Transportation Research Part C 57 (2015) 85–102

Contents lists available at ScienceDirect

Transportation Research Part C

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

Lane-based real-time queue length estimation using license plate recognition data

Xianyuan Zhan [a], Ruimin Li [b], Satish V. Ukkusuri [a,*]

[a] Lyles School of Civil Engineering, Purdue University, 550 Stadium Drive, West Lafayette, IN 47907, USA
[b] Department of Civil Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 14 February 2015
Received in revised form 28 May 2015
Accepted 2 June 2015

Keywords:
License plate recognition data
Real-time queue length estimation
Lane-based estimation
Cumulative arrival–departure curve
Gaussian process
Boundary constrained car-following model

ABSTRACT

License plate recognition (LPR) data are emerging data sources that provide rich information in estimating the traffic conditions of urban arterials. While large-scale LPR system is not common in US, last few years have seen rapid developments and implementations in many other parts of world (e.g. China, Thailand and Middle East). Due to privacy issues, LPR data are seldom available to research communities. However, when available, this data source can be valuable in estimating real-time operational metrics in transportation systems. This paper proposes a lane-based real-time queue length estimation model using the license plate recognition (LPR) data. In the model, an interpolation method based on Gaussian process is developed to reconstruct the equivalent cumulative arrival–departure curve for each lane. The missing information for unrecognized or unmatched vehicles is obtained from the reconstructed arrival curve. With the complete arrival and departure information, a car-following based simulation scheme is applied to estimate the real-time queue length for each lane. The proposed model is validated using ground truth information of the maximum queue lengths from the city of Langfang in China. The results show that the model can capture the variations in queue lengths in the ground truth data, and the maximum queue length for each signal cycle can be estimated with a reasonable accuracy. The estimated queue length information using the proposed model can serve as a useful performance metric for various real-time traffic control applications.

1. Introduction

Queue length has long been recognized as a crucial factor to measure the performance of signalized intersections (Balke et al., 2005) or signal optimization (Webster, 1958;Newell, 1965; Chang and Lin, 2000;Mirchandani and Zou, 2007). Previous studies in queue length estimation rely on information from data sources such as loop detector data (Skabardonis and Geroliminis, 2008), event-based signal and vehicle detection data (Balke et al., 2005; Liu et al., 2009) and mobile traffic sensors (Ban et al., 2011). The first two approaches typically use aggregate level traffic information (e.g. traffic flow, density and velocity) from the fixed sensors, which have limited coverage and incur extra cost for installing additional physical infrastructures. Using data from mobile sensors (e.g. GPS data) on the other hand, does not need fixed sensors and have relatively larger coverage. However, it has its own limitations for practical implementation, such as the need of dedicated fleet of GPS equipped vehicles, low penetration rates and the difficulty in real-time gathering and processing of the decentralized data.

* Corresponding author. Tel.: +1 765 494 2296; fax: +1 765 496 7996.
E-mail addresses: zhanxianyuan@purdue.edu (X. Zhan), lrmin@tsinghua.edu.cn (R. Li), sukkusur@purdue.edu (S.V. Ukkusuri).

http://dx.doi.org/10.1016/j.trc.2015.06.001
0968-090X/

License plate recognition (LPR) data are emerging data sources in urban and highway transportation systems. Due to a wide range of traffic applications, including automatic toll collection, law enforcement, traffic monitoring and emergency operation, few years have seen rapid deployments of large-scale LPR systems in many countries. For example, Beijing has a LPR system with 374 high definition (HD) LPR cameras in 2010, and this number is projected to increase to 3000 by 2015 (Beijing Municipal Committee, 2012). Such LPR systems provide a novel and rich source of data that enable recording of the vehicle departing timestamps at the stop line of an intersection, and allowing for the re-identification of the same vehicle passing consecutive intersections. There are three unique features of the LPR data: first, at the intersection level, the LPR cameras record highly accurate and complete timestamp sequences for all vehicles that depart from the stop line on each lane of the road; second, the recorded license plate numbers enable the tracking of the same vehicle between upstream and downstream intersections, from which the link travel times can be easily obtained; third, since the LPR cameras monitor the stop line for each lane of the arterial, detailed lane-based traffic state estimation can be achieved. Given the unique features for the LPR data, link travel time estimation can be easily achieved and has already been explored in several pioneering researches using LPR data (Bertini et al., 2005; Yasin et al., 2009). However, due to privacy issues, LPR data are seldom available to the research community, and there have been limited research efforts to develop data analytics for more complex transportation applications, such as real-time arterial queue length estimation. On the other hand, because of the lack of related modeling approaches, such valuable data are only recorded and used for law enforcement by local transportation agencies. LPR data provide much better vehicle re-identification accuracy given the uniquely identified license plate numbers compared with other vehicle re-identification data sources, e.g. dual loop detectors (Coifman and Cassidy, 2002; Coifman and Krishnamurthy, 2007) and vehicle signature (Oh et al., 2007; Kwong et al., 2009; Jeng et al., 2010). This is because that the re-identification of a vehicle using LPR data is based on unique license plate information, whereas in other approaches the identifiers suffer from various inaccuracies. For instance, vehicle length is not a unique identifier for a vehicle, and the vehicle length estimation through dual loop detectors is also prone to error. According to Coifman and Cassidy (2002), the length measurements obtained from dual loop detectors may only be accurate to two feet (or worse). On the other hand, the magnetic signal (vehicle signature approach) is also not necessarily unique for each vehicle, and prone to magnetic disturbance (Kwong et al., 2009) which is impacted by various factors, e.g. vehicle speed, resolution of the sensor and environmental conditions. Furthermore, LPR data do not suffer from the issue of low penetration rate that is present in GPS probe vehicle data (Ban et al., 2011; Hofleitner et al., 2012), since LPR cameras record the passing timestamps of almost all vehicles departing from an intersection, whereas the GPS probe vehicle data only monitor a small fraction of vehicles in the traffic. Moreover, since the LPR data are recorded and maintained in a centralized system, this overcomes the implementation difficulties of real-time data gathering and processing that often arises in decentralized GPS data based approaches. These features of LPR data offer significant value in developing new analytical models that utilize LPR data for measuring urban operational performance metrics, such as the real-time arterial queue lengths.

In literature, the queue length estimation methods can be generally classified into two categories (Liu et al., 2009; Ban et al., 2011): the input–output models (Webster, 1958; May, 1975; Akcelik, 1999; Sharma et al., 2007; Vigos et al., 2008) and shockwave models (Lighthill and Whitham, 1955; Richards, 1956; Stephanopoulos and Michalopoulos, 1979; Skabardonis and Geroliminis, 2008; Liu et al., 2009; Ban et al., 2011). The input–output models estimate the queue lengths based on the analysis of cumulative traffic input–output (arrival–departure curve) to a link. This type of models have a simplistic conceptual nature, however, are limited by the inability to capture the spatial queuing in the actual arterial traffic. The shockwave models, which are receiving more attention recently, explain the formation and dissipation of queuing using traffic shockwave theory. The shockwave models provide a nicer analytical framework for queue length estimation, however, may not be directly applicable to the LPR data. This is because that the information obtained from LPR data (discrete vehicle arrival and departure times) are more similar to the type of information used in the input–output models, while the shockwave analysis typically requires more aggregate traffic condition measurements, e.g. flow density, speed and traffic volume, etc. Thus the shockwave analysis cannot fully utilize the accurate and rich vehicle level information in LPR data.

In this study, we propose a new hybrid framework that combines the advantages of both input–output models and shockwave models for real-time queue length estimation, thereby fully utilizing the valuable information provided in the LPR data. The model proposed in this paper estimates the real-time queue length for each lane of an intersection using the LPR data from two consecutive intersections. An interpolation model based on Gaussian process is first developed to reconstruct the equivalent cumulative arrival–departure curve for each lane. The missing arrival times for the unmatched arrival vehicles with unrecognized or erroneously recognized license plate numbers can then be inferred from the reconstructed arrival curve. Next, a boundary constrained car-following model is developed based on a simplified first-order Optimal Velocity (OV) car-following model developed by Tordeux et al. (2015), which enables the overall model to capture the formation and dissipation of spatial queues at the intersection. With the complete arrival–departure information used as boundary conditions, the real-time queue length for each lane can be obtained in the generated trajectories from the car-following model. The LPR data of two consecutive intersections of a 720 m arterial segment in the city of Langfang, Hebei province, China is used to test the model. Two field experiments were conducted to obtain the ground truth cycle maximum queue length data to validate the proposed queue length estimation model.

This work contributes to the literature in following aspects:

1. One of the first studies that exploit and utilize the unique features of LPR data for arterial queue length estimation.
2. Hybrid modeling framework that combines both statistical machine learning technique and well-established traffic flow theory in queue length estimation.
3. Detailed lane level queue length estimates can be obtained from the model, and allow for efficient real-time implementation.
4. Real world LPR data and ground truth queue length data from two field experiments are used for testing and validating the model. Similar types of testing and ground truth data have never been used in the literature.

The paper is organized as follows: the next section presents the methodology of the proposed model; the third and the fourth section present the field experiment design and numerical results; the final section concludes the paper.

2. Methodology

This section presents the details of the proposed lane-based real-time queue length estimation model. In this paper, we refer to the queue length as the number of vehicles in the queue. There are two major modules in the queue length estimation procedure:

(1) Equivalent cumulative arrival–departure curve reconstruction
(2) Queue length estimation via boundary constrained car-following model

In typical LPR systems, a virtual detection zone is set right behind the stop line of each lane, which has the size usually smaller or equal to the area occupied by a vehicle. The LPR camera will take a picture only when a vehicle passes the virtual detection zone. A stationary vehicle that covers the virtual detection zone will not be recorded until it passes the zone. Thus the accurate timestamps when vehicles pass the stop line can be obtained as well as their recognized license plate number. There are certain recognition errors due to lighting (day or night), weather, motion of vehicles etc. Such recognition errors will either result in unrecognizable or erroneously recognized license plate number. Both cases lead to vehicle records with unmatched license plate number between upstream and downstream intersections (referred as *unmatched vehicle*). However, the timestamp of each vehicle passing the intersection can be recorded with very high precision even when the license plate of the vehicle is not properly recognized. Consequently, an accurate departure curve from the downstream intersection of the road can be obtained (since only the passing timestamp information is used). On the other hand, it is insufficient to get the accurate arrival curve for the upstream intersection due to the missing information from the unmatched arrival vehicles. In worst cases, such unmatched arrival vehicles can account for as many as 50–70% of all arrival vehicles in certain situations. A more detailed investigation of the proportion of unmatched arrivals using the field experiment data can be found in Section 3. The failure of matching between the downstream departure and upstream arrival vehicles is due to the following reasons: (1) the failure to recognize the license plate number, which results in vehicles with plate numbers recorded as "unrecognized", such vehicles are referred as *unrecognized vehicles* in the following; (2) erroneously recognized license plate numbers and (3) vehicles enter from the lanes that are not monitored, such as those that take the right-turn lane, as right turning movement on red is often permitted and not monitored. The accurate arrival–departure information is critical for estimating queue length on each lane. The incomplete information from the arrival process prohibits the direct implementation of LPR data for queue length estimation, thus the reconstruction of the arrival process becomes necessary. We propose an *equivalent cumulative arrival–departure curve reconstruction* sub-model based on the Gaussian process. This model infers and reconstructs the equivalent arrival process that satisfies the first-in-first-out (FIFO) rule. Due to the effect of lane changing behaviors, the FIFO is generally not satisfied in the actual LPR data, which introduces significant difficulties in modeling queuing process on the lane level. In this study, we consider the equivalent arrival times rather than the actual arrival times. The equivalent arrival time is not the actual arrival time, but the most likely arrival time by assuming the vehicle in consideration follows the same arrival process characterized by the information provided in all other observed peer arrival vehicles on the lane during the cycle, however, does not perform lane changing operation. The equivalent arrival time takes into account the additional (or less) travel time incurred by performing lane changing within the road segment and enforce strict FIFO for each lane. A car-following model based simulation scheme can then be applied to estimate the queuing process. The next few sub-sections present the methodology and algorithms for the equivalent cumulative arrival–departure curve reconstruction and queue length estimation.

2.1. Equivalent cumulative arrival–departure curve reconstruction

The key component to reconstruct the equivalent cumulative arrival curve is to infer the information for unmatched arrival vehicles. It can be viewed as solving an interpolation problem, which is defined as follows: let $I_k = \{i_1, i_2, \ldots, i_{N_k}\}$ be the cumulative indices for arrivals which correspond to each vehicle in departure curve for signal cycle k, N_k be the total number of arriving vehicles within cycle k, and denote $H_k = \{t_1, t_2, \ldots, t_{N_k}\}$ as the arrival times for arrival vehicles. As we only

observe the arrival times for matched vehicles, we need to infer the equivalent arrival times for the unmatched arrival vehicles. This is equivalent to finding the most likely interpolation of arrival curve given the fixed points specified by the matched vehicles. In this study, we develop a novel interpolation model based on Gaussian process to solve the equivalent cumulative arrival curve reconstruction problem. Gaussian process is a powerful method in Bayesian statistical modeling and machine learning, which models finite linear combination of samples as a joint Gaussian distribution. The Gaussian process defines a prior probability distribution over function directly and works with a distribution over the uncountable infinite space of functions (Roberts et al., 2013). This feature makes Gaussian process an ideal tool for the equivalent arrival curve reconstruction, since the problem is to find the most likely function (arrival curve), rather than predicting on specific values. Our discussion of the Gaussian process in this paper is restricted to the construction of the accurate equivalent arrival–departure curves; more detailed background information can be found in Bishop (2006) and Roberts et al. (2013).

2.1.1. Gaussian process interpolation model

We begin the discussion by first modeling the mean arrival process of vehicles from upstream signalized intersection. In this study, the exact signal timing plans for both the upstream and downstream intersections are known. Consider the signal timing plan with four phases illustrated in Fig. 1. The arrival vehicles that enter the investigated link can be considered as a combination of three arrival processes: (1) arrival vehicles from the through movement during the first phase; (2) arrival vehicles from the right turning movement during the entire cycle[1]; (3) arrival vehicles from the left turning movement of the lanes in the orthogonal approach during the last phase. In this study, the arrival rate for each state of the process is assumed to have constant mean arrival rate and its variation is controlled by a disturbance term η. More specifically, for the first and third arrival process, we introduce two arrival rates: the saturation discharging flow rate $r_{T_s}(r_{L_s})$ and normal arrival flow rate $r_{T_n}(r_{L_n})$ for vehicles from the through movement (left turning movement) of the upstream intersection. The saturation discharging flow corresponds to vehicles that discharge from the queue as the signal turns green, which is assumed to only happen within time period $[0, t_a]$ and $[T_3, t_b]$ for the first and fourth phase. The normal arrival flow corresponds to the vehicles that arrive at the intersection after the queue has already fully dissipated and do not experience queuing process. The saturation discharging flow rates are usually much larger than the normal arrival flow rates ($r_{T_s} > r_{T_n}, r_{L_s} > r_{L_n}$), however, their durations t_a and $t_b - T_3$ are unknown and need to be inferred. For the arrival flow from the right turning movement, since it is unrestricted, we model its mean arrival rate as r_R, which can occur during the entire signal cycle ($[0, T_{cyc}]$, where T_{cyc} is the cycle length). It should be noted that it is allowed for t_a to take values of 0 or T_1, and t_b to take values of T_3 or T_{cyc}. Such cases correspond to extreme scenarios of either no queue ($t_a = 0, t_b = T_3$) or over-saturation state ($t_a = T_1, t_b = T_{cyc}$) that the queue arrivals in a cycle cannot be fully discharged during this cycle.

Denote $\mu(t)$ as the mean cumulative arrival process during a cycle. Based on previous assumptions, the combined mean arrival process can be characterized by a piecewise linear function defined as follows:

$$\mu(t) = \begin{cases} r_{T_s} t + r_R t, & t \in [0, t_a] \\ \mu(t_a) + r_{T_n}(t - t_a) + r_R(t - t_a), & t \in (t_a, T_1] \\ \mu(T_1) + r_R(t - T_1), & t \in (T_1, T_3] \\ \mu(T_3) + r_{L_s}(t - T_3) + r_R(t - T_3), & t \in (T_3, t_b] \\ \mu(t_b) + r_{L_n}(t - t_b) + r_R(t - t_b), & t \in (t_b, T_{cyc}] \end{cases} \tag{1}$$

The cumulative arrival curve can hence be modeled as follows:

$$y(x|\boldsymbol{\theta}) = \mu(x|\boldsymbol{\theta}) + \eta \tag{2}$$

where y is the index of the cumulative arrival vehicle, x is the arrival timestamp, η is the disturbance term that is assumed to be normally distributed, and the parameter $\boldsymbol{\theta} = (r_{T_s}, r_{T_n}, r_R, r_{L_s}, r_{L_n}, t_a, t_b)^T$ is used to specify the mean cumulative arrival process $\mu(x|\boldsymbol{\theta})$. From flow conservation, an arrival vehicle at the upstream intersection must be matched to a departing vehicle with the same index at the downstream intersection. However, such matching information can only be established for matched vehicles, thus the arrival curve is incomplete. The vehicle index and arrival timestamp pair for unmatched vehicle must be inferred to reconstruct the equivalent cumulative arrival curve.

Suppose $(y_1, x_1), (y_2, x_2), \ldots, \left(y_{n_k}, x_{n_k}\right)$ are matched arrival vehicles on the arrival curve. The objective of equivalent cumulative arrival curve reconstruction is to find the most likely arrival curve and further infer the arrival timestamp for unmatched arrivals. This can be viewed as solving an interpolation problem given observed data and specific arrival pattern. To achieve this goal, we model the entire arrival process as a Gaussian process.

Let $\boldsymbol{y} = \left(y_1, y_2, \ldots, y_{n_k}\right)^T, \boldsymbol{x} = (x_1, x_2, \ldots, x_{n_k})^T$. We adopt the most widely used *squared exponential* function with regulation term as the kernel function to measure the covariance between any pair of points on the arrival curve:

[1] This study is based on a field experiment conducted in China, where the right turning movement on red is permitted. For right turning movement restricted cases, the arrival process 2 can be modified based on the actual timing plan.

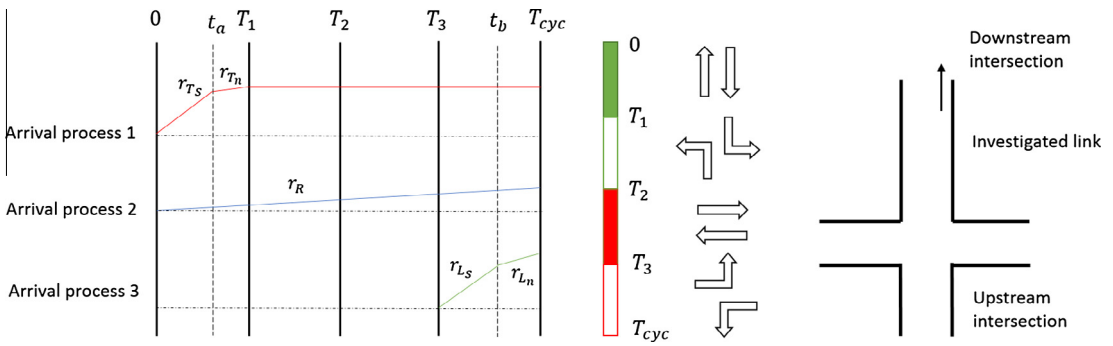

Fig. 1. Illustration of mean arrival process at the upstream intersection.

$$k(x_n, x_m) = h_0 \exp\left(-\left(\frac{x_n - x_m}{\lambda}\right)^2\right) + \eta^2 \delta(x_n, x_m)$$
$$\delta(x_n, x_m) = \begin{cases} 1, & \text{if } x_n = x_m \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where h_0 and λ are the output-scale and input-scale hyperparameters. The function $y(x)$ can thus be viewed as the arrival indices $\boldsymbol{y}$'s drawn at time $\boldsymbol{x}$ from a multivariate Gaussian distribution controlled by parameter $\boldsymbol{\theta}$,

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = N(\boldsymbol{\mu}(x|\boldsymbol{\theta}), \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})) \tag{4}$$

which is a Gaussian process, with the covariance matrix defined as

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}) = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix} \tag{5}$$

Let $\boldsymbol{y}^*, \boldsymbol{x}^*$ be the unmatched arrival indices and timestamps on the arrival curve, the joint distribution of all arrivals can be then represented as the following augmented form:

$$p\left(\begin{bmatrix} \boldsymbol{y}|\boldsymbol{\theta} \\ \boldsymbol{y}^*|\boldsymbol{\theta} \end{bmatrix}\right) = N\left(\begin{bmatrix} \mu(\boldsymbol{x}|\boldsymbol{\theta}) \\ \mu(\boldsymbol{x}^*|\boldsymbol{\theta}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}) & \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}^*) \\ \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}) & \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix}\right) \tag{6}$$

hence the marginal distribution over $\boldsymbol{y}^*$ is still a Gaussian given by

$$p(\boldsymbol{y}^*|\boldsymbol{\theta}) = N(\boldsymbol{m}^*(\boldsymbol{x}^*|\boldsymbol{\theta}), \boldsymbol{C}^*) \tag{7}$$

and the posterior mean and covariance $\boldsymbol{m}^*(\boldsymbol{x}^*|\boldsymbol{\theta})$ and $\boldsymbol{C}^*$ can be computed in closed form as follows:

$$\boldsymbol{m}^*(\boldsymbol{x}^*|\boldsymbol{\theta}) = \boldsymbol{\mu}(\boldsymbol{x}^*|\boldsymbol{\theta}) + \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x})\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})^{-1}(\boldsymbol{y}(\boldsymbol{x}) - \boldsymbol{\mu}(\boldsymbol{x}))$$
$$\boldsymbol{C}^* = \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*) + \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x})\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x})^{-1}\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}^*) \tag{8}$$

2.1.2. Parameter estimation via MCMC

To obtain the most likely arrival curve given the information from the matched arrival vehicles, $\boldsymbol{\theta}$ needs to be inferred. The direct maximization of the likelihood over $\boldsymbol{\theta}$ defined in Eq. (4) is analytically intractable given the complexity introduced by the piecewise linear mean cumulative arrival process. In this study, we develop a Metropolis–Hasting (M–H) algorithm to learn the parameter $\boldsymbol{\theta}$ from the observed matched records $(\boldsymbol{x}, \boldsymbol{y})$. M–H algorithm is a widely used sampling algorithm of Markov Chain Monte Carlo (MCMC) technique, which can be used to estimate parameter values from complex posterior distributions by generating a sequence of random samples. For more information about M–H algorithm and MCMC technique, please refer to Berger (1985) and Bishop (2006).

To allow for faster sampling, we introduce the following non-informative prior distribution based on modified uniform distribution for the flow rate parameters in $\boldsymbol{\theta}$:

$$\theta \sim U_\rho[0, u] \tag{9}$$

in which the probability density distribution is defined on a close interval $[0, u]$, and the probability density function is given as:

$$P(\theta) = \begin{cases} \rho, & \theta = 0 \\ (1 - \rho) \cdot \frac{1}{u}, & 0 < \theta \leqslant u \end{cases} \tag{10}$$

where ρ $(0 < \rho < 1)$ is a predefined hyperparameter. The reason to use this prior is that the distribution of the flow rates, especially the normal arrival flow rates r_{T_n}, r_{L_n} and r_R may be more likely to skew towards 0, especially in signal cycles that

have very few arrival vehicles. Using this prior distribution instead of a standard uniform distribution will ensure more effective sampling of the flow rate parameters in MCMC iterations. The brief outline of the developed M–H algorithm for learning the parameter $\boldsymbol{\theta}$ is given in Algorithm 1.

Algorithm 1. M–H algorithm for learning parameter θ.

Step 1: Initialize $\boldsymbol{\theta}^{(0)}$.

Step 2: Sample $\hat{\boldsymbol{\theta}}$ from following distributions:

$$t_a \sim U[0, T_1], \quad t_b \sim U[T_3, T_{cyc}]$$

$$r_{T_s} \sim U_\rho\left[0, \frac{N_k}{t_a}\right], \quad r_{T_n} \sim U_\rho[0, r_{T_s}]$$

$$r_R \sim U_\rho\left[0, \frac{N_k}{T_{cyc}}\right], \quad r_{L_s} \sim U_\rho\left[0, \frac{N_k}{t_c - T_3}\right], \quad r_{L_n} \sim U_\rho[0, r_{L_s}]$$

where $U[a, b]$ stand for uniform distribution defined on interval $[a, b]$, N_k is the total number of vehicles in the signal cycle found from departure curve. Note from above sampling method, $r_{T_s} \geqslant r_{T_n}, r_{L_s} \geqslant r_{L_n}$ is always guaranteed.

Step 3: Compute and accept $\hat{\boldsymbol{\theta}}$ as $\boldsymbol{\theta}^{(k)}$ according to following ratio

$$r = \min\left\{\frac{p\left(\boldsymbol{y}|\hat{\boldsymbol{\theta}}\right)}{p\left(\boldsymbol{y}|\boldsymbol{\theta}^{(k-1)}\right)}, 1\right\}$$

Step 4: Repeat steps 2–3 until $\boldsymbol{\theta}^{(k)}$ are stable.

Step 5: Discard the first 20–50% of accepted samples as burn-in, and use the mean of the remaining sampled $\boldsymbol{\theta}$ values as the estimated parameter for $\boldsymbol{\theta}$.

After learning the parameter $\boldsymbol{\theta}$, the posterior mean and variance of the arrival process for unmatched arrival vehicles $\boldsymbol{m}^*, \boldsymbol{C}^*$ can be easily computed using Eq. (8). The final interpolation of the arrival curve can be obtained based on the posterior mean estimates $\boldsymbol{m}^*$. Denote T_i^{arr}, T_i^{dep} as the timestamps when vehicle i passes the upstream and downstream intersections. The overall algorithm for equivalent cumulative arrival–departure curve reconstruction is summarized in Algorithm 2.

Algorithm 2. Algorithm for equivalent cumulative arrival–departure curve reconstruction.

For each direction and lane:

Step 1: Construct departure curve

Directly draw departure curve using vehicle departure data. Create vehicle index $(i = 1, 2, \ldots, N_k)$ based on the vehicle departure timestamp sequence. Put the timestamps of all matched arrival vehicles in the ordered set x, and let y be the corresponding ordered index vector. Assign timestamps of all unmatched arrival vehicles as *null*, and let $\boldsymbol{y}^*$ be their corresponding ordered index vector.

Step 2: Remove FIFO violations

Let the sequence $\{m_1, m_2, \ldots, m_{|y|}\}$ be the matched vehicle indices in y. For every matched arrival vehicle m_j, if $T_{m_j}^{arr} > T_{m_{j+1}}^{arr}$, then filter out the m_j in $\boldsymbol{x}, \boldsymbol{y}$ and add the index m_j to $\boldsymbol{y}^*$.

Step 3: Identify cycle starting vehicle

Insert signal cycle starting times $\{T_k^{cyc}, k = 1, \ldots, K\}$, where K is the number of signal cycles. For all matched arrival vehicles in cycle $[T_k^{cyc}, T_{k+1}^{cyc}]$, Do:

1. If any matched vehicle m_j satisfies $T_{m_j}^{arr} > T_k^{cyc}$, but $T_{m_j-1}^{arr} < T_k^{cyc}, m_j - 1 \in \boldsymbol{y}$, then identify vehicle m_j as the cycle starting vehicle.
2. Define a small time window TN_{cyc}, check if any matched arrival vehicles satisfy $0 < T_{m_j}^{arr} - T_k^{cyc} < TN_{cyc}$. The earliest vehicle satisfies this condition is identified as the cycle starting vehicle.
3. If no matched vehicle is found in $[T_k^{cyc}, T_k^{cyc} + TN_{cyc}]$, take out all matched arrival data from this cycle and the neighboring cycles (use more cycles if insufficient data available). Perform piecewise cubic Hermite interpolation and use the earliest interpolated vehicle in the cycle as the starting vehicle.

Step 4: Infer unmatched arrival vehicle

Perform the proposed Gaussian process interpolation model on all identified and matched vehicle records, obtain the posterior mean $\boldsymbol{m}^*$ predicted by the Gaussian process interpolation model.

Step 5: Filter out FIFO violations in $\boldsymbol{m}^*$ and find the corresponding arrival times for each of the unmatched arrivals.

In Algorithm 2, Step 1 pre-processes the data that directly constructs the departure curve and create the index for all vehicles (both matched and unmatched) based on the accurate vehicle departure data. The FIFO violations in the arrival vehicle records caused by lane changing behaviors are then filtered out. These filtered arrival vehicles will be marked as unmatched, and inferred together with other unmatched arrival vehicles. To make sure the kernel function (Eq. (3)) in the Gaussian process interpolation model is well defined, there must exist at least two matched vehicle records for each signal cycle. This is guaranteed in the Step 3 by identify the cycle starting vehicle of each signal cycle, hence at least two matched data points (starting vehicle of the current cycle and the next cycle) exist, and the Gaussian process model is always well-defined and solvable. Identifying the cycle starting vehicles proceeds by sequentially checking 3 conditions. First, if we can directly observe both the last vehicle $i-1$ in the previous cycle and the first vehicle i in the current cycle ($T_i^{arr} > T_k^{cyc}$ and $T_{i-1}^{arr} < T_k^{cyc}$), then the cycle starting vehicle can be directly identified. If not, we inspect if any matched arrivals' timestamps fall inside a small time window $\left[T_k^{cyc}, T_k^{cyc} + TN_{cyc}\right]$ (in the actual implementation, $TN_{cyc} = 5s$ is used), the earliest vehicle satisfies this condition is identified as cycle starting vehicle. Lastly, if the data is too sparse, we use a numerical interpolation method (piecewise cubic Hermite interpolation) with all the matched arrival data in the current cycle as well as the neighboring cycles to create a piece of initial arrival curve interpolation. The earliest interpolation point in cycle k is used as the cycle starting vehicle. The piecewise cubic Hermite interpolation is implemented using the build-in algorithm "pchip" provided in MATLAB. For details about this algorithm, please refer to Fritsch and Carlson (1980), Kahaner et al. (1988). Multiple numerical interpolation algorithms have been tested when developing the proposed model. The piecewise cubic Hermite interpolation is selected as it produces the most realistic initial arrival curve when there are significant amount of unmatched arrival vehicles. Since the piecewise cubic Hermite interpolation is only a numerical interpolation method without considering the underlying traffic dynamics, we only use it to infer one point (cycle starting vehicle) if necessary, and leave the majority of the work to the proposed Gaussian process interpolation model. Step 4 is the most important step in Algorithm 2, which directly implements the Algorithm 1 and further obtains the posterior mean $\boldsymbol{m}^*$ of the unmatched arrival timestamps predicted by the Gaussian process interpolation model. In the last step, by filtering and modifying the FIFO violations in $\boldsymbol{m}^*$, we obtain the most likely equivalent cumulative arrival curve.

2.2. Estimation of link travel time and number of vehicles on the lane

Several traffic condition measures for each lane can be directly estimated from the reconstructed equivalent cumulative arrival–departure curve, such as the real-time link travel time, number of vehicles on the lane, and the average link travel time of a given time period, which is illustrated in Fig. 2. For example, the link travel time and the number of vehicles on the lane can be obtained from the length of the horizontal and vertical line between the cumulative arrival and departure curve respectively. And the average link travel time of the time period $[t_1, t_2]$ can be obtained by the computed shaded area divided by the difference of cumulative number of vehicles ΔN.

2.3. Queue length estimation via boundary constrained car-following model

Although LPR data provide information on when a vehicle passes the upstream and downstream intersections of a road, the actual trajectory of the vehicle within the road segment is unknown. To estimate the queue length given the boundary conditions specified in LPR data, we introduce a boundary constrained car-following based simulation scheme. The challenge in developing a suitable car-following model is to identify one that satisfies the constraints at the boundaries of the trajectories and one that is not overly complicated so that quick computations can be performed. Although many car-following models have been proposed in the literature, the difficulty of applying such models for the queue length estimation problem in this paper arises from several aspects: (1) most of the car-following models require at least some of the vehicle trajectories be known to allow for car-following behavior, which are nonetheless not known in this problem; (2) although a more comprehensive but complex car-following model (e.g. higher order car-following models that simultaneously consider the vehicle's location, speed, and acceleration) will capture the traffic behaviors better, given the limited vehicle state information from LPR data, significant amount of additional assumptions on driver behavior and the boundary control conditions need to be introduced to ensure the satisfaction of the boundary constraints. This will add great difficulty in modeling, calibration and computation of the resulting model. In this study, we propose an elegant *boundary constrained car-following model* that is simple, uses minimal additional assumptions, and perfectly satisfies the boundary conditions specified in the LPR data.

The *boundary constrained car-following model* in this study is modified from a first-order Optimal Velocity (OV) car-following model developed by Tordeux et al. (2015). Tordeux's car-following model can be viewed as a first-order simplification of the classic OV car-following model (Newell, 1961; Bando et al., 1995) that is based only on an optimal speed function and the driver's reaction time. It offers several particularly desirable features for the queue length estimation problem in this study: (1) the speed function in the Tordeux's car-following model only depends on the desired travel speed and gap distance between the current vehicle and its predecessor, which allows great flexibility in incorporating the boundary conditions specified in the LPR data; (2) the number of parameters are minimal, which relieves the difficulty in calibration of the car-following model with the missing trajectory information; (3) the model guarantees collision-free condition and is numerically stable under stop-and-go traffic conditions, which are crucial in modeling the queuing process; (4) the

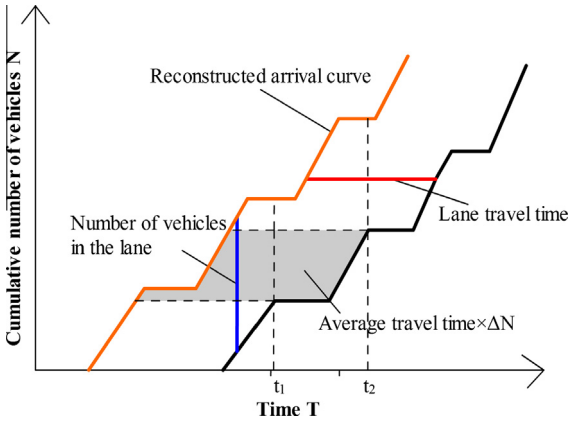

Fig. 2. Illustration of real-time estimation of link travel time and number of vehicles in the lane.

computation cost of Tordeux's car-following model is inexpensive as compared to second-order models and the vehicles can be updated in parallel, which allows for efficient real-time estimation of queue lengths.

The speed function of Tordeux's car-following model has the following bounded linear form:

$$V(d) = \min\{v_d, \max\{0, (d-L)/G\}\} \tag{11}$$

where v_d is the desired travel speed, G is the time gap used, and L is the safe driving distance occupied by a vehicle. Tordeux's car-following model is first-order, the speed of a vehicle is given by

$$\dot{s}_n(t+\tau) = V(\Delta s_n(t)) \tag{12}$$

in which $s_n(t)$ is the position of the head of the vehicle n, τ denotes the reaction time for each vehicle and $\Delta s_n(t) = s_{n-1}(t) - s_n(t)$ is the gap distance between the vehicle n and its predecessor vehicle $n-1$ at time t. An explicit Euler scheme (Eq. (13)) is adopted in Tordeux's car-following model to update the location of each of vehicle:

$$s_n(t+\delta t) = s_n(t) + \delta t V(\Delta s_n(t) - \tau[V(\Delta s_{n-1}(t) - V(\Delta s_n(t)))]) \tag{13}$$

with δt denotes the updating time step size. In actual implementation, δt is taken as 0.5 s. From the above formulation, the current vehicle's location at next time step can be efficiently obtained by only evaluating the gap distance between the current vehicle and its predecessor.

2.3.1. Incorporating boundary constraints

The boundary conditions for each vehicle i include the exact arrival and departure time T_i^{arr}, T_i^{dep}at the upstream and downstream intersection. In particular, to make sure Tordeux's car-following model is correctly specified, we introduce a positive arrival speed v^{ini} $(0 < v^{ini} \leqslant v_d)$ for all vehicles entering the road (ensures $s_{n-1}(t) > s_n(t)$). Numerical results show that the choice of v^{ini} has minor impact on the results. This is because that v^{ini} only impact the initial behavior of a vehicle when it enters the lane. The extra or less time spent on the acceleration or deceleration at the initial stage is negligible compared with the total link travel time if the link is long enough. The boundary condition for each vehicle can be summarized as:

- Initial condition $(t = T_n^{arr})$:

$$s_n(t) = 0, \dot{s}(t) = v^{ini} \tag{14}$$

- Final condition $(t = T_n^{dep})$:

$$s_n(t) = L_R \tag{15}$$

where L_R is the length of the link.

To incorporate the boundary constraints, two stages are introduced to Torduex's car-following model, namely: (1) car-following stage, in which vehicle n has a predecessor and follows it according to Tordeux's car-following model (see Fig. 3(a)); (2) pseudo car-following stage, in which the predecessor vehicle has already discharged from downstream intersection and vehicle n follows a pseudo predecessor that remains stationary at location $L_R + L$ until $T_n^{dep} - \delta t$ (see Fig. 3(b)). Since the speed function and distance update of Tordeux's car-following model only depends on the gap distance $\Delta s_n(t)$, by introducing the pseudo predecessor, the car-following scheme can be preserved even when the actual predecessor vehicle

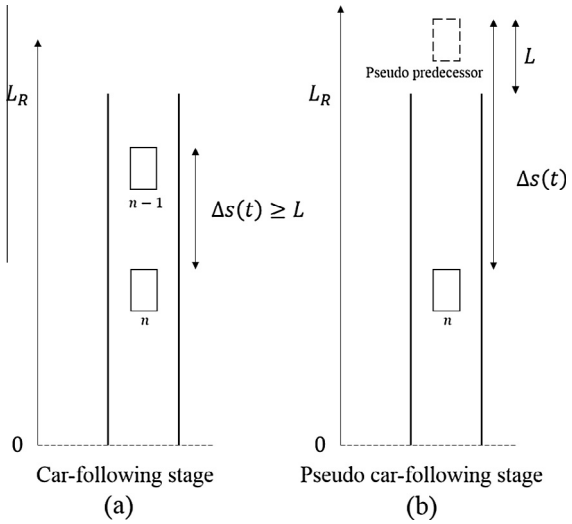

Fig. 3. Illustration of the car-following stages.

has already discharged. Moreover, since the speed function Eq. (11) has a linearly bounded form, the rapid change in gap distance (e.g. original predecessor vehicle discharged, and a pseudo predecessor appear immediately at distance $L_R + L$ at next time step) will not significantly impact the driving behavior of the current vehicle. Due to the appearance of the pseudo vehicle, vehicle n will stop at stopping line ($s = L_R$) before T_n^{dep}, and discharge at T_n^{dep} time step immediately. The successor vehicles will remain in car-following stage and join the queue as vehicle n stops at the stopping line of the downstream intersection until vehicle n discharges. Thus the queuing process can be established under the boundary constraints realistically.

The overall outline of the queue length estimation procedure using the proposed boundary constrained car-following model is presented as follows:

Algorithm 3. Outline of the queue length estimation procedure.

Step 1: Create dummy initial vehicle to enable car-following
Creating a dummy initial vehicle starting at time $T_{min}^{arr} - \Delta t$ with travel speed equal to the desired travel speed v_d, where T_{min}^{arr} is the minimum arrival time in the data and Δt is a threshold time gap for the dummy initial vehicle.

Step 2: Car-following module
Implement the boundary constrained car-following model in Section 2.3.

Step 3: Queue length estimation from the trajectory
Obtaining vehicle trajectory from step 2, obtaining the number of vehicles in the queuing state of each time step as the queue length. A vehicle is identified as in queuing state if:

$$s(t-1) = s(t) \text{ and } \dot{s}(t) = 0$$

3. Field experiment design

The lane-based queue length estimation model proposed in this paper was tested using real world LPR data and validated using ground truth cycle maximum queue length data collected from two field experiments. Both the LPR data and the ground truth data were from the northbound of a 720 meter segment of Heping Road in the city of Langfang, Hebei province, China. The field experiment design is illustrated in Fig. 4. A HD video camera was placed at the downstream intersection to record the video for actual queuing during the experiment. The recorded video data were manually processed, from which the cycle-by-cycle maximum queue lengths for each lane were obtained as the ground truth data. The first field experiment was conducted on July 9th, 2014 from 7:30am to 9:00am, during which the LPR data were obtained. However, the HD video camera was not initially correctly positioned resulting in the inaccurate measurement of the end of the queue. The issue was found and resolved at about 8:00am, thus only the ground truth data from 8:00am to 9:00am were useable to perform the validation of the proposed model. Another more comprehensive field experiment was conducted on November 26th, 2014 from 7:30am to 12:00pm, during which both LPR data and ground truth queue length data were obtained.

Fig. 4. Field experiment design.

Both the upstream intersection (at Beifeng Street) and the downstream intersection (at Xiangyun Street) were equipped with four LPR cameras for each approach, and the LPR data were obtained from the local transportation agency. The LPR data contain the following information for each passing vehicle: (1) an identified license plate number, (2) the actual timestamp when a vehicle passes the stop line, (3) the travel orientation, and (4) the lane from which the vehicle exits. The upstream and downstream intersections of the study share the same lane configurations, which from the leftmost to rightmost lane are: dedicated left turning lane (Lane 1), through lane (Lane 2) and through and right turning lane (Lane 3). Both the upstream and downstream intersection adopted fixed timing plans with the same cycle length of 120s (7:00am – 9:00am) and 100s (9:00am – 12:00pm). However, different phase intervals were used for the upstream intersection and the downstream intersection, which were also recorded in the field experiment.

The LPR system investigated in this work does not possess a very high accuracy for license plate recognition, and is impacted by factors such as resolution of LPR cameras and lighting conditions, etc. Fig. 5 presents the proportion of unrecognized vehicles for the upstream and downstream intersections, and the proportion of unmatched vehicles for every 10 min interval from 7:30am – 12:00pm for the November 26th field experiment data. The proportions of the unrecognized vehicles do not include the vehicles whose license plates were recognized wrongly, but only the vehicles with license plate numbers

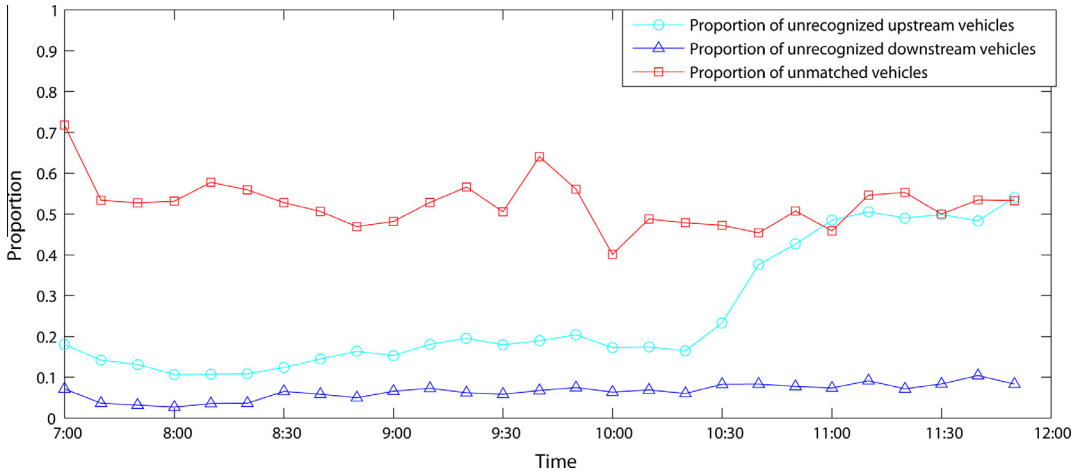

Fig. 5. Proportion of unrecognized vehicles for upstream and downstream intersections and unmatched vehicles for 10 min intervals (November 26th data).

unable to be recognized at all and recorded as "unrecognized". From the LPR data, it is found that the LPR cameras at the upstream intersection have a poorer performance compared with the LPR cameras at the downstream intersection. For unknown reasons, the records of vehicles traveling from west to east direction for upstream intersection after 10:30am were missing in the November 26th LPR data, which caused all vehicles that merging to the studying road segment through the left turning movement became unobserved. This resulted in significant increase in the unrecognized proportion of upstream arrival vehicles. Due to the existence of the unrecognized and the erroneously recognized LPR records, the overall unmatched vehicles account for 40–70% of total data records, thus inferring the missing information from unmatched arrival vehicles becomes critical in the queue length estimation problem.

4. Numerical results

A series of tests were first conducted to test the convergence of the proposed Gaussian process interpolation model using a subset of LPR data. Multiple chains with different initial parameter values were tested and the chain histories were inspected to verify the convergence. The test results showed that around 7000 iterations were sufficient to obtain convergent results regardless of the different initial parameter values used. To be rigorous, we used 10,000 iterations when performing the final numerical experiment. Furthermore, to be conservative and to fully remove the impact from the initial values of the parameters, we discard first 50% of the accepted samples as burn-in and used the remaining samples to compute the final parameter values.

Reconstructing the equivalent cumulative arrival–departure curve takes the greatest amount of computation time in the entire estimation process due to the involvement of parameter sampling in the M–H algorithm. However, since all parameter values are sampled from simple distributions, and the multivariate normal distributed likelihood function $p(\boldsymbol{y}|\boldsymbol{\theta})$ can be easily evaluated, thus sampling 10,000 iterations for one signal cycle using the M–H algorithm only takes about 1 s on a 2.4 GHz CPU laptop. Also, as a light-weighted first-order car-following model is used, the entire queue length estimation can be implemented efficiently in real-time.

Fig. 6 presents a sample section of the reconstructed cumulative arrival–departure curve for the tested through lane (Lane 2) using the LPR data. The blue curve and dots represent the departure curve and departure vehicles. The red curve and dots represent the reconstructed arrival curve and arrival vehicles. The green dashed line represents the estimated posterior mean of the cumulative arrivals curve. Reasonable representation of the equivalent arrival process can be obtained using the proposed Gaussian process interpolation model. Another advantage of adopting the Gaussian process interpolation model is that the equivalent cumulative arrival curve can be modeled as a distribution. In Fig. 6, the black dotted lines encapsulate the area with plus/minus the standard deviation of the estimated posterior mean, which is $\boldsymbol{m}^*(t) \pm \sqrt{\boldsymbol{C}^*(t,t)}, \forall t$. This also provides a measure of confidence for the reconstructed equivalent cumulative arrival curve.

4.1. Calibration of the car-following model

The proposed boundary constrained car-following model needs to be calibrated before used in queue length estimation. Due to the absence of actual vehicle trajectories, in this study, the car-following model was calibrated using parts of the actual cycle maximum queue length data against the model estimated cycle maximum queue lengths. The ground truth

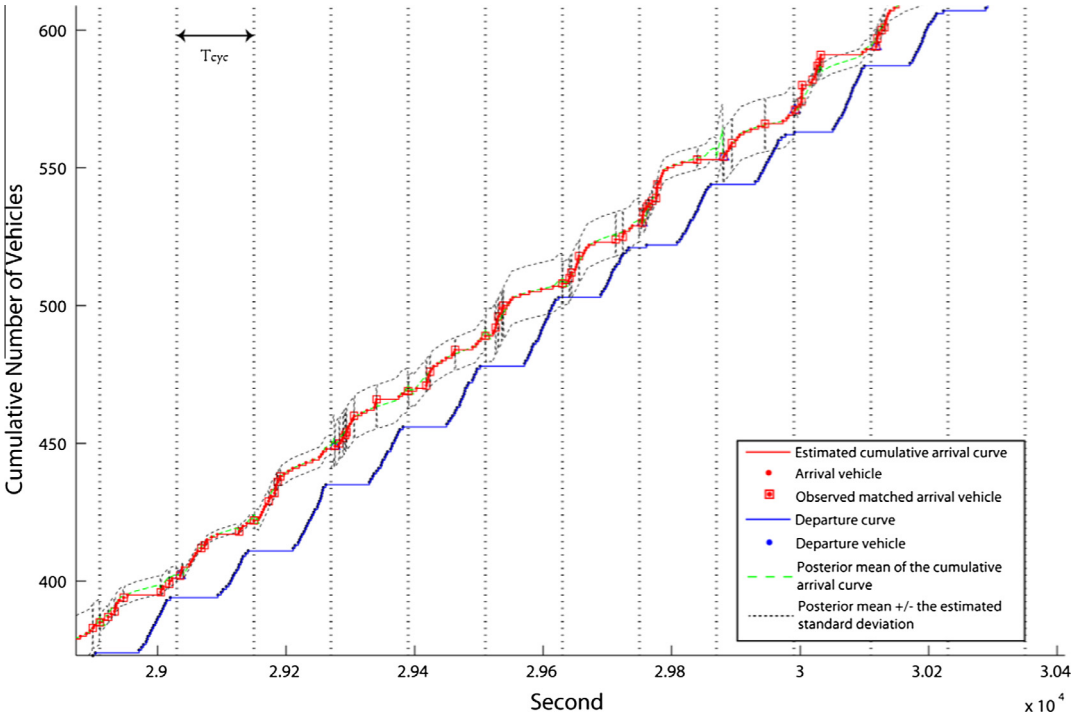

Fig. 6. Sample result for reconstructed equivalent cumulative arrival–departure curve.

cycle maximum queue lengths from 8:00 – 8:20am of the July 9th data and 7:30 – 9:00am of the November 26th data were used to calibrate the car-following model. Define the following loss function:

$$Q(\boldsymbol{\Phi}) = \sum_{i=1}^{NC_c} \left[q_e^i(\boldsymbol{\Phi}) - q_a^i\right]^2 \tag{16}$$

where NC_c is the number of cycles used for calibration, $q_e^i(\boldsymbol{\phi})$ is the maximum queue length estimates given the car-following parameters $\boldsymbol{\phi}\left(\boldsymbol{\phi} = (L, v_d, v^{ini})^T\right)$ and q_a^i is the actual maximum queue length for cycle i obtained from the field experiment. The car-following parameter $\boldsymbol{\phi}$ can thus be calibrated by solving the simulation optimization problem $\boldsymbol{\phi}^* = argmin_{\boldsymbol{\phi}} Q(\boldsymbol{\phi})$. Various simulation optimization techniques can be used for this type of problems, such as stochastic optimization (SO), response surface methodology (RSM), genetic algorithm (GA) and simulated annealing (SA) (Carson and Maria, 1997). In this study, both GA and SA were tested to optimize the loss function defined in Eq. (16). The calibrated car-following parameter $\boldsymbol{\phi}$ found by SA was eventually used to perform the queue length estimation, as lower loss function value was achieved.

4.2. Queue length estimation results

The sample queue length estimation results obtained using the November 26th data from 8:00am to 10:00am are presented in Fig. 7. The results show that the though movement lane (Lane 2) was the most congested lane with the most severe queuing among the three tested lanes during the observation period, while Lane 1, the dedicated left turning lane was the least congested. For all the three lanes, the queuing was much more severe before 8:30am due to the heavy morning peak traffic. The overall congestion patterns for the three lanes in Fig. 7 agree with the congestion pattern found in the ground truth data. To further examine the quality of the estimation results, we insert the cycle starting times obtained from the field experiments and find the maximum queue lengths and their corresponding occurrence times within each signal cycle, which are indicated using the triangle markers in Fig. 7. The obtained estimated cycle maximum queue lengths are then compared with the actual cycle maximum queue lengths from the field experiments for quantitative validation. The validation results for the July 9th and November 26th data are presented in Figs. 8 and 9 respectively.

It can be observed from Figs. 8 and 9 that the estimated cycle maximum queue length curves captures the pattern of the actual cycle maximum queue length curves quite well. However, some discrepancies are also observed. Two measures, namely the root-mean-square error (RMSE) and mean absolute error (MAE) are used to quantitatively evaluate the accuracy of the proposed model. Let NC be the total number of signal cycles that the actual cycle maximum queue length data are available, the RMSE and MAE are computed as:

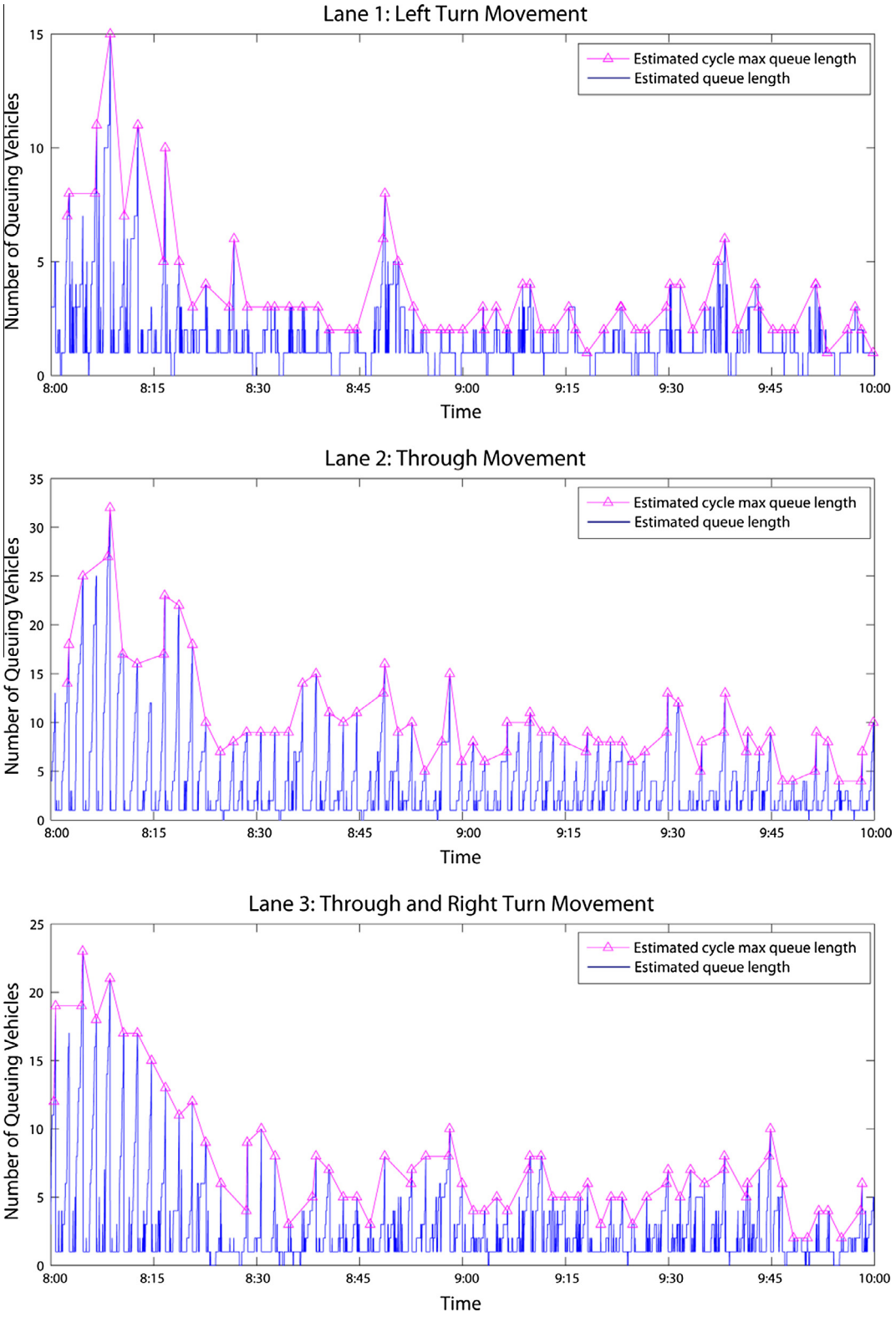

Fig. 7. Sample result for estimated queue length (8:00–10:00am, November 26th).

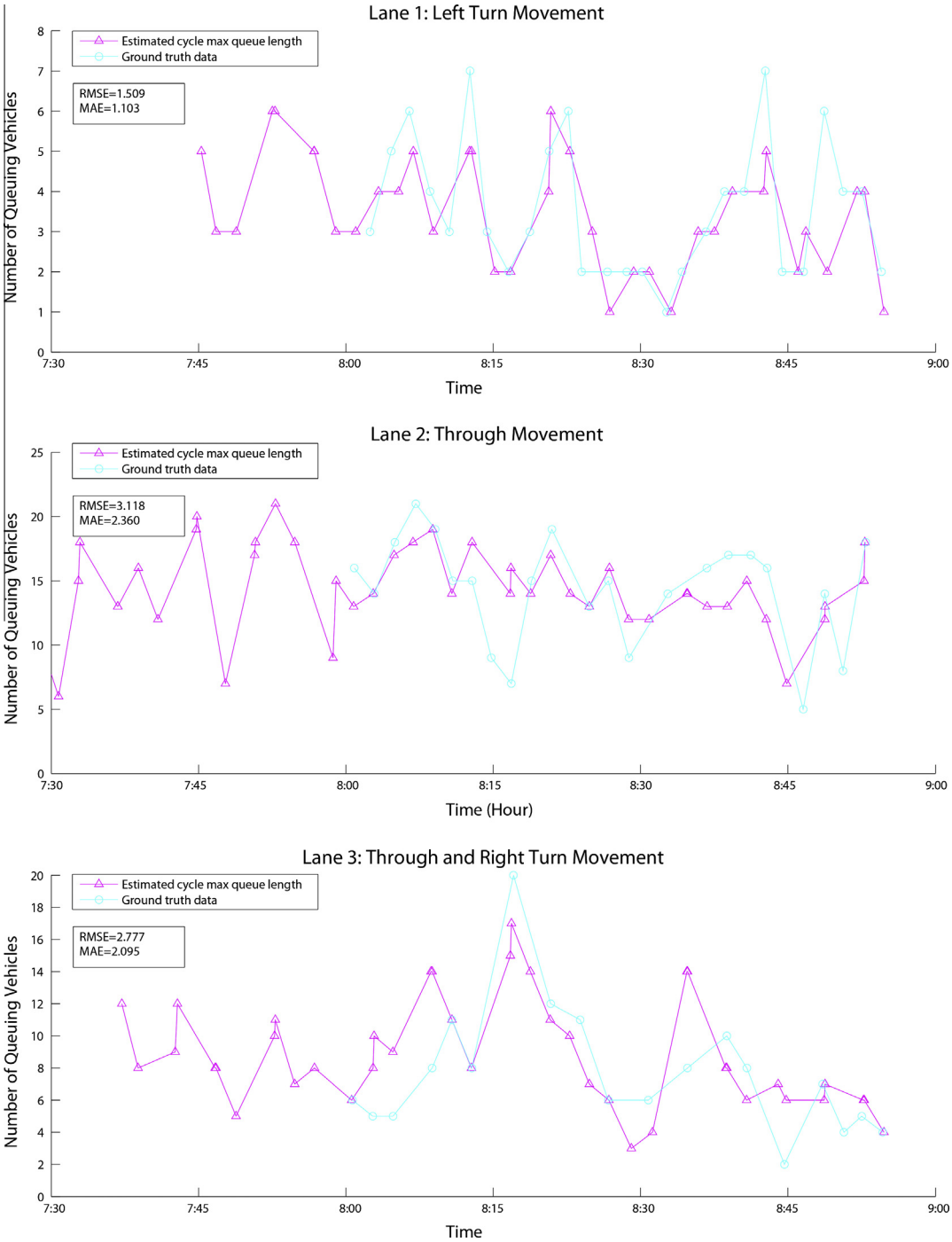

Fig. 8. Cycle maximum queue length comparison (July 9th).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{NC} \left(q_e^i - q_a^i\right)^2}{NC}} \tag{17}$$

$$MAE = \frac{\sum_{i=1}^{NC} \left|q_e^i - q_a^i\right|}{NC} \tag{18}$$

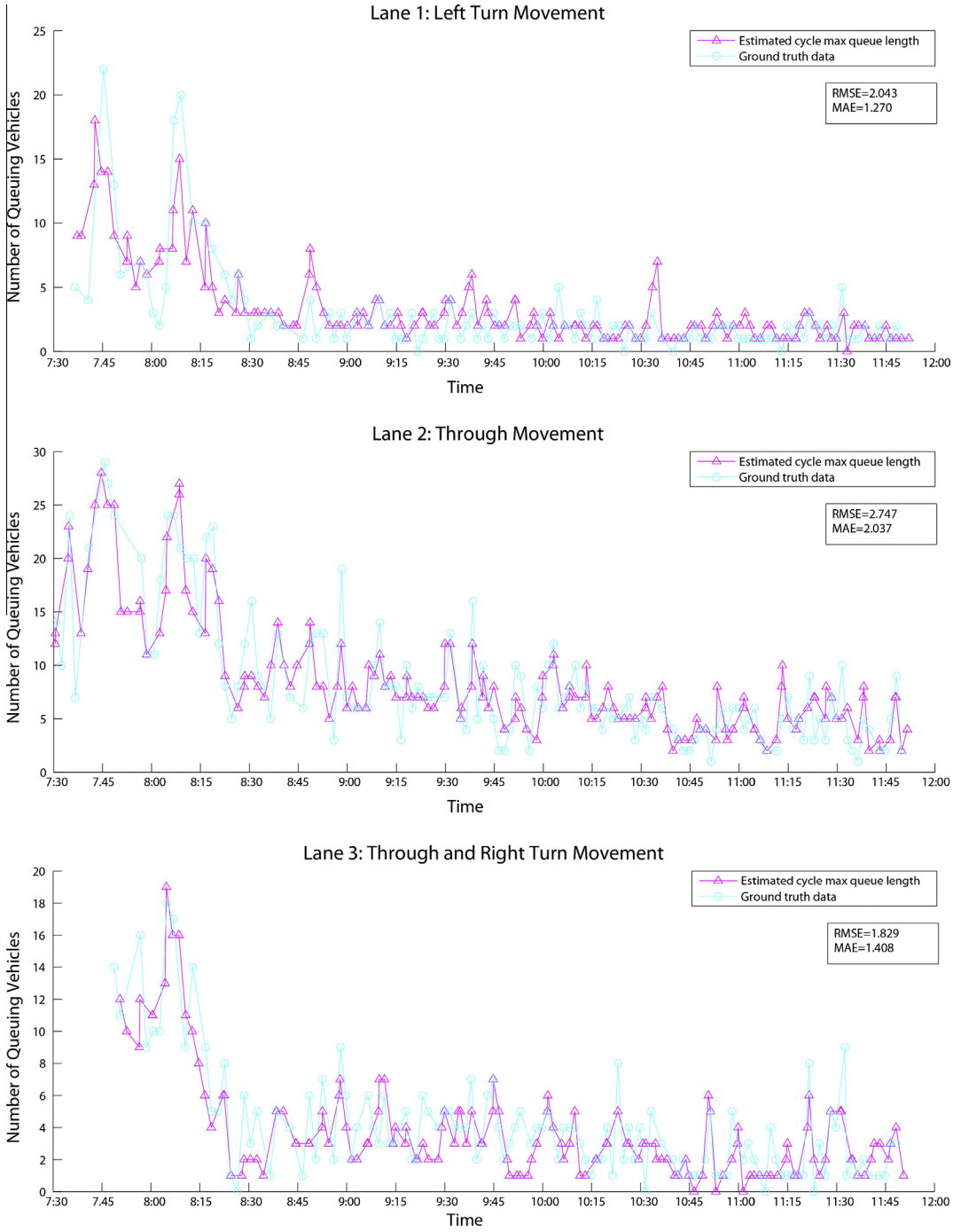

Fig. 9. Cycle maximum queue length comparison (November 26th).

The calculated results for RMSE and MAE are included in Figs. 8 and 9. For all test results of the two field experiment, low level of estimation errors are achieved with the RMSE and MAE controlled below 3.2 and 2.4 vehicles respectively. It is observed that for the less congested lane (Lane 1), the proposed model achieves very good estimation accuracy with the RMSE of 1.509 and MAE of 1.103 for the July 9th data, and the RMSE of 2.043 and MAE of 1.270 for the November 26th data. The error rate is higher for Lane 2, which is also the most congested lane, with the RMSE of 3.118 and MAE of 2.360 for the July 9th data, and RMSE of 2.747 and MAE of 2.037 for the November 26th data. The validation errors for Lane 3 fall between the results of Lane 1 and Lane 2, with the RMSE of 2.777 and MAE of 2.095 for the July 9th data, and RMSE of 1.829 and MAE

of 1.408 for the November 26th data. The relatively better performance of the model in less congested lanes might due to less intense interactions in low congestion situations. Since the boundary constrained car-following model used in this study assumes ideal driving behavior (all vehicles react to the changes of gap distances immediately after reaction time τ), which does not fully capture the complex and non-homogeneous interactions among vehicles especially in highly congested situations. The accuracy can be potentially improved if a more realistic car-following model is used, however, at the cost of additional computational efforts and more complex model calibration.

Several other factors can also contribute to the errors in the estimation, such as: (1) the model assumes homogeneous vehicles having the same driving behavior and vehicle characteristics. However, this is not necessarily the case in the field experiment, as there were a fraction of buses and trucks in the traffic. (2) There were a considerable amount of lane changing behaviors observed from the video camera data. Such lane changing behaviors would cause downstream vehicles to decelerate even when they were far away from the downstream intersection, which caused more severe queuing. (3) The inaccuracy could also arise from the incorrectly inferred arrival times for unmatched arrival vehicles during the equivalent cumulative arrival–departure curve reconstruction, especially when there are too few matched vehicles to perform the statistical inference. The LPR data with higher recognition accuracy are expected to greatly improve the model accuracy due to higher proportion of matched vehicles between the upstream and downstream intersections. (4) Finally, recognition errors from the LPR data are also possible sources that contribute to the model estimation error. For example, incorrectly recognized license plate number will cause failure in matching upstream and downstream vehicles, and introduce inaccuracy discussed in (3). It is also observed in the data that in very rare cases, a single vehicle may be recognized as two different unrecognizable vehicles, which could potentially lead to discrepancy between the estimations and ground truth data.

5. Conclusion

This paper proposed a queue length estimation model using LPR data, which provides efficient queue length estimation at lane level in real-time. The model first reconstructs the equivalent cumulative arrival–departure curve using a novel Gaussian process interpolation model based on underlying arrival process assumptions. With the complete arrival and departure information, a boundary constrained car-following model is applied to estimate the real-time queue length for each of the studying lane. The LPR data used in this research were obtained from a LPR system in the city of Langfang, Hebei province, China. In addition, two field experiments were conducted to obtain the actual cycle maximum queue length data for validation purposes. The results show that the model can well capture the variation patterns in the ground truth data. Good estimation accuracy can be achieved for cycle maximum queue length of each lane even when the underlying LPR system does not process a very high recognition accuracy. Moreover, other detailed traffic state measures such as real-time link travel time, the number of vehicles in the lane can also be obtained from the reconstructed equivalent cumulative arrival–departure curve.

Several future works can be done to further improve the proposed queue length estimation model. First, the proposed model is only validated against the cycle maximum queue lengths, other more comprehensive sources of ground truth data (e.g. real-time queue length and vehicle trajectories) can be collected in the future to further validate the proposed model. Second, the LPR system investigated in this study has relatively low recognition accuracy, which in average about only 50% of vehicles can be matched between the upstream and downstream intersections based on the recognized license plate number. It will be meaningful to test the performance of the proposed model using more accurate LPR data. Third, this paper used a simplified first-order OV car-following model to capture the traffic dynamics, which may not fully capture the complex interaction of vehicles during travel and queuing process. More comprehensive and realistic car-following model can be introduced to improve the estimation accuracy. Lastly, since the LPR data enable the re-identification of the same vehicle at multiple intersections in the network, it is possible to extend the current link level model to a network-wide traffic condition estimation model, in which the proposed model can be used as a sub-module to obtain the link level traffic conditions. All these extensions of the proposed model can contribute to more accurate lane-based real-time queue length estimation, and enable the full utilization of the valuable LPR data for urban traffic monitoring and operations.

Acknowledgement

This research was partially supported by National Natural Science Foundation of China (Grant No. 71361130015), the US National Science Foundation (Grant No. 1017933) and National Key Technology R&D Program (Grant No. 2014BAG03B03). We also thank the Bureau of Traffic Management of Langfang for providing the data.

Appendix A. Summary of notations used in the article

1. Equivalent cumulative arrival–departure curve reconstruction

N_k:	Total number of vehicles arrived/departed on the given lane in signal cycle k
n_k:	Total number of matched arrival vehicles on the given lane in signal cycle k
I_k:	Set of cumulative indices for arrivals which correspond to each vehicle in departure curve for signal cycle $k, I_k = \{i_1, i_2, \ldots, i_{N_k}\}$
H_k:	Set of cumulative indices for arrivals which correspond to each vehicle in departure curve for signal cycle $k, H_k = \{t_1, t_2, \ldots, t_{N_k}\}$
r_{T_s}, r_{L_s}:	Saturation discharging flow rate for vehicles from through and left turning movement respectively
r_{T_n}, r_{L_n}:	Normal arrival flow rate for vehicles from though and left turning movement respectively
r_R:	Arrival flow rate for vehicles from right turning movement
T_1:	The time when green light is ended for through movement, initialized as 0 when signal cycle begins
T_2:	The time when green light is ended for left turning movement, initialized as 0 when signal cycle begins
T_3:	The time when green light is ended for through movement of the orthogonal direction traffic, initialized as 0 when signal cycle begins
T_{cyc}:	Signal cycle length
t_a, t_b:	The time when the queue from the through movement (t_a) and left turning movement in orthogonal direction (t_b) at the upstream intersection is completely discharged, initialized as 0 when signal cycle begins
$\boldsymbol{x}$:	The vector of arrival timestamps for matched arrival vehicles on the arrival curve, $\boldsymbol{x} = (x_1, x_2, \ldots, x_{n_k})^T$
$\boldsymbol{x}^*$:	The vector of arrival timestamps for unmatched arrival vehicles on the arrival curve, $\boldsymbol{x}^* = \left(x_1^*, x_2^*, \ldots, x_{N_k - n_k}^*\right)^T$
$\boldsymbol{y}$:	The vector of cumulative indices for matched arrival vehicles on the arrival curve, $\boldsymbol{y} = \left(y_1, y_2, \ldots, y_{n_k}\right)^T$
$\boldsymbol{y}^*$:	The vector of cumulative indices for unmatched arrival vehicles on the arrival curve, $\boldsymbol{y}^* = \left(y_1^*, y_2^*, \ldots, y_{N_k - n_k}^*\right)^T$
$\mu(\cdot)$:	Function of the mean arrival process
η:	Normal distributed disturbance term
$k(\cdot)$:	Squared exponential kernel function
h_0, λ:	Output-scale and input-scale hyperparameter for the kernel function
$\boldsymbol{\theta}$:	Vector of the parameters to be inferred, $\boldsymbol{\theta} = (r_{T_s}, r_{T_n}, r_R, r_{L_s}, r_{L_n}, t_a, t_b)^T$
$N(\cdot)$:	Multivariate normal (or Gaussian) distribution
$\boldsymbol{m}^*(\cdot)$:	Posterior mean for unmatched arrival vehicles
$\boldsymbol{C}^*$:	Posterior covariance for unmatched arrival vehicles
$U_\rho[0, u]$:	Non-informative prior distribution defined on $[0, u]$ for flow rate parameters, ρ ($0 < \rho < 1$) is a predefined hyperparameter
r:	Sample acceptance ratio used in the M–H algorithm.
T_i^{arr}:	Timestamp for a matched vehicle i passing the upstream stop line
T_i^{dep}:	Timestamp for a matched vehicle i passing the downstream stop line
T_k^{cyc}:	Signal cycle starting timestamp for cycle k
TN_{cyc}:	A small time window for checking cycle starting vehicle

2. Queue length estimation via boundary constrained car-following model

v_d:	Desired travel speed
L:	Safe driving distance occupied by a vehicle
G:	Time gap in the speed function of Tordeux's car-following model, taken as 1.5s in actual implementation
$V(\cdot)$:	Speed function in Tordeux's car-following model
$s_n(t)$:	The position of the head of the vehicle n at time t
$\dot{s}_n(t)$:	The speed of the vehicle n at time t
$\Delta s_n(t)$:	Gap distance between vehicle n and its predecessor vehicle $n-1$ at time t
τ:	Reaction time
δt:	Simulation updating time step size, taken as 0.5s in actual implementation
v^{ini}:	Arrival speed for all vehicles entering the studying lane
L_R:	Length of the studying link
Δt:	The threshold time gap for the dummy initial vehicle

3. Model calibration and evaluation

$Q(\cdot)$:	Loss function for calibrating the car-following model
$\boldsymbol{\phi}$:	Car-following parameters to be calibrated, $\boldsymbol{\phi} = \left(L, v_d, v^{ini}\right)^T$
q_a^i:	Actual maximum queue length for cycle i
$q_e^i(\boldsymbol{\phi})$:	Maximum queue length estimated using the given car-following parameter $\boldsymbol{\phi}$
q_e^i:	Maximum queue length estimated using the proposed model
NC:	Total number of signal cycles that the actual cycle maximum queue length data are available
NC_c:	Total number of signal cycles used in the calibration of the car-following model

References

Akcelik, R., 1999. A Queue Model for HCM 2000. ARRB Transportation Research Ltd., Vermont South, Australia.

Balke, K., Charara, H., Parker, R., 2005. Development of a Traffic Signal Performance Measurement System (TSPMS). Texas Transportation Institute, Report 0-4422-2.

Bando, M., Hasebe, K., Nakanishi, K., Nakayama, A., Shibata, A., Sugiyama, Y., 1995. Phenomenological study of dynamical model of traffic flow. J. Phys. I 5 (11), 1389–1399.

Ban, X., Hao, P., Sun, Z., 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.

Beijing Municipal Committee, July 2012. The Twelfth Five-Year Plan for the Transportation Development of Beijing. Retrieved from <http://zhengwu.beijing.gov.cn/ghxx/sewgh/t1237237.htm>.

Berger, J.O., 1985. Statistical Decision Theory and Bayesian Analysis. Springer.

Bertini, R.L., Lasky, M., Monsere, C.M., 2005. Validating predicted rural corridor travel times from an automated license plate recognition system: Oregon's frontier project. In: Proceedings of 2005IEEE Intelligent Transportation Systems, pp. 296–301.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer, New York.

Carson, Y., Maria, A., 1997. Simulation optimization: methods and applications. In: Proceedings of the 29th Conference on Winter Simulation. IEEE Computer Society, pp. 118–126.

Chang, T-H., Lin, J-T., 2000. Optimal signal timing for an oversaturated intersection. Transport. Res. Part B: Methodol. 34 (6), 471–491.

Coifman, B., Cassidy, M., 2002. Vehicle reidentification and travel time measurement on congested freeways. Transport. Res. Part A: Policy Pract. 36, 899–917.

Coifman, B., Krishnamurthy, S., 2007. Vehicle reidentification and travel time measurement across freeway junctions using the existing detector infrastructure. Transport. Res. Part C: Emerg. Technol. 15 (3), 135–153.

Fritsch, F.N., Carlson, R.E., 1980. Monotone piecewise cubic interpolation. SIAM J. Numer. Anal. 17 (2), 238–246.

Hofleitner, A., Herring, R., Bayen, A., 2012. Arterial travel time forecast with streaming data: a hybrid approach of flow modeling and machine learning. Transport. Res. Part B: Methodol. 46 (9), 1097–1122.

Jeng, S., Tok, Y., Ritchie, S., 2010. Freeway corridor performance measurement based on vehicle reidentification. IEEE Trans. Intell. Transport. Syst. 11 (3), 639–646.

Kahaner, D., Moler, C., Nash, S., 1988. Numerical Methods and Software. Prentice Hall.

Kwong, K., Kavaler, R., Rajagopal, R., Varaiya, P., 2009. Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors. Transport. Res. Part C: Emerg. Technol. 17 (6), 586–606.

Lighthill, M.J., Whitham, G.B., 1955. On kinematic waves. I: flood movement in long rivers. II: a theory of traffic flow on long crowded roads. Proc. Roy. Soc. (Lond.) A229, 281–345.

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.

May, A.D., 1975. Traffic flow theory-the traffic engineers challenge. In: Proceedings of the Institute of Traffic Engineering, pp. 290–303.

Mirchandani, P.B., Zou, N., 2007. Queuing models for analysis of traffic adaptive signal control. IEEE Trans. Intell. Transport. Syst. (1)

Newell, G.F., 1961. Nonlinear effects in the dynamics of car following. Oper. Res. 9 (2), 209–229.

Newell, G.F., 1965. Approximation methods for queues with application to the fixed-cycle traffic light. SIAM Rev. 7, 223–240.

Oh, C., Ritchie, S.G., Jeng, S., 2007. Anonymous vehicle reidentification using heterogeneous detection systems. IEEE Trans. Intell. Transport. Syst. 8 (3), 460–469.

Richards, P.J., 1956. Shock waves on the highway. Oper. Res. 4, 42–51.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrain, S., 2013. Gaussian processes for time-series modelling. Phil. Trans. Roy. Soc. A: Math. Phys. Eng. Sci. 371 (1984).

Sharma, A., Bullock, D.M., Bonneson, J., 2007. Input-output and hybrid techniques for real-time prediction of delay and maximum queue length at a signalized intersection. Transport. Res. Rec. 2035, 88–96.

Skabardonis, A., Geroliminis, N., 2008. Real-time monitoring and control on signalized arterials. J. Intell. Transport. Syst. 12 (2), 64–74.

Stephanopoulos, G., Michalopoulos, P.G., 1979. Modeling and analysis of traffic queue dynamics at signalized intersections. Transport. Res. 13A.

Tordeux, A., Lassarre, S., Roussignol, M., Aguiléra, V., 2015. Generic first-order car-following models with stop-and-go waves and exclusion. In: Traffic and Granular Flow'13. Springer International Publishing, pp. 485–493.

Vigos, G., Papageorgiou, M., Wang, Y., 2008. Real-time estimation of vehicle-count within signalized links. Transport. Res. Part C: Emerg. Technol. 16 (1), 18–35.

Webster, F.V., 1958. Traffic Signal Settings. Road Research Laboratory Technical Paper No. 39, HMSO, London.

Yasin, A.M., Karim, M.R., Abdullah, A.S., 2009. Travel time measurement in real-time using automatic number plate recognition for Malaysian environment. In: Proceedings of the Eastern Asia Society for Transportation Studies, vol. 2009(0), pp. 324–324.