

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/340055427>

Advanced Framework for Microscopic and Lane-level Macroscopic Traffic Parameters Estimation from UAV Video

Article in IET Intelligent Transport Systems · March 2020

DOI: 10.1049/iet-its.2019.0463

CITATION

1

READS

162

4 authors:



Ruimin Ke

University of Washington Seattle

40 PUBLICATIONS 566 CITATIONS

[SEE PROFILE](#)



Shuo Feng

University of Michigan

28 PUBLICATIONS 147 CITATIONS

[SEE PROFILE](#)



Zhiyong Cui

University of Washington Seattle

37 PUBLICATIONS 399 CITATIONS

[SEE PROFILE](#)



Y. H. Wang

Dalian University of Technology

265 PUBLICATIONS 4,946 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Author [View project](#)



Testing and Evaluation of Highly Automated Driving Systems [View project](#)

Advanced framework for microscopic and lane-level macroscopic traffic parameters estimation from UAV video

ISSN 1751-956X

Received on 10th July 2019

Revised 13th February 2020

Accepted on 16th March 2020

E-First on 16th April 2020

doi: 10.1049/iet-its.2019.0463

www.ietdl.org

Ruimin Ke¹, Shuo Feng², Zhiyong Cui¹, Yin Hai Wang¹ ✉¹Smart Transportation Applications and Research (STAR) Laboratory, Department of Civil and Environmental Engineering, University of Washington, USA²Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, USA

✉ E-mail: yinhai@uw.edu

Abstract: Unmanned aerial vehicle (UAV) is at the heart of modern traffic sensing research due to its advantages of low cost, high flexibility, and wide view range over traditional traffic sensors. Recently, increasing efforts in UAV-based traffic sensing have been made, and great progress has been achieved on the estimation of aggregated macroscopic traffic parameters. Compared to aggregated macroscopic traffic data, there has been extensive attention on higher-resolution traffic data such as microscopic traffic parameters and lane-level macroscopic traffic parameters since they can help deeply understand traffic patterns and individual vehicle behaviours. However, little existing research can automatically estimate microscopic traffic parameters and lane-level macroscopic traffic parameters using UAV videos with a moving background. In this study, an advanced framework is proposed to bridge the gap. Specifically, three functional modules consisting of multiple processing streams and the interconnections among them are carefully designed with the consideration of UAV video features and traffic flow characteristics. Experimental results on real-world UAV video data demonstrate promising performances of the framework in microscopic and lane-level macroscopic traffic parameters estimation. This research pushes off the boundaries of the applicability of UAVs and has an enormous potential to support advanced traffic sensing and management.

1 Introduction

Unmanned aerial vehicle (UAV) has demonstrated great potentials for traffic monitoring in a series of transportation studies due to its high flexibility, wide monitoring range, top-view perspective, and low cost, compared to conventional traffic sensors [1–7]. With such benefits, the prospect of UAV for cost-effective and advanced traffic management is brilliant. However, new algorithms and frameworks are needed to turn the data collected by UAV into useful resources. One of the most common UAV sensors is the video camera. While there are many existing algorithms in the area of video-based traffic detection [8], most of them are designed for roadway surveillance cameras at fixed locations [9]. Hence, it is difficult to apply them to UAV video due to the nature of UAV's ego-motion, which causes irregular video background movements.

In preliminary studies, researchers explored the possibilities of using UAV for traffic monitoring with a focus on either UAV videos with static backgrounds [10–13] or some straightforward research tasks (e.g. vehicle detection) that could be completed without handling the UAV ego-motion issue [14–16]. Recently, with the needs and vision for advanced ITS applications and significant progress in video processing techniques, some new studies have been conducted on addressing this issue [17–23]. These achievements in handling UAV ego-motion have laid the foundation for more sophisticated research on UAV-based traffic sensing, where a primary topic is traffic parameters estimation from moving UAV platforms [17, 19–21, 23, 24].

While most previous studies on UAV-based traffic parameters estimation focus on extracting aggregated macroscopic traffic data, there has been extensive attention on higher-resolution traffic data such as microscopic traffic parameters and lane-level macroscopic traffic parameters due to the fact that they can help us deeply understand traffic patterns and individual vehicle behaviours [25, 26]. Despite the recent accomplishments, little existing research realises the estimation of microscopic and lane-level macroscopic traffic parameters from moving UAVs. UAV video, as a cost-effective data source for future transportation, needs to be examined for its availability of providing such data. The key

limitations in the existing research can be summarised into three aspects: (i) instead of making use of long-term temporal information, existing traffic parameters estimation methods for UAV sensing are usually based on two-frame video information, which reduces the reliability and applicability of the application; (ii) there are few processing pipelines to extract sufficient lane information from UAV videos for traffic parameters estimation; (iii) there are few modules for UAV videos to automatically combine and convert vehicle information, lane information, motion information, and traffic flow theories into fundamental traffic parameters.

In this paper, we test the availability of UAV video for microscopic and lane-level macroscopic traffic parameters estimation by proposing an effective processing framework. Specifically, three functional modules consisting of multiple processing streams, algorithms, data structures, and the interconnections among them are carefully designed within the framework with the consideration of UAV video features and traffic flow characteristics [27]. This research pushes off the boundaries of the applicability of UAVs in intelligent transportation systems and has an enormous potential to support applications like advanced traffic sensing, traffic enforcement, and post-disaster traffic management.

The proposed framework targets a critical problem in the field of intelligent transportation systems: increasing the types of fundamental traffic parameters that can be automatically extracted from an emerging sensing platform – UAV. The design of the framework targets balancing efficiency, accuracy, and robustness with a combination of existing algorithms and proposed new algorithms. To our best knowledge, the contributions of this study are summarised as follows:

(i) The development of the framework is among the first efforts to examine moving UAV's availability for microscopic and lane-level macroscopic traffic parameters estimation. Interactions among different modules and streams within the framework are designed to address background motions in UAV video and support reliable traffic parameters estimation.

(ii) A new multiple-vehicle tracking method is specifically developed for UAV videos, which improves the data collection accuracy and enables the estimation of microscopic traffic parameters.

(iii) An efficient and robust lane information extraction pipeline is built to obtain the number of lanes, lane positions, and lane lengths from moving UAV videos.

2 Literature review

The most recent studies in UAV-based traffic surveillance focus on vehicle detection [14–16, 28–34], vehicle tracking [12, 18, 28–30, 32, 34–36], traffic pattern recognition [13, 37], and traffic parameters estimation [3, 10, 17, 19–21, 23–26, 38]. Vehicle detection and vehicle tracking are usually the initialisation process for traffic pattern recognition and traffic parameter estimation. Thus, developing vehicle detectors and trackers with high efficiency, accuracy, and robustness is very important for advanced traffic surveillance tasks. While traditional vehicle detection in UAV videos tended to use background modelling or conventional machine learning with handcrafted features [14, 28–32], more and more studies started to design or implement deep learning based vehicle detectors for UAV surveillance due to the high accuracy of deep neural networks in image classification and localisation [25, 33, 35, 37, 38]. Vehicle detection itself is able to determine traffic parameters like density without the need for motion analysis or vehicle tracking. Zhu *et al.* [38] presented an enhanced single shot multibox detector (Enhanced-SSD) for vehicle detection with their own manually annotated data. It helps collect traffic density with high accuracy.

However, in order for the estimation of other traffic parameters such as speed and volume, motion analysis or vehicle tracking is a must on top of vehicle detection. Great efforts have been made in testing vehicle tracking methods for UAV videos and designing motion analysis frameworks for traffic parameters estimation. Most previous works implemented and tested popular tracking algorithms for UAV videos. Kalman filter, particle filter, and optical flow are the most widely adopted trackers or motion estimators in the existing literature [12, 18, 28, 29, 32, 34]. Among these three, the particle filter is more appropriate than a naïve Kalman filter in real-world traffic scenarios due to the particle filter's non-linearity property [18]. Optical flow, normally Kanade–Lucas–Tomasi (KLT) tracker, is a very flexible tracker and has been widely used in UAV-based applications. Lately, a study compared performances of the state-of-the-art trackers for UAV tracking [36]. Among them, simple online and real-time tracking (SORT), achieves the best overall performance [39] considering tracking accuracy and real-world applications.

Traditional traffic parameter estimation research focused on using static UAV videos to extract macroscopic traffic flow parameters. One of the earliest studies was conducted by McCord *et al.* [10], in which many critical macroscopic traffic parameters were successfully estimated such as annual average daily traffic. Later on, research was conducted to address UAV ego-motion issues and estimate macroscopic traffic parameters at the same time. A pioneering study of this task was proposed by Shastry and Schowengerdt [17]. They developed a method that made use of image registration and motion information to achieve the estimation of basic traffic flow parameters. Recently, Ke *et al.* [20, 21, 23] developed a couple of machine-learning-based methodologies to estimate aggregated traffic flow parameters (speed, density, and volume) and conducted thorough experiments in a variety of scenarios. In addition to macroscopic parameters, because of the needs for higher-resolution traffic data, researchers have been exploring the possibility of UAV for microscopic traffic parameter estimation [25, 26]. For example, Barmponakis *et al.* [25] conducted a study in microscopic traffic parameter estimation from UAV video. In their research, naturalistic trajectory data from UAV video footage at a low-volume intersection and a pedestrian passage is extracted. However, these new studies on microscopic data were still based on static UAV videos. In this paper, we try to study the availability of moving UAV videos for high-resolution

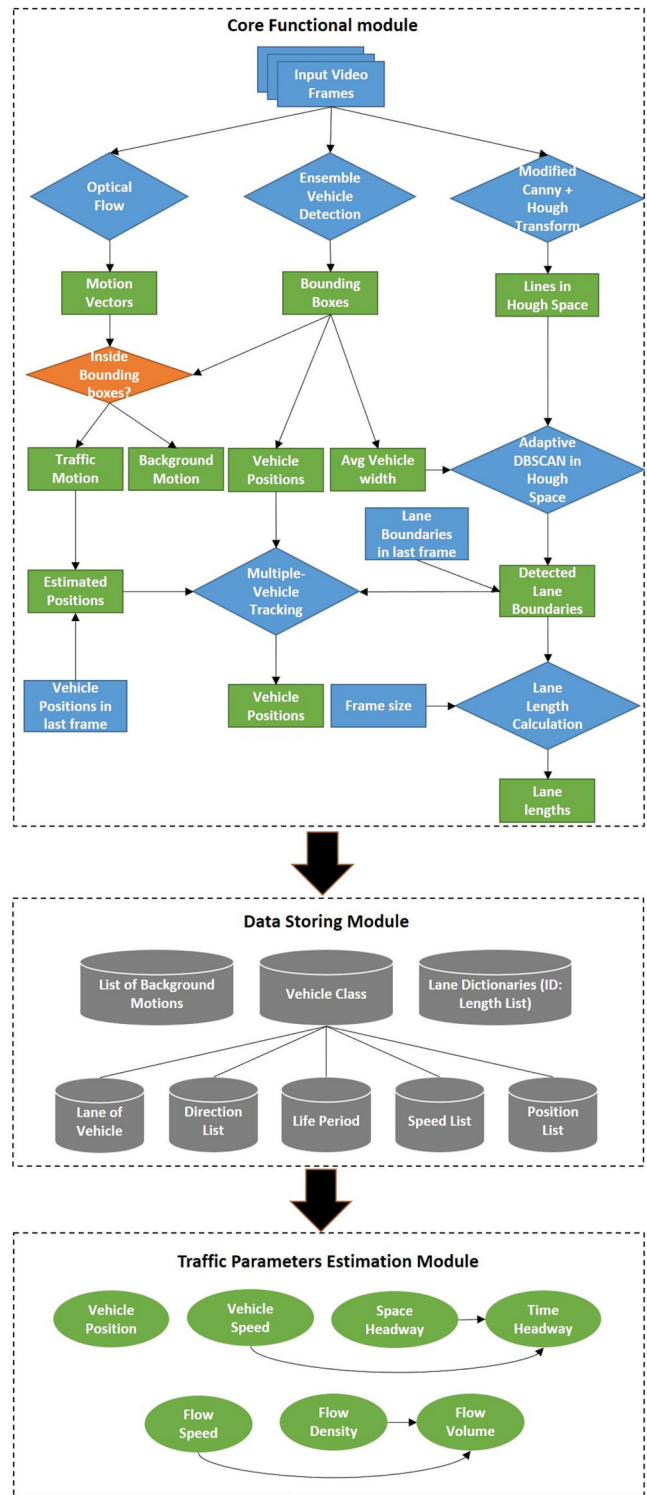


Fig. 1 Proposed framework for microscopic and lane-level macroscopic traffic parameters estimation from UAV videos

traffic parameters estimation (microscopic traffic parameters and lane-level macroscopic traffic parameters).

3 Methodology

3.1 Overview

The proposed framework contains three modules: core functional module, data storing module, and traffic parameters estimation module (see Fig. 1). The core functional module has three processing streams for motion-vector processing, multiple vehicle detection/tracking (see Figs. 2 and 3), and lane information extraction (see Figs. 3 and 4), respectively. Each of them functions as an independent stream, but at the same time, has

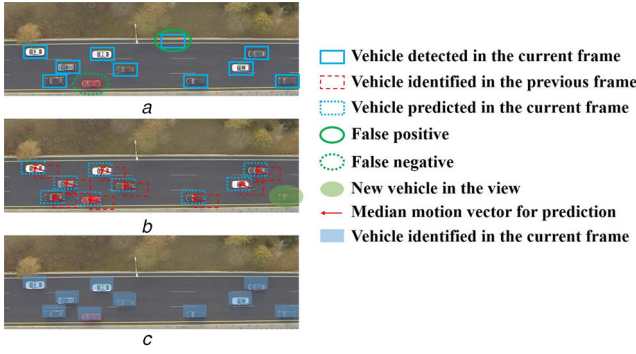


Fig. 2 Diagram of the proposed multiple-vehicle tracking algorithm
(a) Vehicle detection, (b) Vehicle prediction, (c) Vehicle association

Input: detected vehicle list *det_list*, predicted vehicle dictionary *pre_dic*,
Output: identified vehicle dictionary *veh_dic*

```

for pre in pre_dic:
    for det in det_list:
        if the IoU for pre and det is larger than threshold_IoU:
            The position pos for vehicle pre_key is calculated as the
            averaged position of pre and det
            veh_dic appends [pre_id, pos] as an identified vehicle

    if no detection is found to be associated with pre:
        if pre is out of video frame bounds:
            Skip adding pre to the active vehicles veh_dic
        elif pre was not associated with any det in the past five frames:
            An FP is found, stop tracking pre
        else:
            An FN is found, veh_dic appends pre as an identified vehicle

for det in det_list that has not been associated with any prediction:
    if det is within the bounds of the roadway & det is within a certain
    distance to video frame bounds:
        new_id is assigned to det
        veh_dic appends [new_id, det] as a new vehicle into the UAV
        view
    else:
        An FP det is found, so det will be discarded.

```

Fig. 3 Pseudocode of the association step of the proposed multiple-vehicle tracking algorithm

Input: Lane boundary representations in the Hough space with parameter *theta* and *rho*, video frame height *ht*, and video frame width *wt*
Output: The length of the lane *len*

```

For every lane L:
    1) Assume it is bounded by two Hough lines,  $\rho_{h1} = x * \cos(\theta_{h1}) + y * \sin(\theta_{h1})$  and  $\rho_{h2} = x * \cos(\theta_{h2}) + y * \sin(\theta_{h2})$ 
    2) Let  $x = 0$  and  $x = wt$  in the first Hough line equation, get two y values  $y_{11}$  and  $y_{12}$ 
    3) If  $y_{11} \in [0, ht]$ , one end point of lane boundary #1 is  $P_{11} = (0, y_{11})$ ; otherwise, there must be  $y_{12} \in [0, ht]$ , thus  $P_{11} = (wt, y_{12})$ 
    4) Let  $y = 0$  and  $y = ht$  in the first Hough line equation, get two x values  $x_{11}$  and  $x_{12}$ 
    5) If  $x_{11} \in [0, wt]$ , the other endpoint of lane boundary #1 is  $P_{12} = (x_{11}, 0)$ ; otherwise, there must be  $x_{12} \in [0, wt]$ , thus  $P_{12} = (x_{12}, ht)$ 
    6) Repeat 3) – 5) to get the endpoints  $P_{21}$ ,  $P_{22}$  for lane boundary #2.
    7) Calculate the length of the lane as  $len = \frac{1}{2}(\|P_{11} - P_{12}\|_2 + \|P_{21} - P_{22}\|_2)$ , where  $\|\cdot\|_2$  measures the Euclidean norm of a vector

```

Fig. 4 Calculation process of the lane lengths in every frame

interconnections with one another. This design enables the automatic and robust extraction of multiple types of traffic parameters. In the core functional module chart of Fig. 1, the rectangular green boxes represent the outputs that will be further processed within the framework.

The data storing module takes outputs from the core functional module as its inputs, and stores data into different data structures. The traffic parameters estimation module takes the organised data from the data storing module and estimates seven key traffic parameters. Based on traffic flow theories and the proposed preceding-vehicle determination process, the traffic parameters estimation module takes the organised data from the data storing module and estimates four key microscopic traffic parameters including vehicle position, vehicle speed, space headway, and time headway, and the three lane-level macroscopic traffic parameters including traffic flow speed, traffic flow density, and traffic flow volume.

3.2 Core functional module: multiple vehicle detection and tracking

The core functional module has three main functional streams: (i) extracting and grouping motion vectors, (ii) detecting and tracking multiple vehicles, and (iii) extracting lane information (see Fig. 1). In this subsection, we introduce our new multiple-vehicle tracking algorithm under the context of UAV traffic sensing. This algorithm is designed for vehicle tracking in moving UAV videos. It follows a detection-based tracking pipeline, which is composed of three steps: detection, prediction, and association. It not only achieves a high tracking rate but also improves vehicle detection precision and recall rates by effectively filtering out false positives (FPs) and false negatives (FNs). How the proposed tracking pipeline reduces FPs and FNs are introduced later in this section, and you can use Fig. 3 for quick reference.

Fig. 2 is the diagram showing the key steps of the proposed detection/tracking algorithm. In the detection step, the state-of-the-art ensemble vehicle detector [23] is implemented to find the locations and sizes of vehicles from a top-view perspective. The solid blue rectangular boxes in Fig. 2a show the vehicle detection results in a frame. The dashed red boxes in Fig. 2b show where the vehicle bounding boxes are in the previous frame. KLT optical flow tracker [40], which has demonstrated its great performance in UAV-based vehicle motion prediction, is selected as the motion prediction tool in the second step of the tracking. Optical flow motion vectors in a frame are classified into two groups: traffic motion vectors and background motion vectors, with the detected vehicle bounding boxes as the region classifier. Each vehicle's predicted motion is calculated by adding the median of the motion vectors in its bounding boxes to its position in the previous frame (shown as dashed blue boxes in Fig. 2b).

The third step is the association of detections with predictions. In order to better illustrate our association algorithm, the primary processes are summarised as pseudocode in Fig. 3. Detected vehicle boxes and predicted vehicle boxes are the two inputs to the association algorithm. The detected vehicle boxes are stored in a data list, and the predicted vehicle boxes are added to a data dictionary as the values. The dictionary's key is vehicle ID. The algorithm outputs another data dictionary which has the same structure with the predicted vehicle dictionary *pre_dic*, where the key is the vehicle ID and the value is the vehicle bounding box. This dictionary is denoted as *veh_dic* for the identified vehicles, which contains all the updated vehicle positions in the current frame. Intersection-over-Union (IoU) is used to determine whether every prediction box can find a match in the detected boxes list. IoU is selected as the indicator for the association because of its appropriateness for UAV-based vehicle tracking. The appropriateness comes from (i) its low computational complexity, and (ii) the top-view perspective of UAV, which ensures little overlap between any two vehicles. If a prediction matches any detection, the new position of the vehicle, *pos*, in the current frame, is obtained by averaging the positions of its detected boxes and predicted boxes. Then the vehicle dictionary *veh_dic* will be updated by appending *pos*.

In the current frame, sometimes, there could be no detected vehicle box associated with a predicted vehicle box. This situation can be divided into two cases. In the first case, the vehicle appearing in the previous frame is outside of the UAV camera view in the current frame, thereby it will not be added to the latest

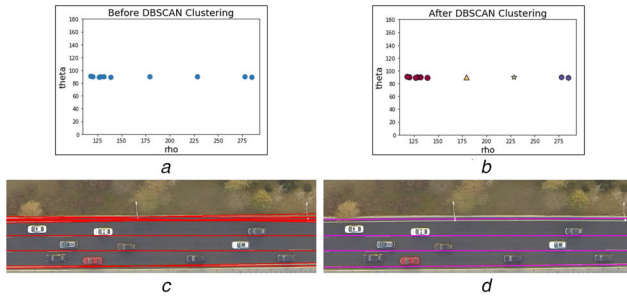


Fig. 5 Adaptive DBSCAN algorithm to cluster redundant Hough lines in the Hough space for traffic lane boundary detection
(a) Displays the corresponding Hough lines detected in (c), (b) Presents the clusters in the Hough space, (d) Shows the post-clustering results in the video

veh_dic. In the second case, the vehicle is still in the camera view, but it is not detected. In other words, it is an FN. Hence, the prediction itself will be taken as the updated vehicle position and appended to *veh_dic*. An FN example is shown in Fig. 2a with a dashed green ellipse. So far in the association algorithm, all the predictions from the previous frame's identified vehicles have been processed. However, in the detection list *det_list*, those that are not associated with any predictions still need more examination. They could be either FPs from the ensemble vehicle classifier or new vehicle appearing in the camera view. We design a rule to determine whether it is an FP or new vehicle with the interaction of the lane information extraction stream: If the centre of a detected box is inside the bound of the roadway (which is determined at the same time by the lane extraction stream) and it is within a certain distance to any of the video frame bounds, it is regarded as a new vehicle. Otherwise, the detection is discarded as an FP. A new vehicle will be assigned a new vehicle ID and added to *veh_dic* along with its detected box. In Fig. 2a, an example of a new vehicle is masked with a green translucent rectangle, and an FP is with a solid green ellipse. Another case of identifying an FP is that: if a bounding box is not associated with any vehicle detected by our detector [23] in the past five frames, it will be deactivated. This operation reduces the possibility of tracking a non-vehicle that is falsely detected within the roadway region. In Fig. 2c, it shows the final vehicle tracking results with vehicles correctly identified.

3.3 Core functional module: lane information extraction

Based on the characteristics of UAV videos, we develop a traffic lane information extraction method as one stream in the core functional module. It is composed of four main steps: (1) modified Canny edge detection and Hough transform, (2) adaptive DBSCAN for redundant line filtering, (3) lane boundary tracking in Hough space, and (4) lane length calculation. The first step is based on a traditional line detection process that applies Canny edge detection and Hough transform [41, 42]. Given that most traffic lane boundaries are within a specific colour range (in general, white, or yellow), we add colour information to the edge detection process besides the gradient information. This can filter out pixels that have large gradients but are not from the lane boundaries. The proposed lane detection method is essentially a combination of a traditional yet effective line detection algorithm and a proposed adaptive unsupervised learning algorithm, with several other rules. We do not use deep learning features here in the lane information extraction with the consideration of processing speed for UAV video analytics.

In the second step, we develop a clustering algorithm for redundant line filtering. This design is based on the problem that a single traffic lane boundary is sometimes associated with multiple detected Hough lines. These boundaries are either double lines or so thick that they are detected as multiple lines. Thus, a clustering method would be a natural solution to the multi-line problem. DBSCAN, which is based on the density of elements, can detect the number of clusters (i.e. the number of traffic lane boundaries) [43]. Therefore, it is selected over other clustering methods for this problem. Two parameters are critical in DBSCAN: (i) *minPts*, the minimum number of samples in a cluster, and (ii) *eps*, the

maximum distance between two samples for them to be considered as in the same cluster. The *minPts* is fixed to 1 in the proposed framework since some lane boundary may be just associated with one detected Hough line. Another parameter, *eps*, should be adjusted adaptively in each frame. Note that from a top-view perspective, lanes' widths are supposed to be identical, which means the distances between each pair of neighbouring clusters should be identical. In practice, *eps* should be set a little smaller than the lane width to avoid grouping adjacent lane boundary lines into one cluster. Also, lane width may constantly change throughout the video clip due to UAV height changes. In every frame, *eps* is set equal to the average vehicle width (in the unit of a pixel), which is determined by our vehicle detector. This specific design of settings takes advantage of the UAV perspective as well as the use of the vehicle detector, and it turns out to work very well. An example is shown in Fig. 5.

Following the redundant line filtering, a lane boundary tracking process is developed in the Hough space by combining detection and prediction results. As aforementioned, motion vectors are grouped into traffic motion and background motion by the vehicle bounding boxes. The predicted motions of the lane boundaries are obtained from the background motion vectors. The detected lines are stored in a data list, and the prediction is made by adding the background motion in the current frame to the lane boundary list of the last frame. In the case any lane boundary is not detected in the current frame, the predicted position will be added to the lane list for this frame. Otherwise, the updated position for a lane boundary will be calculated as the average of its predicted and detected positions. The tracking process further improves the reliability of lane boundary detection and ensures the robustness of the whole system.

Then, we calculate the lanes' lengths. This process is necessary for traffic density calculation. With the traffic lane boundaries extracted, each lane will be assigned a label according to their distance to the origin of the frame. Suppose a traffic lane is bounded by two Hough lines in (1),

$$\begin{aligned}\rho_{01} &= x * \cos(\theta_{01}) + y * \sin(\theta_{01}) \\ \rho_{02} &= x * \cos(\theta_{02}) + y * \sin(\theta_{02})\end{aligned}\quad (1)$$

where ρ_{01} , ρ_{02} , θ_{01} , θ_{02} are the Hough line parameters. Then, with the video frame's dimension known, we can determine the lane length within the frame based on straightforward geometric calculations. The detailed process is shown in Fig. 4. The length in the unit of a pixel will be converted into physical length using a conversion rate r , which is updated in every several frames using a robust online method [23]. In this method, the initial value of r at the beginning of a video needs to be manually inputted by measuring the length in pixels of an object with a known dimension. Then, this conversion rate is updated in an online manner using vehicle bounding boxes and the t -test process.

3.4 Data storing module

In this module, data structures and functions are built to take the outputs of the core functional module and store information for a whole UAV video clip. There are three low-order data structures in this module, which are (1) a data list of video background motion at every timestamp, (2) a vehicle class that stores each individual vehicle's information, and (3) lane dictionaries in which the key is a lane ID and the value is a list of lane lengths at every timestamp. At the high order, the vehicle class contains five different types of vehicle data: (1) a list of IDs showing which lane the vehicle is in at every timestamp, (2) a list of angles showing the vehicle's moving directions, (3) the life period of the vehicle in the video view, (4) a list of speed values of the vehicle, and (5) a list of coordinates showing the positions of the vehicle.

These data will be further used for the estimation of both microscopic and macroscopic traffic parameters. In the lane length calculation process, the instantaneous lane information in a frame is recorded in a dictionary, where the dictionary key is the lane ID and the dictionary value stores a list of lane length values. The background motion list is also updated for every frame by

appending the median background motion vector of that frame. When a new vehicle appears in the UAV camera view, a vehicle object will be constructed with a new vehicle ID. Meanwhile, several empty lists representing the position, speed, moving direction, life period, and lane information variables will be created. The vehicle association algorithm shown in Fig. 3 outputs every vehicle's position in every frame as a 1D list with a fixed length of four (four corners of the bounding box), and it will be appended to the 2D position list in the corresponding vehicle class. To calculate the instantaneous speed of a vehicle, the data storing module records the bounding box's centre C_{cur} of the vehicle for the current frame, the bounding box's centre C_{pre} of the vehicle for the previous frame, the instantaneous background motion vector M_b , the video frame rate fr , and the conversion rate r of pixel length to physical length. Hence, the instantaneous ground speed v_{veh} and moving direction dir_{veh} for a vehicle is calculated in the following equation:

$$\begin{cases} v_{veh} = \|C_{cur} - C_{pre} + M_b\|_2 \times fr \times r \\ dir_{veh} = \tan^{-1}(C_{cur} - C_{pre} + M_b) \end{cases} \quad (2)$$

where $\|\cdot\|_2$ is the Euclidean norm of a vector, and $\tan^{-1}(\cdot)$ is the arctangent function. The speed will be appended to the 1D speed list of the corresponding vehicle object. To determine the instantaneous lane information for a vehicle, we compare the vehicle's centre C_{cur} with the lane boundaries to see which lane the vehicle is in. Then we append the lane ID to the 1D lane list of this vehicle. Note that the 'life period' variable in the vehicle class is a two-element list that records when the vehicle enters and exits the UAV view.

3.5 Traffic parameters estimation module

Speed, density, and volume are the most critical three macroscopic traffic parameters. Equation (3) describes the fundamental relationship for them:

$$q = v \times k \quad (3)$$

where q , v , and k denote volume, speed, and density, respectively. The macroscopic parameters can be aggregated for either each lane or all lanes automatically in our framework since we extracted the lane information. Traffic density can be calculated from vehicle counts and length of the roadway. In this paper, we mainly discuss the estimation of macroscopic parameters for each lane, since this process has not been automated so far in existing UAV-based studies. We have recorded lane lengths as well as the vehicle information for each frame. Therefore, it is feasible to read the data from the data storing module and calculate lane-level density. In any frame, assuming N_t vehicles are identified on lane t , and the length for lane t is len_t , then according to the definition of traffic density, (4) computes the density k_t for lane t .

$$k_t = \frac{N_t}{len_t} \quad (4)$$

The instantaneous macroscopic speed of lane t is obtained by calculating the space mean speed of all vehicles on lane t , which is represented by (5). The volume of each lane is estimated using (3).

$$v_t = \frac{N_t}{\sum 1/v_{veh}} \quad (5)$$

Microscopic traffic parameters depict the behaviours of individual vehicles at a particular timestamp. Typical microscopic traffic parameters include vehicle position, vehicle speed, space headway, and time headway. The vehicle position and vehicle speed are already stored in the vehicle class. We denote the vehicle position list and vehicle speed list pos and spd , respectively, and the starting frame of this vehicle is fr_{start} . Thus, (6) yields the instantaneous

position pos_i and speed spd_i of vehicle V_i from the data storing module in any frame fr_{cur} ,

$$\begin{cases} pos_i = pos[fr_{cur} - fr_{start}] \\ spd_i = spd[fr_{cur} - fr_{start}] \end{cases} \quad (6)$$

where $list[n]$ is the $(n+1)$ th element of the list. To extract a vehicle's space headway and time headway in frame fr_{cur} , the first step is to find out the preceding vehicle of the target vehicle. Mathematically, vehicle V_1 is followed by vehicle V_2 if constraints in (7) are all satisfied:

$$\begin{cases} lane_1 = lane_2 \\ |\tan^{-1}(pos_1 - pos_2) - dir_2| < \pi \\ |pos_1 - pos_2| = \min(pos_{i,down} - pos_2) \end{cases} \quad (7)$$

where the first constraint ensures the lane labels of V_1 and V_2 are the same (i.e. on the same lane), and the second constraint ensures that V_1 is downstream the target vehicle V_2 , and the third constraint ensures V_1 is closer to V_2 than any other downstream vehicle $V_{i,down}$.

With this searching process, it is then possible to compute the headways of the target vehicle V_2 . Assuming the space headway of V_2 is H_{space} , and time headway H_{time} , (8) yields the results of H_{space} and H_{time} in a frame

$$\begin{cases} H_{space} = \|pos_1 - pos_2\|_2 \times r \\ H_{time} = \frac{H_{space}}{spd_2} \end{cases} \quad (8)$$

where r and $\|\cdot\|_2$ are the conversion rate and the Euclidean norm mentioned above. It is worth noting that instantaneous headways are not available for the leading vehicle in any lane due to that the vehicle it follows is out of the UAV view. This could be solved by operating multiple UAVs, however, it is out of the scope of this study. With the microscopic traffic parameters extracted from UAV video, microscopic behaviours such as car following and lane changing can be observed and analysed.

4 Experimental results

4.1 Datasets and preliminary results

In total, over one hour of video clips in various traffic scenarios were collected to test the proposed methodology, which was implemented in Python language. The key extra libraries that were configured for the implementation were OpenCV for image processing and Keras for detector training with Tensorflow backend. Generally, the overall performance was promising that few false detections or missed vehicles showed up. First of all, we carefully examined 600 representative video frames to validate the effectiveness of the methodology. In this video clip, the UAV had irregular background movement patterns including rotation, cruising, and vibration. It also contained other challenging scenarios such as where the UAV was flying over a highway sign gantry that blocked vehicles from the top.

Fig. 6 shows some example output frames, in which traffic lane boundaries and vehicles are marked with lines and bounding boxes. Vehicles from different lanes are differentiated by bounding box colours, with their vehicle IDs on the left-top of the box. The highway sign gantry can be observed in the first three example frames with vehicles partially blocked. It can be seen that most of these blocked vehicles were well tracked and identified by our method. Motion-vectors were marked as short green/brown arrows in the example frames. The green arrows were outside detected bounding boxes for background motion-vectors extraction, and brown arrows were inside the bounding boxes for vehicle motion extraction. Then, another four videos will be added for further analysis.

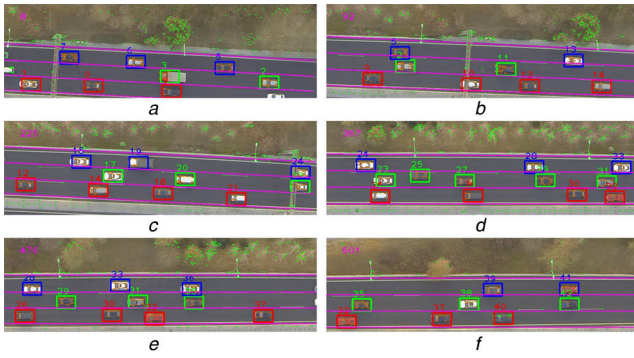


Fig. 6 Example output frames showing the preliminary results in a UAV video with a moving background. Motion vectors, lane boundaries, and vehicles are marked in the video frames (a) Frame #8, (b) Frame #92, (c) Frame #225, (d) Frame #367, (e) Frame #476, (f) Frame #601

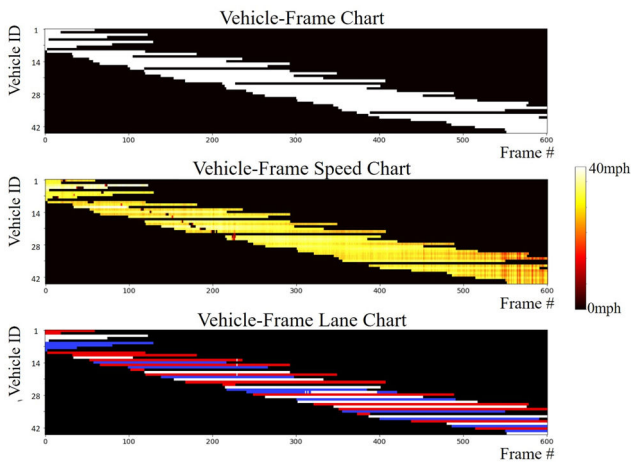


Fig. 7 These are the vehicle-frame charts showing some of the microscopic information of all vehicles' in 2D matrices, and in each chart, the x-axis denotes the video frame number and the y-axis the vehicle ID. White or colourful pixels mark vehicle appearances or certain parameter values

4.2 Microscopic traffic parameters estimation and analysis

Figs. 7–9 present the estimation results of microscopic traffic parameters. Fig. 7 contains three charts, each of which displays one type of extracted parameter for all vehicles in a single chart. We name the charts as ‘vehicle-frame charts’, which are formed by two dimensions: one is the frame number, or in other words, time dimension, and another is the vehicle ID. The vehicle-frame charts are literally matrices that are produced by stacking each vehicle's 1D microscopic parameters throughout the whole video together. These charts record the number of vehicles, vehicles' life periods, and some microscopic parameters for any given vehicle at any timestamp in a clear and straightforward way.

The top vehicle-frame chart in Fig. 7 records all vehicles' life periods using a binary image, where the white part illustrates a case that a vehicle with a certain ID is within the UAV view at a certain frame. In the middle of the figure, the second vehicle-frame chart gives an impression of the speed information, where the lighter colour corresponds to a higher speed. The vehicle-frame lane chart at the bottom shows each vehicle's lane information. There are three colours (red, blue, white) in this chart and each of them represents one of the three lanes.

A better understanding of traffic patterns and vehicle behaviours is likely to be achieved by properly utilising these vehicle-frame charts produced by our system. For example, in the vehicle-frame speed chart, traffic speed patterns can be visualised and give us an intuition on when and how the speed changes. In Fig. 7, we can observe speed oscillations at around the end of the period for vehicles on all lanes (the vehicles' corresponding lane information is shown in the third chart of Fig. 7). With this

observation, we know that the traffic has a trend to slow down, given the knowledge in traffic flow characteristics and congestion formation. From these charts, we see the distribution of vehicles on each lane, and the relationship between traffic speed and lane-changing behaviours (lane #, timestamps, durations, etc.) can also be roughly visualised.

Detailed microscopic parameters for individual vehicles can be visualised in Figs. 8 and 9. Vehicle positions, speeds, and lane information of three representative vehicles throughout the experimental video are displayed in Fig. 8. The position subfigures are the same size as the video frame, thus to display vehicle trajectories in the video. Video frame numbers are the x-axes for speed subfigures and lane subfigures, and y-axes are labelled as mph for speed subfigures and lane # for lane subfigures, respectively. Note that the traffic moves from the right of the frame to the left.

The vehicle with ID #9 is one of the vehicles appearing in the very beginning of the video recording. Therefore, its starting position is in the middle with respect to the x-axis instead of the rightmost. The speeds for vehicle #9 throughout the video are almost 30 mph and it stays in lane #3 for its life period. Vehicle #13 enters the UAV view at about frame #60 and leaves at about #218. From the speed curve, a speedup can be observed in its life period from about 30 to 35 mph, and it can be seen this vehicle stays at lane #1 with no lane changing behaviour. Vehicle #25 enters the view at frame #246, and its speed remains relatively constant with a very small decrease after frame #350.

In general, vehicles #13 and #25 have higher speeds than vehicle #9 during their life periods. Lane changing actions can be observed from the lane information curve of vehicle #25, it moves from lane #1 to lane #2, and then returns to lane #1 after about 80 frames. It is noticed that there are two small peaks in the lane subfigure of this vehicle (#25) for its first-time lane changing from lane #1 to lane #2. This is due to detector estimation errors of the vehicle's location. There are always errors in the vehicle location (or bounding box) estimation, though most time it is small with a good detector and cannot be visualised. However, in a lane changing situation, there is time the vehicle's geometric centre is right on the boundary of two adjacent lanes. And that is when minor localisation error causes the ‘peaks’, or in other words, the vibration pattern in the figure.

As aforementioned, space headway and time headway are key traffic microscopic parameters and can be automatically given out by the proposed framework for traffic data collection and behaviour analysis. An example is shown in Fig. 9, where four vehicles (with vehicle ID 30, 32, 37, and 40) are following one another in the same lane. We analysed sixty frames from #518 to #577 from vehicle #40 entered the view to vehicle #30 left the view. Seven subfigures are included in Fig. 9, where the first one is an original frame with the four target vehicles marked by bounding boxes and vehicle IDs, and the other six subfigures show the headway curves for the 60-frame period. The first row includes three space headway curves and the second row includes three time-headway curves. It can be observed from the video frame in Fig. 9 that vehicle #32 follows closely to vehicle #30, and the other two vehicles have relatively larger following distances.

From the space headway curves, it can be seen vehicles #32 and #30 are within 10 m, which is a short distance given the traffic speed is around 30 mph. The space headway of vehicles #37 and #32 is larger than 20 m, which is the largest of the three. The space headway of vehicles #40 and #37 starts at about 20 m, but an apparent decreasing trend can be observed. In fact, all three space headway curves have decreasing trends. It is inferred that the leading vehicle has a slower cruise speed than average traffic flow speed during that period. This inference has been proved correct based on the extracted speeds of these vehicles. The time headway curves provide additional information to the car following behaviours. Among all three time-headway curves, the first one generally has the smallest values, which makes sense because the associated space headway is much smaller than the others. An interesting observation is that the time headway for vehicle #37/#32 is basically much larger than that for vehicle #40/#37, even though the associated space headways are not that different.

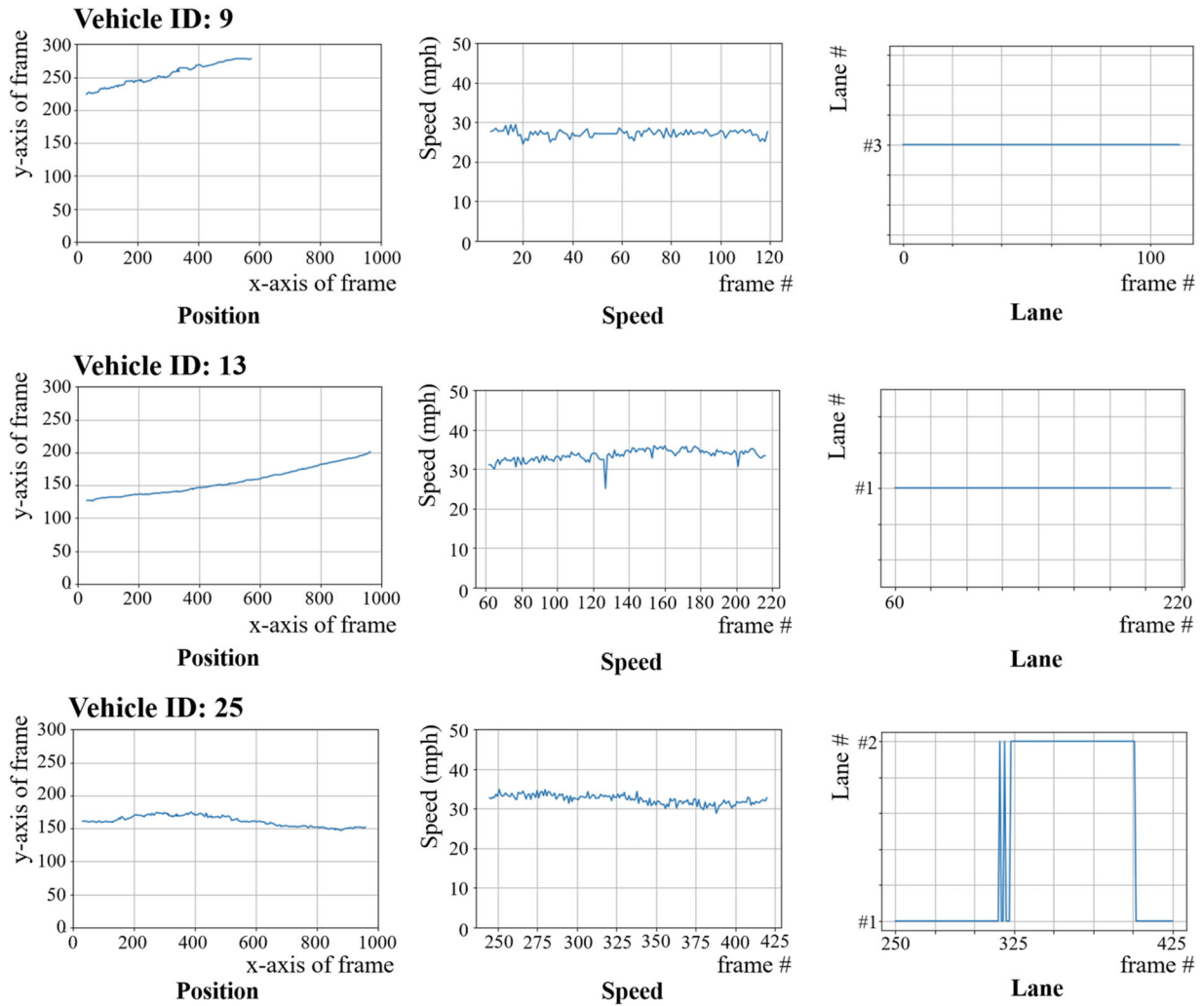


Fig. 8 Examples of individual vehicle tracks, speeds, and lane information extracted using the proposed framework



Fig. 9 Space headway and time headway extraction and car following behaviour analysis

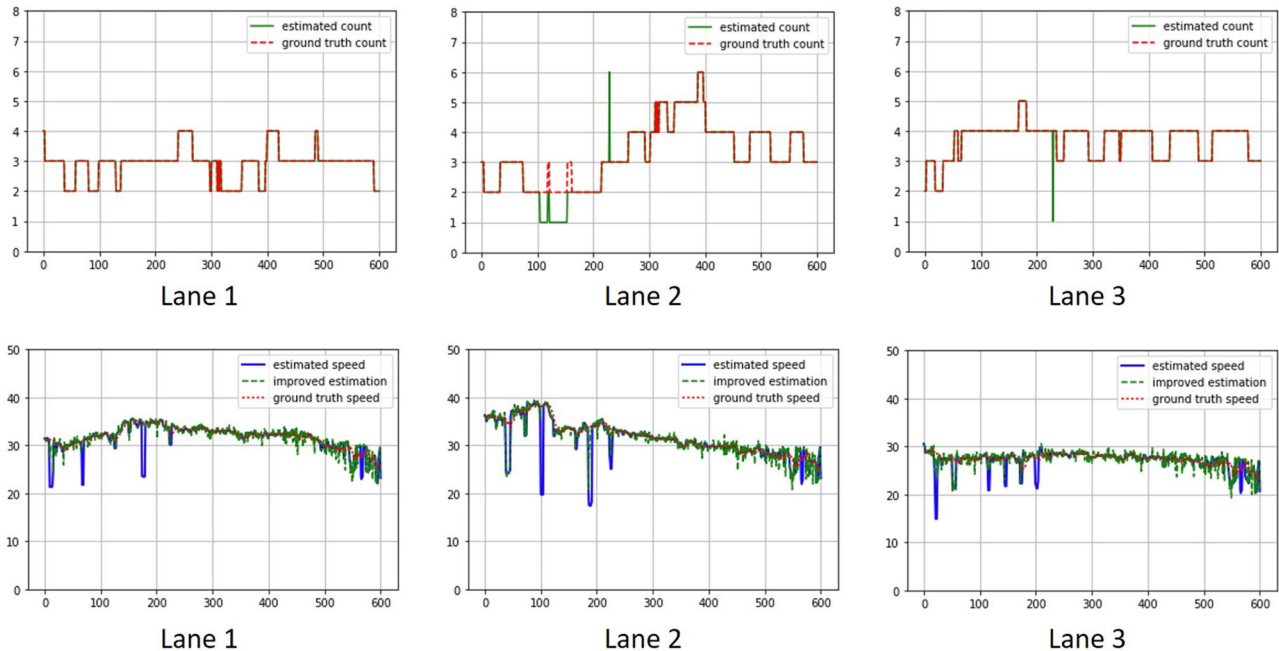
This means that, besides having a larger acceleration rate, vehicle #40 also has a larger speed than vehicle #37. In terms of the drivers' car following behaviours, these observations and analyses lead to the finding that vehicles #32 and #40 are more aggressive than vehicles #30 and #37.

4.3 Macroscopic traffic parameters estimation and analysis

Since reporting instantaneous macroscopic traffic parameters frame by frame is not meaningful in practice, the averaged traffic speeds, densities, and the volumes of all frames are computed and displayed in Table 1 for every and each lane. In this table, we also show the speed estimation accuracies and vehicle count estimation accuracies. Despite a variety of traffic parameters given out by our method, the most fundamental metrics for evaluation of the system

Table 1 Macroscopic traffic parameter estimation results and overall system accuracy evaluation

	Lane #1	Lane #2	Lane #3
estimated speed, mph	32	33	28
estimated density, pc/mi/lane	26	33	38
estimated volume, pc/h	905	1200	1155
speed estimation accuracy, %	97.3	96.6	96.6
count estimation accuracy, %	100	95.4	98.8

**Fig. 10** Accuracy curves for lane-level vehicle counting and traffic flow speed estimation

performance are lane estimation, vehicle speed, and vehicle counting. The summarised overall evaluation results are shown in Table 1 and Fig. 10. Throughout the entire video, no FP was generated, no vehicle was falsely identified with multiple labels, and very few FNs were generated. The only vehicle identified as an FN was the missed tracking for one vehicle on the middle lane from frames #106 to #161. This vehicle was missed after passing through the gantry. It was found that this missed vehicle had a relatively small size and similar colour to the road surface. Hence, when it was blocked by the sign gantry, only a small portion of the motion-vectors extracted was truly eligible for vehicle tracking; at the same time, the vehicle detector missed this vehicle as well due to the vehicle pattern misclassification caused by both the obstruction and the colour issue. This kind of scenario is not common since it occurs only when there are significant errors in both the vehicle detection and vehicle tracking processes. A possible solution is to add a shape estimation on the top of the motion estimation.

In Fig. 10, the red dotted curves are the ground truths, and others are estimations. We can see the vehicle counting estimation for lane #1 fits perfectly to the ground truth. For lane #2, the only missed vehicle (frames #106–#161) contributes to the major part of estimation error. Only 1 out of 600 frames has an incorrect lane boundaries estimation, and this frame number is 230. In Fig. 10, a sudden count increase for lane #2 and a sudden count decrease for lane #3 can be seen at frame #230. While the vehicle detection/tracking works well, three vehicles on lane #3 are grouped into lane #2. This kind of misclassification could happen due to different reasons. But since this problem is rare, it does not have a big influence on any of the parameter estimations. The vehicle counting accuracies for lanes #2 and #3 reach 95.4 and 98.8%. Like in previous studies, the ground truth speed data was manually collected with an on-screen pixel measurement tool [21, 23]. Every individual vehicle's speed has been measured throughout the video, and the mean accuracies are calculated by comparing them with the estimated values. According to Table 1, the speed estimation

accuracy tends to have a positive correlation with vehicle counting accuracy, since the speed accuracies are 97.3, 96.6, and 96.6% for the three lanes.

From Fig. 10, it can be seen that the original speed estimation curves (solid blue curves) contain some significant underestimations of the speeds, which appear like spikes. The spikes show up when there are vehicles right under the highway sign gantry. As aforementioned, only one vehicle was missed after passing through the gantry, but the blockage of UAV view did produce errors for the motion estimation since some motion-vectors on the gantry were inside of the vehicles' bounding boxes. To address this issue, we adopt a straightforward method to filter out those spikes, in which a spike is detected if it is significantly smaller or larger than its nearest neighbours. Specifically, every point on the original speed curve is compared with the median value of a 15-frame long time series from seven frames before it to seven frames after. If it is significantly smaller or larger than the median, its value would be replaced by the median value. We use 15 as the length here because we consider 15 frames are short enough (around 0.6 s) to consider the traffic speed constant and long enough to filter out the sudden changes.

4.4 More validation

To further validate our proposed framework, we added another four UAV videos which were taken in different scenarios (see Fig. 11). Video #2 was collected over a seven-lane urban freeway segment, where the right-most two lanes were ramp lanes connected to the mainstream freeway. Video #3 was a two-lane roadway in a rural area, where the traffic was not as dense as in other videos. In this video, the UAV was hovering over the road segment so that the video background was stationary. The camera view of Video #4 covered six lanes on a two-way road segment. For two-way traffic, note that one minor modification was needed in the macroscopic traffic parameter estimation process: instead of aggregating all vehicles' data, it was necessary to first determine which direction

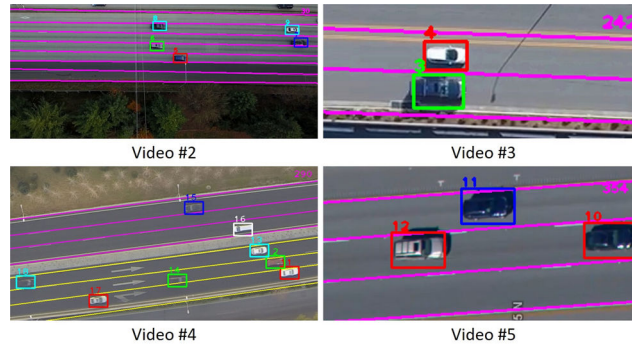


Fig. 11 Four more UAV videos are collected for validating the proposed framework

Table 2 Validation and analysis of five different videos

	Video #1	Video #2	Video #3	Video #4	Video #5
num of lanes	3	7	2	6	3
direction	one	one	one	two	one
UAV motion	medium	slow	zero	medium	fast
camera zoom	no	yes	no	no	yes
speed acc. %	96.8	94.3	96.5	95.7	88.1
count acc. %	97.9	93.1	98.3	96.0	94.4

Table 3 Multiple vehicle tracking comparison with two state-of-the-arts (all based on the same vehicle detector in [23])

		Particle [18]	SORT [39]	Ours
Video #1	FP	5	4	0
	FN	178	130	56
	IDS	10	9	2
	FPS	12	16	13
Video #2	FP	24	18	10
	FN	246	182	95
	IDS	12	9	3
	FPS	10	15	11

Bold values indicates the best performance among the three tracking algorithms.

the vehicles moved towards using their velocities. Video #5 was a video clip we got from the Washington State Patrol Aviation. The key feature of this video was that it was collected by a manned aircraft rather than a UAV. Therefore, the camera movement was faster than in the other four cases.

Table 2 summarises the properties of the five test videos and the performances of our method on them. The overall estimation accuracy is high and stable. It can be seen that the five videos are different and representative. The number of lanes in the five videos ranges from two to seven, which covers most cases in the real world. We also have one video (Video #4) that monitors a two-way road. The purpose of having this video was to test the applicability of our method in extracting traffic information from multi-directional roads. The UAV motion patterns in the test videos also vary. We notice that the moving speed of the aircraft has some impacts on the performance. We can see that the overall speed estimation accuracy for Video #5 is significantly lower than the others. By looking at this video, it is found that the fast speed of the aircraft creates extra challenges in traffic speed estimation because large motions often lead to reduced motion estimation accuracy for optical flow and object tracking. We also evaluated the camera zooming cases. In Videos #2 and #5, there are some camera zooming cases while others not. We do not observe significant performance differences between the time around camera zoom and the remaining time period. This result is encouraging and reasonable because of our vehicle detector and adaptive DBSCAN algorithm. The vehicle detector can detect vehicles in multiple scales, which ensures the detection accuracy in zooming cases; and the adaptive DBSCAN algorithm in the lane detection is proposed to handle camera zooms and UAV height

changes with its adaptive *eps* value. Please note that a sample video for demonstration purposes has been uploaded to this link: https://www.youtube.com/watch?v=1J_3R303r5w.

4.5 Comparison with state-of-the-arts

In this subsection, we compared the vehicle tracking algorithm and the entire framework with the state-of-the-arts. Table 3 shows the comparison of our vehicle tracking algorithm with two state-of-the-art tracking methods that have shown their exceptional ability for UAV-based vehicle tracking [18, 36, 39]. Four widely used metrics for object tracking were picked: False positive (FP), false negative (FN), identity switches (IDS), and frames per second (FPS). FP, FN, and IDS are for accuracy evaluation and FPS are for efficiency evaluation. As aforementioned, FP is the false detection of vehicle, FN is the missed vehicle. In our context, IDS measures the times that a vehicle's ID switches. It usually happens when a vehicle is missed by the tracker and then re-detected with a new ID. Note that ID switches are less common in UAV videos than traditional traffic surveillance videos due to fewer occlusions in UAV videos given their top-view perspective. The smaller the values of FP, FN, and IDS are, the tracking performance is better.

As presented in Table 3, the proposed vehicle tracking algorithm outperforms the state-of-the-arts in FP, FN, and IDS. The numbers in the table are in the unit of vehicle-frame (except FPS), which is the sum of the number of FP/FN/IDS in every frame. In video #1, 0 FPs, 56 FNs, and 2 IDSs occurred for our algorithm. The 56 FNs were caused by the missed tracking of one vehicle. The 2 IDSs occurred when the tracked vehicle was out of the camera view for a while due to the UAV movements. Video #2 was more challenging to the vehicle detector, thus the tracking performances were also influenced by the detector and errors were larger for all three accuracy metrics. The FPs in video #2 was caused by two false detections tracked by our algorithm. But then they were discarded after not being associated with any detection in the next five frames. For FPS, SORT was the fastest among the three, but our algorithm achieved slightly better speed than particle filter and comparable to SORT.

The superiority of our algorithm comes from the special design considering UAV features and traffic information. Besides the FP and FN filtering scheme described in Section 3.2, the IDS is largely reduced because our algorithm tends to keep tracking a vehicle even if the detector loses it for a few frames. However, particle or SORT will assign a vehicle with a new ID if it is not detected for more than one frame. This design does not directly improve the macroscopic traffic parameter estimation much, but it is super beneficial to microscopic data collection since it extracts individual

Table 4 Comparison results with the state-of-the-art frameworks for traffic parameters estimation from moving UAVs

	[19]	[21]	[23]	This study
overall traffic speed	✓	✓	✓	✓
overall traffic density	×	✓	✓	✓
overall traffic volume	×	✓	✓	✓
lane-level traffic speed	×	×	×	✓
lane-level traffic density	×	×	×	✓
lane-level traffic volume	×	×	×	✓
individual vehicle speed	×	×	×	✓
individual vehicle trajectory	×	×	×	✓
space headway	×	×	×	✓
time headway	×	×	×	✓
processing speed	32 fps	24 fps	25 fps	4 fps
percentage of total frames with FP or FN	NA	47%	38%	9%

vehicle information. Moreover, the incorporation of KLT tracker in the algorithm ensures a good tracking accuracy: When just using historical movements to predict the future locations like Kalman filter, particle filter, or SORT, the prediction error can be large due to UAV's random movements, especially when predicting multiple frames ahead. However, our algorithm predicts individual vehicles' immediate motions in every frame using motion vectors extracted in the vehicle bounding boxes. For the processing speed, it is crucial for UAV monitoring since we expect real-time UAV monitoring in the future. While our tracking pipeline has multiple FP/FN filtering steps, its computational complexity is still relatively low, because we only use IoU as the association metric by taking advantage of UAV's top-view perspective, which avoids inter-vehicle occlusions.

In addition to tracking performance analysis, we then briefly compared the proposed framework to three state-of-the-art frameworks [19, 21, 23]. These studies accomplished several key milestones in the area of UAV-based traffic flow parameter estimation. According to the summary in Table 4, the proposed framework in this paper is able to extract many more types of traffic parameters.

The robustness has also been further improved. By running these frameworks on the same video datasets, the new one generates much lower errors than the state of the arts. We use a metric called 'the percentage of total frames with FP or FN,' which means the ratio of the number of frames with any FP or FN to the total number of frames. This metric decreases from 47% in [21] and 38% in [23] to 9%. It is an impressive result, especially considering that the new framework can produce a much broader range of traffic parameters. Note that this metric value is not available (NA) in [19] because this framework has no vehicle detection function. The cost is the computational complexity: in the three state-of-the-art studies, the processing speeds on a desktop computer are all over 24 fps, which enables real-time operation. However, the proposed framework runs in a non-real-time manner with a speed of just 4 fps. It is now appropriate for processing UAV videos offline, but a new machine with stronger computational power or optimisation to the algorithmic part is recommended for real-time applications in the future.

4.6 Applicability of the study

This study has great potential to be applied to a variety of traffic surveillance, data collection, and research tasks. While there are still some practical issues to be addressed for UAV-based traffic applications such as its short battery life and regulation issues, it is believed UAV will play an important role in modern transportation systems. We briefly discuss the applicability of this research in this subsection. We hope it can help the readers to have a better understanding of this work and inspire more ideas.

Advanced traffic sensing: Traditional traffic sensors are usually stationary sensors. To monitor a roadway or traffic network, a large number of sensors are needed, which has a high cost in installation and maintenance. Moreover, traditional traffic sensors collect limited types of traffic data. For example, loop detectors, which are the most common sensors for freeway systems, collect only macroscopic traffic speed, volume, and occupancy data. Probe vehicles, as another widely used sensing method, collects vehicle trajectories and a sample of traffic speed, but not able to collect lane information or volume. Our study offers an option to collect both microscopic and macroscopic traffic parameters using a sensor that is more cost-effective and flexible than traditional sensors.

Traffic enforcement: Over a decade ago, traffic agencies started to install cameras on manned aircraft for the purpose of issuing speeding tickets from the sky. Now some of the agencies have already considered using UAVs for traffic enforcement in order to reduce the cost of aerial ticketing operations. Our framework can be incorporated into unmanned traffic enforcement systems to monitor multiple individual vehicles and extract microscopic parameters from the sky to assist in enforcement decision making. Compared to manned aircraft solution, UAV solution to traffic enforcement has not only a lower cost but also better scalability.

Post-disaster traffic management: Disasters have a big impact on modern cities regarding human and economic losses. As post-disaster lifelines, post-disaster traffic networks are critical to support various operations like evacuation and recovery. However, traffic sensors may more or less stop working after disasters due to the damage to themselves or the traffic systems. At that time, UAVs can serve as alternative sensors to support traffic performance evaluation and post-disaster traffic planning. With UAV's high mobility and flexibility, its surveillance can cover the areas with the highest needs. Our framework and other related studies can be applied to emergent traffic state estimation and management.

Applicability to other traffic scenarios: There are a few more traffic scenarios we would like to discuss regarding the applicability of the framework: curves, urban situations, and occlusions. The current framework is designed for straight roads rather than curves based on two considerations: (i) straight roads are more common than curves so that the parameter extraction would work in most cases, and knowing the traffic situation does not always need the data from every single road segment; (ii) detecting curves would need a different and often more complicated method, which increases the computational cost. As far as urban situations, this framework still applies to urban roadway links but not intersections because intersections have different geometry with roadway links and the set of parameters for analysis is also different (e.g. queue length, number of stops, delay). For occlusion, one advantage of UAV is its top-view perspective, which naturally avoids occlusions in traditional traffic surveillance videos. One example is a truck blocking a car so that the detection and tracking of the car will likely fail. However, from a UAV perspective, this kind of occlusion rarely happens. Also, the proposed tracking method in our framework will keep tracking the vehicle for several frames even if the detector misses it. It means the vehicle's identity will not switch unless it is blocked by something in multiple consecutive frames.

5 Conclusion and future work

In this paper, an advanced framework was developed among the first efforts to test the availability of UAV videos to extract microscopic and lane-level macroscopic traffic parameters. This framework was specifically designed for UAV video processing, particularly for UAV videos with background motions. It contained three functional modules: The core functional module was composed of three interconnected processing streams that have a combination of new algorithms and modified existing algorithms; the data storing module contains a couple of data structures that keep the necessary information extracted from the whole UAV video; the traffic parameters estimation module include the calculations of seven fundamental traffic parameters based on

traffic flow theories and a preceding-vehicle determination process. The proposed framework was implemented and tested on real-world UAV video clips. The experimental results turned out to be very encouraging. It demonstrated that UAV video could serve as a valuable data source for automatic collection of both macroscopic and microscopic traffic parameters. It could be a major option for future ITS applications and transportation monitoring tasks.

Future work of this study will focus on the following three aspects. First, the system implemented in this paper does not support real-time on-board operation at this moment. Since real-time processing speed will be required for some modern ITS applications, we will explore solutions and make plans for implementing our framework on-board by optimising the efficiency and robustness from both the algorithmic aspect and coding aspect. Second, while the main focus of this study is on the development of the new framework for UAV-based traffic parameter data collection, we plan to spend some efforts in evaluating the integration of different algorithms into the framework and summarise their performances. Third, with this research as a baseline for microscopic and lane-level macroscopic traffic parameters estimation in UAV videos, we will develop and test additional methods targeting the long-term objective to handle as many real-world monitoring scenarios as possible.

6 Acknowledgments

The authors express our gratitude to Beihang University and Washington State Patrol Aviation for providing part of the video data. They also thank the editors and reviewers for volunteering their time to review this paper.

7 References

- [1] Barmounakis, E.N., Vlahogianni, E.I., Golias, J.C.: 'Unmanned aerial aircraft systems for transportation engineering: current practice and future challenges', *Int. J. Transp. Sci. Technol.*, 2016, **5**, (3), pp. 111–122
- [2] Kanistras, K., Martins, G., Rutherford, M.J., et al.: 'Survey of unmanned aerial vehicles (UAVs) for traffic monitoring', in Valavanis, Kimon P., Vachtsevanos, George J. (Eds.): 'Handbook of unmanned aerial vehicles' (Springer, USA2015), pp. 2643–2666
- [3] Du, Y., Zhao, C., Li, F., et al.: 'An open data platform for traffic parameters measurement via multirotor unmanned aerial vehicles video', *J. Adv. Transp.*, 2017, **2017**, pp. 1–12
- [4] Coifman, B., McCord, M., Mishalani, R.G., et al.: 'Surface transportation surveillance from unmanned aerial vehicles', Proc. of the 83rd Annual Meeting of the Transportation Research Board, Washington, DC, USA, 2004
- [5] Angel, A., Hickman, M., Mirchandani, P., et al.: 'Methods of analyzing traffic imagery collected from aerial platforms', *IEEE Trans. Intell. Transp. Syst.*, 2003, **4**, (2), pp. 99–107
- [6] Zhou, H., Kong, H., Wei, L., et al.: 'Efficient road detection and tracking for unmanned aerial vehicle', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (1), pp. 297–309
- [7] Freeman, B.S., Al Matawah, J.A., Al Najjar, M., et al.: 'Vehicle stacking estimation at signalized intersections with unmanned aerial systems', *Int. J. Transp. Sci. Technol.*, 2019, **8**, pp. 231–249
- [8] Ke, R., Lutin, J., Spears, J., et al.: 'A cost-effective framework for automated vehicle-pedestrian near-miss detection through onboard monocular vision', IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 2017
- [9] Ke, R., Pan, Z., Pu, Z., et al.: 'Roadway surveillance video camera calibration using standard shipping container', 2017 Int. Smart Cities Conf. (ISC2), Wuxi, People's Republic of China, 2017, pp. 1–6
- [10] McCord, M., Yang, Y., Jiang, Z., et al.: 'Estimating annual average daily traffic from satellite imagery and air photos: empirical results', *Transp. Res. Rec. J. Transp. Res. Board*, 2003, **1855**, pp. 136–142
- [11] Salvo, G., Caruso, L., Scordo, A.: 'Urban traffic analysis through an UAV', *Proc. Soc. Behav. Sci.*, 2014, **111**, pp. 1083–1091
- [12] Khan, M.A., Ectors, W., Bellemans, T., et al.: 'Unmanned aerial vehicle-based traffic analysis: methodological framework for automated multivehicle trajectory extraction', *Transp. Res. Rec. J. Transp. Res. Board*, 2017, **2626**, pp. 25–33
- [13] Kaufmann, S., Kerner, B.S., Rehborn, H., et al.: 'Aerial observations of moving synchronized flow patterns in over-saturated city traffic', *Transp. Res. C, Emerg. Technol.*, 2018, **86**, pp. 393–406
- [14] Cao, X., Wu, C., Lan, J., et al.: 'Vehicle detection and motion analysis in low-altitude airborne video under urban environment', *IEEE Trans. Circuits Syst. Video Technol.*, 2011, **21**, (10), pp. 1522–1533
- [15] Ammour, N., Alhichri, H., Bazi, Y., et al.: 'Deep learning approach for car detection in UAV imagery', *Remote Sens.*, 2017, **9**, (4), p. 312
- [16] Xu, Y., Yu, G., Wu, X., et al.: 'An enhanced Viola-Jones vehicle detection method from unmanned aerial vehicles imagery', *IEEE Trans. Intell. Transp. Syst.*, 2017, **18**, (7), pp. 1845–1856
- [17] Shastry, A.C., Schowengerdt, R.A.: 'Airborne video registration and traffic-flow parameter estimation', *IEEE Trans. Intell. Transp. Syst.*, 2005, **6**, (4), pp. 391–405
- [18] Cao, X., Gao, C., Lan, J., et al.: 'Ego motion guided particle filter for vehicle tracking in airborne videos', *Neurocomputing*, 2014, **124**, pp. 168–177
- [19] Ke, R., Kim, S., Li, Z., et al.: 'Motion-vector clustering for traffic speed detection from UAV video', 2015 IEEE First Int. Smart Cities Conf. (ISC2), Guadalajara, Mexico, 2015, pp. 1–5
- [20] Ke, R.: 'A novel framework for real-time traffic flow parameter estimation from aerial videos', 2016
- [21] Ke, R., Li, Z., Kim, S., et al.: 'Real-time bidirectional traffic flow parameter estimation from aerial videos', *IEEE Trans. Intell. Transp. Syst.*, 2017, **18**, (4), pp. 890–901
- [22] Chen, P., Zeng, W., Yu, G., et al.: 'Surrogate safety analysis of pedestrian-vehicle conflict at intersections using unmanned aerial vehicle videos', *J. Adv. Transp.*, 2017, **2017**, pp. 1–12
- [23] Ke, R., Li, Z., Tang, J., et al.: 'Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow', *IEEE Trans. Intell. Transp. Syst.*, 2018, **99**, pp. 1–11
- [24] Li, J., Chen, S., Zhang, F., et al.: 'An adaptive framework for multi-vehicle ground speed estimation in airborne videos', *Remote Sens.*, 2019, **11**, (10), p. 1241
- [25] Barmounakis, E.N., Vlahogianni, E.I., Golias, J.C., et al.: 'How accurate are small drones for measuring microscopic traffic parameters?', *Transp. Lett.*, 2019, **11**, pp. 332–340
- [26] Kim, E.-J., Park, H.-C., Ham, S.-W., et al.: 'Extracting vehicle trajectories using unmanned aerial vehicles in congested traffic conditions', *J. Adv. Transp.*, 2019, **2019**, 16 pages
- [27] Feng, S., Wang, X., Sun, H., et al.: 'A better understanding of long-range temporal dependence of traffic flow time series', *Phys. A Stat. Mech. Appl.*, 2018, **492**, pp. 639–650
- [28] Teutsch, M., Krüger, W.: 'Detection, segmentation, and tracking of moving objects in UAV videos', 2012 IEEE Ninth Int. Conf. on Advanced Video and Signal-Based Surveillance, Beijing, people's Republic of China, 2012, pp. 313–318
- [29] Rodriguez-Canosa, G.R., Thomas, S., Del Cerro, J., et al.: 'A real-time method to detect and track moving objects (DATMO) from unmanned aerial vehicles (UAVs) using a single camera', *Remote Sens.*, 2012, **4**, (4), pp. 1090–1111
- [30] Tsao, P., Ik, T.-U., Chen, G.-W., et al.: 'Stitching aerial images for vehicle positioning and tracking', 2018 IEEE Int. Conf. on Data Mining Workshops (ICDMW), Singapore, 2018, pp. 616–623
- [31] Breckon, T.P., Barnes, S.E., Eichner, M.L., et al.: 'Autonomous real-time vehicle detection from a medium-level UAV', Proc. 24th Int. Conf. on Unmanned Air Vehicle Systems, Bristol, UK, 2009, pp. 21–29
- [32] Gomaa, A., Abdelwahab, M.M., Abo-Zahhad, M.: 'Real-time algorithm for simultaneous vehicle detection and tracking in aerial view videos', 2018 IEEE 61st Int. Midwest Symp. on Circuits and Systems (MWSCAS), Windsor, Canada, 2018, pp. 222–225
- [33] Najiya, K.V., Archana, M.: 'UAV video processing for traffic surveillance with enhanced vehicle detection', 2018 Second Int. Conf. on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 662–668
- [34] Li, J., Ye, D.H., Chung, T., et al.: 'Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs)', 2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 2016, pp. 4992–4997
- [35] Carletti, V., Greco, A., Saggese, A., et al.: 'Multi-object tracking by flying cameras based on a forward-backward interaction', *IEEE Access*, 2018, **6**, pp. 43905–43919
- [36] Du, D., Qi, Y., Yu, H., et al.: 'The unmanned aerial vehicle benchmark: object detection and tracking', Proc. of the European Conf. on Computer Vision (ECCV), Munich, Germany, 2018, pp. 370–386
- [37] Khan, M., Ectors, W., Bellemans, T., et al.: 'Unmanned aerial vehicle-based traffic analysis: a case study for shockwave identification and flow parameters estimation at signalized intersections', *Remote Sens.*, 2018, **10**, (3), p. 458
- [38] Zhu, J., Sun, K., Jia, S., et al.: 'Urban traffic density estimation based on ultrahigh-resolution UAV video and deep neural network', *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 2018, **11**, (12), pp. 4968–4981
- [39] Bewley, A., Ge, Z., Ott, L., et al.: 'Simple online and realtime tracking', 2016 IEEE Int. Conf. on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 3464–3468
- [40] Lucas, B.D., Kanade, T., et al.: 'An iterative image registration technique with an application to stereo vision', 1981
- [41] Canny, J.: 'A computational approach to edge detection', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1986, **PAMI-8**, (6), pp. 679–698
- [42] Duda, R.O., Hart, P.E.: 'Use of the Hough transformation to detect lines and curves in pictures', 1971
- [43] Ester, M., Kriegel, H.-P., Sander, J., et al.: 'A density-based algorithm for discovering clusters in large spatial databases with noise', Knowledge Discovery and Data Mining (KDD), Portland, OR, USA, 1996, pp. 226–231