Zhou, Qishen ; Mohammadi, Roozbeh; Zhao, Weiming; Zhang, Kaihang ; Zhang, Lihui ; Wang, Yibing ; Roncoli, Claudio; Hu, Simon

Queue Profile Identification at Signalized Intersections with High-Resolution Data from Drones

Published in:
2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2021

DOI:
10.1109/MT-ITS49943.2021.9529337

Published: 16/06/2021

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Zhou, Q., Mohammadi, R., Zhao, W., Zhang, K., Zhang, L., Wang, Y., Roncoli, C., & Hu, S. (2021). Queue Profile Identification at Signalized Intersections with High-Resolution Data from Drones. In *2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2021* IEEE. https://doi.org/10.1109/MT-ITS49943.2021.9529337

Queue Profile Identification at Signalized Intersections with High-Resolution Data from Drones

Qishen Zhou*†, Roozbeh Mohammadi‡, Weiming Zhao‡, Kaihang Zhang*,
Lihui Zhang†, Yibing Wang†, Claudio Roncoli‡, Simon Hu*†
*Zhejiang University/University of Illinois at Urbana-Champaign Institute (ZJU-UIUC Institute), Haining 314400, China
Emails: {qishenzhou, kaihang.17}@intl.zju.edu.cn, simonhu@zju.edu.cn
†College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China
Emails: {lihuizhang, wangyibing}@zju.edu.cn
‡Department of Built Environment, Aalto University, Espoo, Finland
Emails: {roozbeh.mohammadi, weiming.zhao, claudio.roncoli}@aalto.fi

***Abstract*—Queue profile is a crucial measure for traffic management in the vicinity of signalized intersections. In this study, we develop a method to identify queue profile using high resolution data, which can be provided from various sources such as drones. Our methodology has three main components which are signal state estimation, queue profile identification, and lane detection. The developed algorithms are tested on the real-world dataset collected by drones as a case study for validation. Remarkably, our method only uses drone data as input and it is independent from any other data source such as geographic information system data. The results demonstrate satisfactory performance of the methodology in extracting queue profile information from raw drone data. The developed algorithm can be also applied on data collected via connected vehicles in future.**

***Index Terms*—queue length, traffic management, high resolution data, trajectory data, clustering**

I. Introduction

Estimation of the traffic state in urban signalized links is an crucial task for Intelligent Transportation Systems (ITS) [1]. Earlier studies on traffic state estimation have been mainly based on fixed-location sensors (e.g. radar-based devices, loop detectors, cameras, etc.) [2]–[4]. However, the collection of detailed traffic data with fixed-location sensors is a difficult process as it requires a large amount of installed sensors in order to cover the entire network [5]. One approach to improve estimation and overcome the problem of limited fixed-location sensors is to use advanced mobile sensors, like probe vehicles or connected vehicles (CVs) [6]–[9]. Such vehicles will continuously upload their status information (e.g., latitude, longitude, instantaneous speed, and moving direction) to data centers via wireless communications [10]. The trajectory dataset obtained from probe vehicles is pretty useful, but its main limitation will be the low penetration rate of such vehicles in near future.

Recently, Unmanned Aerial Vehicles (UAV) or "drones" have started to take the center stage of traffic monitoring as they can carry high quality cameras and other technological equipment [5]. This provides new opportunities for more accurate estimation of traffic information. pNEUMA is a unique vehicles trajectory dataset obtained from the first experiment using a large number of drones in a dense city center, with unprecedented high resolution and scale [11]. pNEUMA has been used in many fields such as map mapping and lane-changing identification. For example, compared with the well-known Next Generation Simulation (NGSIM) trajectory dataset [12], pNEUMA has a larger scale and higher spatial coverage.

Queues are the main cause of traffic delays at signalized intersections. The accurate estimation of queue profile is crucial for optimization of traffic signals and vehicle trajectories [13]. Therefore, queue profile estimation at signalised intersection has been widely studied [14]–[16]. The focus of this paper is to identify the queue profile from pNEUMA dataset. Since the dataset does not contain lane information and signal timing plan, we have developed a queue profile identification algorithm suitable for multi-lane environment, which can automatically estimate signal timing plans and help identify queue profiles. Moreover, we developed a lane detection algorithm which is useful to find, e.g., the lane where maximum queue occurs. The algorithm is applied to a corridor with three links and three signalized intersections and the results show that the developed algorithm can accurately identify the queue profile at the intersections.

The remainder of this paper is organized as follows. Section II describes the three components of the methodology of this paper which are signal state estimation, queue profile identification, and lane detection. Results of implementing developed methodology on pNEUMA dataset as a case study are presented in Section III. Finally, Section IV summarises and discusses the key findings and outlines further research

This work is supported in part by the key program project by Ministry of Science and Technology, China, grant no.2018YFB1600500. The Aalto University team received funding from the Henry Ford Foundation Finland (Henry Fordin Säätiö) and the FINEST Twins Center of Excellence (H2020) under Grant 856602. This work was supported in part by ZJU-UIUC Joint Research Center Project No. DREMES 202003, funded by Zhejiang University. The work was led by Principal Supervisor Simon Hu.

directions.

II. Methodology

This section discusses our proposed method for queue profile extraction, in which the profiles of the stop wave for each cycle are estimated via a recursive searching process. Our methodology framework is divided into three main parts: (i) the estimation of signal timing parameters using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, (ii) queue profile identification based on dynamic time-space cells, and (iii) lane detection. The detailed flowchart of proposed method is shown in Fig. 1 and the methodology is elaborated as follows.

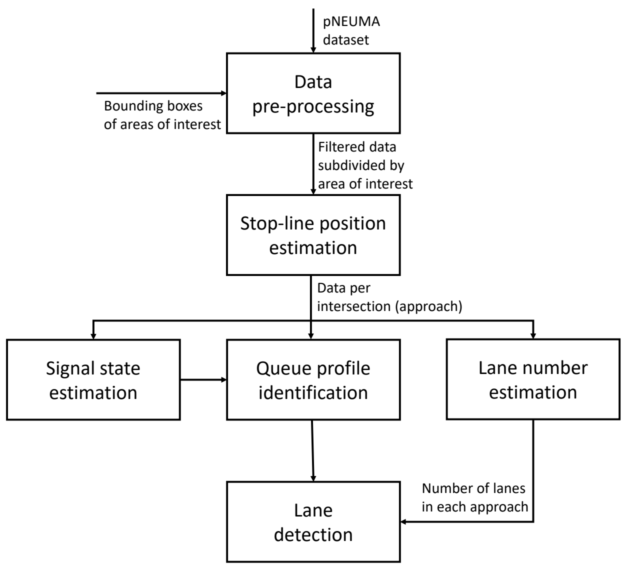

Fig. 1. Flowchart of the proposed methodology

A. Signal Timing Plan Estimation

Signal time is crucial to capture the spatial-temporal characteristics of the queue and it is also an important factor for accurately identifying queue spillback. On urban roads, signalized intersections are the main reason for queuing. Since the vehicles follow the rule of stopping at red and starting-up at green, the stop time of the vehicle at the front of the queue can be assumed being close to the start time of the red light and the start-up time is also close to the start time of the green light. Based on this, we can estimate signal timing plan from the vehicle trajectories.

As shown in Fig. 2, there are many factors that can make vehicles stop, including: queue caused by red lights at signalized intersections (**Case 1**), queue caused by spillback generated downstream (**Case 2**), and some random events, such as pedestrians crossing the street (**Case 3**), so only data belong to **Case 1** are useful for signal timing plan estimation. As we can see, vehicles stopping at red and starting-up at green is periodical due to the fixed signal timing in **Case 1**, which makes possible to classify parking samples (data collected when vehicles stopped) whose speed is less than the preset threshold by cycle using a clustering algorithm, and extract signal timing plan from each cycle's parking samples. Since we have no information about the number of clusters, we use an unsupervised clustering algorithm that is able to find the optimum clusters as well as is able to remove the noise. DBSCAN [17] is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers the points that lie alone in low-density regions (whose nearest neighbors are too far away) [18]. Therefore, using DBSCAN to classify parking samples is expected to achieve a good result. The detailed flow of the algorithm is as follows.

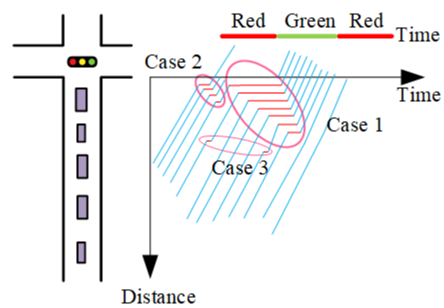

Fig. 2. Example of vehicle trajectories and vehicle stopping cases.

First, we can filter out moving samples (data collected when vehicles are moving) by setting a threshold for vehicle's speed according to

$$VEH^k_{i,c} = \begin{cases} S & VEH^k_{i,v} \le v_{th} \\ M & VEH^k_{i,v} > v_{th} \end{cases} \tag{1}$$

where VEH^k_i is the k-th sample of the i-th vehicle, $VEH^k_{i,v}$ is the speed of the k-th sample of the i-th vehicle, $VEH^k_{i,c}$ is the category of the k-th sample of the i-th vehicle, and v_{th} is the preset speed threshold, which is set as 3.6 km/h in this paper.

Next, we filter out moving samples, since only parking samples near the stop-line in **Case 1** are needed, and the parking time of this part of the parking samples is longer than the parking samples in other cases. Therefore, the data can be further filtered by the parking time. Parking samples in j-th parking for i-th vehicle are filtered as follows:

$${}^{j}VEH_{i,st} = \max_t({}^{j}VEH^k_{i,t}) - \min_t({}^{j}VEH^k_{i,t}) \tag{2}$$

$${}^{j}VEH^k_{i,t=t_{min},c} = S_1 \tag{3}$$

$${}^{j}VEH^k_{i,t=t_{max}}, c = S_2$$

$$((i,j) \in \{(i,j)|({}^{j}VEH_{i,st} \le t_{th}\}$$

$$t_{max} = \operatorname*{argmax}_t {}^{j}VEH^k_{i,t}, t_{min} = \operatorname*{argmin}_t {}^{j}VEH^k_{i,t}) \tag{4}$$

where ${}^{j}VEH_{i,st}$ is the total parking time of the i-th vehicle during the j-th parking period, and ${}^{j}VEH^k_{i,t}$ is the timestamp of the k-th sample during the j-th parking of the i-th vehicle. As we just want to estimate when red light and green light start, only stopping samples S_1 (data collected when vehicle start parking) and starting-up samples S_2 (data collected when vehicles start-up) that are in front of queue are useful. According to shockwave theory, the parking time of vehicles at the front of the queue should be longer than that of vehicles at the back of the queue. So in order to extract those samples, a relatively bigger t_{th} should be set.

After stopping samples S_1 and starting-up samples S_2 in front of queue in **Case 1** are extracted, the DBSCAN algorithm is used to cluster them into different cycles. In order to obtain an accurate number of cycles, we set up a loop to continuously test the two main parameters of DBSCAN, namely eps and $min\_samples$, until the convergence condition is reached, that is, the number of cycles remains unchanged for six iterations. Among them, eps is a parameter specifying the radius of a neighborhood and $min\_samples$ is the minimum number of points required to form a dense region. Taking S_1 for example, the steps are as follows.

- *Step 1*: Initialization: set DBSCAN algorithm parameter $eps = 1$, and $min\_samples = 3$; set $i = 0$;
- *Step 2*: Input the samples **t** into DBSCAN algorithm, output the labels for each data **l** and numbers of class n_0, where **t** are the timestamp set attached to the samples in S_1.
- *Step 3*: if $n_0 > 1$:
  Continue;
  else:
  Set $eps = eps + 1$ and return to *Step 2*;
- *Step 4*: Set $eps = eps + 1$, input the **t** into DBSCAN, output the labels for each data l and numbers of class n;
- *Step 5*: if $n - n_0 == 0$:
  $i = i + 1$
  Continue;
  else:
  if $n > 1$:
  $n_0 = n$
  Skip to *step 7*;
- *Step 6*: if $i == 6$:
  Set $\mathbf{l} = \mathbf{l}^*$, $n^* = n$ and skip to *Step 8*;
  else:
  Return to *Step 4*;
- *Step 7*: Set $i = 0$ and return to *Step 4*;
- *Step 8*: According to labels $\mathbf{l}^*$, cluster samples in S_1 to cycles, and filter out the noise identified by DBSCAN.

Thirdly, after samples in S_1 and S_2 are classified into cycles, the start time of the red and green lights of each period can be determined by the minimum and maximum timestamps of the samples in each period belong to S_1 and S_2.

Finally, it should be noted that, in order to ensure that the number of cycles obtained is accurate, we need to set a larger neighbor distance threshold. In this case, DBSCAN cannot remove noise well. To solve this problem, we clustered the sample twice. First, we cluster the entire sample, setting a larger neighbor distance with the goal of obtaining the exact number of cycles. Then, samples are clustered in each period, setting a smaller neighbor distance to achieve better noise removal effect.

B. Queue Profile Identification

We introduced that parking samples can be extracted by speed threshold v_{th} and, since all parking samples are needed in the process of identifying the queue profile, we cannot filter out the parking samples of **Cases 1,2** by setting the threshold of the parking time. To this end, we designed a queue profile identification algorithm which can describe the spatial-temporal formation and dissipation of queues, and the algorithm can work regardless of whether a spillback occurs or not. Before proceeding to the next step, the distance from the parking sample to the stop line needs to be calculated, so that the parking sample can fall on the time-space plane composed of timestamp and the distance between sample and stop line.

To illustrate our method, let us refer to the scenario depicted in Fig. 3, where the 2^{nd} link is upstream of 1^{st} link, and both intersections are signalized. t_r is the effective red light start time of a certain period of 1^{st} link, and t_g is the effective green light start time of the period. The red line in the graph is the data collected when the vehicle is parked. t_1 is the time when the first vehicle in second space interval starts to stop, t_2 is the time when the first vehicle in second space interval start-up, and t_3 is the is the time when the first vehicle in third space interval starts to stop. In order to filter out the parking samples that do not belong to **Case 1**, we first divide the time-space plane by the space interval Δd. Then, we traverse each distance interval from top to bottom, and determine whether the samples in the current interval belong to **Case 1** by judging whether there are parking samples in the previous interval of the current interval. If there is no sample in a certain interval, it means that the end of the queue is in the previous interval. Although there may be samples in the following intervals, those samples do not belong to **Case 1**. However, since the data in each space interval is generated in multiple cycles, we must also limit the time range of the data in each space interval. To this end, we propose a queue profile identification algorithm based on dynamic time-space cells, it can remove the parking samples that do not belong to **Case 1** and identify the queue profile for each cycle. The basic assumption of the algorithm is that after dividing the time-space plane by the space interval Δd, (i) the time when the first vehicle in the current interval starts to stop is later than the time when the first vehicle in the previous interval starts to stop, that is, t3 is greater than t1; (ii) the time when first vehicle in the current interval starts to stop is earlier than the time when first vehicle in the previous interval starts-up, that is, t3 is less than t2. For example, if we want to identify the profile of the queue generated by the red light (t_r to t_g) of 1^{st} link, the detailed flow of the algorithm is as follows.

- *Step 1*: Extract parking samples S_3 by speed threshold v_{th} and filter out samples whose timestamp is less than

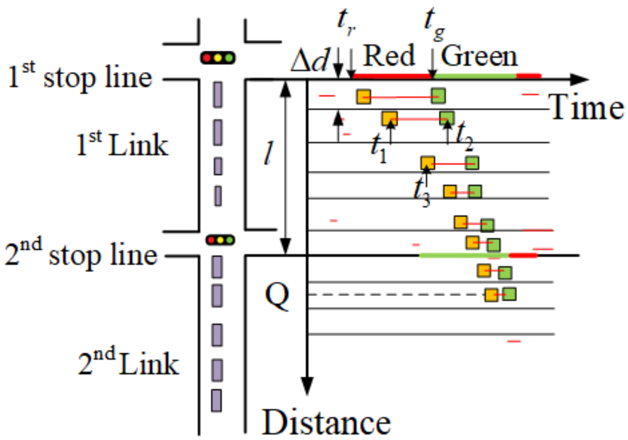

Fig. 3. Queue profile identification.

t_r; calculate the distance between samples and stop-line;

- *Step 2*: Divide S_3 on the time-space plane according to the space intervals Δd to obtain the parking sample set $\{pd_1, pd_2, ..., pd_m\}$;
- *Step 3*: Initialization: $i = 1, t_1 = t_r, t_2 = t_g, profile_f = (tr, 0), profile_d = (tg, 0)$. Among them, $profile_f$ describes the formation of the queue, which is composed of the coordinates of the yellow solid squares in the Fig. 3, and $profile_d$ describes the queue dissipation, which is composed of the coordinates of the green solid squares in the Fig. 3.
- *Step 4*: Find the distance and time when the first vehicle starts to stop in pd_i,which is the coordinates of yellow solid square in the i-th space interval in the Fig. 3: get parking samples S_4 with timestamp between t_1 and t_2 in pd_i; find the sample with the smallest time stamp and smallest distance in S4.

  $t_f = \min_t(VEH^k_{i,t})$
  $d_f = \underset{d}{\text{argmin}}(VEH^k_{i,t=t_f,d})$
  $VEH^k_i \in S_4$
  Insert (t_f,d_f) to $profile_f$

  Where $VEH^k_{i,t}$ is the timestamp of k-th sample of the i-th vehicle, and $VEH^k_{i,t=t_f,d}$ is the distance of k-th sample of the i-th vehicle from the stop-line at time t_f.
- *Step 5*: Find the distance and time when the first vehicle starts-up in pd_i, which is the coordinates of green solid square in the i-th space interval in the Fig. 3;

  Find first vehicle set in pd_i, since the research is carried out in a multi-lane environment, the vehicle cannot be uniquely determined by time and distance, So there may be more than one so-called first vehicle.

  $\mathbf{I} = \{i|\exists k \text{ make } VEH^k_{i,t} = t_f \text{ and } VEH^k_{i,d} = d_f\}$
  $VEH^k_i \in S_4$

  Where $\mathbf{I}$ is the set of vehicles with samples whose timestamp and distance equal to t_f and d_f.

  Find the sample with the largest timestamp and smallest distance belonging to the vehicle in the vehicle set I in S_3;

  $t_d = \underset{t}{max}(VEH^k_{i\in\mathbf{I},t})$
  $d_d = \underset{d}{\text{argmin}}(VEH^k_{i\in\mathbf{I},t=t_d,d})$
  $VEH^k_i \in S_3$
  insert (t_d,d_d) to $profile_d$
- *Step 6*: $i = i + 1$;
- *Step 7*: Repeat step3 - step5 until there is no data in a certain space interval;

C. Lane detection

Lane detection is performed on each separated short segment, rather than on the whole road, because of the possible changes in the number of lanes in the road, as well as potential curvatures. The motorcycle data are filtered out in the lane detection because motorcycles do not follow the lane rule closely and their data create noise in the lane detection. The 2-dimensional trajectory data is converted to 1-dimensional data by calculating the distance of each vehicle position to the boundary of the road in each segment. Then, the Gaussian Mixture Model (GMM) is applied to perform clustering, considering the position of both stopping and moving vehicles.

Determining the optimal number of clusters is one of the main challenges in applying GMM. Three factors are used in this paper, i.e., Akaike Information Criterion (AIC), the difference of the mean of a cluster, and the number of points in clusters. After the clustering, the lane index for each cluster is determined by the mean distance between the cluster and the road boundary. The leftmost lane along the road is labelled as lane 1, while the shoulder lane has the largest index. The lane information for every vehicle position is determined by predicting which cluster it belongs to. So when the position of maximum queue length is estimated, the lane that the maximum queue length belongs to can also be obtained. The procedures of lane detection are summarized in the following.

- *Step 1*: Get the trajectories data within each segment and remove the data from motorcycles;
- *Step 2*: Calculate the distance of each trajectory point to the road boundary;
- *Step 3*: Use GMM to determine the clustering on the distance data with the number of clusters varying from 1 to 8;
- *Step 4*: Select the number of clusters with the smallest AIC;
- *Step 5*: Check the difference between the mean of clusters. If any of the difference of means of every Gaussian distribution component is smaller than a threshold (2.2 m is used in this paper, which is representative of the minimum acceptable lane width), the number of clusters is reduced by 1;
- *Step 6*: Check the number of points in clusters. If the minimum number of points in a cluster is less than 20% of the second minimum number of points, the number of clusters is reduced by 1.
- *Step 7*: Select the best number of clusters and return the result of clustering. The lane id of each point is

determined by the probability of belonging to every cluster.

III. Case Study and Results

In order to verify the effectiveness of the proposed algorithm, this paper utilises part of the Leof. Alexsandras road, which is shown in the red area in Fig. 4, located in central Athens, Greece. The red area contains three links and three signalized intersections.

Fig. 4. Research area.

Since the pNEUMA dataset does not directly allows to filter data for a specific road, it is necessary to extract the data for the road in analysis. After extracting the data for the red area, different links are also distinguished for specific analysis of each road section. The final result is shown in Fig. 5, where **Link_1** is the most downstream link among the three links. Moreover, the data shows that spillbacks often occur during peak periods in the region.

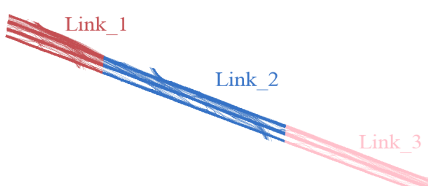

Fig. 5. Trajectories separated by links.

We proceed by first estimating the signal timing plans of the three signalized intersections. Fig. 6 shows a box-plot of the estimated cycle length of three signalized intersections with different periods. It can be seen that the estimated cycle length of the three links does not fluctuate much and is relatively stable, which conforms to the characteristics of fixed signal timing. The black dashed line in Fig. 7 represents the time when the red light starts in each cycle of **Link_1**. It can be seen that the estimate can reflect the real situation.

The results of queue profile identification on **Link_2** is shown in Fig. 8, where blue lines are the trajectories of vehicles on **Link_2** and **Link_3**. Red and green lines describe the spatio-temporal formation and dissipation of queues formed by the red lights of **Link_2**. It can be seen from the Fig. 8 that our algorithm has high accuracy and robustness, because

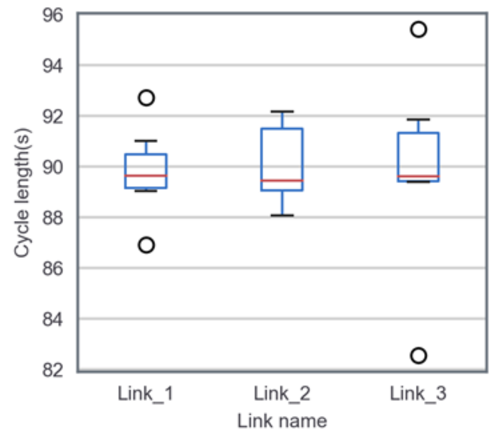

Fig. 6. Estimated cycle length distribution.

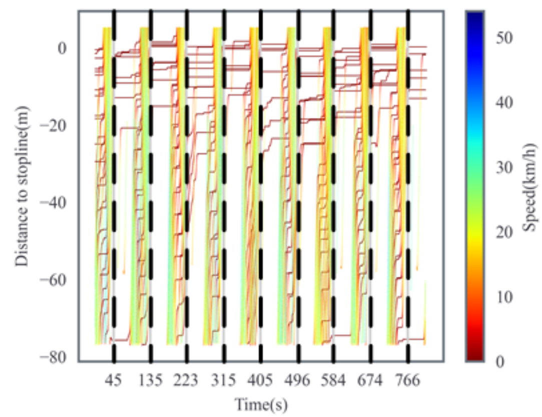

Fig. 7. Start time of red light in **Link_1**.

it can not only accurately identify the queue profile without spillback, but also accurately identify the overflow part of the queue when a spillback occurs.

The graphical representation of lane identification of three links is presented in Fig. 9. As can be seen from the Fig. 9, the data is well classified into lanes. In this way, we can obtain some key information in the queue, such as in which lane the maximum queue length occurs.

IV. Conclusion

In this paper, we introduced a comprehensive method to extract queue profile information as well as other prerequisites information,i.e., signal state from drone data. In particular, we use a set of machine learning-based methods to obtain the required information. In the first step, we implemented the DBSCAN algorithm in order to estimate signal state at each time step. Then, we used a queue profile identification algorithm based on dynamic time-space cell in order to identify queue profile, which works also when spillback appears. Furthermore, we deployed GMM clustering model to separate the vehicles based on lane, which is useful to find the lane where maximum queue occurs. The findings demonstrate that our method estimates signal state, identifies queue profile, as

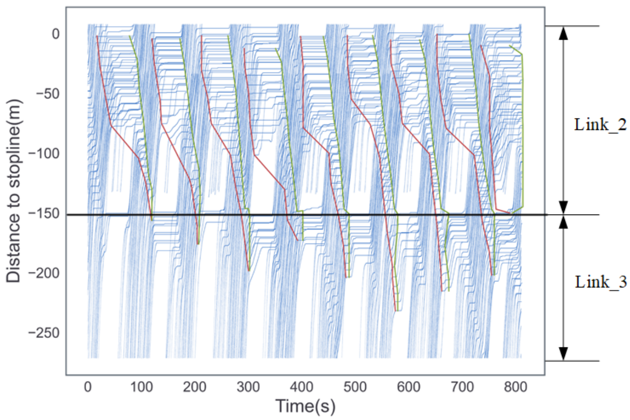

Fig. 8. The results of queue profile identification on **Link_2**

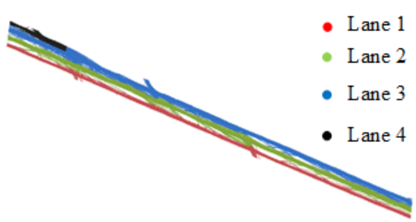

Fig. 9. Lane identification result.

well as position of maximum queue length, and detects queue lane by acceptable and useful accuracy.

The merit of this work is that the process of extracting the queue profile is done without recurring to any other external data sources, such as, e.g., geographic information system data. In other words, our methodology is able to identify the queue profile only by employing trajectory data, obtained from drones. This enables the application of our methodology on any other similar dataset, allowing to extract queue profile information at lane level. Our developed methodology can be used for various purposes. Firstly, identifying historical queue profile using drone data is extremely promising for future traffic management and developing efficient signal timing plans at signalized intersections. Moreover, our methodology may be extended for usage in a CV environment, where all vehicles can transmit similar trajectory data to the controller via V2I (vehicle-to-infrastructure) communication systems in real-time. However, further investigations are required in order to operate the method in low penetration rate of CVs. Furthermore, our method can be used as baseline for other queue profile estimations method,e.g., using point detectors or probe vehicle data.

V. Acknowledgment

The authors are grateful to Urban Transport Systems Laboratory (LUTS) at Swiss Federal Institute of Technology Lausanne (EPFL) for providing drone dataset used in this study: "Data source: pNEUMA – open-traffic.epfl.ch"

References

[1] M. Rostami Shahrbabaki, A. A. Safavi, M. Papageorgiou, and I. Papamichail, "A data fusion approach for real-time traffic state estimation in urban signalized links," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 525–548, 2018.

[2] H. X. Liu, X. Wu, W. Ma, and H. Hu, "Real-time queue length estimation for congested signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 4, pp. 412–427, 2009.

[3] G. Vigos, M. Papageorgiou, and Y. Wang, "Real-time estimation of vehicle-count within signalized links," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 1, pp. 18–35, 2008.

[4] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic engineering*. Pearson/Prentice Hall, 2004.

[5] M. Khan, W. Ectors, T. Bellemans, D. Janssens, and G. Wets, "Unmanned aerial vehicle-based traffic analysis: A case study for shockwave identification and flow parameters estimation at signalized intersections," *Remote Sensing*, vol. 10, no. 3, 2018.

[6] M. Ramezani and N. Geroliminis, "Queue profile estimation in congested urban networks with probe data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 6, pp. 414–432, 2015. [Online]. Available: https://dx.doi.org/10.1111/mice.12095

[7] H. X. Liu and W. Ma, "A virtual vehicle probe model for time-dependent travel time estimation on signalized arterials," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 1, pp. 11–26, 2009.

[8] J. Zheng and H. X. Liu, "Estimating traffic volumes for signalized intersections using connected vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 347–362, 2017. [Online]. Available: https://dx.doi.org/10.1016/j.trc.2017.03.007

[9] D. Nam, R. Lavanya, R. Jayakrishnan, I. Yang, and W. H. Jeon, "A deep learning approach for estimating traffic density using data obtained from connected and autonomous probes," *Sensors*, vol. 20, no. 17, p. 4824, 2020.

[10] L. Li, R. Jiang, Z. He, X. Chen, and X. Zhou, "Trajectory data-based traffic flow studies: A revisit," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 225–240, 2020.

[11] E. Barmpounakis and N. Geroliminis, "On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 50–71, 2020.

[12] "NGSIM – Next Generation Simulation," 2008a, [Online; accessed 18. Oct. 2008]. [Online]. Available: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[13] H. Zhang, H. X. Liu, P. Chen, G. Yu, and Y. Wang, "Cycle-based end of queue estimation at signalized intersections using low-penetration-rate vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3257–3272, 2020.

[14] H. X. Liu, X. Wu, W. Ma, and H. Hu, "Real-time queue length estimation for congested signalized intersections," *Transportation research part C: emerging technologies*, vol. 17, no. 4, pp. 412–427, 2009.

[15] X. J. Ban, P. Hao, and Z. Sun, "Real time queue length estimation for signalized intersections using travel times from mobile sensors," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1133–1156, 2011.

[16] Z. Wang, L. Zhu, B. Ran, and H. Jiang, "Queue profile estimation at a signalized intersection by exploiting the spatiotemporal propagation of shockwaves," *Transportation Research Part B: Methodological*, vol. 141, pp. 59–71, 2020.

[17] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*. AAAI Press, 1996, pp. 226–231.

[18] Contributors to Wikimedia projects, "DBSCAN - Wikipedia," Aug 2020, [Online; accessed 18. Oct. 2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=DBSCAN&oldid=975786449