.
[94] C. Chen, A. Skabardonis, and P. Varaiya, "Systematic identification of freeway bottlenecks," *Freeway Operations and Traffic Signal Systems 2004,* pp. 46-52, 2004.
[95] H Chang, Y Lee, B Yoon, and S Baek, "Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences," *IET Intelligent Transport Systems,* vol. 6, pp. 292-305, 2012.
[96] Wenxin Qiao, Ali Haghani, and Masoud Hamedi, "A nonparametric model for short-term travel time prediction using bluetooth data," *Journal of Intelligent Transportation Systems,* vol. 17, pp. 165-175, 2013.
[97] JPl Lewis, "Fast template matching," in *Vision Interface*, 1995, pp. 15-19.
[98] INRIX. (2012). *http://www.inrix.com/trafficinformation.asp*. Available: http://www.inrix.com/trafficinformation.asp
[99] J. W. C. van Lint and N. J. van der Zijpp, "Improving A Travel Time Estimation Algorithm by Using Dual Loop Detectors," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1855, pp. 41-48, 2003.
[100] Zuduo Zheng and Dongcai Su, "Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm," *Transportation Research Part C: Emerging Technologies,* 2014.
[101] S Lim and C Lee, "Data fusion algorithm improves travel time predictions," *Intelligent Transport Systems, IET,* vol. 5, pp. 302-309, 2011.
[102] TL Pan, A Sumalee, RX Zhong, and N Indra-payoong, "Short-Term Traffic State Prediction Based on Temporal–Spatial Correlation," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 14, pp. 1242-1254, 2013.
[103] Lyudmila Mihaylova, Andreas Hegyi, Amadou Gning, and René K. Boel, "Parallelized Particle and Gaussian Sum Particle Filters for Large-Scale Freeway Traffic Systems," *IEEE Transactions on Intelligent Transportation Systems,* vol. 13, pp. 36-48, 2012.

[104] Zhirui Ye and Yunlong Zhang, "Speed estimation from single loop data using an unscented particle filter," *Computer-Aided Civil and Infrastructure Engineering,* vol. 25, pp. 494-503, 2010.
[105] Hesham Rakha, Hao Chen, Ali Haghani, and Kaveh Farokhi, "Assessment of Data Quality Needs for use in Transportation Applications," MAUTC-2011-01, 2013.
[106] Steven I-Jy Chien and Chandra Mouly Kuchipudi, "Dynamic Travel Time Prediction with Real-Time and Historic Data " *Transportation Research Record: Journal of the Transportation Research Board,* vol. 129, pp. 608-616, 2003.
[107] Mei Chen and Steven I. J. Chien, "Dynamic Freeway Travel-Time Prediction with Probe Vehicle Data: Link Based Versus Path Based," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1768, pp. 157-161, 2001.
[108] C. P. I. van Hinsbergen, J. W. C. van Lint, and H. J. van Zuylen, "Bayesian combination of travel time prediction models," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2064, pp. 73-80, 2008.
[109] Dan Mikami, Kazuhiro Otsuka, and Junji Yamato, "Memory-based Particle Filter for Tracking Objects with Large Variation in Pose and Appearance," in *The 11th European Conference on Computer Vision*, 2010, pp. 215-228.
[110] F. Gustafsson*, et al.*, "Particle filters for positioning, navigation, and tracking," *Signal Processing, IEEE Transactions on,* vol. 50, pp. 425-437, 2002.
[111] Arnaud Doucet, Nando de Freitas, and Neil Gordon, "An introduction to sequential Monte Carlo methods," *Sequential Monte Carlo methods in practice,* pp. 3-14, 2001.
[112] Miodrag Bolić, Petar M. Djurić, and Sangjin Hong, "Resampling Algorithms for Particle Filters: A Computational Complexity Perspective," *EURASIP Journal on Applied Signal Processing,* vol. 1, pp. 2267-2277, 2004.
[113] Jaimyoung Kwon, Benjamin Coifman, and Peter Bickel, "Day-to-day Travel-time Trends and Travel-time Prediction from Loop-detector Data," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1717, pp. 120-129, 2000.
[114] Tung Bui and Jintae Lee, "An Agent-based Framework for Building Decision Support Systems," *Decision Support Systems,* vol. 25, pp. 225-237, 1999.
[115] Lei Zhang and David Levinson, "Agent-based Approach to Travel Demand Modeling: Exploratory Analysis," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1898, pp. 28-36, 2004.
[116] Yuan Luo, Kecheng Liu, and Darryl N Davis, "A Multi-agent Decision Support System for Stock Trading," *IEEE Network,* vol. 16, pp. 20-27, 2002.
[117] Ehsan Shafiei*, et al.*, "An Agent-based Modeling Approach to Predict the Evolution of Market Share of Electric Vehicles: A Case Study from Iceland," *Technological Forecasting and Social Change,* vol. 79, pp. 1638-1653, 2012.
[118] Toshiyuki Sueyoshi and Gopalakrishna R Tadiparthi, "An Agent-based Decision Support System for Wholesale Electricity Market," *Decision Support Systems,* vol. 44, pp. 425-446, 2008.

[119] Samer Hassan, Javier Arroyo, José Manuel Galán, and Juan Pavón, "Asking the Oracle: Introducing Forecasting Principles into Agent-Based Modelling," *Journal of Artificial Societies and Social Simulation,* vol. 16, p. 13, 2013.
[120] J. W. C. Van Lint, N. J. Van der Zijpp, and Rotterdamse Rijweg, "An improved travel-time estimation algorithm using dual loop detectors," in *82nd Transportation Research Board Annual Meeting*, 2002.
[121] Terry Connolly and Lisa Ordóñez, *Judgment and decision making*: Wiley Online Library, 2003.
[122] Manuel D Salas, "Shock fitting method for complicated two-dimensional supersonic flows," *AIAA Journal,* vol. 14, pp. 583-588, 1976.