Research Report
Agreement T2695, Task 20
AVL Vehicles as Speed Probes

AVL-Equipped Vehicles as Speed Probes

(Phase 2)

By

Daniel J. Dailey, Professor
University of Washington
Dept. of Electrical Engineering
Box 352500
Seattle, Washington 98195-2500

Fredrick W. Cathey, Research Scientist
University of Washington
Dept. of Electrical Engineering
Box 352500
Seattle, Washington 98195-2500

Washington State Transportation Center (TRAC)
Univeristy of Washington, Box 354802
University District Building, Suite 535
1107 N.E. 45th Street
Seattle, Washington 98105-4631

Washington State Department of Transportation
Technical Monitor
Pete Briglia
ITS Program Manager

Sponsored by

Washington State
Transportation Commission
Department of Transportation
Olympia, Washington 98504-7370

Transportation Northwest (TransNow)
University of Washington
135 More Hall, Box 352700
Seattle, Washington 98195-2700

in cooperation with
U.S. Department of Transportation
Federal Highway Administration

December 2003

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO. WA-RD 579.1	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
4. TITLE AND SUBTITLE AVL-Equipped Vehicles as Speed Probes (Phase 2)		5. REPORT DATE December 2003 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Daniel J. Dailey, Fredrick W. Cathey		8. PERFORMING ORGANIZATION REPORT NO.
9. PERFORMING ORGANIZATION NAME AND ADDRESS Washington State Transportation Center (TRAC) University of Washington, Box 354802 University District Building; 1107 NE 45th Street, Suite 535 Seattle, Washington 98105-4631		10. WORK UNIT NO. 11. CONTRACT OR GRANT NO. Agreement T2695, Task 20
12. SPONSORING AGENCY NAME AND ADDRESS Research Office Washington State Department of Transportation Transportation Building, MS 47372 Olympia, Washington 98504-7372 Doug Brodin, Project Manager, 360-705-7972		13. TYPE OF REPORT AND PERIOD COVERED Research Report 14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

This study was conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. ABSTRACT

Performance monitoring is an issue of growing concern both nationally and in Washington State. Travel times and speeds have always been of interest to traveler-information researchers, but as they become a key measure in performance monitoring, this interest is now greater than ever. However, instrumenting the roadway infrastructure to obtain this type of data is very expensive. In this project, we use transit vehicles as probe sensors and develop a framework to use the vehicle position estimates as a speed sensor.

This report documents the second phase of a three-phase project that will create a robust set of virtual sensors for freeways and arterials. The first phase was a proof of principle that examined the statistics of successfully using transit vehicles as traffic probes. The results of the second phase are presented in this report. An optimal filter method is described that estimates acceleration, speed, and position as a function of space and time. The third phase will implement a server to place speed estimates from the transit probe virtual sensors into the WSDOT Northwest Region's operational Traffic Management System.

WSDOT will benefit from this work by gaining additional traffic management sensing capabilities without the additional installation and maintenance costs of cabinets, loops, and communications. The traveling public in metropolitan Seattle will benefit from having additional traveler information about arterials that can be used as alternatives to freeway travel. Publication of results that validate the techniques used to derive virtual sensors from transit probe vehicles will have a national impact as an increasing number of cities use transit fleet management systems.

17. KEY WORDS Bus, transit, probe, Calman filters, Kalman smoother, GIS, AVL, TCIP, MML, virtual sensors, traffic probes, transit probes, speed sensors, geographic information systems, road traffic, traffic control, traffic engineering computing, transportation	18. DISTRIBUTION STATEMENT No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22616

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
None	None		

DISCLAIMER

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. This document is disseminated through the Transportation Northwest (TransNow) Regional Center under the sponsorship of the U.S. Department of Transportation UTC Grant Program and through the Washington State Department of Transportation. The U.S. government assumes no liability for the contents or use thereof. Sponsorship for the local match portion of this research project was provided by the Washington State Department of Transportation. The contents do not necessarily reflect the official views or policies of the U.S. Department of Transportation or Washington State Department of Transportation. This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

Technical Title Page i

Disclaimer iii

Executive Summary ix

1. Theory 1

1.1 Introduction 1
1.2 Transit Database and AVL Data 2
1.3 Kalman Filter Models 4
1.3.1 Determination of Filter Parameters 9
1.4 GIS Index System 10
1.5 Results 12

2. Application 17

2.1 Introduction 17
2.2 Tracker Component 18
2.3 Results 19
2.4 Summary 21

3. Corridor Travel Times 23

3.1 Corridor Concept 23
3.2 Transit Database and AVL Data 24
3.3 Tracker 25
3.4 Corridor Estimator 25
3.4.1 Covering Arcs Builder 26
3.4.2 Corridor Builder 26
3.4.3 Corridor Estimator Component 26
3.5 Applications 28
3.6 Summary 31

References 32

LIST OF FIGURES

Figure 1. Time series of reported distance-into-trip. 4

Figure 2. Time series of distance-into-trip. 8

Figure 3. Time series of estimated speed. 8

Figure 4. Residuals between measurements and estimates. 8

Figure 5. Smoother velocity error estimates. 9

Figure 6. Chain of oriented arcs. 11

Figure 7. Time series of the speed estimate at a single point. 12

Figure 8. Speed as a function of location along I-5. 13

Figure 9. Speed as function of time and distance. 14

Figure 10: Contour plot of speed; darker is slower. 15

Figure 11. Travel time as a function of departure time. 15

Figure 12. Data flow for the ProbeView application. 18

Figure 13. On the left are probe data from Aurora Avenue on Wednesday, June 13, 2001. On the right are probe data from I-5 on Friday, June 14, 2001 19

Figure 14. The left plot shows probe and speed trap data for SR 520 on Wednesday, June 13, 2001. The right plot shows probe and Lane 1 speed trap data on the same day 20

Figure 15. The left side shows probe data from December 17, 2001, when an incident was reported between 17:29 and 18:50. On the right is a map of I-5 North with the sensor locations identified 20

Figure 16. Virtual sensor data from the sensor nearest to an inductance loop location. 21

Figure 17. ProbeView application output. 22

Figure 18. Probe corridor design. 24

Figure 19: Corridor probe trajectories in space (feet into corridor) and time of day (minutes past midnight), with Friday on the left and Monday on the right. 28

Figure 20: Corridor probe speed (miles per hour) surface as a function of time of day (minutes past midnight) and distance-into-corridor (feet), Friday on the left, Monday on the right. 29

Figure 21. Comparison of probe and inductance loop speed-trap speeds (miles per hour) as a function of time of day (minutes past midnight), Friday on the left, Monday on the right. 29

Figure 22. Corridor probe travel time (minutes) as a function of departure time of day (minutes past midnight), Friday on the left, Monday on the right. 30

Figure 23. Corridor probe travel time (minutes) as a function of departure time of day (minutes past midnight) using instantaneous speed measurements, Friday on the left, Monday on the right. 31

EXECUTIVE SUMMARY

This report documents the second phase of a three-phase project that will create a robust set of virtual sensors for freeways and arterials. The first phase was a proof of principle that examined the statistics of successfully using transit vehicles as traffic probes. The results of the second phase are presented in this report. An optimal filter method is described that estimates acceleration, speed, and position as a function of space and time. The third phase will implement a server to place speed estimates from the transit probe virtual sensors into the Northwest Region's operational Traffic Management System.

The overall project will accomplish several goals, it will: (1) create a mass-transit vehicle tracking system that computes smooth estimates of speed and distance traveled for each vehicle in the fleet, (2) create a network of virtual "speed sensors" on selected road segments by using the "probe" vehicle speed estimates produced by the tracking system, (3) make the speed measurements available for traffic analysis purposes and data fusion with CCTV camera and inductance loop measurements, (4) create graphical applications for visualization of current and historical traffic conditions, (5) create tools for predicting point-to-point travel times on the basis of smoothed historical speed data, and (6) create a corridor speed estimate and travel time.

WSDOT will benefit from this work by gaining additional traffic management sensing capabilities without the additional installation and maintenance costs of cabinets, loops, and communications. The traveling public in metropolitan Seattle will benefit from having additional traveler information about arterials that can be used as alternatives to freeway travel. Publication of results that validate the techniques used to derive virtual sensors from transit probe vehicles will have a national impact as an increasing number of cities use transit fleet management systems

1. THEORY

1.1 Introduction

Performance monitoring is an issue of growing concern both nationally and in Washington State. Travel times and speeds have always been of interest to traveler-information researchers, but as a key measure in performance monitoring, this interest is now greater than ever. However, instrumenting the roadway infrastructure to obtain this type of data is very expensive. In this project, we used transit vehicles as probe sensors and developed a framework to use the vehicle position estimates as a speed sensor.

This report documents the second phase of a three-phase project that will create a robust set of virtual sensors for freeways and arterials. The first phase was a proof of principle that examined the statistics of successfully using transit vehicles as traffic probes. The results of the second phase are presented in this report. An optimal filter method is described that estimates acceleration, speed, and position as a function of space and time. The third phase will implement a server to place speed estimates from the transit probe virtual sensors into the Northwest Region’s operational Traffic Management System.

The goals of the “buses as probes” effort by the University of Washington ITS research group include the following:

- Create a mass-transit vehicle tracking system that computes smooth estimates of speed and distance traveled for each vehicle in the fleet.
- Create a network of virtual “speed sensors” on selected road segments by using the “probe” vehicle speed estimates produced by the tracking system.
- Make the speed measurements available for traffic analysis purposes and data fusion with CCTV camera and inductance loop measurements.
- Create graphical applications for visualization of current and historical traffic conditions.
- Create tools for predicting point-to-point travel times on the basis of smoothed historical speed data.

In this report, we describe the use of vehicle locations produced by an automatic vehicle location (AVL) system, the King County Metro Transit AVL System [1], to construct “speed sensors” on every road segment over which transit vehicles travel. We present the resulting estimates of speed and use those data to estimate travel times.

To describe our algorithms, we first define terms and discuss essential concepts related to the transit scheduling system. This background information is necessary to understand the AVL data and present the problem of associating AVL data with an actual road segments.

We present the framework for the measurement and process models of a Kalman filter/smoother that estimates vehicle state (position, speed, and acceleration) from AVL sensor position reports. The parameters for these models, variances of the measurement and process noise, are determined experimentally by using a recently developed maximum marginal likelihood algorithm [2].

To clarify the mapping of speed estimates onto roadways, we describe the geographical road segment database from which the bus schedule database is created. We introduce a geographic information system (GIS) approach for organizing vehicle state estimates according to an “oriented road segment” indexing system.

Finally, we present some example data and applications. In the first example, we select a time and then sample the data at many locations to produce a geographical map of traffic speed at that time. In the second example, we select a location and then sample over time to produce a time series of traffic behavior at a geographic point. The third example samples in both time and space at points along a chain of oriented road segments, and from this we construct a function that gives speed as a bivariate function of time and distance, $dx/dt = f(x,t)$. Using integration techniques, we then estimate travel times between points at various times of day. These examples demonstrate that it is possible to create a speed and travel time sensor by using transit vehicles as probes.

1.2 Transit Database and AVL Data

Our basic assumptions in using a mass transit system as a speed sensor are as follows:

1. There is a fleet of transit vehicles that travel along prescribed routes.
2. A “transit database” defines the schedule times and the geographical layout of every route and time point.

3. An automatic vehicle location (AVL) system equips each vehicle in the fleet with a transmitter that periodically reports its progress back to a transit management center.

To clarify the terminology used here, a description of the AVL data, as well as a conceptual description of relevant elements of the database in terms of ITS Transit Communications Interface Profile (TCIP), is necessary. There are five relevant terms: (1) time-point (TP), (2) time-point-interval (TPI), (3) pattern, (4) trip, and (5) block.

- A *time-point* is a named location. The location is generally defined by two coordinates, either in Cartesian state-plane coordinates (as is the case in King County) or by geodetic latitude and longitude. Transit vehicles are scheduled to arrive or depart time-points at various times.
- A *time-point-interval* (TPI) is a named polygonal path directed from one time-point to another. The path is geographically defined by a list of "shape-points," a shape-point being simply an unnamed location. Since one frequently needs to determine the distance of a vehicle along a path, each shape-point is often augmented with its own distance into a path.
- A *pattern* is a route specified by a sequence of TPIs, the ending time-point of the i^{th} TPI being the starting time-point of the $(i+1)^{st}$.
- A *trip* is a pattern with an assignment of schedule times to each of the time-points on the pattern.
- A *block* is a sequence of trips. Each transit vehicle is assigned a block to follow over the course of the day. Some blocks are long and are covered by different vehicles at different times of the day.

A typical AVL system will produce real-time reports of vehicle location based on technologies such as dead reckoning or satellite GPS position measurements. The King County Metro AVL system is based on odometer measurements. Real-time reports from this system are freely available in the Self Describing Data (SDD) format [3] from UW host "sdd" on port 8412. Archived data are available for post-processing at http://avllog.its.washington.edu:8080/. The Metro AVL system produces a vehicle location report that includes the following information:

- vehicle-identifier

- pattern-identifier
- time
- distance-into-pattern.

The vehicle-identifier is required to correlate the report with the correct vehicle track. As we will show in Section 1.4, the pattern-identifier provides the link between vehicle distance-into-pattern and vehicle position on an identifiable road, permitting the geographical indexing of estimated speed.

Figure 1 shows a time-series plot of reported distance-into-pattern for a vehicle traveling along its first trip of the day. Here, time is measured in minutes (min) after midnight and distance in feet. The circles represent scheduled time-points and the continuous curve rising from left to right is the polygonal line joining these points. The short horizontal line segments are drawn to help visualize estimated schedule deviations of the vehicle.

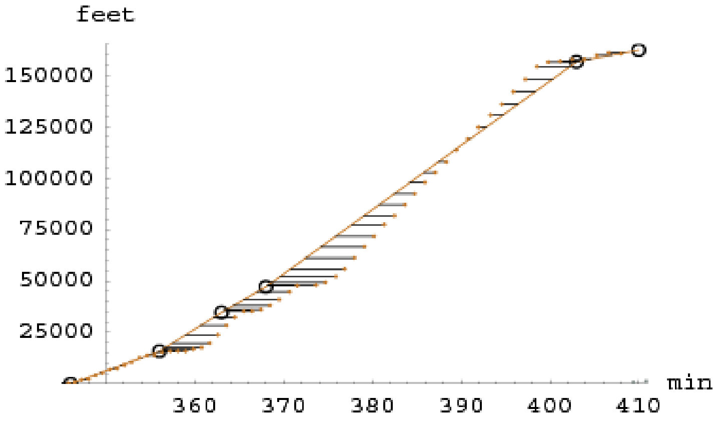

Figure 1. Time series of reported distance-into-trip.

1.3 Kalman Filter Models

We use a Kalman filter/smoother to transform a sequence of AVL measurements into estimates of vehicle dynamical state, including vehicle speed. In this section, we describe the measurement and process models used in the filter framework. The models depend on several parameters, including the variances for measurement and process noise. We estimated representative values experimentally by using the method of maximum marginal likelihood as specified by Bell [2], and described in Section 1.3.1 below.

To implement a Kalman filter, the following must be specified:

- a state-space
- a measurement model
- a state transition model
- an initialization procedure.

Once these items have been specified, one may employ any one of a number of implementations of the Kalman filter/smoother equations (see Bell [2], Jazwinski [4], Rauch et al. [5], or Anderson and Moore [6]) to transform a sequence of measurements into a sequence of vehicle state estimates.

To represent the instantaneous dynamical state of a vehicle, we selected a three-dimensional state space. We denote a vehicle state vector by

$$X = \begin{pmatrix} x & v & a \end{pmatrix}^T , \tag{1}$$

where x is distance-into-pattern, v is speed, and a is acceleration. (The superscript T denotes transpose.) We use the foot as unit of distance and minute as unit of time.

A measurement, z, provided by the AVL system, is simply a noisy estimate of the vehicle's distance into trip, and our measurement model is given by

$$z = HX + e = x + e . \tag{2}$$

Here, $H = (1 \quad 0 \quad 0)$ is the "measurement matrix," and e denotes a random measurement error, assumed to have a Normal distribution with variance R. The variance is treated as a model parameter with a nominal value of R = $(500 \text{ ft})^2$.

We assume a simple dynamics for evolution of state, defined by the first order system of linear stochastic differential equations

$$\begin{aligned} dx &= v \; dt \\ dv &= a \; dt \; . \\ da &= dw \end{aligned} \tag{3}$$

Here, dt is the differential of time and dw is the differential of Brownian motion, representing randomness in vehicle acceleration. By definition of Brownian motion (see Chapter 3, Section 5 of Jazwinski [4]), the expectation is

$$E(dw^2) = q^2 dt , \tag{4}$$

where q^2 is a model parameter. In the absence of a measurement correction, the variance of acceleration grows linearly with time. We selected a value for $q^2$of (264 ft/min^2)^2/min = (3 mph/min)^2/min.

The differential equations, (3), are written in vector form as follows,

$$dX = F\ X\ dt + G\ dw, \tag{5}$$

where

$$F = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad G = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{6}$$

Integrating over a time interval $(t, t+\delta t)$, we obtain the state transition model

$$X(t+\delta t) = \Phi(\delta t) X(t) + W(\delta t), \tag{7}$$

where $X(t)$ and $X(t+\delta t)$ denote vehicle state values at times t and $t + \delta t$, respectively.

$$\Phi(\delta t) = \exp(F\ \delta t) = \begin{pmatrix} 1 & \delta t & \delta t^2/2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{pmatrix} \tag{8}$$

is the “transition matrix” and also where the accumulated error $W(\delta t)$ has covariance

$$Q(\delta t) = \begin{pmatrix} \delta t^5/20 & \delta t^4/8 & \delta t^3/6 \\ \delta t^4/8 & \delta t^3/3 & \delta t^2/2 \\ \delta t^3/6 & \delta t^2/2 & \delta t \end{pmatrix} q^2. \tag{9}$$

See the text following Theorem 7.1 of Jazwinski [4] for a discussion of integration.

Finally, to run a Kalman filter/smoother, we need an initialization procedure, a method for computing an initial value for the state vector and its associated error covariance matrix. These initial values, $\hat{X}_0$ and P_0, are based on the initial measurement z_0 (at time t_0) and measurement variance R. We set $\hat{X}_0 = (z_0 \quad 0 \quad 0)$ and set P_0 equal to a diagonal matrix with entries R, (30 mph)^2, and (16 mph/min)^2. The initialization data above are specified by a number of “hard-coded” parameters. One could attempt to determine their “optimal” values experimentally using the techniques of Section 1.3.1, but we did not do so.

Now, given a sequence of measurements $z_1, z_2, \ldots z_N$ at times $t_1, t_2, \ldots t_N$, the Kalman filter transition equations are

$$\begin{aligned} \hat{X}_k^- &= \Phi_k \hat{X}_{k-1} \\ P_k^- &= \Phi_k P_{k-1} \Phi_k^T + Q_k \end{aligned}, \tag{10}$$

where $\hat{X}_k^-$ is the prediction of the state vector at the k^{th} step, Φ_k is the transition matrix between the k-1 to the k^{th} step, and Q_k is the corresponding transition covariance matrix. The data update equations are

$$\begin{aligned} \hat{X}_k &= \hat{X}_k^- + K_k \left(z_k - H\hat{X}_k^- \right) \\ P_k &= \left(I - K_k H \right) P_k^- \end{aligned}, \tag{11}$$

where

$$K_k = P_k^- H^T \left(HP_k^- H^T + R \right)^{-1}. \tag{12}$$

The equations for the smoothed estimates of state $\overline{X}_k$ for k = 1…N are given by

$$\overline{X}_{k-1} = \hat{X}_{k-1} + P_{k-1}\Phi_k^T \left(P_k^- \right)^{-1} \left(\overline{X}_k - \hat{X}_k^- \right), \tag{13}$$

where $\overline{X}_N \hat{X}_N$. See Rauch et al. [5] for a derivation of these Kalman filter/smoother formulas.

The particular implementation of the Kalman filter/smoother that we used is identified as Algorithm 4 of Bell [2]. This algorithm, in addition to computing the smoothed estimates of state, also computes the marginal likelihood of the measurement sequence. This algorithm is appropriate for our application, namely the post-processing of historical data and the determination of filter parameters, but for a real-time application, we would use an ordinary Kalman filter.

Figures 2 and 3 show example time-series plots of vehicle distance-into-trip and speed, in miles per hour (mph), as produced by a Kalman smoother. Figure 4 is a plot of residuals, the measurement minus the estimate. The straight horizontal lines indicate the 500-foot measurement uncertainty, and the slightly slanted horizontal lines show the 1-sigma distance error estimate computed by the smoother. Figure 5 shows a plot of the 1-sigma speed error estimate computed by the smoother. The AVL reports for this example span a time-point-interval containing a stretch of freeway.

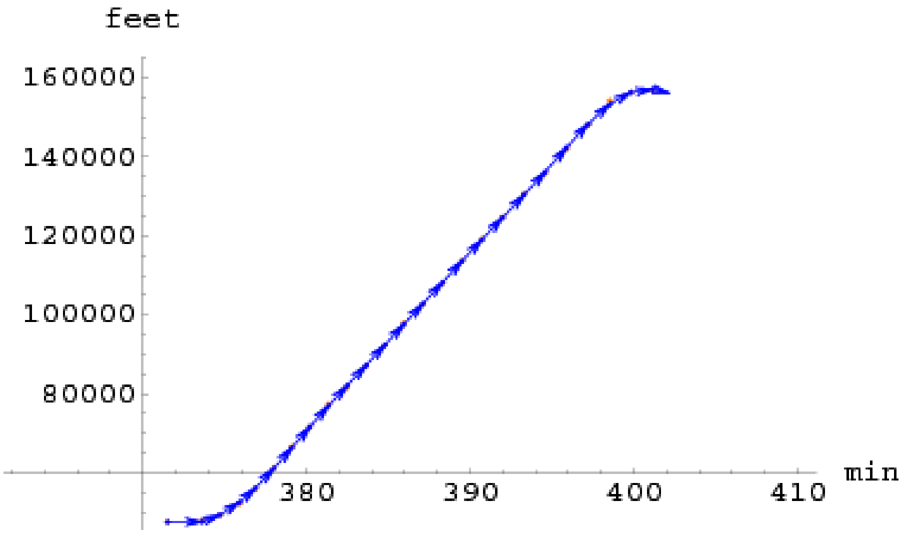

Figure 2. Time series of distance-into-trip.

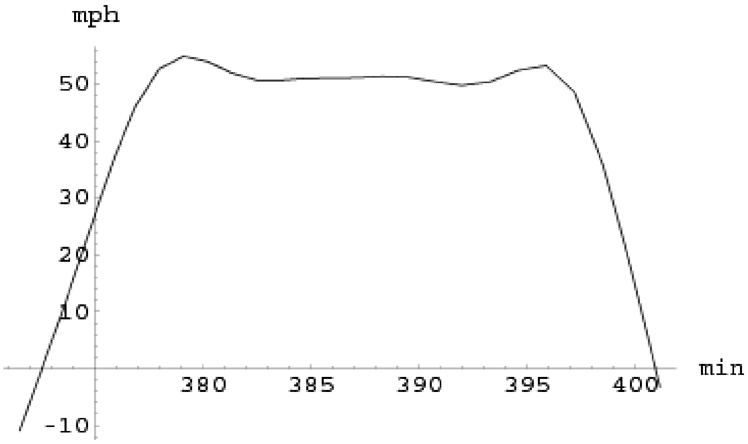

Figure 3. Time series of estimated speed.

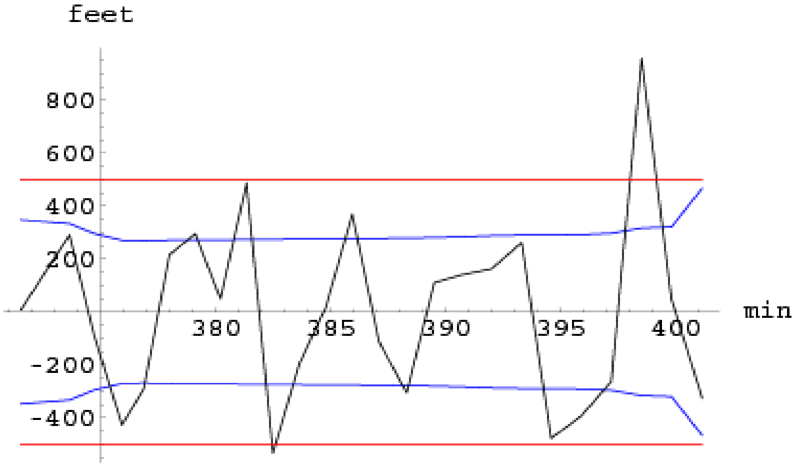

Figure 4. Residuals between measurements and estimates.

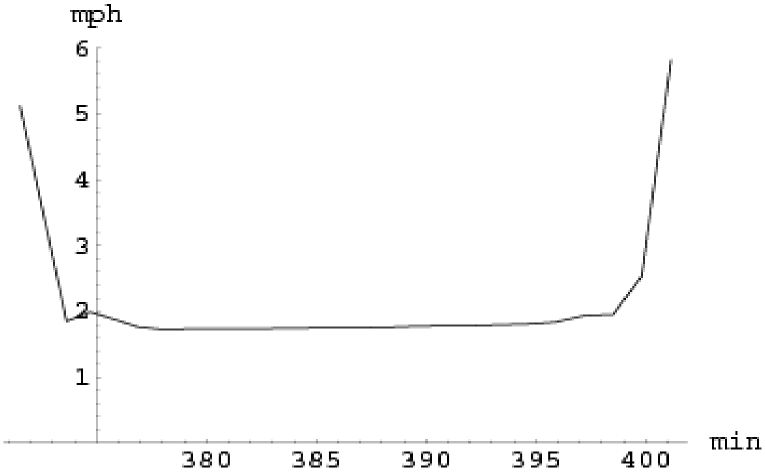

Figure 5. Smoother velocity error estimates.

1.3.1 Determination of Filter Parameters

As pointed out above, our measurement and process models depend on two parameters: the measurement variance, R, and the rate of change of the variance of process noise, q^2. By using the method of "maximum marginal likelihood," we obtained "optimal" values for these parameters in a number of experiments with different measurement sequences. Our goal was not to perform an exhaustive statistical analysis but rather just to find some representative values that would give reasonable filter performance. We observed that the values obtained in each experiment were roughly the same, i.e., fluctuated around $R = (500\text{ ft})^2$ and $q^2 = (3\text{ mph/min})^2/\text{min}$.

Although the method of maximum marginal likelihood for parameter estimation is well known to statisticians, the fact that it can be used effectively for estimating parameters in the setting of Kalman-Bucy filters is not widely reported. This method was first proposed by Bell [2], who provided an effective procedure for evaluating the marginal likelihood function. We briefly describe the theory.

Let **Z** and **X** denote vector-valued random variables representing a measurement sequence and a corresponding sequence of state vectors, and let ξ denote the parameter vector (R, q^2). A formula is given in Equation 4 of Bell [2] for the joint probability density function $p_{\mathbf{Z},\mathbf{X}}(z, w; \xi)$ in terms of the measurement and process models like those described above. (The symbols z and x in this context denote real-valued vectors, and usage is not to be confused with that in the preceding section.) The cited reference provides an algorithm

(Algorithm 4) which, when given a measurement vector *z* and parameter vector ξ, simultaneously computes the maximum likelihood estimate for the state vector sequence (the Kalman smoother estimate)

$$x^*_{z,\xi} = \underset{x}{\arg\max} \left\{ P_{\mathbf{x}|\mathbf{z}}(x|z;\xi) \right\}, \tag{14}$$

and evaluates the marginal density function

$$p_{\mathbf{z}}(z;\xi) = \int p_{\mathbf{z},\mathbf{x}}(z,x;\xi)dx. \tag{15}$$

As usual, the algorithm actually works in terms of the negative logs of the various probability densities.

In each of our experiments, we used the cited algorithm to define an objective function $f_z(\xi) = -\log(p_{\mathbf{z}}(z;\xi))$, depending on the measurement sequence *z*, and then used Powell's conjugate direction minimization algorithm (Chapter 10, Section 5 of Press et al. [7]) to find the optimal parameter values for each measurement sequence *z* ,

$$\xi^*_z = \underset{\xi}{\arg\min} \left\{ f_z(\xi) \right\}. \tag{16}$$

1.4 GIS Index System

In this section, we describe a geographical system for indexing traffic data, and we describe a procedure for assigning probe vehicle state data to appropriate indices. One may imagine a virtual speed sensor located at each index that provides a speed measurement whenever a probe vehicle passes by. The indexing system described here uses the GIS road-segment database that underlies the construction of the bus schedule time-points and time-point-intervals described in Section 1.2. We note that the database for King County is based on U.S. Census Bureau TIGER files.

The GIS road-segment database is an instance of a mathematical structure called a "directed graph." A set of "nodes" represents points on roads. and a set of "arcs" represents segments of the road between nodes. Like a time-point-interval, an arc has a start node, an end node, and a sequence of shape points that define a polygonal path from start to end. Unlike time-point-intervals, however, there is never more that one arc connecting two nodes. In particular, if an arc is directed from node A to node B, then no arc is directed back from B to A.

We define a GIS index for traffic data to be a triple, that consists of an arc, an orientation, and a selected point. The orientation is +1 if the direction of traffic is along the direction of the arc and is -1 otherwise. Points are selected symmetrically on each arc to achieve a nominal spacing on long arcs of between 2,000 and 3,000 feet. The following rule was used: if the length *L* of an arc is such that $L \le d = 200 \text{ feet}$, then the center point of the arc is selected; otherwise, *k* points are chosen with distance into arc $d_1/2, 3d_1/2, \ldots (2k-1)d_1/2 = L - d_1/2$, where the integer *k* is chosen so $L/d \le k < L/d + 1$ and where $d_1 = L/k$.

Given a sequence of time-tagged vehicle state estimates along a scheduled pattern, we can interpolate the time and speed at any intermediate distance-into-pattern. In order to associate vehicle state data with GIS indices, we require that, for every pattern, there be a correspondence between the GIS indices and pattern distances. Because every pattern is a list of time-point-intervals (TPI), it suffices to have a correspondence between the GIS indices and TPI distances. To construct such a correspondence, we exploit the known relationship between the GIS database and the transit database:

- each time-point is a GIS node
- each time-point-interval is the result of "welding together" a chain of oriented GIS arcs. In Figure 6 each arc is labeled with an "A" and the directionality of the arc is indicated by the sign, plus to the right and minus to the left, and further the nodes are labeled with an "N."

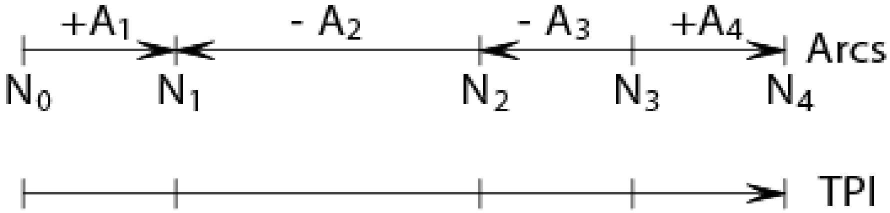

Figure 6. Chain of oriented arcs.

The bus schedule database does not identify the chain of GIS arcs used to construct a TPI. The method we used to determine this information is described by the following algorithm.

First, determine the unique GIS node, N_0, with the same position as the TPI start time-point. Also determine the unique GIS arc, A_1, which either starts or ends at N_0 and whose other node, N_1, has the same position as a TPI shape-point farther down the list. If this arc is directed from N_0 to N_1, assign it the positive orientation, otherwise the negative orientation.

Assume inductively that $i > 0$ and that we have determined the i^{th} node, N_i, and oriented arc, $\pm A_i$, in the chain. Iterate the following step until N_i has the same position as the TPI end time-point.

Find the unique arc, A_{i+1}, which either starts or ends at N_i and whose other node, N_{i+1}, has the same position as a TPI shape-point farther down the list. If this arc is directed from N_i to N_{i+1}, assign it the positive orientation, otherwise the negative. Set $i := i+1$.

We maintained the GIS nodes in a hash table keyed by position (with integer coordinates) in order to efficiently look up a node given its position. In addition, each node had an associated list of incident arcs so that we could quickly determine which arcs started or ended at that node.

1.5 Results

Figure 7 shows a time series plot of speed collected from 222 vehicle passages on a weekday at a single GIS index. The continuous curve was fitted to the data by using the seven-point local smoothing procedure described in Section (6) of Efron [8]. This particular index corresponds to a GIS arc which, for the weekday schedule, lies on 19 TPIs, 36 patterns, 235 trips, and 143 blocks. With perfect coverage by the AVL system, one would expect 235 samples at this index.

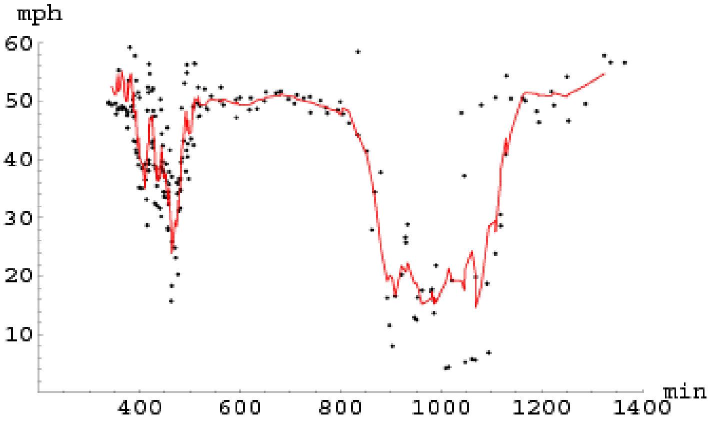

Figure 7. Time series of the speed estimate at a single point.

In Figure 8, dots are at the locations where pseudo speed sensors are placed on a northbound stretch of the I-5 freeway in Seattle. Next to each of the psudeo speed sensors is the speed estimate at 4:40 PM on a weekday. To produce this plot, the most recent observations of speed were selected at GIS indices that had data within the last 15 minutes. At this time of day, the number of samples in a 15-minute window varied from 1 to 4. (During the morning rush hour, the number of samples observed was as high as 16.)

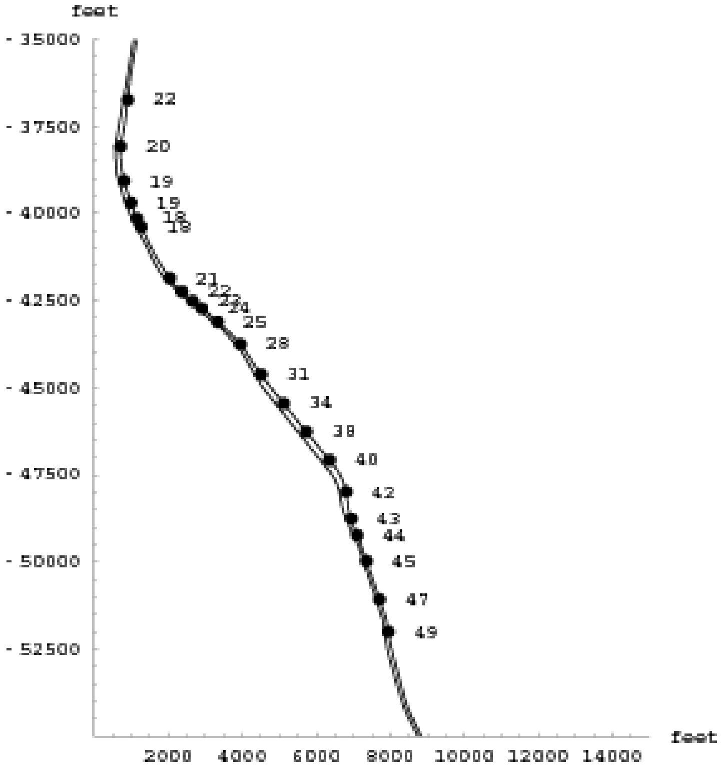

Figure 8. Speed as a function of location along I-5.

Figure 9 shows a surface plot of speed as a function of time and distance, $v = f(x,t)$, along a 2-mile stretch of the I-5 freeway. The function was defined by two-dimensional interpolation of all speed values collected at GIS indices on a chain of arcs defining the freeway.

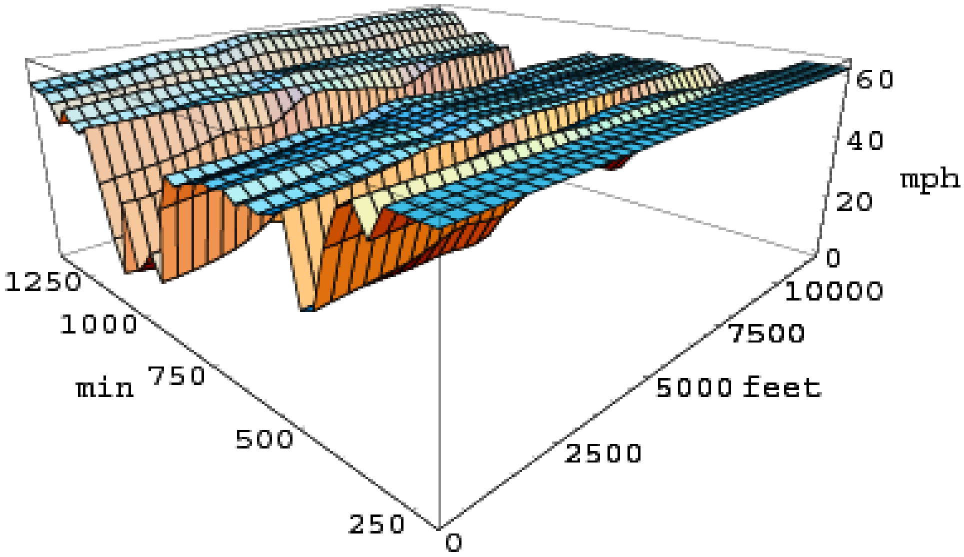

Figure 9. Speed as function of time and distance.

Figure 10 shows a contour plot of the speed as a function of time and space, where the darker internal regions are slower speeds. To estimate travel times given this speed function, the ordinary differential equation ,

$$\frac{dx}{dt} = f(x,t), \tag{17}$$

is solved numerically using Euler's method. The two heavy white lines and the central heavy black line in Figure 10 are the trajectories of three solutions, and the shape of the trajectory depends heavily on the shape of the speed function. In particular, note the character of the bottom heavy white line that traverses a period and region that have slow speeds. This demonstrates that to accurately estimate travel time, speed must be an explicit function of space and time. For example, to estimate the travel time between two points as a function of time, we select a start time t_0 and solve the ODE for time t_1 subject to constraints $x(t_0) = 0$ feet and $x(t_1) = 11{,}000$ feet to obtain travel time $t_1 - t_0$. Figure 11 shows a plot of the travel times for this stretch of road as a function of departure time. The largest travel time peak, found at 1,050 minutes, is associated with the bottom solution trajectory in Figure 10.

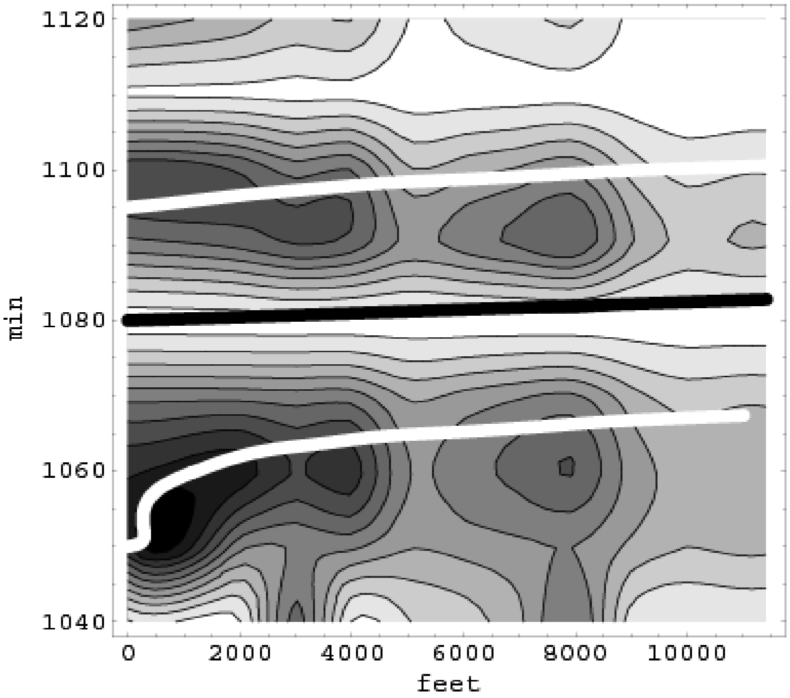

Figure 10: Contour plot of speed; darker is slower.

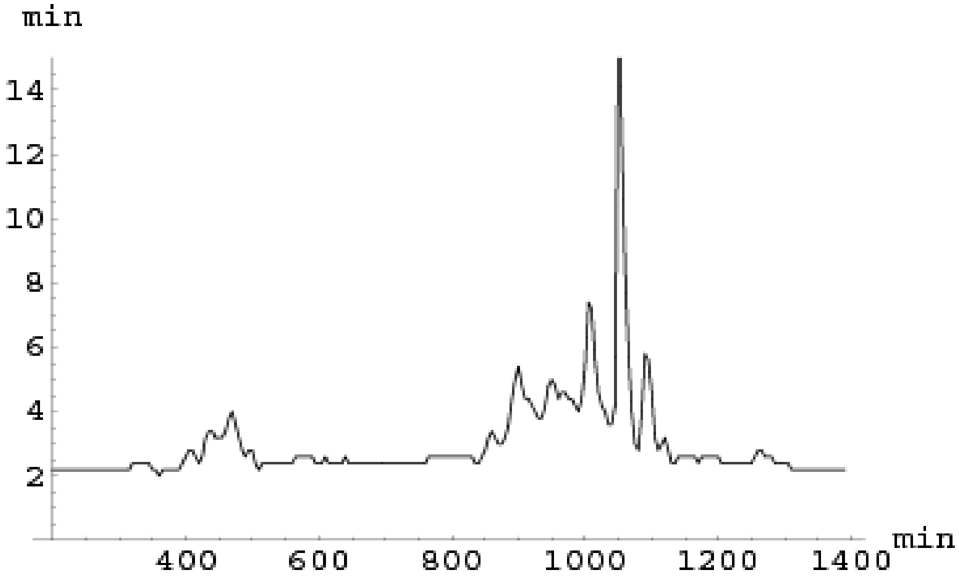

Figure 11. Travel time as a function of departure time.

2. APPLICATION

2.1 Introduction

This section describes how data from a transit management automatic vehicle location (AVL) system, the King County Metro Transit AVL System [1], can be used to construct “virtual sensors” that produce smoothed speed estimates. While the King County Metro AVL system uses a “dead reckoning” method to determine vehicle location, our probe sensor framework can also be used with AVL systems that employ the Global Positioning System (GPS) [9].

The “virtual sensor” system provides travel time and speed measurements at user-defined “probe sensor locations” on both arterials and freeways throughout King County, Washington. These measurements are made readily available in a Self Describing Data (SDD) stream [3]. Depending on the probe sensor location and time of day, reported speeds may or may not reflect surrounding traffic conditions. The speeds of transit vehicles in high occupancy vehicle (HOV) lanes on freeways and major arterials are generally greater than the speed of surrounding traffic. On the other hand, average transit speeds on arterials with bus stops are generally somewhat less. Careful selection of probe sensor locations and an understanding of the dependencies of probe data on traffic conditions are used to produce a system that benefits traffic managers, transit operators, and developers of traveler information systems by producing dependable speed estimates.

Figure 12 is a data-flow diagram of the deployed system’s architecture. The basic components are (1) a Tracker, (2) a Probe Speed Estimator, and (3) a display application, ProbeView. The Tracker process receives the real-time stream of AVL position reports and outputs a corresponding stream of “Track” reports, including filtered vehicle position and speed estimates. The Probes process receives the track data and outputs probe speed reports when vehicles cross specified locations. The data flow between the components adheres to the (SDD) [3].

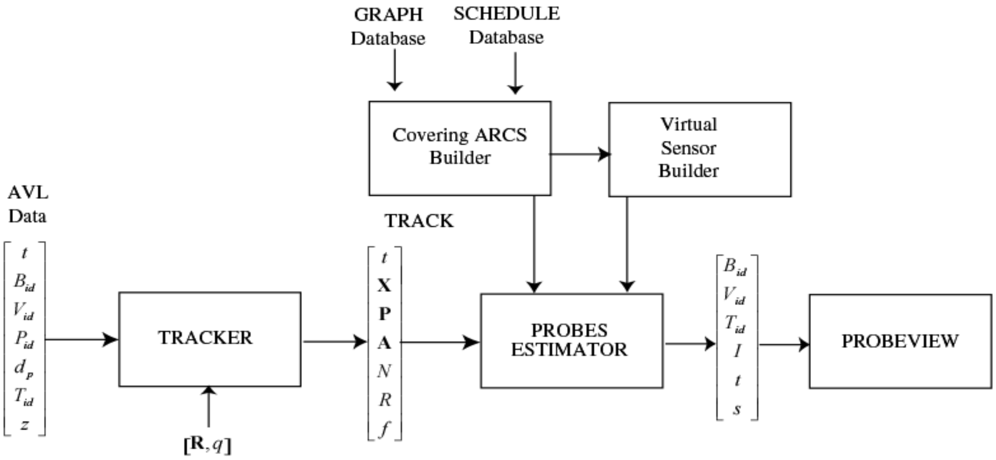
GRAPH Database
SCHEDULE Database
Covering ARCS Builder
Virtual Sensor Builder
AVL Data
t B_{id} V_{id} P_{id} d_p T_{id} z
TRACKER
$[\mathbf{R}, q]$
TRACK
t $\mathbf{X}$ $\mathbf{P}$ $\mathbf{A}$ N R f
PROBES ESTIMATOR
B_{id} V_{id} T_{id} I t s
PROBEVIEW

Figure 12. Data flow for the ProbeView application.

In the next sections, we describe the logic of the Tracker and present in more detail the framework for the measurement and process models of the Kalman filter/smoother that estimates vehicle state (position, speed, and acceleration) from AVL position reports. Finally, we present some examples. The first example shows the quality of the probe speed estimates at several locations. The second example compares inductance loop speed trap data with probe data at several locations and shows that probe sensor output is similar to that of loops. In the third example, probe speed samples are collected during periods of normal traffic flow and during abnormal non-recurring congestion due to incidents. These examples demonstrate that it is possible to create a speed and travel time virtual sensor using transit vehicles as probes.

2.2 Tracker Component

The purpose of the Tracker component is to filter the stream of AVL reports and provide smoothed estimates of vehicle location and speed. In order to perform its task, the Tracker maintains an internal “Track" data structure for each block identified in the transit schedule database. A Track consists of (1) time, t; (2) Kalman filter state, $\mathbf{X}$, and covariance, $\mathbf{P}$; (3) last AVL report, $\mathbf{A}$; (4) number of updates, N; (5) number of consecutive rejected reports, R; and (6) speed validity flag, f. For each AVL report received, the Tracker determines the track with a corresponding block-identifier and then takes one of three actions: Track *initialization*, Track *update*, or report *rejection*.

2.3 Results

We first present examples of the data coming from the virtual speed sensors, created with the transit probes, for both a freeway and a major arterial. The left of Figure 13 shows superpositions of smoothed (the heavy line) and raw (the lighter line) probe speed data collected from a virtual sensor on the arterial Aurora Avenue North. Note that transit vehicle traffic appears bimodal in that it maintains an average speed of about 40 MPH throughout the day, with the exception of the period 1,100 minutes past midnight (MPM) (6:20 PM) to 1,150 MPM (7:10 PM). Similar performance is observed on all other days.

The right of Figure 13 shows a superposition of smoothed and raw probe data on Friday, June 15, 2001, at probe location 417 on I-5 South in Tukwila. Once again there is bimodal behavior consisting of a normal operating speed, approximately 50 MPH throughout the day, and a lower speed mode, approximately 20 MPH between 950 MPM (3:50 PM) and 1,075 MPM (5:55 PM), which corresponds to the afternoon peak on that facility.

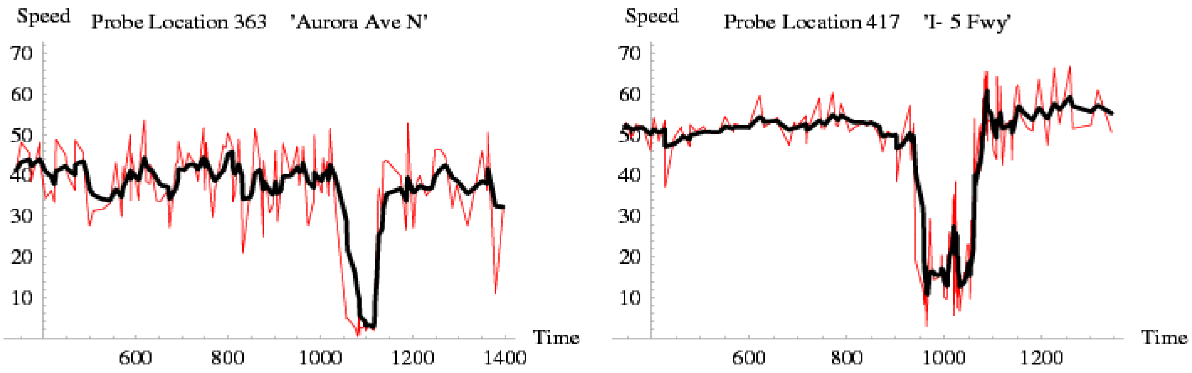

Figure 13. On the left are probe data from Aurora Avenue on Wednesday, June 13, 2001. On the right are probe data from I-5 on Friday, June 14, 2001.

The left of Figure 14 shows probe data (the heavy line) superimposed over speed trap-data for three lanes of west-bound traffic at cabinet ES-520 on State Route 520 just west of Lake Washington Boulevard NE on Wednesday, June 13, 2001. The right of Figure 14 shows that the probe data compare most closely to speed trap data for lane one. The median offset between probe and trap speeds was less than 1 mile per hour on all days examined. Again, ground truth was unavailable, but the example demonstrates the viability of using virtual sensors to create data similar in character to inductance loop data.

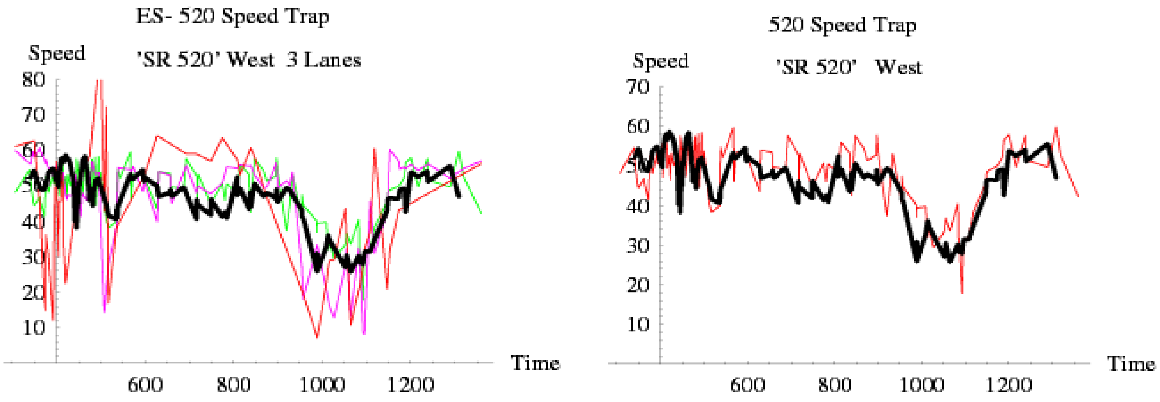

Figure 14. The left plot shows probe and speed trap data for SR 520 on Wednesday, June 13, 2001. The right plot shows probe and Lane 1 speed trap data on the same day.

The right side of Figure 15 shows the locations of virtual sensors 399, 24, 420, and 15 on I-5 North in Seattle, from south to north respectively. The incident identified by WSDOT camera operators is at North 85th Street, and virtual sensor 15 coincides with an inductance loop location. On the left of Figure 15 are the time histories of the four virtual sensors upstream of the incident that began at 17:29 and ended at 18:50. The virtual sensors clearly identify the incident.

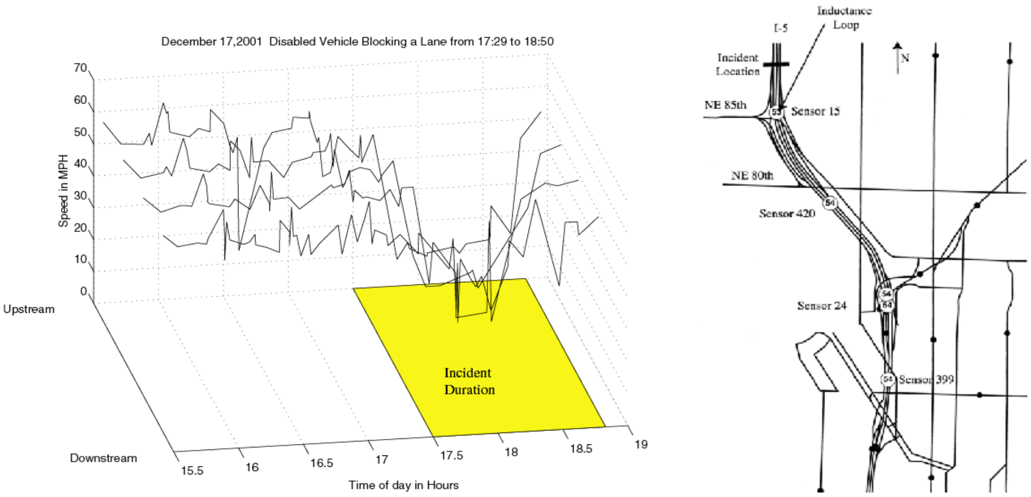

Figure 15. The left side shows probe data from December 17, 2001, when an incident was reported between 17:29 and 18:50. On the right is a map of I-5 North with the sensor locations identified.

Figure 16 is a comparison of the speed estimated by the inductance loop and that estimated by the virtual sensor derived from transit probes. It is clear from this plot that the virtual sensors, if properly deployed, can create a data stream similar in character to inductance loop data for similar traffic conditions. Figure 17 shows the output of the ProbeView application

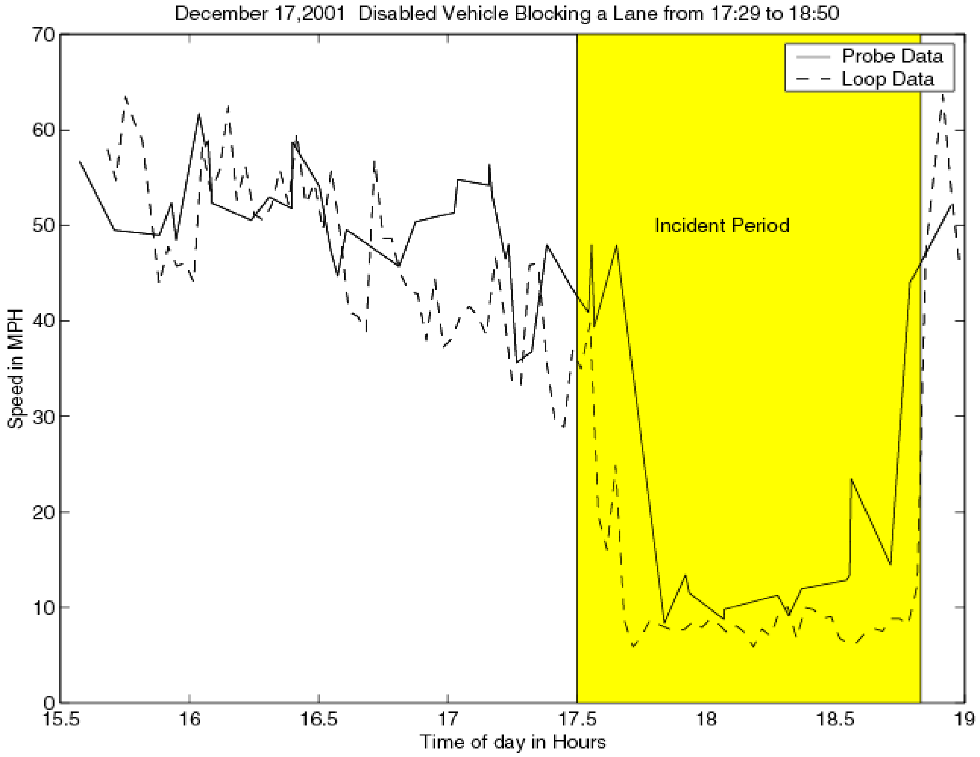

Figure 16. Virtual sensor data from the sensor nearest to an inductance loop location.

2.4 Summary

In this section describe a mass transit tracking system based on AVL data and a Kalman filter to estimate vehicle position and speed. We also describe a system of “virtual” probe sensors that measure transit vehicle speeds using AVL data. Probe sensor data for the King County Washington, area are made readily available to researchers, planners, and developers in a Self Describing Data (SDD) stream (see http://www.its.washington.edu/backbone).

Examples showing the correlation between probe data and inductance loop data demonstrate the viability of the approach and suggest that judiciously selected virtual sensors on freeways and arterials can extend the limited availability of speed traps.

Future research should include the estimation of filter parameters for specific arterials and evaluation of probe speed and travel time measurements with respect to ground truth. The relationship between probe sensor data and the surrounding traffic conditions needs be explored in more detail to allow for the specification of probe placement and density that will ensure accurate traffic data. In addition, methodologies to relate speed and travel times derived from probe sensors and inductance loops need to be developed with an eye toward data fusion of these two sources and the development of traveler information applications.

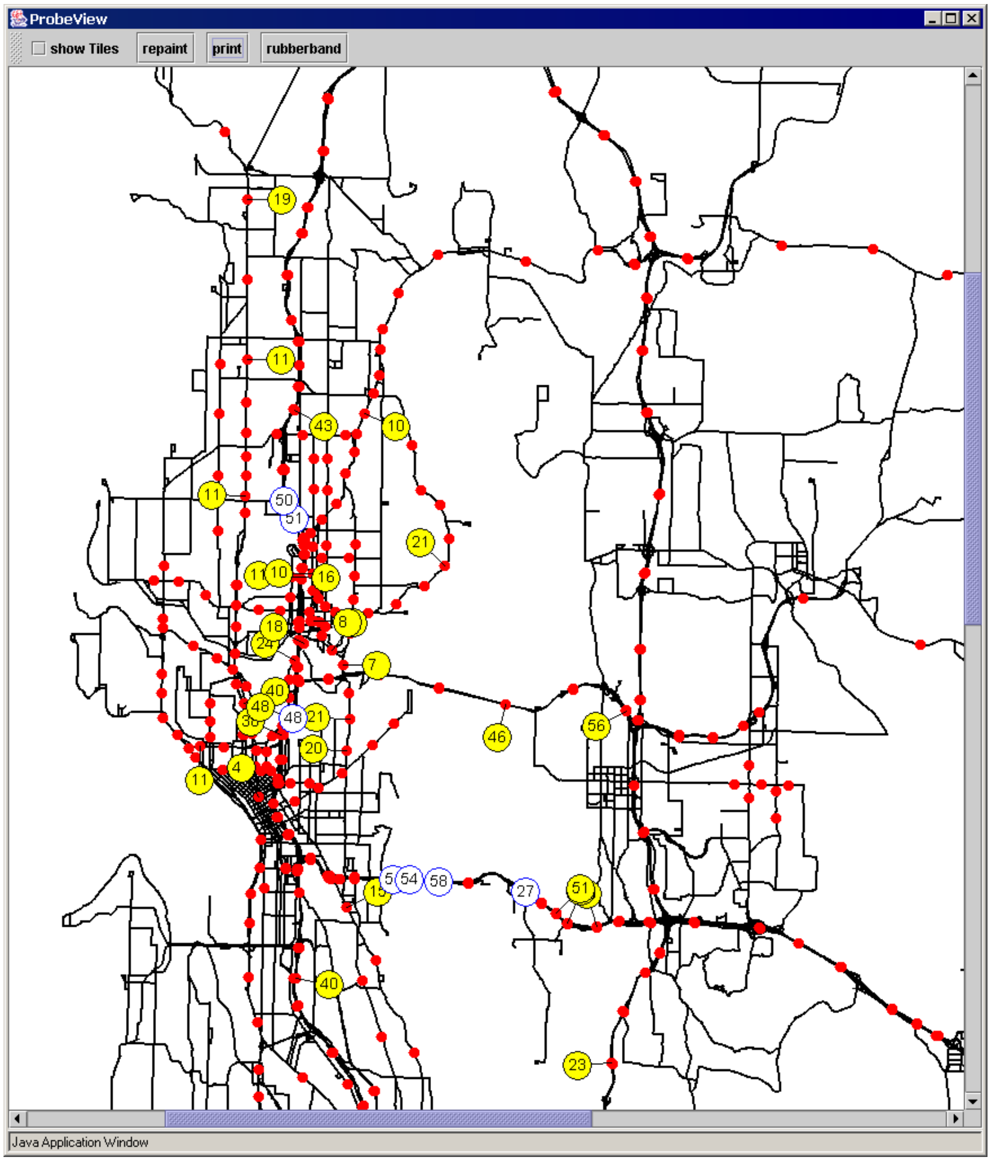
ProbeView
☐ show Tiles
repaint
print
rubberband
19
11
43
10
11
50
51
21
11
10
16
18
24
8
7
40
48
36
48
21
20
46
56
4
11
54
58
15
27
51
40
23
Java Application Window

Figure 17. ProbeView application output.

3. CORRIDOR TRAVEL TIMES

3.1 Corridor Concept

In this section we present a second approach to measuring traffic speed, unlike the preceding approach where the user specified “virtual speed sensors” on various roadway segments, and transit vehicle speeds were determined whenever a vehicle crossed a virtual sensor location. In the approach described here, the user defines a set of one or more roadway corridors, and transit vehicle speed and position along the corridor are determined for every vehicle on the corridor. This scheme provides a richer data set and is more amenable to travel time calculations.

We again use the data from a transit management automatic vehicle location (AVL) system, the King County Metro Transit AVL System [1], to construct a system of probes that produce smoothed speed estimates along specified corridors. While the King County Metro AVL system uses a “dead reckoning” method to determine vehicle location, our probe framework can also be used with AVL systems that employ the Global Positioning System (GPS) [9].

The corridor probe system provides speed measurements along user-defined corridors on both arterials and freeways throughout King County, Washington. These measurements are made readily available in a Self Describing Data (SDD) stream [3]. Depending on the probe location and time of day, reported speeds may or may not reflect surrounding traffic conditions. The speeds of transit vehicles in high occupancy vehicle (HOV) lanes on freeways and major arterials are generally greater than the speed of surrounding traffic. On the other hand, average transit speeds on arterials with bus stops is generally somewhat less. Collation and filtering of probe reports, together with an understanding of the dependencies of probe data on traffic conditions, will lead to a system that benefits traffic managers, transit operators, and developers of traveler information systems.

Figure 18 is a data-flow diagram of the deployed system’s architecture. The basic components are: (1) a Tracker, (2) a Corridor Estimator, and (3) Travel Time Applications. The Tracker receives the real-time stream of AVL position reports and outputs a corresponding stream of “Track” reports. Each track includes filtered estimates of speed and vehicle position with respect to its scheduled path. The Corridor Estimator receives the track data and outputs probe speed reports for vehicles on specified corridors. Its primary task is to parameterize track location as distance along the corridor. Two Corridor Travel Time applications are currently being developed : (a) computation of historical averages of travel times, and (b) prediction of current travel times. The data flow between the components adheres to the (SDD) [3].

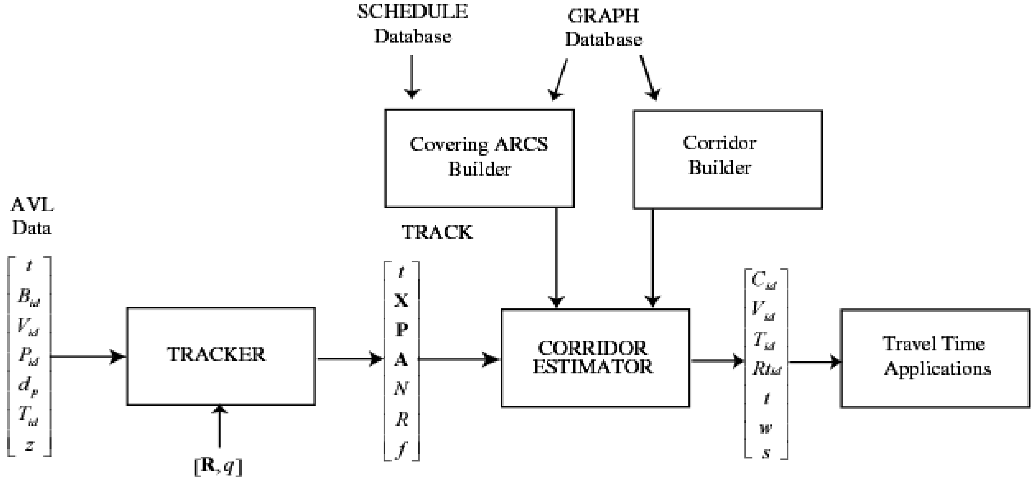
SCHEDULE Database
GRAPH Database
Covering ARCS Builder
Corridor Builder
AVL Data
t
B_{id}
V_{id}
P_{id}
d_p
T_{id}
z
TRACKER
[R, q]
TRACK
t
X
P
A
N
R
f
CORRIDOR ESTIMATOR
C_{id}
V_{id}
T_{id}
Rt_{id}
t
w
s
Travel Time Applications

Figure 18. Probe corridor design.

We will first define terms and discuss essential concepts related to the transit scheduling system. This background information is necessary to understand the AVL data and to present the problem of associating AVL data with actual road segments. We briefly describe the Tracker and present the framework for the Corridor Estimator in more detail. Finally, we present some examples.

3.2 Transit Database and AVL Data

Our basic assumptions in using mass transit system vehicles as speed probes are (1) there is a fleet of transit vehicles that travel along prescribed routes, (2) a “transit database” defines the schedule times and the geographical layout of every route and time point, and (3) an automatic vehicle location (AVL) system equips each vehicle in the fleet with a transmitter that periodically reports its progress back to a transit management center.

As before a vehicle report produced by the King County Metro AVL system includes the following information: (1) time, t; (2) block-identifier, B_{id}; (3) vehicle-identifier, V_{id}; (4) pattern-identifier, P_{id}; (5) distance-into-pattern, d_p; and (6) trip-identifier, T_{id}. We repeat these definitions in order to assign variable names to each value. In addition, for tracking purposes it is convenient to identify how far the vehicle has traveled into its block. Therefore, in software upstream of the Tracker, we augment the information included in the AVL report with distance-into-block, z. Distance-into-block is simply the current distance-into-pattern plus the sum of the lengths of the preceding patterns on the block.

The King County Metro AVL system is based on a dead reckoning technique that uses odometer measurements with position corrections whenever the vehicle passes a a small radio beacon Metro calls a "sign-post." However, the vehicle positioning technique is not critical, and GPS-based AVL systems can also be used to generate the required information, as shown by Cathey and Dailey [9]. The AVL reports are created for each transit vehicle traveling the roadways and are transmitted at a nominal rate of once per minute. These reports are the dynamic input data for the Tracker that is the first of the components used to produce the corridor speed data.

3.3 Tracker

As before the purpose of the Tracker component is to filter the stream of AVL reports and provide smoothed estimates of vehicle location and speed. The Kalman filter state space used here is three-dimensional, $X = (x, v, a)$, where x is distance-into-block, v is speed, and a is acceleration. Details on the Kalman filter implementation, including derivation of filter parameters $\mathbf{R}$ and q, are given by Cathey and Dailey [10].

AVL reports for a given vehicle are received at an average rate of one per minute, and track data for the vehicle are periodically propagated and output at a nominal rate of once every 20 seconds. Because of the low sample rate, the starting and stopping of a bus at bus stops is not generally observable. This "fine grain" behavior is compensated for in the noise terms of the Kalman filter [11]. For buses that stop frequently, filtered speeds will be lower than they actually are.

For a more detailed discussion of the Tracker Component and the Kalman filter, see Cathey and Dailey [10].

3.4 Corridor Estimator

In this section, we describe the procedure in use by the Corridor Estimator component to determine the corridor locations, times, and speeds of transit vehicles using reported track data. As shown in Figure 18, this component is supplied with static information from three sources: (1) a transit schedule database, (2) a “Covering Arcs Builder” component, and (3) a “Corridor Builder” component.

3.4.1 Covering Arcs Builder

The Covering Arcs Builder component provides an index to map road-segments into the spatial schedule information. The Covering Arcs Builder uses an industry standard “geographical information system” (GIS) road-segment database that underlies the construction of the bus schedule time-points and time-point-intervals. It is the purpose of the Covering Arcs Builder is to decompose each TPI into the corresponding chain of oriented arcs. These chains are referred to as “TPI covering arcs.” Figure 6 illustrates the relationship between TPIs and covering arcs.

3.4.2 Corridor Builder

In the context of this report, a “corridor” is a geographic path consisting of road-segments connecting some specified starting point to a specified ending point. Corridors are defined in much the same way as schedule patterns: a corridor is a sequence of “corridor intervals” in which each corridor interval is a chain of oriented GIS arcs. In order to make the construction of arbitrary corridors a feasible task, we developed an interactive, map-based graphical application. This program displays the GIS database in map form and allows the user to create, edit, and save corridor definition files.

To create a corridor, the user first selects a starting point by clicking the mouse near the image of a road. Corridor intervals are generated automatically between successive mouse clicks along the desired path. This is accomplished with the use of an A^* shortest path algorithm [12] so that the user need not manually specify every individual arc. A corridor path can consist of hundreds of arcs.

3.4.3 Corridor Estimator Component

The purpose of the Corridor Estimator component is to determine which corridors the transit vehicles are traveling along and to compute the locations of the vehicles with respect to these corridors. The algorithm that performs this task is based on a mapping between block locations and corridor locations, defined with the help of the static information described above. This mapping, which is constructed during the initialization phase of the algorithm, is defined as follows.

Each block *B* is decomposed into a chain of oriented roadway arcs. This is possible because the structure of the transit database allows a block to be reduced to a sequence of TPIs, and each TPI has an associated chain of arcs provided by the Covering Arcs Builder. The orientation of an arc with respect to a block is the same as its orientation with respect to the TPI from which it is obtained. To facilitate vehicle location, each member, *A*, of the chain of arcs on a block is augmented with its distance-into-block, $d_B(A)$. Thus, given vehicle distance-into-block, *x*, we can quickly determine the corresponding roadway arc, *A*, and vehicle distance along the arc, *y*, as follows: *A* is the last arc in the chain such that $d_B(A) < x$, and

$$y = \begin{cases} x - d_B(A) & \text{if } A \text{ is positively oriented with respect to } B \\ |A| - (x - d_B(A)) & \text{otherwise} \end{cases} \tag{18}$$

Here $|A|$ denotes the length of *A*.

For each block, the corresponding chain of arcs is partitioned into a list of sub-chains, where each sub-chain is either a sub-chain of a corridor or is disjoint from all corridors. Each sub-chain of a corridor is augmented with the distance-into-block for both its start node and end node. Thus, given vehicle distance-into-block, we can quickly determine whether the vehicle is traveling on a corridor.

Each corridor, *C*, is decomposed into a chain of arcs by joining together the corridor intervals that make up that corridor. As above with the block decomposition, each member, *A*, of the chain of arcs on a corridor is augmented with its distance-into-corridor, $d_C(A)$.

With this structure in place, we now specify the correspondence between block location and corridor location. Given a block location, specified by distance-into-block, *x*, we determine whether the location lies on a corridor sub-chain. If it does not, then there is no corresponding corridor location. Otherwise, *x* determines an arc, *A*, and a distance into arc, *y*, such that *A* lies on some corridor *C*. The corresponding distance-into-corridor, *w*, is then given by

$$y = \begin{cases} y + d_c(A) & \text{if } A \text{ is positively oriented with respect to } C \\ |A| - y + d_c(A) & \text{otherwise} \end{cases} \tag{19}$$

Data transmitted on the probe output stream include: (1) corridor-name, C_{id}; (2) time, *t*; (3) distance-into-corridor, *w*; (4) speed, *s*; (5) vehicle-identifier, V_{id}; (6) block-identifier, B_{id}; (7) route-identifier, Rt_{id}; and (8) trip-identifier, T_{id}.

The block, route, and trip identifiers are used for downstream collation and filtering of speed data. For example, vehicles on some routes and trips are known to be significantly slower than others at certain corridor locations and are not indicative of general traffic speed. This is often the case for vehicles entering and exiting the corridors.

3.5 Applications

To test the Corridor Estimator system, we defined a number of sample corridors along highways I-5 and I-90, and along the arterial SR 99. Corridor speed data were collected and analyzed for several weeks of July 2002.

Figure 19 is a time-space diagram of the progress of the probe vehicles on an I-5 southbound corridor, in south Seattle, running from the West Seattle Freeway to Orillia (south of Southcenter). The left plot is for Friday, July 19th, when significant congestion was observed in the corridor, and the right plot is for Monday, July 22, 2002, when there was little congestion. These are plots of the distance along the corridor in feet, for all the vehicles using the corridor, as a function of time of day, reported in minutes past midnight. The time span presented here is from about 3:00 PM to 8:00 PM. Probe vehicle speed at a specific corridor, location, and time is given by the slope of the curve at that point. In the right figure, the constant slope of the lines suggests nearly constant speeds; however, in the left figure, the decrease in slope near the end of the corridor indicates a decrease in speed.

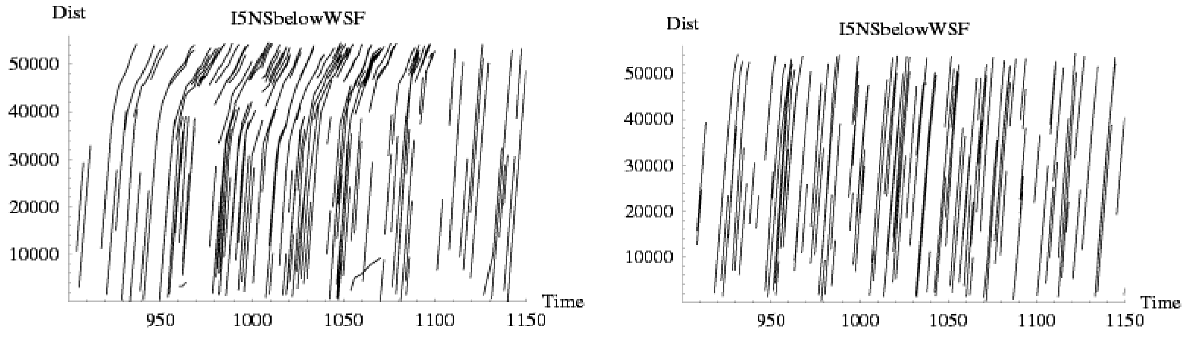

Figure 19: Corridor probe trajectories in space (feet into corridor) and time of day (minutes past midnight), with Friday on the left and Monday on the right.

We use probe vehicle data to construct an interpolating function, $v = f(x, t)$, that approximates speed (MPH), v, at any corridor location and time, (x, t). Figure 20 shows a plot of the surface corresponding to this function. Note the depression in the speed, centered at 5:00 PM, near the end of the corridor on Friday. To validate this surface, we compare the value on the speed surface at the location of an inductance loop speed trap in Figure 21. The gray points on the figure are the lane-by-lane, 20-second average speed; the heavy dots are the 10-minute average speeds from the probe vehicles; and the heavy line is the slice from the speed surface at the location of the speed-trap. In general, the probes travel on the slow side of the observed loop measurements but track the major changes in speed.

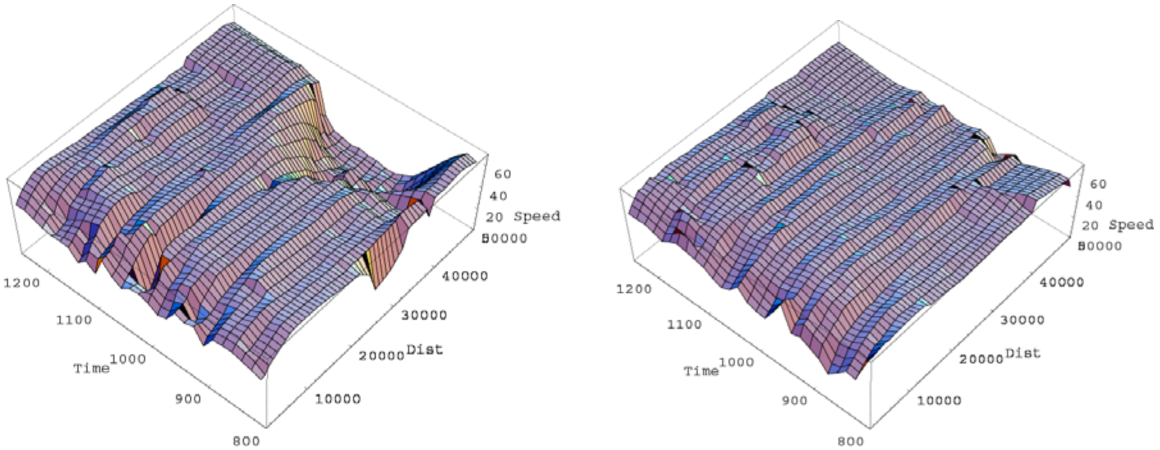

Figure 20: Corridor probe speed (miles per hour) surface as a function of time of day (minutes past midnight) and distance-into-corridor (feet), Friday on the left, Monday on the right.

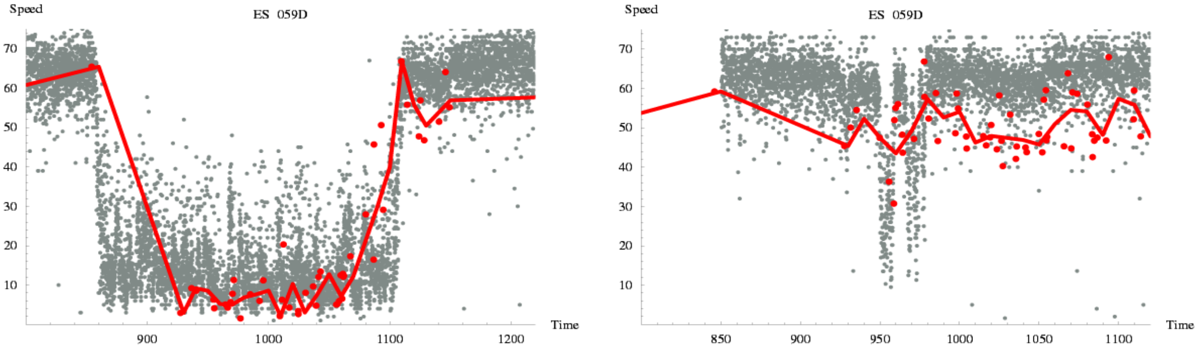

Figure 21. Comparison of probe and inductance loop speed-trap speeds (miles per hour) as a function of time of day (minutes past midnight), Friday on the left, Monday on the right.

A solution of the differential equation

$$\frac{dx}{dt} = f(x,t) \tag{20}$$

corresponds to the space-time trajectory of a vehicle traveling the corridor. To estimate travel time in the corridor, for a vehicle entering the corridor ($x = 0$) at time t_0, we do as follows. Solve Equation 20 with initial conditions, $t = t_0$, $x = 0$, to obtain a trajectory function, $x = \omega(t)$. Next, determine the value $t = t_1$, for which $\omega(t)$ is the corridor length. Then, $t_1 - t_0$ is the corridor travel time for a vehicle starting out at time t_0.

Figure 22 is a plot of corridor travel time, in minutes, as a function of time of day in minutes past midnight. The plot on the right is for an un-congested day, and the plot on the left is for a day with significant congestion. The corridor is approximately 10 miles long, and for the un-congested data, the vehicle travel times are approximately 12 minutes, suggesting that the travel-time estimate for the corridor will be on the conservative side.

In real-time applications, the corridor travel time is often estimated by using instantaneous speeds along the corridor so that the speed function, $f(x,t)$, is only a function of x, and t is set to "now." The travel time is then computed,

$$T = \int_0^L \frac{1}{f(x, t_{now})} dx , \qquad (21)$$

where L is the length of the corridor. This type of estimate, for the data set presented here, is shown in Figure 23. Comparing figures 22 and 23, it is clear that the instantaneous estimate lags, in time, the real increase in travel time, and furthermore, the maximum travel times are significantly larger for the instantaneous estimates in the congested case. This suggests that real-time travel-time estimates may benefit from the use of historical information that may mitigate these two effects.

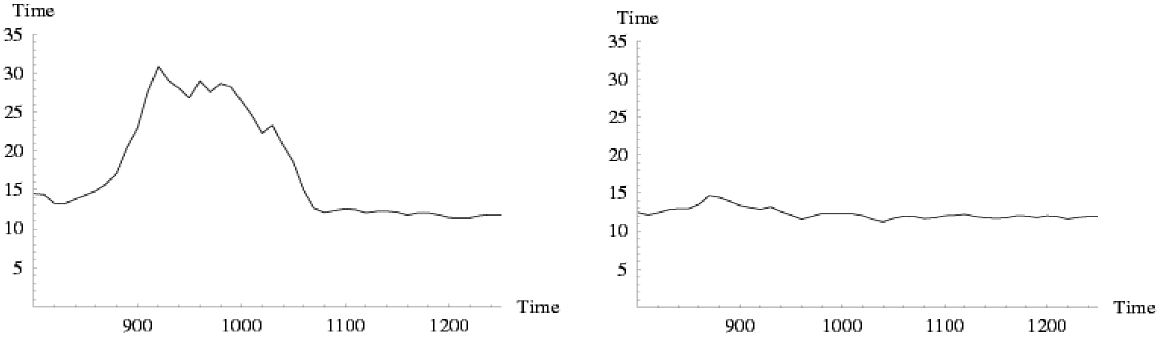

Figure 22. Corridor probe travel time (minutes) as a function of departure time of day (minutes past midnight), Friday on the left, Monday on the right.

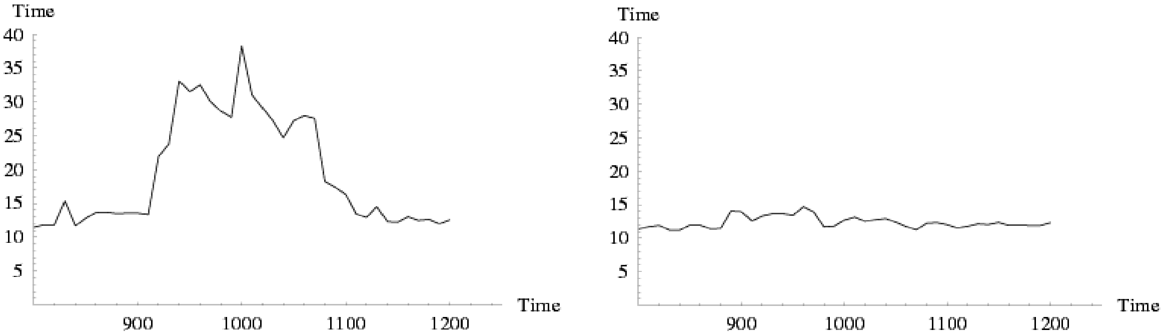

Figure 23. Corridor probe travel time (minutes) as a function of departure time of day (minutes past midnight) using instantaneous speed measurements, Friday on the left, Monday on the right.

3.6 Summary

We present a corridor approach to travel time estimates using transit vehicles as probes. These estimates produce more information density along the corridor than does use of only probe information at specified points. It provides speed estimates that track the significant changes identified in inductance loop data, but it seems to provide a conservative estimate of the speed. Comparison of instantaneous travel times, often used for real-time applications, and travel time computed with a corridor speed surface indicate that the instantaneous travel times have a delay in tracking changes in the corridor and a higher maximum travel time. This result suggests that additional research should be done to correct these biases. These corrections may take the form of incorporating historic information into the instantaneous travel estimates used by real-time applications. In addition in the future, probe vehicle data might also prove useful for preemption and signal timing activities.

REFERENCES

1. Dailey, D.J., and M.P. Haselkorn, K. Guiberson, and P. Lin. *Automatic Transit Location System*. Washington State Transportation Center - TRAC/WSDOT, Final Technical Report WA-RD 394.1, 49 pages, February 1996.
2. Bell, B.M. "The Marginal Likelihood for Parameters in a Discrete Gauss-Markov Process." *IEEE Transactions on Signal Processing*, Vol. 48, No. 3, pp. 870-873, March 2000.
3. Dailey, D.J., D. Meyers, and N. Friedland. "A Self Describing Data Transfer Methodology for ITS Applications." *Transportation Research Record 1660*, pp. 140-147, 1999.
4. Jazwinski, A.H. *Stochastic Processes and Filtering Theory*. New York, Academic Press, 1970.
5. Rauch, H.E., F. Tung, and C.T. Striebel. "Maximum Likelihood Estimates of Linear Dynamic Systems." *American Institute of Aeronautics and Astronautics Journal*, Vol. 3, pp. 1445-1450, August 1965.
6. Anderson, B.D.0. and J.B. Moore. *Optimal Filtering*. Englewood Cliffs, New Jersey: Prentice Hall, c1979.
7. Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge; New York: Cambridge University Press, 1992.
8. Efron, B. "Computer-Intensive Methods in Statistical Regression." *Siam Review*, Vol. 30, No. 3, pp. 421-449, September 1988.
9. Cathey, F.W. and D.J. Dailey. "A Prescription for Transit Arrival/Departure Prediction Using Automatic Vehicle Location Data." *Transportation Research C: Emerging Technologies, Traffic Detection and Estimates,* Vol. 11, Nos. 3-4 pp. 241-264, June-August 2003.
10. Cathey, F.W. and D.J. Dailey. "Transit Vehicles as Traffic Probe Sensors." *Transportation Research Record 1804*, pp. 23-30, 2002.
11. Dailey, D.J., S.D. Maclean, F.W. Cathey, and Z. Wall. "Transit Vehicle Arrival Prediction: An Algorithm and a Large Scale Implementation." *Transportation Research Record, Transportation Network Modeling, 1771*, pp. 46-51, 2001.

12. Bar, Avron and Edward A. Feigenbaum, edited by. *The Handbook of Artificial Intelligence*. Reading, Massachusetts: Addison-Wesley, 1986.