Communications in Transportation Research 1 (2021) 100014

Contents lists available at ScienceDirect

Communications in Transportation Research

journal homepage: www.journals.elsevier.com/communications-in-transportation-research

Video-based trajectory extraction with deep learning for High-Granularity Highway Simulation (HIGH-SIM)

Xiaowei Shi [a], Dongfang Zhao [a], Handong Yao [a], Xiaopeng Li [a,*], David K. Hale [b], Amir Ghiasi [b]

[a] Department of Civil and Environmental Engineering, University of South Florida, Tampa, FL, 33620, United States
[b] Leidos, Inc, Saxton Transportation Operation Laboratory, McLean, 22101, Virginia, United States

ARTICLE INFO

Keywords:
Video analytics
Image processing
Vehicle trajectory extraction
Deep learning
Microsimulation

ABSTRACT

High-granularity vehicle trajectory data can help researchers develop traffic simulation models, understand traffic flow characteristics, and thus propose insightful strategies for road traffic management. This paper proposes a novel vehicle trajectory extraction method that can extract high-granularity vehicle trajectories from aerial videos. The proposed method includes video calibration, vehicle detection and tracking, lane marking identification, and vehicle motion characteristics calculation. In particular, the authors propose a Monte-Carlo-based lane marking identification approach to identify each vehicle's lane. This is a challenging problem for vehicle trajectory extraction, especially when the aerial videos are taken from a high altitude. The authors applied the proposed method to extract vehicle trajectories from several high-resolution aerial videos recorded from helicopters. The extracted dataset is named by the High-Granularity Highway Simulation (HIGH-SIM) vehicle trajectory dataset. To demonstrate the effectiveness of the proposed method and understand the quality of the HIGH-SIM dataset, we compared the HIGH-SIM dataset with a well-known dataset, the NGSIM US-101 dataset, regarding the accuracy and consistency aspects. The comparison results showed that the HIGH-SIM dataset has more reasonable speed and acceleration distributions than the NGSIM US-101 dataset. Also, the internal and platoon consistencies of the HIGH-SIM dataset give lower errors compared to the NGSIM US-101 dataset. To benefit future research, the authors have published the HIGH-SIM dataset online for public use.

1. Introduction

Naturalistic vehicle trajectories have significant values in studying various traffic phenomena, such as car-following (Pei et al., 2016) and lane-changing behaviors (Li et al., 2021; Soleimaniamiri et al., 2020; Wang et al., 2019), traffic oscillation propagation (Li et al., 2012), and traffic capacity drops (Shi and Li, 2021a). As specified in Zhao and Li (2019), existing vehicle trajectory datasets can be classified into four categories, such as lidar-based trajectory datasets (Coifman et al., 2016; Zhao et al., 2017), radar-based trajectory datasets (Victor, 2014), GPS-based trajectory datasets (Milanés and Shladover, 2014; Shi and Li, 2021b), and aerial video-based trajectory datasets (Babinec et al., 2014; Kim et al., 2019; Xu et al., 2017). Due to the emergence of unmanned aerial vehicle (UAV) technology, which facilitates the collection of flexible, economical, and unbiased aerial videos (Kim and Cao, 2010), the investigation of aerial video-based trajectory datasets has attracted wide attention from researchers in both industry and academia (Apeltauer et al., 2015; Azevedo et al., 2014; Ke et al., 2019; Kim et al., 2019; Xu et al., 2017).

In the literature, well-known aerial video-based trajectory datasets include the Next Generation Simulation dataset (Federal Highway Administration [FHWA], 2006), the HighD dataset (Krajewski et al., 2018), and the pNEUMA dataset (Barmpounakis and Geroliminis, 2020). For the NGSIM dataset, FHWA extracted the vehicle trajectory data from videos taken by multiple digital video cameras installed at different locations near the freeway segments of interest (Kim and Malik, 2003). A feature-based vehicle detection algorithm detected vehicles in these videos, and a zero-mean cross-correlation matching algorithm tracked the detected vehicles (Kim et al., 2009). Researchers identified lane markings manually to provide lane numbers to the vehicles. However, based on the result analysis of the dataset, the process failed to successfully detect more than 10 percent of the vehicles, and vehicle tracking could fail for several consecutive frames. Because of the vehicle detection errors, after accounting for vehicle length, the trajectories in the dataset often overrun their leaders, seemingly resulting in "collisions of trajectories". Moreover, many researchers revealed that the speed and

* Corresponding author.
E-mail address: xiaopengli@usf.edu (X. Li).

https://doi.org/10.1016/j.commtr.2021.100014
Received 10 November 2021; Received in revised form 16 November 2021; Accepted 16 November 2021
Available online 11 December 2021

acceleration of the trajectories in the dataset often exhibit unrealistic results (Coifman and Li, 2017; Montanino and Punzo, 2015; Punzo et al., 2011). Obviously, these errors degrade the effectiveness of using the NGSIM dataset to study and validate various traffic phenomena.

In comparison with the NGSIM dataset, the HighD dataset (Krajewski et al., 2018) collected by the Ika team in the RWTH Aachen University provides a more accurate highway vehicle trajectory dataset. They adopted the U-Net (Çiçek et al., 2016), a common neural network architecture, to detect and track vehicles in aerial videos. UAVs hovering next to the German highways recorded the aerial videos and captured traffic from a bird's-eye view of the road sections. Results showed that the HighD dataset has higher detection accuracy than the NGSIM dataset, because of the advanced detection algorithm and high video resolution. Despite the success of the HighD dataset, the trajectory extraction method used in the HighD dataset might not always be applicable to videos shot at a high altitude (e.g., from a helicopter) where small onboard cameras' rotations or vibrations would cause drastic shifts of vehicles across the video frames. Moreover, lane markings were also identified manually, which will be tedious and unreliable when the video is long, and the coverage range is wide. Lane marking identification is critical for vehicle trajectory extraction because vehicles' lane numbers are calculated based on the locations of the lanes. The wrong classification of a vehicle's lane number may easily cause the "collisions of trajectories" issue. Most existing lane marking identification algorithms were proposed for automated driving (Chen and Wang, 2006; Kreucher et al., 1998; Lee and Moon, 2018; Yim and Oh, 2003; Zhao et al., 2017). The lane marking identification for automated driving is completely different from the investigated problem. For example, for the investigated problem, the background of aerial videos is almost static but that of the automated driving is dynamic. Also, since the videos are shot at a high altitude, the investigated problem has a much higher requirement on the accuracy of the lane marking identification. Due to these reasons, the algorithms for automated driving cannot be utilized by the investigated problem. Thus, there is still a need to propose an efficient lane marking identification method to cluster vehicle trajectories while extracting the trajectories from aerial videos.

Unlike the two aforementioned highway datasets, researchers collected the pNEUMA dataset (Barmpounakis and Geroliminis, 2020) over multiple days, using UAVs hovering over the Athens central business district. The UAVs recorded traffic streams in a congested 1.3 km^2 area with more than 100 km-lanes of the road network and close to half a million trajectories, which allow the deep investigation of several critical traffic phenomena. Despite the contribution of the study, we would like to state that the vehicle trajectories in the pNEUMA dataset are not temporally continuous. That is, the flight time of the UAVs is constrained by the battery capacity, and the aerial videos were missed while the UAVs were changing batteries and returning to the hovering point. Further, the investigated road facilities are mostly urban arterials with short blocks. Thus, the recorded aerial videos may not capture the comprehensive life cycle of certain traffic phenomena, e.g., bottleneck development and dissipation on freeways.

To fill out the aforementioned gaps, this paper proposes an advanced vehicle trajectory extraction method that integrates a Monte-Carlo-based lane marking identification approach. The paper proposes a multi-point feature-matching-based video calibration algorithm to circumvent the camera rotation and shifting issues. Then, the proposed method utilizes an efficient object detection deep learning neural network YOLOv3 to detect vehicles, and a feature-based vehicle tracking method to track vehicles according to their features and vehicle kinematics. The authors applied the proposed method to extract vehicle trajectories from several high-resolution aerial videos recorded by helicopters. Collaborating with a car-following-based trajectory connection method (Shi et al., 2021), we generated a 2-h 30fps vehicle trajectory dataset with 8000 ft coverage, which is named the High-Granularity Highway Simulation (HIGH-SIM) vehicle trajectory dataset. To demonstrate the effectiveness of the proposed method and understand the quality of the HIGH-SIM dataset, we compared the HIGH-SIM dataset with a well-known dataset, the NGSIM US-101 dataset, regarding the accuracy and consistency aspects. The result showed that the HIGH-SIM dataset has more reasonable speed and acceleration distributions than the NGSIM US-101 dataset. Also, the internal and platoon consistencies of the HIGH-SIM dataset give lower errors compared to the NGSIM US-101 dataset. To benefit future research, the authors published the generated HIGH-SIM dataset online. Researchers can find the data sharing link at both the Federal Highway Administration, U.S. Department of Transportation (https://highways.dot.gov/), and at the GitHub page of Connected and Autonomous Transportation Systems Lab, University of South Florida (https://github.com/CATS-Lab-USF). Further, the authors encapsulated all code files into a package named the Video-Based Intelligent Road Traffic Universal Analysis Tool (VIRTUAL), which is available for download at https://github.com/CATS-Lab-USF.

The disposition of this paper is as follows. Section 2 describes the proposed vehicle trajectory extraction method, including the video calibration, vehicle detection and tracking, Monte-Carlo-based lane marking identification algorithm, and vehicle motion characteristics calculation. Also, we describe the criteria for analyzing the extracted trajectories. Section 3 explains where we extracted the HIGH-SIM dataset and compares the quality of the HIGH-SIM dataset with the NGSIM US-101 dataset. Section 4 concludes the paper and discusses future research directions.

2. Methodology

2.1. Vehicle trajectory extraction method

Fig. 1 shows the steps of the proposed vehicle trajectory extraction method. With the given aerial videos and camera parameters (camera height, resolution, and angle), Step (a) calibrates all frames in the video, and calculates the rotation and shifting parameters. With the rotation and shifting parameters, Step (a) obtains the aerial videos with the static background. Then, Step (b) identifies and tracks vehicles in each frame. To identify vehicles' lane numbers, we propose a Monte-Carlo-based lane marking identification approach in Step (c). With the vehicles' positions (obtained by Step (b)) and lane numbers (obtained by Step (c)), Step (d) calculates the vehicles' motion characteristics at each time point, including location, speed, and acceleration. In the end, the method outputs the extracted trajectory dataset.

2.1.1. Step (a): camera rotation and shifting correction

Since disturbances such as camera shake, rotation, and shifting will cause the drift of a reference point across frames, finding rotation and shifting variations between any two consecutive frames are critical for vehicle trajectory extraction. To calculate the rotation and shifting parameters between the two consecutive frames, we applied the ORB feature matching algorithm (Rublee et al., 2011) to obtain the perspective transformation.

According to the ORB feature matching algorithm, we firstly extract the FAST features (Rosten and Drummond, 2006) of the two consecutive frames, e.g., frames f_i, f_{i+1}, where i and $i+1$ are the indices of the frames, as shown in Fig. 2. We define S_i, S_{i+1} as the sets of feature points extracted from frames f_i, f_{i+1}, respectively. With the extracted features, the BRIEF descriptors (Calonder et al., 2010) can be calculated based on the detected features. We define B_i, B_{i+1} as the BRIEF descriptors of the set of feature point S_i, S_{i+1}. With the BRIEF descriptors of the two frames, the perspective transformation is obtained with the RANSAC algorithm (Fischler and Bolles, 1981) as described in OpenCV (https://docs.opencv.org). We define $M_{i,i+1}$ as the perspective transformation from frames f_{i+1} to f_i. Therefore, the transformation between the i_{th} frame f_i to the very first frame f_1 will be $M_{1,i} = \prod_{j=1}^{i-1} M_{j,j+1}$, where we let $M_{1,1} = I$. Thus, the j th point $P_{j,i}$ in frame f_i can be matched to frame f_1 with the transformation matrix: $P_{j,1} = M_{1,i} * P_{j,i}$, where we define $P_{j,1}$ as the

position of the pixel point $P_{j,i}$ in frame f_1.

With the ORB feature matching algorithm, we can form a relatively static background with the perspective transformation $M_{1,i}$, $i \in [1, I]$. However, since the perspective transformation matrix is calculated with the multiplication of the perspectives of every two consecutive frames, there will be cumulative errors as the frame number i increases. Therefore, we applied a lane detection and tracking algorithm in Step (b) to further correct the cumulative error.

2.1.2. Step (b): vehicle detection and tracking

To identify and track vehicles in each frame in Step (a) efficiently and correctly, we propose to locally train the YOLOv3 (Redmon and Farhadi, 2018) and apply the model to detect vehicles in each frame of the aerial videos. To train the YOLOv3 locally, we generate a training dataset from the aerial video using a background extraction algorithm. We first apply the Gaussian mixture-based background/foreground segmentation algorithm (KaewTraKulPong and Bowden, 2002) to extract the background and foreground of each frame. Since the vehicles are moving across the frames, the contours of the vehicles will be extracted by the algorithm with noise. Therefore, we match the contours in two consecutive frames by their distances and sizes. Otherwise, we will discard the contours if no feasible matching of the contours is found. In this manner, a portion of vehicles in the video will be detected. We extract the contours of these detected vehicles and create training data for vehicles. Afterward, we train the YOLOv3 with the training data. With the locally trained YOLOv3, we are able to detect vehicles in the video accurately.

2.1.3. Step (c) lane marking identification

In Step (c), we propose a Monte-Carlo-based lane marking identification method to help position vehicles and find lane numbers of vehicles. The proposed method is described in Algorithm 1. The algorithm takes advantage of the distribution properties of lane markings, i.e., the lane markings of different lanes are parallel, and the distance of any two connected lane markings on the same lane is consistent. The algorithm assumes that the locations of lane markings in images are the perspective transformation of their locations on any other images. We first detect lane marking locations loc_i with a locally trained YOLOv3 detector D: $l_i[j] = D(f_i, j)$, where $l_i[j]$ is the detected location of lane marking j in frame f_i, $i \in [0, N]$. We then select a subset of indices $r \in R \subset \{1, \ldots, j, \ldots, J\}$ for the locations of the detected lane markings. With an initial lane marking locations l^* distributed following the mentioned distribution properties and the perspective transformation matrix calculation function T, which is introduced in OpenCV (https://docs.opencv.org/3.4/d7/dff/tutorial_feature_homography.html), we are able to calculate the perspective transformation matrix $M_i = T(l^*[r], l_i[r]), r \in R$ with index r. The perspective transformation matrix M_i is then used to calculate the expected locations of all lane markings in frame f_i, $l_i' = M_i l_m$. The sum of the Euclidean distance $S = \sum_{j \in J} \left|\left| l_{i,j} - l'_{i,j} \right|\right|$ for all expected locations l_i' and detected locations loc_i is calculated to estimate the correctness of the perceptive transformation matrix. The random selection of index r will be repeated several times to find the optimal perceptive transformation matrix. With the optimal perceptive transformation matrix, the expected lane marking location can be calculated by $l_i^* = M_i^* l^*$.

Algorithm 1. Monte-Carlo-based lane marking identification method

Algorithm 1 Monte-Carlo-based lane marking identification method

Required: Frames $f_i, i \in [1, N]$, detector D: $l_i[j] = D(f_i, j)$, the initial distribution of lane markings l_m, perceptive transformation matrix calculation $M_i = T(l^*, l_i)$, perceptive transformation M_i, optimal perceptive transformation M_i^*, threshold ρ, random index selected $r \in R$, a large number n^{big}

Ensure: detection rate of detector $p(D) \geq 90\%$

for each frame $f_i, i \in [1, N]$ **do**
  $S_i^* = n^{big}, M_i^* = I, loc_i^*$
  for $k \in [0, M]$ **do**
    $J = [0, |l_i|]$
    for $j \in J$ **do**
      $l_i[j] = D(f_i, j)$
    end for
    $r \in R$
    $M_i = T(l^*[r], l_i[r])$
    $l_i' = M_i * l^*$
    $S = \sum_{j \in J} ||l_i'[j] - l_i[j]||$
    if $S < S_i^*$ **then**
      $S_i^* = S, M_i^* = M_i, l_i^* = l_i'$
    end if
    if $S < \rho$ **then**
      break
    end if
  end for
end for

2.1.4. Step (d) vehicle motion characteristics calculation

With the detected vehicles (Step (b)) and lane markings (Step (c)), the vehicle positions can be calculated. Denote the position of vehicle j in frame i as p_{ij}. With the perspective transformation matrix M_i obtained from Step (c), the vehicle position in the static background can be calculated as $p_{ij}^S = M_i * M(\theta_i^A, S_i^A) * p_{ij}$, where $M(\theta, S)$ is the transform ma-

trix with rotation parameter θ and shifting parameter S. With the lane marking positions, the lane label can be calculated throughout the image with a polynomial regression method. We define $L_k = \{(x_0^k, y_0^k), \ldots, (x_N^k, y_N^k)\}$ as the lane markings' positions of lane $k \in [1, K]$ in an image with K lanes. As shown in Algorithm 2, by applying a polynomial regression method to the location array (https://en.wikipedia.org/wiki/-Polynomial_regression), we can obtain a polynomial function $x^k = f_k^P(y^k)$. Considering a position of a vehicle $p_{ij}^S = (x_{ij}^S, y_{ij}^S)$, the lane number of the position can be determined by the regressed lane marking function f_k^P. With the vehicle positions, the speed and acceleration of the vehicle trajectories can be calculated as well.

Algorithm 2. Lane number calculation based on the detected lane marking

Algorithm 2 Lane number calculation based on the detected lane marking

Required: lane marking positions of lane k, $L_k = \{(x_0^k, y_0^k), \ldots, (x_N^k, y_N^k)\}, k \in [1, K]$
vehicle j's position in image i $p_{ij}^S, i \in [0, I], j \in [0, J]$
polynomial regression function β: $[\boldsymbol{X}, \boldsymbol{Y}]$
lane number of vehicle position $L(x_{ij}^S) = -1$
Ensure: $|L_k| > 3$ for polynomial regression
1: **for** each lane $k \in [1, K]$ **do**
2: calculate polynomial regression parameter β
obtain function $x^k = f_k^P(y_k)$
3: **End for**
4: **for** each lane $k \in [1, K]$ **do**
5: $\hat{x}_{ij}^k = f_k^P(y_{ij}^k)$
6: **if** $x_{ij}^S < \hat{x}_{ij}^k$ **do**
7: $L(x_{ij}^S) = k$
8: **end if**
9: **end for**
Return lane number of vehicle position $L(x_{ij}^S) == -1 ? K : L(x_{ij}^S)$

2.2. Trajectory analysis method

To evaluate the performance of the proposed vehicle trajectory extraction method, we adopt the trajectory accuracy analysis method proposed by Punzo et al. (2011). The trajectories are analyzed from the following aspects.

2.2.1. Internal consistency

The internal consistency analysis aims to check the consistency of the differentiation of vehicle trajectory with its speed and acceleration. The internal consistency of space, ε_t^S, is calculated as:

$$\varepsilon_t^S = \hat{s}_t - \left(\hat{s}_0 + \int_0^t \hat{v}_t dt \right),$$

where $\hat{s}_t$ is the observed location of a vehicle and $\hat{v}_t$ is the observed speed of the vehicle at time t. Similarly, the internal consistency of speed, ε_t^V, is calculated as:

$$\varepsilon_t^V = \hat{v}_t - \left(\hat{v}_t + \int_0^t \hat{a}_t dt \right).$$

where $\hat{a}_t$ is the observed acceleration of a vehicle at time t.

2.2.2. Platoon consistency

The platoon consistency is adopted to estimate the consistency of trajectories of vehicle pairs in vehicle trajectory data. The platoon consistency of space can be calculated as:

$$\varepsilon_{npt}^{PS} := (\hat{s}_{n0} - \hat{s}_{p0}) + \left(\int_0^t \hat{v}_{nt} dt - \int_0^t \hat{v}_{pt} dt \right),$$

where n, p denote the ID of the subject vehicle and its following vehicle, respectively. Similarly, the speed platoon consistency can be calculated as:

$$\varepsilon_{npt}^{PV} := (\hat{v}_{n0} - \hat{v}_{p0}) + \left(\int_0^t \hat{a}_{nt} dt - \int_0^t \hat{a}_{pt} dt \right).$$

The measurement of the bias ($\varepsilon : \varepsilon_t^S, \varepsilon_t^V, \varepsilon_{npt}^{PS}, \varepsilon_{npt}^{PV}$) in a vehicle trajec-

Table 1
Format of the HIGH-SIM dataset.

Column Name	Explanation
Vehicle ID	ID number for each vehicle
Global Time	Time in seconds from 12:00:00 am of the day
Frame ID	Frame number in the corresponding video
Local X (ft)	Position in the direction perpendicular to the road
Local Y (ft)	Position in the direction along the road
Global X (Longitude)	Vehicle's GPS longitude location
Global Y (Latitude)	Vehicle's GPS latitude location
Width (ft)	Vehicle width
Length (ft)	Vehicle length
Class (1 motor; 2 auto; 3 truck)	Vehicle class
Speed (ft/s)	Vehicle speed
Acceleration (ft/s^2)	Vehicle acceleration
Lane Num	Lane number
Space Highway (ft)	Distance between this vehicle's front bumper to its following vehicle's front bumper

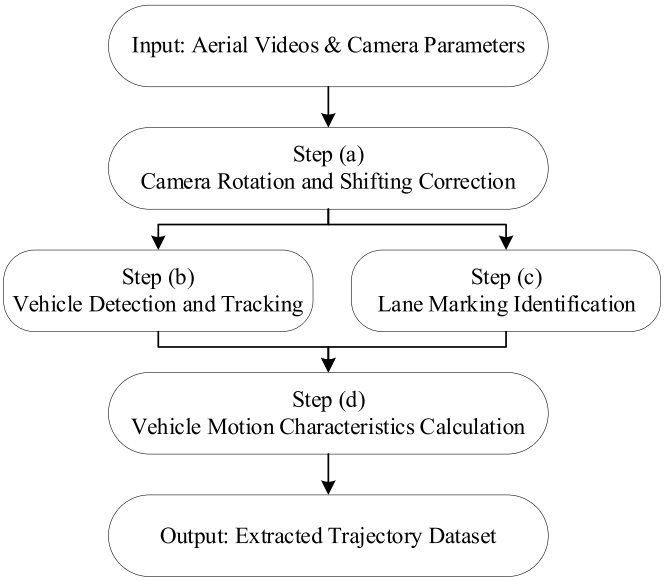


Fig. 1. Vehicle trajectory extraction method.

tory data is summarized as follows: (1) the minimum bias: $\min(\varepsilon)$; (2) the maximum bias: $\max(\varepsilon)$; (3) the mean bias: $mean(\varepsilon) = \sum \varepsilon / N$; (4) root mean square error (RMSE): $\sqrt{\sum (\varepsilon^2)/N}$.

3. HIGH-SIM dataset

3.1. Experiment settings

In the experiments, we applied the proposed vehicle trajectory extraction method to a series of aerial videos. As shown in Fig. 3, the aerial video data were collected by three 8K cameras in a helicopter from 4:15–6:15 p.m. on Tuesday (May 14, 2019) over an 8000 ft long segment of the I-75 freeway in Florida (28°08′37.2″N 82°22′58.8″W to 28°10′16.2″N 82°23′38.0″W), United States. A set of detected vehicles are highlighted in Fig. 3 in red (in the Web version) rectangle boxes. Note that the proposed method extracted the trajectories from different aerial videos sequentially. To connect the extracted trajectories from different aerial videos and thus generate a long-coverage dataset, we adopted a car-following-based trajectory connection method proposed in Shi et al. (2021). The method successfully connected the trajectories extracted from the aerial videos shot by different cameras. The newly generated long-coverage dataset is named the HIGH-SIM dataset. The dataset contains vehicle trajectories from three regular lanes and one off-ramp with a frequency of 30 Hz. The format of the HIGH-SIM dataset is shown in Table 1, which is consistent with the NGSIM dataset for the convenience of further trajectory analysis and future public use. The trajectories are plotted in Fig. 4. As shown in Fig. 4, there is traffic congestion on lane 0 and the off-ramp. In comparison, the trajectories on lane 1 and lane 2 are much smoother.

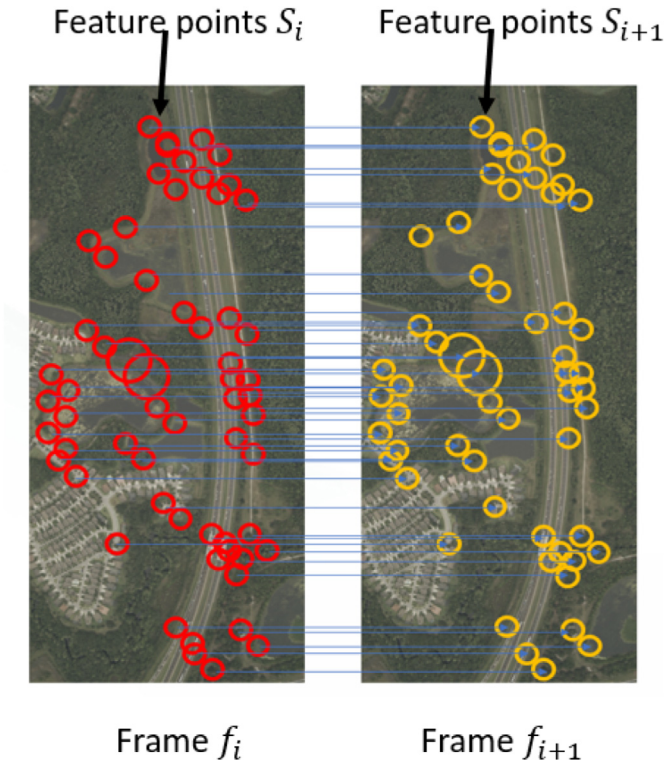


Fig. 2. ORB feature matching algorithm.

3.2. HIGH-SIM dataset analysis

To illustrate the effectiveness of the proposed method and help readers understand the quality of the HIGH-SIM dataset, we compare the HIGH-SIM dataset with a well-known publicly available dataset, the NGSIM US-101 dataset, which was collected on southbound US 101 in Los Angeles, CA. According to Punzo et al. (2011), the NGSIM US-101 dataset is the one with the best quality among all NGSIM datasets. The study area includes five mainline lanes and an auxiliary lane between the on-ramp at Ventura Boulevard and the off-ramp at Cahuenga Boulevard. The duration of the dataset is 45 min, which is separated into three 15-min periods, including 7:50 a.m. to 8:05 a.m., 8:05 a.m. to 8:20 a.m., and 8:20 a.m. to 8:35 a.m. The data set covers a road segment of 2100 feet. The data record frequency is 10 Hz per frame.

3.2.1. Speed and acceleration distributions

We first compare the distributions of speed and acceleration within the two datasets. As shown in Fig. 5, the distribution of the speed of the HIGH-SIM dataset is more uniform than the NGSIM US-101 dataset. There are mainly three flat speed peaks (or modes) in the HIGH-SIM dataset, e.g., 0 ft/s, 25 ft/s, 100 ft/s, which correspond to the traffic jam, traffic congestion, and free flow scenarios plotted in Fig. 4. In comparison, there is only one speed peak in the NGSIM US-101 dataset at 35 ft/s, which indicates that the NGSIM US-101 dataset contains relatively homogeneous traffic scenarios. Also, the speed range of the HIGH-SIM dataset is [0, 150] ft/s ([0, 45.72] m/s or [0, 165] km/h), which is significantly wider than that of the NGSIM US-101 dataset, as shown in Fig. 5.

The distributions of the acceleration of the two datasets are shown in Fig. 6. The acceleration of the HIGH-SIM dataset is in the range of [−20, 20] ft/s^2 ([−6.10, 6.10] m/s^2), and the acceleration of the NGSIM US-101 dataset is in the range of [−12.5, 12.5] ft/s^2 ([−3.81, 3.81] m/s^2). We can observe that the HIGH-SIM dataset also has a wider acceleration range than that of the NGSIM US-101 dataset. One possible reason for this result is that the HIGH-SIM dataset contains the trajectories with extreme operating conditions, e.g., emergency acceleration and deceleration. The incorporation of the trajectories with these extreme operating conditions definitely can provide more hints for researchers to study the characteristics of the traffic flow. The other reason for this wider acceleration range may be because the adopted car-following-based trajectory connection algorithm (Shi et al., 2021). We connect two broken trajectories by considering vehicle kinematics constraints, e.g., maximum and minimum acceleration. It is possible that the optimal transition trajectory for connecting the two broken trajectories is the trajectory with the maximum or minimum acceleration. Interested readers can refer to Section 3.2 in Shi et al. (2021) for more details about

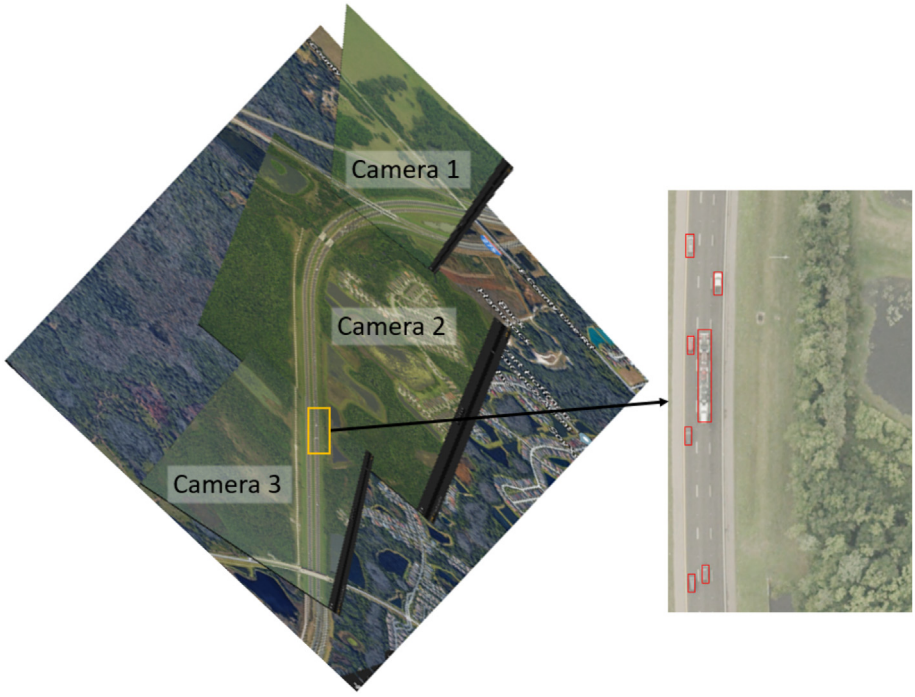


Fig. 3. The study area of the HIGH-SIM dataset.

the adopted trajectory connection algorithm.

3.2.2. Internal and platoon consistencies

We also compare the two datasets in terms of internal and platoon consistencies (Punzo et al., 2011). Table 2 shows the results of the space/speed internal consistency, and Table 3 shows the results of the speed/acceleration internal consistency of the two datasets, respectively.

We can observe that the maximum bias, minimum bias, and mean bias for the HIGH-SIM dataset are close to zero, which are much lower than those of the NGSIM US-101 dataset. In particular, the percentage of bias greater than 1 for both the internal space/speed (shown in Table 2) and speed/acceleration consistencies (shown in Table 3) of the HIGH-SIM dataset are 0, while those of the NGSIM US-101 dataset are 29.2% and 65.2%, respectively. Also, the HIGH-SIM dataset has a much lower value in terms of the RMSE than the NGSIM US-101 dataset. The superior performance of the HIGH-SIM dataset on these criteria indicates that the HIGH-SIM dataset has much higher internal space/speed and speed/acceleration consistencies than the NGSIM US-101 dataset.

The platoon consistencies of the space/speed and speed/acceleration of the HIGH-SIM and NGSIM US-101 datasets are shown in Table 4 and Table 5, respectively. The results are similar to the results of the internal consistencies analysis. That is, the HIGH-SIM dataset has much lower values on the maximum bias, minimum bias, mean bias, percentage of bias greater than 1, and RMSE than the NGSIM US-101 dataset.

Despite the fairly good performance of the HIGH-SIM dataset on both the internal and platoon consistencies, we would like to point out one flaw of the dataset. That is, due to the wrong detections and classifications, there still are some unreasonable trajectories in the HIGH-SIM dataset. As shown in Table 4, the number of vehicle pairs with negative inter-vehicle spacing for the HIGH-SIM dataset is 1232. However, due to a large number of vehicle pairs in the HIGH-SIM dataset (i.e., 7,057,678 vehicle pairs), the HIGH-SIM dataset has a much lower percentage of vehicle pairs with negative inter-vehicle spacing comparing to the NGSIM US-101 dataset, as shown in Table 4.

Overall, based on the superior performance of the HIGH-SIM dataset in terms of these criteria, we can conclude that the HIGH-SIM dataset outperforms the NGSIM US-101 dataset regarding internal and platoon consistencies. Thus, the HIGH-SIM dataset has a much higher data quality than the NGSIM US-101 dataset, which is expected to provide a more accurate representation of real traffic and thus facilities future traffic flow-related studies.

4. Conclusions

In this paper, we proposed an advanced vehicle trajectory extraction system to extract longer and more accurate vehicle trajectory data from aerial videos. The proposed system applied YOLOv3 for vehicle detection and a feature matching method for vehicle tracking. In addition, a novel Monte-Carlo-based lane marking identification method was proposed to track lane markings across frames. By implementing the proposed system, we developed a vehicle trajectory tool called VIRTUAL and extracted a new and long trajectory dataset – HIGH-SIM. We analyzed the quality of HIGH-SIM and compared the internal and platoon consistency of HIGH-SIM and the NGSIM US-101 dataset. Results show that the new dataset HIGH-SIM provides much higher quality trajectory data. It not only indicates the efficiency of the proposed vehicle extraction method, but also provides a high-quality trajectory dataset for trajectory-related research activities. The HIGH-SIM dataset can help the theoretical development of new behavioral models, estimate model parameters, and validate them.

The proposed trajectory extraction method can be generally applied to extracted vehicle trajectories from aerial videos. The process will help build maps of areas of interest with higher accuracy. The accuracy of object detection and location extraction can be improved by applying

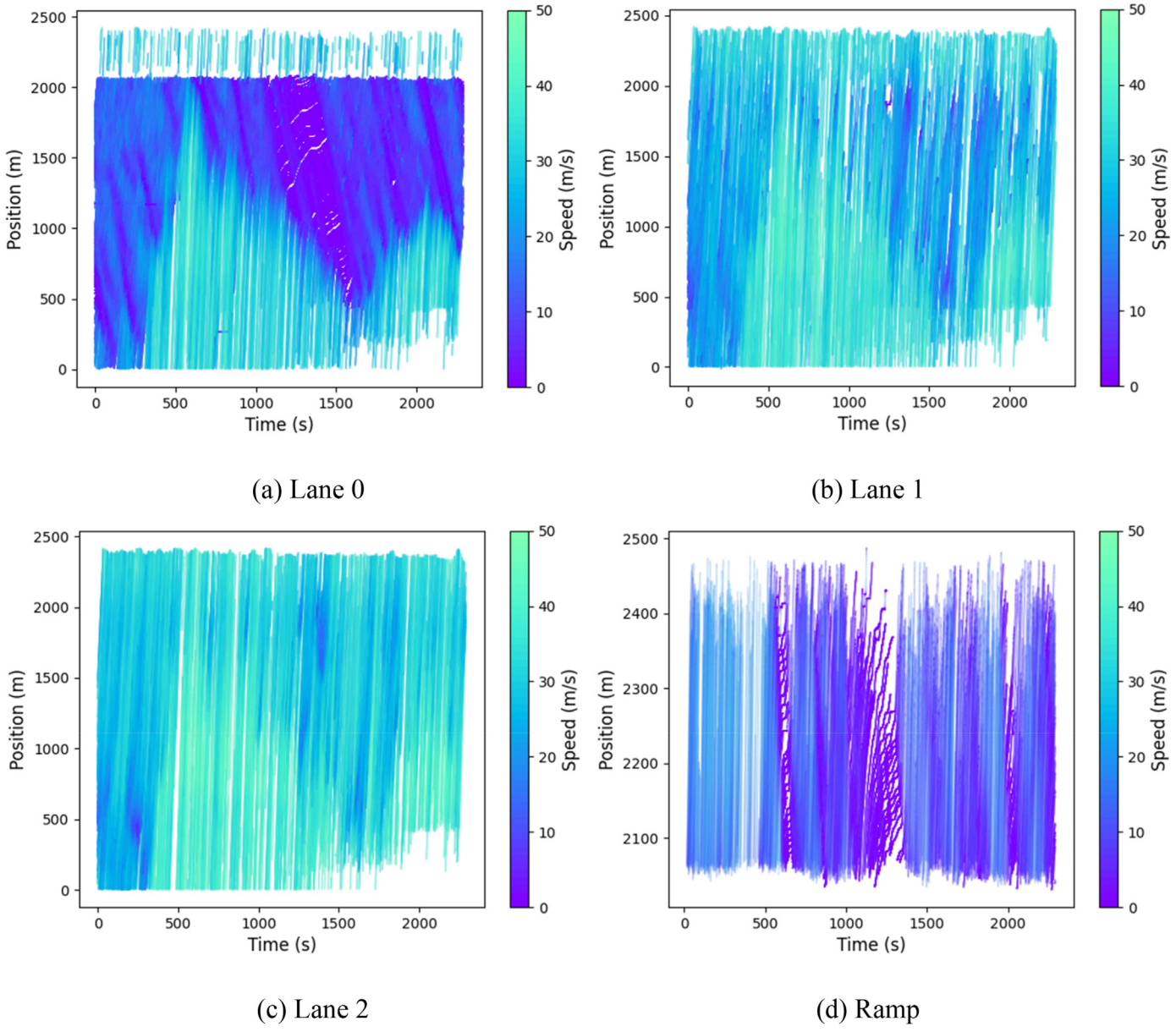


Fig. 4. Examples of vehicle trajectories in the HIGH-SIM dataset.

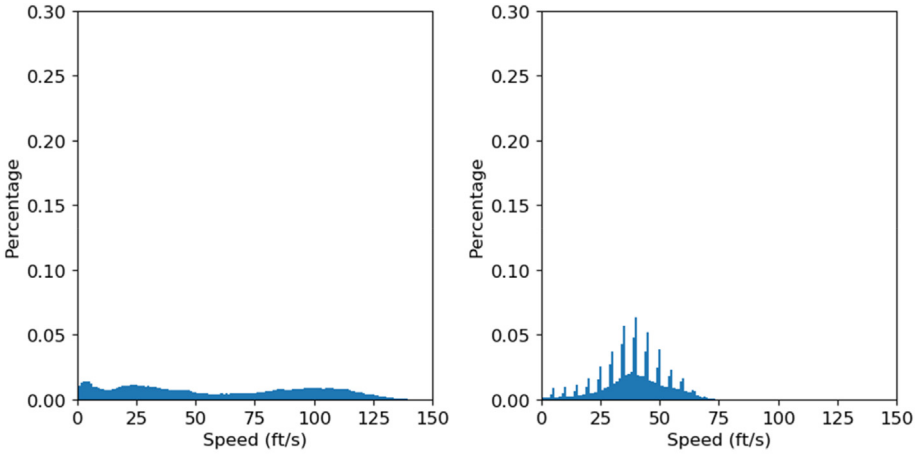
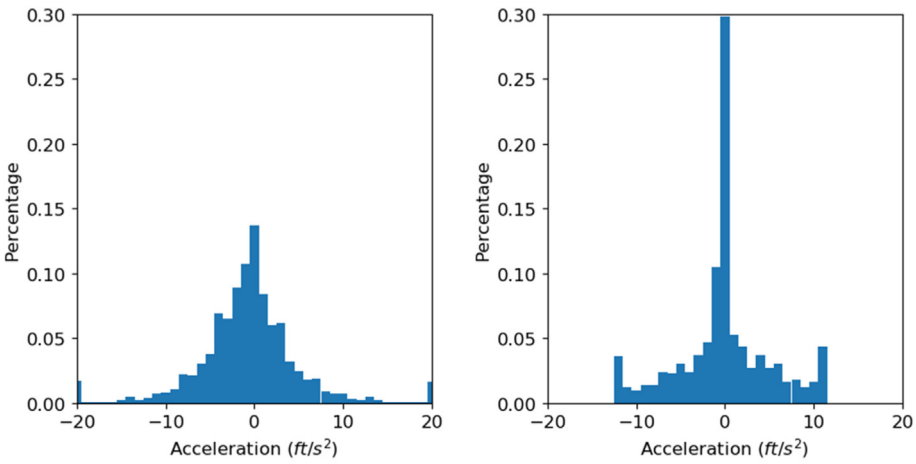


Fig. 5. Distributions of the speed of the HIGH-SIM and NGSIM US-101 datasets.



(a) HIGH-SIM (b) NGSIM US-101

Fig. 6. Distributions of the acceleration of the HIGH-SIM and NGSIM US-101 datasets.

Table 2
Internal space/speed consistency of HIGH-SIM and NGSIM US-101 datasets.

	HIGH-SIM	NGSIM US-101
Maximum bias max(ε^S)	0.06	6.63
Minimum bias min(ε^S)	−0.06	−28.22
Mean bias *mean*(ε^S)	5.76e-5	−0.01
Percentage of bias greater than 1 (ft) ($P_{\varepsilon^S>1}(\%)$)	0.00	29.20
RMSE of bias (*RMSE*(ε^S))	0.0018	0.36

Table 3
Internal speed/acceleration consistency of HIGH-SIM and NGSIM 101 datasets.

	HIGH-SIM	NGSIM US-101
Maximum bias max(ε^V)	0.01	56.56
Minimum bias min(ε^V)	−0.001	−63.75
Mean bias *mean*(ε^V)	7.68e-6	0.30
Percentage of bias greater than 1 (ft/s) ($P_{\varepsilon^V>1}(\%)$)	0.00	65.20
RMSE of bias (*RMSE*(ε^V))	0.002	3.12

Table 4
Platoon space/speed consistency of HIGH-SIM and NGSIM 101 datasets.

	HIGH-SIM	NGSIM US-101
Maximum bias max(ε^{PS})	0.01	11.25
Minimum bias min(ε^{PS})	−0.06	−4.40
Mean bias *mean*(ε^{PS})	1.19e-6	−0.03
Percentage of bias greater than 1 (ft) ($P_{\varepsilon^{PS}>1}(\%)$)	0.0	9.93
RMSE of bias (*RMSE*(ε^{PS}))	7.96e-4	0.23
Total number of vehicle pairs	7,057,678	985,552
Number of vehicle pairs with negative inter-vehicle spacing	1232	1438
% of vehicle pairs with negative inter-vehicle spacing	1.74e-4	0.0014

Table 5
Platoon speed/acceleration consistency of the HIGH-SIM and NGSIM US-101 datasets.

	HIGH-SIM	NGSIM US-101
Maximum bias max(ε^{PV})	1.33	88.28
Minimum bias min(ε^{PV})	−0.87	−88.80
Mean bias *mean*(ε^{PV})	−2.39e-4	0.04
Percentage of bias greater than 1 (ft/s) ($P_{\varepsilon^{PV}>1}(\%)$)	1.45e-4	65.88
RMSE of bias ($RMSE(\varepsilon^{PV})$)	0.11	4.51

RGBD cameras to obtain aerial videos. With the depth dimension of each detected object, the detection and location errors caused by shadows can be eliminated.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported in part by the United States National Science Foundation Grant #1932452 and Federal Highway Administration Grant #DTFH6116D00030. The review effort and helpful comments from Rachel James (FHWA) are acknowledged with great gratitude. We also thank Dr. Zhenyu Wang (University of South Florida) for the suggestion to apply the YOLOv3 object detection network in the proposed trajectory extraction system.

References

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.

Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. BRIEF: Binary Robust Independent Elementary Features, pp. 778–792. https://doi.org/10.1007/978-3-642-15561-1_56.

Chen, Q., Wang, H., 2006. A real-time lane detection algorithm based on a hyperbola-pair model. IEEE Intell. Veh. Symp. Proc. https://doi.org/10.1109/ivs.2006.1689679.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3D U-net: learning dense volumetric segmentation from sparse annotation. Lect. Notes Comput. Sci. https://doi.org/10.1007/978-3-319-46723-8_49.

Coifman, B., Li, L., 2017. A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset. Transp. Res. Part B Methodol. 105, 362–377. https://doi.org/10.1016/j.trb.2017.09.018.

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. Transport. Res. C Emerg. Technol. 72, 254–271. https://doi.org/10.1016/j.trc.2016.09.001.

FHWA, 2006. The Next Generation Simulation.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM. https://doi.org/10.1145/358669.358692.

KaewTraKulPong, P., Bowden, R., 2002. An improved adaptive background mixture model for real-time tracking with shadow detection. Video-Based Surveill. Syst. https://doi.org/10.1007/978-1-4615-0913-4_11.

Ke, R., Li, Z., Tang, J., Pan, Z., Wang, Y., 2019. Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow. IEEE Trans. Intell. Transport. Syst. https://doi.org/10.1109/TITS.2018.2797697.

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. Transport. https://doi.org/10.1155/2019/9060797, 2019.

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

Kim, Z.W., Gomes, G., Hranac, R., Skabardonis, A., 2009. A machine vision system for generating vehicle trajectories over extended freeway segments. In: Intell. Transp. Soc. Am. - 12th World Congr. Intell. Transp. Syst. 2005.

Kim, Z.W., Malik, J., 2003. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. Proc. IEEE Int. Conf. Comput. Vis. https://doi.org/10.1109/iccv.2003.1238392.

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.

Kreucher, C., Lakshmanan, S., Kluge, K., 1998. A driver warning system based on the LOIS lane detection algorithm. IEEE Int. Conf. Intell. Veh.

Lee, C., Moon, J.H., 2018. Robust lane detection and tracking for real-time applications. IEEE Trans. Intell. Transport. Syst. https://doi.org/10.1109/TITS.2018.2791572.

Li, Q., Li, X., Mannering, F., 2021. Assessment of discretionary lane-changing decisions using a random parameters approach with heterogeneity in means and variances. Transp. Res. Rec. J. Transp. Res. Board. https://doi.org/10.1177/0361198121992364.

Li, X., Wang, X., Ouyang, Y., 2012. Prediction and field validation of traffic oscillation propagation under nonlinear car-following laws. Transp. Res. Part B Methodol. 46, 409–423. https://doi.org/10.1016/j.trb.2011.11.003.

Milanés, V., Shladover, S.E., 2014. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. Transport. Res. C Emerg. Technol. 48, 285–300. https://doi.org/10.1016/j.trc.2014.09.001.

Montanino, M., Punzo, V., 2015. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. Transp. Res. Part B Methodol. https://doi.org/10.1016/j.trb.2015.06.010.

Pei, X., Pan, Y., Wang, H., Wong, S.C., Choi, K., 2016. Empirical evidence and stability analysis of the linear car-following model with gamma-distributed memory effect. Phys. A Stat. Mech. Appl. 449, 311–323. https://doi.org/10.1016/j.physa.2015.12.104.

Punzo, V., Borzacchiello, M.T., Ciuffo, B., 2011. On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data. Transport. Res. C Emerg. Technol. 19, 1243–1262. https://doi.org/10.1016/j.trc.2010.12.007.

Redmon, J., Farhadi, A., 2018. YOLOv3: an Incremental Improvement.

Rosten, E., Drummond, T., 2006. Machine learning for high-speed corner detection. Lect. Notes Comput. Sci. https://doi.org/10.1007/11744023_34.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544.

Shi, X., Li, X., 2021a. Constructing a fundamental diagram for traffic flow with automated vehicles: methodology and demonstration. Transp. Res. Part B Methodol. 150, 279–292. https://doi.org/10.1016/j.trb.2021.06.011.

Shi, X., Li, X., 2021b. Empirical study on car-following characteristics of commercial automated vehicles with different headway settings. Transport. Res. C Emerg. Technol. 128, 103134. https://doi.org/10.1016/j.trc.2021.103134.

Shi, X., Zhao, D., Li, X., 2021. A Car-Following-Based Method for Vehicle Trajectory Connection.

Soleimaniamiri, S., Shi, X., Li, X., Shaw), Hu, Y., 2020. Incorporating Mixed Automated Vehicle Traffic in Capacity Analysis and System Planning Decisions.

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, Z., Shi, X., Li, X., 2019. Review of lane-changing maneuvers of connected and automated vehicles: models, algorithms and traffic impact analyses. J. Indian Inst. Sci. https://doi.org/10.1007/s41745-019-00127-7.

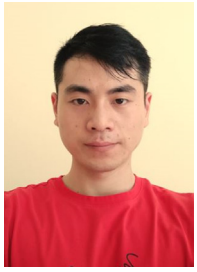
Xu, Y., Yu, G., Wu, X., Wang, Y., Ma, Y., 2017. An enhanced Viola-Jones vehicle detection method from unmanned aerial vehicles imagery. IEEE Trans. Intell. Transport. Syst. 18, 1845–1856. https://doi.org/10.1109/TITS.2016.2617202.

Yim, Y.U., Oh, S.Y., 2003. Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving. IEEE Trans. Intell. Transport. Syst. https://doi.org/10.1109/TITS.2003.821339.

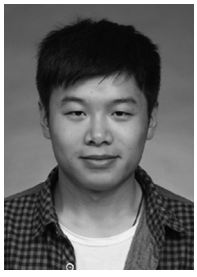
Zhao, D., Li, X., 2019. Real-world trajectory extraction from aerial videos - a comprehensive and effective solution. In: 2019 IEEE Intell. Transp. Syst. Conf. ITSC, pp. 2854–2859. https://doi.org/10.1109/ITSC.2019.8917175, 2019.

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. Transport. Syst. 18, 192–205. https://doi.org/10.1109/TITS.2016.2571726.

Dr. Xiaowei (Tom) Shi received his Bachelor and Master degree from Beijing Jiaotong University, Beijing, China in 2015 and 2018. He received his second Master degree and Ph.D. degree from the University of South Florida, Tampa, Florida, the United States in 2020 and 2021. His main research interests focus on the designs and applications of emerging technologies on future transportation systems (e.g., connected, automated, modular, and electric vehicles).



Dr. Dongfang Zhao received his B.S. in physics and B.E. in computer science from University of Scisnce and Technology of China (2016) and Ph. D in Civil Engineering in the University of South Florida (2021). He is working on car sharing fleet management and facility design models and vehicle trajectory data collection projects.



Dr. Handong Yao is a Postdoctoral Research Associate in the Department of Civil and Environmental Engineering at University of South Florida. He received his Ph.D. and M.S. degree in transportation engineering from Harbin Institute of Technology, Harbin, China. His main research interest is machine-learning based data analysis, connected and autonomous vehicle trajectory optimization modeling.

Dr. Xiaopeng (Shaw) Li received a B.S. degree (2006) in civil engineering with a computer engineering minor from Tsinghua University, China, a M.S. degree (2007) and a Ph.D. (2011) degree in civil engineering along with a M.S. degree (2010) in applied mathematics from the University of Illinois at Urban-Champaign, USA. He is an associate professor and Susan A. Bracken Faculty Fellow (first holder) in the Department of Civil and Environmental Engineering at the University of South Florida (USF). He is also the director for one USDOT national university transportation center, National Institute for Congestion Reduction (NICR). His research interests include automated connected, electric and shared (ACES) vehicles, interdependent infrastructure systems, operations research & artificial intelligence.

Dr. David Hale is a senior transportation engineer and certified Project Management Professional for Leidos, Inc. His 27-year career has focused on urban traffic mobility research and development. He has delivered 50 traffic analysis workshops across 22 U.S. states plus Australia, 85 conference presentations, and 66 peer-reviewed publications. Since joining Leidos in 2014, he has served as a principal investigator and/or project manager for 27 FHWA projects. He is a member of the Transportation Research Board (TRB) standing committee on traffic simulation (ACP80). His service includes contributions to the TRB Joint Simulation Subcommittee, Highway Capacity Committee, Traffic Signal Systems Committee, and Institute of Traffic Engineers (ITE) Traffic Engineering Council Executive Committee.

Dr. Amir Ghiasi has over 8 years of successful experience in traffic engineering and transportation planning, and has published tens of research papers in the most prestigious journals and conference proceedings, which received a few hundred citations. He has conducted several research projects at the Federal Highway Administration Saxton Transportation Operations Laboratory, Tuner Fairbank Highway Research Center. He is currently a Principal Investigator for two CAV research projects that aim to enable cooperative driving automation (CDA) to interact with infrastructure to enhance infrastructure efficiency and ultimately improve transportation system safety and efficiency and reduce traffic congestion. Additionally, he has extensive experience in traffic modeling and simulation, calibration and validation.