Title

Implementation of a Tool for Measuring ITS Impacts on Freeway Safety Performance

Permalink

https://escholarship.org/uc/item/2nn3j1sd

Authors

Golob, Thomas F.
Marca, James
Recker, Will

Publication Date

2007-07-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Implementation of a Tool for Measuring ITS Impacts on Freeway Safety Performance

Thomas F. Golob, James Marca, Will Recker
University of California, Irvine

California PATH Research Report
UCB-ITS-PRR-2007-9

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation, and the United States Department of Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

Final Report for Task Order 5307

July 2007

ISSN 1055-1425

Implementation of a Tool for Measuring ITS Impacts on Freeway Safety Performance

Thomas F. Golob

James Marca

Will Recker

Abstract

The research was undertaken to develop a tool for assessing the impacts of changes in freeway traffic flow on the level of traffic safety. Safety is measured in terms of the probability of a reportable accident, and the tool is so far restricted to urban freeway mainlines with substantial traffic levels. The tool will: (1) monitor the safety level of freeway operations (2) aid in freeway planning. The tool was calibrated by applying advanced statistical models to actual data combined from two sources: Vehicle Detector Station (VDS) data for freeways in Orange County (District 12), and data on all reported accidents in Orange County from the Traffic Surveillance and Analysis System (TASAS). The analytical engine that drives the safety tool is based on models that are highly effective in identifying those myriad aspects of traffic flow that are statistically related to accident probabilities. It is recommended that Caltrans invest in projects that will validate the current work, and subsequently: (1) improve the accuracy of the safety predictions; (2) extend the applicability of the modeling approach to other Caltrans districts; and (3) evaluate the dissemination of safety predictions in real time.

Keywords

Traffic safety, VDS loop detectors, traffic flow models, real-time services

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

Executive Summary

The research was undertaken to develop a tool for assessing the impacts of changes in freeway traffic flow on the level of traffic safety. Safety is measured in terms of the probability of a reportable accident, and the tool is so far restricted to urban freeway mainlines with substantial traffic levels. (The tool does not apply to surface arterials, freeway ramps and connectors, and freeway mainlines with low traffic levels.) To the best of our knowledge, such a tool is available nowhere else in this or any other nation. Its development was made possible by Caltrans commitment to its mission to provide the safest transportation system in the nation.

The tool is meant to allow Caltrans and others to accomplish two things: (1) Monitor the safety level of freeway operations (2) for use in planning. In monitoring mode, the tool uses real-time 30-second data from single inductive loop detectors and processes these data in a way that is analogous to the travel time performance monitoring systems that display speed ranges and travel times to various waypoints. This tool produces continuous measurements of the odds of different types of accidents for all loop stations providing data. These measurements are then converted into ranges that can be displayed in terms of the familiar red-yellow-green markings that are used for levels of service. This advance warning provided by the tool can be used in pre-positioning resources. It can eventually provide critical information for managing traffic flow using metering systems, motorist warning devices and informational signs, and other technologies. The output of the tool also provides an extensive archive that is available for evaluating the safety levels of different sections of freeway, without having to wait a year or more for TASAS data to become available.

In planning mode, the tool can be used to compare the safety level of changes in traffic flow. This comparison can be accomplished using either actual or simulated data. Using actual data, safety levels of situations before and after an intervention specifically aimed at improving safety, or any other change imposed on a section of freeway, can be analyzed using loop detector data from the two periods. As accidents are rare events, accident counts from TASAS and other sources are not effective in such assessments. Using simulated data, the tool can use loop detector data generated by micro-simulation models, such as Paramics, to test the safety level of hypothetical situations. This adds a safety dimension to project evaluations that have up until now been limited to calculations of travel time delay. The testing of do-nothing scenarios is seen as particularly valuable, as is the evaluation of the potential safety of traffic management schemes at Caltrans worksites.

The tool was calibrated by applying advanced statistical models to actual data combined from two sources: Vehicle Detector Station (VDS) data for freeways through Orange County (District 12), and data on all reported accidents in that county from the Traffic Surveillance and Analysis System (TASAS). The models compare the traffic conditions at the time and place of accidents against data elsewhere. The calibrated

models were then implemented in the California ATMS Testbed using live data from the Testbed Intertie with Caltrans District 12. The implementation generates safety predictions for freeway segments throughout Orange County all day long, and stores those predictions in a database. The implementation also has a web-based data browser component that allows analysts to examine the current predictions, as well as to explore historical predictions from all of the estimated safety models.

Our results lead us to conclude that the analytical engine that drives the safety tool is based upon models that are highly effective in identifying those myriad aspects of traffic flow that are statistically related to accident probabilities. Specifically, we discovered that certain ways of manipulating the raw loop detector data lead to clear and accurate precursors of the likelihood of a traffic accident.

It is recommended that Caltrans invest in projects that can serve to validate the current work, and then seek ways to improve the accuracy of the safety predictions, extend the applicability of the modeling approach to other Caltrans Districts, and evaluate the dissemination of safety predictions in real time.

The most important step toward implementation is to validate the current models based on TASAS data now being collected. Once that crucial step is taken and has resulted in an acceptable level of confidence in the accuracy of the models, the current project has already created a blueprint for implementation. The models are being actively applied to real-time data, and a live web server provides the ability to examine current safety conditions, and explore and download past conditions. Following validation, the step remaining is to disseminate the model and its predictions to a wider audience, including Caltrans Transportation Management Centers (TMCs), the California Highway Patrol, and other agencies. Our models are based on the premise that it is the mixing of drivers in a traffic stream that decreases safety and increases the odds for different kinds of accidents. Developing strategies to inform travelers of risks and instructing them to take preventative actions, such as staying in their lane, are important steps in calming traffic streams, so that accident risks can be minimized.

1 OBJECTIVES AND SCOPE

Our goal is to calibrate and verify a model that translates traffic flow, as measured by ubiquitous single loop detectors, into safety performance. By quantifying the safety benefits accrued from smooth and efficient traffic operations, transportation management agencies will be able to incorporate safety measures in assessment of performance gains resulting from deployment of Intelligent Transportation System (ITS) measures, such as system-wide ramp metering (SWARM), freeway service patrol (FSP) and other incident response measures, and advanced driver information. In the research reported here, our objective was to capture the relationships between traffic flow, as measured by an extensive set of statistical parameters, and the types of accidents that occur under different types of traffic flow conditions. The work builds upon previous work by the authors (Golob and Recker, 2003, 2004; Golob, Recker, and Alvarez, 2002, 2004a, 2004b; and Golob, Recker and Pavlis, 2007).

The project was conducted using data for six major Orange County freeways: Interstate Routes (I) 5 and 405, and State Routes (SR) 22, 55, 57, and 91. This study area covered 130 centerline miles. The period of study is March through August, 2001 (six full months).

METHODOLOGY

Our method can be broken down into eight distinct steps:

1. Design a set of statistical variables that capture as many aspects of traffic flow as possible using only 30-second loop detector raw volume and occupancy data.
2. Combine accident and traffic flow data to identify the traffic conditions at time and place of an accident.
3. Draw a comparable sample of non-accident traffic conditions.
4. Use probabilistic models to determine what traffic conditions are most likely to be associated with different types of accidents. Modify the statistical variables as necessary to optimize the performance of the probabilistic models.
5. Using the models results, code a tool that forecasts accident probabilities as a function of traffic flow.
6. Develop a user interface for the tool.
7. Store real-time input and forecasts for future validation studies.
8. Demonstrate potential planning and TMC monitoring tools.

2 CAPTURING TRAFFIC FLOW CONDITIONS

2.1 Data Source

Our traffic flow data are drawn from the Vehicle Detection System (VDS) thirty-second single loop detector data received directly from front-end processors (FEP) using the UCI ATMIS Testbed Intertie with Caltrans District 12. These loop detectors record volume (flow) and occupancy (the percent of time a vehicle is within the detection field of a loop) for each freeway lane at thirty second (30-s) intervals. There are approximately 8,000 VDS locations on California freeways, typically spaced one-third to one-half mile apart (Varaiya, 2005). The study area in Orange County contains 457 mainline loop detector stations.

From volume and occupancy, traffic density and point speed can be derived, but only under very restrictive assumptions of uniform speed and average vehicle length, and taking into account the physical installation of each loop. Such assumptions are relevant for aggregated data over extended periods of time. For our disaggregated purposes, there is no accurate information on average vehicle lengths for each 30-s interval, or for any aggregation of data over the periods of time (say, between ten and thirty minutes) needed to relate accident hazards to traffic flow conditions. Also, while some types of traffic flow detectors are able to identify the proportions of large trucks (long vehicles) for a given duration of traffic flow on different lanes, such data are generally not available at the time and location of very many accidents.

Consequently, we assume that absolute density and speed are *not* determinable, due to the absence of accurate effective average vehicle length for each 30-s observation. We do compute variables based on the parameter calculated to be the ratio of volume to occupancy (defined only for intervals with occupancy greater than zero), but we do not convert this ratio to speed by applying any conversion factors. We assume that occupancy and the ratio of volume/occupancy are scale-free parameters. We will demonstrate that, with a sufficient number of 30-s observations across multiple lanes of a freeway, we can capture important traffic flow characteristics without estimating actual speeds.

2.2 The Traffic Flow Variables

Repeated measurements of 30-second intervals of volume (flow) and occupancy for each of the freeway lanes are used to capture variations in various aspects of traffic flow. Sensitivity analyses revealed that we need at least thirty observations in order to estimate the traffic flow variables found to be important in our models. Because missing 30-s intervals are quite common in the VDS data we used, we needed the data calculations to be fault tolerant so that we could capture the traffic flow conditions at the time and place of as many accidents as possible. We settled on using twenty minute periods, so that we had a maximum of forty 30-s observations, allowing for a 25% rate of missing data. This maximized the number of acceptable times and locations without overly extending the time duration needed for the calculations.

To allow consistent definitions for all locations within our case study network, we use data for three lanes: (a) the leftmost, number one, or median lane (excluding any HOV lanes, designated "1"): (b) one interior lane (whichever has least missing data, designated "M" for middle), and the rightmost or curb lane (designated "R"). We confirmed that there are strong correlations of the three parameters (volume, occupancy and the ratio of the two) across all interior lanes, so there is no loss of information in using only one interior lane in locations where there are more than three lanes in one direction.

Twenty-seven statistical variables, falling into four types, were found to be useful. These are summarized in Table 1 and described below.

> Central tendency: The only one of the three traffic flow parameters for which we have a true scale is volume, so this is the only variable for which mean values are calculated over each twenty minute period of observation. Means are computed for each of the lanes, left (number 1 lane), interior, or middle (M), and curb or right (R).
>
> Statistical dispersion (variation): For volume, it is possible to use standard deviation, the root of the second order moment of the distribution over the twenty minute period. For occupancy and volume divided by occupancy, for which the scale is unknown, we use the coefficient of variation, defined as the ratio of the standard deviation to the mean. This is a dimensionless measure of variation in terms of units of the mean. The variations of volume, occupancy, and volume divided by occupancy are each computed for the three lanes (1, M, and R), resulting in nine parameters.
>
> Cross-lane correlations of traffic conditions: To measure the synchronization of the three parameters of traffic conditions across the lanes, correlations are computed for the corresponding values of volume, occupancy, and volume divided by occupancy in pairs of lanes. The three lane combinations are left versus middle (1 vs. M), left versus right (1 vs. R), and middle versus right (M vs. R).
>
> Autocorrelations, the correlation of a variable at one 30-s interval with the value of the same variable in the previous 30-s interval, for all adjacent time intervals in the 20 minute period: This is a scale-free measure of the level of systematic change in a parameter over the twenty minute period of observation. In theory, we can compute this for all three of our parameters, but tests showed that the autocorrelations of the volume/occupancy parameter were unstable and potentially misleading. Thus, autocorrelations were only computed for the two raw data parameters, each for the three lanes, resulting in six variables.

Table 1 The twenty-seven Traffic Flow Variables

variable type	measurement	lanes	variable	Abbreviated
Central tendencies	Volume	left (1)	mean volume lane 1	mean_vol1
		middle (M)	mean volume lane M	mean_volM
		right (R)	mean volume lane R	mean_volR
Standard deviations	Volume	1	standard deviation volume 1	sd_vol1
		M	standard deviation volume M	sd_volM
		R	standard deviation volume R	sd_volR
Coefficients of variation	Occupancy	1	coef. of var.occupancy 1	CV_occ1
		M	coef. of var.occupancy M	CV_occM
		R	coef. of var.occupancy R	CV_occR
	Volume/Occupancy	1	coef. of var. vol./occ. 1	CV_volocc1
		M	coef. of var. vol./occ. M	CV_voloccM
		R	coef. of var. vol./occ. R	CV_voloccR
Correlations across lanes	Volume	1 vs. M	correlation volume 1 vs. M	corr_vol1M
		1 vs. R	correlation volume 1 vs. R	corr_vol1R
		M vs. R	correlation volume M vs. R	corr_volMR
	Occupancy	1 vs. M	correlation occupancy 1 vs. M	corr_occ1M
		1 vs. R	correlation occupancy 1 vs. R	corr_occ1R
		M vs. R	correlation occupancy M vs. R	corr_occMR
	Volume/Occupancy	1 vs. M	correlation vol./occ. 1 vs. M	corr_volocc1M
		1 vs. R	correlation vol./occ. 1 vs. R	corr_volocc1R
		M vs. R	correlation vol./occ. M vs. R	corr_voloccMR
autocorrelation	volume	1	autocorrelation volume 1	autocorr_vol1
		M	autocorrelation volume M	autocorr_volM
		R	autocorrelation volume R	autocorr_volR
	occupancy	1	autocorrelation occupancy 1	autocorr_occ1
		M	autocorrelation occupancy M	autocorr_occM
		R	autocorrelation occupancy R	autocorr_occR

2.3 Demonstration of the Traffic Flow Variables for a Real Time and Place

These variables are best understood by demonstrating them for real times and places. The 27 traffic flow variables are computed at any location (mainline loop station) for any given 30-s interval based on the volume and occupancy data for that interval and the preceding 39 intervals (19.5 minutes). After each 30 second interval, the variables are recomputed after adding the new observation and dropping the oldest observation. Thus, after the initial calibration period of 20 minutes, the 27 variables each form a time series for the location. Viewing these time series graphically contributes greatly to an understanding what aspects of traffic flow the variables are capturing.

A location within the study area was chosen at random, with the only consideration being that there be a sufficient amount of valid loop detector data for at least a twelve hour period that involved both free flow and congested conditions. The location chosen was southbound SR-55 at 17th Street. The morning period was chosen for the day of November 28, 2006. Graphed in Figures 1 through 3 are all the raw loop detector data for the three lanes on which the computations of the variables are based. In each graph, occupancy is recorded on the left hand axis and volume (flow) on the right hand axis. The overall patterns are the same for the three lanes, but there are subtle differences that are exposed in our variables.

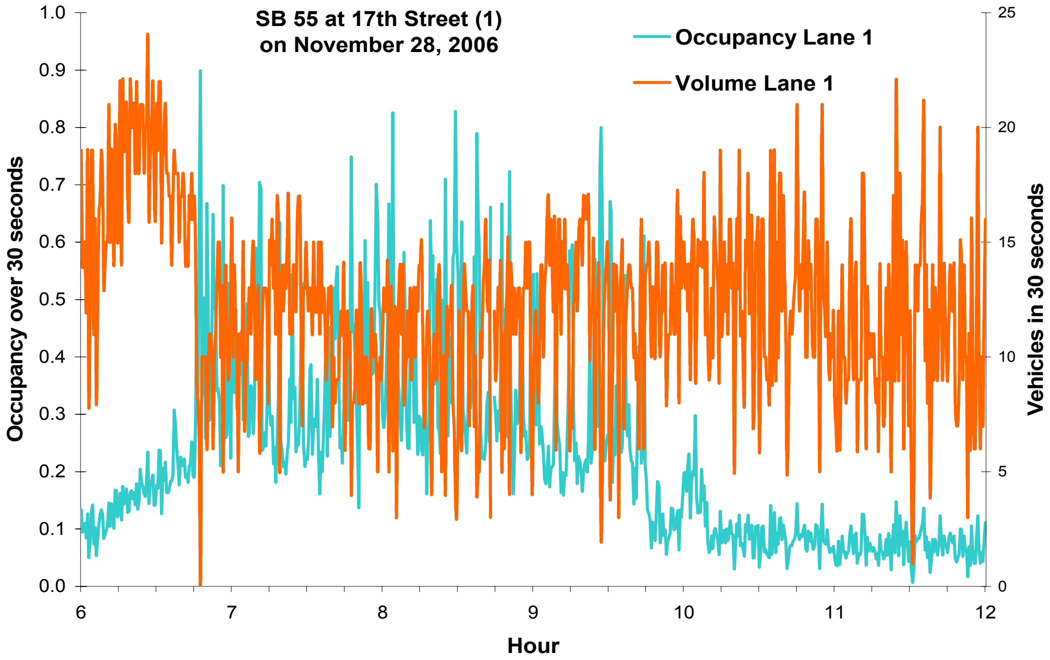


Figure 1 Raw Left Lane Volume and Occupancy Data for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

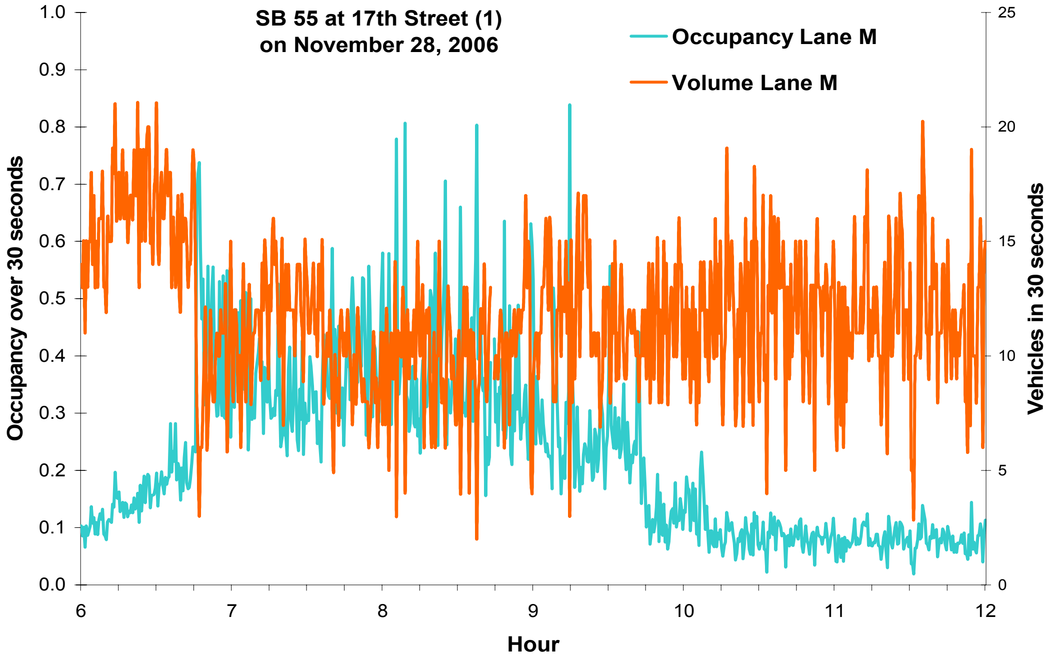


Figure 2 Raw Interior Lane Volume and Occupancy Data for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

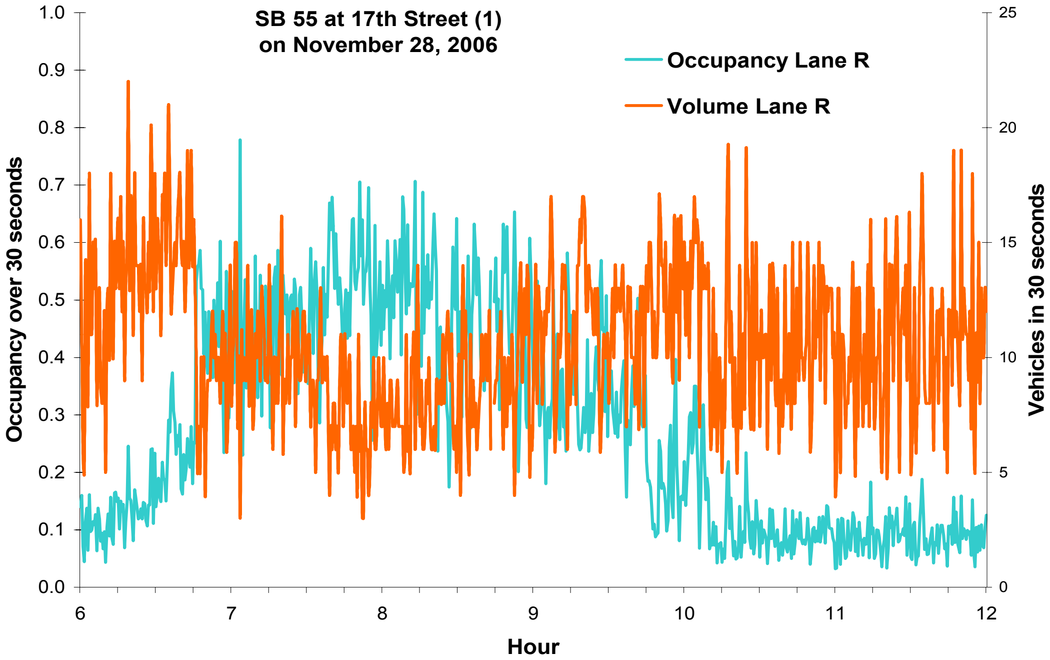


Figure 3 Raw Right Lane Volume and Occupancy Data for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

The third raw data parameter is simply the ratio of volume to occupancy, defined for all intervals with non-zero occupancy. These ratios are graphed in Figure 4 for lanes 1 and R. For clarity, lane M will be ignored in the remainder of this demonstration. It can be seen in Figure 4 that the freeway at this location was operating in free flow mode prior to 6:15 in the morning and later, after about 10:15. From 6:15 until 6:45 there was a steady decay in speeds, presumably due to excess demand. From about 6:45 until 9:45, the road is operating in what is commonly known as congested condition. The recovery to free flow after 9:45 was interrupted by increased congestion around 10:00.

In the free-flow regimes, the ratio of volume to occupancy in the right lane is consistently lower than the corresponding ratio in the left lane. This might indicate lower right lane speed, but it more probably indicates longer average vehicle lengths due to trucks in the right lane. This illustrates the problem of estimating speed from volume and occupancy. The wide variations of the ratio of volume to occupancy in the right lane during free flow operation are due to a combination of different vehicle lengths and different speeds that is impossible to separate for 30-s intervals.

All our traffic flow variables are computed from the raw data such as these graphed in Figures 1-4. In the remainder of this section, we superimpose various sets of our variables on the two time series of raw data parameters graphed in Figure 4 in order to illustrate how these variables relate to congested and uncongested flow regimes while exposing new aspects of traffic flow.

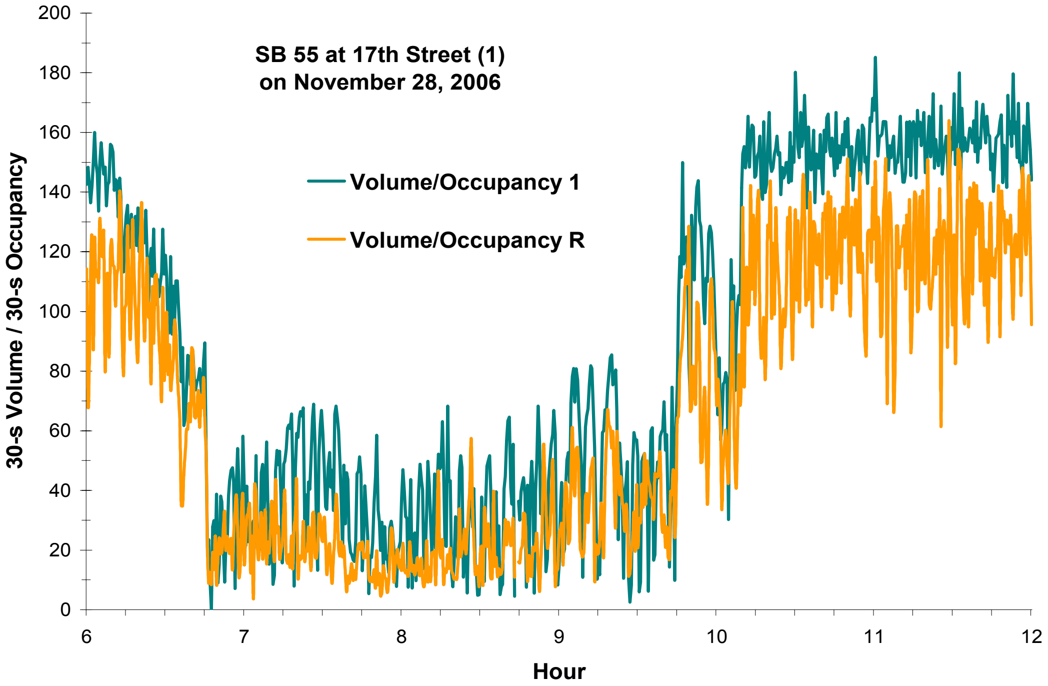


Figure 4 Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

The central tendency variables Mean_vol1 and Mean_volR are plotted in Figure 5. Their values, particularly for the left lane (Mean_vol1) are highest in the early morning free-flow period. At 6:35, a left lane average of 19.45 is attained for the past twenty minutes in the left lane; if sustained, this would equal 2334 vehicles per left lane per hour. This can be seen in Figure 1, where both volumes and occupancies are high in the pre-congestion period, indicating relatively close vehicle spacing at high speeds. Mean left-lane volumes do not change much from the congestion to post-congestion periods, but right-lane volumes generally improve substantially after attaining a minimum near the middle of the congestion period, where right-lane volume dips as low as 6.95 vehicles in 30 seconds (834 vehicles per lane per hour). Average left-lane volume is usually greater than right-lane volume, likely due to the presence of (longer) trucks in the right lane.

The variations of left and right lane volumes are plotted in Figure 6. These both spike at the breakdown of the road from free to congested flow. The variation in left-lane volume is generally always greater than the variation in right-lane volume, although the two are most similar during periods of transitions between periods of free and congested flow.

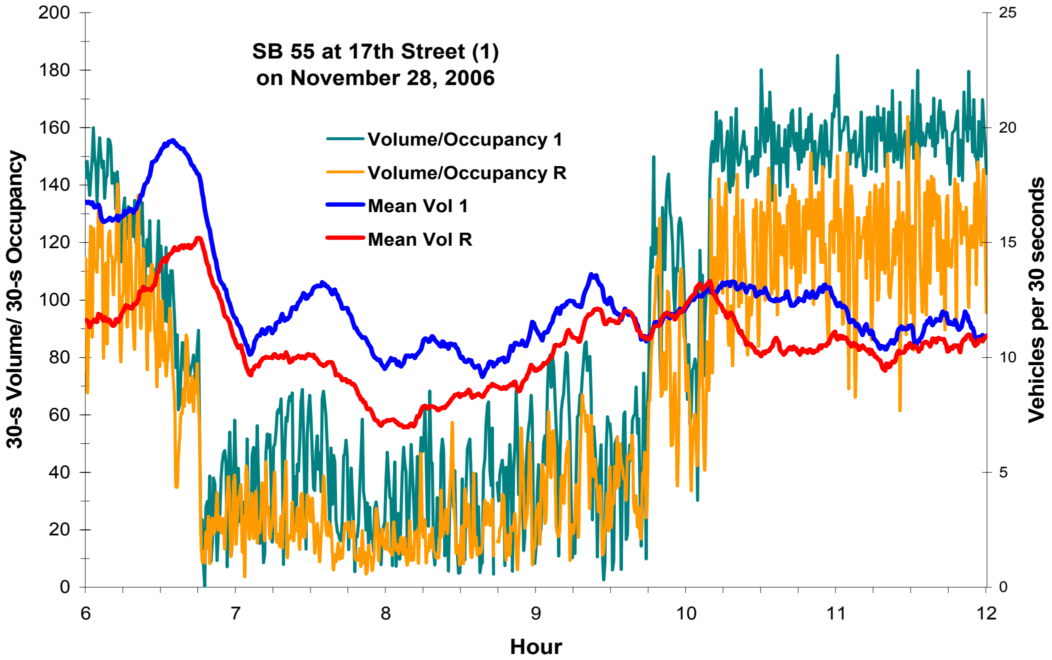


Figure 5 Means of Volume with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

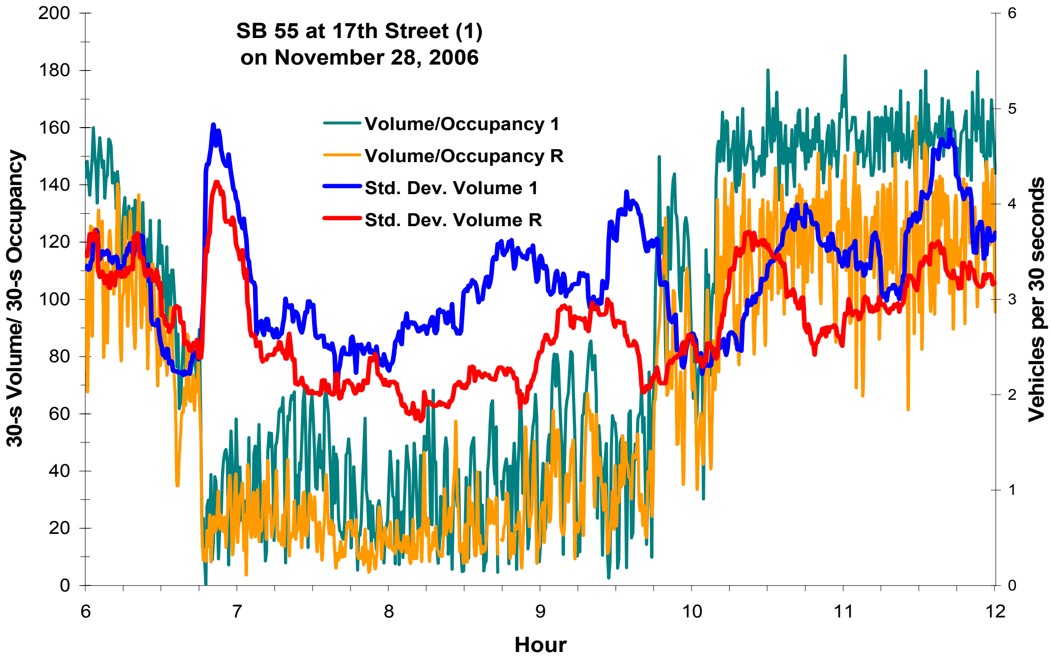


Figure 6 Standard Deviations of Volume with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

Variations in occupancy for the two lanes, measured by their coefficients of variation, are graphed in Figure 7. These standardized variations in parameters proportional to density spike for both lanes in both the periods of breakdown and recovery. The variation in the density-related parameter is consistently greater for the left lane during the period of congested flow; the two variables are more similar during periods of free flow.

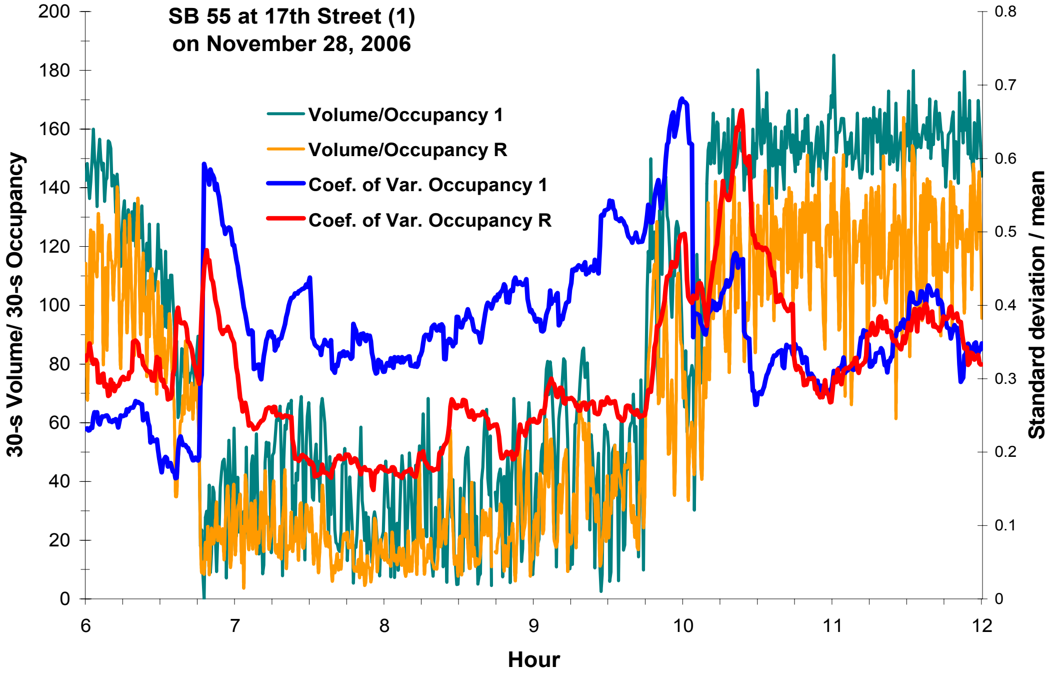


Figure 7 Coefficients of Variation of Occupancy with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

Variations in the ratio of volume to occupancy, measured by their coefficients of variation, are graphed in Figure 8. These standardized variations in parameters proportional to speed are much higher during periods of congested flow, indicating that these variables are an effective way of measuring the level of congestion. Also, right and left lane variations are more similar during congestion; during conditions of free flow, variation in this parameter is greater in the right lane than in the left lane. There is also a spike for both lanes marking the period of breakdown in flow.

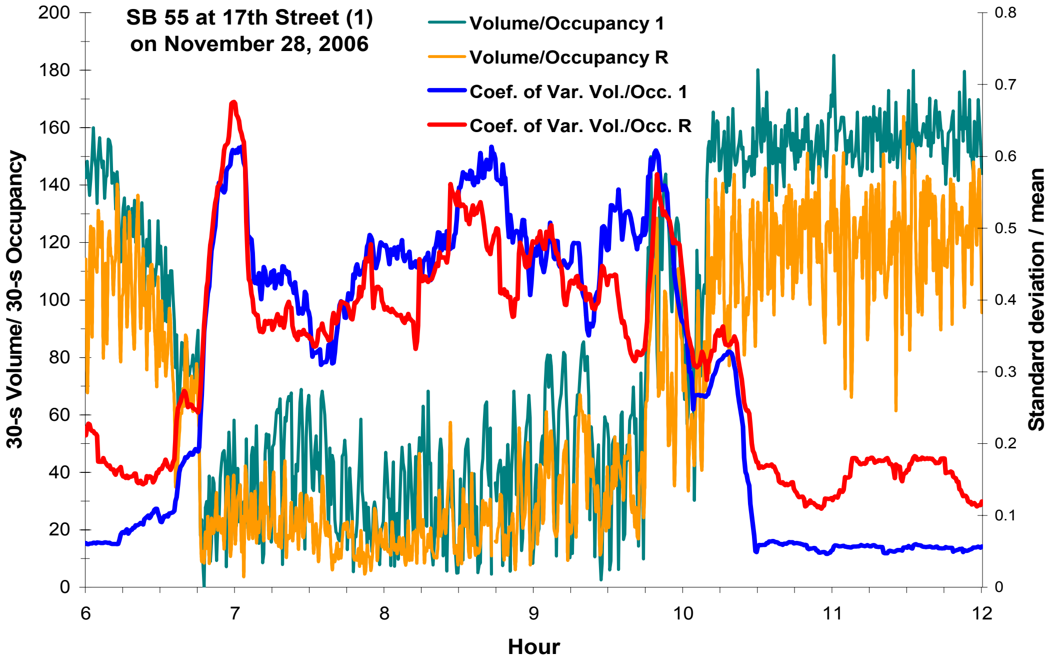


Figure 8 Coefficients of Variation of Volume / Occupancy with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

The next set of variables are the cross-lane correlations of each of volume, occupancy, and the ratio of the two. Correlations of volume are graphed in Figure 9. For ease of interpretation we show only two of the three correlations, ignoring for now the correlation between the observations in the interior and right lane (designated M-R). As in many of the previous variables, a spike in both cross-lane correlations of volume is detected at the onset of congestion. There is also a general upward trend in volume correlation between the right and left lanes over the course of the extended period of congestion. During periods of free flow, cross-lane correlations of volume are generally low, especially the correlation between the left and right lanes. A falling correlation of volume between the right and left lanes during a period of free flow appears to be a precursor of impending congestion.

Cross-lane correlations of occupancy are plotted in Figure 10. The patterns are similar to those displayed in the previous graph. Density correlations are high both at the point of a transition that marks a breakdown from free flow to congestion, and at a recovery from congestion to free flow.

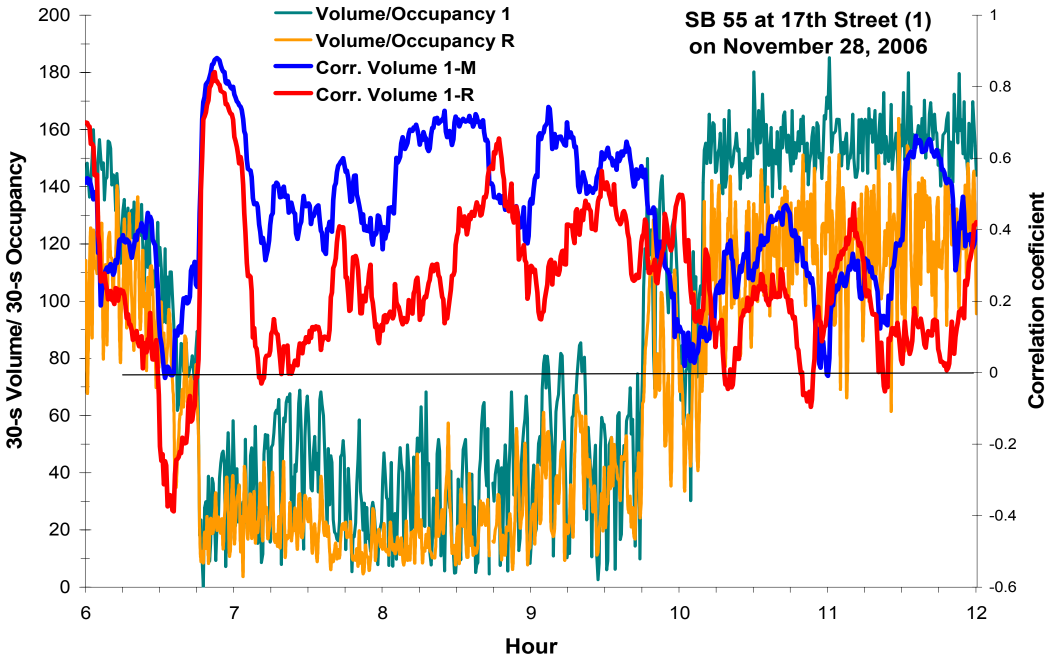


Figure 9 Cross-Lane Correlations of Volume with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

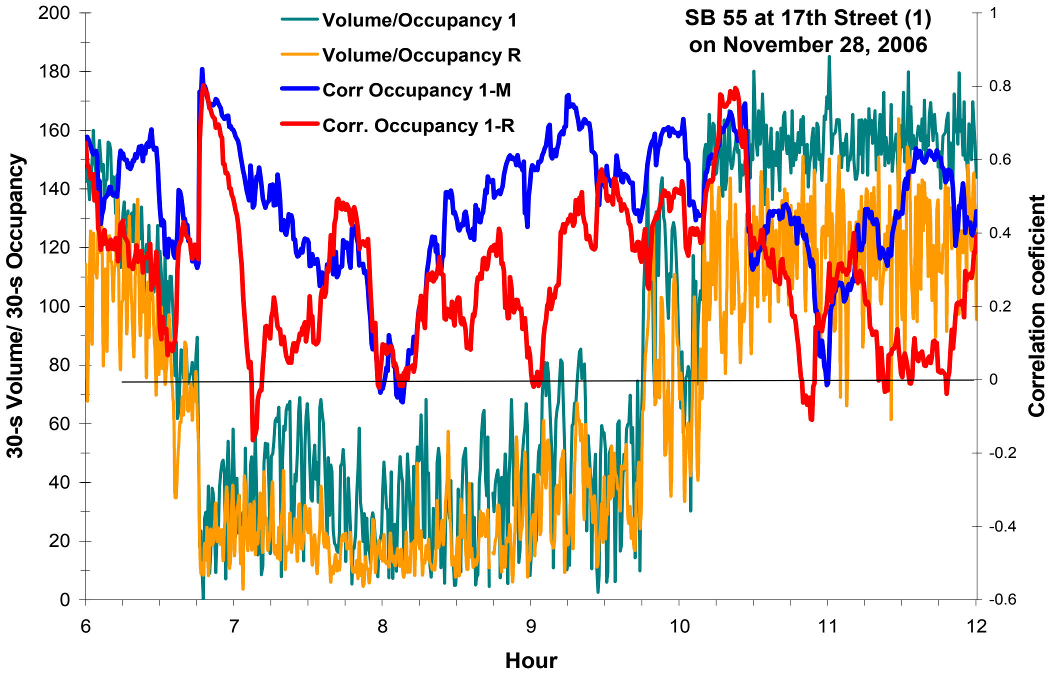


Figure 10 Cross-Lane Correlations of Occupancy with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

Correlations of the ratio of volume to occupancy are graphed in Figure 11. The correlation of this parameter across the outer lanes (1 and M), which is likely to be closely related to a correlation of speeds, since vehicle lengths are less likely to vary substantially over time for these two lanes, is high for the entire period of congestion. This presumably captures synchronized stop and go driving in the outer lanes. The correlation is much lower during the later free-flow period. The correlation between the left and right lanes, on the other hand, oscillates wildly when the road is congested. It approximates zero during free flow, and a rising level of this correlation is likely to signal a breakdown from free flow to congestion.

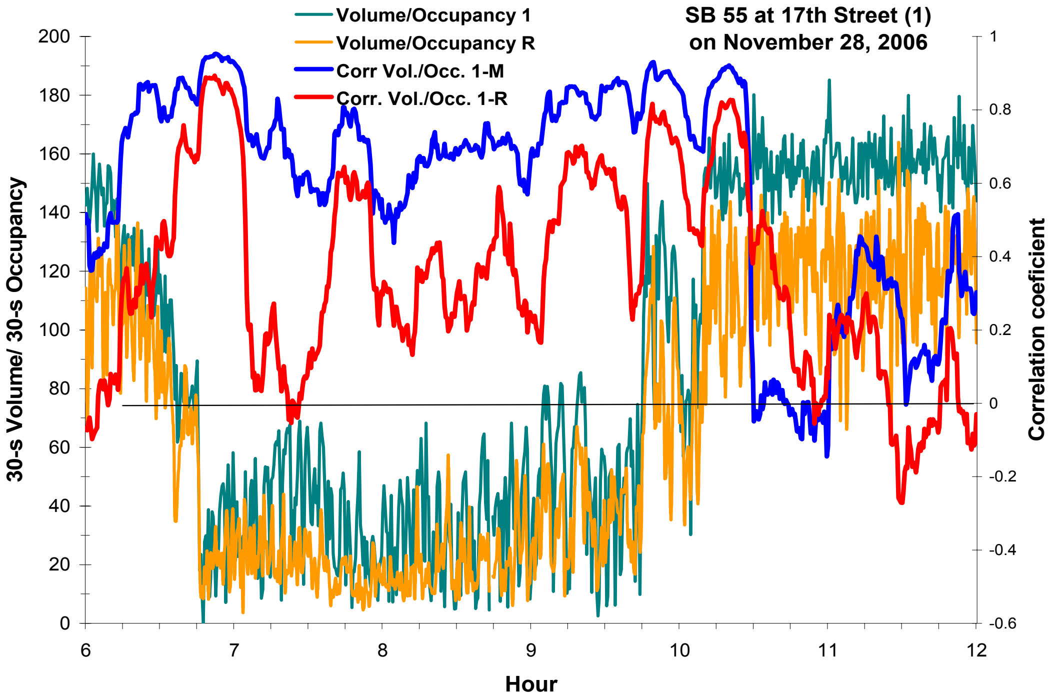


Figure 11 Cross-Lane Correlations of Volume / Occupancy with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

Finally, the autocorrelation variables for volume and occupancy are graphed in Figures 12 and 13, respectively. The autocorrelation of volume (Figure 12), measuring the degree to which a flow for a 30-s interval predicts the flow for the following interval, spikes at the onset of slow-speed, high density travel, immediately following the road coming to an almost complete stop. This is true for either lane. During free flow periods these autocorrelations fluctuate around zero. Autocorrelations of density (Figure 13) are similar, with the exception that the autocorrelation of the parameter proportional to density in the right lane also appears to be high at the onset of the recovery period.

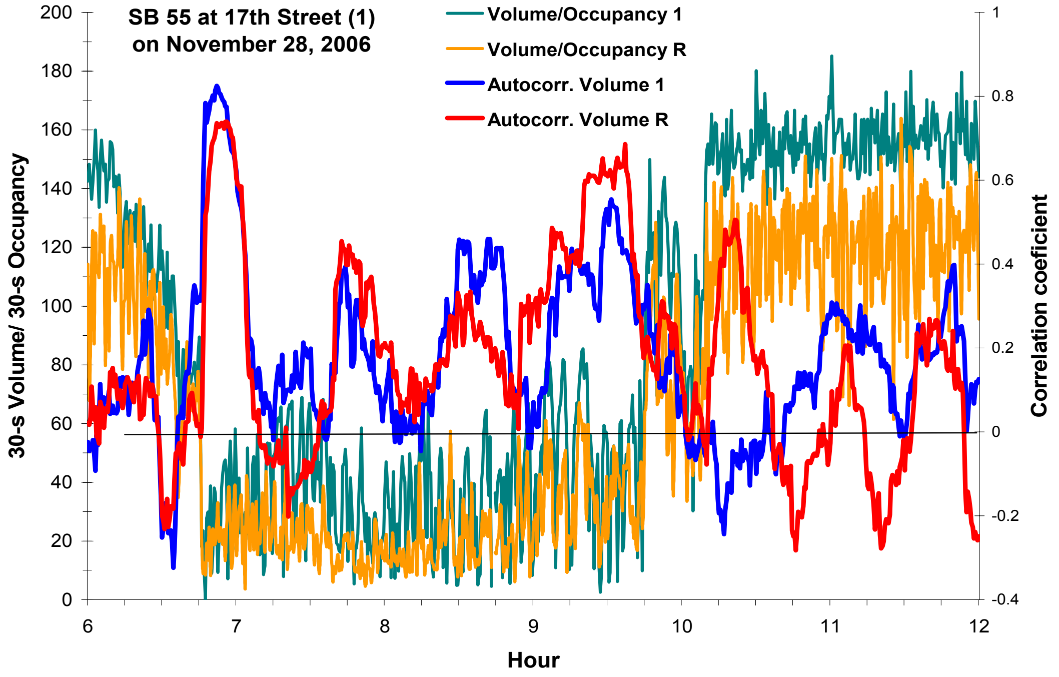


Figure 12 Autocorrelations of Volume with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

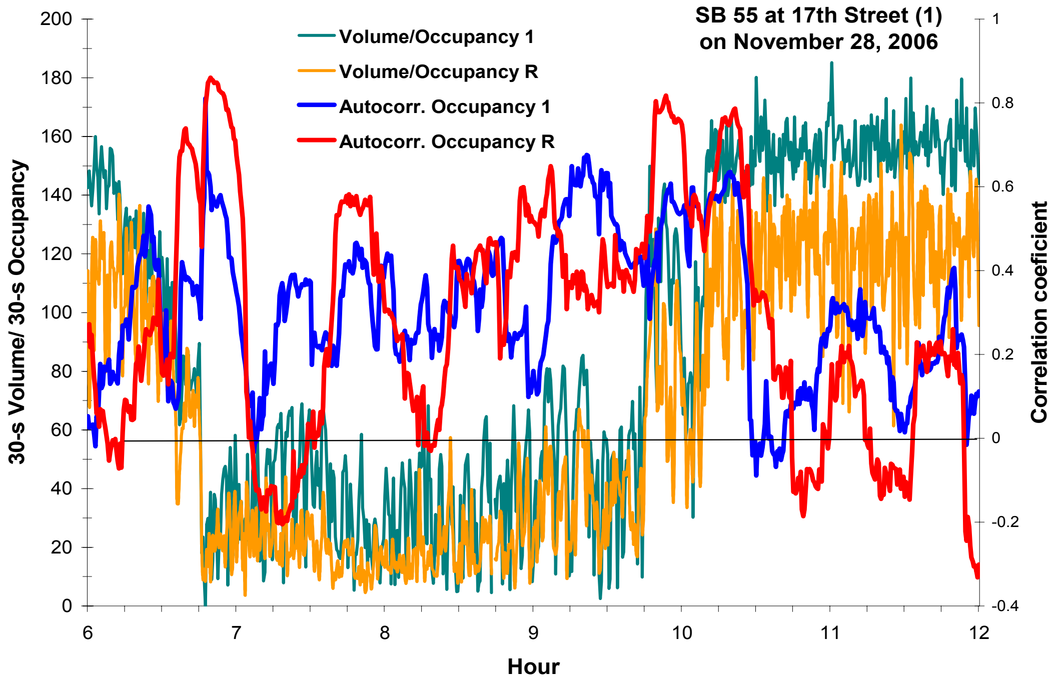


Figure 13 Autocorrelations of Occupancy with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

These general relationships between our traffic flow variables and the raw 30-s loop detector data should be verified with further comparisons of data at different times and places. This is outside the scope of the present project, but all of the examples we have seen lead us to believe that the patterns shown above are fairly typical. The extent to which these variables capture phenomena that is not readily apparent in the raw data is a subject for future research.

3 ACCIDENT DATA

The second data used in this project are from the Traffic Accident Surveillance and Analysis System (TASAS) database (Caltrans, 1993; FHWA, 2000), which covers all police-reported accidents on the California State Highway System. Most of the crashes included in the TASAS database were investigated in the field, but some were reported after the fact. The database does not cover crashes for which there are no police reports. We are concerned only with highway (mainline) crashes on urban freeways.

The TASAS data available to us contain the following types of crash characteristics: (1) the type of collision (rear-end, sideswipe, broadside, head-on, overturn), (2) the collision factor (e.g., speeding, following too close, illegal turn, alcohol), (3) number of vehicles and other parties involved, (4) the movements of each vehicle prior to collision (e.g., proceeding straight ahead, slowing, stopping, turning), (5) the location of the collision involving each vehicle (e.g., left lane, interior lanes, right lane, right shoulder area, off-road beyond right shoulder area), (6) the object struck by each vehicle (e.g., another vehicle, guardrail, bridge abutment), (7) number of injured and fatally injured persons per vehicle, and (8) environmental conditions, such as lighting, weather, and pavement conditions. No information was available concerning drivers and extent of injuries.

3.1 Capturing Traffic Conditions at the Time and Place of an Accident

Of the 4,412 accidents on the mainlines of our six freeways recorded in TASAS over the six-month study period, we have sufficient loop detector data for analysis for 1712 (about 39%) of these accidents. Chi-square tests performed on contingency tables revealed that the subset of accidents with sufficient traffic flow data is a random selection with respect to: (1) type of collision (e.g., rear end, sideswipe, hit object), (2) the number of vehicles involved, (3) the types of vehicles involved (e.g., automobile, panel or pickup truck, large truck), (4) location (which lane or side of road where the primary collision was located), (5) timing (time of day, day of week), and accident severity (injury or fatality versus property damage only (PDO)). The fortunate conclusion is that the subset of accidents with sufficient traffic flow data for analysis are a random sample of all reported accidents in the case study area over six months of 2001.

The time of each crash is not known with precision. An inspection of the crash times, presumably obtained from eyewitness accounts documented in police reports, reveals that almost 88 percent of the accidents in our study have reported times in minutes that fall precisely on the twelve five-minute intervals that comprise an hour. Thus, reported crash times must be treated as likely being rounded to the nearest five-minute interval, with a lesser secondary rounding to the nearest quarter hour. Since it is important that the traffic data in this study represent pre-crash conditions (rather than conditions arising from the crash itself), the period of the traffic-flow data used in the analysis was cut off 2.5 minutes before the nominal crash time. Under the presumption that any "rounding" is to the nearest five-minute mark, this ensures that in cases in which "rounding up" occurs data used is from the pre-accident conditions; for the "rounding down" case, there is no data conflict.

3.2 Random Sample of non-accident Traffic Conditions

A corresponding random sample of traffic conditions not at the time or place of any reported accident was drawn for the same study area and six-month study period. This draw was across 529,920 time periods, which is the number of 30-second intervals in 184 days. With 457 loop locations, there were thus approximately 2.4 x 10^8 potential time-space points in the sampling universe. A random sample of 4,441 observations was drawn from this universe. The test used with the accident sample for rejecting selections with insufficient data was also used with the random sample. The total final sample for modeling is thus 6,153, composed of 1,712 accident events and 4,441 events non-accident events.

4 Accident Probability as a Function of Traffic Flow

The models are designed to determine how the traffic flow is related to the probability of an accident. In other words, to what extent do our traffic flow variables describe accident potential? Logit (logistic regression) models are used to uncover any relationship between the traffic flow variables, based on the probabilities of occurrence of an accident event.

A binomial (or binary) logistic regression is used in the case of the event of any type of accident. In a simple (bivariate) logistic regression, a dichotomous criterion variable, e.g., accident is of type y (1="type y") or it is not (0="not type y"), is defined, and a model of the form:

$$\text{Prob}\,(y=1|x)=\frac{\exp(x\beta)}{1+\exp(x\beta)}$$

predicts the logit, that is, the natural log of the odds of the occurrence of the value being 1 for the dichotomous variable. That is,

$$\ln(Odds)=\ln\left(\frac{\hat{Y}}{1-\hat{Y}}\right)=x\beta$$

where $\hat{Y}$ is the predicted probability of the event $y=1$.

Multinomial logistic regression involves nominal response variables in more than two categories and is used to distinguish the occurrence of more than two types of outcomes. These models are used in explaining the traffic conditions under which injury versus property damage only accidents occur, and accident severity, in explaining collision location, and in explaining the number of involved vehicles. Multinomial logit models are multi-equation models. A response variable with k categories will generate k-1 equations. Each of these k-1 equations is a binary logistic regression comparing a group to the reference group. Multinomial logistic regression simultaneously estimates the k-1 logits.

For example, suppose that we had a response variable with three levels, say "no accident" (y=1), "property damage only accident" (y=2) and "injury accident" (y=3). The probabilities for each of the levels would be obtained as follows:

$$\text{Prob}\,(y=1|x)=\frac{\exp(\beta_1 x)}{\sum_{j=1}^{3}\exp(\beta_j x)}\ ;\quad \text{Prob}\,(y=2|x)=\frac{\exp(\beta_2 x)}{\sum_{j=1}^{3}\exp(\beta_j x)}\ ;\quad \text{Prob}\,(y=3|x)=\frac{\exp(\beta_3 x)}{\sum_{j=1}^{3}\exp(\beta_j x)}$$

This system of equations is unidentified; that is, there is more than one solution to the coefficients that lead to the same probabilities. To make the system identifiable, one of the coefficients is set to 0, called the reference category. It does not matter which one since they each yield the same probabilities. If we set $\beta_1 = 0$, i.e., define the category "no accident" as the reference category, the expressions above yield:

$$\text{Prob}\,(y=1|x) = \frac{1}{1+\sum_{j=2}^{3}\exp(\beta_j x)} \;;\; \text{Prob}\,(y=2|x) = \frac{\exp(\beta_2 x)}{1+\sum_{j=2}^{3}\exp(\beta_j x)} \;;\; \text{Prob}\,(y=3|x) = \frac{\exp(\beta_3 x)}{1+\sum_{j=2}^{3}\exp(\beta_j x)}$$

which, in turn, lead to the following probabilities relative to the reference group, in this case, y=1:

$$\frac{\text{Prob}\,(y=2|x)}{\text{Prob}\,(y=1|x)} = \exp(\beta_2 x) \;;\; \frac{\text{Prob}\,(y=3|x)}{\text{Prob}\,(y=1|x)} = \exp(\beta_3 x)$$

The two coefficients, β_2 and β_3 represent the log odds of being in the groups "property damage only accident" (y=2) and "injury accident" (y=3), respectively, relative to the reference group "no accident" (y=1).

Logit models apply maximum likelihood estimation after transforming the dependent variables into natural logarithms of the odds of whether or not an outcome occurs. The exponential function of each coefficient for each dependent category in a logit model gives the multiplicative effect of that variable on the odds of occurrence of the event in question.

4.1 The Likelihood of Any Type of Accident

The model results for any type of accident are shown in Table 2. To aid in interpretation and to optimize the precision of the estimators, the model only includes those variables with coefficients statistically different from zero at the 95% confidence interval, meaning that the probability of the coefficient being in fact zero is less than 5% (p < .05). The model goodness-of-fit is quite acceptable according to commonly accepted criteria for such types of probabilistic models. The model Nagelkerke Pseudo-R^2, an analogy to R^2 in linear regression, is 0.291. The fact that nine traffic flow variables were found to be statistically significant in terms of explaining the probability of an accident occurring is testimony to the robustness of the variables in explaining different aspects of traffic conditions. At least one of every type of traffic flow variable – central tendency, dispersion, cross-lane correlations, and autocorrelation – contributed to the explanation of accident likelihood. In addition to the nine significant variables, there are three interaction terms, which capture combined effects that are greater than or less than the sum of the individual contributions of the two separate variables. In all three cases, the significant interaction involves variables of different types.

The Chi-square statistic given for each traffic flow variable in Table 2 is an overall measure of the contribution of that variable to the explanation of the differences between the probabilities of accident and non-accident events. The dependent variable is coded so that the reference category is no accident, meaning that a positive coefficient relates to the event of an accident, and a negative coefficient relates to the event of no accident. The t-statistic of the coefficient is the ratio of the coefficient estimate to the standard error of that estimate. The odds ratios is simply the exponential of the coefficient. This gives the odds of the probability of an accident divided by the probability of no accident for a single unit increase in the traffic flow variable, or q/(1-q), where q = probability of an accident. Finally, *p* is the probability that the variable has no effect on whether or not an accident occurs, which is the probability that the estimated coefficient is zero, or that the odds ratio is equal to unity.

Table 2 Binomial Logit Model of Accident Occurrence as a Function of Traffic Flow Variables (Reference category = no accident)

Explanatory variable	Chi-square [a]	Coefficient	t-statistic	Odds ratio	*p* [a]
mu_vol1	7.91	0.038	2.81	1.04	0.005
mu_volM	34.98	0.100	5.91	1.11	0.000
CV_volocc1	12.30	1.080	3.48	2.94	0.000
CV_voloccM	4.28	0.627	2.13	1.87	0.033
CV_voloccR	4.89	0.553	2.21	1.74	0.027
corr_volocc1M	80.57	1.439	8.87	4.22	0.000
corr_voloccMR	15.82	0.658	3.97	1.93	0.000
autocorr_occM	5.11	0.412	2.26	1.51	0.024
autocorr_occR	22.61	1.424	4.67	4.15	0.000
corr_occ1M * mu_volM	52.63	-0.168	-7.20	0.85	0.000
corr_occ1M * sd_volR	28.40	0.479	5.32	1.61	0.000
corr_occ1M * autocorr_occR	7.58	-1.462	-2.74	0.23	0.006
Constant		-3.194	-20.64		0.000

[a] degrees of freedom = 1 for all Chi-square statistics

[b] denotes the probability that the odds ratio is equal to unity (or that the variable coefficient is zero)

We can interpret the results in Table 2 as follows. The likelihood of an accident occurring is…

- HIGHER with increased flows in all outer lanes, especially flows in the interior lanes;
- HIGHER with increased turbulence in any lane, especially the left lane;
- HIGHER with synchronized cross-lane speeds involving the interior lanes and either the outer or inner lanes;
- HIGHER with systematic changes in density (indicating queue formation and dispersal), especially in the right lane;
- LOWER when right lane density is conforming and interior lane flow is high;
- HIGHER when right lane density is conforming and right lane flow is high; and
- LOWER if right lane density is conforming and turbulence is high.

The most important variable in explaining the likelihood of an accident occurring according to the overall Chi-square likelihood ratio statistics, is the correlation of volume to occupancy ratios across the left and interior lanes. We interpret this to mean that the road is operating under particularly dangerous stop-and-go conditions when speeds in the outer lanes are highly synchronized.

4.2 Accident Severity

The second model is designed to explain how accident severity is related to traffic flow conditions. About one-quarter of all accidents in our case study lead to an injury, the rest are property damage only (PDO). Results of a binomial logit regression model for severity are listed in Table 3. The dependent variable is encoded 1 for injury and 0 for PDO, so that a positive coefficient indicates that injury accidents are more likely for higher levels of the independent variable. In Table 3 and subsequent tables, we only list the odds ratios and their significance levels for each traffic flow variable and for each event. The coefficients are easily calculable as the natural logarithms of these odds ratios, and the t-statistics are only important in determining the listed significance levels. As in the case of the first model (Table 2), the goodness-of-fit statistics for the severity model of Table 3 are convincing. The Nagelkerke Pseudo-R^2 is 0.263. The set of eight significant variables and their interactions is slightly different than that of the previous model.

We can interpret the results in Table 3 as follows.

- All Accidents, particularly injury accidents, are MORE likely with increased flow in the outer lanes.
- Injury accidents are MORE likely with increased turbulence in terms of right-lane density.
- PDO accidents, and, to a lesser extent injury accidents, are MORE likely with increased turbulence in right lane speeds.

- Injury accidents, and, to a lesser PDO accidents, are MORE likely with increased turbulence in interior lane speeds.
- PDO accidents in particular are MORE likely when speeds in the interior lanes are more synchronized with speeds in the outer and inner lanes.
- PDO accidents are MORE likely when there are systematic changes in density in the right lane.
- All types of accidents are LESS likely when right lane density is conforming AND there is high interior lane flow.
- All types of accidents are MORE likely when right lane density is conforming AND there is heavy right lane flow.
- All types of accidents are LESS likely when right lane density is conforming when there is turbulence.

Table 3 Multinomial Logit Model of Accident Severity as a Function of Traffic Flow Variables (Reference category = no accident)

Explanatory variable	Chi-square (*dof* = 2)	Property damage only		Injury or fatality	
		Odds ratio	*p* [a]	Odds ratio	*p* [a]
mu_vol1	9.09	**1.04**	**0.013**	**1.87**	**0.014**
mu_volM	28.06	**1.11**	**0.000**	**3.29**	**0.004**
CV_occ1	6.16	0.90	0.649	**4.31**	**0.000**
CV_volocc1	16.58	**3.52**	**0.000**	**1.78**	**0.030**
CV_voloccM	9.14	**2.69**	**0.002**	**4.79**	**0.001**
corr_volocc1M	97.39	**4.83**	**0.000**	**1.05**	**0.030**
corr_voloccMR	19.98	**2.20**	**0.000**	**1.09**	**0.005**
autocorr_occR	27.04	**4.72**	**0.000**	1.00	0.932
corr_occ1M * mu_volM	44.81	**0.85**	**0.000**	**0.84**	**0.000**
corr_occ1M * sd_volR	25.50	**1.61**	**0.000**	**1.67**	**0.000**
corr_occ1M * autocorr_occR	7.17	**0.27**	**0.021**	**0.20**	**0.040**

[a] denotes the probability that the odds ratio is equal to unity (or that the variable coefficient is zero)

As in the case of the likelihood of any accident, the most important variable in explaining the probabilities of injury and property damage accidents is the correlation of speeds across the left and interior lanes. When speeds in the outer lanes are highly synchronized, this indicates that the road is operating under stop-and-go conditions that lead to a heightened level of property damage accidents. The likelihood of injury accidents is also elevated under such conditions, but not nearly as much as the level of property damage accidents. This indicates that the collisions are likely to occur at lower speeds of impact.

Another traffic flow variable that substantially raises the odds of a property damage accident is the autocorrelation of occupancy in the right lane. The interpretation of this is that systematic changes in right-lane density, which might be due to queues formed at egress and ingress points, is an accident precursor. Another precursor is variation in left lane speeds.

The two major precursors of injury accidents are: (1) variation in the volume to occupancy ratio in the interior lanes (and, to a lesser degree, in the left lane), and (2) variation in occupancy in the left lane. These effects indicate that two precursors to injury accidents are (1) turbulence in terms of interior and outer lane speeds and (2) turbulence in terms of left lane density. Finally, another precursor of injury accidents is simply high interior lane volumes.

4.3 Number of Involved Vehicles

The majority of accidents in our sample – 59% – were two-vehicle accidents. About 13% of the accidents in our sample were single-vehicle accidents; these are usually run-off road accidents involving collisions with fixed objects or overturns. Accidents involving three-or-more vehicles made up 28% of the sample. The multinomial model that explains the number of involved vehicles as a function of the traffic flow variables is listed in Table 4. There are thirteen significant variables, plus two interaction terms, and the Nagelkerke Pseudo-R^2 for this model is 0.290, indicating that the number of involved vehicles is easier to predict than accident severity.

Table 4 Multinomial Logit Model of Number of Involved Vehicles as a Function of Traffic Flow Variables (reference category = no accident)

Explanatory variable	Chi-square (*dof* = 3)	Single vehicle		2 vehicles		3+ vehicles	
		Odds ratio	*p* [a]	Odds ratio	*p* [a]	Odds ratio	*p* [a]
mu_vol1	10.26	1.00	0.940	**1.05**	**0.013**	**1.06**	**0.013**
mu_volM	26.66	1.04	0.371	**1.10**	**0.000**	**1.15**	**0.000**
mu_volR	8.09	0.95	0.111	1.02	0.226	**1.05**	**0.040**
sd_volR	3.35	1.10	0.725	0.79	0.088	0.98	0.909
CV_occ1	12.82	**2.31**	**0.004**	**0.57**	**0.032**	0.99	0.973
CV_volocc1	37.70	0.60	0.485	**7.41**	**0.000**	1.47	0.441

Explanatory variable	Chi-square (*dof* = 3)	Single vehicle		2 vehicles		3+ vehicles	
		Odds ratio	*p* [a]	Odds ratio	*p* [a]	Odds ratio	*p* [a]
CV_voloccR	9.31	0.87	0.811	**1.90**	**0.041**	**3.14**	**0.005**
corr_volMR	12.86	1.02	0.953	1.18	0.418	**0.44**	**0.002**
corr_occ1M	2.67	0.92	0.934	0.48	0.274	2.56	0.345
corr_volocc1M	112.36	1.37	0.341	**5.01**	**0.000**	**9.18**	**0.000**
corr_voloccMR	15.14	1.54	0.228	**1.84**	**0.003**	**2.28**	**0.003**
autocorr_vol1	11.11	0.95	0.892	0.86	0.456	**2.16**	**0.004**
autocorr_occR	25.27	1.36	0.409	**2.58**	**0.000**	**2.20**	**0.003**
corr_occ1M * mu_volM	35.12	0.99	0.867	**0.83**	**0.000**	**0.74**	**0.000**
corr_occ1M * sd_volR	10.28	1.60	0.321	**2.19**	**0.002**	1.55	0.177

[a] *p* denotes the probability that the odds ratio is equal to unity (that the variable coefficient is zero)

We can interpret the results of the involved vehicles model as follows:

- Multiple-vehicle accidents are MORE likely with increased flow in all lanes, particularly flows in the interior lanes.
- Single-vehicle accidents are MORE likely, and 2-vehicle accident LESS likely, with increased right lane density turbulence.
- Two-vehicle accidents are MORE likely with increased speed turbulence in the left lane, and, to a lesser degree, the right lane.
- Three-or-more-vehicle accidents are MORE likely with increased speed turbulence in the right lane.
- Three-or-more-vehicle accidents are LESS likely with synchronization of right and left lane flows.
- Multiple-vehicle accidents are MORE likely with cross-lane synchronization of speeds involving the interior lanes.
- Three-or-more-vehicle accidents are MORE likely when there are systematic changes in volume in the left lane.
- All multiple-vehicle accidents are MORE likely with systematic changes in the density of the right lane.
- All multi-vehicle accidents are LESS likely when right lane density is conforming AND there is high interior lane flow.
- Two-vehicle accidents are MORE likely when right lane density is conforming AND there is heavy right lane flow.

Once again, as in the previous models, the most important variable is the correlation of speeds across the left and interior lanes. When changes in speeds in the outer lanes are highly synchronized, this indicates that the road is operating under stop-and-go conditions that lead to heightened levels of multiple-vehicle accidents. The likelihood of single-vehicle accidents is also elevated under such conditions, but not nearly as much as the level of three-or-more-vehicle accidents.

Variation in occupancy in the left lane, the same variable that is positively related to the likelihood of an injury accident, is the primary predictor of the likelihood of a single-vehicle accident, as opposed to a two-vehicle accident. The variance of the left-lane ratio of volume to occupancy, indicating variation in left-lane speeds, is a primary precursor of two-vehicle accidents. The variance of the right-lane ratio of volume to occupancy, indicating variation in right-lane speeds, is a primary precursor of large scale, three-or-more-vehicle accidents, as well as two-vehicle accidents.

4.4 Collision Location

The breakdown of accidents by collision location was: 18% off road, 28% left lane, 36% interior lanes, and 17% right lane. The multinomial model that explains collision location as a function of the traffic flow variables is listed in Table 5. Here there are fourteen significant variables, plus three interaction terms, the most explanatory variables of any of our models. The Nagelkerke Pseudo-R^2 is 0.278, indicating that location is easier to predict than severity, but harder to predict than the number of involved vehicles.

Table 5 Multinomial Logit Model of Collision Location as a Function of Statistically Significant Traffic Flow Variables (reference category = no accident)

Explanatory variable	Chi-square (*dof* = 4)	Off road		Left lane		Interior lanes		Right lane	
		Odds ratio	*p* [a]	Odds ratio	*p* [a]	Odds ratio	*p* [a]	Odds ratio	*p* [a]
mu_vol1	18.93	0.97	0.349	**1.10**	**0.000**	**1.05**	**0.023**	1.04	0.174
mu_volM	34.84	**1.10**	**0.021**	**1.09**	**0.019**	**1.13**	**0.000**	**1.17**	**0.000**
sd_vol1	14.80	**1.21**	**0.032**	**1.20**	**0.016**	0.97	0.676	0.87	0.136
sd_volR	6.24	0.78	0.249	**0.68**	**0.050**	0.92	0.590	0.74	0.147
cv_volocc1	15.59	1.71	0.386	**4.20**	**0.001**	**3.88**	**0.000**	2.08	0.229
cv_voloccM	14.18	0.52	0.371	**3.92**	**0.007**	**2.95**	**0.004**	2.55	0.100
corr_vol1R	22.67	0.69	0.273	**0.39**	**0.001**	0.83	0.457	**2.39**	**0.013**
corr_occ1M	1.86	0.88	0.897	0.54	0.521	0.94	0.936	0.26	0.200
corr_occMR	18.45	0.88	0.716	**2.37**	**0.009**	**0.58**	**0.038**	0.48	0.050
corr_volocc1M	98.99	**2.74**	**0.001**	**13.82**	**0.000**	**3.46**	**0.000**	**3.23**	**0.001**
corr_volocc1R	13.10	1.65	0.138	**0.45**	**0.020**	1.59	0.083	1.42	0.357
corr_voloccMR	19.56	1.47	0.268	**2.37**	**0.015**	**1.99**	**0.011**	**3.59**	**0.001**

Explanatory variable	Chi-square (*dof* = 4)	Off road		Left lane		Interior lanes		Right lane	
		Odds ratio	p [a]	Odds ratio	p [a]	Odds ratio	p [a]	Odds ratio	p [a]
autocorr_vol1	14.27	0.98	0.958	**2.34**	**0.002**	0.88	0.598	0.64	0.176
autocorr_occR	27.41	**3.99**	**0.018**	**5.62**	**0.000**	**5.50**	**0.000**	**3.03**	**0.043**
corr_occ1M * mu_volM	35.31	0.87	0.067	**0.83**	**0.003**	**0.77**	**0.000**	**0.79**	**0.001**
corr_occ1M * sd_volR	18.45	1.88	0.128	**2.15**	**0.029**	**2.52**	**0.001**	**3.27**	**0.003**
corr_occ1M * autocorr_occR	10.43	0.34	0.289	**0.12**	**0.008**	**0.18**	**0.016**	0.71	0.711

[a] *p* denotes the probability that the odds ratio is equal to unity (that the variable coefficient is zero)

We can interpret the results of the collision location model as follows:

- Left and interior lane collisions are MORE likely with increased flows in those lanes.
- All types of accidents are MORE likely with increased flow in interior lanes.
- Off-road and left lane collisions MORE likely with higher left lane speed variance.
- Left lane collisions are LESS likely with higher right lane speed variance.
- Likelihoods of left- and interior-lane collisions are HIGHER with more turbulence in either of those lanes.
- Likelihood of a right lane collisions is HIGHER, and that of a left lane is LOWER when right and left lane flows are synchronized.
- All types of accidents are MORE likely with simultaneous queueing in interior and left lanes.
- Likelihood of a left lane accident is LOWER with simultaneous queueing in right and left lanes.
- On-road collisions are MORE likely when there is simultaneous queueing in interior and right lanes.
- left lane collisions MORE likely with left lane queueing.
- All types of accidents MORE likely higher with queueing in right lane.
- On-road collisions LESS likely when right lane density is conforming AND there is high interior lane flow.
- On-road collisions MORE likely when right lane density is conforming AND there is heavy right lane flow.

- Left and interior lane collisions are LESS likely when right lane density is conforming when there is turbulence.

As expected from the previous models, the most important variable in explaining accident probabilities by collision location is the correlation of speeds across the left and interior lanes. Here we can see that this variable, indicating synchronized speed changes in the outer lanes, is primarily a precursor of left lane accidents. The correlation of volume in the two extreme (right and left) lanes is a precursor of right-lane collisions, as opposed to collisions occurring anywhere else.

5 IMPLEMENTATION

The estimation of the safety models is the first step towards a safety tool. The next step is to implement the models using real data. The goals for this implementation were:

1. To discover if it is feasible with current hardware and software to process all District 12 mainline loops every 30 seconds.
2. To predict relative odds of different accidents at each detector with real data and see if the predictions made sense.
3. Over time, validate the models

The implementation has two main components. First, the raw data must be read from the Testbed Oracle database, processed, and then fed into the safety prediction models. This component is always on, processing and storing safety predictions. The second component is a web-based visualization service that enables the inspection of the safety predictions, and provides the ability to download data for further processing offline. This second component runs on demand, but (as with any web service) must accommodate multiple simultaneous connections. This implementation strategy is diagrammed in Figure 14. Raw data from the VDS loops flows into the Oracle database over the Testbed Intertie to Caltrans D12. The first component of the implementation processes the data from Oracle and saves the predictions. The second component makes the predictions available on the web.

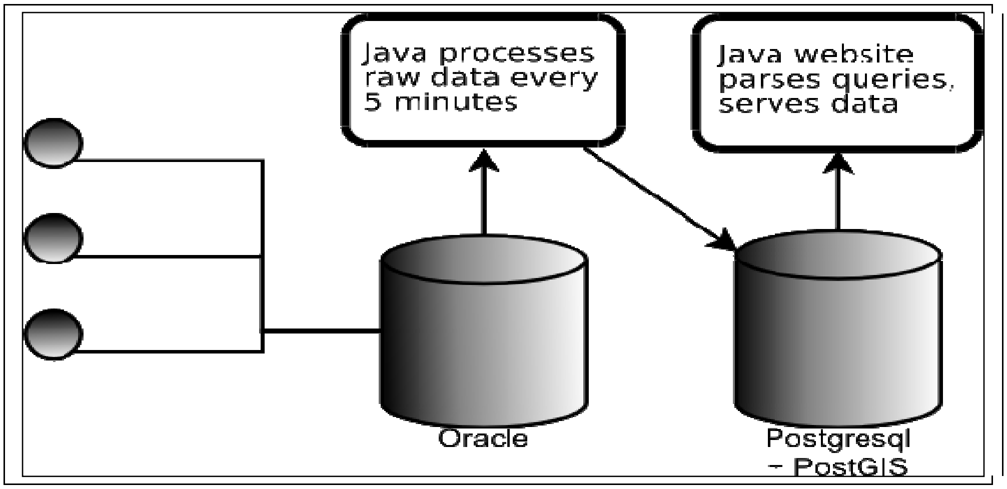

Java processes raw data every 5 minutes
Java website parses queries, serves data
Oracle
Postgresql – PostGIS

Figure 14 Data flow for the safety prediction implementation.

In the implementation discussion that follows, please note that when referring to specific Java classes the complete classpath is given the first time. While this is not visually appealing, it is done to enable developers to find the correct packages and make changes and enhancements as necessary.

5.1 Getting Data from Oracle

Predicting the odds of the different accident types requires as input the raw volume and occupancy measurements for a VDS site for 20 minutes. The required data streams continuously from District 12 into the Testbed ATMS Oracle database. This section discusses how the data are retrieved from the database.

The simplest way to access that data is to set up a program to repeatedly find and retrieve the latest data from the database. This is inefficient, of course. In a perfect world the safety odds calculator would get the same feed as the Oracle database, since it uses the same data. Alternately, one might set up some fancy triggers in the Oracle database to run the safety prediction code whenever any new data is stored. However, those and other "efficient" approaches take time and effort to implement, whereas a simple query for the latest data is easy to implement and easily transferable to other installations. For this reason, the implementation uses the constant query approach.

The Oracle database has a table called HISTORY.RECENT that contains the last 24 hours of observations for every VDS site that is stored in the database. There is also a table called TMCDBA.EQUIPVDSACTIVE that describes the physical details of each VDS, such as the freeway, the number of lanes at the site, the postmile, etc. These two tables have been translated into Java classes with the help of the Torque (The Apache DB Project, 2006) database access library.

Because the safety odds models were only estimated with data from specific freeways, and because they only apply to mainline sections with 3 lanes or more, the Oracle query selects less than the full set of available VDS data. Specifically, a typical query is shown in Figure 15. A careful examination of that query will show that a 25 minute period is being selected rather than a 20 minute period. This will be explained in the discussion of data processing.

Each time the Oracle database query from Figure 15 is run, the database might return anywhere from 11,000 to 14,000 records. Each record is used to populate a list of `Recent` objects that bind each of the selected columns from the database to Java methods for getting those attributes. This list of raw data is the starting point for the data processing and prediction steps that will be discussed in the following sections.

```
SELECT to_char(sample_time, 'yyyy-mm-dd hh24:mi:ss') as SAMPLE_TIME,
HISTORY.RECENT.VDS_ID, LOOP_COUNT, LANE_COUNT, LOOP1_VOL,
LOOP1_OCC, LOOP1_STATUS, LOOP2_VOL, LOOP2_OCC, LOOP2_STATUS,
LOOP3_VOL, LOOP3_OCC, LOOP3_STATUS, LOOP4_VOL, LOOP4_OCC,
LOOP4_STATUS, LOOP5_VOL, LOOP5_OCC, LOOP5_STATUS, LOOP6_VOL,
LOOP6_OCC, LOOP6_STATUS, LOOP7_VOL, LOOP7_OCC, LOOP7_STATUS,
LOOP8_VOL, LOOP8_OCC, LOOP8_STATUS, LOOP9_VOL, LOOP9_OCC,
LOOP9_STATUS, LOOP10_VOL, LOOP10_OCC, LOOP10_STATUS
FROM HISTORY.RECENT
 JOIN TMCDBA.EQUIPVDSACTIVE ON
 (TMCDBA.EQUIPVDSACTIVE.VDS_ID=HISTORY.RECENT.VDS_ID)
 WHERE
    sample_time>sysdate - numtodsinterval(25, 'minute')
    AND TMCDBA.EQUIPVDSACTIVE.VDS_TYPE='ML'
    AND TMCDBA.EQUIPVDSACTIVE.NUM_OF_LOOPS>2
    AND( (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=5 AND
    TMCDBA.EQUIPVDSACTIVE.DIRECTION='N' AND
    TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 0.0 AND 44.0)
    OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=5 AND
    TMCDBA.EQUIPVDSACTIVE.DIRECTION='S' AND
    TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 0.0 AND 44.0)
    OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=22 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='E' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 2.879 AND 12.71)
       OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=22 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='W' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 0.659 AND 12.011)
       OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=55 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='N' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 2.769 AND 17.121 )
       OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=55 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='S' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 2.769 AND 16.711)
       OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=57 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='N' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 11.09 AND 11.221)
       OR ( TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=91 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='E' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 0 AND 18.441)
       OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=91 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='W' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 0 AND 18.441)
       OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=405 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='N' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 0 AND 24.121)
       OR (TMCDBA.EQUIPVDSACTIVE.FREEWAY_ID=405 AND
       TMCDBA.EQUIPVDSACTIVE.DIRECTION='S' AND
       TMCDBA.EQUIPVDSACTIVE.CAL_POSTMILE BETWEEN 0 AND 24.121)
    ) ORDER BY HISTORY.RECENT.SAMPLE_TIME
```

Figure 15 A typical Oracle database query to select the most recent VDS data

5.2 Sorting through the 13,000 VDS Observation Records

The raw observation records found and retrieved from the Oracle database have to be converted into organized matrices of data to be useful. This step is handled in the `edu.uci.its.safety.commands.ProcessDeepFetchAll` class. For each VDS, three matrices are created: one for volume, one for occupancy, and one to keep track of missing data. The missing matrix is made possible by listing all of the time-stamps for all of the observations retrieved from the database. If a 30-second time step is missing (no record in the database) then a non-zero entry is put in the missing matrix. Using the variables that are in the code, each matrix is p rows by n columns, where p is the number of lanes for the VDS site, and n is the number of observations of those lanes, specifically the number of 30 second periods from the earliest time to the latest time retrieved from the database.

5.3 Cleaning the Data

In using the data, a persistent problem was discovered in the raw loop data stream. Following no clear pattern and apparently affecting all VDS sites, there are cases in which consecutive observations are identical, with the third observation being approximately twice what it should be. This problem is illustrated in Table 6. Each of these entries has a loop status of 0 (not shown in table), which means that the data should be good. The last column in the table indicates our estimation of whether the data is in error or not. Note that five of the rows are exact duplicates of the preceding row. This duplication is exact even in the binary data received directly from the Caltrans data feed. This means that the exact same chunk of data is being reported for two different timestamps for the same VDS. This is statistically impossible. Furthermore, as shown in Table 6, the very next record following each duplicate record appears to have twice the volume and twice the occupancy that it should have.

Our explanation for this is as follows. It appears that the VDS might miss a poll for a 30-second period. Loop detectors are essentially accumulators. The detector accumulates counts of vehicles, and adds up the amount of time the loop is occupied. When a poll is received, the accumulator reports its data and resets its counters. When a poll is missed, on the other hand, the front end processor reports the old data at the new time stamp, and the VDS accumulator doesn't ever reset its counters. On the next cycle, the poll is successful, and the accumulator sends back the data for two consecutive 30-second periods. However, the front end processor assumes—incorrectly—that the poll is returning data for 30 seconds. Thus the volume is double what is should be (60 seconds of counts is approximately twice a 30 second count) and the occupancy is also double what is should be (occupancy period in seconds divided by 30 seconds rather than 60 seconds). The example shown in Table 6 is even more interesting. In this time period, the VDS appears to have missed multiple polls in succession, with only every other poll succeeding. In this case, the apparently doubled values (60 seconds of data) are reported for consecutive time steps. These are identified in the error column as being both doubled and duplicate. That is, if every other poll is missed, then the data will remain twice as high as it should be. In order to recover from the doubled state, there must be two consecutive successful polls.

Table 6 An example of bad data in the Oracle database, for loop id 1201254 on Nov 20, 2006. All loops have a status of 0 (not shown), which means good data.

	LOOP1	LOOP1	LOOP2	LOOP2	LOOP3	LOOP3	LOOP4	LOOP4	
TIME	**VOL**	**OCC**	**VOL**	**OCC**	**VOL**	**OCC**	**VOL**	**OCC**	**error?**
9:25:30	18	0.1278	21	0.1522	17	0.1367	17	0.1267	good
9:26:00	19	0.1422	19	0.1333	14	0.16	14	0.1511	good
9:26:30	16	0.1244	19	0.1444	18	0.18	19	0.1756	good
9:27:00	18	0.13	19	0.1433	17	0.14	15	0.1278	good
9:27:30	22	0.1611	18	0.1367	13	0.1489	13	0.15	good
9:28:00	17	0.1222	16	0.12	14	0.1422	14	0.1411	good
9:28:30	17	0.1222	16	0.12	14	0.1422	14	0.1411	duplicate
9:29:00	34	0.2589	33	0.2633	28	0.27	26	0.2667	doubled
9:29:30	34	0.2589	33	0.2633	28	0.27	26	0.2667	duplicate, doubled
9:30:00	44	0.3489	35	0.2967	25	0.2967	25	0.2867	doubled
9:30:30	22	0.17	21	0.1611	13	0.1533	14	0.1567	good
9:31:00	18	0.1344	22	0.1833	22	0.2	21	0.2044	good
9:31:30	16	0.1233	17	0.1322	16	0.1467	17	0.1522	good
9:32:00	16	0.1233	17	0.1322	16	0.1467	17	0.1522	duplicate
9:32:30	43	0.3644	39	0.3244	32	0.3022	31	0.2811	doubled
9:33:00	43	0.3644	39	0.3244	32	0.3022	31	0.2811	duplicate, doubled
9:33:30	38	0.3011	37	0.3044	31	0.3389	31	0.3311	doubled
9:34:00	38	0.3011	37	0.3044	31	0.3389	31	0.3311	duplicate, doubled
9:34:30	38	0.3	31	0.2567	28	0.3	28	0.2878	doubled

Whether our theory about the cause is correct or not, the fix is obvious. We process through all of the data in each observation matrix, and mark as duplicates each row that is a duplicate of the prior time step. After this first pass, all of the records following the duplicate records are marked as being doubled (two time periods in one). Finally, all duplicate and doubled observations are recorded as missing data by setting the corresponding cells in the missing matrix to 1.

5.4 Iterate Over the Raw Data

Once the raw observations have been found, retrieved, stored in matrices, and cleaned of the duplicates and doubles as explained in the preceding section, then the command class `edu.uci.its.safety.commands.IterateOverData` is executed to iterate over the data (as the name suggests).

There are two loops that are executed in the class. The first is to loop over each VDS site. The second is to iterate over consecutive 20 minute periods. Because the data are saved to the database periodically, and because the data query is run periodically, a 25 minute time period is retrieved from the database, containing up to 50 30-second observations. These are sorted by time and VDS id, and a 20 minute window (at most 40 observations) is constructed and run over the data. This is the main job of the `IterateOverData` command.

Once the correct subset of the data matrices are loaded into the context, the chain of commands making up the data processing command sequence is loaded up and executed. It might be the case that the time period has already been stored in the database, so the first command in the chain is to check the database. Once that check has passed, the other commands in the chain are executed in turn, as described in the following sections. When all of the commands have been run, or when the command chain has been aborted (due to known error conditions), the context is cleared, and the next 20 minute window is loaded and the process repeated.

5.5 Computing the Input Variables

The first step is to compute the model input variables that each of the odds prediction models require. A chain of operations has been set up to convert the raw data into usable input variables. The chain of operations is as follows:

1. Pick the best lane to use as the “middle” lane
2. Initialize the processing of raw volume
3. Create variables (based on the raw volume)
4. Initialize the processing of raw occupancy
5. Create variables (based on the raw occupancy)
6. Initialize the processing of raw volume over occupancy
7. Create variables (based on volume over occupancy)

Each of the “create variables” steps—steps 3, 5, and 7 above—runs the identical piece of Java code, with only the raw data being changed. The three initialization steps allow any other checks on the raw data to be performed as necessary. These steps will be described in turn in the following subsections.

5.5.1 Pick a middle lane

Because many of the VDS locations have more than three lanes, it is necessary to choose the best middle lane for use in computing the input variables for the safety calculations. The `edu.uci.its.safety.commands.PickMiddleLane` class follows the following logic. First, if the loop site has less than 3 lanes, stop processing and return an error code. If the site has exactly 3 lanes, then assign the number two lane to the middle lane variable and return. If there are more than three lanes then the program actually has to do some work. There are two criteria for selecting the best middle lane. First, the ideal lane will have the least amount of missing observations. Second, the ideal lane will be closest to the actual middle of the roadway.

To satisfy the first constraint, each non-zero column of the raw missing matrix is counted and the result stored for the corresponding row. Since the safety prediction models require a minimum number of valid observations, the result is used to determine if further processing of the loop site is warranted. For this the first and last values of the tally (corresponding to the left and right lanes of the site, which must be included in any calculation) are checked to ensure that the minimum number of good observations (30) is met or exceeded in the 20 minute period. If this condition is not met, then an error is returned and further processing is halted. The remaining rows of counts (from lane 2 to lane n-1) represent the candidate middle lanes. The minimum value of these is determined, and if that value exceeds the allowed number of missing observations, then none of the middle lanes is suitable and so an error is returned and further processing is halted.

Once the minimum number of missing observations has been determined for the middle lanes, the lanes are searched in a specific order to choose the best middle lane that has the minimum number of missing observations. The sorting rule is to pick the middle-most lane, and settle ties by choosing the left-most lane. For example, suppose there are 8 lanes at a site, with loops numbered 0, 1, 2, 3, 4, 5, 6, and 7. Loops 0 and 7 are kept, as they represent the left and right lanes, leaving lanes 1, 2, 3, 4, 5, and 6 as middle lane candidates. To determine the middle lane, in this case there is no lane 3.5, so the best choice is lane 3 (the left-most lane rule) followed by lane 4. The complete ordering is 3, 4, 2, 5, 1, and 6. Each of these lanes is examined in turn, and the first one containing the lowest number of missing entries is used as the middle lane.

The `PickMiddleLane` command finishes by setting the volume, occupancy, and missing matrices to correspond to the left, right, and the selected middle lane, dropping all other lanes from the matrices. After the command has completed, either an error has been returned, or the volume, occupancy, and missing matrices are each 3 rows by 40 columns.

5.5.2 Initialize the volume calculations

The variable processing loop runs its routines on a matrix of data. In order to tell it which matrix to work on, an initialization step is required. The volume data processing step, coded in the `edu.uci.its.safety.commands.InitializeVolMatrix` class, is the simplest of these. Here all that is done is that the active matrix variable is set to equal the volume matrix.

5.5.3 Initialize the occupancy calculations

After the raw volume data has been processed, the raw occupancy data is initialized by the `edu.uci.its.safety.commands.InitializeOccMatrix` command class. This command does two things. First it checks the raw volume data (which has been processed already) to decide whether the minimum volume requirements have been met. The computed average volumes are examined to make sure that each lane has an average volume of at least 0.5 vehicles for the 20 minute period. If the average volume is lower than that level, then the estimated safety models do not apply, and so further processing is abandoned and an error condition is returned. If the minimum average volume conditions are met, then the second task of the command is to set the occupancy matrix to be the data processing matrix for the “create variables” step.

5.5.4 Initialize the volume over occupancy calculations

The `edu.uci.its.safety.commands.InitializeVolOccMatrix` class is the final initialization class. Since volume/occupancy is not a standard output of the Oracle database, this matrix (`volocc`) must be computed here before it can be manipulated in the data processing command. The `InitializeVolOccMatrix` command creates the `volocc` matrix by dividing each `volume` matrix entry by the corresponding `occupancy` matrix entry. If the occupancy value is zero, then the corresponding cell in the `volocc` matrix is set to zero, and a flag is set in the `missing` matrix to indicate that this value should be considered missing. Once again, there are a minimum number of valid observations required for the safety models. If adding to the `missing` matrix increases the number of missing beyond the maximum allowed, then further processing of this VDS site at the current time is aborted with an appropriate error condition.

5.5.5 Create input variables

The `volume`, `occupancy`, and `volocc` matrices are all processed by the command class `edu.uci.its.safety.commands.CreateVariableStats`. This command creates the means, standard deviations, variance/covariance matrix, the sample correlation matrix, the sample autocorrelation (a one time-step lag function), and the coefficients of variation. While not all of these variables are used for each of volume, occupancy, and volume/occupancy, the operations are performed using a matrix library and are very fast; the extra time spent computing unused variables is negligible.

All of the statistical functions used by the `CreateVariableStats` class are coded in the `edu.uci.its.spatialvds.utis.stat.Descriptive` library class. This class leverages the open source Matrix Toolkits for Java (MTJ) package wherever possible, and uses custom operations on MTJ matrices when necessary.

5.6 Feeding the Input Variables to the Safety Models

After the `volume`, `occupancy`, and `volocc` matrices have been processed to generate the necessary means, variances, covariances, autocorrelations, correlations, and coefficients of variation, the next step is to use these input variables to compute a prediction for each of the estimated safety models. This step is handled by the command class `edu.uci.its.safety.commands.ApplyModels`.

`ApplyModels` first loads up all of the computed input variables, and then uses them to create an instance of `edu.uci.its.safety.utils.modeling.LoopStats`. The `LoopStats` object handles whether or not to ignore variables, and also contains functions that help to store the calculated variables in the PostgreSQL database for later retrieval. The `LoopStats` object also stores the raw volume and occupancy data, specifically to allow saving those values in the database.

There are currently 10 models coded up into Java classes. These are:

- the `AnyAccident` model;
- models of severity: `Injury` and `PDO` (property damage only);
- models of vehicle involvement: `OneVeh`, `TwoVeh`, and `ThreePlusVeh`;
- and models of location: `OffRoad`, `RightLane`, `InteriorLanes`, and `LeftLane`

These models all extend a `BaseAccidentModel` class that provides a common interface and set of operations. Each specialization of `BaseAccidentModel` is named appropriately and saved in the `edu.uci.its.safety.utils.modeling` package. Each specific model class has a distinct model `NAME`, and has a uniquely specified parameters list which contains the specific parameter values estimated for each model. Creating a new model class can be done fairly easily by copying one of the existing classes and altering the parameter list as necessary. The only difficulty is in making sure that the new model class is given a unique `NAME`, so that no conflicts arise in the database. Once all 10 model objects have been instantiated, each is passed the `LoopStats` object in turn, and the output of the model is stored in a list and saved to the global context.

5.7 Saving the Model Output

The last step in processing the safety predictions for a given 20 minute period for a specific VDS location is to save the output of the models to the database. This task is performed by the `edu.uci.its.safety.commands.SaveModelResults` command class. This command wraps up several database actions into a single transaction; the failure of any part of the transaction will cause a graceful rollback of the database status, rather than leaving the database in an inconsistent state. The database transaction first saves the input variables and the raw variables into the `loop_stats` table, and then saves the various model results in the `accident_risk_results` table. The new entries in these tables are joined to each other using a common id field (`stats_ids.id`) and are joined to the relatively static VDS data stored in other tables via the `vdsid` field. Various java classes are invoked in the process of saving data to the database. These classes belong to the `edu.uci.its.spatialvds.schema` package, except as noted in the following discussion.

5.8 Storing Safety Data: the **spatialvds** *Database Schema*

Prior to the implementation of the safety prediction models, there existed a geo-enabled database containing descriptions of the VDS sites, called `spatialvds`. The term "geo-enabled" means that the locations of each of the VDS sites had latitude and longitude coordinates; that the database uses spatial indices to greatly speed spatial searches; and has embedded functions for a standard set of spatial searches over these indices. For example, one can ask for all loops that are to the left of a particular loop, all loops that are within a 1 mile boundary of a given coordinate, or all loops that are within a specific bounding box such as a map window. Lacking a spatial index, similar operations are very slow. Since Oracle's spatial extension is quite expensive, and since the PostgreSQL spatial extension—PostGIS—is open and free, we chose to use PostgreSQL and PostGIS to implement spatial VDS features for that earlier project.

For this project, we needed a database to be able to store the results of the safety computations. We could have saved the results in Oracle, either in the same database or in a separate database, or we had the option of extending the previous work we did in PostgreSQL. We chose to extend our previous work and to add safety-related tables to the `spatialvds` PostgreSQL database.

There are two things that need to be stored in the database: the predicted odds of different kinds of accidents, and the input variables that allow those odds to be computed. In fact neither of these tables is strictly necessary. It is always possible to go back to the original source (the Oracle ATMS database), pull the required data, and recompute the values. However, it makes no sense to continuously redo the predictions when the computation process takes minutes rather than milliseconds. Because a web-interface was desired, and because we anticipate eventually making the computations open to a large number of potential clients, saving the computed values in a database table trades space (memory and disk space) to gain speed.

The input variables are stored in the `loop_stats` table. The schema for this table is reproduced in Figure 16. For each timestamp (`estimate_time`) the table stores all 31 computed variables that are used by the safety odds prediction models (the 27 listed in Table 1, plus 4 crossed terms). The table also stores the raw volume and occupancy for the left, middle and right lanes for that timestamp. The unique primary key for the table is stored in the `id` column. This column is autogenerated by PostgreSQL. For technical reasons relating to integrating a programming language (Java) and a database, the unique keys are generated in a separate table (`stats_ids`). Note that all of the computed input values are required (have `NOT NULL` constraints, while the raw data columns do not have this constraint. It is quite possible that input values can be computed for a 20 minute period even if some or all of the raw data for the most recent 30-second interval are missing. Finally, a separate join table, vds_stats, exists to allow joining each row of loop_stats to the correct VDS.

The predicted odds are stored in the `accident_risk_results` table. The schema for this table is shown in Figure 17. Once again, the unique primary key column, `id` is generated using the table `stats_ids`. The other columns store the estimate time, the model name, and the computed odds. Finally the column `vds_id` contains the id of the VDS site, allowing cross linkage with other tables in the database that provide more details on the VDS (such as the number of loops, the freeway, the post mile, and the spatial coordinates). Technically, good database design would require that this join information go in a separate table, as was done with the `vds_stats` table. However, including the `vds_id` in the table saves a little bit of time when selecting data spatially, and so the index was included in the table directly.

```
CREATE TABLE loop_stats (
id integer NOT NULL PRIMARY KEY,
estimate_time timestamp without time zone NOT NULL,
cv_occ_1 numeric NOT NULL,
cv_occ_m numeric NOT NULL,
cv_occ_r numeric NOT NULL,
cv_volocc_1 numeric NOT NULL,
cv_volocc_m numeric NOT NULL,
cv_volocc_r numeric NOT NULL,
corr_vol_1m numeric NOT NULL,
corr_vol_1r numeric NOT NULL,
corr_vol_mr numeric NOT NULL,
corr_occ_1m numeric NOT NULL,
corr_occ_1r numeric NOT NULL,
corr_occ_mr numeric NOT NULL,
corr_volocc_1m numeric NOT NULL,
corr_volocc_1r numeric NOT NULL,
corr_volocc_mr numeric NOT NULL,
lag1_vol_1 numeric NOT NULL,
lag1_vol_m numeric NOT NULL,
lag1_vol_r numeric NOT NULL,
lag1_occ_1 numeric NOT NULL,
lag1_occ_m numeric NOT NULL,
lag1_occ_r numeric NOT NULL,
mu_vol_1 numeric NOT NULL,
mu_vol_m numeric NOT NULL,
mu_vol_r numeric NOT NULL,
sd_vol_1 numeric NOT NULL,
sd_vol_m numeric NOT NULL,
sd_vol_r numeric NOT NULL,
corr_occ_1m_x_mu_vol_m numeric NOT NULL,
corr_occ_1m_x_sd_vol_r numeric NOT NULL,
corr_occ_1m_x_lag1_occ_r numeric NOT NULL,
cv_volocc_1_x_corr_volocc_1m numeric NOT NULL,
occ_1 numeric,
occ_m numeric,
occ_r numeric,
vol_1 numeric,
vol_m numeric,
vol_r numeric,
FOREIGN KEY (id) REFERENCES stats_ids(id) ON DELETE CASCADE
);
```

Figure 16 The loop_stats table schema

```
CREATE TABLE accident_risk_results (
id integer NOT NULL PRIMARY KEY,
estimate_time timestamp without time zone NOT NULL,
model_name character varying(100) NOT NULL,
odds numeric NOT NULL,
vds_id integer NOT NULL,
UNIQUE (vds_id,estimate_time,model_name),
FOREIGN KEY (vds_id) REFERENCES vds(id) ON DELETE CASCADE,
FOREIGN KEY (id) REFERENCES stats_ids(id) ON DELETE CASCADE
);
```

Figure 17 The accident_risk_results table schema

5.9 Retrieving safety data

Storing the safety predictions and the associated model inputs in a database means that they are always available for future research. In order to simplify sorting and querying the vast quantities of data being stored, a web interface was constructed (discussed in more detail in subsequent sections). The web interface required three different kinds of queries:

1. A spatial, shallow query, in which all VDS locations inside of a bounding box are selected, along with the most recent (or closest to a time parameter) safety predictions
2. A spatial, detailed historical query, in which all safety predictions and input variables between a start date and an end date are selected, for all VDS locations inside of a bounding box
3. A detailed historical query for a specific VDS

Each of these queries was parameterized and coded up into a specific method in the database access class generated by Torque. A companion command class was written to accept the parameters submitted by the web site, translate them into the correct format for the database query, and then return the results to the web browser. This section documents the three database queries and their implementation in Java. To simplify the discussion somewhat, all of the database access classes are part of the `edu.uci.its.spatialvds.schema` package, unless otherwise specified.

5.9.1 Spatial selection of safety predictions

For a spatial query, the inputs are the parameters describing the data that should be retrieved, and the bounding box of the area of interest over which the query should be run. To query the safety predictions for a specific area for a given moment in time, the following variables need to be specified.

- The `timestamp`: the date and time for which one wishes the predictions to be returned, with a default value of the current time. The closest time not exceeding the specified timestamp is returned.
- The `modelList`: the list of models that are of interest. The default is to select all possible models.

After several trials and empirical timings of the resulting code, the following query strategy was devised. The query itself was split into two separate queries. First the spatial bounding box is used to obtain a list of mainline VDS stations. Then this list is used to sequentially execute a prepared query that finds and retrieves the desired safety odds prediction results. Splitting the query into two distinct queries doesn't necessarily increase the program execution time. A programmatic database query has three steps: connecting to the database, preparing a query (which involves the database engine preparing an optimized query plan), and then executing the query. By using prepared queries, the first two steps are performed only once. In this case, the spatial select planning is done once, and the odds prediction select planning is done once. While there might be a theoretical speed up by combining the two queries into one SQL statement, in practice the difficulty of crafting that single query meant that the more complicated statement actually took longer to prepare and longer to execute. Furthermore, as will be shown, splitting up the query makes it easy to extend the query to other cases, such as fetching predictions for a range of times, or selecting the input data for each prediction.

The spatial query is prepared in the class `GeomPoints4326Peer`. The query uses the PostGIS spatial binary operator `&&`, which is used in the idiom `A && B`. This statement returns true if any part of the bounding box of `A` overlaps the bounding box containing `B`, and false otherwise. In this case the VDS locations are coded as points and so the operation returns only those detectors that are inside the bounding box. The other major limitation to the query is to only return mainline detectors.

The odds forecast results query is prepared in the class `AccidentRiskResultsPeer`. This method is called directly by the methods in `GeomPoints4326Peer` that are performing spatial queries, so its use is transparent to the querying classes. The prepared statement is relatively simple. The model name, estimate time, and computed odds (see Figure 17) are selected conditioned on the `vds_id` equaling an unspecified bind variable, and the timestamp being less than or equal to the maximum timestamp that is less than the specified timestamp. This maximum timestamp is determined using a tricky sub-select statement. During testing it turned out that using the sub-select to determine the correct timestamp was equally as fast as (to within tenths of milliseconds) using a separate query to determine the timestamp and then inserting an equals condition. This was due to the way the database engine caches results and how the query planner uses indexing. In a further effort to streamline the SQL queries, no effort is made to limit the results based on the desired list of models. Instead all models are returned which should be fine as long as there are only tens or even hundreds of kinds of models.

Once the spatial and odds queries are prepared, they are run. The logic to process and store the results is complicated. The goal of the query method is to return a list of objects, each object containing all of the necessary data for a particular VDS detector. The object is an instantiation of the class `VDSAccidentRisk`, which is located in the package `edu.uci.its.spatialvds.beans`. The reason this class is used is two-fold. First, it combines the records from several tables into one Java object, with accessor methods that are customized to how the object will be used by the web-based front end. Second, passing the data access objects generated by Torque to a presentation layer is dangerous. A programming mistake in the presentation of data could possibly overwrite the database. The Java Bean class `VDSAccidentRisk` removes all database set and get methods, thus eliminating this risk. To initialize the data retrieval loop, first a `List` object is created, and a database connection is created. This database connection is used to prepare the spatial and model SQL statements. The two queries are run in a two nested loops.

The outer loop executes the spatial query and stores the result in a `ResultSet`, which is part of the standard Java runtime environment SQL library. Each row in the result set represents a specific mainline VDS detector that lies inside of the specified boundary box. The attributes of this VDS station are used to create the previously mentioned `VDSAccidentRisk` Java Bean.

For each VDS result, the `getVdsid()` method of the current `VDSAccidentRisk` object is used to set the bind variable of the model query. The model query is run, and the results are stored in a second `ResultSet`. Control then passes to the inner loop to process this result set. For the basic spatial query for a single timestamp, this inner loop simply uses the `VDSAccidentRisk` method `putModelOutput` to store the model name, timestamp, and odds prediction in the Java Bean.

When the model data has been processed, the new `VDSAccidentRisk` object is added to the List that will be returned at the end of the query, and the outer loop continues processing the next VDS detector. When the outer loop has completed, the connection is closed, errors are handled, and the list of results is passed back to the calling routine.

5.9.2 Spatial query of a range of safety predictions

Finding and retrieving a range of predictions, from a start time to an end time, is similar to getting data for a single timestamp. There are three differences. First the prepared statement to find the model results uses a SQL `BETWEEN` operator to test if the timestamp of a record falls within the desired range, rather than the sub-select statement. Unlike the single timestamp case, this might return an empty list if there are no suitable records. As with the single timestamp case, the SQL statement contains a bind variable for specifying the `vds_id` to select.

Second, whereas the single timestamp results are destined for a simple graphical interface, when a range of timestamps is requested, this usually indicates that the end-user wants to perform a more complicated analysis. This means that most likely the query will also want the input data stored in the `loop_stats` table (see Figure 16). This requires another prepared SQL statement. This syntax for the statement is generated by the class `LoopStatsPeer`, with logic that loosely parallels that used in `AccidentRiskResultsPeer`. Again, the use of this class to prepare the SQL syntax is done within the `GeomPoints4326Peer` class, and is generally transparent to the calling function.

The third difference from the single timestamp case comes in how the inner loop is handled. (In fact the code is identical, but for the simpler case various if statements mask out the more complicated options.) Instead of there being a single inner loop, there are now two—one for the model odds, and one for the input data. Further, since there are most likely several timestamps for each VDS station, some care must be taken to ensure that the timestamps are sorted and standardized.

If the input data is also being requested, then a “get data too” flag will be set to true by the calling function. This causes the data SQL to be prepared for the specified time period outside of the loop, and for that SQL to be executed inside the loop. Since the select statement is quite a bit longer than for the predicted odds table, the results of the query are passed into the active list of `VDSAccidentRisk objects` by code contained within the `LoopStatsPeer` class. This allows the order of the select and the order of parsing the response to be matched exactly.

It was discovered during testing that occasionally one or more timestamps would be missing from an otherwise continuous range of times. It was decided that it is much easier to catch these cases at the source and insert empty records than to require the user or analyst to check the data for gaps. Therefore one final run through the data was coded up, using the standard Java runtime library class `SortedMap`. This class is a hash map, with the added feature that all of its keys are guaranteed to be sorted. In this case, the values are the `VDSAccidentRisk` data containers, and the keys are the timestamps returned by each container's `getEstimateTime` method. Using the map, the minimum and maximum timestamps are determined, and a while loop is executed from the minimum value to the maximum value, incrementing the index variable by 30 seconds each time (actually the increment is determined by looking up a canonical value in a constants package, so that this can easily be changed in a different deployment). For each timestamp, the sorted map is consulted to see if there is a non-null value. If not, then there is a missing value, and so a new, empty record is created using the current VDS data. In this way, every missing observation explicitly has an empty result for predicted odds and input data, and every 30-second time step from the start to the end of the period will have a corresponding record.

5.9.3 Find a range of safety predictions for a single VDS

Selecting the predicted odds and input variables for a single VDS is largely identical to the spatial case, with the exception that the outer spatial loop is eliminated. Instead the "spatial" query is replaced with a query for a specific, known VDS id. This is the only substantial difference between this query implementation and the spatial queries. The lack of a loop over VDS ids also means that the data input SQL and the model output SQL statements can be specified without a bind variable. To speed implementation, the spatial query method was copied, and the necessary changes made throughout. In the future, a re-factoring of the code should probably combine these two kids of queries to call common methods.

5.10 The Web Server, Command Sequences, and The Chain of Responsibility Design Pattern

The web server is organized around the Model-View-Controller (MVC) paradigm. The Model is the data and all algorithms and logic that manipulate that data. In this case, the Model consists of the database and the classes stored under the two `*.schema.*` packages that have been discussed for accessing the Oracle and the PostgreSQL databases. The View is the user interface. The Controller links requests and commands from the View to the data and methods that make up the Model. There is always a gray area between where the Controller ends and where the Model begins. To facilitate making this distinction, we have taken the approach that the Controller should do as little as possible, and should only serve as a switchbox, routing incoming requests to the correct Model commands, and then piping the responses from those commands to the right display programs in the View. To implement the commands as part of the Model, we have chosen to use the Chain of Responsibility design pattern (as implemented by the commons-chain package in the Apache Foundation's Jakarta Commons library).

The Chain of Responsibility pattern can be understood as a chain of commands, all of which acting on a global context. The theory is that each command in the chain is called in turn. The command performs whatever operations it can on the variables stored in the context, and stores the result of that operation in the context as well. Commands return *true* or *false*. A return value of true aborts the execution of the remainder of the chain, and a return value of false passes control along to the next command in the chain. For example, a chain of responsibility that is designed to retrieve a file might contain one command to look in the user's home directory, another command to look at all directories on the local computer, a third which looks for the file on all networked drives, and a fourth which does a Google search for the filename. The desired filename is stored in the context, the chain's execute method is called, and then (hopefully) the desired file is stored in the context when the chain is finished running. In this case, if any of the commands finds the file, it can store the file in the context and return true, thus skipping the remaining commands.

Another variant of the Chain of Responsibility pattern is to have all of the commands return false if they successfully perform their respective tasks, and to return true (and abort the chain) only if they run into problems. This is how the Chain of Responsibility pattern is used in this web server application. Returning to the file finding example, the chain might consist of a command to search the user's hard drive, followed by a command to read the file into memory and create a Java file object. At the end of this chain, the caller can either expect to get true as the output (perversely meaning that the chain failed), or to get false as the output and find the desired file slotted away in the context, ready to use.

The various command chains that make up the data manipulation sequences are described in the following sections. Each command chain is embedded in simple Controllers in the Struts configuration file, so that incoming requests from the View can access the data stored in the Model. For the most part, the commands expect to be called from a modern web interface, and therefore return their data as JavaScript Object Notation (JSON), presuming that the caller is either JavaScript embedded in the web page, or else some other program that can easily parse JSON.

5.10.1 Find the current odds for a bounding box

The first command chain is designed to provide the view with the information needed to paint colored dots on a map. The dots represent the VDS stations, and are placed on the map by means of the latitude and longitude stored in the PostgreSQL database. It was decided that the colors should represent the relative risk of seeing "any accident" for a particular detector compared to all other detectors at that exact snapshot in time. The View expects colors to range from red to green, with red meaning that the loop's odds of "any accident" is in the highest percentile, while green means that the odds of "any accident" at that detector are among the lowest. One obvious future extension is to allow the user to specify which odds model is used to generate the colors, and whether to use cross-sectional information or longitudinal (historical) information for each loop. To perform this task, a chain of responsibility was set up as follows:

1. Login Check: check if the user is properly logged in to the Testbed, and throw a specialized exception (error) if not.
2. Parse Box: parse the parameters of the web request for the bounding box definition.

3. Parse Timestamp: look for a specific timestamp in the request parameters, and if it isn't there, set the timestamp to now.
4. Parse Model List: look for a list of models to add to the query, or else set the list to all known models.
5. Sort the Model List: because the list of known models is not guaranteed to be in any particular order, make sure it is sorted as desired by this application, grouping like models together.
6. Spatial Query of VDS and Safety Data: call the query described earlier in section 6.9.1.
7. Get the Cross-section Odds for the Timestamp: find and retrieve the mean and standard deviation over all detector stations for each model type for the current timestamp.
8. Set Colors: set the colors for each VDS location. This is done by calling a method in the `VDSAccidentRisk` class (the Java bean class described earlier) with the mean and standard deviation for the different odds. A color is set for each model type, so that the view can set the color as it wants.
9. Convert Results to JavaScript Object Notation (JSON): convert the data stored as Java objects in the context to simpler, serialized versions that conform to the JavaScript Object Notation (JSON) specification. Once again the hard work is handled by a method in the `VDSAccidentRisk` class.

5.10.2 Query the historical odds for a bounding box

The next chain of responsibility Queries a range of data from the database for the selected bounding box. Here the use case is that a researcher has moved the map to a particular place, and wants to dial in on a period of time for some stretch of freeway. The expected inputs are a bounding box, a list of freeways, a starting timestamp, and an ending timestamp. Instead of returning a JSON string containing the data, the chain will generate a comma delimited file with the data, zip it up, and then return a JSON string that describes the location of the file. The view can then display the correct link to the user. The following chain of responsibility handles this task.

1. Login Check: as before.
2. Parse Box: as before.
3. Parse Timestamp: as before, but note that the code will also pick out starting and ending times stored in the request. The usage described above is actually a special case of this more general case.
4. Parse Model List: as before.

5. Sort the Model List: as before.

6. Spatial and Historical Query of VDS and Safety Data: parse the request for the freeway list, if any, and then call the database query described Section 6.9.2.

7. Convert Results to Comma Separated Values (CSV): convert the data stored as a list of Java objects in the context to a list of CSV records, with a single header row describing each column. Once again the work is handled by a method in the `VDSAccidentRisk` class.

8. Create a File: dump the CSV list into a file, zip up the file, and then create a JSON string describing the location of the file.

5.10.3 Query the historical odds for a single VDS station

The last chain of responsibility Queries the historical data for a single VDS station. The anticipated use case is to display detailed information about a single detector. After using the map display and downloading detailed data for a period of time, we decided that it would be more convenient to be able to view periods of time in the web interface prior to downloading a data file. We felt this would improve the analysis of the downloaded data, by providing the analyst a better tool to decide which period of time and which areas to download. In the use case, the analyst would set the time period using the same form inputs used to download data, and then click on a single VDS's icon, thus specifying the VDS id and initiating the query. If no time period was set, the default values are the current time for the end time, and thirty minutes prior for the start time. Again, the response to the web browser is a JSON string describing the data, which is handled by the view as necessary. The chain of responsibility that accomplishes this is as follows.

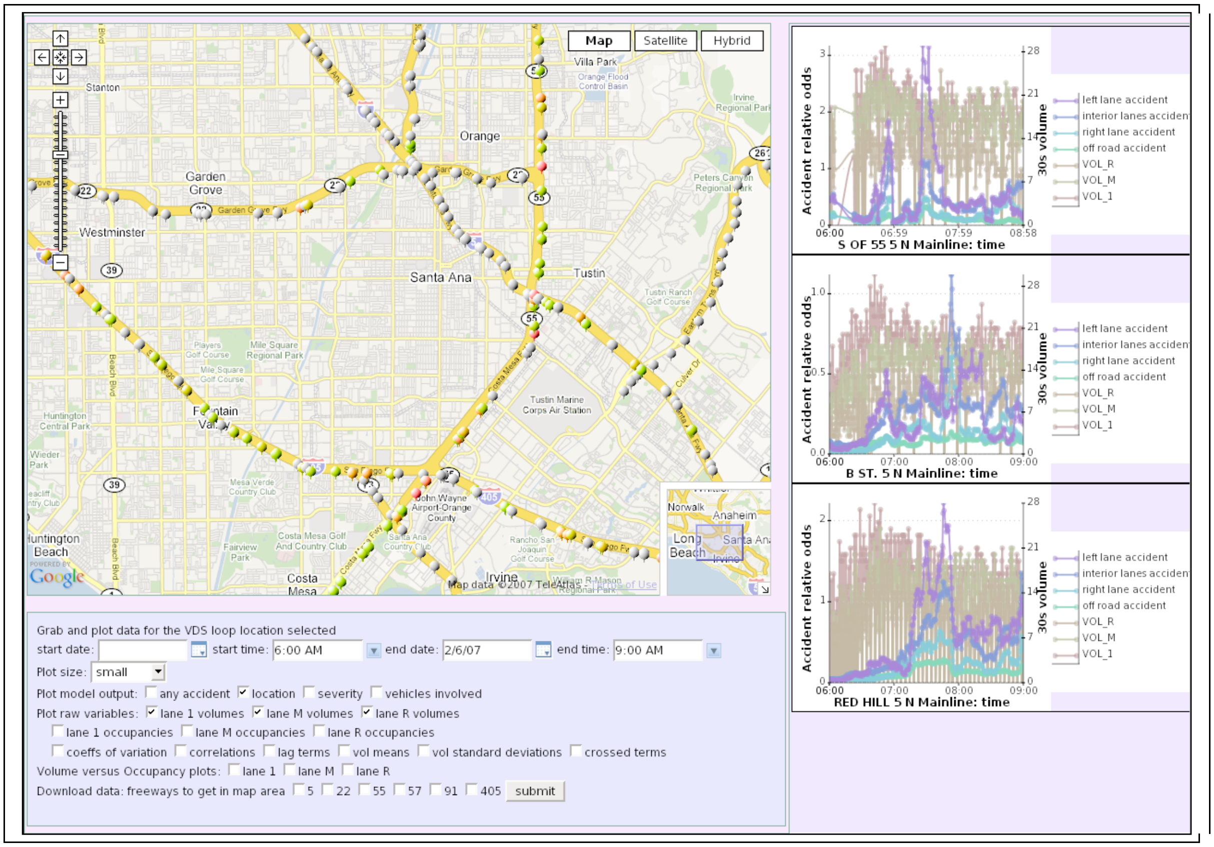
1. Login Check: as before.
2. Parse Timestamp: as before.
3. Parse Model List: as before.
4. Sort the Model List: as before.
5. Historical Query of VDS and Safety Data: parse the request for the specific VDS desired, and then call the database query described in Section 6.9.3.
6. Convert Results for One VDS to JavaScript Object Notation (JSON): This command is slightly different than before. In this case there are multiple timestamps and a single detector, so the output is split into a JSON list containing the data for the VDS over time, and another JSON object describing the VDS's characteristics. Again, the `VDSAccidentRisk` class does the work.

5.11 The Web-based User Interface

The last aspect of the implementation that needs to be described is the user interface, or the View in the Model-View-Controller (MVC) pattern. The previous section described how a very simple Struts controller routed various requests to Chains of Responsibility in the Model. The output of each chain was a JSON string. The reason for this is that the user interface is a modern web interface that uses asynchronous JavaScript requests to change the contents of the web page. This technology is often called "AJAX" although the *X* in AJAX refers to XML, while in this implementation we are using JSON. Nobody uses the acronym AJAJ, so we will continue to use the term AJAX.

In order to bootstrap the application, an initial page must be loaded that contains all of the required HTML and JavaScript. A JSP page was written to do this, called `mapoverlay.jsp`. A simple welcome controller was written that calls a very short chain of responsibility that checks if the user is properly logged in to the Testbed website, and then sends this JSP page to the user. This page loads the Dojo JavaScript library to provide dialog widgets and helpful JavaScript language constructs; loads up a map using the Google Maps API; and finally loads custom JavaScript libraries we have written to add functionality to the Google Map interface and to connect to the safety odds calculations server we have constructed. Then a short inline JavaScript block initializes the Google Map, sets the default position of the map, and hooks up the various form input elements to the appropriate JavaScript controllers. Finally, a block of HTML lays out the page, including a container div (the lexical block defined by a matched pair of `<div>`…`</div>` tags) that holds the main user interface, and an output div that contains links to any files that the user might download. The user interface div contains three child div blocks: a div for the map, a div for plotting graphs, and a div for collecting the input parameters for the queries. The queries div contains various form elements as child nodes.

When the page loads, the first task is to load the Google Map, or echo an error to the screen if the browser is not compatible with Google Map. This is the main advantage of using the Google Maps API—not having to struggle with accommodating all of the different web browsers that exist. If the map is successfully created, then the custom map overlay called `VDSOverlay` is loaded and connected to the map div, the query div, and the data output div. The initialization also specifies which http addresses to use for getting VDS data. When the initialization step returns, the `handleMapMove()` method is called to simulate a map move action, which finds and retrieves the VDS data for the current map view by calling the chain of responsibility described earlier. In addition, a timer is set to call the `handleMapMove` method every few minutes so that the map always shows the recent odds predictions. The user should see the map load, followed by a sprinkling of colored dots on the map, as in the map window shown in Figure 18.



Grab and plot data for the VDS loop location selected

start date: start time: 6:00 AM end date: 2/6/07 end time: 9:00 AM

Plot size: small

Plot model output: □ any accident ☑ location □ severity □ vehicles involved

Plot raw variables: ☑ lane 1 volumes ☑ lane M volumes ☑ lane R volumes

□ lane 1 occupancies □ lane M occupancies □ lane R occupancies

□ coeffs of variation □ correlations □ lag terms □ vol means □ vol standard deviations □ crossed terms

Volume versus Occupancy plots: □ lane 1 □ lane M □ lane R

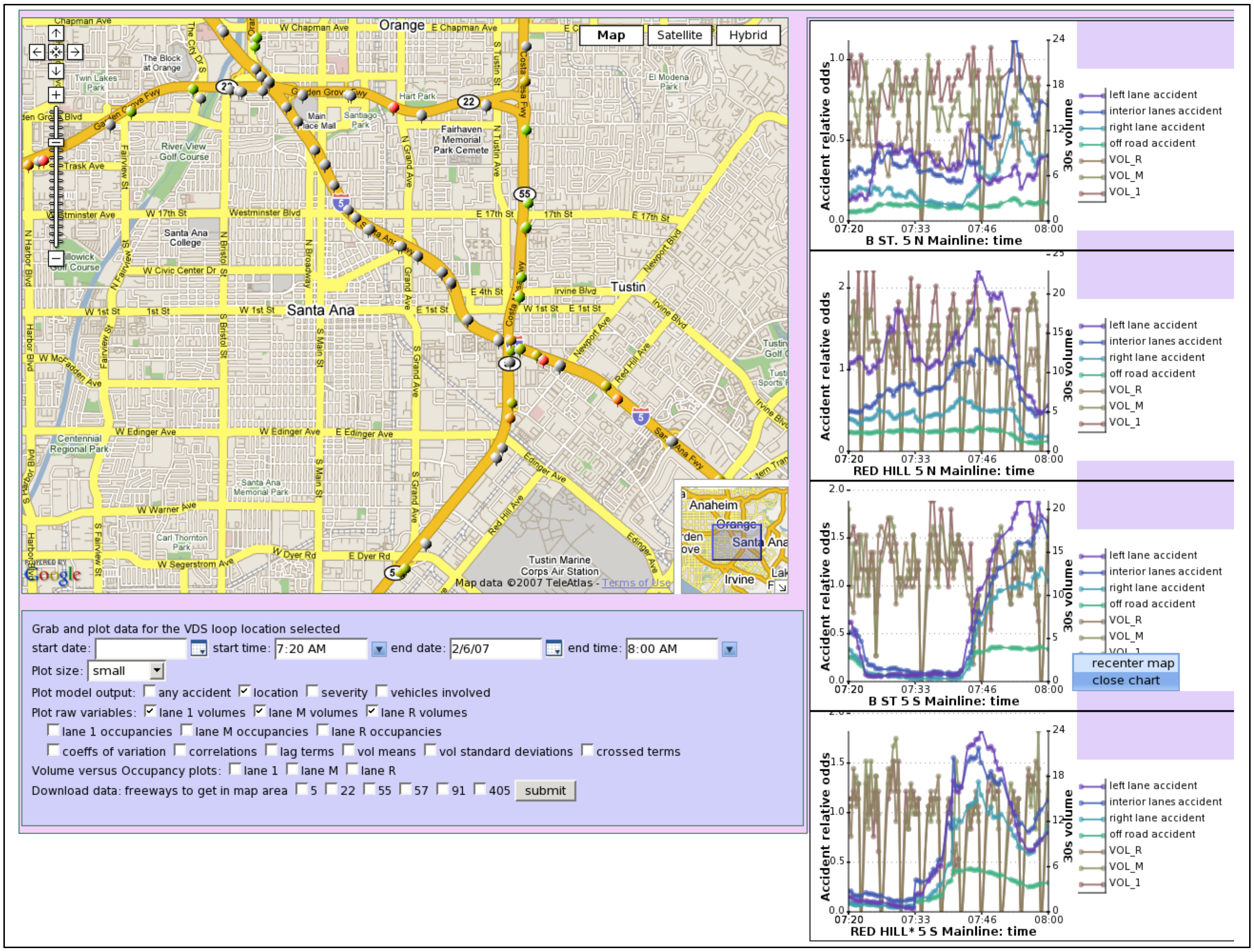
Download data: freeways to get in map area □ 5 □ 22 □ 55 □ 57 □ 91 □ 405 submit

Figure 18 The user interface screen, with the map and the VDS data loaded and three VDS detectors plotted (I5 northbound from 6am to 9am).

Below the map shown in Figure 18 is the query block. The first row sets the times and dates of interest. Note that if the user sets the ending time in the dialog, then this date and time are used to color the VDS icons shown on the screen. Since the user has demonstrated an interest in historical information, the automatic refreshing of the VDS colors is disabled. The next rows specify what data should be shown in the plots of individual VDS detectors. The last line lists the freeways for which we are modeling safety odds, and a submit button. The submit button will send a request to the server to download a data file, calling the chain of responsibility described in Section 6.10.2. To guard against overwhelming the server with a massive data request, the query submit button is disabled when the map is zoomed out too far. The point of this user interface is to allow analysts to zoom in on exactly the stretch of freeway and time period that is interesting, and worthy of further off-line analysis.

To the right of the map in Figure 18 is the charting block, containing three different plots. These plots were generated by clicking on the corresponding VDS icons in the main map window. These charts use the Dojo toolkit's `dojo.charting.Chart` class, coupled with custom code for initializing the chart and handling further interactions. When the webpage first loads, all of the options are set to their defaults, with the blank dates and times indicating that most requests will get the current time.

When the user clicks on a VDS icon, a request is sent to the web server with the starting and ending times and dates. The request calls the "single VDS" chain as described in Section 6.10.3. If no times are specified, the current time is used. As explained in earlier, the results of the query are sent back to the web browser using JSON. The Dojo toolkit has methods to automatically parse the JSON and generate proper JavaScript objects. These objects are used to create the requested plot, and to instantiate an object that controls the plot. The controller object holds the loop data in memory, and listens to the user interface controls in the query block. If one or more of these change, then the chart is redrawn with the new options. If the start or end time change, then the existing data is discarded and a new request is sent to the server, resulting in a new charting object being created. The charting object also creates two menu dialog options that show up on the chart when the user clicks the right mouse button: "recenter" and "close". This menu is shown in Figure 19. "Recenter" centers the map on the VDS that is charted, and "close" closes the chart and its associated objects.



Grab and plot data for the VDS loop location selected

start date: start time: 7:20 AM end date: 2/6/07 end time: 8:00 AM

Plot size: small

Plot model output: □ any accident ☑ location □ severity □ vehicles involved

Plot raw variables: ☑ lane 1 volumes ☑ lane M volumes ☑ lane R volumes

□ lane 1 occupancies □ lane M occupancies □ lane R occupancies

□ coeffs of variation □ correlations □ lag terms □ vol means □ vol standard deviations □ crossed terms

Volume versus Occupancy plots: □ lane 1 □ lane M □ lane R

Download data: freeways to get in map area □ 5 □ 22 □ 55 □ 57 □ 91 □ 405 submit

Figure 19 The user interface screen, getting details on a suspected trouble spot (I5 south of SR55 at 7:45 AM). Note the starting and ending times specified, and a menu dialog is shown on the third chart.

Figures 18 and 19 also show what happens when data is missing. In the main map several of the VDS icons are colored gray in the original color image. This means that those loops did not have any odds predictions for 9 AM (Figure 18) or 8 AM (Figure 19). At any given time, many mainline detectors do not have enough data to generate odds predictions. This is due to the requirements that are placed on the data for generating a prediction. Problems might include the data duplication and doubling errors we described earlier, and not meeting the other requirements noted in the three initialization steps described in the section on computing the input variables. Especially important are the left and right lane loops, as gaps in these are more likely to prevent the odds prediction. In some sense, the odd predictions also serve as a way to examine the quality of the mainline VDS loops in the District 12 study area, as all of the grayed out loops have problems of one sort or another.

Another artifact that should be noted in the charts shown in the screenshots are the repeated drops in volume down to zero. These dips are caused by no valid volume data being available for those time steps, although a valid safety odds prediction was made (the blue lines in the charts). From inspecting the raw data at these times, it is obvious that the cause of these drops is the data duplication and doubling problem noted earlier.

As a consequence of problems with the input data, most VDS detectors have periods of time with no predictions. The first chart shown in Figure 18 provides an example of this. The straight line from about 6 AM to 6:30 AM is a result of the plotting engine interpolating over a gap in the input data. This means that the data processing loop was aborted for that VDS detector for that period of time, and no data was saved. There are many reasons that this might occur, and many are avoidable. Future research should focus on improving the data stream from the loops in the field, to extend the efficacy of our models.

6 Forecasts

To assess the performance of the tool, we investigated forecasts of the models for actual times and places in the study area. Three case studies are presented.

6.1 Case Study I: Southbound SR-55 at 17th Street on November 28, 2006

The first case study is the same as previously used to demonstrate the traffic flow variables. In Figure 20 we superimpose the forecast odds ratio of any type of accident on the two time series of raw data parameters graphed in Figure 4 in order to illustrate how the forecasts are related to congestion. This odds ratio is simply the forecast of the model detailed in Table 2 evaluated at each 30-s interval. These are relative odds only at this stage. For example, if we compare two time periods, and one has a forecast of the relative odds of an accident of 10.0, while the second has a forecast of 2.0, it simply means that the likelihood of accident is five times higher in the first period than in the second period, according to the model.

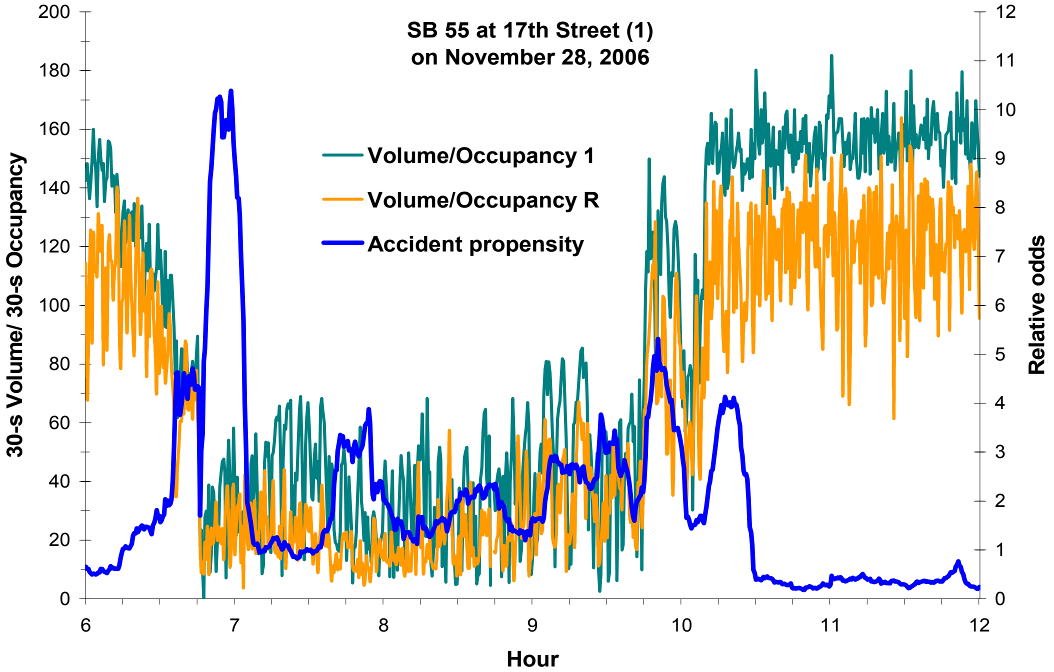


Figure 20 Forecasts of the Accident Propensity Model with Raw Left and Right Lane Volume / Occupancy Ratios for SB SR-55 at 17th Street on the morning of Nov. 28, 2006

Figure 20 demonstrates that the likelihood of any type of accident spikes when the road breaks down from free flow operation to congested operation. The odds of an accident occurring increase by a factor of about five at approximately 6:30AM, when speeds begin to drop. This holds for a period of about ten minutes, until about 6:45AM when the SR-55 freeway grinds to a temporary halt. The model for any accident then forecasts an increase in the odds of an accident to a level of about ten, compared to a pre-congestion level of less than one in the free flow period around 6:00AM. The predicted relative odds then oscillates between about one and four during the extended period of congestion from 6:45AM until about 9:45AM. At 9:45AM the level of service recovers, then degrades again, then finally recovers to stay for the remainder of the morning at about 10:15AM. The forecast relative odds of an accident reflects this blip in the recovery before retreating to a fairly consistent value of about 0.5 for the remainder of the case study period.

The forecasts of the accident severity, vehicle involvement, and location models are graphed for the case study in Figures 21, 22, and 23, respectively. The accident severity model (Figure 21) predicts that the increase in accident likelihood during the free-flow to congested transition is mostly due to the heightened chance of property damage only accidents. As expected, the difference between the odds of property damage and injury accidents is also greater during periods of congestion than during periods of free flow.

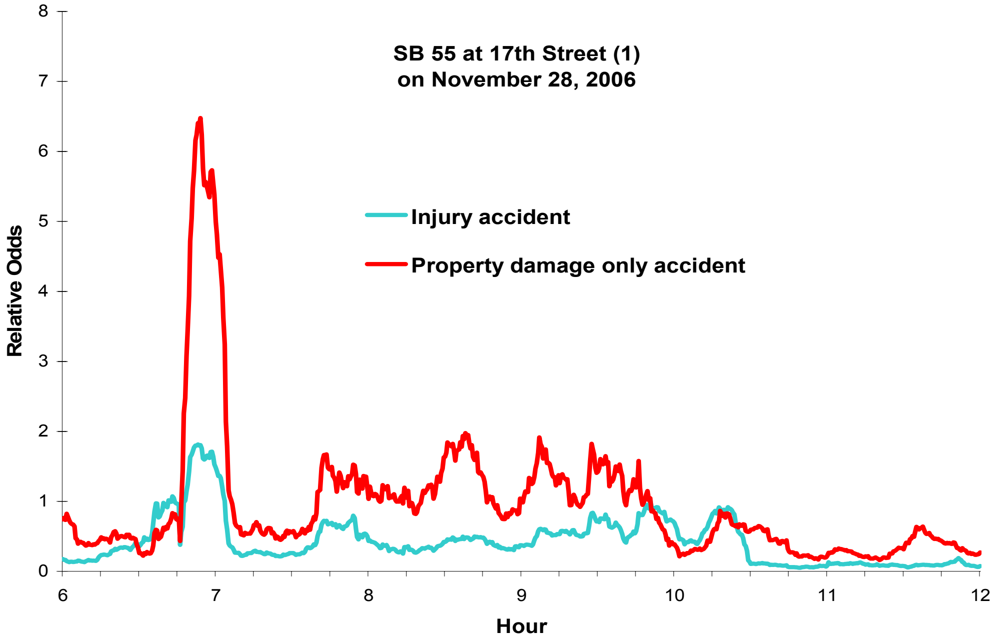


Figure 21 Forecasts of the Accident Severity Model for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

The vehicle involvement model (Figure 22) predicts that large scale accidents are most probable during the initial decay in level of service, beginning shortly after 6:30AM. Two-vehicle accidents then become most likely during the period of congestion.

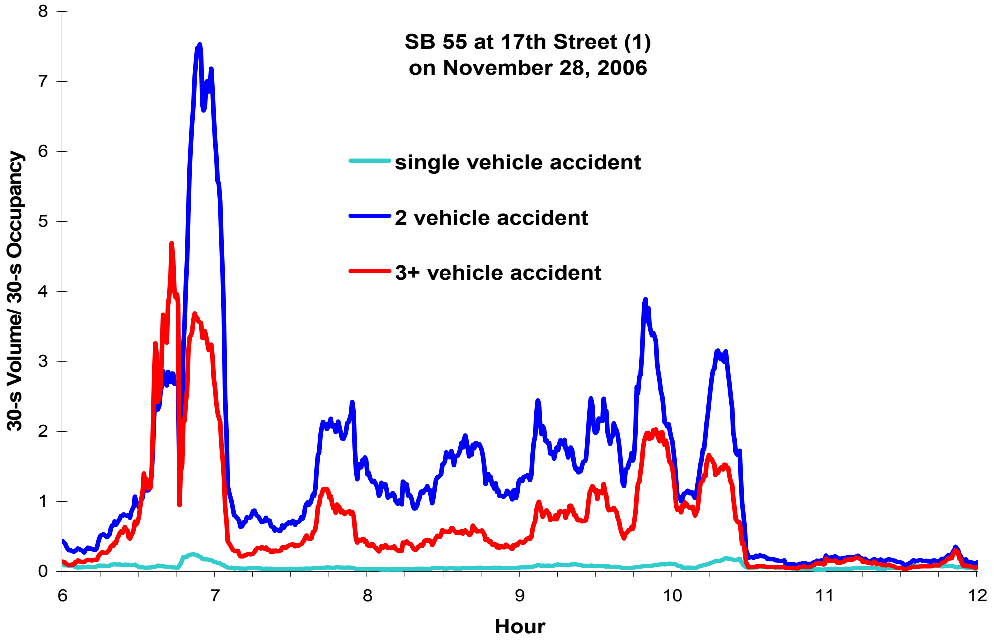


Figure 22 Forecasts of the Vehicle Involvement Model for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

The collision location model (Figure 23) attributes the jump in accident likelihood during the transition periods primarily to left lane accidents, and secondarily to interior lane accidents. The likelihood of right lane accidents is elevated only in the period of the lowest level of service, from about 6:45 to 7:00. During the relatively stable period of congested operation, from about 7:10 until 10:00, the odds of left-lane and interior-lane accidents are similar. The odds of an interior-lane accident is elevated during the period of recovery to free-flow operation, in the 10:15 to 10:30 time frame.

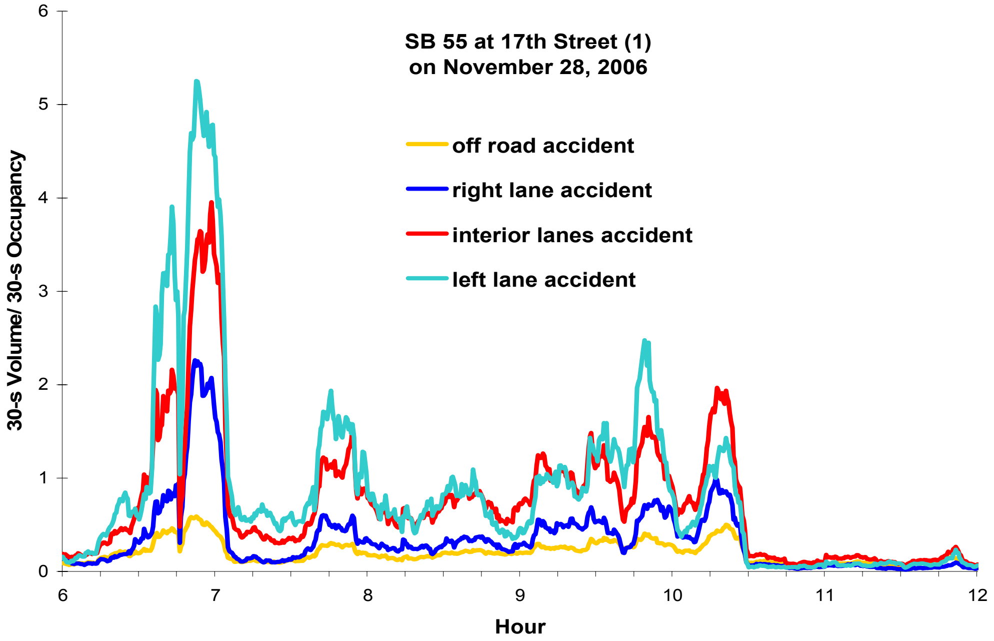


Figure 23 Forecasts of Accident Location Model for SB SR-55 at 17th St. on the morning of Nov. 28, 2006

6.2 Case Study II: Northbound I-5 South of SR-73 on November 6, 2006

The second case study of model forecasts is for northbound I-5 at the loop detector station south of SR-73 on November 6, 2006. We focus on the shorter interval from 7:00AM to 9:00AM. An accident downstream of this location produced a shockwave that caused traffic at this location to come to a complete halt shortly after 7:30. The forecast odds ratio of any type of accident is superimposed on the two time series of raw data volume to occupancy ratios for the left and interior lane in Figure 24. The model predicts a rapid and substantial increase in the probability of an accident due to the stoppage of traffic and subsequent stop and go conditions. The odds of an accident then decreases as traffic is funneled through at very low speeds, but these odds increase again during the period , from 7:58 until about 8:20. During this time speeds are building up but still oscillating greatly across the 30-s intervals. This variance in speeds is generally greater in the left lane than in the interior lanes. The predicted likelihood of an accident finally goes down after the road returns to free flow condition.

The collision location model forecasts are graphed in Figure 25. The model predicts that the spike in accident likelihood is almost entirely due to increases in the probability of left-lane and-interior lane accidents. This is consistent with CHP reports that the downstream incident resulted in blockage of all but the left lane. It appears that the model has picked up the shift in traffic to the left based on the values of the twenty-seven traffic flow variables. One example is the greater variation in left lane speeds apparent in Figure 24.

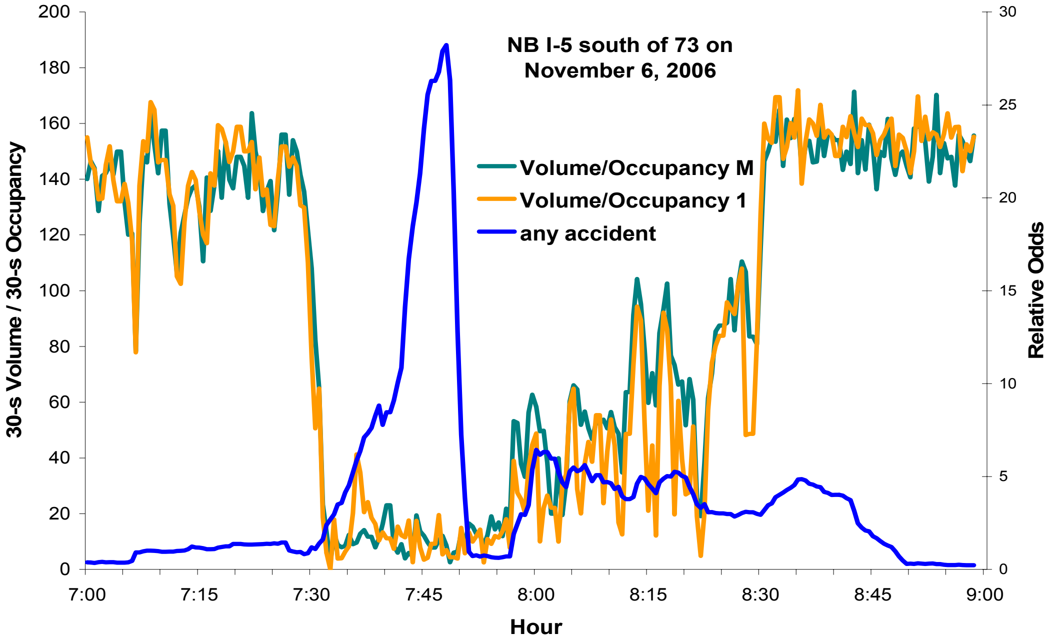


Figure 24 Forecasts of the Accident Propensity Model with Raw Left and Interior Lane Volume / Occupancy Ratios for NB I-5 South of SR-73 on the morning of Nov. 6, 2006

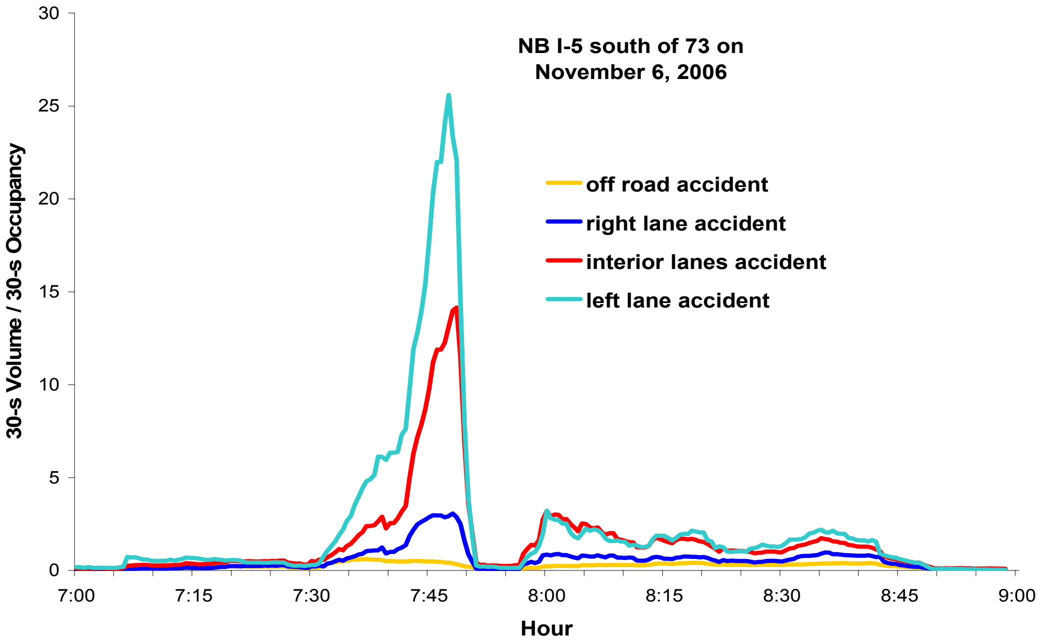


Figure 25 Forecasts of Accident Location Model for NB I-5 South of SR-73 on the morning of Nov. 6, 2006

6.3 Case Study III: Northbound I-405 at Brookhurst on November 30,2006

The final case study is for a all day, from 6:00AM until 11:00PM, for NB I-405 at Brookhurst station 1, on November 30, 2006. Forecasts of the relative odds of any type of accident are superimposed on the graph of volume to occupancy ratio for the interior lane on that day in Figure 26. The volume/occupancy plot shows that this location witnessed a morning with only two very short periods of reduced interior lane speeds, followed by an abrupt period of congested operation during the noon hour, and then an extended period of congested operations in the afternoon and early evening, beginning at about 2:00PM and lasting until about 8:00PM. The forecasts of the accident propensity model track the changes in level of service. The breakdown during the noon hour results in a substantial increase in the predicted likelihood of an accident. The odds of an accident are relatively high throughout the entire afternoon and evening period of congested operation, but there are spikes that appear to accompany subtle transitions. The initial transition from free-flow to congested operation, initiated at about 2:00, appears to be followed by a transition to an even heavier level of congestion approximately one hour later. The model predicts that this second state is even potentially more dangerous. Several other perturbations are detected by the model during the 3:00PM to 8:00PM period. It is clear that the model, which is being driven by all the variables listed in Table 2, is tracking nuances in traffic flow that are not visible raw volume to occupancy ratio for a single lane.

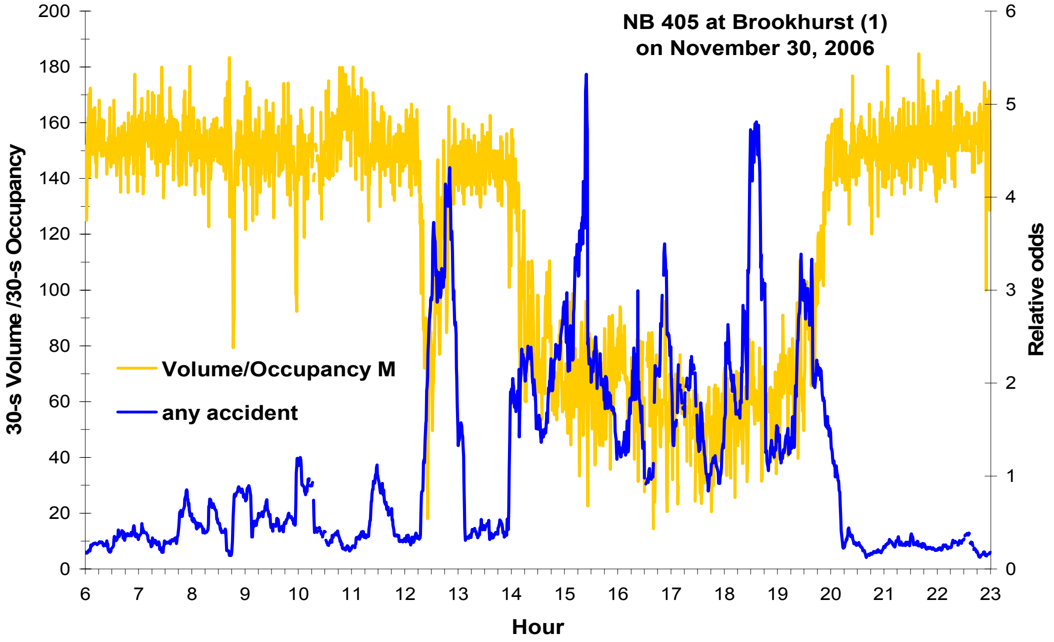


Figure 26 Forecasts of the Accident Propensity Model with Interior Lane Volume / Occupancy Ratios for NB I-405 at Brookhurst on Nov. 30, 2006

The accident location model forecasts for this same location and time period are graphed in Figure 27. This model predicts that the spike in accident likelihood during the non hour is largely attributable to the likelihood of left-lane accidents. This is also true of the increase in the odds of any accident at the onset of the extended period of congestion. However, the spike in the odds of any accident accompanying the secondary transition to heavier congestion after 3:00PM is caused by a spike in the likelihood of interior-lane accidents, as well as right-lane accidents.

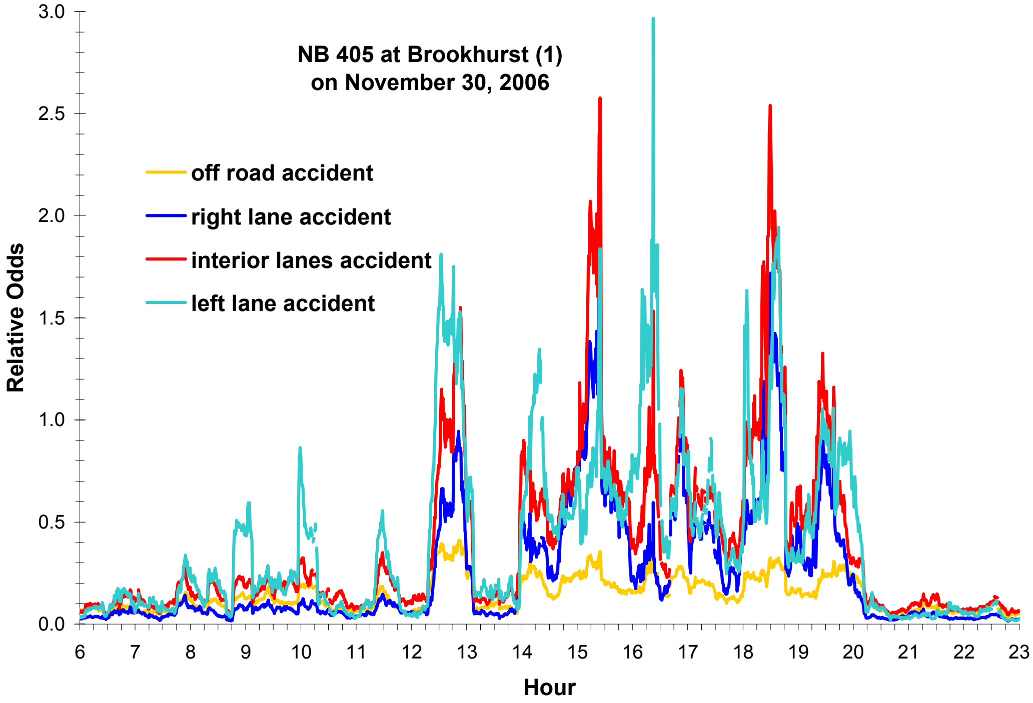


Figure 27 Forecasts of Accident Location Model for NB I-405 at Brookhurst on Nov. 30, 2006

7 Transformation of Probability Results to Measurable Quantities

As noted previously, the models developed yield the log of the odds of specific accident characteristics relative to some reference category. Because the estimation of these models was conducted using a biased, rather than a representative, sample of observations heavily skewed toward the inclusion of accident events, the resulting probability expressions can not be used directly as measuring the probability of the occurrence of any specific event, without first correcting for the sampling bias. Here we present the appropriate correction factors to obtain measurable probabilities from the expressions.

Let $P(acc|\beta_0+\underset{\sim}{\beta}^T\cdot\underset{\sim}{x})$ denote the binary probability of an accident occurrence during any 30-sec time interval, given a vector of traffic conditions $\underset{\sim}{x}$ and coefficients $\beta_0,\underset{\sim}{\beta}$; here and elsewhere the superscript T denotes transpose. Further, let $P(acc\,char\,k|\theta_0^k+\underset{\sim}{\theta}_k^T\cdot\underset{\sim}{x})$ denote the probability of an accident that has occurred having a particular characteristic k, given a vector of traffic conditions $\underset{\sim}{x}$ and coefficients $\theta_0^k,\underset{\sim}{\theta}_k$, as referenced to a base characteristic $k=b$.

7.1 Estimation of $\beta_0,\underset{\sim}{\beta}$

7.1.1 Sampling:

Let n denote the total number of loop stations in the sample year. Let t_i denote the total number of 30-sec reports for loop station $i,\ i=1,\dots,n$ in the data set. Then $N=\sum_{i=1}^{n}t_i$ represents the total number of observations. Let N_{acc} ($N_{acc}=4,412$) denote the total number of accidents in the data set. Then $N_0=N-N_{acc}$ is the total number of "non-accident" observations.

7.1.2 Estimation:

In our estimation, we employ a sample comprised of $N_{acc}^{ok}<N_{acc}$ ($N_{acc}^{ok}=1,712$) accident occurrences for which sufficient data are available, and $N_{non}=4,441$ of randomly selected non-accident observations. The corresponding maximum likelihood estimates for $\beta_0,\underset{\sim}{\beta}$ are determined as $\hat{\beta}_0,\hat{\underset{\sim}{\beta}}$.

7.1.3 Correction:

According to Ben-Akiva and Lerman (1985; p 237-238) the appropriate correction involves only the constant $\hat{\beta}_0$. The correction is given by

$$\hat{\beta}_0^{corrected} = \hat{\beta}_0 - \ln\left(\text{sample rate}/\text{population rate}\right)$$

In our application, the sample rate is $1,712/6,153 = 0.278$; the population rate is N_{acc}/N . So, assuming no bias in missing data, the correction is given by:

$$\hat{\beta}_0^{corrected} = \hat{\beta}_0 - \ln\left(0.278 \cdot N/N_{acc}\right)$$
$$\hat{\beta}_0^{corrected} = \hat{\beta}_0 - \ln\left(0.278 \cdot N/4412\right)$$

7.1.4 Determining *N*:

The accident sample was drawn from a freeway network monitored by a total of 457 loop stations, generally spaced at approximately equal intervals. The sample was taken during the six-month period March through August, 2001—a total of 184 days—leading to a total population of $N = 529,920$ loop station-30 second observations that were populated by the 4,412 recorded accidents. Then,

$$\hat{\beta}_0^{corrected} = \hat{\beta}_0 - \ln\left(0.278 \cdot 529,920/4412\right)$$
$$\hat{\beta}_0^{corrected} = \hat{\beta}_0 - \ln\left(33.41886\right)$$
$$\hat{\beta}_0^{corrected} = \hat{\beta}_0 - \ln\left(33.41886\right)$$
$$\hat{\beta}_0^{corrected} = \hat{\beta}_0 - 3.509$$

And, correspondingly, $\hat{P}(acc|\hat{\beta}_0^{corrected} + \underset{\sim}{\hat{\beta}}^T \cdot \underset{\sim}{x})$ is the appropriate estimated probability of an accident occurrence within any particular freeway section (defined by the midpoint distances between successive loop stations upstream and downstream of the section) during any particular 30-sec time interval.

7.2 Estimation of θ_0^k , $\underset{\sim}{\theta}_k$

7.2.1 Sampling:

Let n_k denote the total number of accidents in the sample year that are classified as having characteristic $k,\ k = 1,\ldots,n_{char}$. Let n_k^{ok} denote the total number of accidents in the data set that are classified as having characteristic $k,\ k = 1,\ldots,n_{char}$ that have sufficient data to be included in the data set used for estimation. $k,\, k=1,\ldots,n_{char}$. Then $\sum_{\forall k} n_k = N_{acc}$ ($N_{acc} = 4{,}412$) and $\sum_{\forall k} n_k^{ok} = N_{acc}^{ok}$ ($N_{acc}^{ok} = 1{,}712$).

7.2.2 Estimation:

In our estimation, we employ a sample of N_{acc}^{ok} ($N_{acc}^{ok} = 1{,}712$) comprised of n_k^{ok}, $k=1,\ldots,n_{char}$ accident occurrences for which sufficient data are available. The corresponding maximum likelihood estimates for $\theta_0^k, \underset{\sim}{\theta}_k, j=1,\ldots,n_{char}-1$ ($j=b$ is the reference characteristic) are determined as $\hat{\theta}_0^k, \hat{\underset{\sim}{\theta}}_k$, with $\theta_0^{k=b} \equiv 0$.

7.2.3 Correction:

Again, according to Ben-Akiva and Lerman (1985; p 237-238) the appropriate corrections involve only the constants $\hat{\theta}_0^k$. The correction is given by

$$\hat{\theta}_0^{k\ corrected} = \hat{\theta}_0^k - \ln\left(\text{sample rate of } k / \text{population rate of } k\right)$$

In our application, the sample rate of k is n_k^{ok}/N_{acc}; the population rate is n_k/N_{acc}. So, assuming no bias in missing data, the correction is given by:

$$\hat{\theta}_0^{k\ corrected} = \hat{\theta}_k^0 - \ln\left(n_k^{ok}/n_k\right)$$

In particular, the correction for the base (reference) characteristic is

$$\hat{\theta}_0^{k=b\ corrected} = 0 - \ln\left(n_{k=b}^{ok}/n_{k=b}\right)$$
$$\hat{\theta}_0^{k=b\ corrected} = -\ln\left(n_{k=b}^{ok}/n_{k=b}\right)$$

In order to preserve $k=b$ as the reference, we need to add $\ln\left(n_{k=b}^{ok}/n_{k=b}\right)$ to all of the corrected terms, yielding

$$\hat{\theta}_0^{k\ corrected} = \hat{\theta}_k^0 - \ln\left(n_k^{ok}/n_k\right) + \ln\left(n_{k=b}^{ok}/n_{k=b}\right);\ k \neq b$$
$$\hat{\theta}_0^{k=b\ corrected} = 0$$

And, correspondingly, $\hat{P}(acc\,char\,k|\hat{\theta}_0^{k\,corrected} + \hat{\underset{\sim}{\theta}}_k^T \cdot \underset{\sim}{x})$ is the appropriate estimated probability of an accident that has occurred having a particular characteristic k.

7.3 Validation

Consider a time period comprised of T 30-second time intervals over freeway sections monitored by N loop stations. Let $\mathbf{T}_j$ denote the set of 30-second time intervals for which loop station j has reported valid data. Denote by $\underset{\sim}{x}_{ij}$ the vector of traffic characteristics associated with the i^{th} element of $\mathbf{T}_j$. Then, the quantity

$$\sum_{j=1}^{N} \sum_{\forall i \in \mathbf{T}_j} \hat{P}(acc \big| \hat{\beta}_0^{corrected} + \underset{\sim}{\hat{\beta}}^T \cdot \underset{\sim}{x}_{ij})$$

is an estimate of the expected value of a lower bound (since, in general $\left|\mathbf{T}_j\right| < T$) to the number of accidents recorded in the TASAS data base for the time period comprised of T 30-second time intervals over freeway sections monitored by N loop stations.

8 DIRECTIONS FOR FURTHER RESEARCH

8.1 Validate Model Performance

The tool can be authenticated to support its use in planning and operations management. Computed traffic flow variables and the odds predictions have been stored, together with the raw loop detector data, for the Orange County study area since November 2006. This is automatically gathering the resources necessary to perform a comprehensive validation study. Once accident statistics become available in the TASAS database or elsewhere, we can compare our predictions to actual accident rates, and evaluate how our models are performing. Time lags in the compilations of the accident databases dictate that such a validation study for archived data beginning in late 2006 be conducted no sooner than 2008.

Such a validation study can only determine probabilistic relationships between the rare event of an accident and model predictions of the likelihood of an accident of a certain type, as described in Section 8.3. Many accidents will also occur at times and places where loop detector data are missing, so that no model predictions are available. However, a validation study will be able to confirm or deny many of this study’s results, and it will surely indicate ways in which our models can be improved. The study will also provide updated information for calibrating the tool along the lines documented in the previous Section 8.

8.2 Test Enhancing the Models to Include Road Layout Variables

It is possible that some explanatory power can be gained by adding geometric design variables in the accident prediction models. Our traffic flow variables capture patterns of traffic flow that can be measured by levels and variations in volume, density, and speed, and the relationships of these parameters across lanes and shifts in them over time. These variables surely reflect many aspects of driver behavior in response to restricted sight distances, merging and weaving vehicles due to on- and off-ramp locations, lane widths, and all changes in road geometry. The major advantage to this approach is that the models do not require any data other than VDS loop detector data; they are thus highly transferable.

Testing whether a road layout variable adds anything to the models is a straightforward analytical exercise, but the difficulty is that the geometric design is unavailable in electronic format for many of the VDS locations. Indeed, studying the map generated by the odds prediction user interface shows that even the latitude and longitude of VDS locations is often only approximate. Linking potentially useful geometric data, such as vertical and horizontal curvature, into a consistent x-y-coordinate that includes VDS locations will require support from Caltrans. However, such an exercise is likely to yield data that will support many other projects involving network simulations.

8.3 Explore the Data Delivery Options of the Safety Information

There are myriad issues involved in the dissemination of safety information such as that provided by this tool. Caltrans can use such information to inform drivers of “trouble ahead” using changeable message signs, advisory radio, or other communication devices. The best ways of presenting the information need to be studied in detail. In Europe, specifically the United Kingdom, motorway drivers are instructed to “stay in lane” when certain traffic flow conditions are detected, and this mandatory instruction is lane specific and photo-enforced. Many other driver directives are possible, depending upon the type of danger and the roadway situation, and these should be explored in a follow-on study.

The safety information should also prove useful to the California Highway Patrol and local police and emergency agencies, for potential “heads up” alert status and possibly for pre-positioning of resources. Police vehicles might also be used in calming or guiding traffic flows. Consultation with experts from such agencies will be useful in alerting them to the existence of the information and in identifying appropriate content and communication channels.

It is also possible that accidents can be reduced by simply informing the traveling public of times and places where the odds of accidents are elevated. This alone could go a long way towards reducing risk, as drivers will be alerted to dangerous conditions and will drive more defensively. To accomplish this, drivers must be able to access the information now available on the website. There are many ways that this can happen. For example, an RSS (family of web feed formats) feed can be implemented for the site to broadcast changing conditions and recent odds elevations. These RSS streams can be accessed by cell phone. Alternately, travelers can sign up to receive instant messages sent to their phone. Another broadcast mechanism would be to encourage radio stations to include safety odds in their usual traffic updates. Liability issues will need to be explored in all of these contexts.

8.4 Investigate the Driver Behavior Phenomena Captured by the Models

The success of the modeling argues strongly that more research should be conducted to better understand what aspects of driver behavior the models are capturing. It is not clear to what extent we are identifying newly defined patterns of driver behavior that are especially effective precursors of accident risk, and to what extent we are, for the first time, relating well known traffic flow phenomena to accident risk. We recommend a systematic examination of the different conditions (as represented by our traffic flow variables extracted from simple loop detector data) that result in elevated risks, and then comparing those conditions to more macroscopic measurements such as aerial photographs or video detection. It might turn out that counts of driver actions, such as lane changing maneuvers of different types and vehicle gap acceptances, are robustly related to some of our traffic flow variables. Such an investigation should contribute to our fundamental knowledge of traffic safety and demonstrates Caltrans' commitment to provide the safest possible transportation system.

8.5 Linking the Variables and Forecasts with Drivers' Perceptions of Risk

An appealing research topic is determining how drivers' perceptions of risk are related to the model forecasts and the traffic flow variables that drive the models. This can be accomplished by designing and executing a survey of drivers in the Orange County study area. To prevent distraction, the survey should probably be implemented using a non-interactive GPS recorder and a post-journey questionnaire. Drivers can be asked where and when they were driving when they felt less or more safe, and these stated attitudes and perceptions can be linked to actual lane positions and lane changing maneuvers they performed, using the GPS record. Drivers can be asked their opinions about what causes their comfort or discomfort with traffic conditions, allowing them to reference freeways and times that may not be in the GPS travel record. For example, a traveler may perceive a certain time of day or section of roadway as too dangerous to use. That important fact will not show up in a travel diary. All of these perceptions and opinions can be related to the loop-based safety predictions and data being archived on the project website.

There are various means of implementing such a survey – web-based, postal, or telephone – and issues of sample size and recruitment need to be explored. Insights into driver interactions and sensing of driver inattention could be quite useful in identifying ways to improve driver education and signage. It is also expected that age and experience will play a role both in the *perception* of safety, and in the observed *driving behavior* under different safety conditions (both actual and perceived). At the same time, many situations will produce consistent perceptions about safety across different classes of drivers (*i.e.* age, experience, gender, car type), providing both an opportunity to calibrate models of different types of drivers, and an indication of specific facilities that would benefit further study by Caltrans.

8.6 Evaluate Integration with Adaptive Cruise Control

Our models link traffic flow characteristics with the likelihood of different types of accidents. The same information might be used to influence driver behavior in those situations with heightened danger levels. This can be accomplished by warnings and driver instructions using changeable message signs and advisory radio. It might also be accomplished through the use of on-board devices such as adaptive cruise control (ACC), now being provided by auto manufacturers on certain luxury models. A useful course of action is to cooperate with vehicle and device manufacturers in exploring means of taking advantage of real-time safety information to influence safer driving.

With further research, it may be possible to identify specific actions that cars in a stream can take to “calm down” relatively risky conditions, such as flashing brake lights, or engage in evasive maneuvers that are more controlled and predictable to upstream drivers. If corrective strategies can be devised for different kinds of traffic conditions, and if the fleet of ACC-equipped vehicles is large enough, then those vehicles could be leveraged to reduce the risks of accidents in some situations.

8.7 Evaluate Use of Virtual Moving Detectors

Safety evaluations of traffic flow can be transferred from fixed VDS locations to vehicles building on the ACC (adaptive cruise control) concept. In the near future, on-board detectors are likely to be capable of measuring speeds and densities of a vehicle and all of its nearest neighbors. It will then be possible to use accumulations of those data to build the same type of models developed in this project, and to predict the relative risk of different kinds of accidents. If successful, on-board sensing could expand the scope of the present tool to a seamless coverage of an entire network.

Research can be done now to help design future on-board detectors and to study how such detector systems could be used to predict accident odds. The present study has shown that traffic flow variables computed from data collected by VDS loop detectors are effective precursors of levels of safety. An important question is: what type of data can be collected by on-board detectors that will serve in a similar role. Test data from floating cars equipped with different types of sensors and GPS devices can be compared to fixed VDS data to evaluate the extent to which on-board data replicate and expand VDS data. Further issues to be studied include methods of pooling data across drivers, and coverage in terms of time, space, and driver sampling.

8.8 Identify VDS Detectors that Would Benefit Most from Maintenance

Finally, to improve the efficacy of the odds prediction techniques described here, we strongly urge Caltrans to improve the data coming out of the loops, as many loops do not produce data of sufficient quality to be used for safety odds predictions. We have identified at least one problem that can be observed nearly every hour of every day—the data duplication and doubling problem described in Section 6.3. It should be a simple task to examine a VDS detector station and identify when data is duplicated. It may be more difficult to determine the exact cause of the problem and devise a solution.

We also have suspicions about the status bit being set on the incoming loop data records. As we have used the safety odds browser to examine what our models are doing, we have tried to find major traffic events in real-time to see how our models respond. In many cases, our safety predictions “switch off” just after an incident occurs and the freeway segment hits jam conditions. Closer inspection of the raw data has shown that often, just after a single loop has hit an occupancy of 1, the bad status bit gets set and our data processing routine automatically drops that observation. Typically this condition will persist for the duration of the incident, with the status of jammed loops only occasionally going back down to the 0 state.

An excellent application of our odds prediction browser would be to properly study how loops behave after incidents, focusing on those cases in which the incoming data are insufficient to generate safety odds predictions. These cases indicate that the loop data are poor, and most likely not good enough for any other kind of modeling as well.

One obvious recommendation would be to expand the descriptive power of the status flag, so that it isn't just "0" or "1", but rather is an error code describing the problem or inconsistency. Our suspicion is that a status of 1 might sometimes mean only that occupancy spiked in one lane relative to other lanes (which may be exaggerated when the duplication and doubling events occur) since we have observed exactly that kind of behavior when following up CHP incident reports in the safety odds browser. A more descriptive error bit would allow us to more intelligently decide whether to keep or reject a loop's measurement, thereby increasing the numbers of odds predictions we can make.

ACKNOWLEDGEMENTS

This research was funded in part by the California Partners for Advanced Transit and Highways (PATH) and the California Department of Transportation (Caltrans). The contents of this paper reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the University of California, California PATH, or the California Department of Transportation.

REFERENCES

Ben-Akiva, M. and Lerman, S.R. (1985). *Discrete Choice Analysis.* MIT Press, Cambridge, MA.

Caltrans (1993). *Manual of Traffic Accident Surveillance and Analysis System.* California Department of Transportation, Sacramento.

Chen, C; Petty, K.F., Skabardonis, A., Varaiya, P.P., and Jia, Z. (2001). Freeway performance measurement system: mining loop detector data. *Transportation Research Record 1748*: 96-102.

Choe, T., Skabardonis, A., and Varaiya, P.P. (2002). Freeway performance measurement system (PeMS): an operational analysis tool. Presented at Annual Meeting of Transportation Research Board, January 13-17, Washington, DC.

FHWA (2000). *Guidebook for the California state Data Files*. Report FHWA-RD-00-137, Federal Highway Administration, U.S. Department of Transportation. http://www.hsisinfo.org/pdf/00-137.pdf (accessed May 23, 2005).

Golob, T.F. and Recker, W.W. (2003). Relationships among urban freeway accidents, traffic flow, weather and lighting conditions. *ASCE Journal of Transportation Engineering,* 129: 342-353.

Golob, T.F. and Recker, W.W. (2004). A Method for relating type of crash to traffic flow characteristics on urban freeways. *Transportation Research, Part A,* 38: 53-80.

Golob, T.F., Recker, W.W. and Alvarez, V.M. (2002). *Freeway Safety as a Function of Traffic Flow: The FITS Tool for Evaluating ATMS Operations.* Final Report prepared for California Partners for Advanced transit and Highways (PATH). Institute of Transportation Studies, University of California, Irvine, CA.

Golob, T.F., Recker, W.W. and Alvarez, V.M. (2004a). A tool to evaluate the safety effects of changes in freeway traffic flow. *ASCE Journal of Transportation Engineering,* 130: 222-230.

Golob, T.F., Recker, W.W. and Alvarez, V.M. (2004b). Freeway safety as a function of traffic flow. *Accident Analysis and Prevention*. 36: 933-946.

Golob, T.F., Recker, W.W. and Pavlis, I. (2007). Probabilistic Models of Freeway Safety Performance using Traffic Flow Data as Predictors. *Safety Science*, in press.

PeMS (2005). Freeway Performance Measurement System. Department of Electrical Engineering and Computer Science, University of California Berkeley http://pems.eecs.berkeley.edu/Public/ (accessed May 11, 2005).

The Apache DB Project (2006). Torque. The Apache Software Foundation, Forest Hill, MD. http://db.apache.org.

Varaiya, P.P. (2001). Freeway Performance Measurement System, PeMS V3, Phase 1: Final Report. Report UCB-ITS-PWP-2001-17, California PATH Program, Institute of Transportation Studies, University of California Berkeley, CA.

Varaiya, P.P. (2005). What we have learned about highway congestion. April 10, 2005. http://paleale.eecs.berkeley.edu/~varaiya/papers_ps.dir/accessF05v2.pdf (accessed May 12, 2005).