PERGAMON

Transportation Research Part C 11 (2003) 241–264

TRANSPORTATION RESEARCH PART C

www.elsevier.com/locate/trc

A prescription for transit arrival/departure prediction using automatic vehicle location data

F.W. Cathey, D.J. Dailey [*]

Department of Electrical Engineering, University of Washington, Seattle, WA 98155, USA

Abstract

In this paper we present a general prescription for the prediction of transit vehicle arrival/departure. The prescription identifies the set of activities that are necessary to preform the prediction task, and describes each activity in a component based framework. We identify the three components, a Tracker, a Filter, and a Predictor, necessary to use automatic vehicle location (AVL) data to position a vehicle in space and time and then predict the arrival/departure at a selected location. Data, starting as an AVL stream, flows through the three components, each component transforms the data, and the end result is a prediction of arrival/departure. The utility of this prescription is that it provides a framework that can be used to describe the steps in any prediction scheme. We describe a Kalman filter for the Filter component, and we present two examples of algorithms that are implemented in the Predictor component. We use these implementations with AVL data to create two examples of transit vehicle prediction systems for the cities of Seattle and Portland.
© 2003 Elsevier Ltd. All rights reserved.

Keywords: Bus; Transit; Prediction; Kalman filter; AVL; TCIP

1. Introduction

In this paper we present a general prescription for the prediction of transit vehicle arrival/departure. This prescription uses a data flow approach and breaks the task of predicting arrival/departure into several component parts. Each of these parts is realized as software that implements an algorithm to accomplish the goals of that component. The identified components are shown in Fig. 1, and the description of each of the components makes up the body of this paper.

The prescription presented is a general framework for the overall prediction task that encompasses a variety of techniques that have been used to date. For example, we show that the

[*] Corresponding author. Tel.: +1-206-543-2493; fax: +1-206-616-1787.
E-mail address: dan@its.washington.edu (D.J. Dailey).

0968-090X/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0968-090X(03)00023-8

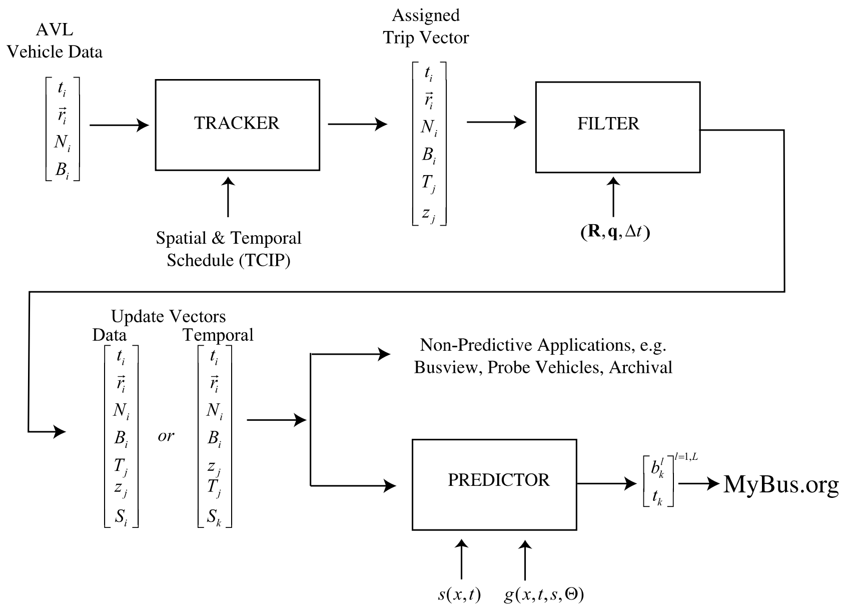
AVL
Vehicle Data
t_i $\vec{r}_i$ N_i B_i
TRACKER
Spatial & Temporal
Schedule (TCIP)
Assigned
Trip Vector
t_i $\vec{r}_i$ N_i B_i T_j z_j
FILTER
$(\mathbf{R},\mathbf{q},\Delta t)$
Update Vectors
Data
Temporal
t_i $\vec{r}_i$ N_i B_i T_j z_j S_i
or
t_i $\vec{r}_i$ N_i B_i z_j T_j S_k
Non-Predictive Applications, e.g.
Busview, Probe Vehicles, Archival
PREDICTOR
$\begin{bmatrix} b_k^l \\ t_k \end{bmatrix}^{l=1,L}$
MyBus.org
$s(x,t)$
$g(x,t,s,\Theta)$

Fig. 1. Overall design for a general prediction system.

"schedule deviation" approach for prediction is a simple version of the Predictor Component, while methodologies to incorporate a variety of external influences are examples of more complex Predictor Components that may incorporate information beyond the schedule such as traffic conditions, weather or historical information. We assert that the prescription presented is a unified framework for the prediction task into which a variety of implementations can be inserted.

The history of using computers to provide transit information goes back to the 1970s when Carlson and Brastow (1974) described placing "route numbers, arrival times, departure times, and transfer points … on a cathode ray tube" and to "perform certain trip planning calculations for urban transportation users." The use of computers and geographical information systems for schedule based planning is reported by Koncz and Greenfeld (1995) and more recently by Peng and Huang (2000). By the middle of the 1990s systems that used automatic vehicle location (AVL) data for traveler information began to appear in the literature. "Countdown," in London, was an early demonstration system that used AVL data to estimate and display bus arrival information at bus stops (Balogh and Smith, 1993). Also in the mid 1990s Dailey and Haselkorn (1994) reported a framework and implementation for a map based real-time transit information display. Ibrahim (1996) reports a GPS based system that provides location information to on-board passengers by flashing LEDs on a wall map type display. The deployment of two types of real-time transit display systems, map based bus locations and prediction based tabular displays, was reported in Dailey et al. (1999). A large scale prediction system, deployed as MyBus.org, that created HTML for web browsers, and Wireless Markup Language (WML) for cell phones, was reported in Maclean and Dailey (2001).

A common thread to the publication history above is that the focus was on describing the systems rather than the underlying science. Early work on algorithms was reported by Oda (1990), and we have reported several prediction algorithms of increasing complexity (Wall and Dailey, 1999; Dailey et al., 2000, 2001). It is the purpose of this paper to create a definitive prescription that places past work in a more general framework.

1.1. Prescription

We make three assumptions in the framework presented here: (1) There is a fleet of transit vehicles that travel along prescribed routes, (2) there is a "transit database" that defines the schedule times and the geographical layout of every route, and (3) there is an automatic vehicle location (AVL) system, where each vehicle in the fleet is equipped with a transmitter and periodically reports its progress back to a transit management center.

Fig. 1 is a data flow diagram for our prescription illustrating the three important components in this design. The prescription for prediction is illustrated by following the data as it flows through these three components. The definition and notation for the data elements in the data stream are defined in subsequent sections, as they are presented. The AVL data from the vehicles is combined with spatial and temporal schedule information in the first component, the "Tracker," to produce an assignment of a vehicle to the task scheduled by the transit operator. Next the data passes through the second component, a "Filter," that is an implementation of a Kalman filter process to make optimal estimates of the location and the speed of the vehicle. The "Filter" also generates a continuous stream of data that is updated at preselected temporal intervals. This data is then used in the third component, a "Predictor," that generates time of arrival/departure at a variety of downstream locations for each vehicle observed. This paper details the data elements and the functionality of each of these components.

2. Tracker

The first component of the prescription is the "Tracker." It assigns each observation of vehicle location, in both space and time, to the correct piece of scheduled work. Both spatial and temporal schedule data from a transit property are necessary for this task. It has been our observation that the schedule data is represented differently in each transit property. The prescription presented is a general framework in part because of the reliance on a generic set of definitions for the data that makes up the temporal and spatial schedule information. This set of generic definitions is taken from the Transit Communications Interface Profiles (TCIP) Framework (AASHTO/ITE/NEMA, 1999). We first describe the TCIP data necessary to accomplish the task of the "Tracker." We then present the AVL data framework being used, and finally we present the tracking and trip assignment algorithm.

2.1. Transit data

To clarify the terminology used here, we present a conceptual description of relevant elements of the database in TCIP terms. There are five relevant terms, (1) time-point, (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 Cartesian state-plane coordinates or geodetic latitude and longitude. For the purposes of this paper, we assume NAD-83 state-plane coordinates and also assume that the transformation from geodetic to state-plane coordinates, and its inverse, are known (Stern, 1989). We choose to use Cartesian coordinates for computational reasons as the geometry and metric are Euclidean rather that ellipsoidal. The time-points are the locations at which the transit vehicles are scheduled to arrive or depart.

A *time-point-interval* (TPI) is a polygonal path representing a stretch of road directed from one time-point to another. The TPI is geographically defined by a list of "shape points," and the length of a TPI is the sum of the lengths of the segments joining successive member shape points.

A *pattern* is a geographical route made up of a sequence of TPI's, where the ending time-point of the ith TPI is the starting time-point of the $(i+1)$th. Note that the sequence of TPI's on a pattern determines a sequence of time-points on the pattern. (The converse is not necessarily true since there may be more that one TPI running from one time-point to another.) The distance-into-pattern of a TPI is defined to be the sum of the lengths of the preceding TPIs and the length of a pattern is the sum of the lengths of its TPIs. For a vehicle traversing a pattern, the index of the TPI that the vehicle is on and the vehicle's distance-into-TPI uniquely determine the distance-into-pattern of the vehicle. Fig. 2 shows a sample pattern plotted in state-plane coordinates (units are feet), with origin translated to the corner of Salmon and 5th in Portland Oregon.

A *trip*, labeled T in Fig. 1, is an assignment of schedule times to time-points on a pattern. More precisely, a trip specifies a start time and end time for every TPI on the pattern in such a way that the end time for the ith TPI is no greater than the start time for the $(i+1)$th. Note that it is possible for the same time-point to be assigned two successive times, in which case we say that a *layover* is scheduled at that point. Note also that a trip specifies a travel time for each of its TPIs and different trips traversing the same TPI may specify different travel times depending on the time of day. The set of all trips is partitioned into blocks as defined below.

A *block*, labeled B in Fig. 1, is a sequence of trips such that the schedule time for the end of the ith trip is no greater than the schedule time for the start of the $(i+1)$th trip. If the end time-point

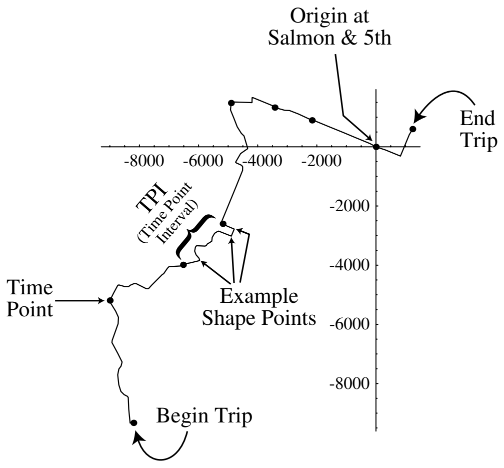

Fig. 2. A pattern with TPI's and shape points identified.

of the ith trip is the same as the start time-point for the $(i+1)$th, but the schedule times are different, then as in the preceding paragraph, we say that a layover is scheduled at that point. Each transit vehicle is assigned a block to follow over the course of the day. Some blocks are of long duration and are covered by different vehicles at different times of day.

2.2. Transit AVL

AVL systems produce real-time reports of vehicle location based on technologies such as dead reckoning or satellite GPS position measurements. In the case of the Portland Tri-Met AVL system each vehicle in the fleet is equipped with a GPS receiver and a radio transmitter. The vehicles periodically transmit location reports back to the transit center. These reports include the following information:

- vehicle-identifier: N in Fig. 1,
- block-identifier: B in Fig. 1,
- time: t in Fig. 1,
- latitude and longitude: $\vec{r}$ in Fig. 1.

Applications that make use of AVL data require the information in a more usable form. For example, a graphical application that displays vehicle location or measures distance between the vehicle and landmarks must transform the reported geodetic coordinates into Cartesian planar coordinates with a minimum of distortion. We use the NAD-83 state-plane transformations to do this, and assume henceforth that each AVL report is automatically augmented with position coordinates in the same state-plane coordinate system used in the schedule database.

Fig. 3 shows a state-plane plot of a sequence of vehicle locations superimposed over a schedule pattern. In this figure the coordinates have been translated so that the origin is at the corner of Salmon and 5th. The vehicle is moving towards the end of the pattern in the lower left quadrant. Fig. 4 shows a corresponding plot of time (in minutes past midnight) versus distance (in feet) from reported position to nearest point on pattern.

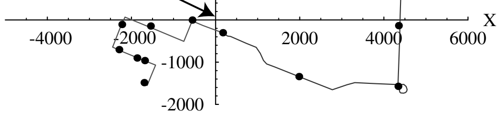

Fig. 3. Vehicle locations in state-plane coordinates.

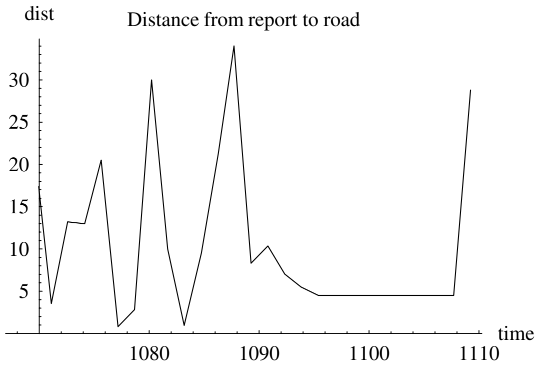

Fig. 4. Distance from reported position to nearest point on pattern.

An application which computes schedule deviation or predicts vehicle arrival times requires information in a form that is directly correlated to the scheduled block of work. These data are the trip identifier and the distance-into-trip, where distance-into-trip is defined to be the distance along the underlying pattern. For example, if the vehicle distance-into-trip equals that of a scheduled time-point, then schedule deviation is just the difference of report time and schedule time.

Given trip and distance-into-trip, we can determine block and state-plane location coordinates. The preceding definition of trip requires that every trip is associated with a unique block, and the pair (trip, distance-into-trip) determines the pair (pattern, distance-into-pattern) exactly. The pair (pattern, distance-into-pattern) uniquely determines both TPI, and distance-into-TPI. By moving the appropriate distance along the sequence of shape points on the identified TPI, we determine a pair of successive shape points on either side of the vehicle. Finally we determine vehicle state-plane coordinates by interpolating between these two shape points.

In the next section we describe a procedure to determine trip and distance-into-trip given the block, time, and vehicle location in state-plane coordinates.

2.3. Tracking and trip assignment

Mapping AVL data to an assigned piece of scheduled work is not straightforward. Vehicles may traverse the same pattern several times, perhaps in different directions, in addition they may be delayed or ahead of schedule. As a result, there are ambiguities when assigning AVL reports to trips. For example, if a vehicle is operating on a block that travels back and forth over the same roadways between two destinations, and the vehicle is late on a trip traveling one way, its position can be such that it could be identified as late or it might be identified as early on the next trip traveling the other way. This type of ambiguity is the reason there is a need for a sophisticated tracker.

We present a tracking methodology that determines trip and distance-into-trip for transit vehicles that provide AVL reports. The basic idea is to maintain a track on each vehicle which records the last report, trip assignment, distance-into-trip and other useful information. Tracks are then updated as new reports are received. The trip assignment logic and distance-into-trip calculation take advantage of previous track values in order to improve efficiency, eliminate ambiguities, and increase probability of update correctness.

The goal of the Tracker is to use an AVL report to produce a *track* that contains the following information:

- vehicle-identifier N,
- last associated AVL report $[t, \vec{r}, N, B]$,
- trip T,
- distance-into-trip z,
- TPI,
- distance-into-TPI d,
- deviation δt,
- validity (true or false),
- number of rejected updates,

where the validity field indicates whether or not the trip data can be used in subsequent processing, (TPI, distance-into-TPI) can be computed from (trip, distance-into-trip), and deviation δt is defined to be the difference between the report time and the *interpolated schedule time* for the distance-into-TPI of this report. The interpolated schedule time $\tilde{t}$ needed for this is the starting time, t_0, for the TPI plus the time to traverse the distance-into-TPI, d_i, portion of the TPI at a known speed (s). The schedule deviation,

$$\delta t = t - \tilde{t} = t - (t_0 + d/s), \tag{1}$$

provides a temporal sanity check for AVL reports.

The validity field of each track is initially marked invalid and remains so until a report is received from which we can compute reasonable trip data. A valid track will be marked invalid if the time since the last report exceeds a specified duration or the number of rejected updates exceeds a specified limit. An update is rejected if no trip data can be computed that is reasonable with respect to the track data. This may be due to measurement error in the report or to previously accrued error in the track.

The essence of the tracking algorithm is based on the classic paradigm of "generate and test" (Charniak and McDermott, 1984). The rationale behind this approach is that typically there are several feasible trip assignments for a report and we must test for the most likely one. For example, in the middle of a TPI a single candidate will normally be determined; however, at the end of a TPI there will be two or more, and near the end of a trip there may be four or more candidates to choose from. The generate and test logic is the following:

1. Look up the track with matching vehicle-identification, and if none are found we create a new track and mark it invalid.
2. Generate a set of candidate trip data tuples (trip, distance-into-trip, TPI, distance-into-TPI, deviation).
3. Apply selection/test rules to determine the most likely choice.
4. Update the track accordingly.

The generation of candidate trips and trip selection rules are detailed below.

2.3.1. Generation of candidate trips

Generating candidate trips involves reasoning in both space and time. When an AVL report is received, the candidate trips are established using geographical and temporal proximity. The generate and test paradigm requires a set of hypotheses from which the "correct" one is selected. In our case, identifying candidate trips is equivalent to generating hypotheses. The algorithm below identifies the candidates and the algorithm in the next section acts to select the correct one from the candidates.

The algorithm to identify the candidate trips is as follows:

1. Obtain an AVL report.
2. Define a circular neighborhood ($N(\vec{r}, \sigma)$) around the reported position $\vec{r}$ where the radius σ bounds the maximum expected measurement error.
3. Determine the tile Γ containing the point $\vec{r}$.

   We use an original "Tiling" technique to provide keys for fast searching of TPIs. In building an implementation of our algorithm it is necessary to create an efficient mechanism for accessing subsets of the large map data set. As a result, a new "tiling" methodology is used for creating a tree to access the TPIs. The details of this new technique can be found in Appendix A.
4. Determine the "time-feasible" pairs, (trip, TPI), associated with the tile.

   Using the reported block and time, select a set of pairs (trip, TPI), where the trip is on the reported block, TPI lies on the trip's underlying pattern, TPI intersects Γ^*, the tile Γ inflated by radius σ, and the pair is "time-feasible." "Time-feasible" means that the time of the report lies in a specified time-window containing the schedule times for the starting and ending time-points for the TPI. For example, in our results presented later, we specified a window whose lower limit was 20 min earlier than the TPI start time and whose upper limit was 90 min later than the TPI end time. A simple search through the block is used to determine the set of these "time-feasible pairs." Moreover, if the track associated with this report is valid, then the search begins with the last (trip, TPI) pair, or, if the last distance-into-TPI is less than σ, with the preceding pair.
5. Determine the set of "space-feasible" triples from the "time-feasible" set.

   Using the reported location ($\vec{r}$) and the pairs just determined we perform geometric processing to compute a set of "space-feasible" triples (trip, TPI, p) where p is a point on the TPI. A triple is "space-feasible," if p locally minimizes distance from the TPI to the reported position p and this distance is less than σ.

   The distance used in this optimization is the distance from a point to a line segment and is described in detail in Appendix B.

   To perform this minimization we proceed as follows. Let the TPI polygonal path be defined by the sequence of shape points $\{p_k | 0 \leqslant k \leqslant N\}$ and represent an arbitrary point p on the path parametrically as a pair (k, d) where k is a shape point index and d is distance along the line segment $\overline{p_k p_{k+1}}$.

   Let $\{p_k | L \leqslant k \leqslant M\}$ denote the subsequence of shape points representing an intersection sub-path of TPI with tile Γ^*. Compute the distance r_k from $\vec{r}$ to each successive line segment $\overline{p_k p_{k+1}}$ on this sub-path, and for each k such that $r_k < \sigma$ is a local minimum compute the distance d along the segment that specifies the minimizing point p. The distance-into-TPI for point p is simply the sum $d + d_k$ where d_k is the distance-into-TPI of the kth shape point.

We assert that the AVL report belongs to one of the remaining trips that are "space-" and "time-feasible."

2.3.2. Selection of trip

In the last section we identified hypotheses or candidate trips that need to be evaluated to determine the best estimate of the trip from which the AVL report arises. In this section we describe the rules and tests that we apply when selecting a candidate trip data tuple to update the *track*. The rules are a set of conditionals.

If the candidate set is empty,
  Then no selection is possible.
    In this case the vehicle track associated with this report will either be marked invalid or its update rejection count will be incremented, as discussed previously.
If the current report is associated with an invalid track,
  Then select the candidate tuple whose deviation has minimum absolute value.
If the associated track is valid,
  Then we subject the set of candidates to some screening tests and eliminate any candidate that fails a test.
    For each candidate trip data tuple, let Δx denote the distance traveled by the vehicle since the last report, assuming that both track and the tuple represents the "truth" i.e., the error in the candidate position measurement is within the specified tolerance σ. Let Δt denote the corresponding change in time. We assume that any error in reported time is insignificant in comparison to position error.

1. The first screening test corresponds to the reasonable assumption that the bus moves forward along any pattern: an acceptable tuple must satisfy $\Delta x > -\sigma$.
2. The second test checks for tuples such that Δx is reasonable given the time step Δt. We predict the maximum distance $\widehat{\Delta x}$ that the vehicle could have traveled in time Δt based on speed limit and schedule data and require that $\widehat{\Delta x} > \Delta x - 2\sigma$.

If the remaining set is empty,
  Then no selection is possible.
If only one candidate survives,
  Then it is selected,
If more than one candidate tuple survives these tests,
  Then we apply preference rules.
    If there are two candidates which together indicate that the vehicle is on a layover between trips (that is, one candidate indicates that vehicle is at the end of the track's last trip and another candidate indicates that it is at the start of the next trip on the block),
    Then in this case, we select the candidate whose deviation has minimum absolute value. (We effectively declare a trip transition halfway through the layover.)
    If the layover rule is not applicable,
    Then we predict an expected distance $\widehat{\Delta x}$ that the vehicle would have traveled in time Δt based on speed limit and schedule data and select the candidate whose value for Δx is closest to $\widehat{\Delta x}$.

Fig. 5 shows a time series plot of computed distance-into-trip for a specific vehicle on several trips. (Here, time is measured in minutes after midnight and distance in feet.) Each continuous curve rising from left to right corresponds to a trip and consists of a polygonal line joining scheduled time-points. Each curve is labeled with two numbers. The upper number identifies the trip, while the lower number identifies the trip's pattern. (Note that the trip's pattern number repeats when the vehicle travels over the same physical streets repeatedly.) The short horizontal line segments are drawn to help visualize estimated schedule deviations of the vehicle. Fig. 6 shows a similar plot for every trip on a block.

Fig. 7 shows an example time series plot of the distance from reported position to nearest point on trip over an entire block.

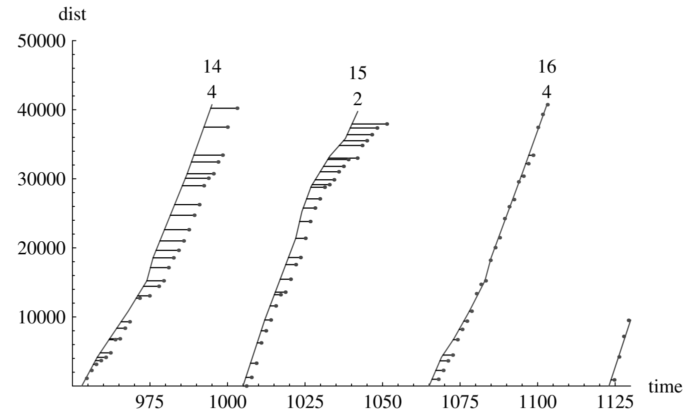

Fig. 5. Time series of computed distance into trip.

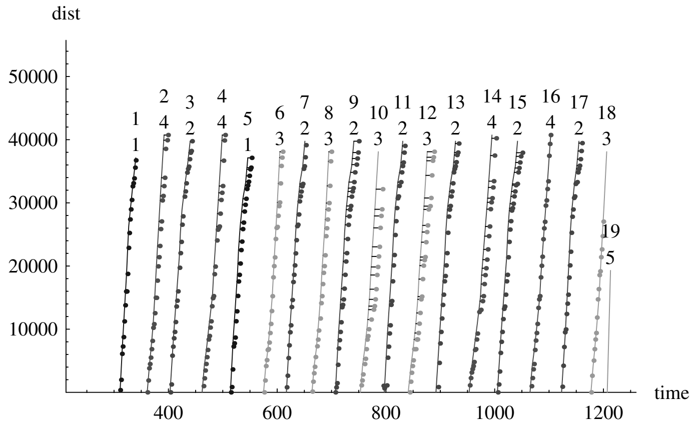

Fig. 6. Time series of computed distance for a block.

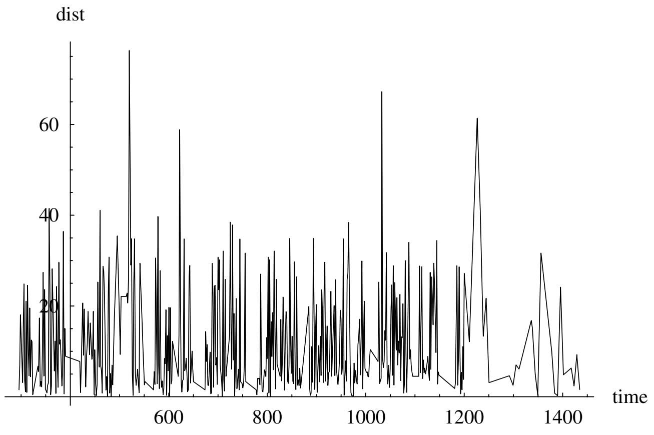

Fig. 7. Time series of distance from report to nearest point on block.

3. Filter

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 using the method of maximum marginal likelihood as specified in Bell (2000), and described in Section 3.2 below.

3.1. Design

Recall that in order 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 are specified, one may employ any one of a number of implementations of the Kalman filter/smoother equations, (see Bell, 2000; Jazwinski, 1970; Rauch et al., 1965 or Anderson and Moore, 1979) 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

$$X = [x \quad v \quad a]^{\mathrm{T}},$$

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 Tracker is an estimate of the vehicle's distance into trip, and our measurement model is given by

$$z = HX + e = x + e.$$

Here $H = (1\ 0\ 0)$ 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 nominal value of $R = (500\ \text{ft})^2$. The value for R was determined as described in Section 3.2.

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

$$\begin{aligned} \mathrm{d}x &= v\,\mathrm{d}t, \\ \mathrm{d}v &= a\,\mathrm{d}t, \\ \mathrm{d}a &= \mathrm{d}w. \end{aligned} \tag{2}$$

Here $\mathrm{d}t$ is the differential of time and $\mathrm{d}w$ is the differential of Brownian motion representing randomness in vehicle acceleration. By definition of Brownian motion (see Chapter 3, Section 5 of Jazwinski, 1970), the expectation

$$E(\mathrm{d}w^2) = q^2\,\mathrm{d}t,$$

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 $(264\ \text{ft/min}^2)^2/\text{min} = (3\ \text{mph/min})^2/\text{min}$ using the method described in Section 3.2.

The differential equations (2), are written in vector form as follows

$$\mathrm{d}X = FX\,\mathrm{d}t + G\,\mathrm{d}w,$$

where

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

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

$$X(t + \delta t) = \Phi(\delta t)X(t) + W(\delta t),$$

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

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

is the "transition matrix," and where the accumulated error $W(\delta t)$ has covariance

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

See the text following Theorem 7.1 of Jazwinski (1970) for a discussion of integration.

Finally, in order 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, $\widehat{X}_0$ and P_0, are based on the initial measurement z_0 (at time t_0) and measurement variance R. We set $\widehat{X}_0 = (z_0\ 0\ 0)$ and set

$$P_0 = \begin{pmatrix} R & 0 & 0 \\ 0 & (30\ \text{mph})^2 & 0 \\ 0 & 0 & (16\ \text{mph}/\min)^2 \end{pmatrix}.$$

The initialization of the covariance above is specified by a number of "hard-coded" parameters. One could attempt to determine their "optimal" values experimentally using the techniques of Section 3.2 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

$$\widehat{X}_k^- = \Phi_k \widehat{X}_{k-1},$$
$$P_k^- = \Phi_k P_{k-1} \Phi_k^{\mathrm{T}} + Q_k,$$

where $\widehat{X}_k^-$ is the prediction of the state vector at the kth step, $\Phi_k = \Phi(t_k - t_{k-1})$ is the transition matrix between the $k-1$ to the kth step and Q_k is the corresponding transition covariance matrix. The data update equations are

$$\widehat{X}_k = \widehat{X}_k^- + K_k(z_k - H\widehat{X}_k^-),$$
$$P_k = (I - K_k H)P_k^-,$$

where

$$K_k = P_k^- H^{\mathrm{T}}(HP_k^- H^{\mathrm{T}} + R)^{-1}.$$

This is the set of equations for use with real-time applications.

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

$$\overline{X}_{k-1} = \widehat{X}_{k-1} + P_{k-1}\Phi_k^{\mathrm{T}}(P_k^-)^{-1}(\overline{X}_k - \widehat{X}_k^-),$$

where $\overline{X}_N = \widehat{X}_N$. See Rauch et al. (1965) for a derivation of these Kalman filter/smoother formulas. This smoother is used for parameter estimation and post processing.

Fig. 8 is a plot of the time series of observed and smoothed distance-into-trip and the versus time. In Fig. 9 the data points are an estimate of speed based on simple differencing and the smooth curve is the result of the Kalman filter. Fig. 10 is a plot of residuals, the measurement minus the estimate where the horizontal lines at plus and minus 500 ft correspond to the measurement variance parameter used by the filter.

3.2. 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. 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

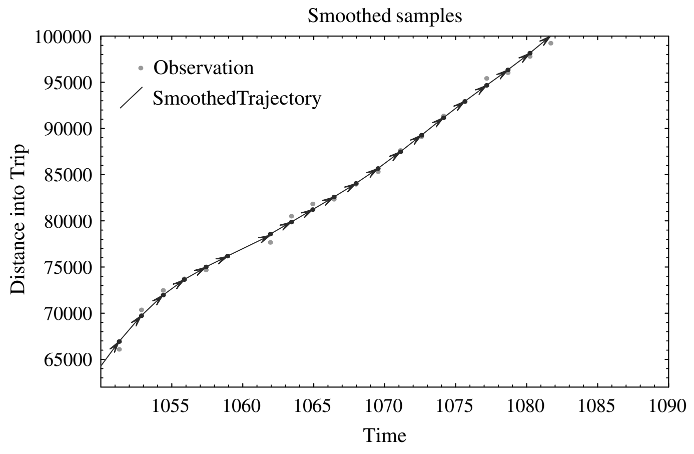

Fig. 8. Time series of distance into trip.

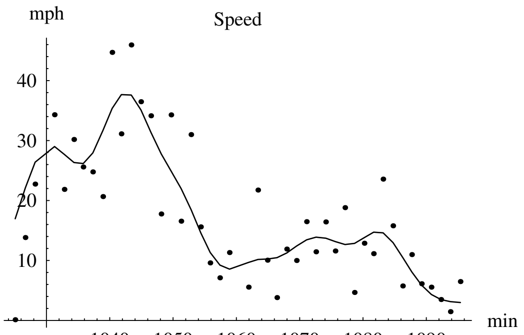

Fig. 9. Time series of estimated speed.

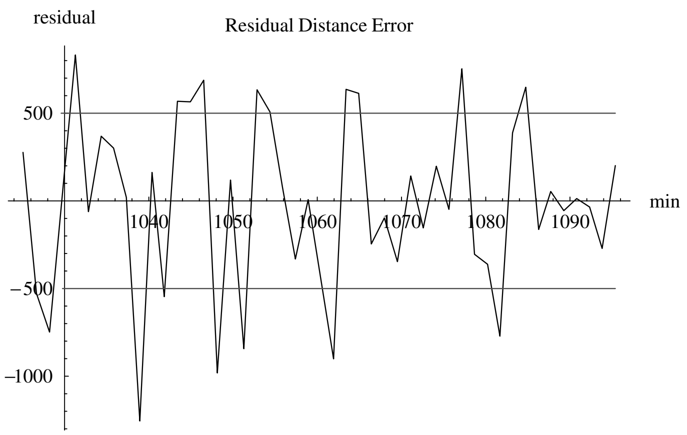

Fig. 10. Residuals between measurements and estimates.

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 in Bell (2000) where an effective procedure is provided for evaluating the marginal likelihood function. We briefly describe the theory.

Let $\mathbf{Z}$ and $\mathbf{X}$ 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 Bell (2000) (Eq. (4)) for the joint probability density function $p_{\mathbf{Z},\mathbf{X}}(z, x; \xi)$ 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)

$$x^*_{z,\xi} = \underset{x}{\operatorname{argmax}}\{p_{\mathbf{X}|\mathbf{Z}}(x|z, \xi)\}$$

and evaluates the marginal density function

$$p_{\mathbf{Z}}(z, \xi) = \int p_{\mathbf{Z},\mathbf{X}}(z, x, \xi)\, \mathrm{d}x.$$

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 $f_z(\xi) = -\log(p_{\mathbf{Z}}(z, \xi))$, that depends on the measurement sequence z, and then used Powell's conjugate direction minimization algorithm (Chapter 10, Section 5 of Anderson and Moore, 1979) to find the optimal parameter values for each measurement sequence z

$$\xi^*_z = \underset{\xi}{\operatorname{argmin}}\{\mathrm{f}_z(\xi)\}.$$

The results presented use a set of parameters derived as just described.

4. Predictor

The third component in our prescription is the Predictor. This component provides accurate predictions of transit vehicle behavior. As indicated in Fig. 1 the Predictor is driven by reports from the Filter (either data or temporal updates), and produces arrival/departure predictions $\{b^l | 0 \leqslant l \leqslant L\}$ at a sequence of L scheduled time-points ahead of the vehicle.

The most general form of the predictor provides a method for predicting arrival/departure times at any point for a vehicle traveling on a known path. To determine the travel time and the travel path $x(t)$ given a scheduled piecewise continuous speed function, $s(x, t)$ where x is position along the path and t is time, we select a starting location and time of (x_0, t_0) and an ending location x_1 and solve

$$\frac{\mathrm{d}x}{\mathrm{d}t} = s(x,t) \quad \text{for } t_1 = t(x_1) \text{ given } (x_0, t_0), \tag{3}$$

This form accounts for scheduled travel.

However, the vehicles are affected by a variety of outside influences. We acknowledge these effects by substituting a general functional form $g(x,t,s,\Theta)$ for $s(x,t)$ in Eq. (3). This form allows for dependence on the scheduled speed function s as well as such things as the statistics of observed vehicle performance, the driver or dispatcher performance, the traffic conditions, the changes in the measurement error and abnormal conditions like weather and natural disasters that depend on position and time (x,t). The travel time is now estimated by solving

$$\frac{\mathrm{d}x}{\mathrm{d}t} = g(x,t,s,\Theta) \quad \text{for } t(x_1) \text{ given } (x_0, t_0), \tag{4}$$

where the vector Θ represents the parameters necessary for $g(\cdot)$. The vector Θ might be coefficients of models for delay, due to things like: traffic, historical observations, time of day, weather or bridge openings. For example,

$$g(x,t,s,\Theta) = \begin{cases} s(x,t) & \text{if } \Theta = 0, \text{ when not snowing,} \\ f(x,t) & \text{if } \Theta = 1, \text{ when snowing,} \end{cases}$$

where, $f(x,t)$ is the historic snow behavior for the buses.

Accurate estimation of travel times is necessary for accurate estimation of arrival times at future destinations. Additional considerations are involved when estimating trip departure times or departures after a layover. In order to clarify the notions of arrival and departure we make the following definitions. Let $x(t)$ be a solution to Eq. (4) and let $x_1 = x(t_1)$. For any $\bar{x}$ such that $x_0 \leqslant \bar{x} \leqslant x_1$ select tolerances δx_a and δx_d and define the *predicted arrival time* at $\bar{x}$ to be

$$t_\mathrm{a} = \min(t | x(t) \geqslant \bar{x} - \delta x_\mathrm{a})$$

and *predicted departure time* at $\bar{x}$ to be

$$t_\mathrm{d} = \min(t | x(t) \geqslant \bar{x} + \delta x_\mathrm{d}).$$

In practice we set $\delta x_\mathrm{a} = \delta x_\mathrm{d} = 0$ when $\bar{x}$ corresponds to a "passing" time-point, i.e., a point between layovers. At a layover, however, we require $\delta x_\mathrm{d} > 0$ to ensure that a correct estimate of departure time is computed.

5. Examples

To demonstrate the use of the general prescription we present three examples. The first example uses schedule data from Portland OR to define $s(x,t)$ in the Predictor Component, and demonstrates that the "schedule deviation" approach is a special case of the generalization presented here. The second example uses probe vehicle data from Seattle WA roadways to construct $g(x,t,s,\Theta)$ for the Predictor Component, AVL data for the input data flow and creates predictions by solving Eq. (4). The third example uses data from both Portland and Seattle. AVL and schedule data are used to demonstrate that the predictions provided by the methodology suggested here are statistically far superior to the schedule information.

5.1. Example 1: Portland, OR

In this example we demonstrate that the "schedule deviation" approach to prediction is a simplified form of the general approach presented here. To accomplish this we define a speed function $s_b(x,t)$ based only on schedule data for block b such that the procedure described above computes the same predictions as the schedule deviation procedure.

The basic idea behind the schedule deviation approach is to predict arrival/departure time at time-points ahead of the vehicle by adding the schedule time for the time-point to the current interpolated schedule deviation of the vehicle as defined in Eq. (1). So, if the current deviation is δt then this is the predicted deviation for future time-points.

This description is an over-simplification as special logic is required to correct the predicted deviation when predicting time of departure from a layover time-point, or a start trip time-point, or points on beyond these. For example, the logic involved to correct deviation for a layover departure is the following: let t_a denote the scheduled time of arrival at the layover, let t_d denote the scheduled time of departure, and let δt denote the current value for schedule deviation. If $t_\mathrm{a} + \delta t \leqslant t_\mathrm{d}$ then zero the deviation, $\delta t := 0$, otherwise decrement the deviation $\delta t := t_\mathrm{a} + \delta t - t_\mathrm{d}$.

In order to define the speed function that corresponds to the schedule deviation approach to prediction, we introduce the notion of *distance-into-block*. Recall that a block can be represented as a series of trips making up an overall piece of work that a transit driver performs in a day. On observing the location and time of a vehicle update we can translate that to a location on a block of trips using a single parameter, *distance-into-block* x defined as follows. Suppose the vehicle location lies on the kth trip of the block and its distance-into-trip is d. Then distance-into-block for this point is defined as the sum of the lengths of the preceding $k-1$ trips plus d. The trajectory for a vehicle in x–t space can be imagined by stacking successive trips in the x-direction. We define the *scheduled trajectory* of a block to be the piecewise linear curve in x–t space that joins consecutive scheduled time-points.

We first define a simple piecewise constant function $\hat{s}_b(x,t)$ which we will then modify around each layover to produce the desired function $s_b(x,t)$. Let $t_1 < t_2 < \cdots < t_N$ be the sequence of times and $x_1, x_2, \ldots, x_N$ be the corresponding sequence of x-coordinates for every scheduled time-point on the block. Then $x_k \leqslant x_{k+1}$ for every k and equality holds only at layover points. Note that for every distance-into-block x there is a unique k such that $x_k \leqslant x < x_{k+1}$. Define $\hat{s}_b(x,t)$ as follows:

$$\hat{s}_b(x,t) = \frac{(x_{k+1} - x_k)}{t_{k+1} - t_k} \quad \text{if } x_k \leqslant x < x_{k+1}. \tag{5}$$

Now modify this function in the vicinity of every layover $x_k = x_{k+1}$ as follows:

$$\begin{aligned} &s_b(x,t) = 0 \quad \text{if} \begin{cases} x_k \leqslant x \leqslant x_k + \delta x, \\ t \leqslant t_{k+1}, \end{cases} \\ &s_b(x,t) = \hat{s}_b(x,t) \quad \text{otherwise.} \end{aligned} \tag{6}$$

Solving Eq. (3) with $s(x,t) = s_b(x,t)$ to determine arrival/departure times is equivalent to using the schedule deviation approach.

Fig. 11 presents an example contour plot of $s_b(x,t)$ created as just described where darker is slower. Each of the bands of constant speed results from the schedule based speed approximation in Eq. (5). Further, a layover is represented by the black or zero speed region located at 40,000 ft

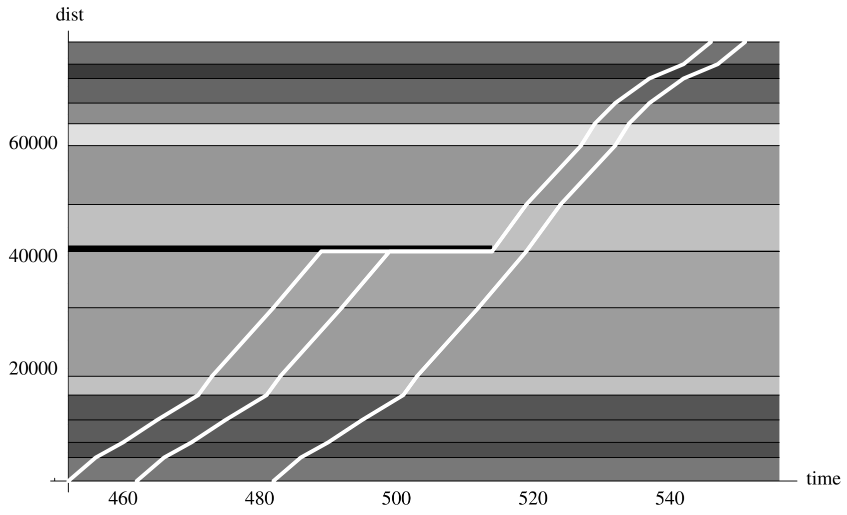

Fig. 11. Schedule based $s(x,t)$ and solution trajectories.

The trajectories of the solutions of Eq. (3) are shown as white bands. The trajectories represent the predicted location of the vehicle as a function of space and time. Each of the three trajectories represents a vehicle with a different starting time. The earliest or left-most travels through space and time until it reaches the distance-into-trip where a layover is scheduled, it then sits still until the layover has expired and continues on its way. The middle trajectory is a vehicle that starts later than the first one and also enters the layover. This vehicle then leaves the layover at the scheduled time such that both the first and second solutions are the same after the layover. The latest or rightmost trajectory is a vehicle that does not arrive at the layover location with sufficient time to stop. If the leftmost trajectory represents the planned behavior for the trip the rightmost represents a vehicle that is late but becomes less so when it bypasses a layover. Note that prior to arrival at the layover, the predicted time of arrival at any intermediate time-point is simply the schedule time plus the initial deviation.

5.2. Example 2: Seattle, WA

To demonstrate the importance of accurate knowledge of the speed function to travel time estimation we present an example of a speed functional $g(\cdot)$ constructed using probe vehicle data that provides speed estimates on roadways on which the transit vehicles travel. The probe vehicle data is created using the techniques from Cathey and Dailey (2001). In this case the schedule speed function is not used directly, although the probe vehicles attempt to adhere to the schedule. Θ is a two-dimensional table of smoothed speeds with indices corresponding to distance and time, and $g(\cdot)$ is a bilinear interpolation function applied to (x,t,Θ). In Fig. 12 is a contour plot of $g(x,t,\Theta)$, a function of time and space, where the darker internal regions are slower speeds.

To estimate travel times given this speed function, Eq. (4) is solved numerically using Euler's method. The two heavy white lines and the central heavy black line in Fig. 12 are the trajectories of three solutions, and the shape of the trajectory depends heavily on the shape of the speed

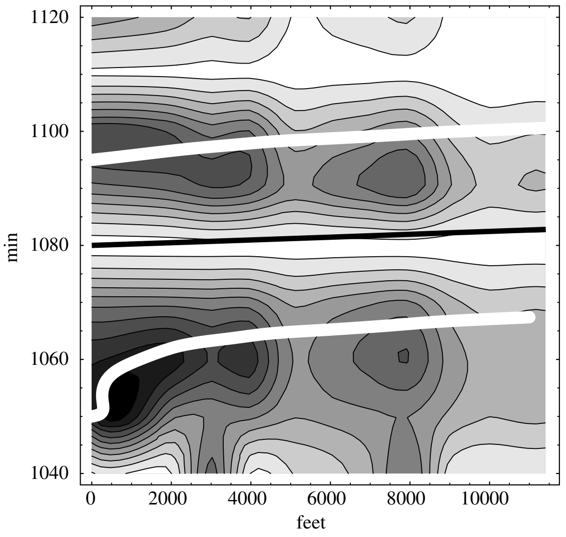

Fig. 12. Contour plot of speed, darker is slower.

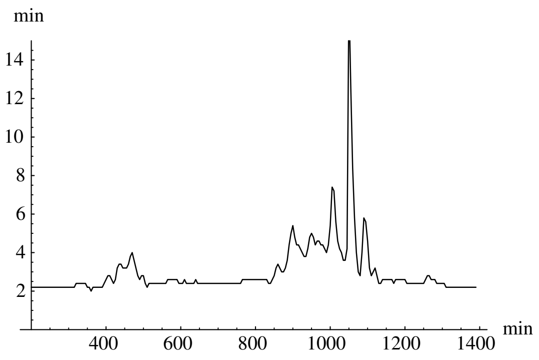

Fig. 13. Travel time as a function of departure time.

function. In particular, note the character of the bottom heavy white line that traverses a period and region that has 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 Eq. (3) for time t_1 subject to constraints $x(t_0) = 0$ ft and $x(t_1) = 11{,}000$ ft to obtain travel time $t_1 - t_0$. Fig. 13 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 1050 min, is associated with the bottom solution trajectory in Fig. 12.

5.3. Example 3: Seattle, WA and Portland, OR

In our third example we present the results of making optimal estimates of the arrival/departure time of transit vehicles. Predicting the future is always a challenging activity. And validating the effectiveness of our predictions is an important measure of the success of the methodology presented.

To compare the predictions with the vehicle behavior, we compare the prediction to the actual arrival time. To accomplish this we need to estimate actual arrival time. As we are tracking the vehicle irregularly, there is no guarantee that the location will be reported just as the vehicle arrives. To get an estimate of the real arrival, we record the location report just before arrival and just after arrival and linearly interpolate the actual arrival time (T_a). The implementation is continuously predicting the arrival as a function of both space and time. We create a deviation between the predicted and actual arrival as a function of time. We then histogram and normalize the resulting deviations to create a probability surface in space and time.

The predictions created using an instantiation of our prescription for Portland Oregon appear in Fig. 14. The right side of Fig. 14 shows the conditional probability, $p(t|T)$, of the error of the prediction t, conditioned on the time before arrival T when the prediction is made. In Fig. 14 the front axis $(-10 \leqslant t \leqslant 10)$ corresponds to the prediction error and the side axis $(0 \leqslant T \leqslant 30)$ corresponds to the time before arrival. A similar function for the relationship between the schedule and the actual behavior is shown on the left side of Fig. 14. The surfaces in Fig. 14 were created using the predictions made over the course of the entire day on September 12, 2000. Comparing these two surfaces suggests that prediction with dynamic information has 2–4 times smaller errors than using the schedule alone.

Fig. 15 is a similar probability plot for data from Seattle, WA. The surfaces in Fig. 15 were created using the predictions made over the course of one day for Seattle Metro Transit's second busiest time-point. This time-point, as well as being the second busiest, has buses passing that begin their trips in all parts of the city, and may be affected by a variety of external influences making it a challenging location for which to make predictions. In this case the errors in the predictions are 5–10 times smaller with our prescription and dynamic information.

Another measure of effectiveness is to ask the question how long in advance of the predicted departure must a passenger arrive to be guaranteed, with some confidence, of getting the bus. The time τ in advance of the predicted departure that allows for a 90% confidence of catching the bus using a prediction made T min in advance can be stated as: find τ such that

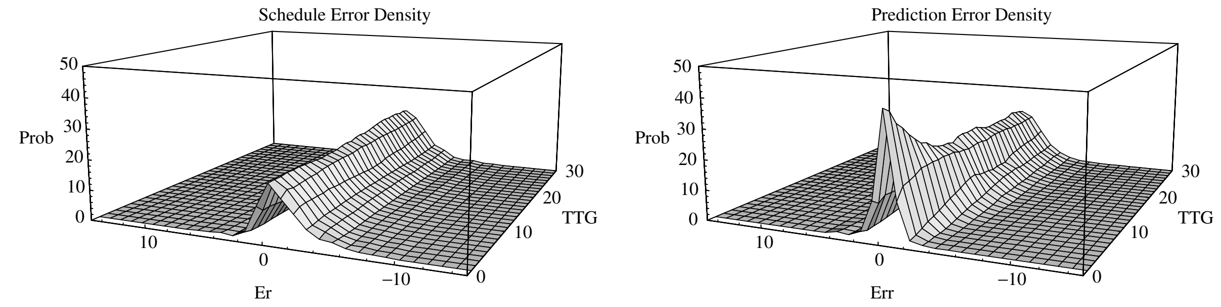

Fig. 14. Portland probability of correct predictions by the schedule on the left and the prescription presented on the right.

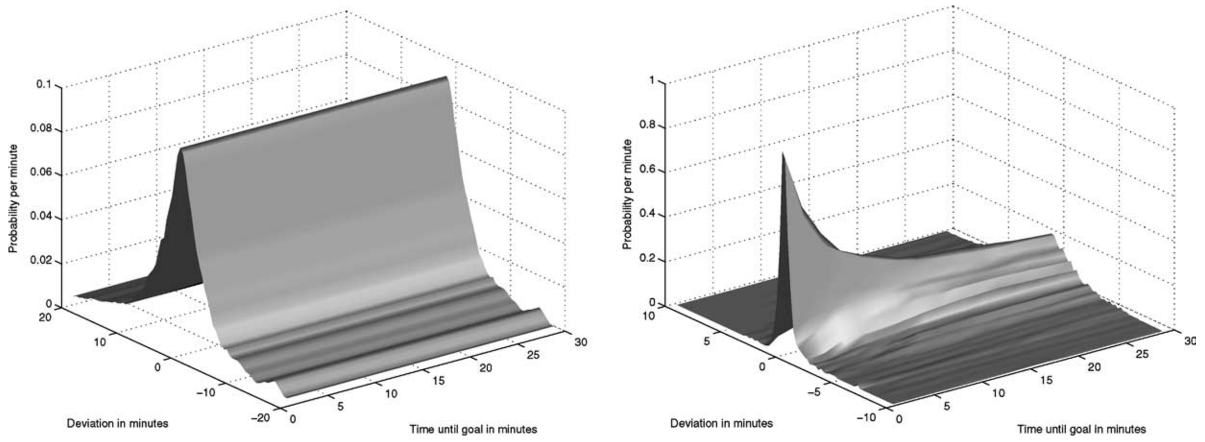

Fig. 15. Seattle: Probability of correct predictions by the schedule on the left and the prescription presented on the right.

Table 1
Time in advance of the predicted departure to have a 90% certainty of getting the bus

	Time before arrival T in minutes				
	1	5	10	20	Schedule
Portland τ in minutes	0.84	1.45	2.10	2.85	3.98
Seattle τ in minutes	0.51	1.35	1.80	2.45	11.03

$$0.9 = \int_{-\tau}^{\infty} p(t|T)\,\mathrm{d}t. \tag{7}$$

Table 1 has the results for both Seattle and Portland, across the top is the time in advance of the predicted arrival at which the prediction is made, including the schedule, and the entries below are the time in advance of the prediction the rider must arrive to have a 90% confidence of catching the bus.

This demonstrates that for transit riders the use of the predictive methodology presented here provides a significant gain in information, over using the schedule.

6. Conclusions

In this paper we present a complete prescription for transit arrival/departure prediction. The prescription is presented in a data flow setting and uses three components, the Tracker, the Filter and the Predictor, through which the data flows. Algorithms suitable for each component are documented, and this provides an open reference to creating accurate prediction of vehicle arrival/departure.

The paper presents results from a set of software components created to implement those described in the paper and used data from Seattle WA and Portland OR to make predictions. These examples demonstrate that the overall prescription is sufficiently general to be useful in any transit settings that (1) have a fleet of transit vehicles that travel along prescribed routes, (2) have a "transit database" that defines the schedule times and the geographical layout of every route, and (3) have an automatic vehicle location (AVL) system, where each vehicle in the fleet is equipped with a transmitter and periodically reports its progress back to a transit management center. Further, these examples demonstrate that there is a significant gain in information, over the schedule, that can be provided to passengers using the AVL data with an implementation of this prescription.

Appendix A. Searching a geographic data set using tiles

In order to create an efficient procedure for indexing and sorting a large geographic data set we introduce a new two-dimensional indexing structure that permits rapid calculation of subsets of TPI's. This structure consists of a regular grid of non-overlapping rectangular regions (tiles) superimposed over the entire region of operation of the transit fleet. Each tile is uniquely determined by two indices, and given the grid origin and tile dimensions, the indices of the tile containing any reported vehicle location can be computed quickly using simple arithmetic with no searching involved. We relate the tiles to TPI's as follows. For each tile Γ form an associated "inflated" tile Γ^* by adding a "halo" of width equal to the maximum expected position measurement error σ. For every TPI in the database associate to it the set of tiles $\{\Gamma_i\}$ such that the TPI intersects the corresponding inflated tile Γ_i^*, and also associate the corresponding intersection sub-paths of the TPI with Γ_i^*. It follows that a circular neighborhood $N(p,\sigma)$ of an arbitrary point p on a tile Γ can possibly intersect a given TPI only if that TPI is associated with tile Γ.

Fig. 16 illustrates the notions of circular neighborhood, tile and associated inflated tile. The neighborhood has a radius of 150 ft and the tile has width and height of 2000 ft.

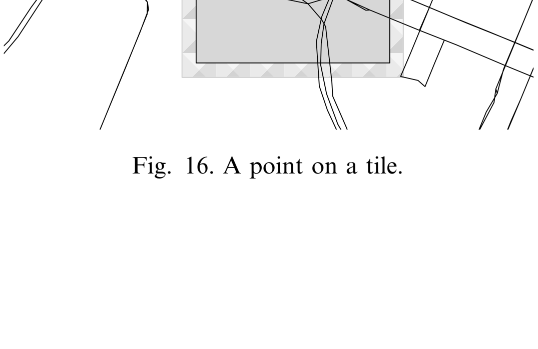

Fig. 16. A point on a tile.

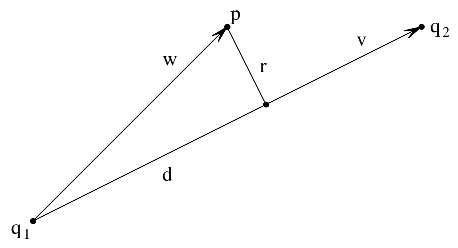

Fig. 17. Distance from point to line segment.

Appendix B. Distance algorithm

We describe a simple algorithm for computing distance r from a point p to a line segment $\overline{q_1 q_2}$ and for computing distance d into the segment for point q which corresponds to this distance. The idea is to use properties of the vector dot-product and consider the projection of p onto the directed line running through q_1 and q_2. See Fig. 17.

Let $v = q_2 - q_1$ and $w = p - q_1$ and recall the dot-product formula $w \cdot v = |w||v| \cos\theta$ where $|w| = \sqrt{(w \cdot w)}$ is the length of w, $|v| = \sqrt{(v \cdot v)}$ is the length of v, and θ is the angle between w and v. The projection of w onto the line determined by v is given $(w \cdot v / v \cdot v)v$. There are three cases to consider.

1. If $w \cdot v < 0$ then θ is obtuse and the projection of p lies before q_1 and hence q_1 is the closest point to p. Thus $r^2 = w \cdot w$ and $d = 0$.
2. If $0 \leqslant w \cdot v \leqslant v \cdot v$ then p projects onto the segment. Hence $d^2 = (w \cdot v)^2 / v \cdot v$ and $r^2 = (w \cdot w) - d^2$.
3. If $v \cdot v < w \cdot v$ then the projection of p lies beyond q_2 and hence q_2 is the closest point on the segment to p. Hence $r^2 = (w - v) \cdot (w - v)$ and $d^2 = v \cdot v$.

(Note that when this algorithm is applied to a chain of segments in the search for a closest point, step (3) can be skipped as redundant on all segments except possibly the last one.)

References

AASHTO/ITE/NEMA, 1999. Transit Communications Interface Profiles (TCIP) Framework Document (NTCIP 1400). Draft 97.01.02, Institute of Transportation Engineers.

Anderson, B., Moore, J., 1979. Optimal Filtering. Prentice-Hall, Inc.

Balogh, S., Smith, R., 1993. London transport's countdown system—a leader in the bus transport revolution. In: IEE Colloquium on 'Public Transport Information and Management Systems'. IEE, London, UK.

Bell, B.M., 2000. The marginal likelihood for parameters in a discrete Gauss–Markov process. IEEE Transactions on Signal Processing.

Carlson, R., Brastow, W., 1974. Interactive trip planning in a computerized transit information system. Bulletin of the Operations Research Society of America.

Cathey, F., Dailey, D., 2001. Transit vehicles as traffic probe sensors. In: Proceedings of the IEEE Intelligent Transportation Systems Conference 2001.

Charniak, McDermott, 1984. Introduction to Artificial Intelligence. Addison-Wesley.

Dailey, D., Haselkorn, M., 1994. Demonstration of an advanced public transportation system in the context of an ivhs regional architecture. First World Congress on Applications of Transport Telematics and Intelligent Highway Systems, pp. 3024–3031.

Dailey, D., Fisher, G., Maclean, S., 1999. Busview and transit watch two products from the Seattle smart trek model deployment. In: Proceedings of the ITS America Eighth Annual Meeting.

Dailey, D., Wall, Z., Mclean, S., Cathey, F., 2000. An algorithm and implementation to predict the arrival of transit vehicles. In: Proceedings of the IEEE Intelligent Transportation Systems Conference.

Dailey, D., Maclean, S., Cathey, F., Wall, Z., 2001. An algorithm and a large scale implementation. Transportation Research Record.

Ibrahim, D., 1996. Gps based public transport route information system. GeoComputation 96. 1st International Conference on Geo Computation. Univ. Leeds, Leeds, UK, vol. 1. pp. 443–453.

Jazwinski, A., 1970. Stochastic Processes and Filtering Theory. Academic Press.

Koncz, N., Greenfeld, O., 1995. The development of a gis-based transit advanced traveler information system. In: URISA Proceedings. Papers from the Annual Conference of the Urban and Regional Information Systems Association. Urban & Regional Inf. Syst. Assoc., Washington, DC, USA, pp. 695–709.

Maclean, S., Dailey, D., 2001. MyBus: helping bus riders make informed decisions. IEEE Intelligent Systems 16 (1), 84–87.

Oda, T., 1990. An algorithm for prediction of travel time using vehicle sensor data. In: Proceedings of the Third International Conference on Road Traffic Control, pp. 40–44.

Peng, Z., Huang, R., 2000. Design and development of interactive trip planning for Web-based transit information systems. Transportation Research Part C (Emerging Technologies) 8C (1–6), 409–425.

Rauch, H., Tung, F., Striebel, C., 1965. Maximum likelihood estimates of linear dynamic systems. American Institute of Aeronautics and Astronautics Journal 3, 1445–1450.

Stern, J.E., 1989. State Plane Coordinate System of 1983. NOAA Manual NOS NGS 5.

Wall, Z., Dailey, D., 1999. An algorithm for predicting arrival time of mass transit vehicles using avl. In: Proceedings of the Transportation Research Board Annual Meeting.