Measuring trajectories and fuel consumption in oscillatory traffic: experimental results

F. Wu, [*] R. Stern, [*] M. Churchill,[*] M. L. Delle Monache,[†] K. Han,[‡] B. Piccoli,[§] D. B. Work[¶]

Abstract

This article presents data collected through a set of experiments with nine to 10 vehicles driving on a ring road constructed on a closed track. Vehicle trajectory data is extracted via a series of vision processing algorithms (for background subtraction, vehicle identification, and trajectory extraction) from a 360-degree panoramic camera placed at the center of the ring. The resulting trajectory data is smoothed via a two-step algorithm which applies a combination of RLOESS smoothing and regularized differentiation to produce consistent position, velocity, and acceleration data that does not exhibit unrealistic accelerations common in raw trajectory data extracted from video. A subset of the vehicles also record real-time fuel consumption data of the vehicles using OBD-II scanners.

The tests include both smooth and oscillatory traffic conditions, which are useful for constructing and calibrating microscopic models, as well as fuel consumption estimates from these models. The results show a an increase in fuel consumption in the experiments in which traffic oscillations are observed as compared to experiments where vehicles maintain a smooth flow. However, this is partially due to the higher average speed at which vehicles travel in the experiments in which oscillatory traffic is observed.

The article contains a complete, publicly available dataset including the video data, the extracted trajectories, the smoothed trajectories, and the OBD-II logs from each equipped vehicle. In addition to the dataset, this article also contains a complete source code for each step of the data processing. It is the first of several experiments planned to collect detailed trajectory data and fuel consumption data with smooth and unsteady traffic flow in a controlled experimental environment.

[*]Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign

[†]Department of Mathematics, Rutgers University - Camden and Inria, Univ. Grenoble Alpes, CNRS, GIPSA-lab, F-38000 Grenoble, France

[‡]Department of Civil and Environmental Engineering, Imperial College, London, UK)

[§]Department of Mathematics, Rutgers University - Camden

[¶]Department of Civil and Environmental Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, (dbwork@illinois.edu)

1 Introduction

1.1 Motivation and Related Work

Traffic instabilities such as stop-and-go waves and speed oscillations are a common phenomenon in dense freeway traffic. Such waves, often referred to as "phantom jams" have been shown to arise in uniformly-flowing traffic, even in the absence of geometric bottlenecks [1]. The development of traffic models that exhibit instabilities such as oscillatory traffic has been of considerable interest in the past few decades [2, 3, 4, 5, 6]. Much of this focus has been on microscopic car-following models, where the trajectory of each vehicle is evolved over time. Often, the evolution equation of each vehicle takes the form of an ordinary differential equation [7]. Despite the great interest in modeling traffic instabilities, there are relatively few high-quality trajectory datasets that exhibit oscillatory traffic and traffic waves. This is largely due to the difficulty in collecting such datasets. Consequently, calibration of microscopic models is hindered by the lack of abundant trajectory data.

One of the most extensive and widely used trajectory datasets is the *Next Generation Simulation* (NGSIM) [8] dataset, which was collected "in support of traffic simulation with a primary focus on microscopic modeling." The dataset was collected using video cameras on a 900m segment of freeway I-80 in California over 30 minutes in December of 2003. In April of 2005, three additional 15 minute duration data collection experiments were performed on a 500m segment of the same highway. Later in 2005, a 640m segment of US-101 in Los Angeles was instrumented, and three consecutive 15 minute datasets were recorded. The NGSIM dataset remains an extremely valuable tool for the micro simulation community, however one must analyze the data with care. As pointed out in several recent works [9, 10, 11], the velocity and acceleration data are prone to large errors caused by the application of a finite difference calculation on the vehicle position. If properly smoothed, the data can be used to calibrate realistic acceleration behaviours and for estimating fuel consumption in congested traffic (see Treiber et al. [12] and Piccoli et al. [13]). The primary limitations of the NGSIM dataset are that it is limited in spatial and temporal scope, and it lacks measured fuel consumption data.

In addition to the field data collection efforts on freeways in California, a pioneering experiment to demonstrate the generation of traffic instabilities that arise from uniformly-flowing traffic was conducted by by Sugiyama et al. [1] in 2008. The experiment took place on a single-lane circular track 74m in diameter. The researchers instructed a number of drivers (22-23 vehicles) to drive around the track at speeds of approximately 30 km/h. In the experiment, the initially-uniform traffic flow devolves into a traffic wave that travels back at approximately 20km/h. The video footage from the experiment is widely used to motivate the need to model stop and go dynamics in microscopic traffic flow models. Datasets containing the position estimates of all vehicles during an approximately four minute test with 22 vehicles, and an eight minute test with 23 vehicles are available from the project website at: `http://traffic.phys.cs.is.nagoya-u.ac.jp`. Unfortunately the dataset does not contain fuel consumption data which is an increasing area of concern at the intersection of transportation engineering and the environment.

Outside of the NGSIM and Sugiyama et al. experiments, other notable data collection efforts have also been executed. For example, the Mobile Century dataset [14] contains detailed GPS trajectory data from a small subset of vehicles for several hours on a freeway in California, but the coarseness of the GPS position data and the lack of trajectory data from any non-GPS-equipped vehicles limits its use for microscopic traffic modeling. At Virginia Tech, the *Center for Data Reduction and Analysis Support* has collected and curated several petabytes of data on a broad range of safety and naturalistic driving topics, including data obtained via an on-board data acquisition system as part of the *Second Strategic Highway Research Program* (SHRP 2) *Naturalistic Driving Study* (NDS). Because the dataset contains human subjects data, the majority of the data is not publicly available for download. Moreover, like the Mobile Century dataset, because not all vehicles in the traffic stream are equipped with the data collection devices, queries about the onset of oscillatory traffic are difficult to answer from this dataset alone.

1.2 Outline and Contributions

In line with the data collection efforts described above, the aim of this paper is to provide an experimental dataset that *i*) exhibits the development of traffic instabilities from uniform traffic, and *ii*) contains vehicle performance data such as fuel consumption as recorded by the OBD-II data from a subset of vehicles, alongside the smoothed trajectory data obtained from video processing and trajectory smoothing. The published dataset with this article contains the raw video, the raw trajectory data extracted from the video, smoothed trajectory data suitable for microscopic model calibration, and fuel consumption data as recorded by *on-board diagnostics II* (OBD-II) devices. A set of four tests (three to five minutes) with nine and 10 vehicles on a 30m diameter track are published with this article. The dataset may contribute to improved fuel consumption modeling in the presence of traffic waves. To this end, the article also provides a complete set of algorithms to extract vehicle-level trajectory information from video data collected using a panoramic 360-degree video camera on a circular track. For the purpose of this article, we consider oscillatory traffic to be traffic in which visible traffic waves are present. The source code for each algorithm applied to the dataset is also available with this article. This work is preliminary, and more tests are planned to increase the number of trajectory datasets available with fuel consumption data.

The remainder of the article is organized as follows. We present a method for conducting a ring-road experiment to verify the presence traffic instabilities in uniform traffic in the *Methodology* section. Specifically, in a subsection on *Video processing for trajectory reconstruction*, we introduce a vision pipeline (i.e., a collection of off the shelf algorithms applied to our specific problem) for tracking vehicles using k-means clustering and *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). In the subsection on *Smoothing methods*, we discuss how to smooth the vehicle trajectories such that they are consistent with respect to differentiation and integration while mitigating the error growth due to differentiation of the positional data. In the next subsection on *Vehicle statistics from OBD-II scanners*, we explain how we instrument vehicles to record real-time fuel consumption data. In the section on *Application of methods to ring road data*, we apply the methods to an experimental dataset collected on June 29, 2016 on a closed track ring road in Champaign, IL, designed in the same spirit of the Sugiyama et al. [1] experiment. Finally in the last section the conclusions and our future plans for larger and more comprehensive experiments are detailed.

2 Methodology

2.1 Overview

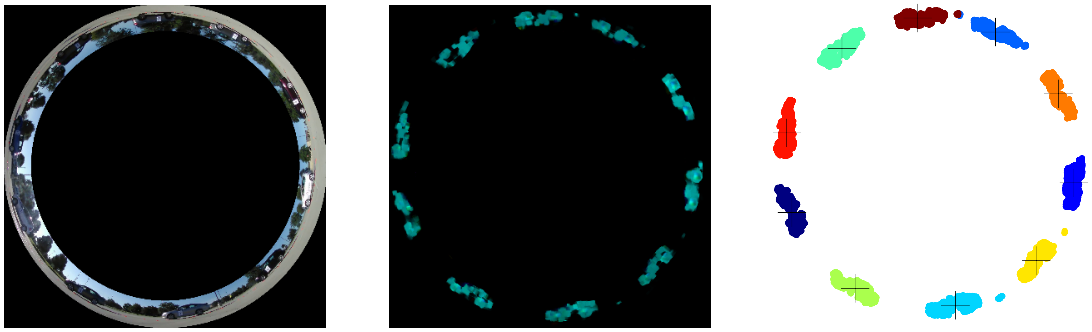
A panoramic camera and several OBD-II scanners are used to reconstruct the vehicle trajectories and record the fuel consumption of the vehicles. The panoramic camera used in the experiment is a 360 degree camera manufactured by VSN Mobil with a resolution of 3840×640 and records at 30 frames per second. The OBD-II scanners used in the experiments are the OBDLink LX model. The panoramic camera is placed at the center of the track, while the OBD-II scanners are installed in the vehicles, and are connected to various Android devices to log the OBD-II data stream. The panoramic camera is chosen because of the ease of implementation and its potential to provide high fidelity trajectory data. Next, the algorithms used to process the video data and extract trajectories are briefly described.

2.2 Video processing for trajectory reconstruction

Vehicle trajectories are reconstructed from the panoramic video by means of background subtraction and image segmentation. Roughly, background subtraction is the process of identifying for each pixel if it is part of the background image, or part of one of the vehicles to be tracked. For pixels that are identified as belonging to a vehicle to be tracked, image segmentation is the process of determining to which vehicle each foreground pixel belongs. Background subtraction and image segmentation are applied frame by frame throughout the video data. A final step to construct trajectories is to assign pixel clusters in consecutive frames as belonging to the same vehicle, thereby connecting the vehicle positions over time. In this work, background subtraction is achieved through *dense optical flow* [15] and image segmentation through k-means clustering. The assignment of clusters across each frame to vehicle trajectories is achieved through the application of DBSCAN. This process is described in more detail below.

To separate vehicle pixels from the background in the background subtraction step, dense optical flow [15] is applied. Dense optical flow is a computer vision algorithm that estimates the apparent motion of pixels in two consecutive frames. Dense optical flow works well when the following assumptions are valid: *i*) the object appearance remains constant across adjacent frames; *ii*) the motion of nearby pixels are similar (e.g., objects to be tracked occupy more than a single pixel); *iii*) points do not move very fast in the image scene (i.e., the object can be found in the same general area in the next frame). When the above assumptions hold true, moving objects are identified at a pixel by pixel level, and the relative motion of each pixel is reported. It is assumed that moving objects correspond to vehicles, and non-moving objects are considered part of the background scene.

Background subtraction via optical flow has limitations that must be addressed when applying it to extract vehicle trajectory data. For example, the algorithm will fail when there are objects in the background that are also moving (e.g., traffic on adjacent roadways, researchers walking around during the experiment, etc.). Optical flow based background subtraction will also fail to extract the vehicle positions if the vehicles come to a complete standstill during part of the experiment. This occurs, for example, at the start of the test, at the conclusion of the test, and if a complete stop-and-go wave is formed. Nevertheless, for



(a) (b) (c)

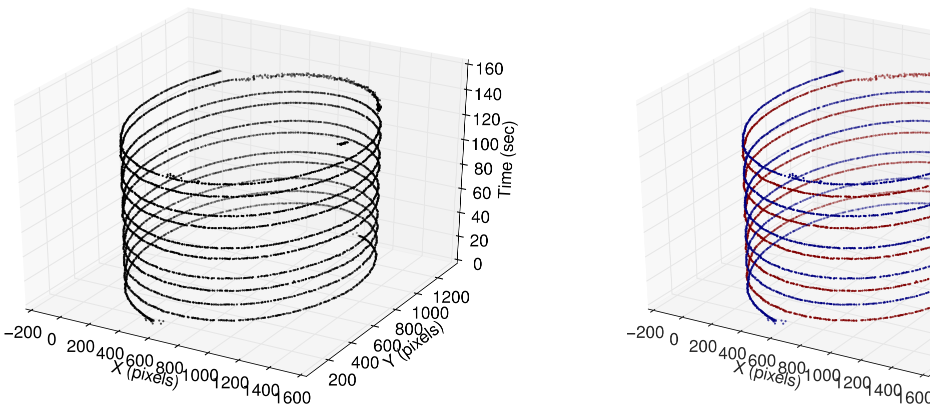
Figure 1: Segmenting vehicles from the background using optical flow and k-means: (a) A frame extracted from the panoramic camera video converted to an annulus so the left and right image boundaries are connected; (b) The separated foreground after using optical flow for background subtraction. Dark region is the removed static background, while the rest are the moving foreground; (c) The result of k-means clustering on the foreground pixels. Distinct clusters are colored differently, with the centers of the clusters marked by a cross. The partitioning correctly expresses the physical presence of the vehicles in the scene.

the experiments conducted as part of this preliminary work, optical flow is shown to provide good performance. Improved algorithms may be necessary in future experiments, as noted in the future work section.

After removing the static background, k-means clustering is then applied to segment foreground pixels into individual vehicles in the frame. Given the number of clusters, the k-means algorithm finds a partitioning that minimizes the variance within each cluster. It is one of the most widely used clustering techniques in the research community, and with the implementation of efficient heuristics, it is able to scale to very large datasets with convex and isotropic clusters with relatively low dimensionality. In addition, since the number of vehicles run during each experiment is known a priori, one is able to simply set coefficient k equal the number of vehicles on the ring, leading to excellent clustering performance.

One difficulty in clustering the foreground data is due to the panoramic video boundary. In polar coordinates on the ring, pixels located at 359 degrees are likely to belong to the same vehicle as pixels located at 1 degree. Unfortunately, in the (x, y) coordinates of the panoramic image, the periodicity of the image is lost, and consequently a direct application of k-means will not cluster the pixels as belonging to the same vehicle. To circumvent this difficulty, the panoramic video is first warped into an annulus as shown in Figure 1, where the left and right boundaries are attached together. Clusters can be calculated directly in the transformed space without concern for the periodic image boundary condition.

After application of k-means to group the foreground pixels associated with each vehicle, the center of each vehicle is estimated as the k-means cluster center. With the cluster centers marked in each frame, the final step is to connect cluster centers in two adjacent frames as belonging to the same vehicle, so the trajectory can be reconstructed. Essentially, each cluster center is located at a position x and y in the annulus frame of reference (i.e., Figure 1, and is also associated with a timestamp identified by the video frame. The set



(a) (b)

Figure 2: Reconstructing vehicles' trajectories through DBSCAN. The raw point cloud of k-means cluster centers in space and time (left) prior to clustering into trajectories. The processed data cloud (right) after applying DBSCAN. Spirals are colored differently to distinguish the trajectories of different vehicles in the test.

of $\{x, y, t\}$ points for all vehicles create a family of points in three dimensional space which can further be clustered into vehicle trajectories. To achieve this clustering across time, DBSCAN is applied.

DBSCAN is a density-based clustering technique that groups together points that are close in space and time while discards sparsely spaced points as outliers [16]. Unlike k-means, it does not take as an input the number of clusters, and instead determines the number of clusters based on two parameters that define how clusters are created. Specifically, it has two parameters that define how high the density of points in space need to be in order to be considered as a valid independent cluster, or if the points will be assigned to an existing cluster. Similar to spectral clustering and unlike k-means, DBSCAN can discover non-convex shaped structures, but runs much faster than spectral clustering. Because the trajectories in space-time are dense non-convex 3D spirals (see Figure 2), DBSCAN is therefore picked over spectral clustering and k-means clustering for reasons of efficiency and accuracy. An application of DBSCAN to two vehicles driving along the ring is shown in Figure 2.

The algorithms described above are very well established methods and there are many off-the-shelf libraries available in the research community. For example, the computer vision library *OpenCV* [17] contains an implementation of dense optical flow based on the algorithm proposed in [15], while the Python machine learning library *scikit-learn* has highly optimized implementations of k-means clustering and DBSCAN [18]. Moreover, it also implements mini batch k-means clustering, a variant of k-means clustering with nearly the same level of accuracy but faster runtime [19]. We leverage the mini batch k-means implementation, which reduces the runtime of the image segmentation step from approximately one hour per experiment down to under 30 minutes. These open-sourced libraries are used in this paper to produce the results presented in the application section to follow.

2.3 Smoothing methods

The trajectories collected with video processing techniques are usually highly noisy. Therefore we need to pre-process data via smoothing methods and robust numerical differentiation to obtain reliable velocity and acceleration profiles.

Several standard smoothing techniques are available for improving the regularity of noisy data, like for example the *local regression using weighted linear least squares and a second-degree polynomial model* (LOESS) and the *local regression using weighted linear least squares and a first-degree polynomial model* (LOWESS). In this paper we use the robust version of LOESS. The robust LOESS (RLOESS) first smooths the value at a given point through local regression using nearby data points (the range of the data points is chosen by a parameter called span) and then it calculates the residuals from the regression and it assigns a robustness weight to each data point within the span. If we apply the RLOESS to position, velocity, and acceleration separately, different spans must be used due to the fact that the regularities of these quantities decrease. It is worth noticing in this context that applying the smoothing separately to locations, velocities, and accelerations means that these quantities may no longer be consistent, i.e., the smoothed speed may be no longer the derivative of the smoothed location.

Instead of applying the smoothing method to individual trajectories of location, velocity, and acceleration we use a numerical differentiation scheme called the Tikhonov regularization to derive the velocity and the acceleration from the smoothed position data. Such a scheme is in fact robust against noise in the data and provides speeds that are consistent with locations. In the following we give a high level description of the method, we refer to [13] for more details.

The main idea of the method is represent the derivative g of a certain function f as the solution to a minimization problem, whose cost function is the sum of two terms: a regularization term and a data fidelity term. The square of the L^2 norm of g' (the derivative of g) is chosen as the regularization term, which penalizes irregularities in the derivative. With such choice the derivative g is forced to be continuous. Instead, the data fidelity term penalizes the discrepancy between the integral of the derivative g and the function f. These methods are used to compute the velocity and the acceleration from the vehicle position data extracted from the video.

2.4 Vehicle statistics from OBD-II scanners

In addition to the vehicle position, speed, and acceleration using the 360-degree camera, we also use OBD-II scanners to connect with the on-board computer on each vehicle and log performance data. OBD-II compatibility is required for all vehicles that are sold in the United States past 1996. Every vehicle is required to publish vehicle performance statistics such as engine speed, velocity, and fuel rate. Using an OBD-II scanner, one can record these quantities at up to 20Hz. This supplements the video trajectory data by giving insight into how much fuel is being used in both uniformly flowing and oscillatory traffic.

For the purposes of this experiment, vehicle speed (km/h), instantaneous fuel economy (l/100km), total fuel economy (l/100km since start of test), fuel rate (l/hr), engine speed (rpm), air mass flow rate (grams per second), absolute throttle position (percent),

Table 1: Summary statistics for each experiment

Test	# of Vehicles	Duration (s)	Avg Spacing +/- Stdev (m)	Avg Headway +/- Stdev (s)
A	10	250	9.31 +/- 1.43	5.44 +/- 1.91
B	9	330	10.34 +/- 2.07	5.76 +/- 2.16
C	10	195	9.41 +/- 1.33	4.89 +/- 2.34
D	9	225	10.30 +/- 2.02	5.19 +/- 2.28

relative throttle position (percent), and acceleration (m/s^2) from the tablet in the vehicle are recorded. These parameters are selected since they can be used to determine how traffic instabilities affect fuel consumption.

While the data is collected at 20Hz, not all parameters are updated at the same rate by the on-board computer. Furthermore, vehicle velocity is only reported to the nearest 1km/h, which makes OBD-II data better for summary statistics of vehicle performance, and suggests that the trajectories obtained using the 360-degree panoramic camera are better suited for precise velocity profiles.

3 Application of methods to ring road data

3.1 Experimental setup

This section details a set of four tests (two each) with nine vehicles and 10 vehicles on a ring with a 30m diameter. While a larger-diameter track may render a more realistic driving scenario with a smaller steering angle, the diameter selected for this study was sufficient to re-create the driving conditions desired to collect vehicle trajectory data in oscillatory traffic. We influence the density of traffic on the ring by removing a vehicle between tests, but keeping the total length of road the same. In order to approximate a road of infinite-length, a circular ring is laid out on a large, flat paved surface. The single-lane ring road is marked using cones on the inside edge of the lane. The circumference of the center line of the road is 94m. This is selected such that the vehicle density on the road is similar to that of the Sugiyama experiment [1] when nine vehicles are on the road. When setting the track, a 3m lane width is considered.

At the center of the ring, a camera capable of recording the surrounding 360 degrees is placed on a tripod in order to record vehicle positions. All 10 rental fleet vehicles are instrumented with OBD-II sensors which are paired via Bluetooth to a tablet which runs an app to log the data. Note that due to technical problems with the data logger, only five of the OBD-II loggers successfully collected data during the experiment. All of the vehicles are labeled with large numbers on the side, and have their *vehicle identification number* (VIN) recorded.

Before each test, vehicles are aligned on the track such that there is an equal inter-vehicle spacing at the start of the test. For tests A and B, drivers try to maintain a constant velocity, while in tests C and D drivers are instructed to follow the vehicle in front of them as closely as they feel comfortable. The number of vehicles in each test and total time for which the test is allowed to run for each test is summarized in Table 1.

Table 2: Summary of vehicles used in experiment and corresponding average fuel rate (l/hr) for each experiment. Note that due to low quality OBD-II loggers on vehicles 1-5, no data was collected.

Veh. #	Make & Model	Fuel rate, Test A (l/hr)	Fuel rate, Test B (l/hr)	Fuel rate, test C (l/hr)	Fuel rate, Test D (l/hr)
6	Ford Focus	1.38	1.26	2.22	2.02
7	Chrysler 200	1.25	1.37	1.74	1.82
8	Ford Fusion	2.09	2.15	2.67	2.79
9	Nissan Versa Note	1.22	1.20	1.26	1.28
10	Nissan Versa Note	1.18	N/A	1.24	N/A

3.2 OBD-II data

The OBD-II data for all experiments conducted are collected and made available online at `https://uofi.box.com/v/RingRoadTRB` for future research.
For analysis, we consider the the instantaneous fuel consumption of vehicle 6, a 2016 Ford Focus with VIN 1FADP3K26GL298016. The fuel consumption collected using an OBD-II scanner of this vehicle in test A (10 vehicles, smooth traffic) and test C (10 vehicles, oscillatory traffic), as well as test B (nine vehicles, smooth traffic) and test D (nine vehicles, oscillatory traffic) is compared. The resulting plot in Figure 3 indicates that vehicle 6 consumed on average nearly twice as much fuel per unit time when driving in oscillatory traffic as compared to when driving in uniformly-spaced traffic. This is confirmed in Table 2, which shows the average fuel rate for each instrumented vehicle for each test. For every vehicle, the fuel consumption is higher for tests with oscillatory traffic (C and D) than smooth traffic (A and B). It is important to note that this is due in part to the lower average speed at which the vehicles travel when in non-oscillatory traffic. Therefore, vehicles are using more fuel, but also traveling further per unit time. Again, note that vehicles 1-5 were equipped with low-quality OBD-II loggers and consequently did not record useful data.

3.3 Vehicle trajectories

The full trajectory dataset is published online at `https://uofi.box.com/v/RingRoadTRB`. The source code is published at: `https://github.com/Lab-Work/TRB_Conference_2016`. As a summary, Table 1 lists each individual test of the dataset, as well as summary statistics regarding the average headway and spacing (to the center of the vehicle as calculated by the vehicle position estimate obtained from the camera).

To give a closer look at the vehicle trajectory data, the trajectories from test A (see Figure 4) and test C (see Figure 5) are plotted. The trajectory data from test A and test C show how different individual driver behavior can collectively affect the global characteristics of the traffic by turning smooth traffic into oscillatory traffic. For example, Test A, drivers are instructed to keep a constant cruising speed throughout the test. As a result, no apparent traffic wave wave is observed in the trajectory plot. In Test C, drivers are instructed to follow the vehicle in front of them as closely as they feel comfortable. As a result of the change

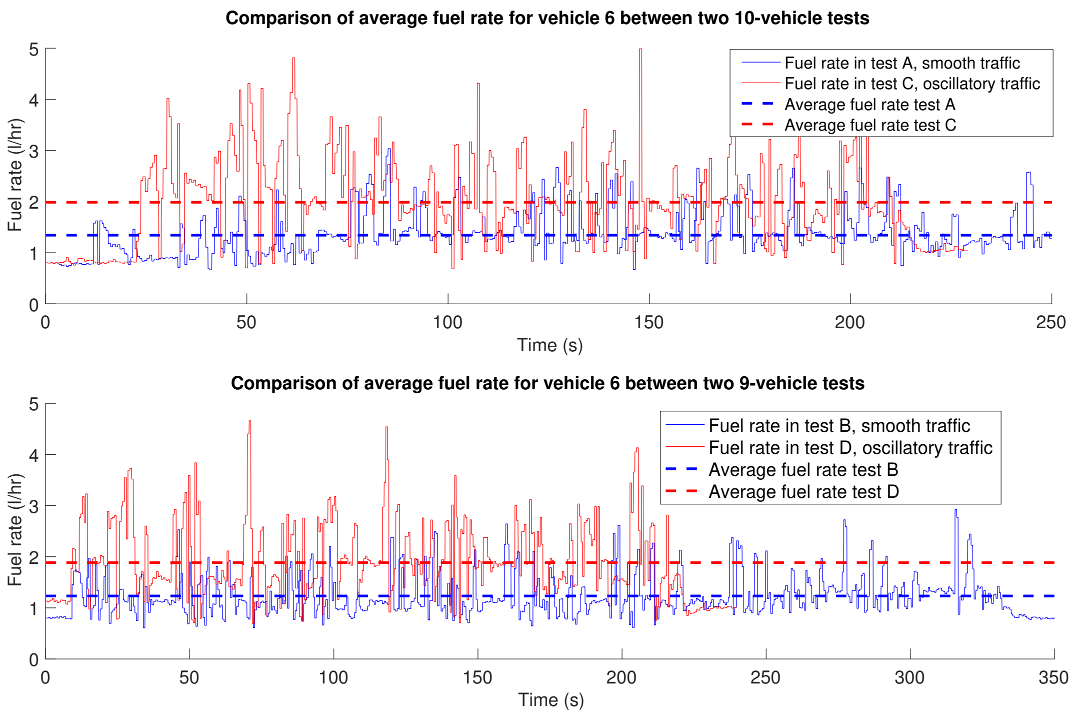


Figure 3: Comparison of fuel burn rate of a 2016 Ford Focus in both smooth and oscillatory traffic. Top subfigure shows comparison of fuel rate for vehicle 6 during two 10-vehicle tests, both with and without oscillatory traffic, while the bottom subfigure makes the same comparison for the two 9-vehicle tests.

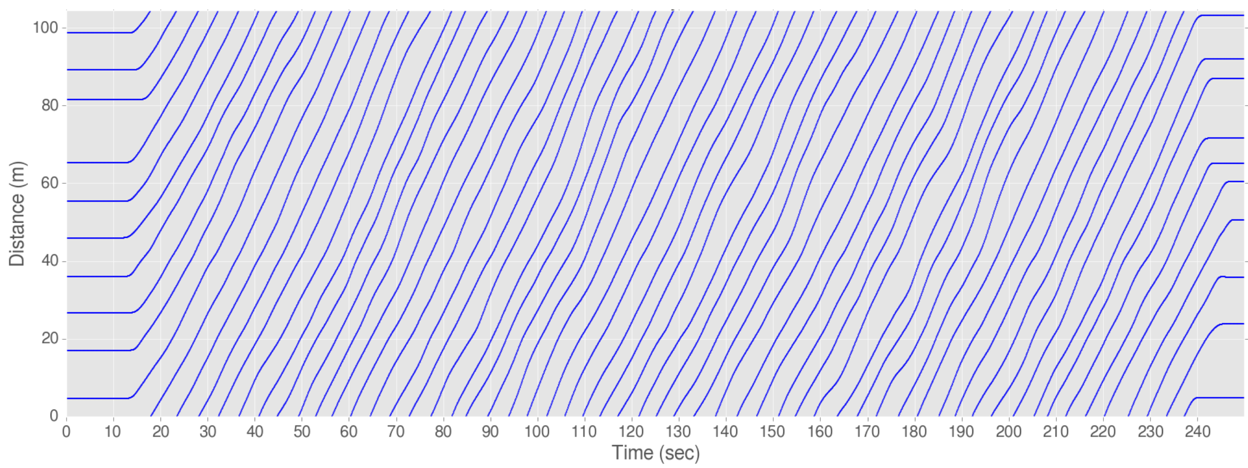


Figure 4: Trajectories for Test A: The test has 10 vehicles which are evenly spaced 10.45m, the same spacing as the 22-vehicle Suigiyama experiment. The test lasts 4 minutes and 10 seconds.

of instructions, oscillations in the traffic flow are observed, with the strongest variations produced in the 60-120s interval.

For tests B and D we present the estimated velocity and acceleration both from the smoothed camera data and from the OBD-II scanner in Figure 6. This demonstrates the higher-quality estimates obtained when smoothing the optical flow velocity as compared to the data obtained from the OBD-II data. Specifically, the acceleration data obtained from the OBD-II scanner produces unrealistically high acceleration rates that oscillate at unrealistically high frequencies. It is also worth noting that the un-smoothed velocity estimates from optical flow are not included in Figure 6 since they are too noisy to be contained in the same plot. When comparing the velocity and acceleration estimates in Figure 6, we observe greater variation in test D than test B. This is a result of the traveling wave observed in test D that was not observed in test B.

4 Conclusion and future work

The main focus of the present article is the collection and processing of trajectory and fuel consumption data from vehicles in oscillatory traffic generated on a closed ring road track. The article provides a detailed description of the vision processing algorithms used to extract noisy trajectory data in the form of a position timeseries. It also includes a method to smooth the trajectories to produce consistent velocity and acceleration trajectory data from noisy position data. It is noted that the estimated velocity data is higher resolution than the OBD-II velocity data, which only reports the velocity to the nearest kilometer per hour. In addition to providing the video, raw trajectory, smoothed trajectory, and OBD-II data from the four tests, this article also contains source code for all processing algorithms used to construct the various datasets. It is hoped this will lead to a more robust data processing pipeline and support additional data collection efforts.

Several areas are open for future work. The present nine and 10 car experiments are

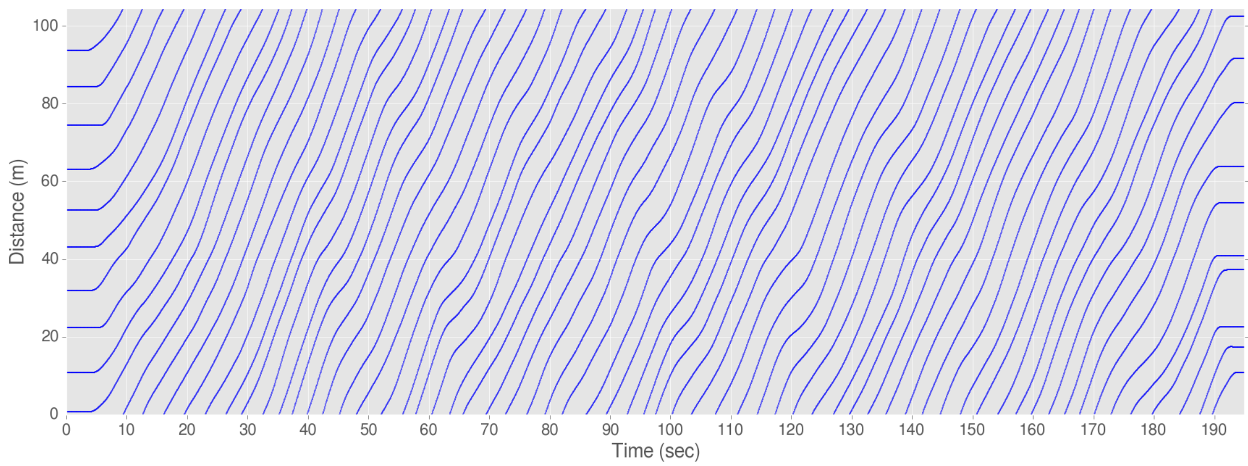


Figure 5: Trajectories for Test C: The whole process lasts for 3 min and 15 sec, where oscillations occur after 30s into the test. The traffic wave is at its strongest within the 60 - 120s interval.

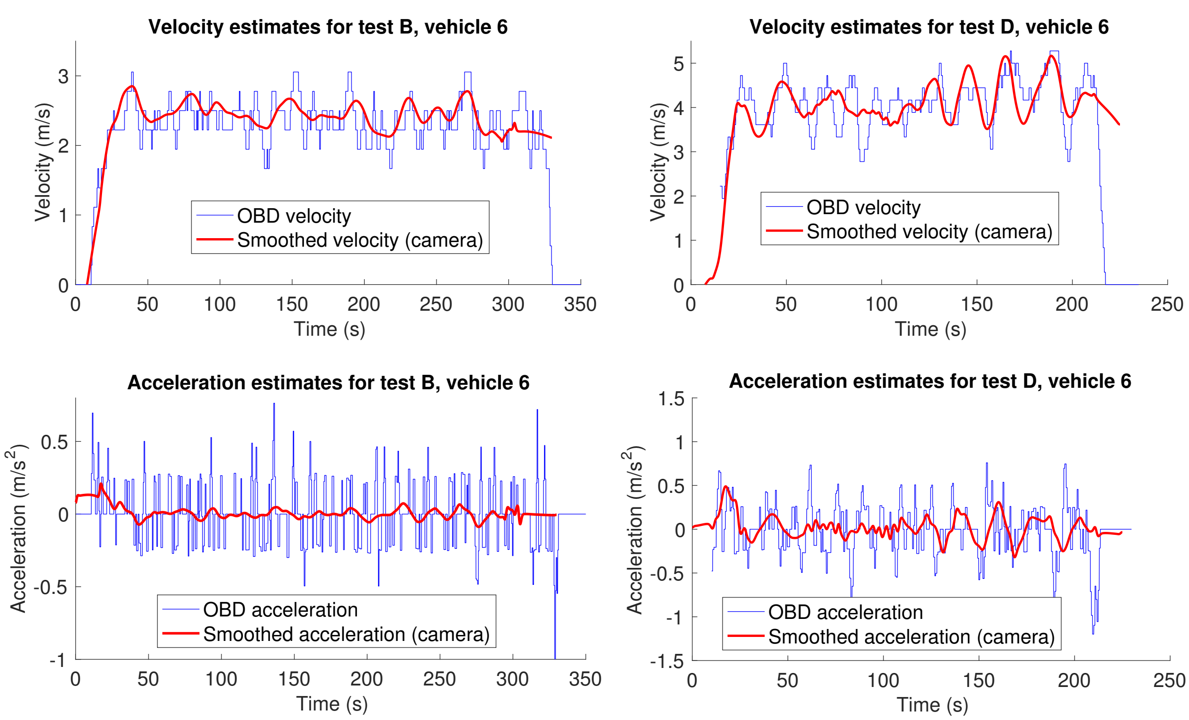


Figure 6: Comparison of velocity and acceleration estimates from OBD-II scanner and optical flow of vehicle 6 for test 2 and 4. OBD-II introduces significant rounding in the velocity estimates and produces noisy acceleration estimates.

part of a larger effort to collect fuel consumption data and detailed trajectory information from vehicles in oscillatory traffic. This experiment is the first and smallest of several planned experiments, which will include full-scale tests to replicate the results in Sugiyama et al. [1] but with additional fuel consumption data collection, as well as other detailed diagnostics provided by the OBD-II scanners.

Due to the physical constraints on the track size, the maximum speed drivers could comfortably reach in the experiments presented was restricted. Thus, the fuel consumption measurements recorded may not be representative of typical highway driving. The larger-scale experiments are expected to see vehicles attaining a higher maximum speed, which will give further insight into the influence of oscillatory traffic on fuel consumption.

Finally, further refinements to the vision pipeline to address noisy background data, vehicle tracking even when vehicles are at rest, and complete OBD-II data from each vehicle on the ring are planned in the near future.

Acknowledgments

The material is based on work supported by the National Science Foundation under Grant No. CNS-1446702. The authors acknowledge Parkland College for granting us access to the parking lot and the participating drivers.

References

[1] Yuki Sugiyama, Minoru Fukui, Macoto Kikuchi, Katsuya Hasebe, Akihiro Nakayama, Katsuhiro Nishinari, Shin ichi Tadaki, and Satoshi Yukawa. Traffic jams without bottlenecks-experimental evidence for the physical mechanism of the formation of a jam. *New Journal of Physics*, 10(3):033001, 2008.

[2] Masako Bando, Katsuya Hasebe, Akihiro Nakayama, Akihiro Shibata, and Yuki Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, 51(2):1035, 1995.

[3] Dirk Helbing and Benno Tilch. Generalized force model of traffic dynamics. *Physical Review E*, 58(1):133, 1998.

[4] Hwasoo Yeo and Alexander Skabardonis. Understanding stop-and-go traffic in view of asymmetric traffic theory. In *Transportation and Traffic Theory 2009: Golden Jubilee*, pages 99–115. Springer, 2009.

[5] Gábor Orosz, Eddie Wilson, Róbert Szalai, and Gábor Stépán. Exciting traffic jams: nonlinear phenomena behind traffic jam formation on highways. *Physical Review E*, 80(4):046205, 2009.

[6] Antoine Tordeux and Armin Seyfried. Collision-free nonuniform dynamics within continuous optimal velocity models. *Physical Review E*, 90(4):042812, 2014.

[7] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805, 2000.

[8] Federal Highway Administration. NGSIM dateset, 2003.

[9] Christian Thiemann, Martin Treiber, and Arne Kesting. Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data. *Transportation Research Record: Journal of the Transportation Research Board*, 2088:90–101, 2008.

[10] Marcello Montanino and Vincenzo Punzo. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transportation Research Part B: Methodological*, 80:82–106, 2015.

[11] Vincenzo Punzo, Maria Teresa Borzacchiello, and Biagio Ciuffo. On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (NGSIM) program data. *Transportation Research Part C: Emerging Technologies*, 19(6):1243–1262, 2011.

[12] Martin Treiber, Arne Kesting, and Christian Thiemann. How much does traffic congestion increase fuel consumption and emissions? applying a fuel consumption model to the NGSIM trajectory data. In *87th Annual Meeting of the Transportation Research Board, Washington, DC*, 2008.

[13] Benedetto Piccoli, Ke Han, Terry L. Friesz, Tao Yao, and Junqing Tang. Second-order models and traffic data from mobile sensors. *Transportation Research Part C*, 52:32–56, 2015.

[14] Juan C. Herrera, Daniel B. Work, Ryan Herring, Xuegang Jeff Ban, Quinn Jacobson, and Alexandre M Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568–583, 2010.

[15] Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. *Lecture Notes in Computer Science*, 2749:363–370, 2003.

[16] Anant Ram, Sunita Jalal, Anand S. Jalal, and Manoj Kumar. A density based algorithm for discovering density varied clusters in large spatial databases. *International Journal of Computer Applications*, 3(6):1–4, Oct 2010.

[17] OpenCV.org. Open source computer vision library. `http://opencv.org/`, 2016.

[18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[19] David Sculley. Web-scale k-means clustering. *Proceedings of the 19th international conference on World wide web - WWW '10*, 2010.