Transportation Research Part C 65 (2016) 16–30

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Reconstructing maximum likelihood trajectory of probe vehicles between sparse updates

Nianfeng Wan [a,*], Ardalan Vahidi [a], Andre Luckow [b]

[a] Department of Mechanical Engineering, Clemson University, Clemson, SC 29634-0921, United States
[b] BMW Group Information Technology Research Center, Greenville, SC, United States

ARTICLE INFO

Article history:
Received 20 February 2015
Received in revised form 17 December 2015
Accepted 2 January 2016
Available online 12 February 2016

Keywords:
Probe vehicular data
Expectation Maximization
Maximum likelihood
Trajectory estimation
Transit bus

ABSTRACT

Data from connected probe vehicles can be critical in estimating road traffic conditions. Unfortunately, current available data is usually sparse due to the low reporting frequency and the low penetration rate of probe vehicles. To help fill the gaps in data, this paper presents an approach for estimating the maximum likelihood trajectory (MLT) of a probe vehicle in between two data updates on arterial roads. A public data feed from transit buses in the city of San Francisco is used as an example data source. Low frequency updates (at every 200 m or 90 s) leaves much to be inferred. We first estimate travel time statistics along the road and queue patterns at intersections from historical probe data. The path is divided into short segments, and an Expectation Maximization (EM) algorithm is proposed for allocating travel time statistics to each segment. Then the trajectory with the maximum likelihood is generated based on segment travel time statistics. The results are compared with high frequency ground truth data in multiple scenarios, which demonstrate the effectiveness of the proposed approach, in estimating both the trajectory while moving and the stop positions and durations at intersections.

1. Introduction

Traditionally costly infrastructure and dedicated sensors have been utilized to collect vehicular traffic data (Soriguera and Robusté, 2011; Singh and Li, 2012). But in recent years data from probe vehicles have provided another viable method for estimating traffic conditions (Herrera et al., 2010; Hofleitner et al., 2012a; Goodall et al., 2012). Probe vehicles are those equipped with Global Positioning System (GPS) devices on board that periodically report their coordinates and other features such as velocity and heading. Traditional loop detector sensors, for instance, can only provide flow or occupancy at fixed locations; probe vehicles on the other hand, can provide traffic information samples at varying locations.

Nowadays the technologies of wireless communication and cloud storage enable collection of probe data more efficiently. But existing probe data sets are spatiotemporally sparse. Higher penetration rate of probe vehicles and higher reporting frequencies are needed for accurate traffic estimation or vehicle control purposes, but are unlikely in the near future. High reporting frequency also raises privacy concerns (Hoh et al., 2007, 2012). Moreover, current available high frequency vehicle trajectory data usually contains large measurement errors and other errors as well. Researchers propose several methods to refine trajectories (Montanino and Punzo, 2013; Punzo et al., 2011). On the other hand, due to their low reporting frequency, current probe data, in raw form, can provide only a very incomplete picture of traffic on the road. But if vehicle trajectories between two sparse updates could be effectively reconstructed, additional virtual data points are generated that perhaps

* Corresponding author.

http://dx.doi.org/10.1016/j.trc.2016.01.010
0968-090X/

help more accurate evaluation of road traffic. One application is shown in Lu and Skabardonis (2007) where the authors use high frequency data to conduct traffic shockwave analysis. This paper proposes a method for reconstruction of vehicle trajectories between probe updates based on historical data and current traffic information such as signal timing.

There are a number of papers that address vehicular trajectory reconstruction (Huang and Tan, 2006; Mehran et al., 2012; Ni and Wang, 2008; Sun and Ban, 2013; Hao et al., 2014). For example in Huang and Tan (2006), the authors proposed a method for short term prediction of a vehicle trajectory based on the usage of nearby vehicles information. In Sun and Ban (2013) trajectories are reconstructed based on the variational formulation of kinematic waves, and results have been tested with NGSIM data and microsimulation data. In Hao et al. (2014), the authors focus on finding the most likely driving mode sequences (deceleration, idle, acceleration, and cruise), to estimate a vehicle's trajectory. The approach in Sun and Ban (2013) is macroscopic while (Hao et al., 2014) employ a microscopic-based approach.

In this paper we employ a different probabilistic microscopic-based approach, and use segment travel time statistics to reconstruct vehicle trajectories. Using historic probe data, we first estimate the travel time statistics across short road segments and also queue patterns at intersections. We then employ a maximum likelihood approach to generate the most likely trajectory of a probe vehicle between consecutive GPS updates.

Because travel time is an important measure of traffic conditions, a number of papers have focused on travel time estimation or prediction for freeways (Wan et al., 2014) and arterials (Hofleitner et al., 2012b; Sun et al., 2008; Hellinga et al., 2008; Zheng and Van Zuylen, 2013). An example of a probabilistic approach to travel time estimation can be found in Hofleitner et al. (2012a). In Coifman (2002) it is shown that travel time can be used in estimating vehicle trajectories.

Most of these existing papers estimate link travel times; the limitation is the implicit assumption of uniform distribution of travel time along an entire link. However, travel time along an arterial road may not be uniformly distributed, and for instance is higher near signalized intersections. To capture this variability, in Wan and Vahidi (2014), we proposed to divide each link to short segments of equal length and estimated statistics of travel time for each segment by using an Expectation Maximization algorithm.

In this paper we build on our earlier results reported in Wan and Vahidi (2014, 2015) to estimate the most likely trajectory of a probe vehicle between two consecutive updates. Presence of signalized intersections between the two updates complicates the problem and is influenced by factors such as queue size, signal timing and phase (Liu et al., 2009; Hao and Sun, 2011). In this paper we present solutions for when the updates span a single or multiple intersections. We evaluate the proposed algorithms using sparse updates from transit buses in the city of San Francisco. We demonstrate the effectiveness of the proposed method in comparison with high frequency data obtained from ground truth measurements of those same buses.

The rest of the paper is organized as follows: Section 2 describes the bus data feed. Section 3 explains estimation of segmental travel time statistics. Section 4 outlines our proposed method for estimating the most likely trajectory of a vehicle including the cases where the vehicle comes to a stop at an intersection queue. Section 5 presents the results as compared to ground truth measurements followed by conclusions in Section 6.

2. Description of the dataset

In this paper, we use a public data feed of transit buses in the city of San Francisco. The data contains GPS time stamp, longitude and latitude, velocity, heading and several other attributes of transit buses and is provided by NextBus. NextBus provides real-time passenger information for over 135 transit agencies and organizations in North America (NextBus, 1-15-2015). The data can be queried in almost real-time in eXtensible Markup Language (XML) interface using URLs with parameters specified in the query string.

Fig. 1 shows aggregated GPS updates from all buses in the city of San Francisco within a twenty-four hour period. As shown in the figure the coverage includes most major streets. Also the fact that buses traverse each route regularly is an advantage of using them as probe vehicles. However, these updates are sparse; at every 200 m or 90 s whichever comes first. Moreover, buses stop not only at intersections but also at bus stops which complicates the trajectory estimation problem. The bus data was among the very few publicly available data feeds that we could find and therefore was used to verify the effectiveness of our proposed algorithms.

3. Estimation of segment travel time statistics

In order to reconstruct the most likely path of a vehicle between its two updates, we first estimate the statistics of travel time for each road segment relying on historical probe data. Two successive updates of a probe vehicle provides a travel time observation. As mentioned before, most existing papers offer methods for estimating "link travel time", which is the travel time between two adjacent intersections. Their implicit assumption is that travel time is uniformly distributed along a link, which is not true in most cases. Moreover, probe vehicle updates normally occur at random positions and times and not necessarily at the two ends of a link. To address these aforementioned issues, we propose to divide a whole path into short segments, and to allocate a travel time to each segment based on probe data.

Using the haversine formula, the reported longitude and latitude coordinates are converted to a linear distance measured from an arbitrary reference point at the upstream end. Let's denote each of such segments by $x_i,\ i = 1, 2, \ldots, N$. All segments

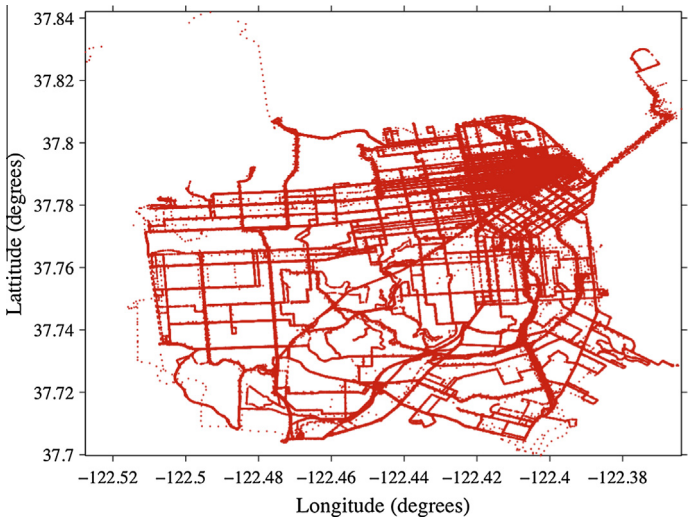

Fig. 1. Aggregated plot of transit bus updates for a period of 24 h in the city of San Francisco (Fayazi et al., 2014).

have the same length with the resolution we choose. The travel time across each segment is denoted by TT_{x_i} which is a random variable. Every observation of this random variable is denoted by $tt^j_{x_i}$, where $j$is the probe vehicle (here a bus) index. As shown in Fig. 2, there is a travel time realization for each bus pass crossing a segment.

Assume there are M bus passes along the route indexed each by $j \in \{1,2,\ldots,M\}$. Each consecutive pair of updates by bus j provides a travel time observation denoted by $tt^j_{[x_{a_1},x_{a_n}]}$ from the beginning of segment x_{a_1} to the end of segment x_{a_n}, where $a_i \in \{1,2,\ldots,N\}$. The interval $[x_{a_1},x_{a_n}]$ denotes the segments that lie between the two consecutive updates. If an update is not right on the segment boundary we assign it to the closest one. The error is acceptable as long as segments are short. The summation of travel time allocated to each segment must equal the total observed travel time, i.e.

$$tt^j_{[x_{a_1},x_{a_n}]} = \sum_{i=a_1}^{a_n} tt^j_{x_i} \tag{1}$$

To generate estimates of travel time for each segment, $tt^j_{x_i}$, we decompose the observed travel times between covered segments. We propose an Expectation Maximization (EM) algorithm to iteratively decompose each observed travel time between segments and subsequently calculate the statistics of travel time for each segment. We assume that each segment travel time has a Gaussian probability density function.[1] We also assume the conditional independence of segment travel times, that is, under the same traffic condition in the same time interval, a segment travel time is independent from all other segments.[2]

Under these assumptions, after allocating travel times to each segment, we can calculate its probability. Then we can readjust the allocation by maximizing its likelihood function, and update the mean and variance for each segment travel time.

At the very first step, segment travel times are initialized evenly, that is, for a pair of GPS updates starting at x_{a_1} and ending at x_{a_n}, we have

$$tt^j_{x_i} = \frac{1}{a_n - a_1 + 1} tt^j_{[x_{a_1},x_{a_n}]} \tag{2}$$

for $i \in \{a_1, a_2, \ldots, a_n\}$. The initialization is repeated for all observations.

Suppose $K_{x_i} \leqslant M$ is the number of overlapping observations at segment x_i, then for each segment x_i, we have K_{x_i} travel time realizations $tt^k_{x_i}$, $k \in \{1,2,\ldots,K_{x_i}\}$. Next we iterate between the following E step and M step till the segment travel times converge.

[1] We are aware that the Gaussian assumption may cause problems such as it may result in negative travel time. Fortunately, the probability is extremely low. Under the resolution we choose, the mid-link mean segment travel time is 0.5 s, the average variance is around 0.1, therefore the chance of allocating a negative travel time (outside of 5 sigma) is 0.0001%. Segment travel time at intersections are higher and have less chance to cause such a problem.

[2] The independence assumption is for simplification. A large amount of ground truth data is needed to verify how strong this assumption is. We only had access to 15 valid high frequency ground truth data traces for three different days, therefore we cannot conclude independence nor we can infer dependence. This is the weakness of this approach, although the results show good precision. More verifications are needed when more ground truth data is available.

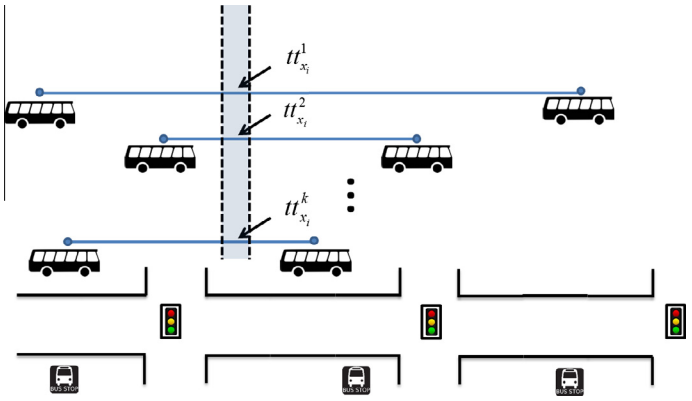

Fig. 2. A schematic display of segment travel times crowd-sourced from many bus passes.

E Step: For each segment x_i, we calculate the mean μ_{x_i} and standard deviation σ_{x_i} of allocated travel times. Having these two parameters, with the assumption that segment travel times are normally distributed, the probability density for $tt^k_{x_i}$ can be calculated as:

$$p(tt^k_{x_i}|\mu_{x_i},\sigma_{x_i}) = \frac{1}{\sqrt{2\pi}\sigma_{x_i}} \cdot e^{-\frac{(tt^k_{x_i}-\mu_{x_i})^2}{2\sigma^2_{x_i}}} \tag{3}$$

This is also the likelihood function of $tt^k_{x_i}$ given μ_{x_i} and σ_{x_i}. We often use the log-likelihood function for computational convenience, that is

$$log[p(tt^k_{x_i}|\mu_{x_i},\sigma_{x_i})] = -\frac{(tt^k_{x_i}-\mu_{x_i})^2}{2\sigma^2_{x_i}} - log(\sqrt{2\pi}\sigma_{x_i}) \tag{4}$$

M Step: In this step, for each update pair (observation), segment travel times $tt^k_{x_i}$ are reallocated such that the log likelihood function is maximized. Suppose an update pair starts at segment x_{a_1} and ends at x_{a_n}, its log likelihood function can be represented as:

$$log[p(tt^k_X|\mu_X,\Sigma_X)] = \sum_{i=a_1}^{a_n} log\left[p(tt^k_{x_i}|\mu_{x_i},\sigma_{x_i})\right] \tag{5}$$

where $X = [x_{a_1}, x_{a_2}, \ldots, x_{a_n}]^T$, $\mu_X = \left[\mu_{x_{a_1}}, \ldots, \mu_{x_{a_n}}\right]^T$, and $\Sigma_X = \begin{bmatrix} \sigma^2_{x_{a_1}} & 0 & .. & 0 \\ 0 & \sigma^2_{x_{a_2}} & .. & 0 \\ 0 & 0 & .. & \sigma^2_{x_{a_n}} \end{bmatrix}$.

Eq. (5) holds under the assumption that segment travel times are independent from each other.

In the M step, the parameters μ_{x_i} and σ_{x_i} are considered to be constants, therefore the terms containing only them can be ignored when maximizing the log-likelihood function with respect to allocated travel times. The problem then can be formulated as

$$\begin{aligned} &\arg\max_{tt^k_{x_i}} \left(\sum_{i=a_1}^{a_n} log\left[p(tt^k_{x_i}|\mu_{x_i},\sigma_{x_i})\right] \right) \\ &= \arg\min_{tt^k_{x_i}} \left(\sum_{i=a_1}^{a_n} \frac{(tt^k_{x_i}-\mu_{x_i})^2}{2\sigma^2_{x_i}} \right) \\ &= \arg\min_{tt^k_{x_i}} \left(\sum_{i=a_1}^{a_n} \frac{(tt^k_{x_i})^2 - 2tt^k_{x_i}\mu_{x_i}}{2\sigma^2_{x_i}} \right) \end{aligned} \tag{6}$$

subject to the equality constraint imposed by Eq. (1). This is a Constrained Quadratic Programming (CQP) problem. We can rewrite the problem in the standard form as,

$$\begin{aligned} &\arg\min_{y} J = \frac{1}{2} y^T Q y + c^T y \\ &\text{s.t.} \sum_{i=a_1}^{a_n} tt_{x_i}^k = tt_{[x_{a_1}, x_{a_n}]}^k \end{aligned} \tag{7}$$

where

$$y = \begin{bmatrix} tt_{x_{a_1}}^k \\ tt_{x_{a_2}}^k \\ \cdot \\ \cdot \\ tt_{x_{a_n}}^k \end{bmatrix}, \; Q = \begin{bmatrix} \frac{1}{\sigma_{x_{a_1}}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_{x_{a_2}}^2} & \dots & 0 \\ 0 & 0 & \dots & \frac{1}{\sigma_{x_{a_n}}^2} \end{bmatrix}, \text{ and } c = \begin{bmatrix} \frac{\mu_{x_{a_1}}}{\sigma_{x_{a_1}}^2} \\ \frac{\mu_{x_{a_2}}}{\sigma_{x_{a_2}}^2} \\ \cdot \\ \cdot \\ \frac{\mu_{x_{a_n}}}{\sigma_{x_{a_n}}^2} \end{bmatrix}.$$

The solution to this CQP problem is the maximum likelihood segment travel time allocation for each observation. Note that the M step is run separately for each pair of GPS updates (observation). After reallocation, the method goes back to the E step to update mean and variance for each segment based on all observations. Iterations are stopped upon convergence of the algorithm, when the difference between consecutive iterations is below a threshold. We have implemented this method in our previous work (Wan and Vahidi, 2014) for all buses (with or without stops). The segment travel time evolution and RMS (Root Mean Square) errors are shown in Fig. 3. The positions of bus stops and intersections are also shown in vertical dashed and dash-dot lines. It is shown that our EM algorithm converges fast. With the initial guess, the RMS error of average segment travel time between iterations starts at 0.2 m, and drops to 10^{-3} m at the 60th iteration. And a reasonable pattern is shown, in which segment travel times increase before bus stops and intersections.

Note that we had to impose a lower bound of 0.01 s^2 on the variance of segment travel time in the E step; otherwise the algorithm had a tendency to accumulate all the uncertainty in isolated segments.[3] This is not an unreasonable assumption as travel time for each road segment has some uncertainty in reality. The comparison between our EM algorithm and a *K*-mean-inspired algorithm in Wan and Vahidi (2014) shows consistency between results. Note that in this paper, we have improved our training procedure over (Wan and Vahidi, 2014), as shown in the next section.

4. Maximum likelihood trajectory estimation

With statistical information about segment travel time, we can proceed to estimating the most likely trajectory between two consecutive probe updates. The problem is complicated by stops and idling at a red lights or at bus stops. Stopping durations and positions are different for each bus, due to their different arrival time, queue size at intersections, and variable time spent at bus stops. Therefore, if we treat all the data similarly when estimating travel time statistics, stop durations and positions will be averaged, and uncertainties will be large around intersections and bus stops. To better capture the differences, we propose to cluster the trajectories into those with stops and those without, and to deal with them separately.

4.1. Estimation of trajectories without stops

In this paper, we label the trajectory between two consecutive updates as "unstopped" if:

1. There is no intersection between the updates and the average velocity is above a threshold (i.e. 25 mph).
2. There is an intersection between the updates and either the time stamps of the former update and the latter update are within the same green light phase, or the average velocity is larger than a threshold (i.e. 25 mph).

Note that traffic signal timings are assumed to be available in this paper; in Fayazi et al. (2014) our group demonstrated that fixed timings of traffic signals can be estimated from the Nextbus sparse updates. In Fayazi et al. (2014), bus position and velocity updates along with their timestamps are used as the inputs for estimating signal timings. This was done by reconstructing the kinematics of bus movement across an intersection and the details can be found in Fayazi et al. (2014).

We first select historical trajectories without stops and apply the method in Section 3 to estimate travel time statistics corresponding to unstopped passes. Next, given two successive updates of an "unstopped" trajectory, we reconstruct the trajectory according to the following two steps:

[3] A *lognormal* distribution for travel time is suggested in the literature. Because the parameters of a lognormal distribution are not a direct indication of segment travel time, we were not able to easily find the appropriate lower bounds for its parameters when we tried it. The algorithm also had a tendency to accumulate all the uncertainty in isolated segments.

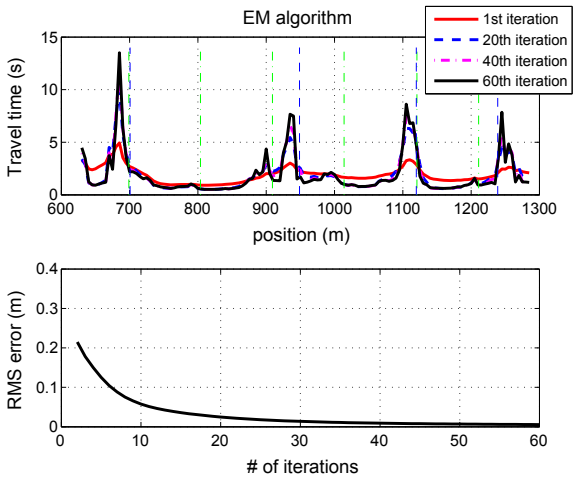

Fig. 3. Segment travel time and RMS errors evolution of our EM algorithm in Wan and Vahidi (2014).

Step 1: Given two new updates, the time stamps of the former and latter updates are denoted by $T(x_{a_1})$ and $T(x_{a_n})$ respectively. We use the time of the former update and mean segment travel times to *predict* (or estimate) the time of the latter update, denoted by $\widehat{T}(x_{a_n})$, as follows,

$$\widehat{T}(x_{a_n}) = T(x_{a_1}) + \sum_{i=a_1}^{a_n} \mu_{x_i} \tag{8}$$

where μ_{x_i} is the mean segment travel time and can obtained as explained in Section 3. We define

$$Delay_{[x_{a_1}, x_{a_n}]} = T(x_{a_n}) - \widehat{T}(x_{a_n}) \tag{9}$$

as the delay in between the update pair, where a positive delay indicates a slower than average trip, and a negative delay indicates the reverse.[4]

Step 2: We use travel time statistics for each segment in the interval $[x_{a_1}, x_{a_n}]$ and allocate the $Delay_{[x_{a_1}, x_{a_n}]}$ to each segment using a maximum likelihood approach. Similar to Eq. (1), we have

$$Delay_{[x_{a_1}, x_{a_n}]} = \sum_{i=a_1}^{a_n} \Delta tt_{x_i} \tag{10}$$

where Δtt_{x_i} is the deviation of travel time in segment x_i from its mean value μ_{x_i}. The corresponding likelihood function for this "adjusted" time Δtt_{x_i} can be obtained based on Eq. (3) and according to:

$$p(\Delta tt_{x_i} | 0, \sigma_{x_i}) = \frac{1}{\sqrt{2\pi}\sigma_{x_i}} \cdot e^{-\frac{(\Delta tt_{x_i})^2}{2\sigma_{x_i}^2}} \tag{11}$$

where $\sigma_{x_i}^2$ is the segment variance as calculated in Section 3. To obtain the maximum likelihood trajectory we need to maximize a likelihood function similar to that of Eq. (6) subject to the equality constraint in Eq. (10). This leads to another CQP problem:

$$\begin{aligned} &\arg\min_y J = \frac{1}{2} y^T Q y + c^T y \\ &\text{s.t.} \sum_{i=a_1}^{a_n} \Delta tt_{x_i} = Delay_{[x_{a_1}, x_{a_n}]} \end{aligned} \tag{12}$$

[4] Note that the usage of the word "delay" in traffic context is often referring to the difference between an actual travel time and the free flow travel time. In our proposed method we are comparing an actual travel time to the mean historic travel time. For the lack of a better word, we refer to this time difference as delay. This is delay with respect to mean historic observations and can assume positive or negative values.

where

$$y = \begin{bmatrix} \Delta tt_{x_{a_1}} \\ \Delta tt_{x_{a_2}} \\ \cdot \\ \cdot \\ \Delta tt_{x_{a_n}} \end{bmatrix}, \; Q = \begin{bmatrix} \frac{1}{\sigma^2_{x_{a_1}}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma^2_{x_{a_2}}} & \dots & 0 \\ 0 & 0 & \dots & \frac{1}{\sigma^2_{x_{a_n}}} \end{bmatrix}, \text{ and } c = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}.$$

The solution to this problem are the travel time adjustments for each segment within the given pair of updates.[5] After these adjustments, we have the most likely segment travel times between two update points, and hence the most likely trajectory.

4.2. Estimation of trajectories with a single intersection stop

A vehicle stops at an intersection if there is a queue in front and/or the traffic signal is in its red phase. The stop time and position are functions of queuing and discharging dynamics. Shock wave theory has been widely used to describe the queue dynamics (Daganzo, 2005; Skabardonis and Geroliminis, 2005). Fig. 4 shows a model for the dynamics of a queue at a red light. Here the point C is the furthest point the queue end can reach. As a traffic signal turns red, the queue starts building up linearly over time as shown by the line AC. Upon start of the next green, the vehicles start leaving the queue in a pace determined by the slope of line BC. The slope of AC is determined by the upstream traffic flow, and the slope of BC is a function of the downstream traffic condition and average acceleration of departing vehicles. In this paper, we assume that the queue discharge rate (slope of BC) is a constant determined by the mean headway and vehicle length (Jin et al., 2009). Furthermore, we assume that the furthest queue end remains constant within a certain time period of day, which we estimate from historical data for different times of day. With these we can estimate the location of point C in Fig. 4, and calculate the upstream flow (slope of AC). The sensitivity of this approach to the assumed queue length and discharge rate is discussed later in this paper when discussing the results.

In this section, we focus only on the update pairs that span one intersection. As shown in Fig. 4, a stopped update pair has potentially many candidate trajectories. Two example candidate trajectories are shown in a dotted line and a dash-dotted line respectively. Points S_1 and S_2 show when and where the bus joins and leaves the queue respectively. Once the stop position, denoted by x_{S_1}, is determined, the stop duration is determined (here $x_{S_1} = x_{S_2}$). In other words, the time coordinates of S_1 and S_2, denoted by $T(x_{S_1})$ and $T(x_{S_2})$, can be obtained. The MLT of the update pair is composed of three parts: the MLT from the former update to point S_1, the idling trajectory from S_1 to S_2, and the MLT from point S_2 to the latter update. The vehicle is assumed to be stationary between S_1 and S_2, and not in the other two intervals. In order to obtain the MLT, the following CQP is formulated and solved:

$$\begin{aligned} &\arg\min_y J = \frac{1}{2} y^T Q y + c^T y \\ &\text{s.t.} \sum_{i=a_1}^{S_1} \Delta tt_{x_i} = Delay_{[x_{a_1}, x_{S_1}]} \\ &\sum_{i=S_2}^{a_n} \Delta tt_{x_i} = Delay_{[x_{S_2}, x_{a_n}]} \end{aligned} \tag{13}$$

where

$$Delay_{[x_{a_1}, x_{S_1}]} = T(x_{S_1}) - \widehat{T}(x_{S_1}) \tag{14}$$

$$Delay_{[x_{S_2}, x_{a_n}]} = T(x_{a_n}) - \widehat{T}(x_{a_n}) \tag{15}$$

$$y = \begin{bmatrix} \Delta tt_{x_{a_1}} \\ \Delta tt_{x_{a_2}} \\ \cdot \\ \cdot \\ \Delta tt_{x_{a_n}} \end{bmatrix}, \; Q = \begin{bmatrix} \frac{1}{\sigma^2_{x_{a_1}}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma^2_{x_{a_2}}} & \dots & 0 \\ 0 & 0 & \dots & \frac{1}{\sigma^2_{x_{a_n}}} \end{bmatrix}, \text{ and } c = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}.$$

[5] Note that we solve two similar CQPs in this paper, but with different goals. The goal of the first CQP is to find mean segment travel times and their variances. The inputs are historical probe bus data, and the outputs are mean segment travel time (in general and not for individual pairs). In the second CQP the goal is to find the segment travel time for a specific pair of probe bus updates. The inputs are two consecutive data updates from the same bus. Solving the latter problem requires the result of the first problem, however, decomposing delays only occurs in the latter problem. Although both problems happened to be a CQP, they have different purposes. For example, in the first problem, the decomposition can only be positive since the variable is travel time, while in the second problem, the decomposition can be either positive or negative since the variable is delay.

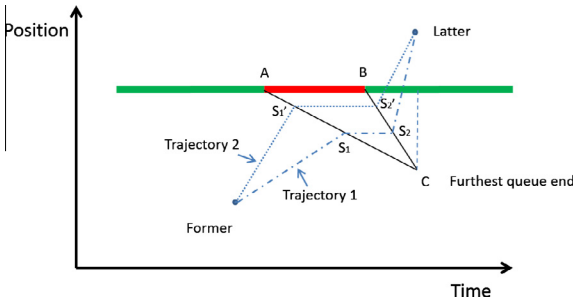

Fig. 4. A schematic of queuing and discharging dynamics at a traffic signal.

Given different stop positions in the queue, different MLTs can be obtained that have different maximum likelihood function values. We try all possible stop position choices, that is, from intersection position *A*to the furthest queue end *C*, and then choose the one with the maximum likelihood function value as the final trajectory.

4.3. Estimation of trajectories with multiple stops

If there are multiple intersections between two successive updates, we need to determine multiple stop positions and durations. Fig. 5 shows an example in which a bus experiences stops at two adjacent intersections. Here we utilize a recursive approach to solve the problem. To estimate the trajectory between the two updates, we first try all possible stop positions for the first intersection. Under the assumption of a fixed queue pattern, each candidate stop position determines S_1 and S_2. Then the problem is to find the MLT connecting point "former" to point S_1 and point S_2 to the "latter" update. In order to find the MLT from point S_2 to the "latter" update, we invoke the estimation method itself again, that is, we try all possible stop positions at the second intersection and find the one which results in the maximum likely trajectory. In other words, for each candidate stop position at the first intersection, we need to solve another set of MLT estimation problems for stopped vehicles at the second intersection. The trajectory with the maximum likelihood of all possible trajectories is then chosen as the final result. Note that the number of possible stop positions is determined by the furthest queue end at the intersection and the resolution we choose. Suppose we have n candidates for the first intersection and m for the second; the computation time complexity is $O(n \times m)$. With the increment of number of intersections, the computation time complexity grows exponentially. However, estimating a trajectory crossing many intersections is not meaningful in general since the uncertainty would dominate the result. We found that with a small number of intersections, our approach is computationally manageable.

5. Results and validation in comparison with ground truth data

To validate our approach, we chose a portion of the southbound path on Van Ness Avenue in the city of San Francisco, spanning two adjacent intersections: Lombard street and Greenwich street. Historical data was accumulated, using the sparse Nextbus feed, of bus routes 47 and 49 from January to September of 2013. The hourly segment travel time statistics for different time periods of day were then calculated. On Van Ness avenue the signals are pre-timed, and the SPaT information is gathered manually.[6] However signals have different offset times for different times of day. Our approach has taken these offsets into account when obtaining segment travel time statistics. And because of clock drift of traffic light we also used a crowdsourcing approach to estimate green initiations in near real-time based on latest bus passes (Fayazi et al., 2014). There are other sources of uncertainty beyond traffic signals, such as slow traffic, that introduce errors in our estimations. If the signals were actuated, current approach is expected to have had larger errors.

In order to evaluate the accuracy of the proposed MLT estimator, a set of high frequency ground truth data was gathered on November 3rd, 4th, and 5th, 2014. Because high frequency data was not directly available from San Francisco transit agency or other sources, we collected this data using a GPS receiver and while physically riding on multiple buses on Van Ness avenue. We used a high-sensitivity, 12-parallel-channel, USB-connected Garmin GPS 18× receiver that recorded its GPS coordinates at a frequency of 1 Hz. To ensure the best possible signal reception, we tried to place the receiver as close as possible to a window, depending on the bus occupancy level. Route 47 typically used standard buses with a length of 40 feet while route 49 buses were articulated with a length of 60 feet. To ensure consistency in GPS recordings, we did our best to place the recorder at almost the same distance from the front of a bus in every ride. After three days of recording, 15 valid high frequency GPS tracks had been recorded.

[6] In many cities the potential exists to obtain signal timings in real-time from the traffic control center. Alternatively in our group we have shown that timing of fixed time signals can be inferred from sparse vehicular probe data (Fayazi et al., 2014).

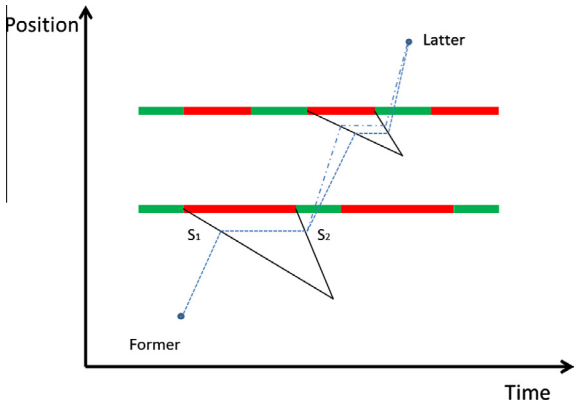

Fig. 5. A schematic of queuing and discharging dynamics at multiple intersections.

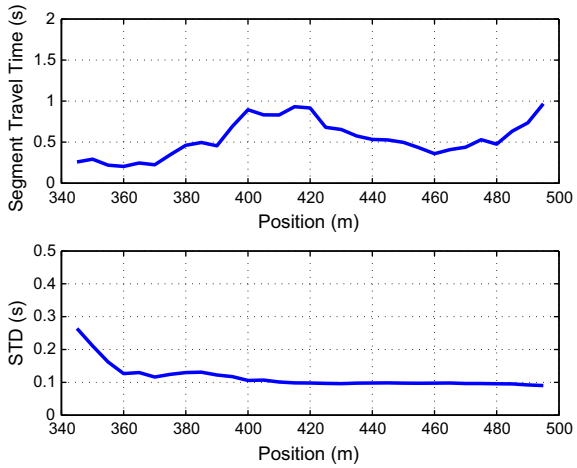

Fig. 6. Travel time statistics for unstopped trajectories. With respect to the reference point, the Lombard intersection lies at 377 m, and Greenwich intersection lies at 480 m.

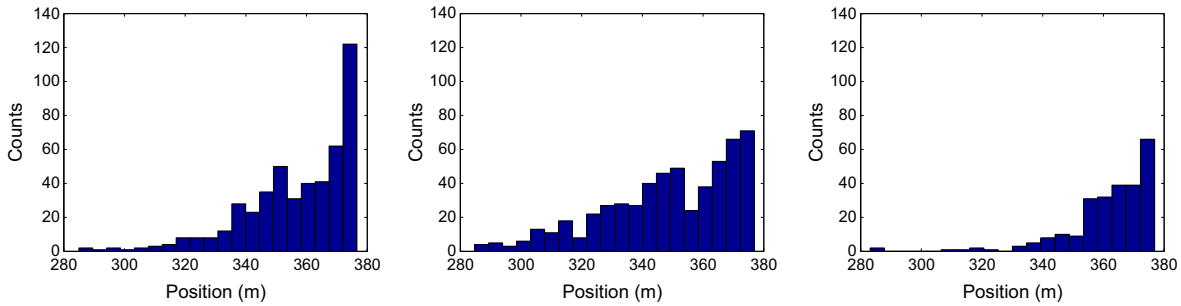

Fig. 7. Histogram of position of buses stopped at Lombard intersection as captured from January to September of 2013.

We first estimate the statistics of segment travel times using only the unstopped trajectories of the training data. The unstopped trajectories are selected based on the two steps outlined in Section 4.1 and using a lower velocity threshold of 6.5 m/s. Each segment length is chosen to be 5 m. Fig. 6 shows the mean travel time allocation and its standard deviation after convergence of the EM algorithm for the update pairs that qualify the “no stop” conditions.

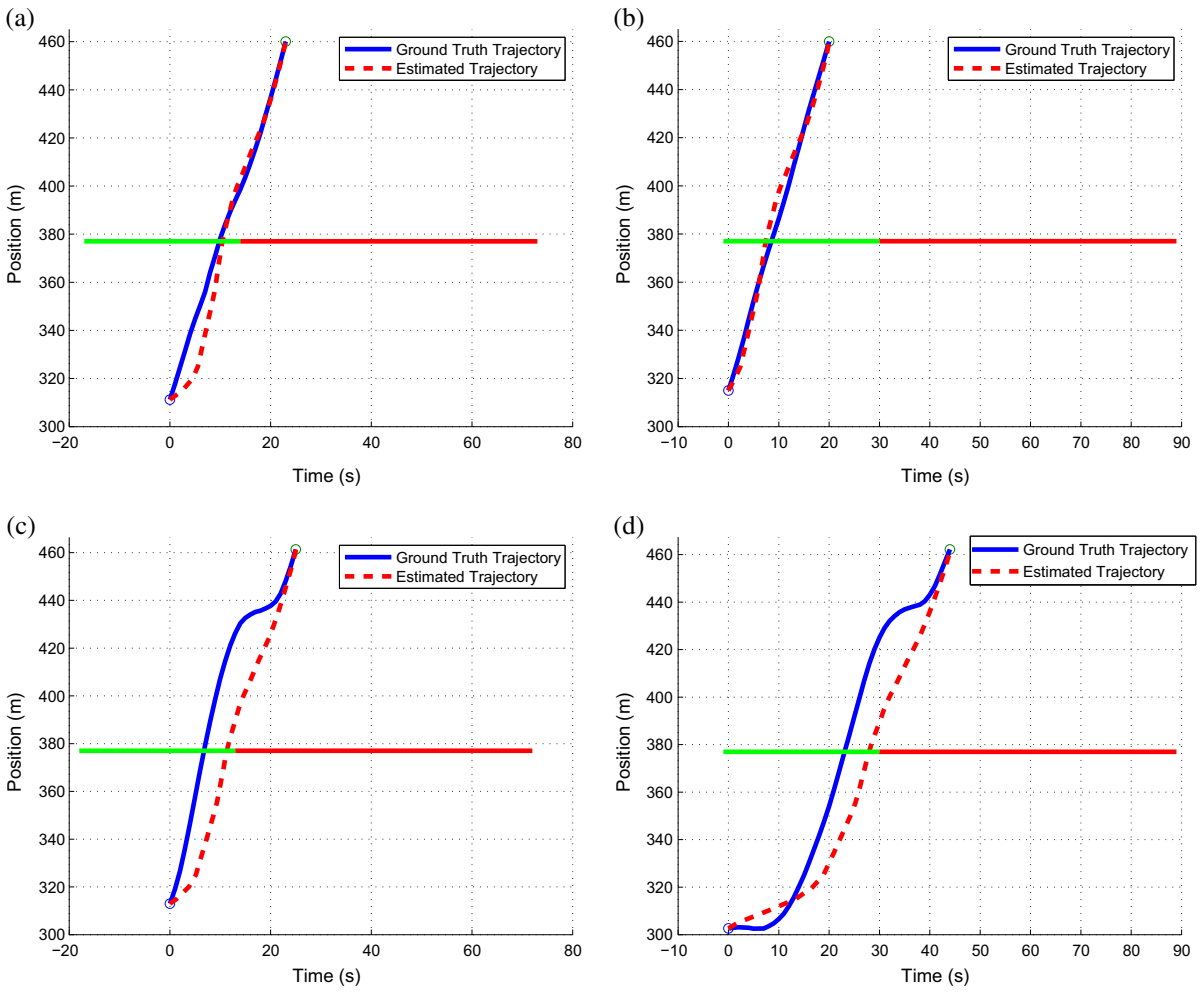

Fig. 8. Estimated and ground truth trajectories spanning a single intersection with no stop.

The maximum queue size at each intersection is also learned from training data. For example, Fig. 7 shows a position histogram of buses with zero velocity at Lombard intersection over a time span of a few months, and for three different times of day. From this histogram an estimate for the furthest queue end during the morning rush hour is 80 m and is what we use in this paper for estimating trajectories in the specific hour of 8:00–9:00 AM. To calculate the discharging rate of the queue (slope of line BC in Fig. 4), we assumed a headway of 1.4 s and average vehicle length of 5.5 m (Jin et al., 2009), resulting in the queue discharge rate of 3.9 m/s. Here we assume that all intersections have the same discharging rate.

To estimate the spatiotemporal location of a queue the timing of the traffic signal is also needed as seen in Fig. 4. The signals at Lombard and Greenwich intersections are fixed time and their baseline timing was known to us from the city timing cards. The starting time of a green phase was estimated once using ground truth measurements and then cyclically mapped forward in time as explained in Fayazi et al. (2014). To correct for potential clock drift of the traffic signal, the green initiation time could be periodically corroborated/adjusted by a crowd-sourcing algorithm that estimates initiation of a green phase based on probe data in real-time. Details and performance of such an algorithm are described in Fayazi et al. (2014).

Given travel time statistics for each segment, traffic signal timings, and historical queue patterns, we calculated the most likely trajectory between two distant updates. The results were compared against high resolution ground truth recordings. Fig. 8 shows this comparison for 4 unstopped trajectories spanning a single intersection. The estimations are close to ground truth readings for the most part. Note that there is an unidentified short slow downs happening at 440 m in 8(c) and (d), and at 300 m in 8(d). These unusual mid-link delays could not be captured in historical data, therefore the estimator did not locate the delay and it just distributed it between segments.

In the second scenario we estimate stopped trajectories that span a single intersection. Fig. 9 shows six trajectories spanning Lombard intersection. It can be seen that our approach finds the stop position in queue and stop duration with little

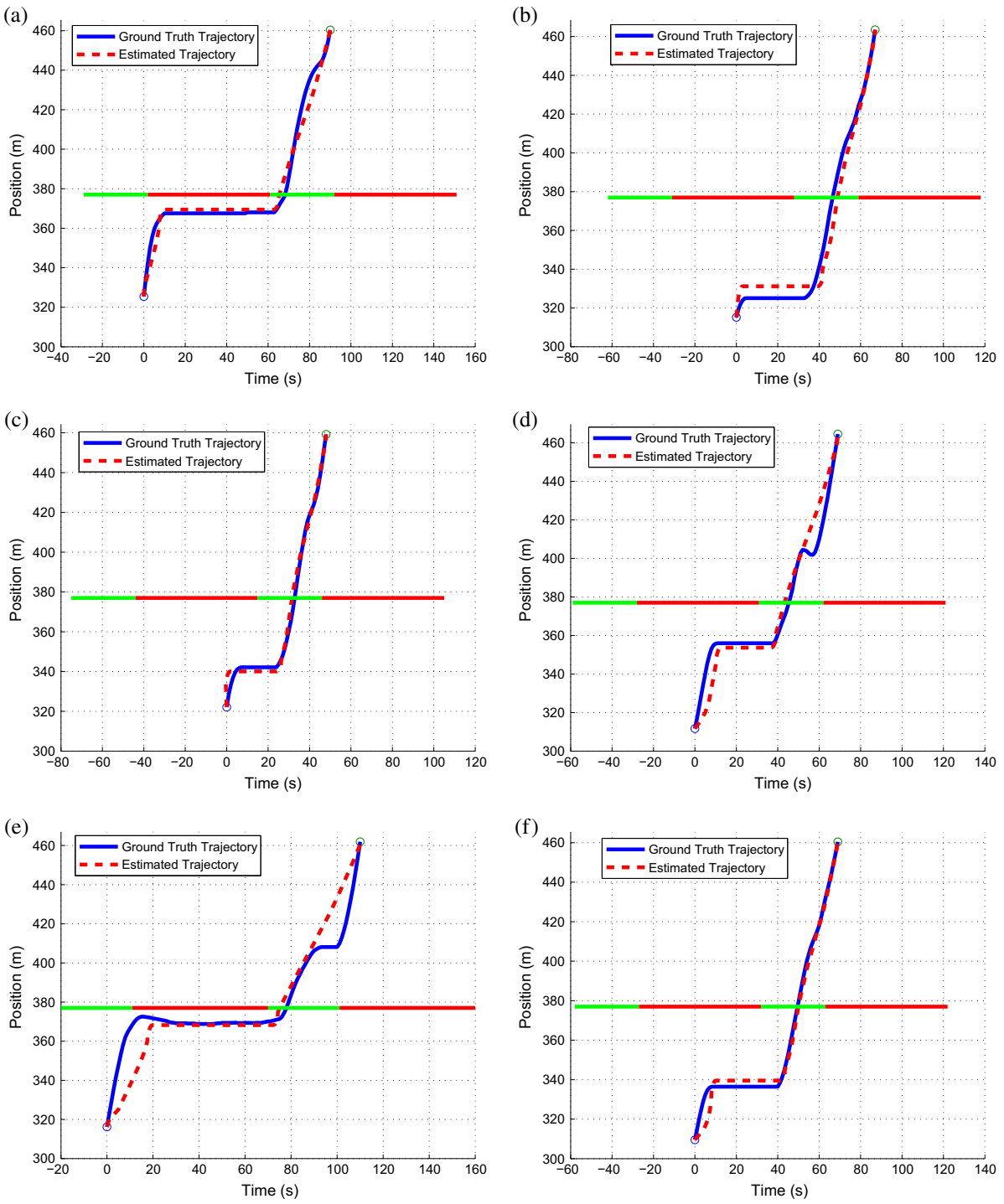

Fig. 9. Estimated and ground truth trajectories spanning a single intersection with a stop.

error in most cases. Note that our estimator also takes acceleration and deceleration constraints into account. This helps accuracy of the estimator right before and right after a stop position.

The third scenario considers trajectories spanning multiple intersections. Fig. 10 shows buses crossing Lombard and Greenwich intersections with one or two stops. Again the estimator is able to capture the actual trajectory well in most instances. Even in 10(d) where the bus experiences stops at each intersection, the estimator is able to detect relatively

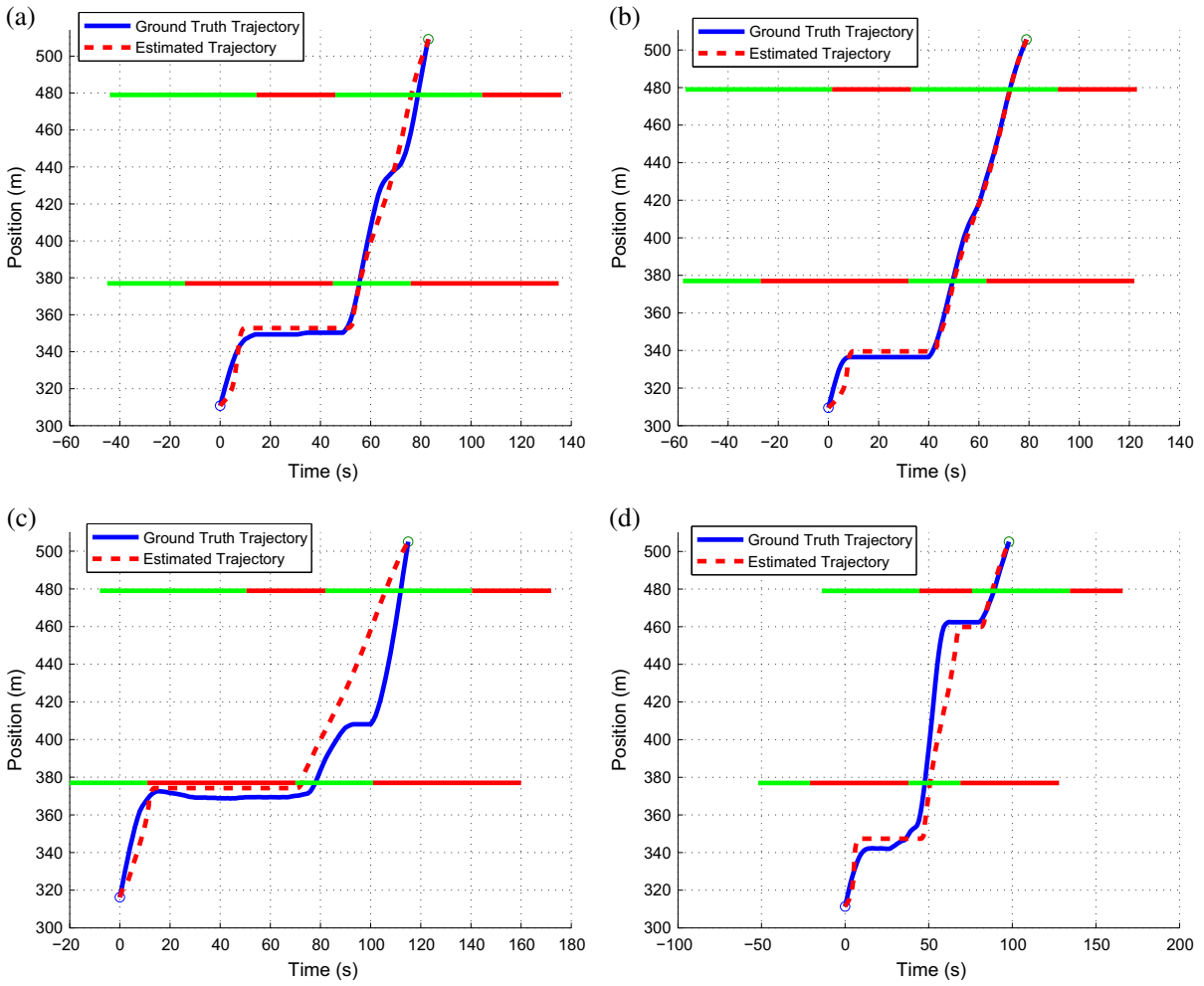

Fig. 10. Estimated and ground truth trajectories spanning multiple intersections.

accurately the position in queue at each intersection. To emphasize, the results illustrate that our approach has convincing estimation performance even when intersections lie in between the updates.

To further demonstrate the effectiveness of our approach, a benchmark (naive) approach is proposed here. The naive approach assumes that signal timings are known and vehicles travel at average free flow speed (e.g. 7.5 m/s). It uses shock-wave theory to determine the stop position and duration in queues. The queue length is assumed to be the historical average. Fig. 11 compares the results in different scenarios. It can be seen that although our approach has a similar performance when estimating moving vehicle trajectories, it is more accurate in estimating stop locations and stop duration at intersections.

To quantitatively evaluate the estimation results, we calculate the Mean Absolute Error (MAE) between the estimated and the ground truth trajectories. The MAE is defined as:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|e_i| = \frac{1}{n}\sum_{i=1}^{n}|\hat{x}_i - x_i| \tag{16}$$

where $\hat{x}_i$ is the estimated position and x_i is the actual position at time step i.

We list MAEs for each path in Table 1. The stop position in queue is also important for stopped buses and is included in Table 1.

The estimation accuracy is affected by various factors. Obviously the likelihood of errors increases with a longer distance between two updates. Moreover, the uncertainty increases with larger number of stops. The time period of day also influences the results, since during off peak hours the queue size may have more cycle variability, and the assumption of a fixed queue pattern does not hold. We assumed the traffic signal timing clock time is accurately predictable, but there can be slow drifts in a signal clock that would skew our results. The ground truth GPS data itself contains errors. For example, Fig. 9(d)

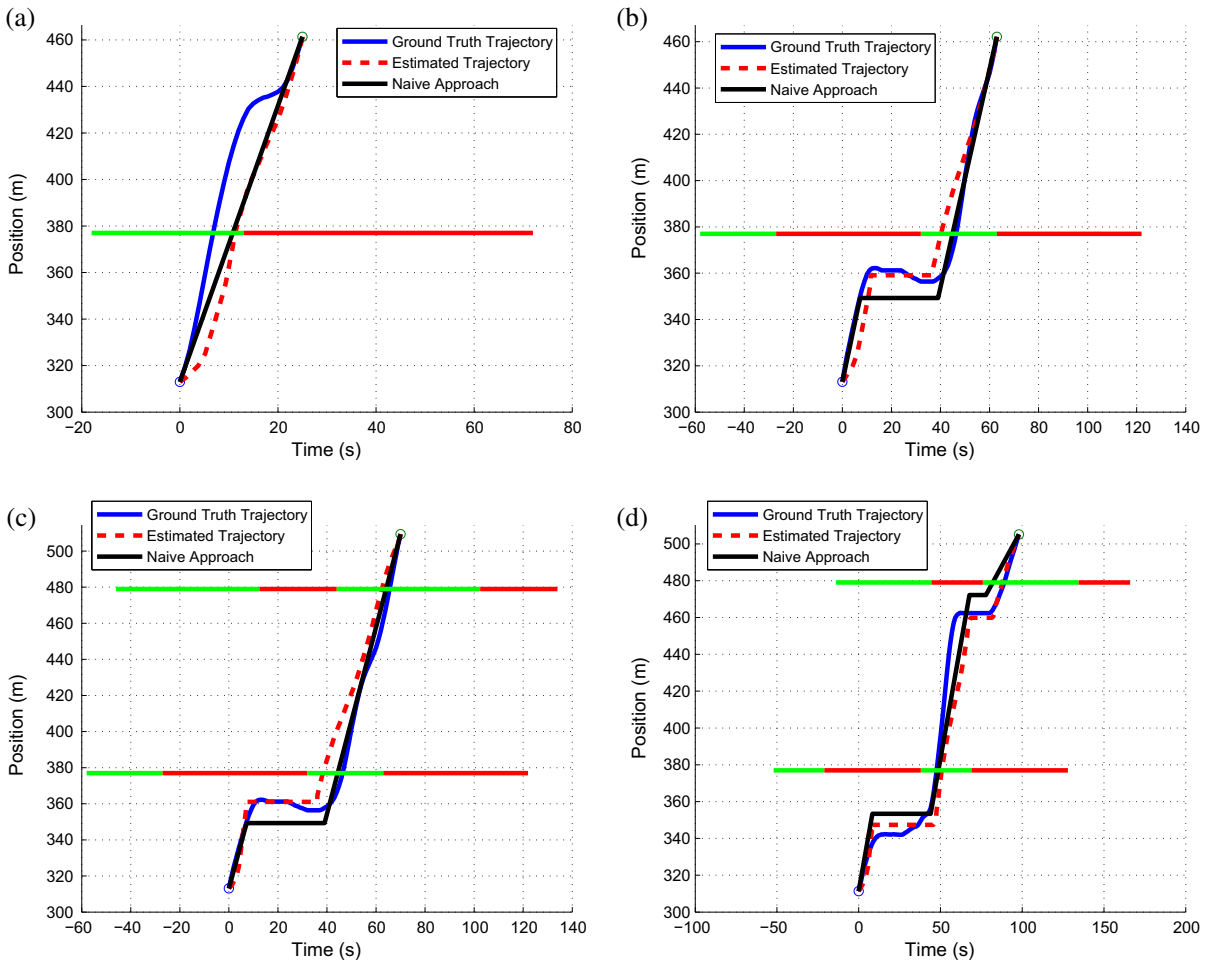

Fig. 11. Comparison of the maximum likelihood trajectory estimation and a naive approach.

Table 1
Trajectory estimation errors.

Pass ID	MAE (m)	# of stops	Error-in-Queue (m)
1	4.78	0	–
2	3.50	1	1.86
3	24.06	0	–
4	6.62	1	6.14
5	3.81	0	–
6	6.53	1	2.34
7	2.49	0	2.05
8	6.54	1	4.33
9	7.57	0	–
10	7.44	1	0.61
11	3.61	1	3.11
12	5.93	1	3.40
13	6.60	1	8.00
14	3.40	1	0.64
15	11.25	2	5.25, 2.48
Average	6.94		3.36

and (e) shows that the ground truth position decreases over some time intervals, indicating that the bus traveled backwards, which is highly improbable and could be due to missing or erroneous GPS updates.

One may have the concern that we only use average headways and queue patterns in our approach, however in reality they are not constant, are affected by traffic congestion levels and vary with time. We are aware that such assumptions

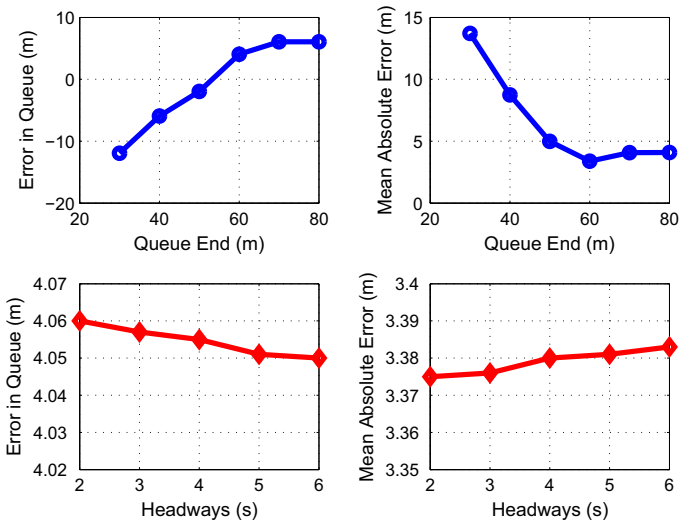

Fig. 12. Sensitivity analysis with different headways and queue ends. It is shown that the results are not highly sensitive to headways. The estimation of stop position is sensitive to the queue end.

introduce errors. However our approach relies only on historical probe data, which does not contain more information. We have also performed a sensitivity analysis on the queue end and headways. Given two updates, we use different queue end and headways to estimate the trajectory and calculate the errors in stop position and the mean absolute errors. The results are shown in Fig. 12.

The computation time is another factor to be evaluated. The computer we used had I5-2310 4 cores 2.9 MHz CPU, 8 Gb memory, 1 Tb Hard Disk and Windows 7 Home Premium 64 bit operation system. For the mean segment travel time CQP, in which the link length is 800 m, containing 9 intersections or bus stops, using 6 months probe bus data, the overall computational time is 40 min which can be done off line. For each pair of updates, the delay decomposition CQP computational time is from 5 to 40 s, depending on the length between the updates and whether there are signalized intersections in between or not. These numbers show that our approach has the potential to be applied in real time. For example in a connected vehicle scheme, a cloud server can solve the problem within seconds. Vehicles connected to the server can receive the trajectory estimation results before entering the link.

6. Conclusions

This paper presented a new method for reconstructing the trajectory of vehicles between sparse updates using a maximum likelihood approach. We introduced an iterative method for estimating travel time statistics for short segments of an arterial road using only sparse updates from probe vehicles. Using the same data feed we could estimate historical queue patterns at intersections. Relying on travel time and queue statistics and prior knowledge of signal timings, we described an algorithm that effectively estimates the most likely trajectory of vehicles in between sparse position updates. Moreover the estimated trajectories spanning an intersection could capture the stop-and-go pattern induced by traffic signals and the most likely position of the vehicle in the queue.

The results are consistent with ground truth readings and are quite promising given uncertainties arising from traffic signal drift, variability in queue size, inconsistencies in GPS recordings, and the limitation posed by the assumptions we have made when constructing the estimator.

Acknowledgments

This research was partly sponsored by BMW Information Technology Research Center in South Carolina. The authors would like to thank Mr. Seyed Alireza Fayazi, who has collected, parsed, and preprocessed the probe bus data and traffic signal timings for the use of this paper.

References

Coifman, B., 2002. Estimating travel times and vehicle trajectories on freeways using dual loop detectors. Transport. Res. Part A: Policy Pract. 36 (4), 351–364.

Daganzo, C.F., 2005. A variational formulation of kinematic waves: basic theory and complex boundary conditions. Transport. Res. Part B: Methodol. 39 (2), 187–196.

Fayazi, S., Vahidi, A., Mahler, G., Winckler, A., 2014. Traffic signal phase and timing estimation from low-frequency transit bus data. IEEE Trans. Intell. Transport. Syst. PP (99), 1–10.
Goodall, N., Smith, B.L., Park, B., 2012. Microscopic estimation of freeway vehicle positions using mobile sensors. In: 91st Annual Meeting of the Transportation Research Board, Washington, DC.
Hao, P., Boriboonsomsin, K., Wu, G., Barth, M., 2014. Probabilistic model for estimating vehicle trajectories using sparse mobile sensor data. In: 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1363–1368.
Hao, P., Sun, Z., et al, 2011. Real time queue length estimation for signalized intersections using travel times from mobile sensors. Transport. Res. Part C: Emerg. Technol. 19 (6), 1133–1156.
Hellinga, B., Izadpanah, P., Takada, H., Fu, L., 2008. Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments. Transport. Res. Part C: Emerg. Technol. 16 (6), 768–782.
Herrera, J.C., Work, D.B., Herring, R., Ban, X.J., Jacobson, Q., Bayen, A.M., 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: the mobile century field experiment. Transport. Res. Part C: Emerg. Technol. 18 (4), 568–583.
Hofleitner, A., Herring, R., Abbeel, P., Bayen, A., 2012a. Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network. IEEE Trans. Intell. Transport. Syst. 13 (4), 1679–1693.
Hofleitner, A., Herring, R., Bayen, A., 2012b. Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach. In: 91st Transportation Research Board Annual Meeting. No. 12-0798.
Hoh, B., Gruteser, M., Xiong, H., Alrabady, A., 2007. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In: Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM, pp. 161–171.
Hoh, B., Iwuchukwu, T., Jacobson, Q., Work, D., Bayen, A.M., Herring, R., Herrera, J.-C., Gruteser, M., Annavaram, M., Ban, J., 2012. Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines. IEEE Trans. Mob. Comput. 11 (5), 849–864.
Huang, J., Tan, H.-S., 2006. Vehicle future trajectory prediction with a DGPS/ins-based positioning system. In: American Control Conference, 2006. IEEE, pp. 6–pp.
Jin, X., Zhang, Y., Wang, F., Li, L., Yao, D., Su, Y., Wei, Z., 2009. Departure headways at signalized intersections: a log-normal distribution model approach. Transport. Res. Part C: Emerg. Technol. 17 (3), 318–327.
Liu, H.X., Wu, X., Ma, W., Hu, H., 2009. Real-time queue length estimation for congested signalized intersections. Transport. Res. Part C: Emerg. Technol. 17 (4), 412–427.
Lu, X.-Y., Skabardonis, A., 2007. Freeway traffic shockwave analysis: exploring the NGSIM trajectory data. In: 86th Annual Meeting of the Transportation Research Board, Washington, DC.
Mehran, B., Kuwahara, M., Naznin, F., 2012. Implementing kinematic wave theory to reconstruct vehicle trajectories from fixed and probe sensor data. Transport. Res. Part C: Emerg. Technol. 20 (1), 144–163.
Montanino, M., Punzo, V., 2013. Making NGSIM data usable for studies on traffic flow theory: multistep method for vehicle trajectory reconstruction. Transport. Res. Rec.: J. Transport. Res. Board (2390), 99–111.
NextBus, 1-15-2015. <www.nextbus.com> (accessed on).
Ni, D., Wang, H., 2008. Trajectory reconstruction for travel time estimation. J. Intell. Transport. Syst. 12 (3), 113–125.
Punzo, V., Borzacchiello, M.T., Ciuffo, B., 2011. On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (NGSIM) program data. Transport. Res. Part C: Emerg. Technol. 19 (6), 1243–1262.
Singh, K., Li, B., 2012. Estimation of traffic densities for multilane roadways using a markov model approach. IEEE Trans. Ind. Electron. 59 (11), 4369–4376.
Skabardonis, A., Geroliminis, N., 2005. Real-time estimation of travel times on signalized arterials. In: Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic Theory, 2005.
Soriguera, F., Robusté, F., 2011. Estimation of traffic stream space mean speed from time aggregations of double loop detector data. Transport. Res. Part C: Emerg. Technol. 19 (1), 115–129.
Sun, L., Yang, J., Mahmassani, H., 2008. Travel time estimation based on piecewise truncated quadratic speed trajectory. Transport. Res. Part A: Policy Practice 42 (1), 173–186.
Sun, Z., Ban, X.J., 2013. Vehicle trajectory reconstruction for signalized intersections using mobile traffic sensors. Transport. Res. Part C: Emerg. Technol. 36, 268–283.
Wan, N., Gomes, G., Vahidi, A., Horowitz, R., 2014. Prediction on travel-time distribution for freeways using online expectation maximization algorithm. In: Transportation Research Board 93rd Annual Meeting, No. 14-3221.
Wan, N., Vahidi, A., 2014. Probabilistic estimation of travel times in arterial streets using sparse transit bus data. In: 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), pp. 1292–1297.
Wan, N., Vahidi, A., 2015. Maximum likelihood estimation of vehicle trajectory at intersections using sparse transit bus data. In: Transportation Research Board 94th Annual Meeting, No. 15-3180.
Zheng, F., Van Zuylen, H., 2013. Urban link travel time estimation based on sparse probe vehicle data. Transport. Res. Part C: Emerg. Technol. 31, 145–157.