Active Transportation and Demand Management (ATDM) Trajectory-Level Validation

State of the Practice Review

www.its.dot.gov/index.htm

Final Report—April 2016
FHWA-JPO-14-193

U.S. Department of Transportation

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-JPO-14-193		

4. Title and Subtitle Active Transportation and Demand Management (ATDM) Trajectory-Level Validation State of the Practice Review	**5. Report Date** April 2016 **6. Performing Organization Code**
7. Author(s) Michalis Xyntarakis, Vassili Alexiadis, Robert Campbell, Erin Flanigan	**8. Performing Organization Report No.**
9. Performing Organization Name And Address Cambridge Systematics 100 Cambridge Park Drive, Suite 400 Cambridge, MA 02140	**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.** DTFH61-12-D-00042
12. Sponsoring Agency Name and Address U.S. Department of Transportation ITS Joint Program Office-HOIT 1200 New Jersey Avenue, SE Washington, DC 20590	**13. Type of Report and Period Covered** Final Report, September 2015 – April 2016 **14. Sponsoring Agency Code**

15. Supplementary Notes

FHWA GTM: James Colyar, Office of Transportation Management

16. Abstract

This state of the practice review is a literature and industry review of existing vehicle trajectory datasets, vehicle trajectory collection methods, and traffic simulation model validation techniques. This report has the following four sections and presents the current state of the practice, as well as advances and knowledge gaps in trajectory validation, a novel field:

- In the first section, existing vehicle trajectory datasets have been identified and categorized by source, purpose, and relevance to this project among other typologies.
- In the second section, existing vehicle trajectory collection methods and tools are presented as well as innovative trajectory collection methods from industries beyond transportation. Even though video detection has seen significant advances in the last 10 years and it is ready for commercial applications, project research indicates that it has a barrier of entry that cannot be met in this project given the available resources and the required data accuracy.
- In the third section, trajectory validation processes and tools are presented including computational engines, and spreadsheet-based tools developed for the purpose of validating traffic-simulation models at both aggregate and disaggregate levels.
- In the final fourth section, validation efforts, cases where simulation models, algorithms, and logic have been validated at the disaggregate level are documented.

17. Key Words	18. Distribution Statement
Traffic simulation, data collection, vehicle trajectories, validation	No restrictions.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	60	N/A

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

Table of Contents

Chapter 1 Introduction **1**
Chapter 2 Existing Vehicle Trajectory Datasets **3**
Chapter 3 Existing Trajectory Collection Methods and Tools **20**
GPS Technologies 20
Radar 23
Video Detection 23
Unmanned Aerial Vehicles 24
Conventional Traffic Monitoring 25
Chapter 4 Validation Processes and Tools **27**
Aggregate Tools 27
Disaggregate Tools 33
Chapter 5 Validation Efforts **39**
Trajectory Comparison Function for Validation 39
Validating Algorithms and Logic 39
Calibrating and Validating Car-Following Models 41
Validating Lane-Change Models at the Trajectory Level 46
Trajectory Clustering 46
Chapter 6 Summary **50**

List of Tables

Table 2-1. Trajectory datasets. 4
Table 2-2. Next Generation Simulation, US 101. 5
Table 2-3. Next Generation Simulation, I-80. 6
Table 2-4. Next Generation Simulation, Lankershim Boulevard. 7
Table 2-5. Next Generation Simulation, Peachtree Street. 8
Table 2-6. Strategic Highway Research Program 2 Naturalistic driving study. 9
Table 2-7. I-80 Prototype dataset. 10
Table 2-8. Washington County, Minnesota. 10
Table 2-9. Tucson, Arizona. 11
Table 2-10. System for Assessment of the Vehicle Motion Environment (SAVME). 11
Table 2-11. JHK and FHWA Datasets from Los Angeles and Washington, D.C. 12
Table 2-12. University of Florida. 13
Table 2-13. Ohio State. 14
Table 2-14. BOSCH Dataset. 15
Table 2-15. MOCoPo (PREDIT, French DOT). 15
Table 2-16. University of Napoli. 16
Table 2-17. Israel Driving Simulator. 17
Table 2-18. Hokkaido Test Track, Japan. 18
Table 2-19. The Netherlands. 18
Table 2-20. Intelligent Vehicle Safety Systems. 19
Table 3-1. Vehicle trajectory collection methods. 21
Table 3-2. Comparison of GPS technologies. 22
Table 4-1. Vehicle trajectory analysis system uniform database format. 34
Table 5-1. Driving regimes for the intelligent driver model. 43
Table 5-2. Impact of sampling rate on calibrated parameters. 46

List of Figures

Figure 4-1. Flow chart. SHRP L04 trajectory processor. 28
Figure 4-2. Screenshot. Strategic Highway Safety Plan L04 trajectory visualizer. 29
Figure 4-3. Charts. SHRP2 IDEA visualizations. 31
Figure 4-4. Graphs. Corridor volume plot from Dynamic Traffic Assignment Anyway. 32
Figure 4-5. Heatmap. Cambridge Systematics spreadsheet-based simulation results processor. 33
Figure 4-6. Graphs. Trajectory visualization in Vehicle Trajectory Analysis System. 35
Figure 4-7. Visualization. Trajectory explorer. 36
Figure 4-8. Charts. Cambridge Systematics computational engine visualizations. 37
Figure 4-9. Visualizations. PKU trajectory visualization system. 38
Figure 5-1. Illustration. Fréchet distance. 39
Figure 5-2. Diagram. Driver interaction diagram showing spacing versus speed. 42
Figure 5-3. Graphs. Trajectory calibration landscape. 45

Chapter 1 Introduction

Recent analysis tools research efforts, such as the Integrated Corridor Management (ICM) Analysis, Modeling, and Simulation (AMS) project, Strategic Highway Research Program (SHRP) 2 C10 project, and SHRP2 L08 project, have demonstrated that the capability for existing AMS tools to properly reflect the impacts of a proactive management approach on driver behavior through the full trip chain is limited. The trip chain here refers to the full range of decisions made by travelers, such as destination choice, time-of-day choice, mode choice, route choice, and facility/lane choice. Detailed simulation tools (meso or micro) are required when modeling the tactical portions of the trip chain (facility/lane choice), which requires modeling behaviors, such as merging, lane changing, and car following).

Current approaches to develop, calibrate, and validate simulation tools are based on time-consuming approaches that use aggregate-level field data, such as 5- or 15-minute averages of fixed-point loop detector data. Many efforts undertaken by the U.S. Department of Transportation (USDOT), such as the Basic Safety Message (BSM) Emulator project, Surrogate Safety Assessment Model (SSAM) development, and the ATDM/Dynamic Mobility Applications (DMA) AMS Testbed Development and Evaluation project, rely on trajectory data from microsimulation models. Microsimulation models, while simulating the detailed position (trajectories) of vehicles on a subsecond level, are for the most part not validated at that level. Even though the algorithms embedded in these models were developed based on vehicle trajectories, they have been validated at the aggregate performance measure level. Much more accurate AMS tools can be developed if validated based on detailed vehicle trajectory data over a variety of operational conditions (e.g., work zones, incidents, adverse weather, special events) to capture reliability measures in the base condition and with ATDM/DMA Operational Strategies utilized.

In this project, in a timeframe of 20 months, the project team will:

- Compile existing publicly available vehicle trajectory datasets.
- Collect a limited amount of new trajectory data.
- Develop a trajectory-level validation process/methodology.
- Develop a trajectory validation computational engine.
- Complete a Proof of Concept application using the computational engine to demonstrate the validation process.

This state of the practice review is a literature and industry review of existing vehicle trajectory datasets, vehicle collection methods and tools, and traffic simulation model validation methods and tools. In addition to a literature review spanning transportation, artificial intelligence (moving object detection), and computer science, we have conducted a number of interviews with the project stakeholders and embedded their suggestions and insights through this document. This report has the following four sections and presents the current state of the practice, as well as advances and knowledge gaps in trajectory validation, a novel field:

- In the first section, we have identified existing vehicle trajectory datasets; and we have categorized them by source, purpose, and relevance to this project among other typologies. The Generation SIMulation (NGSIM) datasets, in combination with the Naturalistic Driving Study (NDS) data that will become available in early 2015, contain a wealth of information that can significantly advance our knowledge of car-following and lane-changing behavior.
- In the second section, we describe existing vehicle trajectory collection methods and tools, as well as innovative trajectory collection methods from industries beyond transportation. Even though video detection has seen significant advances in the last 10 years and it is ready for commercial applications, project research indicates that it has a barrier of entry that cannot be met in this project given the available resources and the required data accuracy. Recent advances in global positioning system (GPS) technology allow us to conduct extended floating car studies with similar or better accuracy than NGSIM at a relative low cost per vehicle, allowing us to record driver behavior and lane selections over long paths from origin to destination.
- In the third section, we present validation processes and tools, computational engines, and spreadsheet-based tools developed for the purpose of validating traffic-simulation models at both aggregate and disaggregate levels. We document indepth two trajectory-based tools that supported recent SHRP 2 reliability projects, and we identify the elements of which that are relevant to this project. Furthermore, we present aggregate and disaggregate computational engines that Cambridge Systematics and others have developed to mine trajectory data at different temporal resolutions.
- In the final fourth section, we document validation efforts, cases where simulation models, algorithms, and logic have been validated at the disaggregate level. We discuss the different measures that have been used to compare trajectories in car-following trajectory-based calibration studies. We present important properties of simulation algorithms that increase model realism and the capability of AMS tools to properly reflect the impacts of a proactive management. Finally, we provide insights on the proper time-step necessary for trajectory comparisons, and we examine how the objective function used for trajectory validation is affected by small changes in car-following model parameters.

Chapter 2 Existing Vehicle Trajectory Datasets

In this section, we present research findings on available trajectory datasets in the U.S. and abroad categorized by source, location, duration, application, resolution, detection, sample size, and lane-changing information. Below, we provide a summary of the findings on the researched datasets with an overall assessment about their relevance to this study.

Based on the research conducted, the NGSIM family of datasets, despite being collected almost 10 years ago, is the most information-rich trajectory datasets available. Several stakeholders and project partners indicated that the NGSIM datasets have not been fully mined, especially for the purposes of this project. Despite of their overall high accuracy levels, as low as a few feet, many researchers have indicated in the past that NGSIM data contain outliers that limit their full utilization for some applications. However, based on prior research and for the purpose of trajectory validation, outliers in the calculation of speed and acceleration do not play as important a role as initially anticipated. For example, recent research by Dr. Treiber has demonstrated that applying smoothing techniques to trajectories to remove outliers does not improve the performance of a car-following model fitted to the data. Furthermore, careful selection of a trajectory similarity objective function further eliminates the influence of unrealistic observations. Therefore, before embarking on resource-intensive and elaborate smoothing techniques on the NGSIM dataset, the project team will investigate simple and efficient alternatives, such as reducing the sampling rate to one second, provided that resolution reduction does not limit our analysis.

The second primary dataset with a high potential for this study is the Naturalistic Driving Dataset (NDS) collected under SHRP2 in the course of several years up to 2013. NDS, for the purpose of this project, contains extended floating car data that include the position of a vehicle and the positions of close-by vehicles (up to eight) via a forward-facing radar. All data processing and preparation tasks are scheduled to be completed by the end of 2014, after which data will become available to the public. In addition to the GPS and radar information, NDS-equipped cars carry a forward- and a rear-facing camera and record speed, acceleration, and steering wheel angle from in-vehicle sensors. Virginia Tech has developed a lane-detection algorithm that measures the lateral position of a vehicle inside its own lane, and already has run the algorithm on all the video collected from the forward-facing camera. Regardless of the accuracy of the automatic-lane detection results, which are still under investigation, it is within the means of the project team to manually determine lane location for 10,000 vehicle miles of collected data, a value that is higher than the 2.5 thousand vehicle miles in the I-80 NGSIM dataset. The addition of lane information will help us calculate aggregate measures about lane changing, such as the number of mandatory and discretionary lane changes per mile under different conditions. Information on the speed and acceleration coming from the vehicle sensors can augment the GPS data to improve location accuracy.

Overall, even without lane detection, NDS contains the type of extended floating car data for which many car-following calibration studies have relied on in the past. Obtaining vehicle positions behind a vehicle prior to a lane change would be desirable, but the video from the rear-facing camera is not of high quality and may not be used to identify accurately enough the positions and speed of the following vehicles. Overall, NDS is by any means an enormous dataset containing data from 2,600 instrumented vehicles traversing 3,700 vehicle-years (1.5 years of data for each vehicle). Because of its size and coverage of different cities, NDS is an excellent source for studying intradriver variations of behavior, and how driver behavior changes by time of day, weather, incident, or other factors, as well as interdriver variations between drivers of the same or different cities. The dataset is new and it has not been used in any transportation research studies. As a result, additional resources may be needed on processing and cleaning parts of it.

In table 2-1, we present a summary of available datasets that could be useful in this study, along with our assessment of the relevance of each investigated dataset to the objectives of this project. All the datasets, except the NGSIM or NDS ones, are considered of low relevance for various reasons that are explained in detail for each dataset in tables 2-2 through 2-18. A dataset is deemed of high relevance to this study if it contains trajectories for 100 percent of the vehicles with a positional accuracy of a few feet. The NDS is considered of high relevance because it can be used to study car following in different cities and for different weather, time-of-day, and incident conditions.

Table 2-1. Trajectory datasets.

Dataset	Relevance
NGSIM Datasets	
US 101 – Los Angeles, California	High
I-80 – Bay Area, California	High
Lankershim Boulevard – Los Angeles, California	High
Peachtree Street – Atlanta, Georgia	High
Other U.S. Datasets	
SHRP2 NDS	High
I-80 Prototype Dataset	Low
Washington County, Minnesota	Low
Tucson, Arizona	Low
System for Assessment of the Vehicle Motion Environment (SAVME)	Low
JHK Dataset – Los Angeles, California, and Washington, D.C.	Low
University of Florida	Low
Ohio State	Low
International Datasets	
BOSCH Dataset – Germany	Low
MOCoPo (PREDIT, French DOT) – France	Low
University of Napoli – Italy	Low
Israel Driving Simulator – Israel	Low
Hokkaido Test Track – Japan	Low
Utrecht A2 Motorway – The Netherlands	Low

Table 2-2. Next Generation Simulation, US 101.

Source:	http://www.fhwa.dot.gov/publications/research/operations/07030/07030.pdf http://arxiv.org/pdf/0804.0108.pdf http://www.webpages.uidaho.edu/niatt/Internal/directors_notes/UIdaho%200406_NGSIM%20and%20Simulation_JColyar.pdf http://www.researchgate.net/publication/228450500_Video-Based_Vehicle_Trajectory_Data_Collection http://trid.trb.org/view.aspx?id=882483
Date:	June 15, 2005
Location:	2,100-foot section of US 101 Southbound in Los Angeles, California
Duration:	45 minutes, segmented into three 15-minute periods
Application:	Freeway weaving sections, freeway-lane selection, lane changing at a freeway merge and across a weaving section, and development of new driver behavior algorithms
Resolution:	10 hertz (Hz)
Detection:	Video camera atop a 35-story building. The NG-VIDEO software was used to produce the trajectories of all vehicles.
Sample size:	6,101 vehicles
Lane changing:	Determined by trajectory path of subject vehicle
Other traffic:	Trajectories are available for the full-traffic stream
Relevance:	High
Notes:	Additional video was collected (seven hours in the p.m. peak for the northbound direction, and five hours in each peak period for the southbound direction), but was not processed.

Table 2-3. Next Generation Simulation, I-80.

Source:	http://www.fhwa.dot.gov/publications/research/operations/06137/06137.pdf http://arxiv.org/pdf/0804.0108.pdf http://www.webpages.uidaho.edu/niatt/Internal/directors_notes/UIdaho%200406_NGSIM%20and%20Simulation_JColyar.pdf http://www.researchgate.net/publication/228450500_Video-Based_Vehicle_Trajectory_Data_Collection http://trid.trb.org/view.aspx?id=882483
Date:	April 2005
Location:	500-meter section of Interstate 80 in Emeryville, California
Duration:	Three 15-minute intervals
Application:	Freeway-Lane Selection, Cooperative Merging, Oversaturated Freeway Modeling
Resolution:	10 Hz
Detection:	Video detection
Sample size:	5,648 vehicles
Lane changing:	Determined by trajectory path of subject vehicle
Other traffic:	Trajectories are available for the full-traffic stream
Relevance:	High
Notes:	Additional video was collected (five hours in the a.m. peak and five hours in the p.m. peak) for both directions, but was not processed.

Table 2-4. Next Generation Simulation, Lankershim Boulevard.

Source:	http://www.webpages.uidaho.edu/niatt/Internal/directors_notes/UIdaho%200406_NGSIM%20and%20Simulation_JColyar.pdf http://www.researchgate.net/publication/228450500_Video-Based_Vehicle_Trajectory_Data_Collection http://www.fhwa.dot.gov/publications/research/operations/07029/07029.pdf http://trid.trb.org/view.aspx?id=882483
Date:	June 16, 2005
Location:	500-meter section of Lankershim Boulevard (three- to four-lane arterial) with four signalized intersections north of US 101 in Los Angeles, California
Duration:	30 minutes in both directions during the a.m. peak
Application:	Arterial-lane selection and driver behavior
Resolution:	10 Hz
Detection:	Five video cameras mounted on the roof of a 36-story building
Sample size:	Approximately 2,450
Lane changing:	Determined by trajectory path of subject vehicle
Other traffic:	Trajectories are available for the full-traffic stream
Relevance:	High
Notes:	Additional video was collected (five hours in the a.m. peak and five hours in the p.m. peak) for both directions, but was not processed.

Table 2-5. Next Generation Simulation, Peachtree Street.

Source:	http://www.entpe.fr/content/download/3451/21866/file/09-3831.pdf http://www.webpages.uidaho.edu/niatt/research/final_reports/klk712_n10-07.pdf http://ce571f2013.weebly.com/uploads/1/0/2/4/10249279/data-analysis-report-0400pm-0415pm.pdf http://trid.trb.org/view.aspx?id=882483
Date:	November 8, 2006
Location:	2,100-foot segment of Peachtree Street in Atlanta, Georgia, with four signalized intersections and one unsignalized intersection
Duration:	30 minutes; collected between 12:45 p.m. and 1:00 p.m. and between 4:00 p.m. and 4:15 p.m.
Application:	Arterial-lane selection
Resolution:	10 Hz
Detection:	These data were collected using eight video cameras mounted on a 30-story building, which is located at 1100 Peachtree Street NE, Atlanta, Georgia.
Sample size:	2,337 vehicles
Lane changing:	Determined by trajectory path of subject vehicle
Other traffic:	Trajectories are available for the full-traffic stream
Relevance:	High
Notes:	Additional video was collected on November 8, 2006 (6.5 hours; from 9:30 a.m. to 1:30 p.m., and from 4:00 p.m. to 6:30 p.m.), but was not processed. Video data also was collected on November 9, 2006 (8:00 a.m. to noon), but was not processed. In addition to the vehicle trajectory data, signal indication information is available for the dataset. Data can be downloaded here: http://www.webpages.uidaho.edu/ngsim/resources0001.htm.

Table 2-6. Strategic Highway Research Program 2 Naturalistic driving study.

Source:	https://insight.shrp2nds.us/home http://onlinepubs.trb.org/onlinepubs/trnews/trnews282SHRP2nds.pdf
Date:	2012 to November 2013
Location:	Bloomington, Indiana – 150 vehicles Central Pennsylvania – 150 vehicles Tampa Bay, Florida – 441 vehicles Buffalo, New York – 441 vehicles Durham, North Carolina – 300 vehicles Seattle, Washington – 409 vehicles
Duration:	Varies by trip; estimated 3,700 years of data total
Application:	Intended for traffic safety analyses; to support the development of new and improved safety countermeasures; to prevent traffic collisions and injuries.
Resolution:	One-second resolution (based on examination of sample datasets)
Detection:	Instrumented vehicles
Sample size:	1,891 instrumented vehicles, 2,600 participants
Lane changing:	Determined through manual referencing of the dashboard footage
Other traffic:	Provided by forward-facing radar for up to eight other vehicles in the vicinity of the subject vehicle. Positions are given relative to the subject vehicle. Data has gaps that must be addressed with preprocessing/imputation before using.
Relevance:	This data can be used for lane changing and car following, but would require much cleanup of the forward and lateral distance data, along with manual processing of the video data for lane-changing events, for car-following and lane-changing applicability. The datasets contain information about nearby vehicles, but do not fully describe the entire traffic stream.
Notes:	The relevant data from this study include GPS location data and roadway data that can be linked to the GPS data. Roadway data includes horizontal curvature, grade, cross slope, lane and shoulder information, speed limit signs, and intersection locations and characteristics. Forward-facing camera video is available for manual determination of lane position, along with turn signal activation information for potential aid in identifying lane-changing maneuvers. Radar data are recorded for up to eight objects at any given time in front of the vehicle (i.e., lateral and forward distance to object/lead vehicle). Data must be formally requested along with a research proposal. Access will be contingent upon a successful review outcome of the proposal and access request.

Table 2-7. I-80 Prototype dataset.

Source:	https://escholarship.org/uc/item/48s0p5gb http://arxiv.org/pdf/0804.0108.pdf
Date:	December 3, 2004
Location:	2,950-foot section of I-80 Northbound in Emeryville, California
Duration:	Two one-half-hour, peak-period-time segments
Application:	Freeway-lane selection, cooperative merging, oversaturated freeway modeling
Resolution:	15 Hz temporal resolution
Detection:	Video camera atop a building of 100 meters. A machine vision system was used to produce the trajectories of all vehicles.
Sample size:	4,733 vehicles
Lane changing:	Determined by trajectory path of subject vehicle
Other traffic:	Trajectories are available for the full-traffic stream
Relevance:	Low
Notes:	

Table 2-8. Washington County, Minnesota.

Source:	http://www.cts.umn.edu/Publications/ResearchReports/reportdetail.html?id=2362
Date:	2014
Location:	Neal Avenue and 6th Street, Washington County, Minnesota
Duration:	30 minutes
Application:	Before and after study of driver response to new signage
Resolution:	20 Hz
Detection:	Two radar sensors
Sample size:	26 vehicles
Lane changing:	Not applicable; there is only one lane per direction
Other traffic:	Not completely described; some vehicles on the roadway do not have trajectory data, so it is not possible to represent all surrounding vehicles for a particular subject vehicle.
Relevance:	The low volumes in this dataset make it unsuitable for car-following applications, and the roadway configuration of one lane in each direction make it unsuitable for lane-changing applications.
Notes:	

Table 2-9. Tucson, Arizona.

Source:	http://www.researchgate.net/publication/3427860_Methods_of_analyzing_traffic_imagery_collected_from_aerial_platforms
Date:	May 2002
Location:	4.69 km segment of Speedway Blvd between Euclid Avenue and Alvernon Way
Duration:	Approximately 10 minutes
Application:	Estimation of macroscopic traffic flow parameters (e.g., travel time)
Resolution:	Not specified
Detection:	Aerial photography (helicopter based)
Sample size:	One platoon of nine vehicles
Lane changing:	Available from the trajectory data
Other traffic:	Only positions of other vehicles in the platoon being followed
Relevance:	Low. The small sample size of this data set limits its applicability to the current project. However, it does include complete trajectory data for a platoon of vehicles.
Notes:	None.

Table 2-10. System for Assessment of the Vehicle Motion Environment (SAVME)

Source:	http://deepblue.lib.umich.edu/handle/2027.42/1324 Ervin, R. D., et al. *System for assessment of the vehicle motion environment (SAVME)*. Vol. 2. University of Michigan, Transportation Research Institute, 2000.
Date:	1996
Location:	An approximately 500-foot, five-lane arterial in Ann Arbor, Michigan
Duration:	18 hours
Application:	Design of driver assistance systems, study of driving behavior, and safety evaluations of specific roadways
Resolution:	10 Hz temporal resolution
Detection:	Digital video images from roadside towers
Sample size:	30,500 vehicles
Lane changing:	Determined by trajectory path of subject vehicle
Other traffic:	Trajectories are available for the full-traffic stream
Relevance:	Trajectory data are recorded for each vehicle that traverses the roadway. All data were collected and analyzed prior to 2001 and may have used video detection techniques that are not as accurate as those deployed in NGSIM. The project team has not been able to find a contact with knowledge of the dataset. Research conducted and published using the SAVME dataset is very limited.
Notes:	Data includes explicit lane locations, absolute positioning coordinates, and intervehicle variables (e.g., range, range rate, and angle from each host to every other vehicle). A visualization tool, the VME Animator, also is available for the data.

Table 2-11. JHK and FHWA Datasets from Los Angeles and Washington, D.C.

Source:	http://trid.trb.org/view.aspx?id=273798
Date:	Spring 1983.
Location:	Various freeway sections in the Los Angeles and Washington D.C. area, ranging between 1,200 and 3,200 feet. Generally in the p.m. peak traffic periods.
Duration:	One hour for each segment
Application:	Gather data for studying vehicular traffic flow across selected freeway section types, for enhancing freeway simulation models (including acceleration/deceleration profiles, car-following headways, lane changing and merge gap acceptance).
Resolution:	1 frame per second (fps)
Detection:	Aerial photography from circling Short Take-Off and Landing (STOL) aircraft
Sample size:	All vehicles passing through the roadway sections (typically 100,000 to 200,000 per segment and hour). Total of 14 datasets.
Lane changing:	Determined by trajectory path of subject vehicle
Other traffic:	Trajectories are available for the full-traffic stream
Relevance:	Low
Notes:	Data is available on nine-track magnetic tape from the FHWA Office of Research. An address and phone number are provided. Data include speed, position coordinates, and lane number.

Table 2-12. University of Florida.

Source:	Soria, Irene, Lily Elefteriadou, and Alexandra Kondyli. "Assessment of car-following models by driver type and under different traffic, weather conditions using data from an instrumented vehicle." *Simulation Modelling Practice and Theory* 40 (2014): 208-220.
Date:	Various dates in 2012
Location:	Jacksonville Florida, two routes in the AM and PM each of which lasting about one hour and 15 minutes to traverse
Duration:	31 subjects drove an instrumented vehicle
Application:	Simulation modeling and driver behavior
Resolution:	1 Hz
Detection:	Instrumented sport utility vehicle (SUV) vehicle with GPS and four onboard cameras capturing an fps.
Sample size:	About 31 subjects of different gender and age drove approximately one hour and 15 minutes each under congested and uncongested regions with and without rain.
Lane changing:	Lane changes were identified manually from the four onboard cameras
Other traffic:	The position of other vehicles on the same and adjacent lanes was extracted from the video cameras manually.
Relevance:	Subjects may not drive as they usually do in each of their hourly test runs. Dataset is small to be representative of the driver population, but it can and has been used in targeted studies of car following and lane changing by Dr. Elefteriadou. (Hill, Corey, Lily Elefteriadou, and Alexandra Kondyli. 2014. "Exploratory Analysis of Lane Changing on Freeways Based on Driver Behavior." Journal of Transportation Engineering. Soria, Irene, Lily Elefteriadou, and Alexandra Kondyli. 2014. "Assessment of car-following models by driver type and under different traffic, weather conditions using data from an instrumented vehicle." Simulation Modelling Practice and Theory 40: 208-220.)
Notes:	None.

Table 2-13. Ohio State.

Source:	http://www2.ece.ohio-state.edu/~coifman/documents/#DataSets Xuan, Y., Coifman, B., "Identifying Lane Change Maneuvers with Probe Vehicle Data and an Observed Asymmetry in Driver Accommodation," Journal of Transportation Engineering, ASCE, Vol 138, No 8, 2012, pp. 1051-1061.
Date:	Various dates from 2005 to 2011
Location:	Two different overlapping tour routes on I-71 in Columbus Ohio. The longer route covered 28 miles round trip, while the shorter route was roughly 14 miles long and covered the most congested portion of the longer tour.
Application:	Driver behavior
Resolution:	October 2008 to Aug 2011: LIght Detection And Ranging (LIDAR) @ 37 Hz, DGPS @ 5 Hz June 2005 to October 2008: LIDAR @ 3 Hz, DGPS @ 1 Hz or 5 Hz
Detection:	LIDAR, Differential GPS (DGPS), Radar. Loop detectors roughly one-third mile apart. Front and rear cameras.
Sample size:	In terms of vehicle miles traveled (VMT), the dataset is 40 times larger than NGSIM. The corridor is 70 times longer than I-80 NGSIM.
Lane changing:	Developed an algorithm to identify lane changes
Other traffic:	The position of other vehicles on the same and adjacent lanes can be extracted from the LIDAR and radar readings.
Relevance:	Can be used for car-following and lane-changing research involving a single vehicle.
Notes:	While the raw data have been collected over several years, Ohio State has not secured sufficient funds to develop the tools necessary to extract the vehicle trajectories from the LIDAR data. The current goal is to extract data for a single day (trajectory data from two round trips over the long tour, plus the concurrent loop detector data).

Table 2-14. BOSCH Dataset.

Source:	Database description/survey provided by Punzo
Date:	Prior to 2003
Location:	Stuttgart City and nearby freeway, Germany. Signalized intersections included
Duration:	4, 5, 7, and 20 minutes. Three runs for arterials, one run for freeway.
Application:	Car-following models.
Resolution:	10 Hz
Detection:	Instrumented vehicles with radar and accelerometer sensors for precise measurement of spacing between leader and follower vehicles.
Sample size:	Small
Lane changing:	Not applicable for the type of data collected
Other traffic:	Relative position, speed, and acceleration data are available for the subject vehicle and the lead vehicle only.
Relevance:	Limited to car-following model validation, as position information for other vehicles on the roadway is limited to only the vehicle being followed.
Notes:	Data is reported as being available online. However, provided web links do not work.

Table 2-15. MOCoPo (PREDIT, French DOT).

Source:	Rivoirard, Lucas, et al. "Using Real-World Car Traffic Dataset in Vehicular Ad Hoc Network Performance Evaluation." International Journal of Advanced Computer Science and Applications (IJACSA) 7.12 (2016): p390-398. https://hal.archives-ouvertes.fr/hal-01503210/document
Date:	Collected September 12-16, 2011.
Location:	Three RN87 freeway sections, 1.3 km long, South of Grenoble, France. Two lanes in each direction. Freeway sections include merge section and weaving section.
Duration:	One hour each, 3 sites, 5 days, for a total of 15 hours. Only 8 hours will be processed.
Application:	Lane changing in merging and weaving zones. Car-following behavior, with emphasis on distance preservation behavior of drivers inside platoons across homogeneous sections.
Resolution:	20 Hz
Detection:	Trajectory recording via aerial footage from helicopter at 500 meters, with high-precision GPS data for validation
Sample size:	Over 20,000 vehicles
Lane changing:	Determined by trajectory path of subject vehicle
Other traffic:	Trajectories will be available for the full-traffic stream
Relevance:	This trajectory dataset is anticipated but not yet available (apart from a small sample), currently making it infeasible as a data source for this project.
Notes:	This trajectory data currently is being processed.

Table 2-16. University of Napoli.

Source:	Database description/survey provided by Punzo. http://trid.trb.org/view.aspx?id=803580
Date:	2004
Location:	Unsignalized streets in Naples, Italy
Duration:	4.2 minutes for rural data. 3.3 min, 6.0 min, 5.3 min, and 5.0 min for urban data
Application:	Car-following studies
Resolution:	10 Hz
Detection:	Instrumented vehicles (DGPS), with participants instructed to follow a lead vehicle.
Sample size:	Four vehicles
Lane changing:	Not applicable for the type of data collected, based on the instructions provided to study participants
Other traffic:	Position data are available for the instrumented vehicles, enabling car-following behavior modeling.
Relevance:	Limited to car-following model validation, as position information for other vehicles on the roadway is limited to only the instrumented vehicles.
Notes:	

Table 2-17. Israel Driving Simulator.

Source:	Database description/survey provided by Punzo. http://trid.trb.org/view.aspx?id=803580 http://toledo.net.technion.ac.il/files/2012/12/TRR_PassingGapDefinition_11.pdf
Date:	Not mentioned. Inferred from publication dates to be 2008.
Location:	Simulated environment of a 7.5-kilometer (km) rural road with uncongested conditions and no intersections.
Duration:	Approximately 40 minutes to complete four scenarios
Application:	Passing behavior
Resolution:	10 Hz
Detection:	Recorded by simulator
Sample size:	100 drivers, each presented with 4 of 16 different driving scenarios
Lane changing:	In a passing scenario only; the simulated environment was a divided two-lane highway with one lane per direction.
Other traffic:	Positions of all other vehicles are precisely known.
Relevance:	Using simulated driving data to calibrate or validate simulation models is circular and inappropriate. Thus, this data is unsuitable for the current project needs.
Notes:	STSIM simulation platform.

Table 2-18. Hokkaido Test Track, Japan.

Source:	http://elib.dlr.de/21349/1/FOVUS2004_Brockfeld.pdf http://elib.dlr.de/78741/1/WSC12_paper_DLR-TS_Nippold_Wagner.pdf http://trid.trb.org/view.aspx?id=729373
Date:	October 2001
Location:	Hokkaido single-lane 3-km circular test track, Japan
Duration:	Eight experiments, each about 15-30 minutes
Application:	Car-following behavior
Resolution:	10 Hz
Detection:	DGPS and real-time kinematic measurements used to gather detailed position information.
Sample size:	Nine cars in each experiment (plus one lead vehicle driven by a researcher)
Lane changing:	Not applicable, as this was a single-lane track.
Other traffic:	Positions of all vehicles are known throughout the experiments.
Relevance:	Can be used for car-following validation and calibration, but not for lane changing due to the single-lane setup.

Table 2-19. The Netherlands.

Source:	Hoogendoorn, Serge P., et al. "Traffic data collection from aerial imagery." IFAC Proceedings Volumes 36.14 (2003): 89-94.
Date:	Not mentioned in the paper. Prior to 2003 based on publication date.
Location:	Different motorway sites near the Dutch city of Utrecht, in particular on the A2 motorway. (210 meters maximum).
Duration:	35-second clips
Application:	Driver behavior modeling, such as lane changing and car following
Resolution:	The spatial resolution was 20 cm; the temporal resolution is 8.6 Hz.
Detection:	Helicopter with digital camera
Sample size:	No additional details given.
Lane changing:	Determined by trajectory path of subject vehicle.
Other traffic:	Trajectories are available for the full-traffic stream.
Relevance:	The 35-second maximum duration limits the usefulness of these trajectory datasets for the purpose of this project, but the vehicle volumes and high degree of accuracy make this a relevant dataset if needed.
Notes:	None.

Table 2-20. Intelligent Vehicle Safety Systems.

Source:	http://onlinepubs.trb.org/onlinepubs/shrp2/SHRP2_S2-S09-RW-1.pdf https://www.cvl.isy.liu.se/en/research/projects/completed-projects/IVSS/IVSS_intersection_accidents_main_report.pdf
Date:	March 2007 to May 2008
Location:	One semirural intersection (at Savenas) east of Gothenburg, Sweden. One rural intersection (at Jung) on E20 between Gothenburg and Stockholm, Sweden.
Duration:	About 626 hours of video were recorded at Savenas between January 2006 and July 2008; 95 hours were recorded at Jung between March 2007 and May 2008. Only the period between 9:00 a.m. and 3:00 p.m. was recorded.
Application:	Intersection safety
Resolution:	20 Hz
Detection:	Ground-based overhead camera footage
Sample size:	744,000 objects were tracked at Savenas; 152,000 at Jung. About 70 percent of these tracked objects were automatically filtered out from the trajectory datasets as nonvehicle objects or unreliable objects. Only midday video was used for processing (11:00 a.m. to 2:00 p.m.).
Lane changing:	Each intersection approach had only one lane for each movement; thus, no discretionary lane changes occurred, and not enough of the upstream segments are observable to provide complete information about mandatory lane changes.
Other traffic:	Based on testing, approximately 13 percent of the traffic stream may have gone undetected.
Relevance:	Given the lack of multiple lanes for any particular direction/movement, the dataset is poorly suited for lane-changing analysis. As the focus of the study was on intersection safety, the trajectory coverage is largely focused on the intersection area itself, making it poorly suited for car following as well.
Notes:	None.

Chapter 3 Existing Trajectory Collection Methods and Tools

In this section, we have compiled information on existing and innovative vehicle collection methods and tools that have been used in the collection and processing of vehicle trajectories in transportation or other fields. Table 3-1. Vehicle Trajectory Collection Methods summarizes and categorizes the available methods in terms of accuracy and applicability specifically for this project.

Even though video detection technology has improved and there are widely used open-source libraries, such as OpenCV that perform object detection, according to the stakeholder interviews, the technology has still a very high barrier of entry for the type of application we are primarily interested, trajectory extraction of the full-traffic stream. An alternative that will provide vehicle position data of similar accuracy over a longer path requires equipping an instrumented vehicle with a high-accuracy GPS device. Such a device will allow for lane identification, information that currently is missing from most Floating Car datasets that have been used for car-following calibration and validation. In addition to the positioning information, accelerometer measurements from smartphones or car-manufactured devices can be used to augment GPS-derived speed and acceleration measurements, which typically contain considerable noise. In the following sections, we describe each data collection technology in higher detail.

GPS Technologies

The accuracy of GPS receivers has significantly improved over the years, and it is anticipated to improve even further with the addition of European and Russian satellites. A typical setup requires a GPS antenna, signal receiver/processor, and data logger. The National Marine Electronics Association (NMEA) has established a communications standard (NMEA-0183) to promote interoperability between GPS receivers and data loggers made by different manufacturers (http://www.catb.org/gpsd/NMEA.html). In some cases, these components may be available as consolidated pieces of hardware; for example, the iPhone combines all GPS components into a single unit. The survey of candidate technologies presented in table 3-2 assumes that in-vehicle hardware setup can be performed in advance before drivers begin making field runs. When evaluating the costs and benefits associated with each technology listed, it also is important to note that a standard traffic lane is approximately 12 feet (3.6 meters) wide.

Post-Processing for Improved Accuracy

Post-processing of GPS data, based on correction factors for the times and locations collected, are a common method for improving GPS accuracy after the data have been collected. However, this is not a viable strategy for the purpose of this project because consumer-level GPS devices do not collect the data needed for a post-processing workflow. There is one possible exception, which we describe below (the Piksi receiver). The Continuously Operating Reference Stations (CORS) network operated by the National Oceanic and Atmospheric Administration (NOAA) provides a free online service

(Online Positioning User Service (OPUS)) that allows users to post-process their detailed GPS data, assuming it contains the parameters needed for post-processing (http://www.ngs.noaa.gov/OPUS/).

Table 3-1. Vehicle trajectory collection methods.

Technology	Accuracy	Applicability
GPS	Greater than 30 feet	Inexpensive. Has been used in many car-following studies. Lane location cannot be determined.
GPS with Wide Area Augmentation System (WAAS)	Less than 3 feet	Inexpensive. Can be used to determine lane location using map-matching.
GPS operating on StarFire I Network	Less than 1 foot	Affordable. Can be used to determine lane location using map-matching.
Speed and Acceleration from onboard vehicle sensors	Higher than GPS, depends on vehicle sensors	Speed and acceleration can be collected by manufacturer-equipped in-vehicle sensors.
Radar	1.5 feet	Affordable and necessary for determining the distance of the leader and follower vehicles.
LIDAR	Determines 3D objects in the surrounding environment up to 150 feet away	Each device is more than $50,000. Used by Google in their autonomous vehicles. The limited range will not allow the identification of vehicles more than a second away if the vehicles are traveling with 70 mi/h.
Video Detection	Depends on resolution. Can be less than a few feet	Technology has improved and has been partially commoditized. However, it has still a high barrier of entry.
Unmanned Aerial Vehicles	Same as video detection.	Video detection is harder than recording from ground-based structures.
Video Stabilization Techniques	Depends on each particular case	When camera is placed on an aerial or moving object, stabilization, software or hardware based, is necessary.

GPS Innovations

There is a new technology just coming to the market (Piski) now that provides very accurate positioning information (to a few centimeters) using receivers that cost approximate $500 each. The setup requires one receiver to be installed in a fixed position in the field (the "base station"), and remain in range of all other mobile receivers. According to the developers, the bundled antenna of the installed base provides a line-of-sight range of about one mile, but the documentation suggests that a range of up to 20 km would be possible with the proper radio antenna. Due to the very limited range of the Piski base station, using Piski GPS devices is recommended only when investigating the movement of vehicles on one particular corridor, such as one of the already surveyed NGSIM corridors.

Table 3-2. Comparison of GPS technologies.

	Global Positioning System (GPS)	Wide Area Augmentation System (WAAS)	StarFire I Network by John Deere
Background	The core element of the majority of positioning systems today, GPS is a free publicly available resource provided by the U.S. government. The iPhone uses GPS data in combination with Wi-Fi data to evaluate its current position.	This system was originally intended to improve GPS accuracy for assisted landing and flight operations among pilots. Whereas, the NDGPS system uses land-based beacons to correct GPS coordinates, WAAS uses a network of satellites for its GPS corrections.	Used primarily for autonomous agricultural equipment applications, the StarFire I network provides very high spatial resolution. Even greater accuracy is available via the StarFire II network, which requires a subscription.
Typical spatial accuracy (under favorable conditions)	15 meters in general (http://www8.garmin.com/aboutGPS/waas.html). 5 meters for the iPhone 5S (http://racerender.com/TrackAddict/docs/Accuracy.html).	30 inches (http://www.gps.gov/technical/ps/2008-WAAS-performance-standard.pdf). Less than 2 meters in NPA mode. Less than one meter in PA mode (http://www.nstb.tc.faa.gov/REPORTS/waaspan47.pdf). 2.5 meters for portable units (iPhone attachments, Bad Elf receiver) (http://gps.dualav.com/explore-by-product/xgps150a/; http://www.emprum.com/ultimategps.php; http://bad-elf.com/pages/be-gps-2200-detail).	10 inches (http://www.gps.gov/technical/ps/2008-WAAS-performance-standard.pdf).
Typical cost per vehicle	$649 for an iPhone 5S, which includes all hardware needed.	$400 for Furuno receiver (http://www.furunousa.com/products/ProductDetail.aspx?product=BBWGPS). $259 for SI-TEX receiver (http://www.si-tex.com/downloads/retail_price_list.pdf). $360 for data logger (http://homepages.ihug.com.au/~robk/price.html). $100 for iPhone WAAS attachment, plus $649 for iPhone 5S (http://gps.dualav.com/explore-by-product/xgps150a/; http://www.emprum.com/ultimategps.php) $199 for standalone portable Bad Elf receiver and data logger (http://bad-elf.com/pages/be-gps-2200-detail).	$3,195 for receiver (http://www.deere.com). $360 for data logger (http://homepages.ihug.com.au/~robk/price.html).

Table 3-2. Comparison of GPS technologies (continuation).

	Global Positioning System (GPS)	Wide Area Augmentation System (WAAS)	StarFire I Network by John Deere
Temporal resolution	Every five seconds (typical) using an iPhone with Runmeter installed (Original research with the Runmeter app on an iPhone 4S).	Every second (Furuno receiver) (http://www.furuno.com/en/gnss/). Every one to two seconds (for data logger) (http://homepages.ihug.com.au/~robk/price.html).	Every second (http://manuals.deere.com/omview/OMPFP1384619/?tM=). Every one to two seconds (for data logger) (http://homepages.ihug.com.au/~robk/price.html).
Notes	The iPhone method does not require a separate data logger—the RunMeter app can be used to this effect with a storage rate of once every five seconds (Original research with the Runmeter app on an iPhone 4S).	The Furuno receiver supports NMEA-0183 with manual rewiring (http://www.furuno.com/en/gnss/). The SI-TEX receiver supports NMEA-0183 (http://www.si-tex.com/index.php/product-information/gps/gpk-11-detail). The Bad Elf portable receiver has a built-in data logger.	StarFire receivers support the NMEA-0183 standard, which allows them to be connected to other brands of data loggers (http://www.farmergps.com/install.htm). The previous receiver model, the StarFire300, has been discontinued but may be available used at a lower price (http://www.machinefinder.com/ww/en-U.S./machine/2568321).

Radar

Radar is an object detection system that uses electromagnetic waves to identify the range, direction, and speed of moving and fixed objects. The radio transmitter emits radio waves, which hit the nearby object and are scattered in all directions based on reflectivity. The emitted signal is partly reflected back; and although it may be weak, it can be amplified to detect the relative position and angle of the nearby object. Short- and long-range radars are part of Adaptive Cruise Control systems that have been steadily gaining popularity among drivers and car manufacturers. Long range radars have been frequently used in instrumented vehicle studies and have a detection range of 2 to 600 feet. In the NDS, forward-looking long-range radar was used to record the relative distance of up to eight nearby vehicles.

Video Detection

Traffic camera installations are ubiquitous, but using oblique camera angles is not conducive to extraction of complete trajectory information as a result of many interfering factors, such as shadows, view obstructions created by vehicles in the foreground, and difficulties in extracting precise position information in the direction of the optical axis (http://photo.stackexchange.com/questions/12434/how-do-i-calculate-the-distance-of-an-object-in-a-photo). Therefore, an approximately perpendicular view of the roadway surface and traffic stream is preferable, and can be obtained via the methods similar to those implemented in NGSIM. Furthermore, best results may be obtained on cloudy days or near midday to avoid shadow issues (e.g., no vehicle shadows being cast across several lanes).

To maintain a minimum resolution of one foot per pixel, the maximum possible coverage range for a camera recording at 1080p (1920×1080 image size) would be approximately one-third mile on the longer horizontal dimension. The height needed to accomplish this degree of roadway coverage is a function of two camera parameters: its sensor size and its focal length (http://photo.stackexchange.com/questions/12434/how-do-i-calculate-the-distance-of-an-object-in-a-photo). As an example, given the specifications of the iPhone 5, the estimated viewing height needed to capture one-third mile of roadway in the image frame is approximately 2,200 feet (http://www.gizmag.com/camera-sensor-size-guide/26684/). Stitching together video streams from multiple cameras could be used to overcome limitations in terms of maximum feasible recording distance from the roadway surface or to increase the coverage zone, but stitching comes with its own complexities due to potential focusing issues and edge distortion created by the camera lens.

The European UDRIVE project, opted for cameras in place of Radar as the technology for recording the vehicle's immediate surroundings/environment (http://www.udrive.eu/). Mobileye is a system that uses a forward-facing camera to measure distance, relative speed, and acceleration of visible vehicles with a range of 100 meters and a 38-degree field of view (http://www.mobileye.com/technology/applications/vehicle-detection/).

Unmanned Aerial Vehicles

Commercially available Unmanned Aerial Vehicles (UAV) already are available for as little as $500, or $800 with an integrated high-resolution (1920×1080) video camera that can be monitored from the ground (http://www.dji.com/product/phantom). For the model without an integrated camera, a camera can be mounted using accessories from the vendor. UAVs can ascend to 1,000 feet in as little as one minute, and have the ability to hover in a roughly stationary position automatically using onboard instrumentation (including GPS). The $500 model allows for up to 15 minutes of continuous flight time and supports a maximum range of 1,000 feet, while the $800 model allows for up to 25 minutes of flight time and supports a maximum range of 1,600 feet. Both models are designed to return automatically to their flight origins in the event of communication failure or imminent battery depletion.

The biggest hurdle to the use of UAVs for traffic surveillance in general is institutional, as the Federal Aviation Administration (FAA) regulates the use of UAVs for anything aside from recreational use (and specifies a maximum height of 400 feet for recreational purposes) (http://www.faa.gov/documentLibrary/media/Advisory_Circular/91-57.pdf). Permission must be granted from the FAA for any other UAV applications, and involves an application for Certificate of Authorization (COA), which must be sponsored by a public agency (http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/systemops/aaim/organizations/uas/coa/). Washington State DOT (WSDOT) has successfully applied for a waiver, though the approval process took six months and the COA was only granted for one year (http://www.wsdot.wa.gov/research/reports/fullreports/703.1.pdf). The FAA is expecting to publish a proposal for looser regulations on UAVs under 55 pounds (of which the UAVs discussed here qualify), but that the timeline for adoption may be longer (http://www.usforacle.com/news/view.php/845153/Library-drone-plan-hits-turbulence).

Viewing the image being recorded onboard the UAV is supported by many equipment manufacturers, so that the UAV can be appropriately positioned remotely (http://www.dji.com/product/phantom-2/spec). Image stabilization is a major problem, but may be addressed through software-based algorithms in post-processing, or may require additional stabilizing equipment to be mounted on the camera itself during recording. Such stabilization and control (pan and tilt) equipment is commercially available for use on UAVs (http://www.dji.com/product/zenmuse-z15/feature).

Instead of using UAVs, chartered helicopter rides can be used to avoid the legal barriers of using UAVs. The biggest problem, however, is image stabilization, which renders vehicle detection harder than the NGSIM effort. For the sake of completeness, we provide the following information about helicopter-mounted cameras.

For cameras placed inside the helicopter, the chartered helicopter must have suitable downward-facing windows for filming through (sometimes called a vertical reference window), which may not be available in all areas (http://www.airbushelicopters.ca/optional-equipment/vertical-reference-window/; http://www.chinookaviation.com/window.htm). For the externally mounted camera approach, a wireless camera will be needed that supports remote control and viewing of the recorded image, since adjustments will not be possible through direct contact with the camera during flight. The GoPro Hero 3+ ($400) is one such camera with these capabilities, with recording sizes of 1080×1920 (http://gopro.com/cameras/hd-hero3-black-edition%20/). However, mounting cameras to the exterior of a helicopter, such as on the landing skids, is somewhat of a legal gray area; typically, the equipment will need to be approved by the helicopter operator and installed by an authorized technician (http://helicopterforum.verticalreference.com/topic/17221-gopro-camera-on-my-skids-faa-ok/). Different operators have different policies on externally mounted camera equipment, and most do not publish policies at all (http://goldengatehelicopters.com/). Some operators offer external camera mounting equipment (including stabilization equipment) as part of their helicopter packages (http://helistream.com/services/aerial-filming-company-california/; http://goldengatehelicopters.com/). Other providers may offer helicopter charter services specifically for surveying and aerial footage applications, with rates of $600 per hour being typical (http://www.helicopter-training-tours.com/price-list.html).

Regardless of the filming approach used, image stability will be an issue, and a gyro stabilizer will likely be needed to provide suitable image quality for trajectory tracing, even with software-based stabilization in post-processing (http://www.aneclecticmind.com/2009/04/05/on-helicopters-video-and-stabilization/). Such stabilizers can cost thousands of dollars to purchase, but are sometimes available for rent as well. One product that was recommended for helicopter use on several online forums is the Ken-Lab KS-6, which sells for $2,400, but can be rented for as little as $65 per day (http://www.liteflighthelicopters.com/photography/; http://www.ken-lab.com/; and http://blueskyaerials.com/).

Conventional Traffic Monitoring

Traffic cameras and detectors are two of the most prevalent data collection systems available to transportation departments; and although they are not well-suited for trajectory tracing applications, they can still be used to obtain auxiliary data for use in this project. Specifically, detector data can be used to measure empirical headway distributions at specific points along the roadway, while video data can be used to record the number of lane-changing maneuvers in spatial bins over time. By linking these two variables, it also is possible to explore the relationship between lane-changing behavior and macroscopic traffic parameters (e.g., headway distribution, density, speed):

- **Lane Changing.** The California Department of Transportation (Caltrans) provides several traffic camera feeds at 640×480 resolution and high-frame rates (e.g., the Orange County feeds), which can be used to measure the number of lane changes that occur across different segments of freeway in the image frame (http://video.dot.ca.gov/). Software can be configured to automatically record the timing and location (pixel coordinates) of each lane-changing maneuver by clicking on each vehicle as it performs a lane-changing maneuver. A correspondence table could then be used to translate the pixel coordinates into lane number and/or freeway segment, subject to the suitability of the camera angle.
- **Headway Distributions.** The Berkeley Highway Laboratory collected and archived detailed detector data at 1/60 second resolution using custom controller software, which allows for high-resolution headway estimation (http://www.its.berkeley.edu/publications/UCB/2007/CWP/UCB-ITS-CWP-2007-2.pdf; http://www.escholarship.org/uc/item/0248v7w8). This software could be deployed on other controllers in a similar manner to obtain precise empirical distributions for headways at compatible vehicle detection stations.

Chapter 4 Validation Processes and Tools

In the transportation field, exporting vehicle trajectories from mesoscopic or microscopic models has not been used frequently, relying instead on aggregated flow and travel time measures provided by the software itself. The increasing focus on travel time reliability has opened up the possibility of processing additional measures that can only be obtained by analyzing observed or simulated vehicle trajectories. In this section, we have classified the reviewed trajectory tools as aggregate and disaggregate. Aggregate are the tools primarily concerned with the properties of an entire trajectory, such as travel time. Disaggregate are those tools that allow the user to calculate microscopic measures, such as gaps or acceleration at any given time step.

Aggregate Tools

NEXTA (SHRP 2 L04)

Dr. Xuesong Zhou has developed a large collection of open-source modeling tools, including a mesoscopic simulator, a Dynamic Traffic Assignment (DTA) model, and the NEXTA visualization platform. As part of the SHRP L04 project, Dr. Zhou developed a trajectory processor that is primarily targeted at the extraction of reliability-related measures from mesoscopic trajectories. Figure 4-1 shows the workflow between the scenario manager and the trajectory processor, including some of the reliability measures that can be extracted. Different weather and incident scenarios are developed with the scenario manager; and the resulting trajectories are post-processed by the trajectory processor to extract measures, such as the travel-time variance and buffer index. The user is able to specify a subset of origin-destination (OD) pairs for which the distributions of travel time for the observed and simulated data are visualized in the Graphical User Interface (figure 4-2).

The SHRP L04 trajectory post-processor has several components that are of interest in this project:

- A map-matching algorithm that snaps observed trajectories from TomTom to the simulation network. Data collected from GPS units can benefit from using map-matching to identify the properties of the link the vehicle is on, including speed limit or slope. Map-matching is necessary when the lane number has to be identified from the GPS data, a task that requires data to be snapped not only to the appropriate link, but also to the closest lane. Map-matching is a trivial task only when there is no error in the GPS data. A vast amount of research has been conducted to determine the best methodology to deal with statistical noise and outliers given the fact that the obvious choice of snapping a GPS point to the closest roadway feature produces discontinuous paths with many lateral and longitudinal inconsistencies. Even though map-matching is an important and well researched problem, only a few open-source tools exist and only one commercial four-step planning package has related functionality.

- A visualization engine that allows the user to select paths on a map and visualizes macroscopic properties of the paths using graphs (figure 4-2). The visualization engine requires that paths are saved in csv format to be visualized in Google Earth and in the NEXTA desktop application.

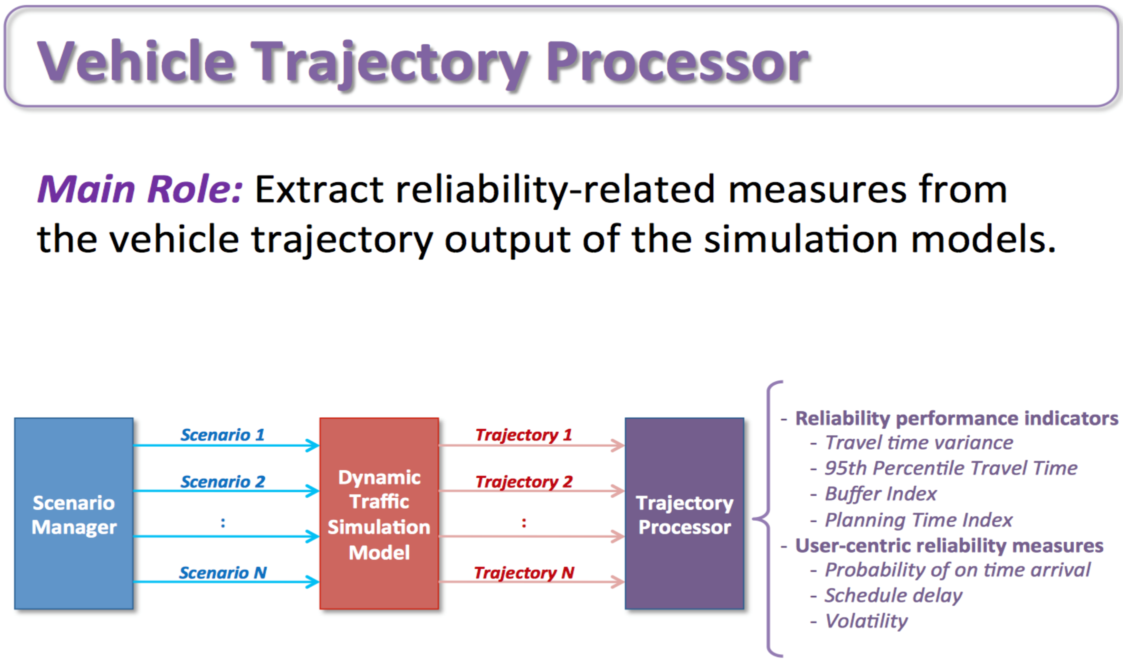

Figure 4-1. Flow chart. SHRP L04 trajectory processor.
(Source: http://onlinepubs.trb.org/onlinepubs/shrp2/RFPL38/L04webinarpresentation.pdf.)

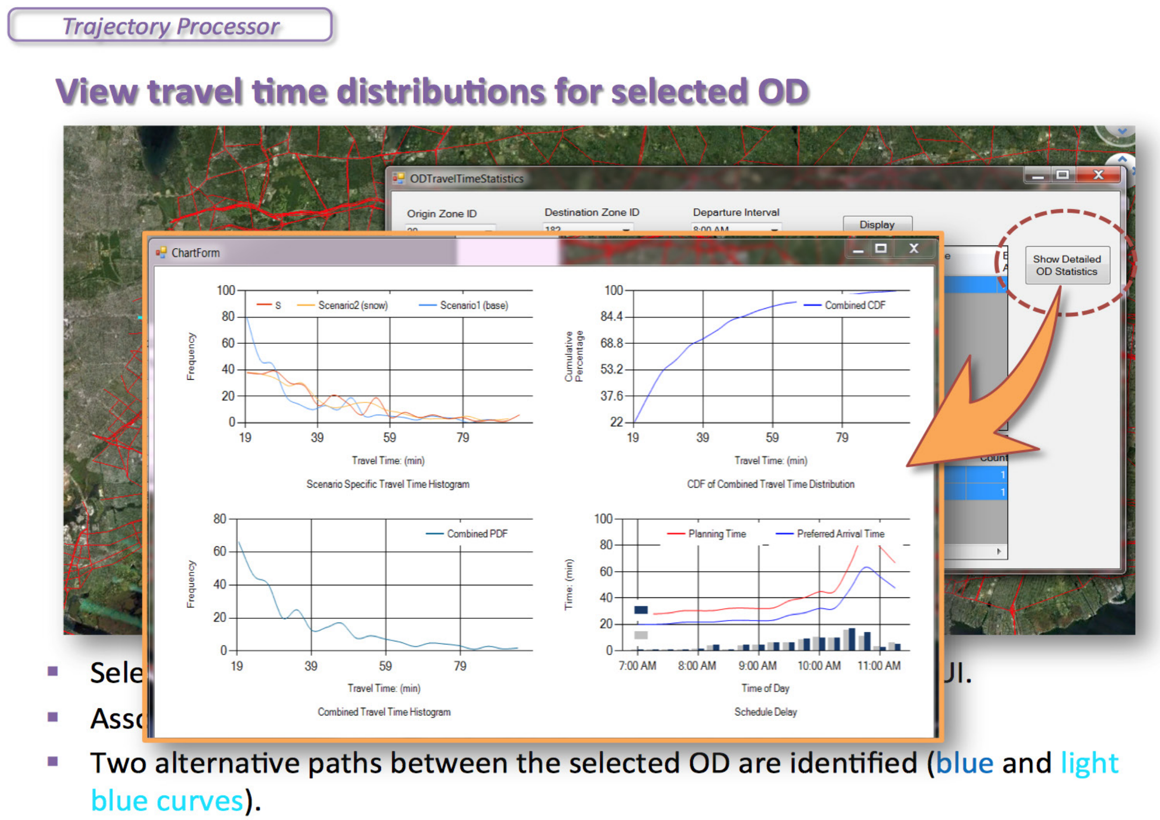

Figure 4-2. Screenshot. Strategic Highway Safety Plan L04 trajectory visualizer.
(Source: http://onlinepubs.trb.org/onlinepubs/shrp2/RFPL38/L04webinarpresentation.pdf.)

The computation engine inside the SHRP L04 processor stores and analyzes trajectories at the aggregate level because its primary objective is not the analysis of vehicle interactions, but the calculation of travel time from origin to destination. To achieve this, time-dependent link travel times are stored in multiple matrices, one matrix for each time period. To calculate travel time from origin to destination, a vehicle is traced along its path keeping a record of link arrival times. Linear interpolation is used to identify the weights to associate to link travel times of different time periods in the travel time calculations. Currently, the trajectory simulator does not allow us to calculate the location of all vehicles at each time step. Instead, vehicle locations are registered only when vehicles move from one link to the next, regardless of the time interval that has elapsed. This is a parsimonious method to store and process trajectories in a regional simulation involving hundreds of thousands or even millions of vehicles that are simulated by a mesoscopic tool. Storing trajectory information every second or subsecond would require several Terabytes even for a midsize network and for a few hours of simulation. Even though the computational engine behind the SHRP L04 project may not be directly applicable to this project, significant insights and functionalities can be copied from the open-source codebase.

SHRP Reliability IDEA Program: Urban Travel Reliability Analysis with Consumer GPS Data

To analyze route travel-time reliability, Dr. Marco Nie from Northwestern University built a similar tool with that of Dr. Zhou, but utilized a different set of technologies, including the relational database PostgreSQL. The objective of the project was to compare the reliability of the link and route travel times from ground sensors with TomTom provided data. To achieve this, Dr. Nie started with TomTom's MultiNet dataset, a collection of matrices that contains average link travel times every five minutes, the TrafficStat dataset that contains selected route travel times, and the ground sensor data. All data were mapped to the regional planning network shown in Figure 4-3. The regional planning network was used as a base for all subsequent analysis and for finding path travel times on alternative routes. The route segments TomTom uses often are referred to as the Traffic Message Channel's (TMC), and are different than the links of the regional roadway network in length and shape. To overcome this obstacle, Dr. Nie used map-matching to associate one or more TMC links with a single planning link. After doing so, his team was able to identify alternative paths from a selected set of origins and destinations and calculate their reliability. Path travel-time calculations were performed in a similar fashion with the SHRP L04 project: vehicles were traced along their paths and proper travel times were selected based on the time a vehicle arrived at each link. Paths, as it was the case with the L04 project, were stored as a sequence of visited links, as opposed to a sequence of visited points for every time interval.

The computational engine and visualizer used in the SHRP IDEA project is relevant to this project for the following reasons:

- Instead of storing segment travel times as matrices, the engine stores them as records in PostgreSQL, an industrial strength relational database. This database allows for querying and grouping the travel-time results easily using the SQL scripting language. Cambridge Systematics proposes that we use the PostgreSQL database and extend it with spatiotemporal capabilities.
- By using a custom built visualization tool developed in Visual Studio and the .NET programing environment, the Northwestern team has built a simple, powerful, and extensible user interface that can be extended to visualize other elements in addition to networks and graphs. Even though the NEXTA visualizer developed by Dr. Zhou contains more features, the simplicity of the code developed in Northwestern is particularly important for this project.

In terms of processing trajectories on a second-by-second or subsecond basis, the Northwestern tool has the same drawbacks as the L04 project. It stores trajectories as a sequence of links and records when a vehicle arrives at the beginning of a link. As such, it is not capable of querying the position of the vehicles in any particular time, unless significant calculations are performed outside the database. Some of the SHRP 2 IDEA visualizations are presented below in Figure 4-3 and include area-wide network depictions or link-specific histograms. Given that this project is mostly focused on linear corridors and less on city-wide networks, the visualizations related to link properties are relevant to the project objectives.

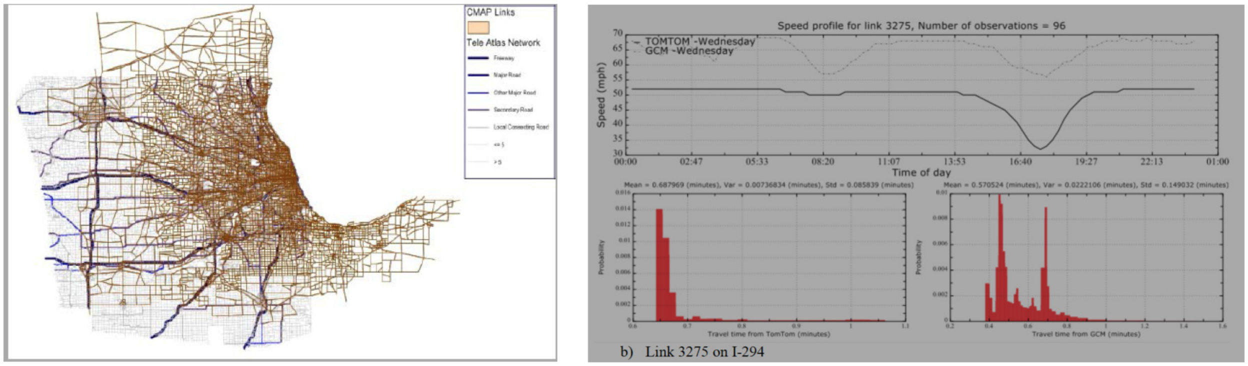

Figure 4-3. Charts. SHRP2 IDEA visualizations.
(Source: SHRP 2 Reliability IDEA L15(D).)

DTA Anyway

The San Francisco County Transportation Authority (SFCTA) has developed a large set of python classes that read, write, and modify mesoscopic DTA data. Although the current functionality does not include reading and writing mesoscopic trajectories it can be easily extended to do so. The current functionality includes:

- Reading and writing mesoscopic DTA network information including, nodes, links, movements, and signals.
- Reading mesoscopic simulation results for querying, processing, and reporting.
- Converting four-step planning networks into mesoscopic DTA networks.
- Importing transit route information from four-step planning packages or the General Transit Format Specification into a mesoscopic simulation software.
- Importing time signals stored in Excel tables into pretimed signals.
- Storing and reporting count data into a separate utility called CountDracula.
- Visualizing flows, counts, travel times, and other simulation results.

The SFCTA's code base can be found in https://code.google.com/p/dta/. The following picture shows one of the more complex visualizations that can be achieved with DTA Anyway. Specifically, it shows the simulated flows on the Geary corridor versus the reported counts at each intersection of Geary and cross-street (flow of traffic is from right to left). The bottom chart in the plot depicts flow on the corridor with a blue line, while count data are shown as black circle points. Perpendicular dotted lines signify an intersection with a cross-street whose name is shown on the bottom chart. The top chart shows the movement flows that enter the corridor, while the middle plot shows the movement flows (and counts) that leave the corridor. This plot is an example of how several aggregate link and movement quantities can be shown simultaneously on the same complex chart to provide valuable insight on corridor operations at the aggregate level.

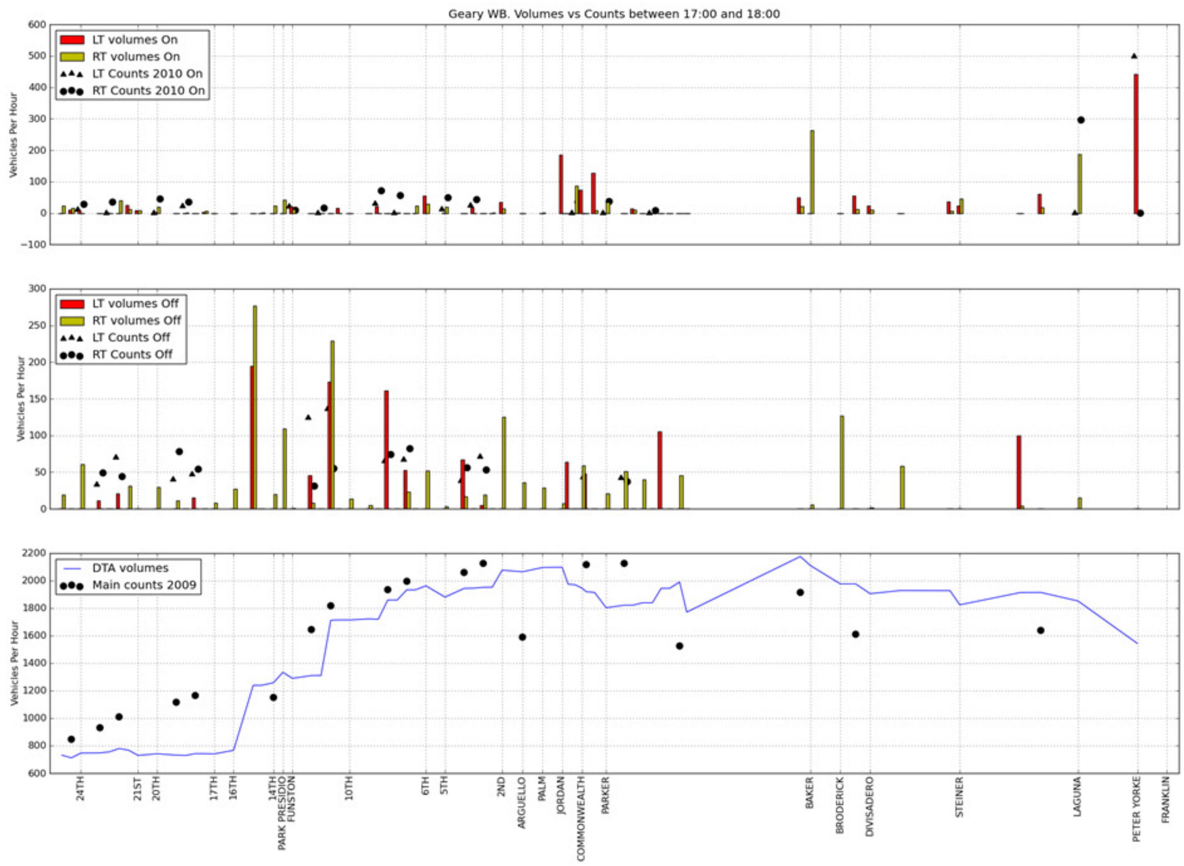

Figure 4-4. Graphs. Corridor volume plot from Dynamic Traffic Assignment Anyway.
(Source: San Francisco Transportation Authority.)

Privately Developed Tools

Simulation modelers often build a set of tools that read aggregate simulation results and produce the set of measures appropriate for calibration/validation purposes. Typical measures include:

- **Traffic counts or volumes.** These often are reported for time intervals different than those required for model calibration and validation (five minutes versus hourly intervals).
- **Travel times.** Travel times usually are reported at the movement, link, or roadway segment level; and it is up to the user to aggregate them at the corridor level and for different time periods.

Cambridge Systematics has built a spreadsheet-based tool that incorporates the above functionality and interfaces with VISSIM to obtain lane, link, and detector information. The time-dependent data obtained from VISSIM are filtered, refactored, aggregated, and reformatted to be displayed in several spreadsheets inside the same workbook. The tool is able to average and combine simulation results from different iterations, time periods, and spatial locations; and then visualize this information using the capabilities of Excel. Figure 4-4 shows a time-space speed contour plot that has been populated automatically inside Excel by the data coming from the post-processor. The open-source technologies we have used in the Cambridge Systematics computational engine are similar to SPSS or the R statistical language, which means that aggregation, statistical operations, and filtering are done simply and efficiently. The entire process is driven by the spreadsheet software itself, and all the data manipulations and statistical computations are done in the background. Implementing a statistical computational engine outside Excel proved to be always reliable and 50 times faster than doing the same computations using custom code in Excel VBA.

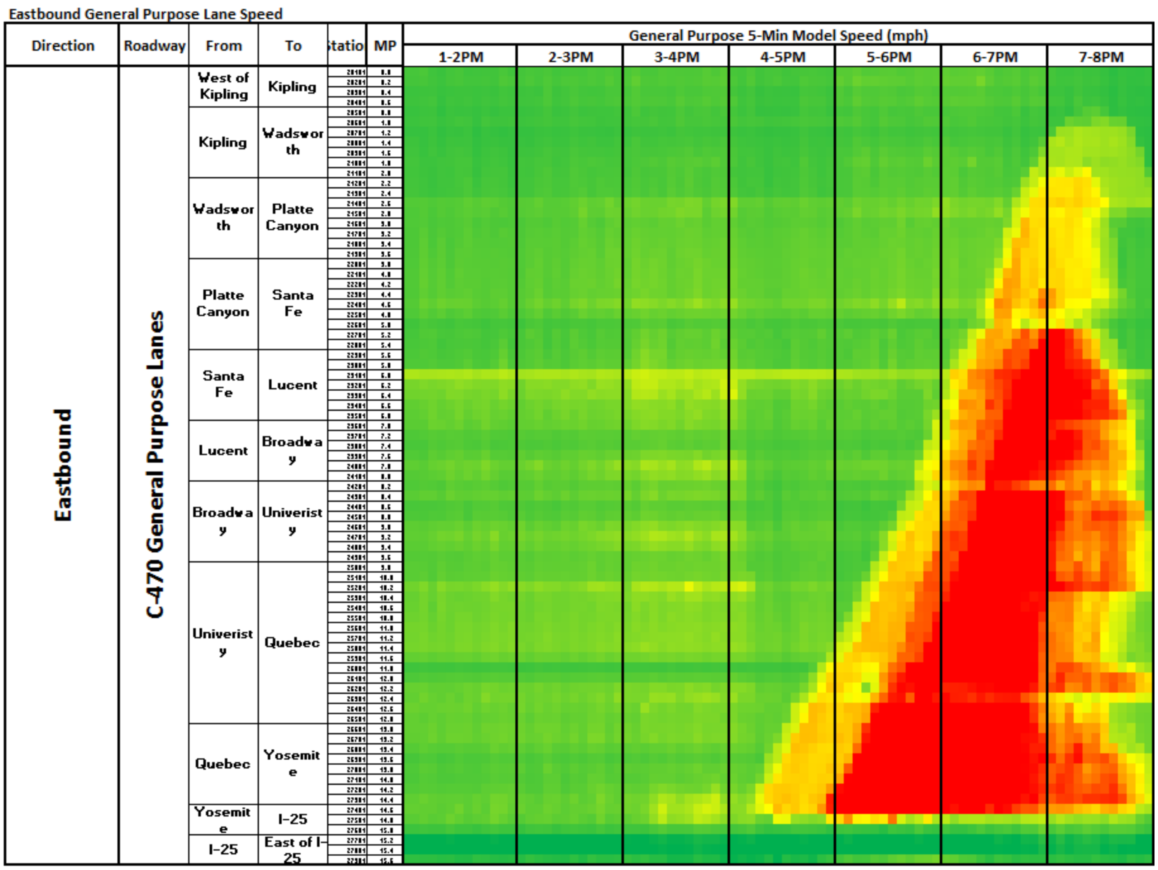

Figure 4-5. Heatmap. Cambridge Systematics spreadsheet-based simulation results processor.
(Source: Cambridge Systematics, Inc.)

Disaggregate Tools

Disaggregate tools read, store, and visualize individual trajectories, either from observed data or from simulation models. Trajectory files very often contain a large number of records that may not fit in a spreadsheet program or in memory. For example, simulating as many as 5,000 vehicles on a five-mile corridor generates more than 1 million record entries per hour when second-by-second positions are recorded. Therefore, the ability to window-in on a specific link or corridor is a useful feature that allows for faster computation and visualization. Calculating aggregate derivative performance measures, such as speed and density, is not a straightforward process; and minor implementation differences can result in inconsistencies. Visualization is an indispensable tool that allows the researcher to understand vehicle interactions and vehicle movements, both at the individual vehicle level and at the aggregate link or corridor level.

VTAPE

Vehicle Trajectory Analysis System (VTAPE) is trajectory analysis software for processing and visualization of vehicle trajectories, either from observed or simulated data (Lu, Bin, and Scott Washburn. 2014. “Vehicle Trajectory Analysis System for Estimating HCM Compatible Performance Measures”. Presented at the 93rd TRB Annual Meeting, Washington, D.C.). The software has been developed by Dr. Scott Washburn at the University of Florida over the course of several years to support the calculation of Highway Capacity Manual (HCM) compatible performance measures from trajectories. VTAPE employs internally a uniform trajectory database that allows the software to apply the same analytical procedures for performance measurement and visualization, regardless of the source data format. The uniform data format is shown in table 4-1 below and contains information on vehicle ID, position, lane ID, and time among other attributes. If the input data format does not contain all the necessary fields, a custom reader is used that calculates the required parameters as the data are being read. VTAPE has the significant advantage to be able to calculate queue, delay, density, and other performance measures based on the exact definitions included in the HCM. The software has been developed using C#, .NET, and the Microsoft Windows Presentation Framework in a modular way that supports expandability.

Table 4-1. Vehicle trajectory analysis system uniform database format.

Property	Data Type	Description
Vehicle ID	Integer	Vehicle identification number
Time	Double	Time step identification number
Position (Link)	Double	Distance traveled by the vehicle from the upstream end of the link
Velocity	Double	Instantaneous velocity of the vehicle
Acceleration	Double	Instantaneous acceleration of the vehicle
Link ID	Unsigned Integer	Link identification number
Link Length	Double	Link longitudinal length
Lane ID	Unsigned Integer	Lane identification number
Vehicle Length	Double	Vehicle longitudinal length
Leader ID	Integer	Vehicle ID of the leader vehicle in car-following movement

Source: Lu, Bin, and Scott Washburn. 2014. “Vehicle Trajectory Analysis System for Estimating HCM Compatible Performance Measures”. Presented at the 93rd Transportation Research Board (TRB) Annual Meeting, Washington, D.C.

VTAPE is capable of visualizing trajectories in two-dimensional time-space and other plots, as shown in figure 4-6. Useful features include the ability to color code trajectory position based on vehicle speed and the capability of the user to click on a specific trajectory trace interactively to obtain additional information.

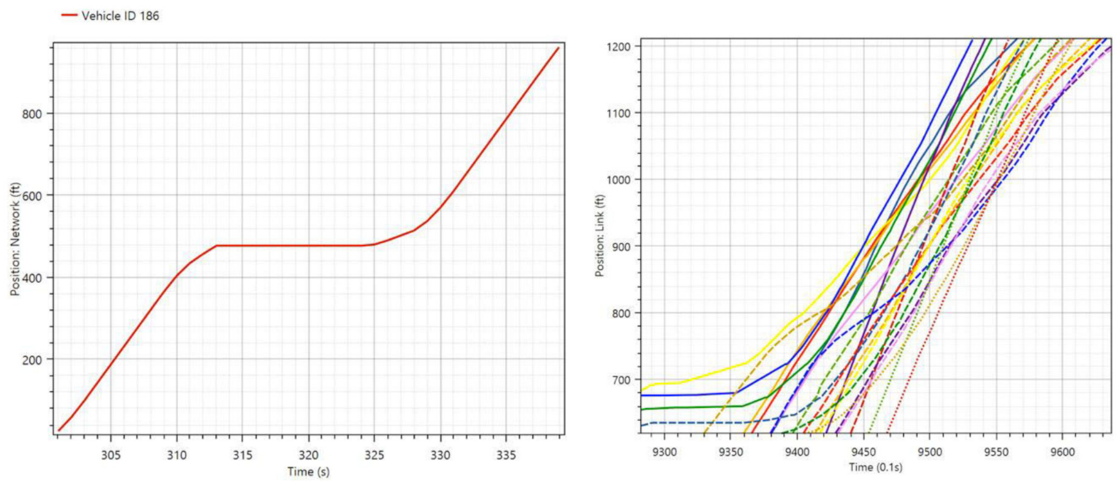

Figure 4-6. Graphs. Trajectory visualization in Vehicle Trajectory Analysis System.
(Source: Lu, Bin, and Scott Washburn. "Vehicle Trajectory Analysis System for Estimating HCM Compatible Performance Measures". Presented at the 93rd TRB Annual Meeting, Washington, D.C., 2014.)

Trajectory Explorer

Dr. Jorge Laval at Georgia Tech has developed a windows-based trajectory visualizer named Trajectory Explorer that is free to download and use (http://trafficlab.ce.gatech.edu/node/2001). Similarly to the tool developed by Dr. Washburn, trajectories are visualized in a time-space diagram and using color to visualize speed. The user can zoom-in and zoom-out interactively to take a closer look at specific locations and time-windows. This is a very useful feature that helps visually identifying vehicle movements at various levels of detail. In addition, the user can interactively specify cut-lines or rectangles for which the program calculates aggregate measures, such as density and flow. Figure 4-7 is a snapshot of the Trajectory Explorer Interface. Other advanced features include the ability to save data to text or bitmap files and the ability to derive the fundamental flow-density diagrams based on selected trajectories.

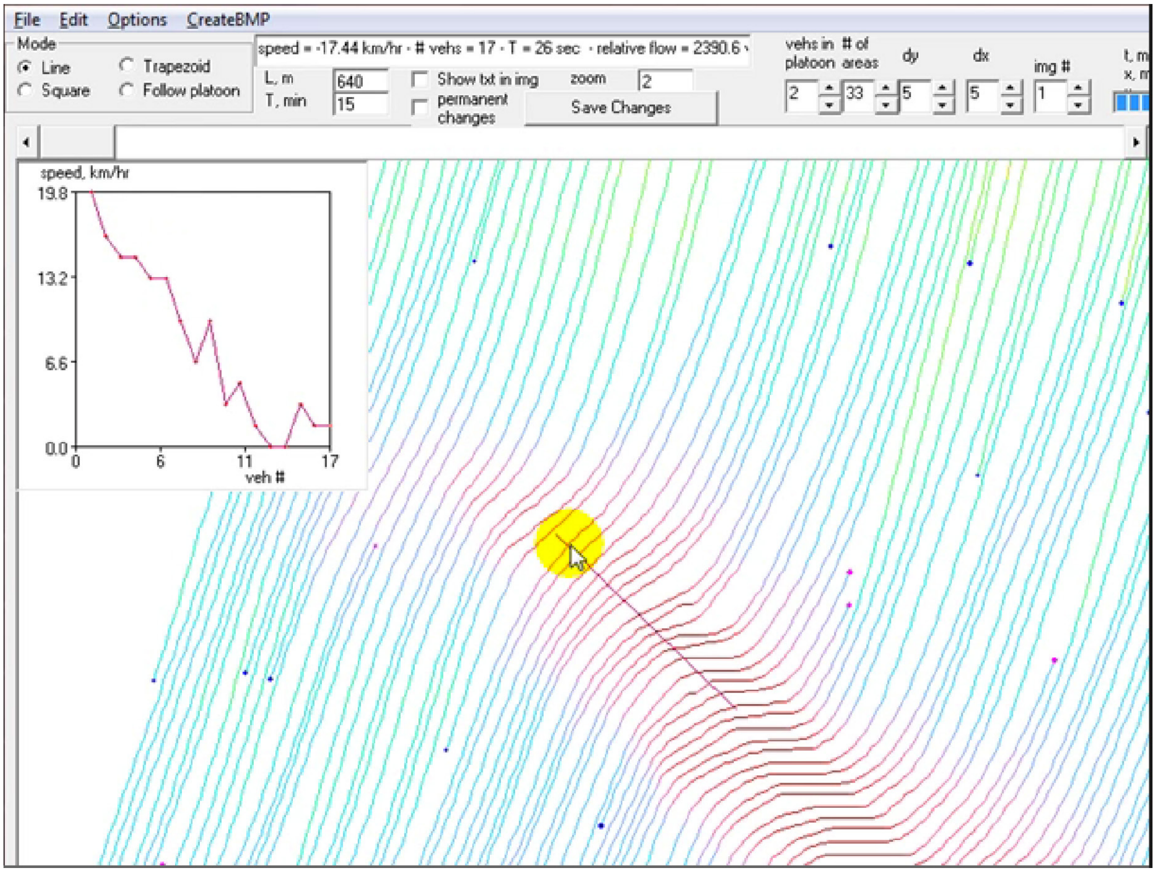

Figure 4-7. Visualization. Trajectory explorer.
(Source: http://trafficlab.ce.gatech.edu/sites/default/files/files/annimations/flash/te/tutorial2/tutorial2.html).

Cambridge Systematics' Trajectory Engine

The Cambridge Systematics trajectory engine uses an open-source spatiotemporal database to perform the following tasks:

- Individual Trajectory Visualization (to plots or shapefiles).
- Calculation of densities, flows, and aggregate trajectory visualizations.
- Computations of various derivative measures, such as distances between vehicles, gaps a few seconds before a lane change happens, and trajectory similarity measures, such as the Fréchet distance explained in chapter 5.

Figure 4-8 shows three visualizations from the trajectory engine, a trajectory plot on the left, a space-time diagram showing the rate of lane changes per mile in a corridor in the top right, and a scatterplot showing speed versus acceleration derived from NGSIM data.

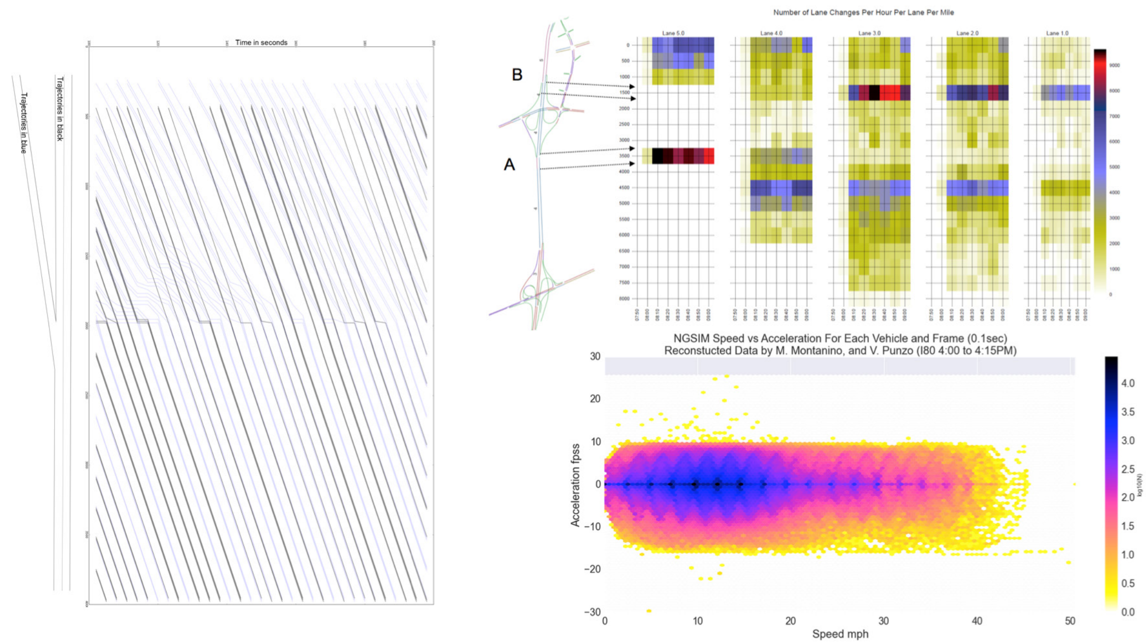

Figure 4-8. Charts. Cambridge Systematics computational engine visualizations.
(Source: Cambridge Systematics, Inc.)

PKU Trajectory Visualization System

PKU is a trajectory visualization platform developed by Peking University for both macro- and micro-scale trajectory visualization and analysis. The developers have used the software to publish research papers on sparse trajectory exploration, trajectory timeline visualization, traffic jam analysis from GPS trajectories, traffic density rendering, and micro-behaviors analysis at the intersection level. Documentation and animation videos are presented on the Web site: http://vis.pku.edu.cn/trajectoryvis/en/index.html. From a pure visualization standpoint, PKU is the most advanced software we encountered allowing for multiple views of the same data in coordinated charts that are all synchronized. In figure 4-9, the micro-behavior of vehicles traveling an intersection is analyzed across several dimensions. Multiple vehicle paths are shown on the top left of the picture (section a), while the multidimensional parameter space of all vehicles is shown in section c. Specifically, in section c, each trajectory dimension, such as start time, total time, average speed, max speed, and acceleration, is a separate vertical axis. Subsequently, each trajectory is visualized as a multipart line that connects different values in each of the axes (dimensions) from left to right. The plot allows the user to see the variability of the data in each dimension and how different values are correlated with each other in the multi-dimensional trajectory property space. PKU is developed in C++ using the Qt visualization toolkit. Even though the software does not contain the custom-specific charts frequently used in simulation, we believe it is worth exploring for its diverse capabilities that cannot be found in trajectory software developed in our field.

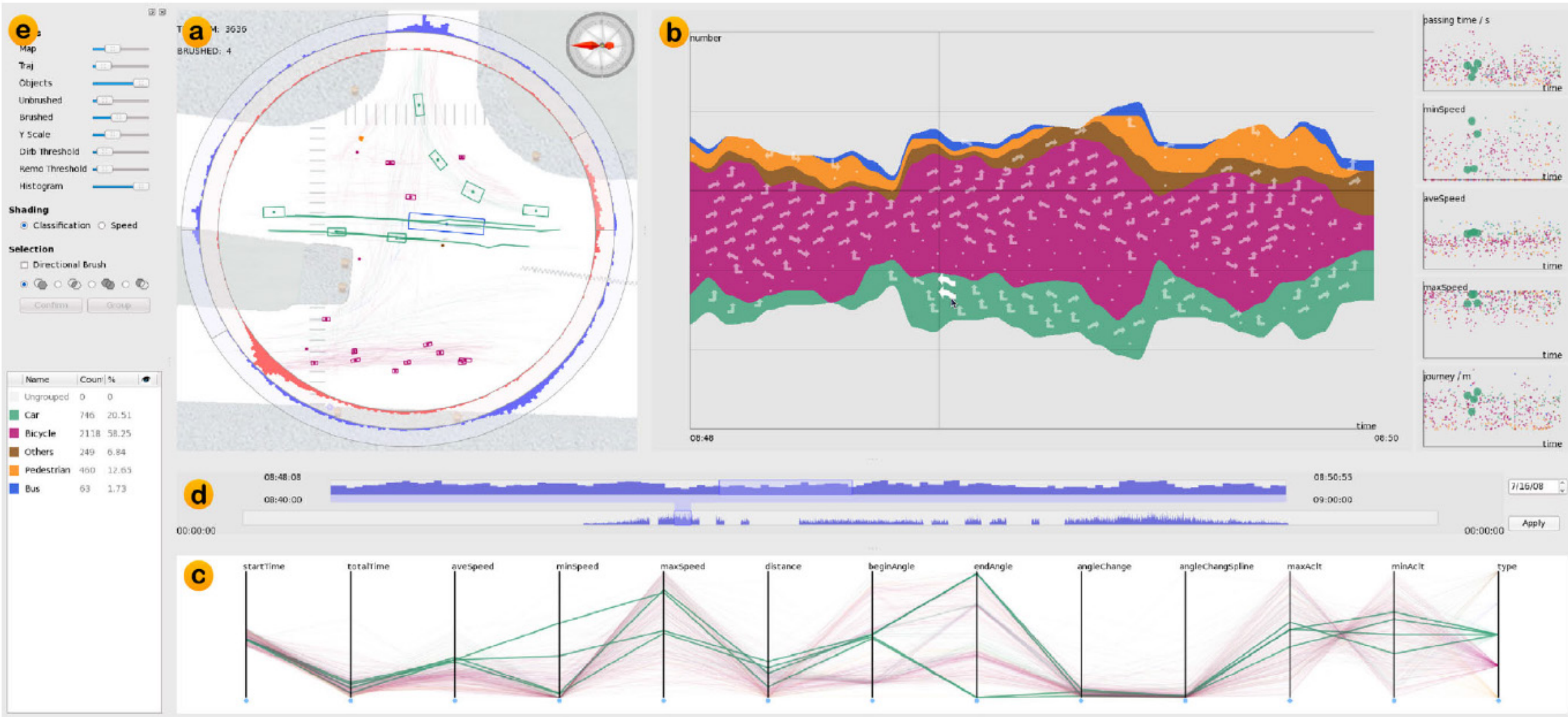

Figure 1: Interface of Triple Perspective Visual Trajectory Analytics (TripVista) visualizing traffic trajectory data at a road intersection. (a) Spatial traffic view showing geometrical trajectory information; (b) Temporal views of ThemeRiver and scatterplots; (c) Parallel coordinates plot showing multiple properties of the multi-dimensional data; (d) Time sliders for two-level time range selection; (e) Control panel for system parameter settings and data classification.

Figure 4-9. Visualizations. PKU trajectory visualization system.

(Source: Guo, Hanqi, et al. 2011. “TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection." Pacific Visualization Symposium (PacificVis), 2011 Institute of Electrical and Electronics Engineers (IEEE))

Chapter 5 Validation Efforts

In this section, we identify applications and cases where simulation models, algorithms, and logic have been calibrated and validated at the disaggregate trajectory level. Researching calibration, in conjunction with validation at the disaggregate level, is necessary because individual trajectories cannot be compared (or validated) on a one-to-one basis unless they belong to the same driver type. Performing calibration prior to validation ensures that the inputted car-following and lane-changing parameters are fitted to the properties of the observed trajectory.

Trajectory Comparison Function for Validation

When both trajectories correspond to the same time window, a straightforward measure to compare them is the sum of the square distances (or errors) between corresponding points. A more elaborate mathematical measure in the computational geometry field is the Fréchet distance, which can be loosely defined as the maximum distance between the two trajectories (Figure 5-1). According to Wikipedia, the Fréchet distance between two curves is the minimum length of an imaginary leash required to connect a dog (or vehicle 1) and its owner (or vehicle 2) constrained on two separate paths, as they move along their respective trajectories from one endpoint to another. For validation purposes and in contrast to the sum of square errors, the Fréchet distance has a straightforward interpretation and units.

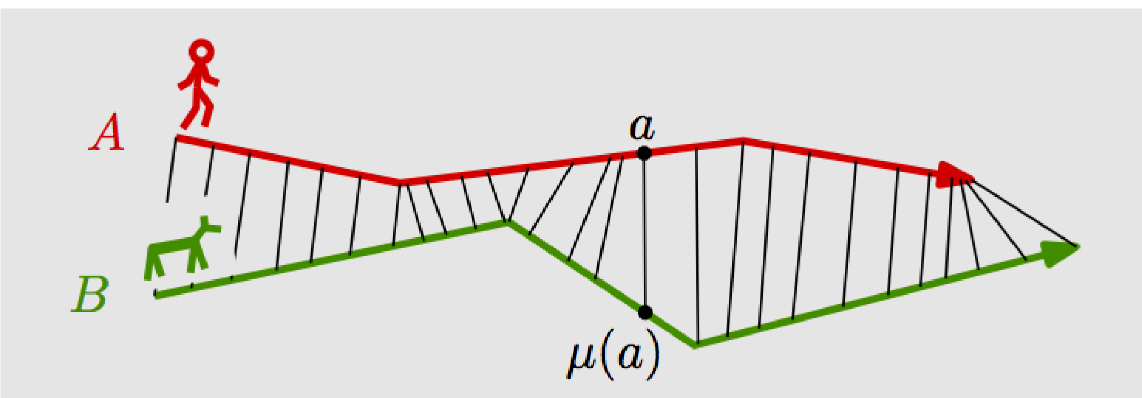

Figure 5-1. Illustration. Fréchet distance.
(Source: http://www.win.tue.nl/~mdberg/Onderwijs/AdvAlg2013/Selected-CG-Problems.pdf.)

Validating Algorithms and Logic

It is common practice to compare the fit of a model against observed data using an objective function, such as root mean square error, the rho-squared coefficient of determination or other, assuming, implicitly, that a model with better fit has always a higher predictive power. Aside from the particularities in the data that are used to calibrate or validate a model, Dr. Treiber proposes that the fitting quality should be one of the criteria for assessing models and not the only one. Additional criteria include robustness, parsimony, and parameter orthogonality. Each of these criteria is explained in more detail below along with intradriver, and interdriver variations that have an impact to model fit.

Robustness

Sometimes, different random seeds of the same simulation result to a gridlock or exhibit significant variations in model outcomes while the average of the outcomes, nevertheless, constitutes an acceptable fit. Sensitivity tests are required to ensure that small variations in input parameters do not result in significant variations in simulation outputs. Otherwise, the predictive ability of such a sensitive model may be limited.

Parsimony

In statistics, there are tests for parsimoniousness that balance the number of parameters against fit quality, such as the likelihood-ratio tests. However, transportation engineers often tweak microsimulation parameters on a link-by-link basis to replicate certain traffic phenomena introducing implicitly additional parameters to the simulation model that are not behaviorally based. In general, between two models of the same fit, the one that depends on fewer parameters has higher predictive power. Therefore, introducing a large number of parameters to improve the fit to the base-year condition may result in an over-fitted model with limited predictive power.

Parameter Orthogonality

Microsimulation models often include many parameters with overlapping influence on driving behavior. When a single parameter modification changes various aspects of driving behavior, it makes calibration and validation harder. In contrast, in a model for which there is one parameter for each aspect of modeled behavior (parameter orthogonality), it is easier to determine the "all other things being equal" effects of a change.

Intradriver Variations

Every driver changes behavior based on how alert he or she is under different driving conditions, such as accidents, mandatory or discretionary lane changes. Therefore, estimating a single set of driver parameters for the entire journey may be an oversimplification. Intradriver variation can be incorporated endogenously into the model by introducing event-oriented parameter changes or time-dependent adjustments (driving in the morning versus night). Intradriver variations may be one of the main reasons why even the best-calibrated car-following models have an average minimum error around 20 percent (Treiber, Martin, and Arne Kesting. 2013. "Microscopic Calibration and Validation of Car-Following Models–A Systematic Approach." Procedia-Social and Behavioral Sciences 80). Another plausible reason may be that we do not identify and model properly driver's anticipatory reactions to traffic past the leader vehicle or to lane-changing maneuvers.

Interdriver Variations

These variations pertain to differences in the driving behavior of the entire population and have to do with physiological characteristics, vehicle characteristics, or localized attitudes that differentiate drivers between different states, cities, and countries. Microsimulation models often assume a universal distribution of driver and vehicle characteristics neglecting such differences. Floating Car data from the NDS can be used to obtain insights on the observed distribution of driver characteristics. In addition to establishing the distribution of driver parameters, it also is important to acknowledge the correlations between parameter values that pertain to the same driver type (i.e., an aggressive driver may have a higher desired speed and acceleration targets.

Calibrating and Validating Car-Following Models

Researchers have used two types of data to calibrate and validate individual trajectories from simulation models:

1. Extended Floating Car data that contain the position, speed, and acceleration of a vehicle pair. Such data are collected by instrumented vehicles that carry a GPS device and a forward-facing radar that records the gap between the instrumented vehicle and the leader.
2. Trajectory information of all the vehicles in the traffic stream, such as the NGSIM dataset. Having a trajectory for all the vehicles in the traffic stream does not necessarily mean that driver parameters or driver type can be estimated for all the drivers since some drivers may encounter only one traffic condition (e.g., free flow) that does not give additional information on their behavior.

A significant number of researchers have developed a successful methodology to calibrate car-following models using NGSIM or Extended Floating Car data. Punzo and Simonelli were among the first to calibrate car-following models using trajectory data obtained from individual vehicles (Punzo, Vincenzo, and Fulvio Simonelli. 2005. “Analysis and comparison of microscopic traffic flow models with real traffic microscopic data.” Transportation Research Record: Journal of the Transportation Research Board 1934.1: 53-63). Their general methodology and the choice of distance and objective functions have been followed by subsequent researchers, such as Soria, Elefteriadou, and Kondyli, and form the basis of this section (Soria, Irene, Lily Elefteriadou, and Alexandra Kondyli. 2014. “Assessment of car-following models by driver type and under different traffic, weather conditions using data from an instrumented vehicle.” Simulation Modelling Practice and Theory 40: 208-220). Soria et al investigated how different car-following models, incorporated in AIMSUN, CORSIM, and MITSIM, perform in different operational conditions, including congestion or weather (rain or clear sky). Their study provided insights to the relationship between car-following parameters and different driver types, a relationship that, according to the authors, had attracted limited research.

Sanster, Rakha, and Du, in a recent study, and for the first time, used NDS data to calibrate and compare four different car-following models (Rakha, Hesham, John Sangster, and Jianhe Du. 2013. Naturalistic Driving Data for the Analysis of Car-Following Models. No. VT-2010-01). Their study focused on the cost and benefits of using naturalistic data, and concluded that “any project seeking to use naturalistic data should plan for a complex and potentially costly data reduction process to extract mobility data.” Specifically, the radar detection technology used for data collection was found unreliable, requiring manual verification using the forward-facing video camera to extract trajectory segments, for which a complete and reliable set of measurements existed. In addition, speed measurements coming from the vehicle network (OBD port) were found more reliable from GPS data that tended to oscillate. As a result, only 50 percent of the corridor-specific extracted trajectories were used for analysis. Despite problems with the reliability of the measurements, the authors noted that the unique combination of driver information, coupled with the vast amounts of recorded data, can

shed light on a number of important topics, such as driver heterogeneity (interdriver variation), and the relationship between driver behavior and roadway type (intradriver variation).
After cleaning the data, the authors used a methodology similar, but not identical, to Punzo and Treiber, to calibrate and validate four car-following models using NDS trajectory data. They concluded that the Rakha-Pasumarthy-Adjerid (RPA) model had the best model fit and was the most capable in explaining the variability of behavior in the dataset. An example of driver variability from their research is shown in figure 5-2 below. Black lines represent the combinations of spacing and speed in the dataset. Grey lines represent model estimates from the best-fit car-following model. Based on this figure, the authors concluded that the observed variability in the relationship between speed and spacing can only be partially replicated by model results.

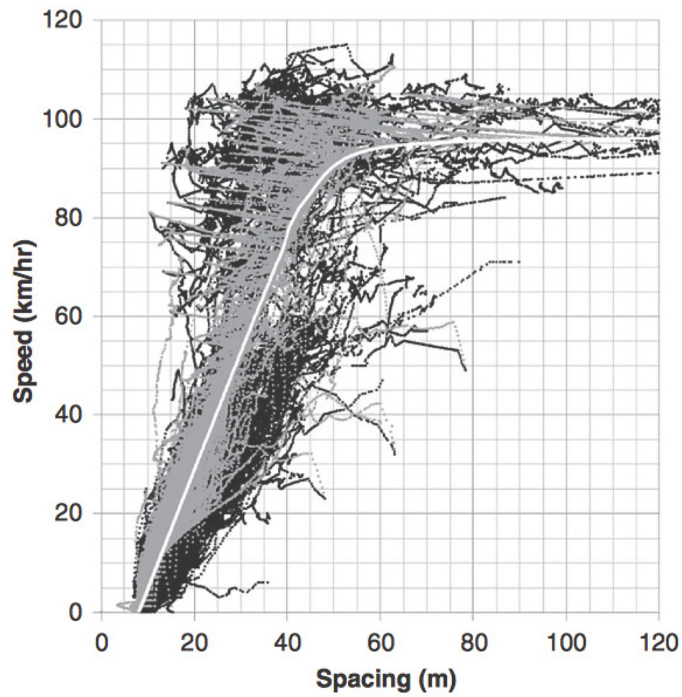

Figure 5-2. Diagram. Driver interaction diagram showing spacing versus speed.
(Source: Rakha, Hesham, John Sangster, and Jianhe Du. 2013. Naturalistic Driving Data for the Analysis of Car-Following Models. No. VT-2010-01.)

Very recently, Kesting and Treiber built on the methodology of Punzo and Simonelli and developed a detailed methodological framework for calibrating car-following models (Kesting, Arne, and Martin Treiber. 2008. "Calibrating car-following models by using trajectory data: Methodological study." Transportation Research Record: Journal of the Transportation Research Board 2088.1: 148-156). The rest of this section will borrow heavily from their thorough approach, which built and expanded previous research. Specifically, Kesting and Treiber investigated the minimum number of traffic states required for fully calibrating a car-following model against a given trajectory, the temporal resolution of the input data, and the impact of the form of the trajectory comparison function on the results.
The number of traffic state regimes contained in the data should relate to the traffic state regimes modeled by the car-following model at hand (Gipps, Optimal Velocity, or other). Different microsimulation models describe driver behavior through a different set of regimes; all of which must

be contained in the observed data to perform calibration. In contrast, and if driver type can be somewhat inferred, trajectories pertaining to any combination of traffic states can be validated using the Fréchet or other distance measure. Even though clustering techniques can be used to group similar trajectories, trajectory similarity does not necessarily imply driver type similarity because there are situations, such as stop-and-go traffic, in which there can be relatively little differentiation between drivers. Rather, the clustering mechanism should be able to identify similarity of responses under similar conditions pertaining to the traffic regimes modeled. In table 5-1 below, we show the different driver regimes of the Intelligent Driver Model (IDM) model and the associated model parameter of each regime. A notable characteristic of the IDM is that all of its parameters are orthogonal, with each parameter describing one particular aspect of driver behavior only. The column "Identification Criteria" in the table below depicts how different driving regimes can be identified in the raw data. This is a data-mining step that is necessary in order to identify trajectories that can be used for estimating all the parameters of the IDM.

Table 5-1. Driving regimes for the intelligent driver model.

Intelligent Driver Model Parameter	Driving Regime	Identification Criteria
Desired speed	Cruising in free traffic conditions	Speed and the time gap are above data-driven limits v_c and T_c, respectively, and the absolute acceleration is below a limit a_c.
Maximum acceleration	Free Acceleration, (nonsteady state flow)	If the time gap is above T_c, the acceleration above a_c, and the conditions for cruising are not fulfilled.
Minimum space gap	Creeping and Standing Traffic	Essentially standing if the gap $s < s_c$.
Maximum desired deceleration	Approaching (nonsteady state flow)	If $v > v_l$, the time gap $s/v > T_c$, and the kinematic deceleration $(v - v_l)^2/(2s) > a_c$.
Desired time gap	Steady state car following	$T < T_c$ and none of the above conditions applies.

Source: Treiber, Martin, and Arne Kesting. 2013. "Microscopic Calibration and Validation of Car-Following Models—A Systematic Approach." Procedia-Social and Behavioral Sciences 80: 922-939.

Objective Function

The most commonly used method for model calibration involves making model runs to obtain complete simulated trajectories, which are then compared to the corresponding observed data using an objective function that is a function of the sum of the square differences/errors (SSE). The measure in SSE need not necessarily be distance between the corresponding points of the two trajectories; it also can be the gap between the leader and follower, speed, or acceleration. Nevertheless, distance measures in the objective function are favored because they can more easily capture all aspects of driver behavior, such as the minimum distance between vehicles while standing. The choice of objective function is important because it can highlight or obscure model responses under different traffic regimes. For example, if we are to use gaps in the objective function, then the longer gaps at free flow speeds are going to dominate the cumulative SSE value giving relative little weight to gaps related to stop-and-go traffic. Taking the logarithm of the gaps reduces the impact of the longer gaps, and at the same time dampens the effect of outliers in the observed data.

Minimizing the objective function identifies the combination of parameter values in the car-following model that minimize the distance between the observed and simulated trajectory, as shown for example in the top right corner of Figure 5-3. In more detail in Figure 5-3, Treiber and Kesting show the 'fitted landscape,' the combinations of car-following parameters that result in the lowest value of the objective function. Each plot in this figure shows how the combination of two specific car-following model parameters affects the overall fit of the simulated trajectory. For example, the bottom left chart shows how the different combinations of desired speed and deceleration impact the value of the SSE. Blue and purple areas in the plot have the lowest SSE values of less than 100 square meters, while red areas have an SSE greater then 500. A horizontal line has been drawn that shows the value of the deceleration parameter that minimizes the SSE across all charts at the bottom. The modeler can infer from the collection of plots below not only the best fitted values, but also the sensitivity around the minimum and the range of parameter values that result to a solution close to the optimum. The following plots also can give a quantitative answer to the question of simulating a different driver that has all the same characteristics but one.

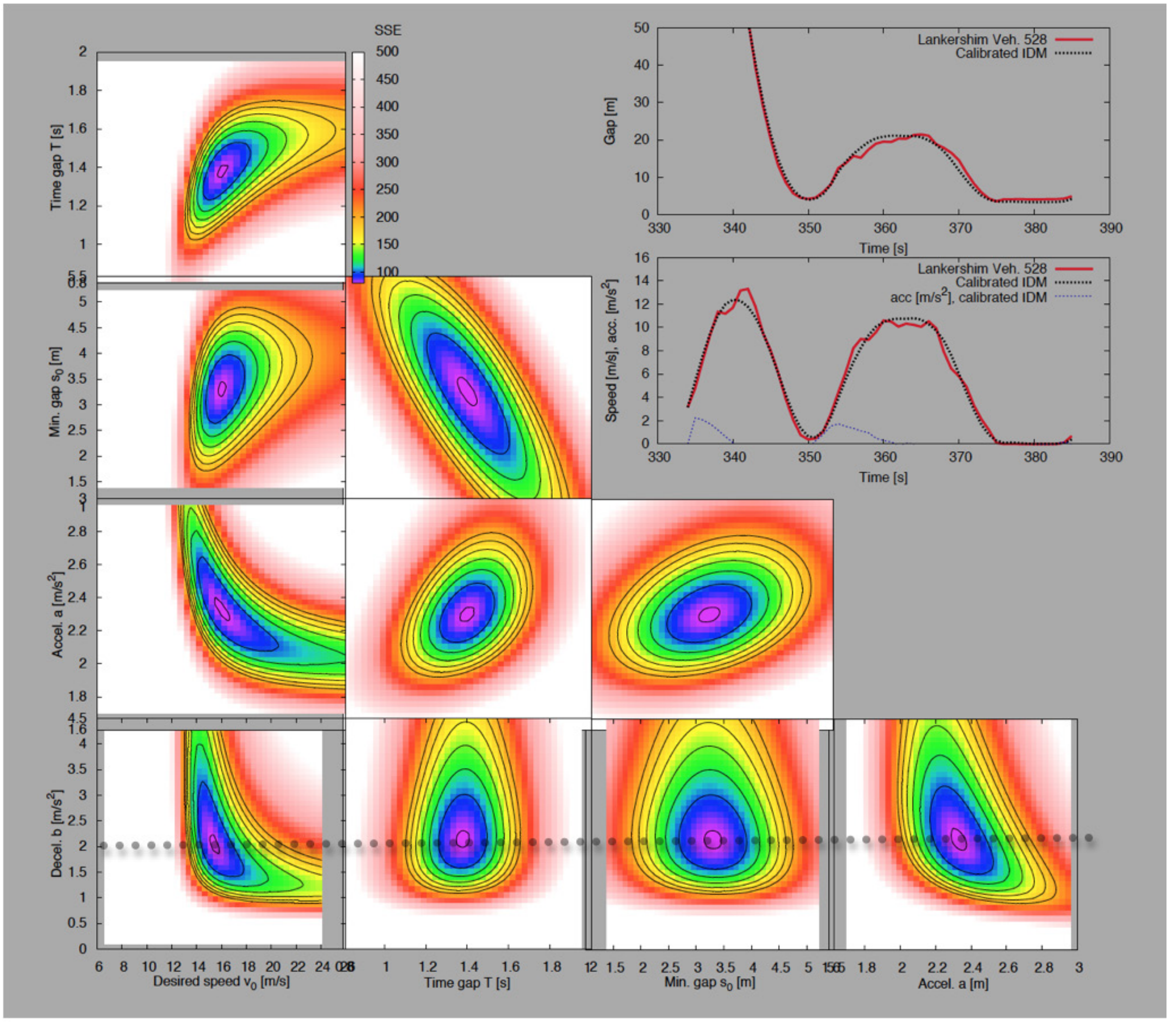

Figure 5-3. Graphs. Trajectory calibration landscape.

(Source: Treiber, Martin, and Arne Kesting. 2013. "Microscopic Calibration and Validation of Car-Following Models— A Systematic Approach." Procedia-Social and Behavioral Sciences 80: 922-939.)

Time Resolution

The sampling rate used in comparing observed and simulated trajectories is important for data collection purposes and for calculating an SSE value that is sensitive to differences. NGSIM and detailed floating car data have a sampling rate of 10 Hz, while other datasets such as the NDS use 1 Hz for the GPS component. Treiber and Kesting have experimented with different sampling rates by eliminating portions of the data. They conclude that, as far as car-following calibration is concerned, a sampling rate of 1 Hz yields the same parameters with a sampling rate of 10 Hz, including the same fitting landscape presented in Figure 5-3. This is an important conclusion for both calibration and validation because, if the error in fitting a trajectory is the same for sampling rates less than or equal to 1 Hz, datasets such as the NDS or others with a similar sampling rate can be used for validation and calibration purposes without biasing the results. Table 5-2 contains the fitted parameters for different sampling intervals with each column pertaining to the car-following parameters under a specific sampling interval. It is clear from the individual values, as well as the overall fit in the last row, that sampling rates up to 1 second yield the same model results.

Table 5-2. Impact of sampling rate on calibrated parameters.

	Sampling Interval (Seconds)				
	0.1	0.2	1.0	2.0	5.0
Desired speed (m/s)	16.1	16.2	16.3	15.8	14.8
Time gap (s)	1.20	1.21	1.22	1.12	0.87
Minimum space gap (m)	1.53	1.54	1.58	2.05	3.12
Maximum acceleration (m/s^2)	1.39	1.38	1.37	1.35	1.24
Minimum desired deceleration (m/s^2)	0.65	0.65	0.66	0.76	0.28
Error percentage	17.4	17.2	17.7	19.9	32.2

Source: Treiber, Martin, and Arne Kesting. 2013. “Microscopic Calibration and Validation of Car-Following Models—A Systematic Approach.” Procedia-Social and Behavioral Sciences 80: 922-939.

Validating Lane-Change Models at the Trajectory Level

The fact that individual trajectories cannot be compared unless they belong to the same driver type also holds when investigating lane changing in addition to car following. Algorithmic parameters of the lane-changing model often depend on certain parameters of the car-following model, making lane change conditional to car following. Therefore, the proper way to calibrate and validate a lane-changing model would be to perform the calibration of the car-following parameters first, and the calibration of the lane-changing parameters later. Lane-changing trajectory tracing tests can be constructed as follows:

- A vehicle’s car-following parameters are calibrated to same-lane observed data.
- A vehicle’s micro-environment (surrounding vehicles) in the same and adjacent lanes is replicated exactly based real observations.
- The vehicle’s lane-changing behavior is compared against observed data. Practically, this means that the first lane change the vehicle makes is compared against observed data.

Trajectory Clustering

In general, clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than those in other groups (clusters). Trajectories due to their spatiotemporal nature are more complex than simpler data types, such as vectors or tuples for which there is a rich literature of clustering methods. In the next paragraphs, we will describe some methods for clustering that have been adapted for trajectory analysis.

Many of the popular clustering methods rely on the notion of similarity or distance between objects. For example, in clustering a vector of attributes, the similarity or distance measure can be the Euclidian distance. Such a measure when applied to vehicle trajectory data can reveal drivers with the same spatiotemporal profile along the entire route that sped, decelerated, and accelerated approximately at the same time. However, given that vehicle trajectories are both constrained and influenced by the position, velocity, and acceleration of the surrounding vehicles, such a clustering may not reveal significant information about driver types, given the fact that the same driver may choose a completely different trajectory if the circumstances are different. If, on the other hand, the distance measure between trajectories is elaborate enough to encapsulate the inner structure of the data and the complete environment of driver decisions, a more insightful or behavioral clustering can arise. Independent of the choice of the distance measure, which will be analyzed in more detail in a following section, popular clustering algorithms that have been applied in other fields and are breaking into our practice include:

- K-means is a process that partitions all input objects into k clusters, where k is a parameter defined by the user. This is a heuristic method that attempts to find those k clusters that minimize intra-cluster difference at the same time that they maximize inter-cluster difference. The method starts by randomly selecting a partitioning and progressively refining it through iterations in which objects are swapped between clusters. K-means is probably the most popular clustering technique with several open-source machine learning and data-mining packages implementing it, including R, scikit-learn (Python), or Octave.
- Hierarchical clustering organizes objects into a multilevel tree structure of clusters and subclusters. The final output of the method is a dendrogram, in which each cluster is a node and subclusters are leafs to the node they belong. An example of such a dendrogram can be constructed by clustering the sequential letters {b,c,d,e,f,g,h,i}. Adjacent letters are more similar to each other and are clustered together (e.g., *b* and *c* together to form *bc* and *d* and *e* together to form *de*). The resulting clusters are clustered again in a hierarchical fashion (i.e., *bc* together with *de*) to form another level of clustering and so on until we reach at the root of the dendrogram.
- Density-based clustering identifies clusters of objects based on a similarity threshold epsilon without imposing a spherical shape to the resulting clusters, such as the k-means algorithm. As a result, clusters are formed in areas of higher density, while points in sparse areas separate the clusters among them. According to Wikipedia, DBSCAN and OPTICS are two of the most popular clustering algorithms currently available. Support for such algorithms exists in R, scikit-learn (Python), and ELKI (Java).

Popular clustering algorithms such as the ones described above need to convert trajectories into multidimensional vectors by means of a suitable distance function in order to operate. However, in our domain, two trajectories can be considered similar (or belonging to the same driver type) even if they do not fully coincide in space, have similar shapes, or have common start or end points. Vice versa, two nearly identical trajectories may not correspond to the same drive type if they are derived from a different environment or if they are derived from a traffic regime, such as stop-and-go traffic where differences between driver behaviors are minimized.

A Euclidian distance measure can be used only when the time window of comparison is fixed to ensure the same number of elements is being compared. Variations of the Euclidian distance measure have been used in trajectory tracing tests, in which the observed and simulated vehicles follow exactly the same path. More elaborate distance functions between trajectories that enable comparisons between trajectories of different length or time window include:

- Dynamic Time Warping (DTW) is a method that can be used when trajectories differ in length. DTW allows a sequence to “stretch” or “shrink” in order to better fit with another sequence.
- Longest Common Subsequence (LCSS), similar to the previous method, allows time series to be “stretched,” while allowing some elements of the sequences to be unmatched.
- Edit Distance on Real Sequences (EDR) is based on applications in bioinformatics that quantify the difference between two strings. EDR calculates the minimum number of insertions, deletions, and replacements in order for two strings to become identical.

Zhao et al. compare Euclidean, DTW, LCSS measures and clustering algorithms on the NGSIM Lankershim dataset (http://dcslab.cse.unt.edu/~zzm/FR.docx). CLUTO, the clustering software used, contains different classes of clustering algorithms, all of which were applied to NGSIM data (http://glaros.dtc.umn.edu/gkhome/views/cluto/). The researchers first clustered trajectories based on origin and destination into four distinct groups, and then applied three different similarity measures using three different clustering algorithms. Their results indicate that DTW and LCSS produced better clustering schemata compared to Euclidian distance. A more extensive comparison among six distance measures and seven clustering algorithms that was done by Morris and Triventi also highlighted the importance of a proper distance function in identifying a set of clusters (B. Morris and M. Trivedi, 2009. “Learning trajectory patterns by clustering: Experimental studies and comparative evaluation,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 312 to 319). In fact, based on the results of Triventi, the choice of distance function is often more important that the choice of clustering algorithm (i.e., k-means or other).

A selection of trajectory specific algorithms that build on top of k-means or density-based clustering methods is briefly discussed below:

- CenTR-I-FCM is a clustering algorithm that is based on k-means and transforms trajectories into vectors (Pelekis, Nikos, et al. 2009. "Clustering trajectories of moving objects in an uncertain world." Data Mining. ICDM'09. Ninth IEEE International Conference on. IEEE). The algorithm considers uncertainty, which can come from noisy data or sampling errors by allowing each data element to belong to different clusters by a certain probability of membership. For each cluster, the centroid trajectories are being produced and used in identifying patterns visually. Even though CenTR-I-FCM is a complex algorithm involving many steps, it is efficient and it has the advantage that clusters may not have the spherical shape of the traditional k-means algorithm.
- T-OPTICS is a clustering algorithm that is based on the vector-based OPTICS algorithm, a powerful and popular density-based clustering algorithm (Nanni, Mirco, and Dino Pedreschi. 2006. “Time-focused clustering of trajectories of moving objects." Journal of Intelligent Information Systems 27.3: 267-289; Ankerst, Mihael, et

al. 1999. "Optics: Ordering points to identify the clustering structure." ACM Sigmod Record. Vol. 28. No. 2. ACM). T-OPTICS is more efficient than hierarchical clustering methods and has the ability to construct clusters of arbitrary shape that are robust with respect to noise in the data. Furthermore, an algorithm for temporal focusing is included in T-OPTICS and aimed at finding the best time intervals for clustering. This is useful because two trajectories that are otherwise very different can be very similar in a specific time window.

- TRACLUS is a trajectory clustering algorithm aimed at discovering local patterns in portions of trajectories (Li, Zhenhui, et al. 2010. “Incremental clustering for trajectories.” Database Systems for Advanced Applications. Springer Berlin Heidelberg). To achieve this, trajectories are simplified into a number of line segments or subpaths and clustered in so called micro-clusters. Micro-clusters are then used to store compact summaries of similar trajectory line segments. TRACLUS uses the density-based clustering methodology DBSCAN (Birant, Derya, and Alp Kut. 2007. “ST-DBSCAN: An algorithm for clustering spatial–temporal data." Data & Knowledge Engineering 60: 208-221).
- FlowScan is an algorithm for discovering popular routes from trajectory data that have not been mapped to a network. A “hot route” is a general traffic flow pattern of nearby moving objects not necessarily adjacent.

Most of the algorithms presented above also can be used to find a representative trajectory for each cluster that can be either artificial or selected from the sample. Such a representative trajectory can be used to make conclusions about driver behavior and vehicle dynamics of the particular population it represents.

Clustering methods are becoming more and more popular in our field. In the last 10 years, there has been some notable research in trajectory clustering from the perspective of traffic simulation. Higgs et al. used a sample of 20 different drivers to identify car-following behaviors based on eight state-action variables: the longitudinal acceleration, the lateral acceleration, the yaw rate, the vehicle speed, the lane offset, the yaw angle, the range, and the range rate (Higgs, Bryan, and Montasir Abbas. 2014. “Segmentation and Clustering of Car-Following Behavior: Recognition of Driving Patterns”: 1-10). The results of this methodology are state-action clusters that define the driving pattern of drivers. The characteristics and frequency of recognized driving patterns are provided in the paper, along with the corresponding modeling parameters of each pattern from a traffic simulation perspective. Higgs et al., in a complementary paper to the previous one, identified the combination of state variables (speed, lane offset, yaw angle, range, and range rate) and action variables (longitudinal acceleration, lateral acceleration, and yaw rate) that constitute clusters using discriminant analysis (Higgs, Bryan, and Montasir Abbas. 2014. "Identification and classification of state-action clusters of car-following behavior." Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. IEEE). In another paper, Higgs et al. study intra-driver variations of car-following behavior by braking up car-following periods and clustering those that are similar together (Higgs, Bryan, and Montasir Abbas. 2013. “A two-step segmentation algorithm for behavioral clustering of naturalistic driving styles.” Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on. IEEE).

Chapter 6 Summary

Current approaches to develop, calibrate, and validate simulation tools are based on time-consuming approaches that use aggregate-level field data, such as 15-minute averages. Aggregate link flow or travel-time data may be insufficient to calibrate or validate simultaneously the four major components that comprise a microsimulation model, namely: 1) car-following; 2) lane-changing; 3) route choice; and 4) time-dependent demand. When using aggregate data, the model calibration space is a high-dimensional field that provides many options to achieve the same model fit. For example, to match modeled flow with the observed count on a link, the modeler can change the demand, or route choice, or car-following parameters that affect travel time on the corridor. All of these alternative approaches can improve the goodness of fit of the model. However, if the search for the best fit is not systematic and supported with data that can be used to calibrate each aspect of the microsimulation model in isolation, there is the danger of overfitting the model to observed base year conditions. Unintentionally, and in search of the best fit, the modeler may modify car-following or lane-changing parameters in a way that unrealistic driver behavior at the trajectory level is being produced involving, for example, too many transitions from acceleration to deceleration or lane changes. Without the proper trajectory analysis tools and the proper validation metrics at the trajectory level, modelers may not have way to measure the impacts of the calibration process on driver behavior and the resulting energy consumption or emissions.

In this document, we have researched trajectory datasets that have been collected since 1980 and have been researched by transportation professionals. For the purpose of this study, complete information for 100 percent of the traffic stream is required in order to reconstruct not only the movement of a single instrumented vehicle, but also the movement of *all* the vehicles around it that constrain or stimulate driver behavior. When complete trajectory information exists, researchers can study conditions and precursor events to lane-changing maneuvers on the same and adjacent lanes to develop causal (deterministic or probabilistic) models of driver behavior. Absence of information on adjacent lanes prohibits the study of lane changes at the trajectory level, except at an aggregate level that describes lane changing in statistical terms over the entire driver population. Trajectory observations that do not span the entire trip of the instrumented vehicle may not contain all the necessary information to gauge the purpose of a lane change (mandatory or discretionary). The study of lane changing from origin to destination requires datasets that we currently do not have in our disposal. Nevertheless, there is still a lot to mine from existing NGSIM and other trajectory datasets according to our stakeholders. The new naturalistic types of data hold significant promise provided that lane-changing maneuvers can be reliably identified.

The study of car following is less data intensive than lane changing because it does not require vehicle positions on the adjacent lanes. This is because most car-following equations relate driver acceleration to the gap, speed, and acceleration of the leader vehicle on the same lane. As a result, calibrating a car-following model can be done with a dataset containing positions of the instrumented vehicle and its leader, a dataset that is often obtained though a GPS and a radar device. The Naturalistic Driving Dataset, unlike datasets obtained by video detection, can provide insights about a driver's car-following behavior in a variety of conditions, including time of day, incident, or weather and, therefore, allow the study of driver behavior among a large population of drivers whose demographics are known.

Trajectory collection equipment and trajectory processing techniques continually advance, making it cheaper now to collect vehicle positions with an accuracy of a few feet that is necessary to determine lane-changing and car-following dynamics. GPS devices that use a ground base station, in addition to satellite information, provide increased accuracy at low cost. In the section of Existing Trajectory Collection Methods and Tools, we researched GPS technologies, radar, video detection, and unmanned aerial vehicles in extracting trajectory information. Video detection is the only method to provide trajectories for 100 percent of the traffic stream. However, even though significant advancements have been made in the last 10 years, advances that have resulted in the commercialization of the technology, a considerable amount of resources still needs to be committed to apply video detection in this project.

The almost ubiquitous GPS sensors in smartphones have propagated the amount of trajectory data that are collected from individuals or commercial vehicle fleets. New types of databases called spatiotemporal databases have been developed to store and query trajectory data. Researchers in the transportation field have developed software that processes vehicle trajectories, such as the ones coming from NGSIM, and allows the visualization and computation of a number of important metrics. The Validation Processes and Tools section introduces a number of such trajectory processing tools describing their capabilities, software framework, and limitations.

Finally, the section on Validation Efforts describes the complexity of the trajectory validation problem. Disaggregate trajectory validation requires that the same driver type operates the same vehicle in the same traffic environment. Recent car-following research efforts that have identified driver types by virtue of calibration are reviewed to obtain insights of how well an observed and simulated trajectory can match. Transferability issues between sites and traffic flow conditions make it harder to validate simulated trajectories unless trajectory data are collected for the same site for which a simulation model exists. In such a case, individual trajectories can be validated by performing trajectory-tracing tests, a technique that models a single vehicle in a completely controlled environment dictated by the observed trajectories of the surrounding vehicles. In addition to the disaggregate tracing tests, aggregate validation tests that compare aggregate measures from the site-specific simulated and observed trajectories also can be conducted and can reveal differences in driving patterns that traditional validation methods cannot uncover.

APPENDIX A. List of Acronyms

AMS	Analysis, Modeling and Simulation
ATDM	Active Transportation and Demand Management
BSM	Basic Safety Message
Caltrans	California Department of Transportation
COA	Certificate of Authorization
CORS	Continuously Operating Reference Stations
DGPS	Differential GPS
DMA	Dynamic Mobility Applications
DTA	Dynamic Traffic Assignment
DTW	Dynamic Time Warping
EDR	Edit Distance on Real Sequences
FAA	Federal Aviation Administration
FHWA	Federal Highway Administration
fps	Frame per second
GPS	Global Positioning System
HCM	Highway Capacity Manual
ICM	Integrated Corridor Management
IDM	Intelligent Driver Model
IEEE	Institute of Electrical and Electronics Engineers
LCSS	Longest Common Subsequence
LIDAR	Light Detection And Ranging
NDS	Naturalistic Driving Study
NGSIM	Next Generation SIMulation
NMEA	National Marine Electronics Association
NOAA	National Oceanic and Atmospheric Administration
OD	Origin-Destination
OEM	Original Equipment Manufacturers
OPUS	Online Positioning User Service
RPA	Rakha-Pasumarthy-Adjerid
SAVME	System for Assessment of the Vehicle Motion Environment
SHRP	Strategic Highway Research Program
SQL	Structured Query Language
SSAM	Surrogate Safety Assessment Model
SSE	Sum of Square Errors
STOL	Short Take-Off and Landing
SUV	Sport Utility Vehicle
TMC	Traffic Message Channel
TRB	Transportation Research Board
UAV	Unmanned Aerial Vehicles
USDOT	U.S. Department of Transportation
VMT	Vehicle Miles Traveled
VTAPE	Vehicle Trajectory Analysis System
WAAS	Wide Area Augmentation System
WSDOT	Washington State Department of Transportation

U.S. Department of Transportation
ITS Joint Program Office-HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

Toll-Free "Help Line" 866-367-7487
www.its.dot.gov

FHWA-JPO-14-193

U.S. Department of Transportation