Corridor Travel Time using Transit Vehicles as Probes

F.W. Cathey
Electrical Engineering Box 352500
U of Washington
Seattle, WA 98195
fritz@its.washington.edu
Fax 206-616-1787
Phone 206-616-3185

D.J. Dailey
Electrical Engineering
Box 352500
U of Washington
Seattle, WA 98195
dan@its.washington.edu
Fax 206-616-1787
Phone 206-543-2493

ABSTRACT

In this paper we present a corridor approach to travel time estimates using transit vehicles as probes. These estimates increase the information density along the corridor over using only probe information at specified points. It provides speed estimates that track the significant changes identified in inductance loop data, but seems to provide a conservative estimate of the speed. Comparison of instantaneous travel times, often used for real-time applications, and travel time computed using a corridor speed surface indicate that the instantaneous travel times have a delay in tracking changes in the corridor and higher maximum travel time.
Corridor Travel Time using Transit Vehicles as Probes
F.W. Cathey, D.J. Dailey
Department of Electrical Engineering
University of Washington
Seattle, Washington, USA.
fritz@its.washington.edu, dan@its.washington.edu

1 Introduction

Performance monitoring is an issue of growing concern both nationally and in Washington State [1]. Travel times and speeds have always been of interest to traveler information researchers, planners and public agencies, and as a key measure in performance monitoring, this interest is now greater than ever. However, instrumenting the roadway infrastructure with inductance loops, cameras and other sensors to obtain this type of data is very expensive and hence an alternative source of travel time and speed data is desirable. In this paper, we present transit vehicles as probe sensors and develop a framework to use the probe data as corridor speed sensors and travel time estimators.

The use of probe vehicles described here differs from that presented in [2]. In the latter approach, 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 on corridor are determined for every vehicle on a corridor. This scheme provides a richer data set and is more amenable to travel time calculations.

We use the data from a transit management Automatic Vehicle Location (AVL) System, the King County Metro Transit AVL System [3], 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) [4].

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 [5]. 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 1 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 corridor. Two Corridor Travel Time applications are currently in work: (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) protocol [5].

In the next sections, 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 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.

2 Transit Database and AVL Data

Our basic assumptions in using vehicles of a mass transit system as speed probes are: (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 time point, (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.

To clarify the terminology used here, we review the description of the transit database and AVL data presented in [2]. The database is described in terms defined by the ITS Transit Communications Interface Profile (TCIP). 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,” where a shape-point is simply an unnamed location. Since one frequently needs to determine the distance of a vehicle along a path, each shape-point is augmented with its own distance into path. A pattern is a route specified by a sequence of TPI’s, where the ending time-point of the i-th TPI is the starting time-point of the (i + 1)-st TPI. 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.

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 \( T_{id} \). For tracking purposes it
is convenient to identify how far the vehicle has traveled into its block. In software upstream of the Tracker, we augment the AVL report with (7) 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 called a “sign-post” by Metro. However, the vehicle positioning technique is not critical and GPS-based AVL systems can also be used to generate the required information as shown in [4]. 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 Tracker

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 $X$ and covariance $P$, (3) last AVL report $A$, (4) number of updates $N$, (5) speed validity flag $f$.

The Kalman filter state space used here is 3-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 $R$ and $q$ are given in [2].

AVL reports for a given vehicle are received at an average rate of one per minute and track data for the vehicle is periodically propagated and output at a nominal rate of once every 20 seconds. Due to 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 [6]. 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 [2].

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 1, 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.
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. The GIS road segment database is a realization of a mathematical structure called a “directed graph.” There is a set of “nodes” that represent points on roads and a set of “arcs” that represent segments of road between nodes. Like a time-point-interval, an arc has a start node and an end node and has a sequence of shape points that define a polygonal path from start to end. The GIS database and the transit database are related as follows:

- each time-point is a GIS node and
- each time-point-interval (TPI) is the result of “welding together” a chain of oriented GIS arcs.

The orientation of an arc on a TPI is positive if the direction along the arc from start node to end node is the same as the direction of the TPI, otherwise the orientation is negative. The transit database does not specify which arcs are welded to form which TPIs. 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 2 illustrates the relationship between TPI’s and covering arcs.

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” where 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.

In order 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 [7] so that the user need not manually specify every individual arc. A corridor path can consist of hundreds of arcs.
4.3 Corridor Estimator Component

The purpose of the Corridor Estimator component is to determine which corridors the transit vehicles are traveling along and compute the locations of the vehicles with respect to these corridors. The algorithm which 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. In order 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

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

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 or not the vehicle is traveling on a corridor.

Each corridor $C$ is decomposed into a chain of arcs by joining together the corridor intervals that comprise 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 or not 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

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

Data transmitted on the Probe output stream includes: (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 not indicative of general traffic speed. This is often the case for vehicles entering and exiting the corridors.

5 Applications

In order to test the Corridor Estimator system, we defined a number of sample corridors along interstates I-5 and I-90, and along the arterial state route 99. Corridor speed data was collected and analyzed for several weeks of July, 2002.

Figure 3 is a time-space diagram of the progress of the probe vehicles on an I-5 south-bound corridor, in south Seattle, running from the West Seattle Freeway to Orillia (south of SouthCenter). The left plot is for Friday July 19 when significant congestion was observed in the corridor and the right plot is for Monday July 22 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 PM to 8 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.

We use probe vehicle data to construct an interpolating function \( v = f(x, t) \) that approximates speed, \( v \) (MPH) at any corridor location and time, \((x, t)\). Figure 4 shows a plot of the surface corresponding to this function. Note that there is a depression in the speed, centered at 5PM, 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 5. 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, 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.

A solution of the differential equation

\[
\frac{dx}{dt} = f(x, t)
\]

(1)
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 (1) 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 6 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 ten 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 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,

\[
T = \int_0^L \frac{1}{f(x, t_{\text{now}})} \, dx
\]  

(2)

where \( L \) is the length of the corridor. This type of estimate, for the data set presented here, is shown in Figure 7. Comparing Figures 6 and 7 it is clear that the instantaneous estimate lags, in time, the real increase in travel time, and further, 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.

6 Conclusion

We present a corridor approach to travel time estimates using transit vehicles as probes. These estimates increase the information density along the corridor over using only probe information at specified points. It provides speed estimates that track the significant changes identified in inductance loop data, but seems to provide a conservative estimate of the speed. Comparison of instantaneous travel times, often used for real-time applications, and travel time computed using a corridor speed surface indicate that the instantaneous travel times have a delay in tracking changes in the corridor and higher maximum travel time. This result suggests that additional research 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.

ACKNOWLEDGMENT

This work was supported by funding from Washington State Department of Transportation and Transportation Northwest (TransNow), a USDOT University Transportation Center.

References


Figure 1: Probe Corridor Design.

Figure 2: Covering arcs example.
Figure 3: Corridor Probes trajectories in space (feet into corridor) and time of day (minutes past midnight) with Friday on the left, and Monday on the right.

Figure 4: Corridor Probes 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 5: 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.

Figure 6: Corridor Probes travel time (minutes) as a function of departure time of day (minutes past midnight), Friday on the left, Monday on the right.
Figure 7: Corridor Probes 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.