A Car Following-Based Method for Vehicle Trajectory Connection

Preprint · July 2021

DOI: 10.13140/RG.2.2.28609.28001/1

Xiaowei Shi
University of Michigan
22 PUBLICATIONS **156** CITATIONS

Dongfang Zhao
University of South Florida
10 PUBLICATIONS **88** CITATIONS

Xiaopeng Li
University of South Florida
172 PUBLICATIONS **3,814** CITATIONS

A Car-Following-Based Method for Vehicle Trajectory Connection

Xiaowei Shi, Dongfang Zhao, Xiaopeng Li*

Department of Civil and Environmental Engineering, University of South Florida, Tampa, FL, 33620, USA

Abstract

High-accuracy long-coverage vehicle trajectory data can benefit the investigations of various traffic phenomena. However, most of the existing vehicle trajectory datasets may miss parts of the trajectories due to sensing limitation and thus contain substantial broken vehicle trajectories. This may restrict the analyses of traffic dynamics and the validations of corresponding findings. To address this issue, this paper proposed a car-following-based (CF-based) vehicle trajectory connection method that can connect broken vehicle trajectories. The proposed method can not only fill missing data points caused by detection errors but also can help connect vehicle trajectory datasets from different sensors. To illustrate the performance of the proposed CF-based method, the proposed method was employed to process a series of vehicle trajectory datasets. The datasets were extracted from aerial videos recorded at several successive spaces of Interstate-75, United States. Compared with several benchmark trajectory connection methods, the results showed that the proposed method has advantages in both trajectory connection accuracy and trajectory consistency aspects. To the best of the authors' knowledge, the dataset processed by the proposed CF-based method, named by HIGH-SIM, is the longest vehicle trajectory dataset in the literature. The dataset has been published online for public use.

Keywords: Car-Following Model, Vehicle Kinematics, Vehicle Trajectory Connection, Vehicle Trajectory Dataset

1. Introduction

Vehicle trajectories, as the positions of a stream of vehicles over time along a guideway (Daganzo, 1997), can provide informative insights into various traffic-related studies, such as traffic flow theory, traffic simulation modeling, traffic safety measures, and traffic management. Kim and Cao (2010) classified existing vehicle trajectory data collection methods into two categories, including vehicle-based methods (Anuar and Cetin, 2017; Coifman et al., 2016; Victor, 2014; Zhao et al., 2017) and video-based methods (Babinec et al., 2014; Kim et al., 2019; Xu et al., 2017; Zhao and Li, 2019). The vehicle-based methods suggest collecting vehicle trajectory data by probe vehicles. The probe vehicles equipped with position/distance measurement sensors (e.g., Lidar, Radar, GPS) travel along the testing road segment and thus the trajectories of the probe vehicles, as well as the surrounding vehicles, can be obtained. The drawback of the vehicle-based methods is apparent. Since only the trajectories of the probe vehicle and its surrounding vehicles will be collected, the data penetration rate with respect to the entire traffic is often very low. On the other hand, the video-based methods extract vehicle trajectories from traffic videos, which are recorded by roadside or aerial cameras over the investigated road segments. By tracking each vehicle's motions in the videos, the trajectories of all vehicles traveling along the road segments can be obtained.

Due to the rapid developments of aerial video recording technologies, e.g., unmanned aerial vehicles with high-definition cameras, video-based trajectory data are becoming more appealing with advantages such as scalability, flexibility, economy, and unbiasedness (Kim and Cao, 2010). Thus, in recent years, the collection of video-based trajectory data has attracted wide attention of researchers both from industry and academia (NGSIM, 2006; Apeltauer et al., 2015; Azevedo et al., 2014; Barmpounakis and Geroliminis, 2020; Chen et al., 2020; Krajewski et al., 2018; Punzo et al., 2011; Shi et al., 2021). Despite these merits of the video-based data collection method with the advanced technologies, we would like to point out two types of fundamental issues in the datasets, including *detection errors* and *limited ranges* as specified below.

Detection errors can be further divided into two categories in terms of the origin of the errors, such as the source errors and extraction errors. The source errors are caused due to the partially losing video feed. Since the videos are recorded in the air, it is possible that the target sites are partially blocked by facilities (e.g., bridges, signals, billboards, buildings) around the road, and thus the vehicle motions in the blocked areas are lost. As illustrated in Figure 1, a bridge across the recorded road segment leads to the loss of the trajectories of all vehicles underneath.

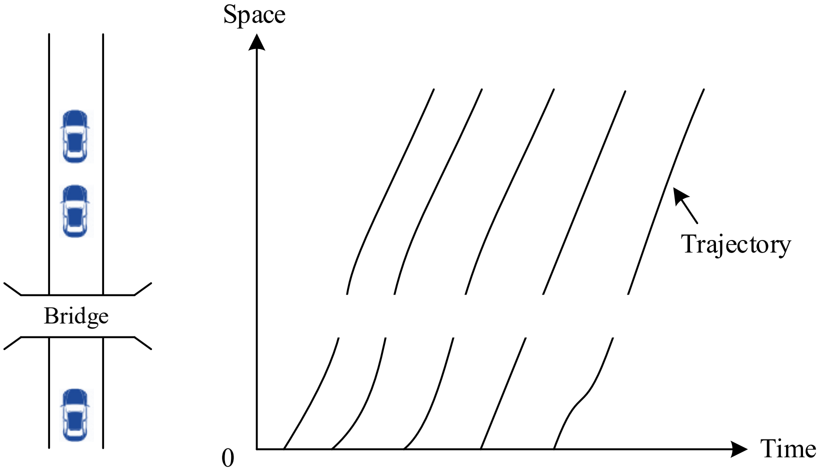


Figure 1. Source errors illustration.

The extraction errors are caused by trajectory extraction methods. To obtain trajectory data from video sources, various trajectory extraction methods have been proposed in the literature to track each vehicle's motions in videos. Reliable vehicle tracking is an extremely challenging problem, which falls into the computer vision field and has attracted intensive studies in the past few years (Jazayeri et al., 2011; Wang et al., 2008; Zhang et al., 2007). Although plenty of high-performance methods have been proposed to tackle this problem, as far as we know, none of them can always guarantee 100% detection rates. Particularly, some exogenous factors, such as the weather, light, wind, camera angle, etc., may impact the quality of the recorded video as well as the detection rates. Thus, missing detections in the trajectory extraction process seem inevitable. Once a missing detection happens, the original long trajectory will be broken into shorter trajectories by the missing points, which degrades the quality of the obtained dataset. As illustrated in Figure 2, due to the angles of the cameras, the vehicle sizes in the video are gradually reduced as approaching the end of the road segment. This leads to the decrease of the detection rates and thus the extracted trajectories are broken into small pieces in the downstream.

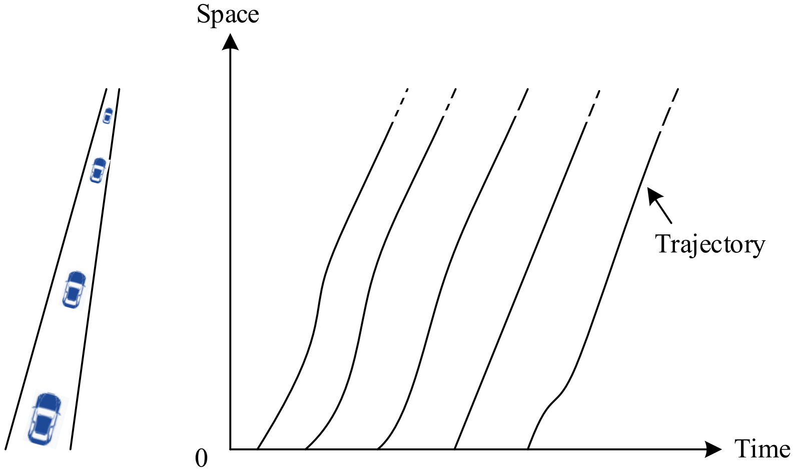


Figure 2. Extraction errors illustration.

To overcome the *detection errors* issue, data post-processing studies on trajectory smoothing were conducted (Lee and Krumm, 2011; Punzo, 2009; Siddique, 2019; Wu, 2018; Xin, 2008). Interested readers can refer to Lee and Krumm (2011) for a detailed review of vehicle trajectory post-processing. The studies investigating the vehicle trajectory connection are scarce in the literature. Kim et al. (2019) connected broken trajectories by extending the trajectory of each vehicle in three seconds with a constant speed. Tong et al. (2017) connected broken trajectories by extending the linear interpolation method considering the historical data and contextual arrival information. Zhang and Jin (2019) performed vehicle trajectory data cleaning and connection (or stitching) while extracting the data, but they did not specify the adopted trajectory connection method. One can see that although this topic started to draw attention in recent years as the emergence of aerial video recording technologies, the existing methods in the literature still are fairly simple, which may only yield results with limited quality without capturing driving behavior and physics (i.e., car-following behavior). To the best of the author's knowledge, Sazara et al. (2017) is the only study that connects broken vehicle trajectories considering car-following characteristics. They collected raw vehicle trajectory data by the vehicle-based method using Lidar sensors. To connect the broken trajectories, they first extended the broken trajectories with Gipp's car-following model, and then a simple reshaped operation was proposed to connect the two pieces of the broken trajectories. Despite the success of this study, we would like to point out that there is a strong underlying assumption. That is, the vehicle IDs of the broken trajectories are given and thus they can be easily matched by each individual vehicle. This assumption can be satisfied easily if the datasets are collected by the vehicle-based method (e.g., Lidar, Radar, etc.). However, how to efficiently match two broken trajectories without explicit vehicle ID tags when the size of the extracted dataset is large (collected by the video-based methods) is an intriguing problem. This problem even becomes harder if the recorded videos include multiple lanes, in which broken vehicle trajectories across different lanes may belong to the same vehicle due to possible lane change maneuvers. Figure 3 illustrates the described lane change maneuvers on a simple two-lane road. Figure 3 (a) and (b) are the trajectories from lane 1 and lane 2, respectively. If a vehicle makes a lane change from lane 1 to lane 2, as highlighted in both Figure 3 (a) and (b), the trajectory of this vehicle for each lane is "incomplete". A wrong connection may happen if a broken trajectory is coincidently near to the "incomplete" trajectory, as illustrated in the Figure 3 (b). In addition, without further investigations of feasible ranges for vehicle trajectories in the space-time diagram, the reshaped operation may easily generate trajectories that violate vehicle kinematic constraints.

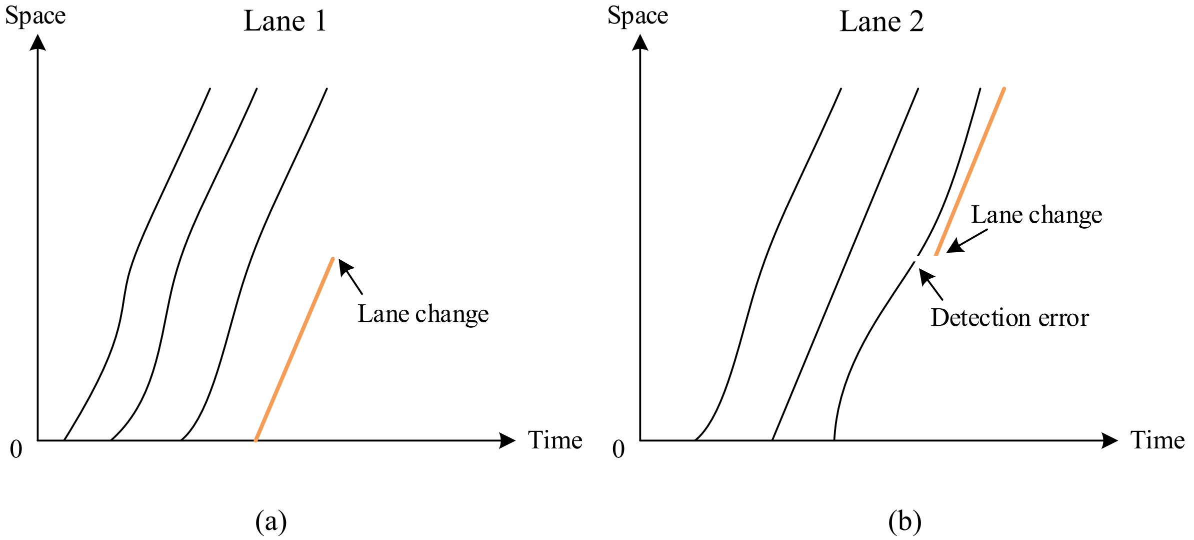


Figure 3. Lane change scenario illustration.

The second fundamental issue that may exist in the data is *limited ranges*. Compared to the vehicle-based collection methods, the video-based collection methods have advantages on the data collection scale (e.g., the number of collected trajectories). However, the detection ranges (e.g., the length of vehicle trajectories) are greatly constrained by both the detection accuracy and the altitude of the cameras. That is, the cameras with a higher altitude can record the traffic video of a greater range, yet the requirement on the detection accuracy of the cameras will be higher. Table 1 lists the length of several video-based trajectory datasets reported in the literature. We see that the longest trajectory dataset is the NGSIM dataset, the length of which is 640 meters. However, it is still insufficient to observe a full life cycle of traffic phenomena (e.g., traffic bottleneck development and dissipation). To break the constraints on the detection ranges, a direct solution is to either improve the performance of cameras (detection accuracy) or fly the cameras to a higher altitude. However, these solutions belong to the optics or mechanical field and are out of the knowledge scope of a transportation engineer. To resolve this issue from the transportation engineering perspective, Raju et al. (2021) proposed a concept of stitching trajectory data from different cameras. However, methods of stitching videos from different cameras reliably have not been proposed following this seminal concept.

Table 1. Trajectory length comparisons among video-based trajectory datasets.

Publication	Location	Trajectory length	Availability	Link
NGSIM	Highways, United States	640 meters	Yes	https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm
Krajewski et al., (2018)	Highways, Geman	420 meters	Yes	https://www.highd-dataset.com/#
Azevedo et al. (2014)	Motorway, Portugal	500 meters	No	/
Kim et al. (2019)	Expressways, Korea	188 meters	No	/
Babinec et al. (2014)	Ring road, Czech Republic	300 meters	No	/
Xu et al. (2017)	Freeway and urban roads, China	160 meters	No	/
Barmpounakis and Geroliminis (2020)	Urban roads, City of Athens, Greece	350 meters	Yes	https://open-traffic.epfl.ch/
Our dataset: HIGH-SIM	Highways, United States	2,438 meters	Yes	https://github.com/CATS-Lab-USF

Overall, one can see that the existing video-based datasets are constrained by the aforementioned two issues, which may restrict the analyses of traffic dynamics and the validations of corresponding findings. To help circumvent these issues, this paper proposed a car following-based (CF-based) method for vehicle trajectory connection, in which the broken vehicle trajectories were connected based on the car-following theory. The proposed method can not only fill missing data points caused by *detection errors* (solving the *detection errors* issue) but also can help connect trajectory data from different sensors (solving the *limited ranges* issue). To illustrate the performance of the proposed CF-based method, we processed a series of vehicle trajectory datasets that were extracted from aerial videos recorded at several successive segments of Interstate-75 (28°08'37.2"N 82°22'58.8"W to 28°10'16.2"N 82°23'38.0"W), United States (See Shi et al. (2021) for the detailed trajectory extraction method). The results showed that the proposed method outperformed several benchmark methods in both trajectory connection accuracy and trajectory consistency aspects. Moreover, the dataset processed by the proposed CF-based method, named by HIGH-SIM, has been published in the data shared link of both Federal Highway Administration, U.S. Department of Transportation (https://highways.dot.gov/) and Connected and Autonomous Transportation Systems Lab, University of South Florida (https://github.com/CATS-Lab-USF) for public use. Based on the best of the authors' knowledge, the HIGH-SIM dataset is the longest high-resolution vehicle trajectory dataset capturing all vehicles in the traffic stream among the publicly available ones, which includes a full life cycle of the bottleneck. This paper focuses on the trajectory connection method we adopted to generate the dataset. Interested readers can refer to Shi et al. (2021) for a detailed introduction about the HIGH-SIM dataset.

The disposition of this paper is as follows. Section 2 describes the trajectory connection problem investigated in this paper. Section 3 exhibits the proposed CF-based vehicle trajectory connection method. A series of numerical experiments are conducted in Section 4 to demonstrate the performance of the proposed method. Section 5 concludes the paper and discusses the limitations of the proposed method and possible solutions.

2. Problem Statement

The investigated trajectory connection problem is stated as follows. Let $\mathcal{N}$ denote the set of N vehicle trajectories in the investigated dataset, and each trajectory is labeled as $i \in \mathcal{N} := \{1,2,\cdots,N\}$. The trajectory data are captured in a spatial range $[0, Y]$ within a continuous time period $[0, \mathbb{T}]$. In practice, the data are usually only available at discrete time points, and thus the time period is discretized into time points $\mathcal{T} := \{0,1,2,\cdots,T\}$ with time interval $\Delta := \mathbb{T}/T$. Each trajectory $i \in \mathcal{N}$ is defined as the composition of a pair of arrays, $(\mathscr{y}_i, \mathscr{t}_i)$ with time point array $\mathscr{t}_i = [t_{i1}, t_{i2}, \cdots, t_{in_i}]$ denoting the consecutive time points and location array $\mathscr{y}_i = [y_{i1}, y_{i2}, \cdots, y_{in_i}]$ denoting the corresponding location coordinates (e.g., mileposts) of the trajectory at these time points, where n_i is the total number of the data points in trajectory i. In addition, $\mathscr{v}_i := [v_{i1}, v_{i2}, \cdots, v_{in_i}]$, $\mathscr{a}_i = [a_{i1}, a_{i2}, \cdots, a_{in_i}]$, and $\mathscr{p}_i = [p_{i1}, p_{i2}, \cdots, p_{in_i}]$ denote the velocities, accelerations, and preceding trajectory's labels of trajectory i at the corresponding time points, respectively. If there is no preceding trajectory preceding to trajectory i at time t_{ik}, then p_{ik} is set to 0. The length of the vehicle associated with trajectory i is denoted by L_i.

It is expected that a great number of trajectories in dataset $\mathcal{N}$ only capture a portion of the subject vehicles' motions due to the aforementioned issues. These trajectories, referred as *broken trajectories*, can be identified if they satisfy either of the following conditions: (1) if $t_{i1} > 0$ and $y_{i1} > 0, \forall i \in \mathcal{N}$; or (2) if $t_{in_i} < T$ and $y_{in_i} < Y, \forall i \in \mathcal{N}$. We collect all broken trajectories from $\mathcal{N}$ and denote the broken trajectory dataset as $\mathcal{N}^b \subseteq \mathcal{N}$. To understand the missing segment of each broken trajectory, according to the conditions, we further classified the broken trajectory dataset $\mathcal{N}^b$ into three subsets, such as $\mathcal{N}_1^b, \mathcal{N}_2^b, \mathcal{N}_{12}^b$.

For those broken trajectories that satisfy condition (1), we store them into $\mathcal{N}_1^b$, which means that these trajectories are broken at the origin side (i.e., a segment before (y_{i1}, t_{i1}) is missed). For those broken trajectories that satisfy condition (2), we store them into $\mathcal{N}_2^b$, which means that these trajectories are broken at the end side (i.e., a segment after (y_{in_i}, t_{in_i}) is missed). For those broken trajectories that satisfy both conditions (1) and (2), we store them into $\mathcal{N}_{12}^b$, which means that these trajectories are broken at both sides. Thus, we have $\{\mathcal{N}_1^b \cup \mathcal{N}_2^b\} = \mathcal{N}^b$, and $\{\mathcal{N}_1^b \cap \mathcal{N}_2^b\} = \{\mathcal{N}_{12}^b\}$. For the specific example shown in Figure 4, $\mathcal{N} = \{1,2,\cdots,8\}$, $\mathcal{N}^b = \{2,3,5,6,7\}$, $\mathcal{N}_1^b = \{3,6,7\}$, $\mathcal{N}_2^b = \{2,5,6\}$, $\mathcal{N}_{12}^b = \{6\}$.

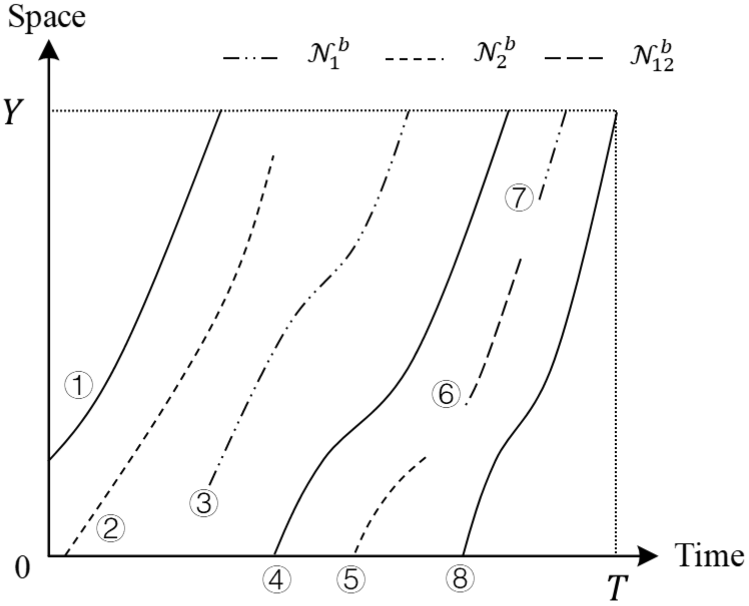


Figure 4. Problem statement.

This paper aims to connect these broken trajectories considering vehicle driving characteristics (i.e., car-following behavior) and thus enhances the quality of the dataset. To this end, this paper proposes a CF-based method for vehicle trajectory connection, in which the broken vehicle trajectories can be connected based on car-following theory.

3. Methodology

The proposed CF-based vehicle trajectory connection method includes two steps: (1) car-following model calibration; and (2) vehicle trajectory connection.

3.1 Car-following model calibration

To connect the broken vehicle trajectories considering the car-following behavior, we first calibrate a car-following model with the dataset. The car-following model adopted in this paper is the Pitt car-following model (Drew, 1968), which is shown in Equation (1).

$$d = L + 3.04878 + c \cdot w_2 + b \cdot c \cdot (w_1 - w_2)^2, \tag{1}$$

where d is the spacing headway between the leading vehicle and following vehicle, L is the length of the leading vehicle, c is a sensitivity factor, w_2 is the speed of the following vehicle, w_1 is the speed of the leading vehicle, and b is a calibration constant ($if\ w_2 > w_1, b = 0.1$; otherwise, $b = 0$). Note that this paper focuses on connecting the broken trajectories considering the car-following behavior. We select the

Pitt car-following model without further comparing it to other car-following models. The comparison among different car-following models may be out of the scope of this paper, and interested readers can refer to (van Hinsbergen et al., 2015, Rahman, 2013) for detailed comparisons.

With the Pitt car-following model, a series of car-following trajectory pairs, including the time ($\mathscr{t}$), location ($\mathscr{y}$), and speed ($\mathscr{v}$) of both leading and following vehicles, is extracted from the dataset to calibrate the model. By referring to the preceding trajectory label array (i.e., $\mathscr{p}$), the car-following trajectory pairs can be obtained easily, i.e., $if\ p_{ik} \neq 0$, extracting t_{ik}, y_{ik}, v_{ik} and $t_{p_{ik}k}, y_{p_{ik}k}, v_{p_{ik}k}$. With the car-following trajectory pairs, we calibrate the Pitt car-following model with a greedy algorithm. The error measurement function is used to evaluate the fitness of the Pitt car-following model, as shown in Equation (2):

$$F_e = \sqrt{\sum_{i=1}^{I} \sum_{k=1}^{n_i} (\hat{y}_{ik} - y_{ik})^2 / I}, \tag{2}$$

where F_e is the standard error of the estimated location ($\hat{y}_{ik}$) and the real location (y_{ik}), I is the total number of the car-following trajectory pairs, n_i is the number of the data captured for trajectory pair i.

The detailed calibration procedures are shown in Figure 5. The calibration starts by initializing the model parameters b^1 and c^1 with random values $b^1 \in [0,0.1]$, $c^1 \in [0,2]$, and inputting the convergence criteria ϵ and step size γ^b, γ^c. In each iteration m, the variable b^m, c^m will be updated as $b^m = \gamma^b * random(0,1) + b^{m-1}$, $c^m = \gamma^c * random(0,1) + c^{m-1}$, where $random(0,1)$ is a random number in the range of $(0,1)$. In the next step, the error of the fitness is calculated according to Equation (2). These procedures will be repeated until the variation of the estimation error $|F_e^m - F_e^{m-1}| \leq \epsilon$, which means that the optimal parameters b, c are obtained.

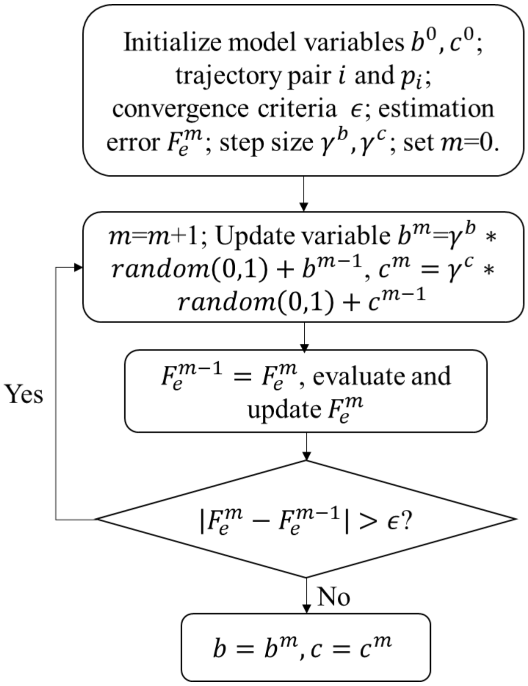

Initialize model variables b^0, c^0; trajectory pair i and p_i; convergence criteria ϵ; estimation error F_e^m; step size γ^b, γ^c; set m=0.
m=m+1; Update variable b^m=$\gamma^b * random(0,1) + b^{m-1}$, $c^m = \gamma^c * random(0,1) + c^{m-1}$
$F_e^{m-1} = F_e^m$, evaluate and update F_e^m
Yes
$|F_e^m - F_e^{m-1}| > \epsilon$?
No
$b = b^m, c = c^m$

Figure 5. Pitt car-following model calibration process

3.2 Vehicle trajectory connection

With the calibrated Pitt car-following model, the proposed CF-based vehicle trajectory connection method connects the broken trajectories with the following two sub-steps. Sub-step 1: Extending the broken vehicle trajectories with the calibrated car-following model. Sub-step 2: Connecting the broken trajectories considering vehicle kinematics constraints.

Sub-step 1:

For those broken trajectories in $\mathcal{N}_1^b$ (or $\mathcal{N}_2^b$), the trajectories are extended backward (or forward) t^e time intervals with the calibrated Pitt car-following model. We name the extended trajectories as the transition trajectories in this paper. The time length of the transition trajectories is denoted by t^e, which is a given value dependent on the quality of dataset $\mathcal{N}$. That is, a large t^e value is set if the time length of the missing trajectories is long and vice versa. The effects of different t^e values on the trajectory connection will be analyzed in Section 4. Assume that there are two broken trajectories labeled by $\alpha \in \mathcal{N}_1^b$ and $\beta \in \mathcal{N}_2^b$, and the preceding vehicle trajectory are labeled by α' and β'. The arrays of the transition trajectory for trajectory $\alpha \in \mathcal{N}_1^b$ (i.e., $\mathscr{y}_\alpha^{b_1}, \mathscr{t}_\alpha^{b_1}, \mathscr{v}_\alpha^{b_1}, \mathscr{a}_\alpha^{b_1}, \mathscr{p}_\alpha^{b_1}$) are generated according to Equations (3)-(8), where superscript b^1 indicates that the trajectory extends backward.

$$\mathscr{y}_\alpha^{b_1} \coloneqq \left\{ y_{\alpha k}^{b_1} = \begin{cases} y_{\alpha 1}, k = t^e \\ y_{\alpha(k+1)}^{b_1} - v_{\alpha(k+1)}^{b_1} \cdot \Delta, \forall k \in \{1,2,\cdots,t^e-1\} \end{cases} \right\}, \tag{3}$$

$$\mathscr{t}_\alpha^{b_1} \coloneqq \left\{ t_{\alpha k}^{b_1} = t_{a1} - (t^e - k) \cdot \Delta, \forall k \in \{1,2,\cdots,t^e\} \right\}, \tag{4}$$

$$\mathscr{v}_\alpha^{b_1} \coloneqq \left\{ v_{\alpha k}^{b_1} = \begin{cases} v_{\alpha 1}, k = t^e \\ \arg_{w_2}\left(y_{\alpha'\left(\arg_{k' \in \mathscr{t}_{\alpha'}}(t_{\alpha' k'} = t_{\alpha k}^1)\right)} - y_{\alpha k}^{b_1} = A \right), \forall k \in \{1,2,\cdots,t^e-1\} \end{cases} \right\}, \tag{5}$$

$$A = L + 3.04878 + kw_2 + bk\left(v_{\alpha'\left(\arg_{k' \in \mathscr{t}_{\alpha'}}(t_{\alpha' k'} = t_{\alpha k}^{b_1})\right)} - w_2 \right)^2, \tag{6}$$

$$\mathscr{a}_\alpha^{b_1} \coloneqq \left\{ a_{\alpha k}^{b_1} = \begin{cases} a_{\alpha 1}, k = t^e \\ (v_{a(k+1)}^{b_1} - v_{\alpha k}^{b_1})/\Delta, \forall k \in \{1,2,\cdots,t^e-1\} \end{cases} \right\}, \tag{7}$$

$$\mathscr{p}_\alpha^{b_1} {:=} \left\{ p_{\alpha k}^{b_1} = \alpha', \forall k \in \{1,2,\cdots,t^e\} \right\}. \tag{8}$$

Similarly, the arrays of the transition trajectory for trajectory $\beta \in \mathcal{N}_2^b$ (i.e., $\mathscr{y}_\beta^{b_2}, \mathscr{t}_\beta^{b_2}, \mathscr{v}_\beta^{b_2}, \mathscr{a}_\beta^{b_2}, \mathscr{p}_\beta^{b_2}$) can be calculated according to Equations (9)-(14), where superscript b^2 indicates that the trajectory extends forward.

$$\mathscr{y}_\beta^{b^2} \coloneqq \left\{ y_{\beta k}^{b^2} = \begin{cases} y_{\beta n_\beta} + v_{\beta k}^{b^2} \cdot \Delta, k = 1 \\ y_{\beta(k-1)}^{b^2} + v_{\beta(k-1)}^{b^2} \cdot \Delta, \forall k \in \{2,3,\cdots,t^e\} \end{cases} \right\}, \tag{9}$$

$$\mathscr{t}_\beta^{b^2} \coloneqq \left\{ t_{\beta k}^{b^2} = t_{\beta n_\beta} + k \cdot \Delta, \forall k \in \{1,2,\cdots,t^e\} \right\}, \tag{10}$$

$$\mathscr{v}_\beta^{b^2} \coloneqq \left\{ v_{\beta k}^{b^2} = \arg_{w_2}\left(y_{\beta'\left(\arg_{k' \in \mathscr{t}_{\beta'}}(t_{\beta' k'} = t_{\beta k}^{b^2})\right)} - y_{\beta k}^{b^2} = B \right), \forall k \in \{1,2,\cdots,t^e\} \right\}, \tag{11}$$

$$B = L + 3.04878 + kw_2 + bk\left(v_{\beta'\left(\arg_{k' \in \mathscr{t}_{\beta'}}(t_{\beta' k'} = t_{\beta k}^{b^2})\right)} - w_2 \right)^2, \tag{12}$$

$$a_{\beta}^{b^2} := \left\{ a_{\beta k}^{b^2} = \begin{cases} (v_{\beta k}^{b^2} - v_{\beta n_{\beta}})/\Delta, k = 1 \\ (v_{\beta k}^{b^2} - v_{\beta (k-1)}^{b^2})/\Delta, \forall k \in \{2,3,\cdots,t^e\} \end{cases} \right\}, \tag{13}$$

$$\wp_{\beta}^{b^2} := \left\{ p_{\beta k}^{b^2} = \beta', \forall k \in \{1,2,\cdots,t^e\} \right\}. \tag{14}$$

We illustrate the obtained transition trajectories with trajectory 6 described previously, which belongs to $\mathcal{N}_{12}^{b}$ including both backward and forward extensions. As shown in Figure 6, by referring to the preceding vehicle trajectory, the transition trajectories are generated regarding the Pitt car-following model colored with green. Here we denote the set of all transition trajectories as $\mathcal{N}^e$. For the specific example that we described in Section 2, $\mathcal{N}^e := \begin{Bmatrix} \mathscr{y}_3^{b^1} & \mathscr{y}_7^{b^1} & \mathscr{y}_2^{b^2} & \mathscr{y}_5^{b^2} & \mathscr{y}_6^{b^1} & \mathscr{y}_6^{b^2} \\ \mathscr{t}_3^{b^1}, & \mathscr{t}_7^{b^1}, & \mathscr{t}_2^{b^2}, & \mathscr{t}_5^{b^2}, & \mathscr{t}_6^{b^1}, & \mathscr{t}_6^{b^2} \end{Bmatrix}$.

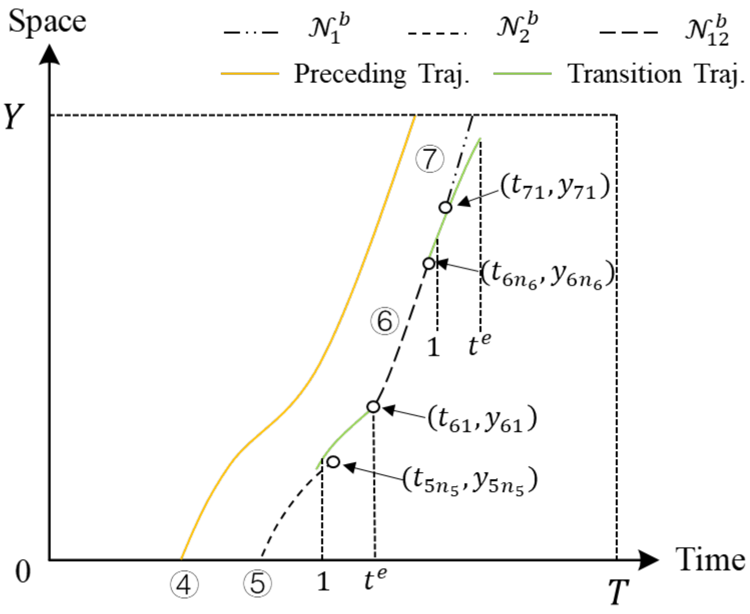


Figure 6. Transition trajectory illustration.

Sub-step 2:

In step 2, we propose the criterion for connecting two broken trajectories through the transition trajectories. In addition, vehicle kinematics constraints are considered while connecting the trajectories.

The criterion for trajectory connection is defined as follows. Assume that there is an arbitrary transition trajectory $\mu_i^* \in \mathcal{N}^e$ ($*$ can be either b_1 or b_2). For each trajectory $i' \in \mathcal{N}^b, i' \neq i$, the location difference between the transition trajectory and the broken trajectory, which is denoted as $\tau_{\mu_i^* i'}$, can be calculated by

$$\tau_{\mu_i^* i'} := \left(\sum_{k_1 = \arg_{k \in \{1,2,\cdots,t^e\}}(t_{ik}^* = C)}^{\arg_{k \in \{1,2,\cdots,t^e\}}(t_{ik}^* = D)} y_{ik}^* - \sum_{k_2 = \arg_{k' \in \mathscr{t}_{i'}}(t_{i'k'} = C)}^{\arg_{k' \in \mathscr{t}_{i'}}(t_{i'k'} = D)} y_{i'k_2} \right) \cdot \frac{\Delta}{D - C}, \tag{15}$$
$$C = \max(t_{i1}^*, t_{i'1}), D = \min\left(t_{it^e}^*, t_{i'n_{i'}}\right).$$

If we have $\tau_{\mu_i^* i'} < \epsilon^S$ and $\tau_{\mu_i^* i'} = \min\left\{\tau_{\mu_i^* i'}\right\}_{i' \in \mathcal{N}^b, i' \neq i}$, we consider the two trajectories i and i' are the trajectories of the same vehicle and thus we can connect them. ϵ^S is a given error term for evaluating the location difference between the transition trajectory and the broken trajectory. Note that a large ϵ^S value may cause a wrong connection, while a small ϵ^S value may reject a correct connection due to the estimation errors. In the practice implementation of the proposed algorithm, different ϵ^S values shall be given regarding the quality of the raw datasets. We will further analyze the performance of different ϵ^S values in Section 4.

Note that the transition trajectory may not perfectly connect to the origin/end point of other broken trajectories due to the errors of the trajectory estimation (i.e., ϵ) as illustrated in Figure 7. It can be seen that there is a gap between the transition trajectory and the broken trajectory. To connect the trajectories (trajectories i and i') considering vehicle kinematics constraints, we propose a time-space-cone-based trajectory connection method as follows.

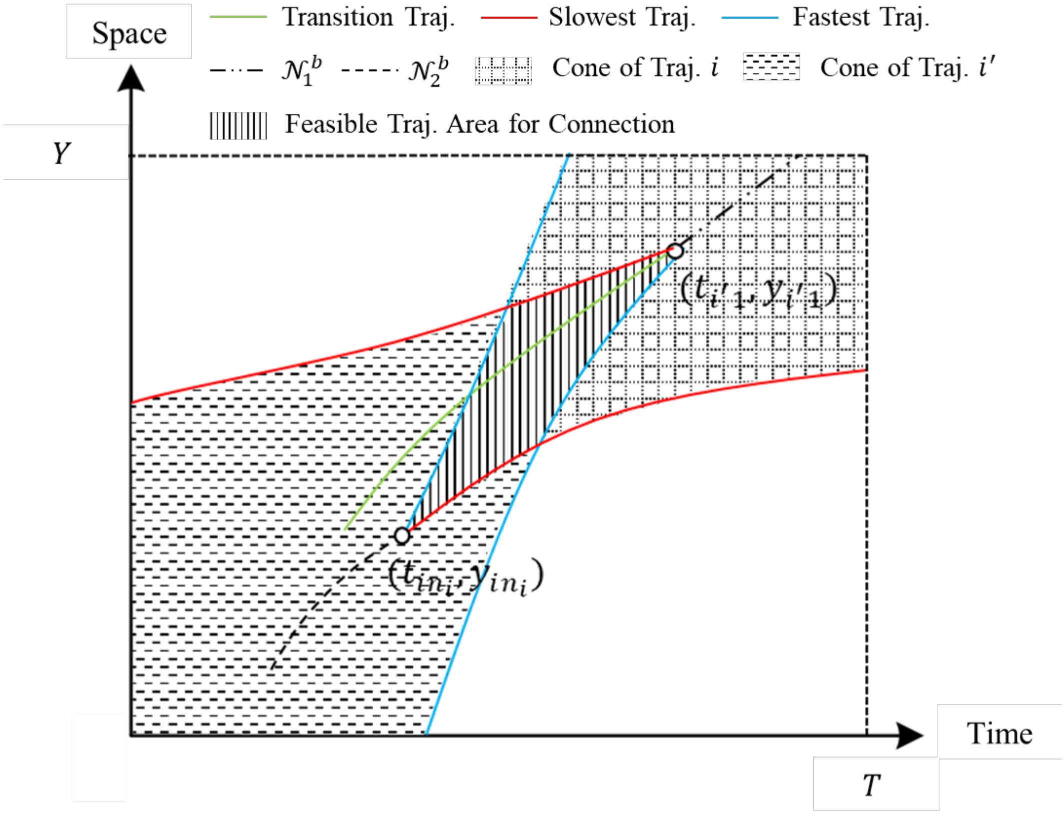


Figure 7. Time-space cone illustration.

Assume that the kinematics constraints for each vehicle are given, including the maximum speed, maximum and minimum accelerations, which are denoted as $v_{max}, a_{max}, a_{min}$ in this paper. Then for each pair of ready-to-connect trajectories $i, i' \in \mathcal{N}^b$, we can uniquely generate two boundary trajectories starting from the end (or the origin) of the trajectories, named as the slowest and fastest trajectories as illustrated in Figure 7. Each pair of the proposed boundary trajectories (a slowest trajectory and a fastest trajectory), which starts at the same point (e.g., either (t_{in_i}, y_{in_i}) or $(t_{i'1}, y_{i'1})$), covers all feasible trajectories passing the point and forms a cone-shaped area in the time-space graph. Starting from the end (or the origin) of the trajectory, the slowest trajectory is generated by operating the vehicle forward (or backward) with the minimum acceleration (i.e., $\min\{a_{min}, 0\}$) until time T (or time 0) in the time-space graph. The physical meaning of the slowest trajectory is that the slowest trajectory is a lower bound to all feasible trajectories, which suggests that all trajectories starting from the end (or the origin) of the trajectory operate faster than the slowest trajectory. Similarly, the fastest trajectory is generated by operating the vehicle forward (or backward) with the maximum acceleration (i.e., a_{max}) until the speed of the vehicle getting to the maximum speed (i.e., v_{max}). Also, the fastest trajectory is generated until the trajectory reaching time T (or time 0) in the time-space graph. The physical meaning of the fastest trajectory is that the fastest trajectory is an upper bound to all feasible trajectories, which suggests that all trajectories starting from the end (or the origin) of the trajectory operate slower than the fastest trajectory.

The equations for generating the slowest and fastest trajectories for trajectory i are shown in Equations (16)-(23). To avoid repetition, we only show the equations for trajectory i, and those of trajectory i' can be obtained easily by considering that the vehicle operates reversely in the time-space graph.

$$\mathcal{y}_i^{slow} := \left\{ y_{ik}^{slow} = \left\{ \begin{matrix} y_{in_i} + v_{in_i} \cdot \Delta - 0.5 \cdot a_{in_i} \cdot \Delta^2, k = 1 \\ y_{i(k-1)}^{slow} + v_{i(k-1)}^{slow} \cdot \Delta - 0.5 \cdot a_{i(k-1)}^{slow} \cdot \Delta^2, \forall k \in \{2,3,\cdots,(T - t_{in_i})/\Delta\} \end{matrix} \right. \right\}, \tag{16}$$

$$\mathcal{t}_i^{slow} := \{t_{ik}^{slow} = t_{in_i} + k \cdot \Delta, \forall k \in \{1,2,\cdots,(T - t_{in_i})/\Delta\}\}, \tag{17}$$

$$\mathcal{v}_i^{slow} := \left\{ v_{ik}^{slow} = \left\{ \begin{matrix} \max(0, v_{in_i} - a_{min}), k = 1 \\ \max(0, v_{i(k-1)}^{slow} - a_{min}), \forall k \in \{2,3,\cdots,(T - t_{in_i})/\Delta\} \end{matrix} \right. \right\}, \tag{18}$$

$$\mathcal{a}_i^{slow} := \left\{ a_{ik}^{slow} = \left\{ \begin{matrix} a_{min}, if\ k \le \lceil v_{in_i}/(a_{min} \cdot \Delta) \rceil \\ 0, otherwise \end{matrix} \right. \right\}, \tag{19}$$

$$\mathcal{y}_i^{fast} := \left\{ \begin{matrix} y_{in_i} + v_{in_i} \cdot \Delta + 0.5 \cdot a_{in_i} \cdot \Delta^2, k = 1 \\ y_{i(k-1)}^{fast} + v_{i(k-1)}^{fast} \cdot \Delta + 0.5 \cdot a_{i(k-1)}^{fast} \cdot \Delta^2, \forall k \in \{2,3,\cdots,(T - t_{in_i})/\Delta\} \end{matrix} \right\}, \tag{20}$$

$$\mathcal{t}_i^{fast} := \{t_{ik}^{fast} = t_{in_i} + k \cdot \Delta, \forall k \in \{1,2,\cdots,(T - t_{in_i})/\Delta\}\}, \tag{21}$$

$$\mathcal{v}_i^{fast} := \left\{ v_{ik}^{fast} = \left\{ \begin{matrix} \min(v_{max}, v_{in_i} + a_{max} \cdot \Delta \cdot k), if\ k \le \lceil (v_{max} - v_{in_i})/(a_{max} \cdot \Delta) \rceil \\ v_{max}, otherwise \end{matrix} \right. \right\}, \tag{22}$$

$$\mathcal{a}_i^{fast} := \left\{ a_{ik}^{fast} = \left\{ \begin{matrix} a_{max}, if\ k \le \lceil (v_{max} - v_{in_i})/(a_{max} \cdot \Delta) \rceil \\ 0, otherwise \end{matrix} \right. \right\}. \tag{23}$$

With these boundary trajectories, all feasible trajectories that not only satisfy the vehicle kinematics constraints but also connect the two broken trajectories are restricted into the shadow area as illustrated in Figure 7. By considering the transition trajectory that we obtained in Step 1, which we denote as the old transition trajectory (i.e., $\left\{ \begin{matrix} \mathcal{y}_i^* \\ \mathcal{t}_i^* \end{matrix} \right\}$), the new transition trajectory that can connect the two broken trajectories (trajectories i and i') is obtained by the trajectory that has the minimum location difference with the old transition trajectory in the shadow area. We denote the new transition trajectory as $\left\{ \begin{matrix} \mathcal{y}_i' \\ \mathcal{t}_i' \end{matrix} \right\}$. The equation to obtain the new transition trajectory is shown in Equations (24) and (25). The $\mathcal{v}_i'$ and $\mathcal{a}_i'$ sets of the new transition trajectory can be obtained according to Equations (24) and (25) with a given location value.

$$\mathcal{y}_i' := \left\{ y_{ik}' = \arg_{y_{ik}^{slow} \le x \le y_{ik}^{fast}} \min(|y_{ik_1}^* - x|), \forall k \in \{1,2,\cdots,(t_{i'1} - t_{in_i})/\Delta\} \right\}, \tag{24}$$

$$\mathcal{t}_i' := \left\{ t_{ik}' = t_{in_i} + k \cdot \Delta, \forall k \in \{1,2,\cdots,(t_{i'1} - t_{in_i})/\Delta\} \right\} \tag{25}$$

Figure 8 illustrates the new transition trajectory. With this, the broken trajectories i and i' are connected. The new trajectory is formed by combing the arrays of these three trajectories, such as trajectories i, new transition trajectory, and trajectory i'. By repeating these two steps, the issues we revealed previously, i.e., *detection errors* and *limited ranges*, can be successfully fixed. Nonetheless, we would like to point out one obvious limitation of the proposed algorithm. That is, the new transition trajectory is obtained without considering the acceleration variations, which suggests that the acceleration of the transition trajectory may dramatically change at the intersection point of two trajectories, e.g., the intersection point of the old transition trajectory and fastest trajectory as shown in Figure 8. This limitation can be circumvented by some trajectory smoothing techniques (Lee and Krumm, 2011), e.g., the merging operation proposed by (Li and Li, 2019) can connect two quadratic trajectories with smooth acceleration variations. However, the investigation of this technique may be out of the scope of this paper, and interested readers can refer to (Li and Li, 2019) for more details.

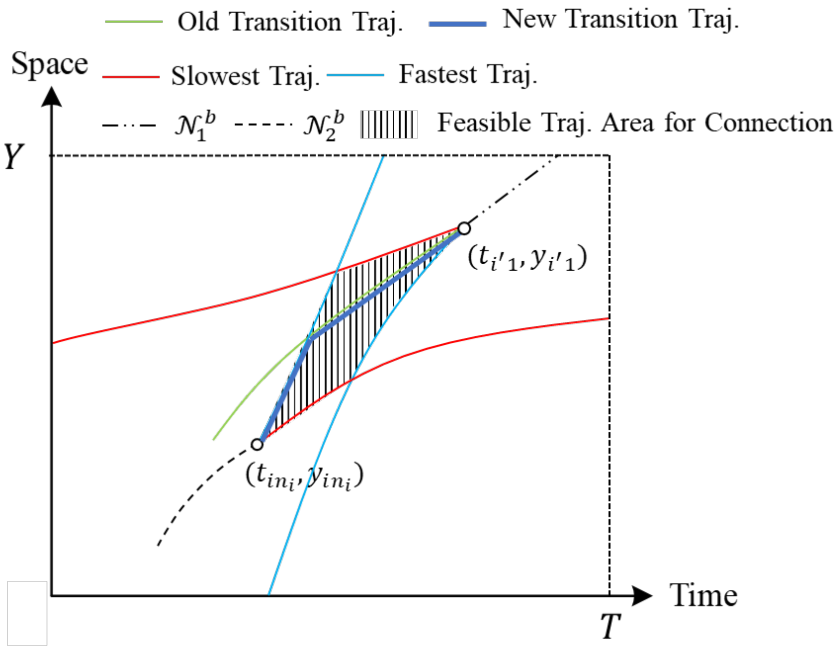
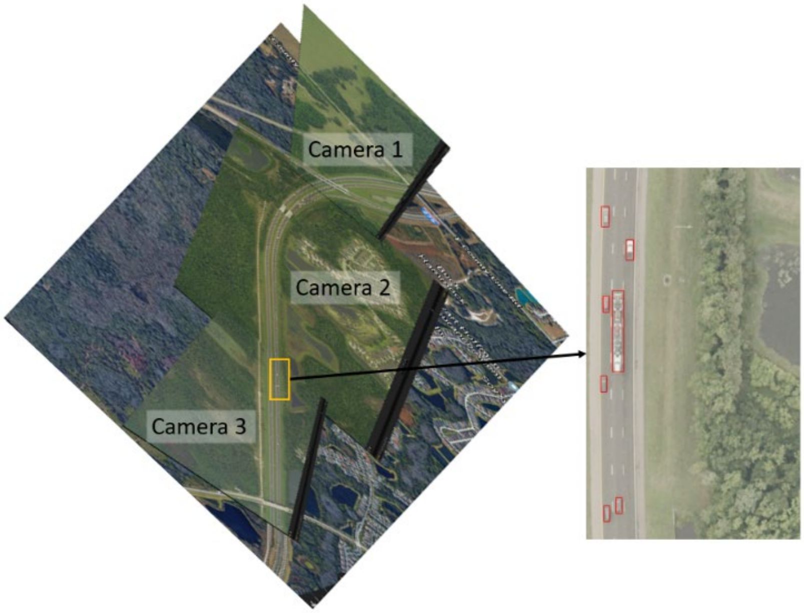


Figure 8. New transition trajectory illustration.

4. Numerical Experiment

4.1 Dataset

In the numerical experiment, we demonstrate the proposed CF-based vehicle trajectory connection method with a set of raw vehicle trajectory datasets extracted from aerial videos (Shi et al., 2021). As shown in Figure 9 (a), the aerial videos were collected by three 8K cameras on a helicopter from 4:15 – 6:15 pm on Tuesday (May 14, 2019) at 8,000 feet (2,438 meters) long segment of Interstate-75 in Florida, United States. The segment includes bi-directional traffic flow, and we only use vehicle trajectories operating from south to north (down to up in Figure 9 (a)) in this paper. The extracted trajectory datasets contain vehicle trajectories of three regular lanes for vehicle proceeding. From left to right in Figure 9 (a), the three lanes are named Lane 2, Lane 1, and Lane 0, respectively. A sample of the detected vehicles in the trajectory extraction process is shown in Figure 9 (b), in which the detected vehicles are marked with red boxes. The frequency of the extracted datasets is 30 Hz, and the format of the datasets is consistent with the NGSIM dataset for the convenience of further trajectory analysis and public use.



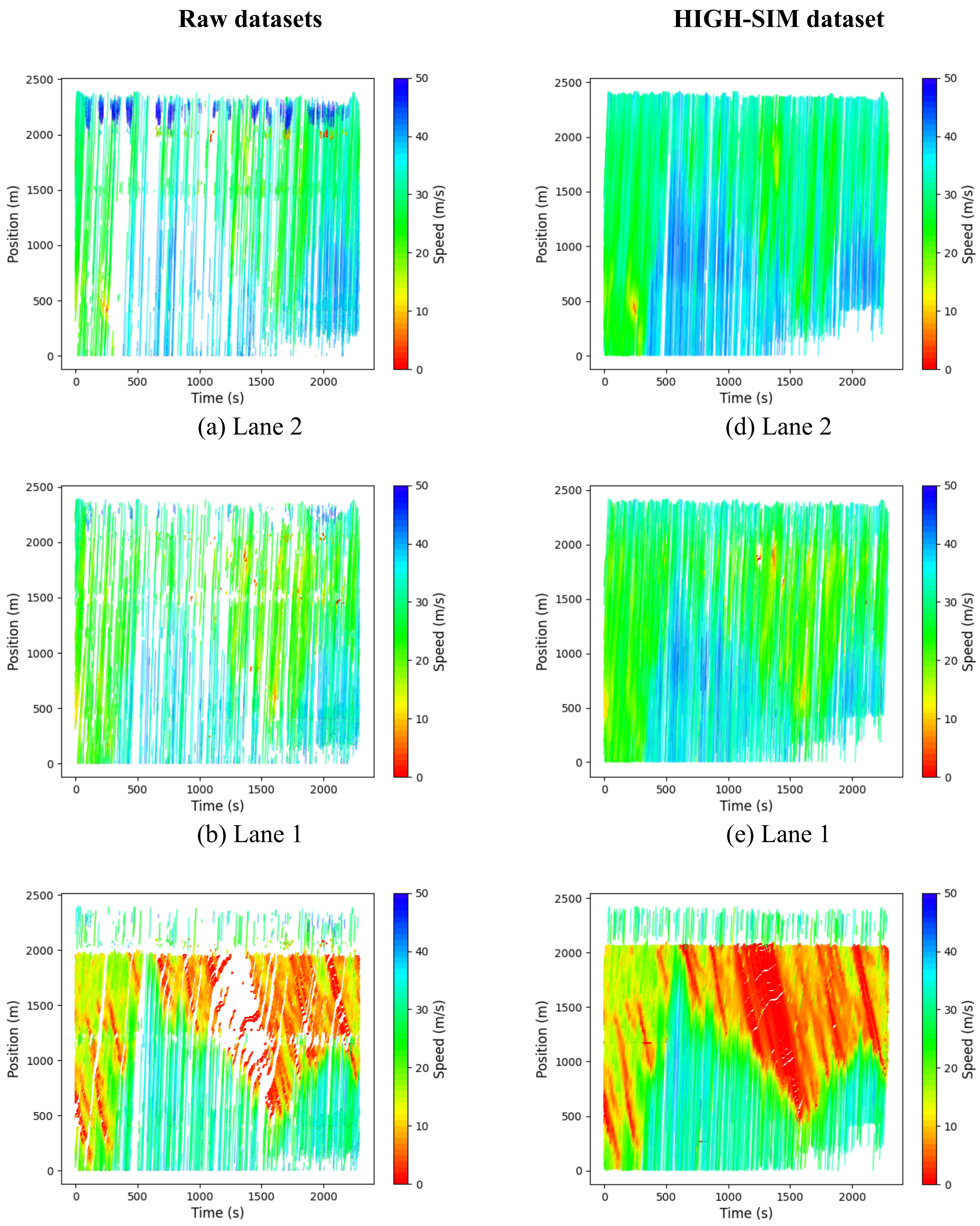
(a) (b)

Figure 9. Study area for collecting the aerial videos (Shi et al. (2021)).

4.2 Trajectory Connection Result

Before the trajectory connection, the raw datasets include 283,501 broken vehicle trajectories, which were caused due to the issues we mentioned previously, e.g., missing detections, wrong detections, etc. The speed and acceleration ranges of the trajectories in the raw datasets are [0, 150] ft/s ([0, 45.72] m/s or [0, 165] km/h) and [-20, 20] ft/s^2 ([-6.10, 6.10] m/s^2).

By using the proposed CF-based trajectory connection method to process the extracted raw datasets, 283,501 broken vehicle trajectories are eventually connected to 2,184 vehicle trajectories. The processed dataset is named as the HIGH-SIM dataset that has been published online for public use. We plot the vehicle trajectories both before and after the connection in Figure 10 to help readers understand the performance of the proposed method. It can be seen in Figure 10 (a)-(c) that the raw vehicle trajectories have some common brokens, suggesting that the trajectories are extracted from the aerial videos shot by different cameras. We can also observe that the raw vehicle trajectories are broken into a bunch of small pieces while approaching the end of the segment. This is because of the camera angle issue as we described in Figure 2. That is, as vehicles are away from the camera, the vehicle sizes on the video are gradually reduced and thus the detection rates decrease. However, after processing the raw datasets with the proposed method, most of the vehicle trajectories are successfully connected, as shown in Figure 10 (d)-(f). The results show that the HIGH-SIM dataset has more reasonable speed and acceleration distributions than a well-known trajectory dataset, the NGSIM US-101 dataset. We explicitly study the quality of the HIGH-SIM dataset in Shi et al. (2021), and interested readers can refer to it for more details.



(a) Lane 2

(d) Lane 2

(b) Lane 1

(e) Lane 1

(c) Lane 0

(f) Lane 0

Figure 10. Comparison between the raw datasets and the processed datasets.

4.3 Different Methods Comparison

We compare the performance of the proposed CF-based vehicle trajectory connection method with several benchmark methods, including Kim et al. (2019)'s method (denoted as the linear-based method), nonlinear-based method (extending the broken trajectory by a quadratic function), and Kalman filter. The testing dataset is generated by broking a set of complete vehicle trajectories of the extracted raw dataset. The raw dataset is used to prevent the external factors that potentially influence the results. For example, if we use the HIGH-SIM dataset, the vehicle trajectory connection rate by the proposed method definitely will be higher than other methods. The connection rate is denoted by $r := c^T / b^T$, where c^T is the number of connected vehicle trajectories and b^T is the number of broken vehicle trajectories. We compare the connection rate of different methods by varying the mean of the broken time length $\vartheta_1 := mean(l)$ and the variance of the broken time length $\vartheta_2 := variance(l)$ of the trajectories, where l is the time length of the trajectory segment that we deleted from each trajectory. The varying range of the mean broken time length is [0.1, 3] second and that of the variance of the broken time length is [0, 64].

Figure 11 plots the connection rate for each method with varying the mean of the broken time length. We can observe that the connection rate ranges of the studied methods are [0.9, 1], [0, 0.2], [0.2,0.5], [0.5,0.9] for the proposed method, the linear-based method, the nonlinear-based method, and Kalman filter, respectively. The proposed method outperforms all benchmark methods regarding the connection rate. One reason for this superior performance is that the proposed method incorporates a car-following model into the trajectory connection and thus well captures the driving characteristics, which provides benefits on the trajectory connection. It also can be observed that as the mean of the broken time length increases, the connection rate appears decreasing trend for each method. This indicates that the longer the time length of the missing trajectory is, the harder the trajectory connection will be. However, the connection rate of the proposed method decreases slower than those of the benchmark methods, which further supports the superior performance of the proposed method.

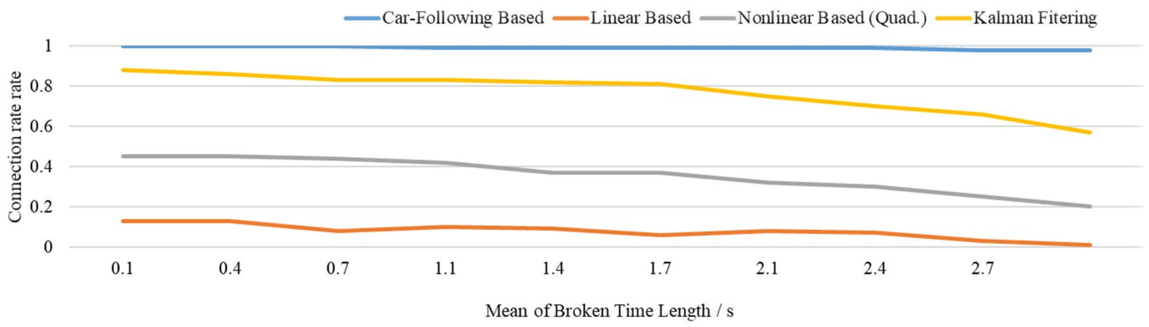


Figure 11. Connection rate comparison with varying mean broken time length.

Further, we change the variance of broken time length (i.e., ϑ_2) to study the robustness of the proposed method. The variance of the broken time length is varied from 0 to 64, and the connection rates of the methods are shown in Figure 12. We can observe that as the variance of the broken time length increases, the connection rate also appears decreasing trend for each method. This is because that a higher variance of the broken time length gives a higher probability of long broken trajectories and thus leads to a lower trajectory connection rate. Note that although the connection rate of the proposed method is degraded with a high variance of the broken time length, the overall connection rate of the proposed method still is the highest among the four methods, which varies within [0.6, 1]. This result indicates that the proposed method has relatively good robustness when dealing with different quality datasets and thus strengthens the transferability of the proposed method.

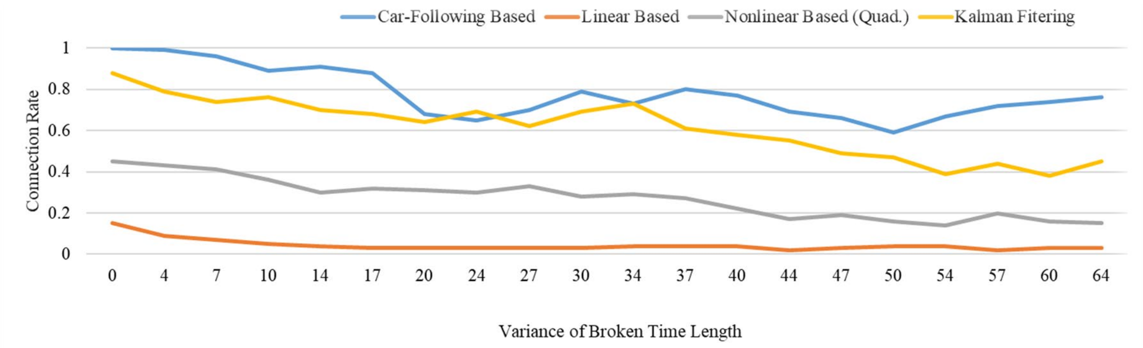
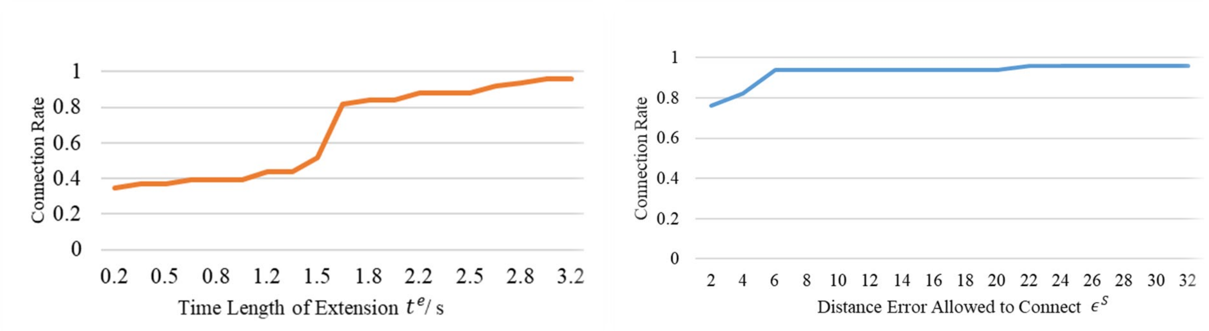


Figure 12. Connection rate comparison with varying the variance of broken time length.

Moreover, we are also interested in the effects of the two critical parameters on the connection rate of the proposed method, such as the time length of the transition trajectory (i.e., t^e) and the error term for evaluating the location difference between the transition trajectory and the broken trajectory (i.e., ϵ^S). Note that the default values of t^e and ϵ^S are set to 1.67 seconds and 5, respectively. When we vary the values of one parameter, we keep the value of the other parameter the same as the default value. Then, as shown in Figure 13 (a), by increasing the values of t^e from 0.2 to 3.2 seconds, the connection rates increase from 0.4 to 0.9. There is a significant increase in the connection rate when t^e equals to around 1.5 s, which indicates that most of the broken time length of the trajectories (around 80%) is less than 1.5 s. As can be seen in Figure 13 (b), as the values of ϵ^S increase, the connection rates increase fast at the beginning (from 2 to 6). However, when the values of ϵ^S are greater than 6, the keep increasing of the values of ϵ^S does not much impact the connection rates. This is because that for a correct trajectory connection, the calculated distance error between the two trajectories is dependent on the accuracy of the adopted car-following model. The maximum distance error for a given car-following model is a bounded value and is regardless of the dataset. Once ϵ^S equals to the maximum distance error for a correct trajectory connection, the keep increasing of the values of ϵ^S will increase the probability of a wrong connection and will not impact the connection rate that reflects the percentage of correct connections.



(a) (b)

Figure 13. Sensitivity analysis for two critical parameters (t^e and ϵ^S).

5. Conclusion

This paper proposed a car-following-based (CF-based) vehicle trajectory connection method that can connect broken vehicle trajectories based on car-following theory. The proposed method can not only fill missing data points caused by detection errors but also can help connect trajectory data from different sensors. To illustrate the performance of the proposed CF-based method, the proposed method was employed to process a series of vehicle trajectory datasets that were extracted from aerial videos recorded at several successive spaces of Interstate-75, United States. Comparing with several benchmark methods, the results showed that the proposed method has advantages in both trajectory connection accuracy and trajectory consistency aspects. The dataset processed by the proposed method, named by the HIGH-SIM dataset, was the longest vehicle trajectory dataset in the literature based on the authors' knowledge, which contains a full life cycle of the bottleneck. The dataset has been published online for public use.

Future research can be conducted in a few directions. It will be interesting to adopt some learning-based methods (e.g., machine learning, deep learning, etc.) to extend the broken trajectory for trajectory connection. Moreover, it will be interesting to validate the existing macro- and microscopic traffic characteristics with the extracted datasets.

Acknowledgment

This research is supported by the US National Science Foundation through Grants Crisp 1 and Crisp 2.

References

Anuar, K., Cetin, M., 2017. Estimating Freeway Traffic Volume Using Shockwaves and Probe Vehicle Trajectory Data. Transp. Res. Procedia 22, 183–192. https://doi.org/10.1016/j.trpro.2017.03.025

Apeltauer, J., Babinec, A., Herman, D., Apeltauer, T., 2015. Automatic vehicle trajectory extraction for traffic analysis from aerial video data. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 40, 9–15. https://doi.org/10.5194/isprsarchives-XL-3-W2-9-2015

Azevedo, C.L., Cardoso, J.L., Ben-Akiva, M., Costeira, J.P., Marques, M., 2014. Automatic Vehicle Trajectory Extraction by Aerial Remote Sensing. Procedia - Soc. Behav. Sci. 111, 849–858. https://doi.org/10.1016/j.sbspro.2014.01.119

Babinec, A., Herman, D., Cecha, S., 2014. AUTOMATIC VEHICLE TRAJECTORY EXTRACTION FOR TRAFFIC ANALYSIS FROM AERIAL VIDEO DATA.

Barmpounakis, E., Geroliminis, N., 2020. On the new era of urban traffic monitoring with massive drone data : The pNEUMA large-scale field experiment. Transp. Res. Part C 111, 50–71. https://doi.org/10.1016/j.trc.2019.11.023

Chen, X., Li, Z., Yang, Y., Qi, L., Ke, R., 2020. High-Resolution Vehicle Trajectory Extraction and Denoising From Aerial Videos. IEEE Trans. Intell. Transp. Syst. 1–13. https://doi.org/10.1109/tits.2020.3003782

Coifman, B., Wu, M., Redmill, K., Thornton, D.A., 2016. Collecting ambient vehicle trajectories from an instrumented probe vehicle: High quality data for microscopic traffic flow studies. Transp. Res. Part C Emerg. Technol. 72, 254–271. https://doi.org/10.1016/j.trc.2016.09.001

Daganzo, C.F., 1997. Fundamentals of Transportation and Traffic Operations. Fundam. Transp. Traffic Oper. https://doi.org/10.1108/9780585475301

Drew, D.R., 1968. Traffic Flow Theory and Simulation. New York, McGraw-Hill.

Jazayeri, A., Cai, H., Zheng, J.Y., Tuceryan, M., 2011. Vehicle detection and tracking in car video based on motion model, in: IEEE Transactions on Intelligent Transportation Systems. pp. 583–595. https://doi.org/10.1109/TITS.2011.2113340

Kim, E.J., Park, H.C., Ham, S.W., Kho, S.Y., Kim, D.K., Hassan, Y., 2019. Extracting Vehicle Trajectories Using Unmanned Aerial Vehicles in Congested Traffic Conditions. J. Adv. Transp. 2019. https://doi.org/10.1155/2019/9060797

Kim, Z.W., Cao, M., 2010. Evaluation of feature-based vehicle trajectory extraction algorithms. IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC 99–104. https://doi.org/10.1109/ITSC.2010.5625278

Krajewski, R., Bock, J., Kloeker, L., Eckstein, L., 2018. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 2118–2125. https://doi.org/10.1109/ITSC.2018.8569552

Li, L., Li, X., 2019. Parsimonious trajectory design of connected automated traffic. Transp. Res. Part B Methodol. 119, 1–21. https://doi.org/10.1016/j.trb.2018.11.006

Punzo, V., 2009. Estimation of vehicle trajectories from observed discrete positions and Next-Generation Simulation Program (NGSIM) data.

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. Transp. Res. Part C Emerg. Technol. 19, 1243–1262. https://doi.org/10.1016/j.trc.2010.12.007

Raju, N., Arkatkar, S., Easa, S., Joshi, G., 2021. Developing extended trajectory database for heterogeneous traffic like NGSIM database. Transp. Lett. 00, 1–10. https://doi.org/10.1080/19427867.2021.1908490

Sazara, C., Nezafat, R.V., Cetin, M., 2017. Offline reconstruction of missing vehicle trajectory data from 3D LIDAR. IEEE Intell. Veh. Symp. Proc. 792–797. https://doi.org/10.1109/IVS.2017.7995813

Tong, C., Chen, H., Xuan, Q., Yang, X., 2017. A framework for bus trajectory extraction and missing data recovery for data sampled from the internet. Sensors (Switzerland) 17. https://doi.org/10.3390/s17020342

van Hinsbergen, C.P.I.J., Schakel, W.J., Knoop, V.L., van Lint, J.W.C., Hoogendoorn, S.P., 2015. A general framework for calibrating and comparing car-following models. Transp. A Transp. Sci. https://doi.org/10.1080/23249935.2015.1006157

Victor, T., 2014. Analysis of Naturalistic Driving Study Data: Safer Glances, Driver Inattention, and Crash Risk, Analysis of Naturalistic Driving Study Data: Safer Glances, Driver Inattention, and Crash Risk. Transportation Research Board, Washington, D.C. https://doi.org/10.17226/22297

Wang, G., Xiao, D., Gu, J., 2008. Review on vehicle detection based on video for traffic surveillance, in: 2008 IEEE International Conference on Automation and Logistics. IEEE, pp. 2961–2966. https://doi.org/10.1109/ICAL.2008.4636684

Xu, Y., Yu, G., Wu, X., Wang, Y., Ma, Y., 2017. An Enhanced Viola-Jones Vehicle Detection Method from Unmanned Aerial Vehicles Imagery. IEEE Trans. Intell. Transp. Syst. 18, 1845–1856. https://doi.org/10.1109/TITS.2016.2617202

Zhang, G., Avery, R.P., Wang, Y., 2007. Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. Transp. Res. Rec. 138–147. https://doi.org/10.3141/1993-19

Zhang, T., Jin, P.J., 2019. A longitudinal scanline based vehicle trajectory reconstruction method for high-angle traffic video. Transp. Res. Part C Emerg. Technol. 103, 104–128. https://doi.org/10.1016/j.trc.2019.03.015

Zhao, D., Li, X., 2019. Real-World Trajectory Extraction from Aerial Videos - A Comprehensive and Effective Solution. 2019 IEEE Intell. Transp. Syst. Conf. ITSC 2019 2854–2859. https://doi.org/10.1109/ITSC.2019.8917175

Zhao, H., Wang, C., Lin, Y., Guillemard, F., Geronimi, S., Aioun, F., 2017. On-Road Vehicle Trajectory Collection and Scene-Based Lane Change Analysis: Part I. IEEE Trans. Intell. Transp. Syst. 18, 192–205. https://doi.org/10.1109/TITS.2016.2571726

Next Generation Simulation, 2006. Source: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

Shi, X., Zhao, D., Yao, H., Li, X., James, R., Hale, D., and Ghiasi, A., 2021. An Open Database Generation with Monte Carlo Based Lane Marker Detection and Critical Analysis of Vehicle Trajectory - High-Granularity Highway Simulation (HIGH-SIM). Preprint: https://doi.org/10.13140/RG.2.2.30725.06887

Wu, M., 2018. Collecting Ambient Vehicle Trajectories from an Instrumented Probe Vehicle and Fusing with Loop Detector Actuations. Dissertation. The Ohio State University.

Lee WC., Krumm J., 2011. Trajectory Preprocessing. In: Zheng Y., Zhou X. (eds) Computing with Spatial Trajectories. Springer, New York, NY. https://doi.org/10.1007/978-1-4614-1629-6_1

Rahman, M., 2013. Application of Parameter Estimation and Calibration Method for Car-Following Models. Clemson University.