Real-Time Bidirectional Traffic Flow Parameter Estimation From Aerial Videos

Ruimin Ke, Student Member, IEEE, Zhibin Li, Sung Kim, John Ash, Zhiyong Cui, and Yinhai Wang

Abstract—Unmanned aerial vehicles (UAVs) are gaining popularity in traffic monitoring due to their low cost, high flexibility, and wide view range. Traffic flow parameters such as speed, density, and volume extracted from UAV-based traffic videos are critical for traffic state estimation and traffic control and have recently received much attention from researchers. However, different from stationary surveillance videos, the camera platforms move with UAVs, and the background motion in aerial videos makes it very challenging to process for data extraction. To address this problem, a novel framework for real-time traffic flow parameter estimation from aerial videos is proposed. The proposed system identifies the directions of traffic streams and extracts traffic flow parameters of each traffic stream separately. Our method incorporates four steps that make use of the Kanade–Lucas–Tomasi (KLT) tracker, k-means clustering, connected graphs, and traffic flow theory. The KLT tracker and k-means clustering are used for interest-point-based motion analysis; then, four constraints are proposed to further determine the connectivity of interest points belonging to one traffic stream cluster. Finally, the average speed of a traffic stream as well as density and volume can be estimated using outputs from previous steps and reference markings. Our method was tested on five videos taken in very different scenarios. The experimental results show that in our case studies, the proposed method achieves about 96% and 87% accuracy in estimating average traffic stream speed and vehicle count, respectively. The method also achieves a fast processing speed that enables real-time traffic information estimation.

Index Terms—Aerial video, motion-vector clustering, optical flow, traffic flow parameter estimation, traffic image analysis, unmanned aerial vehicle.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have been considered a novel traffic monitoring technology used to collect information about traffic conditions on roads. Compared to more traditional methods, roadway monitoring with unmanned aerial vehicles has several advantages. First, traditional monitoring devices such as loop detectors, surveillance video cameras and microwave sensors are usually placed at fixed locations to achieve a fixed surveillance coverage range; this may not be cost effective because a large number of these devices are needed for a single road segment [1], [2], [36]. Additionally, the maintenance of any of the fixed detectors leads to additional fees and would inevitably interrupt the normal traffic. In contrast, the UAV is a cost-effective platform that can both monitor a large continuous stretch of roadway and focus on a specific road segment. Also, maintenance on UAVs can be conducted off site and would this not lead to congestion on the roadway. The UAV also has another advantage in being able to provide rapid assessment and reconnaissance of an incident site for emergency response where no traditional sensors are installed and even in locations that humans may have difficulty accessing [3], [4]. Further, by achieving a top-view perspective, the aerial videos have the potential to be used to provide fast and accurate estimations of traffic information in multiple travel directions at the same time. For the aforementioned reasons, a UAV equipped with a camera is considered to be a low-cost and flexible platform that can provide efficient data acquisition [4]–[6]. Although the privacy issue and short battery life appear to be the two main concerns limiting practical use currently, it is widely believed that the UAV will achieve broader use in the near future once these concerns are addressed.

Due to the advantages of UAVs, aerial video-based traffic surveillance has become an active study topic in the transportation engineering field. Relevant research was initially conducted by State Departments of Transportation in Ohio, Florida, Georgia, and California [6]. Several papers were published by researchers participating in those projects that laid the foundation for UAV studies [4], [7], [8]. Although not placing tremendous weight on technical details, these papers present the “big picture” issues associated with UAVs and provide researchers with guiding information for future research.

Besides traditional challenges in surveillance video-based detection such as those associated with occlusion, shadows, and reflections, the most challenging issue in the aerial video image processing is that both the background and foreground in the image are moving due to the motion of the camera mounted on the UAV. Thus, several traditional image-based traffic information extraction technologies for fixed camera videos such as background subtraction and blob detection do not work well for UAV-based videos. Over the past two decades, some studies have focused on extracting traffic information from aerial videos; most of the work focused on applying traditional image processing techniques to detect and track vehicles in aerial videos [7]–[11]. Once vehicles are able to be properly detected and tracked, specific traffic information...
can then be extracted from video. Such methods require large computing workloads resulting in slow processing speeds.

To address these problems, our research team considered to make use of the motion information in aerial videos instead of converting moving background into static background as applying image registration in most previous studies. Given the intuition that no matter how a UAV moves, the motion of vehicles in the traffic stream is normally different from the motion of the background, extracting low-cost features that can represent the motions of the video background and moving traffic comes to mind.

In this paper, we proposed a new framework to achieve fast and accurate extraction of traffic flow parameters, i.e., speed, density, and volume, in two travel directions simultaneously. Specifically, our framework includes four components, which are interest point tracking, motion-vector clustering, connected graph-based vehicle detection and counting, and traffic flow parameter estimation. The key contributions of this paper are summarized as follows: (1) A framework for estimating multi-directional traffic flow parameters from aerial videos is proposed; (2) A novel method combining the KLT tracker, k-means clustering, and connected graphs for vehicle detection and counting is built; (3) Our system identifies traffic streams and extracts traffic information in a real-time manner; and (4) Our system works well in both daytime and nighttime settings, and is not sensitive to UAV movements (i.e., regular movement, vibration, drifting, changes in speed, and hovering).

II. LITERATURE REVIEW

Some systems have already been developed to meet the demand for transportation surveillance. Among such systems, some are equipped with inertial measurement units, infrared detectors, or high-precision position and orientation systems, which can provide additional information besides simply collecting aerial video [8], [13]–[16]. However, the costs of these systems are relatively high and thus limit their use in practical applications [17]. Other systems are only equipped with cameras and may be more cost-effective, but require more sophisticated computer vision techniques to achieve similar performance to the more advanced systems.

In term of research objectives, previous studies in the area can be roughly divided into three categories. The first category is road detection [6], [18]–[20]. UAV-based road detection is important because these approaches can be applied to vision-based navigation of UAVs [19]. Moreover, road detection can help automatically determine the region of interest (ROI) in a given traffic monitoring scenario. For example, Kim et al. [19] presented a unique real-time approach to detect various types of roads and other corridors. Their method learns a road structure from a single image and can then be applied for detecting and localizing the road in successive frames of a video. Zhou et al. [6] proposed an efficient road detection and tracking method for UAVs. This was the first work to introduce a tracking technique to speed up the localization of the road in a UAV video.

The second category of relevant research is vehicle detection and tracking [21]–[30]. These studies focus on the methodological part and have made contributions in improving the detection and tracking performance, such as increasing detection and tracking rate, reducing false-positive rate, and speeding up computing time. These studies also demonstrate tremendous possibility of supporting UAV-based transportation surveillance and traffic parameter estimation. For instance, Yu et al. [20] proposed a tensor voting computational framework to detect and segment motion patterns in a 4D space. The results show that some difficult problems that challenge the existing UAV systems can be addressed, but a long sequence is needed to detect motion patterns in their system, and it thus cannot meet the real-time requirement. Cao et al. [29] proposed a novel framework for UAV-based vehicle tracking using KLT features and a particle filter. Their method achieves very good tracking performance, but automatic detection of vehicles is not incorporated.

The third category focuses on traffic parameter estimation such as extracting speed, density, annual average daily traffic (AADT), travel time, and delay from aerial videos [4], [7]–[11], [13], [35]. When methods for detection and tracking developed in the aforementioned studies are combined with concepts and models in transportation engineering, useful traffic information can be extracted from the videos. Hence, automatic traffic monitoring can be partially achieved in practice. For example, Angel et al. outlined methods to estimate speeds, travel times, densities, and queueing delays from aerial imagery [8]. Their work is one of the milestones in estimating multiple key traffic parameters and works reasonably well. However, multiple data sources including a global position system (GPS) unit and an inertial measurement unit (IMU) are used in their system which increases the cost and data processing time. McCord et al. proposed a method to estimate AADT from satellite imagery and aerial photos using density information as algorithm inputs [7]. Their results showed the estimation accuracy is very high. However, their work focused on the modeling part rather than the automatic detection part. Shastry et al. [11] successfully incorporated KLT trackers in their framework to estimate traffic flow parameters, but their KLT trackers were applied to image registration which decreases computing speed. Their system is hence incapable of achieving real-time performance.

III. METHODOLOGY

A. Overview

The framework for the proposed traffic flow parameter estimation can be segmented into four consecutive steps, which are described in details in the following sub-sections. In the first step, interest points are identified in a pair of consecutive frames; we used interest points where the eigenvalue of the second-moment matrix is large, i.e., Shi-Tomasi features. The Kanade-Lucas optical flow algorithm is then used to track interest points between frame pairs [12]. In the second step, the speed and direction of interest points on both vehicles and the background are used as inputs to a clustering algorithm. We found that the background of a frame is the cluster with the most points; thus the background cluster can be identified [28]. In the third step, in order to get the count of vehicles in each direction of travel, a connected graph method is applied to further determine the membership of interest points in each
traffic flow cluster. The connectivity of two points is determined by the rule stating that interest points from one vehicle should have similar positions and velocities. Vehicle counts for each direction of traffic flow are thus the number of connected graphs per direction. Then, the speed of a vehicle can be calculated as the average speed of all interest points on that vehicle, but such a speed is in pixels per frame rather than a more sensible/intuitive speed unit. Finally, the actual traffic speed, density, and volume are estimated using reference markings on the roadway and the relationship among these three parameters.

Fig. 1 shows the workflow of the proposed method.

B. Interest Point Tracking

One of the advantages of the optical flow-based interest point tracking approach is that it makes use of the spatial intensity gradient of the image to guide the correspondence search. Thus, it can achieve a faster image processing speed and interest point matching accuracy [31]. Selecting “good” features (i.e., those that can minimize an error criterion) is critical for tracking features robustly across image frames. The Harris corner detector is the most well-known feature detector [32]. However, Shi and Tomasi proposed another corner detector and proved it could outperform the Harris corner detector.

To select “good” Shi-Tomasi features, let the matrix

\[
G = \sum_{p_x+w_x} \sum_{p_y+w_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}
\]

be the second-moment matrix of image \(I\) about point \(u = (p_x, p_y)\) in the window \(\omega\) of size \((2\omega_x + 1) \times (2\omega_y + 1)\). Then, the interest points in \(I\) are located at the points \(u_i\) where \(G\) is non-singular, and the minimum eigenvalue \(\lambda_{\text{min}} = \min(\lambda_1, \lambda_2)\) of \(G\) is above a specific threshold, \(\lambda_{\text{th}}\). To provide non-maximal suppression, any interest point \(u_i\) is not considered if there is another interest point \(u_i'\) in a \(3 \times 3\) neighborhood about \(u_i\) with a larger \(\lambda_{\text{min}}\). Finally, any interest points after the first \(n\), sorted in order of decreasing \(\lambda_{\text{min}}\), are not considered.

After interest points in frames \(I(x, y, t)\) and \(I(x, y, t + 1)\) have been identified, an interest point \(u_i\) can be tracked from time \(t\) to \(t + 1\) with the Kanade-Lucas algorithm for optical flow. In order to track points across distances on the order of several pixels, we used an iterative implementation with image pyramids [33]; the method is summarized in the following.

Let \(I^L\) be the pyramidal image of \(I\) at pyramid level \(L\). Then \(I^L\) is related to the original image \(I\) by the relation

\[
u^L = \frac{u^L}{2^L}
\]

where \(u\) is any point in \(I\). The objective is to find the optimal displacement \(s^*\) such that the error function \(\in (s_x, s_y)\) is minimized, that is

\[
s^* = \arg \min \{\in (s_x, s_y)\}
\]

where the error function

\[
\in (s_x, s_y) = \sum_{p_x+w_x} \sum_{p_y+w_y} \left( I^L(p_x, p_y) - J^L(p_x+s_x, p_y+s_y) \right)^2
\]

is the windowed sum-of-squared differences between images \(I^L\) and \(J^L = I^L(t + 1)\).

If \(s^{L+1} = g^{L+1} + v^K\) is an approximation for \(s^*\) at layer \(L + 1\), then the initial guess for the displacement for \(s^L\) is \(s^0 = g^L + v^0\), where

\[
\nu^0 = [0 \ 0]^T
\]

\[
g^L = 2s^{L+1} = 2(g^{L+1} + v^K).
\]

The update rule for \(\nu\) is based on the first-order Taylor approximation for the partial derivative of \(\in (s_x, s_y)\) (33). For iteration \(k\)

\[
\nu^K = \nu^{k-1} + \eta^K
\]

where

\[
\eta^K = G^{-1} \delta K
\]

\[
\delta K(p_x, p_y) = I^L(p_x, p_y) - J^L(p_x + s^{k-1}x, p_y + s^{k-1}y)
\]

\[
= I^L(p_x, p_y) - J^L(p_x + g^Lx + v^{k-1}x, p_y + g^Ly + v^{k-1}y).
\]

The iteration terminates either when \(k > K - 1\), or \(||\eta^K||\) is less than a specified threshold. The aforementioned iterative process described is executed for each layer \(L \in [0, L_m]\), starting from the image-layer \(L_m\), with the guess

\[
g^{L-1} = [0 \ 0]^T.
\]

Hence, for the layer \(L = 0\) corresponding to the original image \(I\), an interest point \(u\) can be found at the point \(u + s^0\) in \(J\).

C. Motion-Vector Clustering

To ensure that interest points from traffic streams and the video background can be correctly separated, both the background and vehicles in a given traffic stream should satisfy a “similar motion” criterion. That is, interest points from the background should have small variation in \(l\) and \(\theta\) between interest points. Likewise, for traffic clusters the variation in motion for vehicles within a given traffic stream should not be too big. Intuitively, if the motion criterion is violated, the clusters corresponding to different traffic streams, or between a traffic stream and the background, may become mixed. In practice, the motion criterion could be violated by conditions such as heavy congestion.

The result of the optical flow described in the previous section is a set of vectors \(V\) with elements \(v_i = (l_i, \theta_i)\), where
Fig. 1. Workflow chart of the proposed framework. Basically, there are four steps: Shi–Tomasi interest point extraction and tracking, motion-vector clustering, connected-graph-based multivehicle detection and counting, and bidirectional traffic flow parameter estimation. The instantaneous traffic flow parameters are calculated in each frame. In the end, the average speed, density, and volume are computed and recorded. The preclustering plot and postclustering plots in the flowchart correspond to Shi–Tomasi interest point tracking and motion-vector clustering processes, respectively. The precluster plot displays the motion distribution of interest points in the 2-D velocity space; the postclustering plot shows the motion-vector clustering result. In the clustering plots and corresponding frames, the points with different colors are clustered as different groups corresponding to the image background or traffic in either direction of travel.
and θ are given by

\[ l = \sqrt{s_x^2 + s_y^2} \]
\[ \theta = \arctan(s_y, s_x) \]
\[ s^0 = (s_x, s_y). \]  

(11)

Each element in V corresponds to an interest point \( u_i \) tracked from \( t \) to \( t + 1 \), where \( s^0 \) is the displacement for \( u_i \) calculated from the optical flow. Given V, its elements can be clustered with respect to \( l \) and \( θ \) in a 2D velocity space using a standard k-means algorithm. For \( k \) traffic streams, the number of clusters should be set to \( k + 1 \) to account for interest points in the background. Specifically, for bi-directional traffic, \( k \) should be set to three.

D. Connected Graph Based Vehicle Detection and Counting

So far, interest points have been grouped into clusters representing the background and different traffic streams. In each traffic stream cluster, interest points could come from different vehicles or the same vehicle. Determining the memberships of those interest points is crucial for vehicle counting and traffic stream speed estimation. Considering that vehicles are rigid objects, interest points on the same vehicle should share the same motion. Moreover, for an aerial video, interest points from one vehicle should be in close proximity in the 2D image space. Hence, to determine cluster memberships, a connected graph based method is proposed. Assume an interest point is a vertex \( P_i(x_i, y_i, l_i, θ_i) \) in a 4D space, where \( x_i \) and \( y_i \) denote the x and y coordinates in the current image frame, \( l_i \) and \( θ_i \) denote the displacement and direction of \( P_i \) from the previous frame to current frame, respectively. The group constraints are defined as follows:

\[ |x_i - x_j| < α \]
\[ |y_i - y_j| < β \]
\[ |l_i - l_j| < γ \]
\[ |θ_i - θ_j| < δ \]

where \( P_i(x_i, y_i, l_i, θ_i) \) and \( P_j(x_j, y_j, l_j, θ_j) \) are any two interest points in a specific traffic flow cluster. \( α, β, γ, \) and \( δ \) are the thresholds which determine the maximum difference of pixel numbers in the x and y directions and displacement. If all four constraints are satisfied, \( P_i \) and \( P_j \) are determined to be connected. According to this connectivity criterion, interest points can be grouped. In the grouping process, for each vertex that has not been assigned to a group, this vertex will be compared to existing vertex groups. The vertex will then be added to the group which satisfies the aforementioned criteria. If none of the groups satisfied the criteria, the vertex will be added to a new group and it will be the first element in this group. Ideally, one group represents one vehicle, thereby the motion of that vehicle can be estimated as the average motion of those interest points in the group. Likewise, the location of the vehicle in the current frame can be estimated as the centroid of the interest points.

E. Traffic Flow Parameter Estimation

Speed, volume, and density are the three most important parameters of traffic flow. Our method can estimate these parameters in free-flow and moderately congested traffic conditions. For speed estimation, suppose there are \( k \) clusters (\( k = 2 \) for bi-directional traffic) corresponding to traffic streams and a single background cluster such that the cluster center of the background is \( v_{bg}(l_{bg}, θ_{bg}) \). For traffic stream \( i, i ∈ [1, k] \), \( v_{i,j}(l_{i,j}, θ_{i,j}) \) and \( p_{i,j}(x_{i,j}, y_{i,j}) \) denotes the velocity and position of vehicle \( j \) in stream \( i \), respectively. Suppose the average velocity of traffic stream \( i \) relative to the background is \( v_{i,avg}(l_{i,avg}, θ_{i,avg}) \) where

\[ l_{i,avg} = \sqrt{d_{i,x}^2 + d_{i,y}^2} \]
\[ θ_{i,avg} = \arctan(d_{i,y}, d_{i,x}). \]

\[ d_{i,x} = \frac{\sum_j l_{i,j}}{\sum_j θ_{i,j}} \cos \left( \frac{\sum_j θ_{i,j}}{\sum_j 1} \right) - l_{bg} \cos(θ_{bg}) \]
\[ d_{i,y} = \frac{\sum_j l_{i,j}}{\sum_j θ_{i,j}} \sin \left( \frac{\sum_j θ_{i,j}}{\sum_j 1} \right) - l_{bg} \sin(θ_{bg}). \]  

(12)

To covert distance in pixels to speed in miles per hour, we used the video frame rate and reference markings from video frames. For a video with frame rate \( f \), where the pixel length and actual length of a reference marking are \( l_p \) and \( l_a \), respectively, the actual speed \( s \), of a vehicle that moves \( d_p \) in one frame pair is determined by

\[ s = \left( \frac{l_a}{l_p} \right) \times \frac{d_p}{f}. \]  

(13)

To estimate density, suppose the vehicle count for traffic stream \( i \) is \( n_i \) and the length of the road segment in the ROI is \( l_{seg} \) for traffic in both directions, and the number of traffic lanes per direction is \( m_i \). Further suppose the UAV flies at a constant height. Then \( l_{seg} \) can also be determined by the ratio of pixel and actual lengths using reference markings. For bi-directional traffic, a UAV flies along a stretch of a roadway thereby \( l_{seg} \) stays relatively constant. Hence, the density of traffic stream \( i \) is given by

\[ k_i = \frac{n_i}{l_{seg} \times m_i}. \]  

(14)

Then, to determine volume, since our method aims to solve parameter estimation of free-flow and moderately congested traffic conditions, the volume \( v_i \) for traffic stream \( i \), is given by

\[ v_i = s_i \times k_i \times m_i \]  

(15)

where \( s_i \) and \( k_i \) denote estimated speed and density of traffic stream \( i \).

IV. EXPERIMENTAL RESULTS

A. Platform and Parameter Settings

The method was implemented with C++ and OpenCV 2.4.11 [34]. The first test dataset, which will be discussed in detail later, consisted of a 280-frame video at a 960 × 540 resolution, taken at 24 frames-per-second by a UAV traveling above a
freeway segment. The ROI was selected to include six lanes of traffic moving in two directions (i.e., three lanes per direction), denoted Direction A (with traffic moving towards the left) and Direction B (with traffic moving towards the right), resulting in a window of $500 \times 220$ pixels (see Fig. 2). As mentioned previously, reference markings were used to compute the ratio of pixel to actual length. Different reference markings can be used, such as the length of a school bus [11]. Here, lane markings were used with a measured pixel length and actual length of 36 pixels and 6 meters, respectively.

The setting of five parameters is important, the first of which is the number of tracked interest points. The interest points are ranked by their matching errors. If $N$ points are tracked, the first $N$ points with the smallest errors would be tracked. Thus, if $N$ is set too small, some vehicles would not be detected because there would be no interest points on them. On the other hand, if $N$ is set too large, the motions of some interest points would be incorrectly estimated leading to some false-positives (non-vehicles detected as vehicles) in vehicle detection. We also observed that $N$ influences the processing speed substantially. The other four parameters are the four thresholds in the connected graph based vehicle detection and counting step, i.e., $\alpha, \beta, \gamma, \delta$. If either $\alpha$ or $\beta$ is too small, interest points from the same vehicle may be classified into more than one group. Thus, one vehicle can be erroneously detected as two or more vehicles. In contrast, if $\alpha$ or $\beta$ is too large, two vehicles that are close to each other can be incorrectly classified as one vehicle. Likewise, if $\gamma$ and $\delta$ are too small, due to the estimation errors in tracking interest points, interest points coming from the same vehicle can sometimes have small differences in speed or direction values. These differences can also result in erroneously counting one vehicle as two or more. If $\gamma$ or $\delta$ is too large, in some cases the 4D space for the connected graph based grouping is reduced to 2D or 3D, resulting in less effective classification.

Since the influence of $\alpha$ and $\beta$ is the most intuitive, an example showing the influence of these parameters is presented in Fig. 3. Four combinations of $\alpha$ and $\beta$ were tested, showing the influence of thresholds on vehicle counting. Comparing the upper-left snapshot with the lower-left one, in the 80th frame, larger thresholds resulted in detecting the two white vehicles in Direction B as one vehicle. However, in the 280th frame, where there is a bus, the larger thresholds correctly classified the interest points on the bus as one connected group. The smaller threshold combination, however, failed to count it as one vehicle, and instead three buses were detected. Obviously, the errors in classification affect traffic parameter estimation. Hence, setting proper parameter values is important to both
The results are considered reasonable because in the free-line the average speed of the traffic streams was relatively stable directions. The results are shown in Fig. 4. In both directions, the traffic parameters were calculated for the two travel di-


calculated by speed and density, the variation of volume over some frames but fewer vehicles in other frames). As volume is different frames (i.e., there could be multiple vehicles within the same frames but fewer vehicles in other frames). As volume is calculated by speed and density, the variation of volume over different frames was expected.

In practice, reporting frame-by-frame traffic information is unnecessary and tedious. Instead, aggregated traffic flow parameters similar to those reported from inductive loop detectors are more useful to people, such as traffic engineers, than instantaneous results. In our study, we further calculated aggregated traffic flow parameters from the frame-based information. The aggregated speed, density, and volume for Direction A were 54.6 mph, 25.4 pc/mi/lane, and 4177.6 pc/h; for Direction B, they were 42.5 mph, 41.4 pc/mi/lane, and 5222.9 pc/h. We see that the traffic stream in Direction A had a higher speed and lower density, a result which is quite intuitive (see Fig. 4).

C. Estimation Accuracy Analysis

To validate the traffic parameter estimation accuracy of our proposed methods, ground truth data on the average speed and discouraging at first glance. As such, our research team per pixel by frame and vehicle count for traffic flow in each travel direction were measured from the aerial videos. An on-screen pixel measurement tool was used to manually collect ground truth speed data. The speed of individual vehicles was measured over intervals of five consecutive frame pairs. There were two factors we considered for choosing our measurement interval: 1) Generally, the smaller the interval is, the larger the measurement error would be; and 2) if the interval is too large, resolution of the speed data would deteriorate. Normally, in five frames an individual vehicle moved over 20 pixels in the test video, and the actual time interval of five frames was less than 0.2 seconds; in such a period, the speed of a vehicle can be viewed as constant. For the aforementioned reasons, the decision to use five frame pairs as the measurement interval was made.

The measured ground truth speed, estimated speed by our method, and the error rate for each travel direction are presented in Fig. 5(a). Let \( \varepsilon \) denote error rate, and \( y_{\text{estimated}} \) and \( y_{\text{truth}} \) denote estimated value and ground truth, respectively. Then, \( \varepsilon = \frac{|y_{\text{estimated}} - y_{\text{truth}}| \times 100}{y_{\text{truth}}} \). Clearly, the trend of ground truth speeds and estimated speeds are both rather similar for Direction A and B. That being said, our method achieved promising and accurate results in speed estimation. The average error rates for Direction A and B were 2.366% and 2.634%, respectively (see Table I); the maximum of error rates was less than 10%. We also found that the variances of the estimated speeds were both slightly larger than the ground truth data. This is because there were some false-positives in vehicle detection for each travel direction. Such detections were clustered as vehicles and their motions were added to the real motion. Since their motions were random and the traffic flow motion was relatively constant, they led to increased variance in parameter estimates. Such an issue could happen when an interest point from the background is mistakenly matched with another nearby interest point which shares high similarity in terms of intensity and gradient changes in the KLT tracking process.

Compared to speed, error rates associated with vehicle count had higher averages and larger standard deviations. The average error rates were 17.393% and 17.055% for Directions A and B, respectively (see Table I). From Fig. 5(b) it can be seen that in most frames, the estimated vehicle counts were larger than the ground truth. This was due to the parameter settings of the number of tracked interest points. On one hand, the tracked interest points were ranked by their matching errors. Thus, increasing the associated parameter a small amount resulted in tracking all vehicles, but also increased the probability of the occurrence of false-positives. During the experiments, we observed that if this parameter was set too small, the vehicle count would decrease and some vehicles were not detected. Ultimately, we set the thresholds of the parameter values associated with the connected graph process each to be relatively small. As we discussed previously, use of smaller thresholds generates more vehicles in the estimation process.

It is important to note that the maximum error rates for vehicle count estimation reached 100% for Direction A and 75% for Direction B. These error rates may be quite surprising and discouraging at first glance. As such, our research team
examined the data for the frames with high error rates. We found that large errors occurred most often when the image frames included very few vehicles. For example, when a 100% error rate appeared, there was only one vehicle within the detection zone, but our algorithm recognized it as two objects resulting in the extremely high calculated error rate. When there are some more vehicles within the detection zone, the error rate for vehicle count tends to be much lower. That said, even if we include these abnormal frames in the calculations of the average count estimates, we still got relatively accurate detection results, demonstrating the robustness of our method.

To further validate the performance of our method, four more videos taken in different scenarios were examined. The total monitoring time length was about forty minutes and very good detection results were observed. In order to carefully evaluate the performance, 1690 frames were manually examined frame-by-frame to obtain the ground truth values of speed and traffic count/volume. The traffic condition, time of day and the movement of the UAV were also different in each video; however, the performance of our system was fairly stable, staying at a high-accuracy level. The summary of the estimation results and performance are presented in Table II. In the second test video, for the two traffic streams, speed estimation accuracy reached 97.300% and 94.220%; count estimation accuracy reached 86.612% and 82.731%. In the third test video, speed estimation accuracy was 94.995% and 93.368%; count estimation accuracy was 81.449% and 93.335%.

Other challenging scenarios, different from those previously tested, were tested in videos #4 and #5. One was an aerial video taken at night, in which the UAV experienced drift during its regular movement along the freeway. Normally, vehicle detection is much more difficult at night than during the day. However, we noticed that the count estimation in our test video #4, i.e., the video taken at night, obtained the highest accuracy among all test videos, almost reaching 90% (see Table II). Fig. 6 shows the vehicle detection and counting results in randomly selected frames. It can be seen there are very few false negatives and false positives. We notice that as long as vehicles are driving with their lights on at night, they will be detected with high probability and thus few false negatives will be generated. This observation is due to the fact that interest points most often come from locations in a frame where there is light when the frame was filmed at night time. Also, the number of false positives generated from this observation of this video was quite fewer; this can be explained using Fig. 7. False positives occur when non-vehicles are detected as vehicles. In other words, they usually result when points in the image background are mistakenly recognized as
TABLE II
ESTIMATED TRAFFIC FLOW PARAMETERS AND PERFORMANCE EVALUATION SUMMARY OF THE PROPOSED SYSTEM

<table>
<thead>
<tr>
<th></th>
<th>Test Video #1</th>
<th>Test Video #2</th>
<th>Test Video #3</th>
<th>Test Video #4</th>
<th>Test Video #5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Frames</td>
<td>280</td>
<td>180</td>
<td>380</td>
<td>550</td>
<td>300</td>
</tr>
<tr>
<td>Dimension of ROI</td>
<td>500 × 220</td>
<td>500 × 175</td>
<td>500 × 170</td>
<td>800 × 300</td>
<td>500 × 160</td>
</tr>
<tr>
<td>UAV Movement</td>
<td>Move left with constant speed and vibration</td>
<td>Move right with constant speed and vibration</td>
<td>Move with changing speed and vibration</td>
<td>Move with drifting</td>
<td>Hover over the freeway segment</td>
</tr>
<tr>
<td>Video Background</td>
<td>Moving</td>
<td>Moving</td>
<td>Moving</td>
<td>Moving</td>
<td>Fixed</td>
</tr>
<tr>
<td>Time</td>
<td>Daytime</td>
<td>Daytime</td>
<td>Daytime</td>
<td>Nighttime</td>
<td>Daytime</td>
</tr>
<tr>
<td>Estimated Speed (mph)</td>
<td>Dir. A 54.6</td>
<td>Dir. B 42.5</td>
<td>Dir. A 25.4</td>
<td>Dir. B 41.1</td>
<td>Dir. A 4177.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated Density (pc/mi/lane)</td>
<td>Dir. A 42.4</td>
<td>Dir. B 31.1</td>
<td>Dir. B 46.2</td>
<td>Dir. B 29.7</td>
<td>Dir. B 4329.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated Volume (pc/h)</td>
<td>Dir. A 48</td>
<td>Dir. B 45.8</td>
<td>Dir. A 25.8</td>
<td>Dir. B 41.6</td>
<td>Dir. B 3715.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speed Estimation Accuracy (%)</td>
<td>Dir. A 97.634</td>
<td>Dir. B 97.366</td>
<td>Dir. A 94.995</td>
<td>Dir. B 95.475</td>
<td>Dir. A 6833.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count Estimation Accuracy (%)</td>
<td>Dir. A 82.607</td>
<td>Dir. B 86.612</td>
<td>Dir. A 81.449</td>
<td>Dir. B 90.234</td>
<td>Dir. B 6833.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processing Speed (fps)</td>
<td>Bi-direction</td>
<td>27.294</td>
<td>26.360</td>
<td>27.164</td>
<td>32.168</td>
</tr>
</tbody>
</table>

Fig. 6. Selected frames showing the vehicle detection results on test aerial video #4, which was taken by a UAV flying along a freeway at night.

Fig. 7. Snapshots showing the motion vectors extracted in the region of interest. The upper image was taken during the day, whereas the lower image was taken at night.

vehicles. In our framework, false positives are primarily caused by inaccurate motion estimation of interest points from the background. In Fig. 7, it can be seen that during the day there were many more interest points in the frame background than at night, hence increasing the possibility of inaccurate motion estimation. Therefore, it is reasonable that our system performed even better at night than for some cases in which the video was taken during the day.

In order to compare the performance of our system on moving-background videos and fixed-background videos, another test video was taken when the UAV was hovering over a freeway segment, with no vibration or drifting, i.e., the
video background was not moving at all. Table II presents the estimation and evaluation results of video #5. The results prove that the proposed system works well on both moving-background aerial video and fixed-background aerial video, thus further demonstrating the system’s ability to adapt to different monitoring conditions.

By considering the results of the analyses collectively, we can conclude that the performance of our method is not sensitive to the dimensions of the ROI, UAV movements, or light conditions. The slight differences in detection accuracy that were observed could be caused by the differences in heavy vehicle ratios between videos or system parameter settings.

D. Discussion on Real-Time Performance

As real-time traffic information is so important for traffic control or route guidance, the processing speed of our method is evaluated in this section. The experiments were conducted on a computer with an Intel i5-2310 CPU @ 2.9 GHz processor and 6G of memory. Under current parameter settings, our system operates in real-time; the average processing speeds for the five videos were 27.294 fps, 26.360 fps, 27.164 fps, 32.168 fps, and 28.529 fps, respectively (see Table II). Considering our videos have a frame rate of 24 frames-per-second, real-time traffic parameter extraction is achieved.

The high processing speed of our system results from key differences between our processing logic and framework when compared to previous ones discussed in the literature review. Most existing work has focused on trying to turn the problem into an image processing problem with a fixed background. Then, image registration is used to match features in two frames, but the overall process is quite time consuming and complicated. Each of the three algorithms that played an integral role in our methodology, i.e., optical flow, k-means clustering, and connected graph, have low computational complexities. We also observed that the processing speed of our method was mainly influenced by the number of interest points (i.e., the parameter \( N \)) and the accuracy level of the interest point extraction. The parameter \( N \) determines the elements in the array storing interest points and affects the processing speeds of each of the optical flow, k-means clustering and connected graph algorithms. Ultimately, one should attempt to make a balance between the processing speed and estimating accuracy. Our test results show that when \( N \) is set to 50, the average processing speed can reach up to 40 fps. When \( N \) is set to 200, the average processing speed decreased to approximately 15 fps on our machine. Clearly processing speed can be improved by using higher-performance computers, thus better enabling real-time traffic parameter extraction from aerial videos.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a modeling framework to identify traffic streams and extract the bi-directional traffic flow parameters (i.e., speed, density, and volume) from aerial videos in a real-time manner. Our framework includes four consecutive steps. The first step is Shi-Tomasi interest point detection and Kanade-Lucas optical flow based tracking across frames. The second step clusters optical flow vectors via the k-means algorithm based on their speed and direction. The third step groups interest points in each travel direction and calculates the motion of each individual vehicle. The final step calculates speed, densities, and volumes for the two travel directions. Experiments on five datasets from aerial videos show that the proposed method yields about 96% and 87% accuracy in estimating average speed and vehicle count of the two traffic streams considered, respectively, and operates in real-time. Based on the experiments and analysis, our system achieves high accuracy and processing speed in both daytime and nighttime settings. Further, it is not sensitive to UAV movements (i.e., regular moving, vibration, drifting, changes in speed, and hovering).

Future research work can focus on addressing the following issues. First, our current method works well for free-flow and moderately congested traffic flow conditions because the motion criteria depend on similar movement of both background and traffic interest points. Similarly, our method works well for traffic on straight road segments. Testing our method on heavily congested traffic conditions and curved road segments, and adjusting it to improve performance would also be insightful.

Second, our algorithm sometimes recognizes trucks, buses, and other large/heavy vehicles as multiple passenger cars. Future work may investigate to improve accuracy for estimation of large vehicles and to improve the overall performances of our method.

ACKNOWLEDGMENT

The authors would like to thank Beihang University for providing the aerial video data sets.

REFERENCES


Mr. Ke is a Technical Program Committee Member at the 2015 IEEE International Smart Cities Conference.

Zhibin Li received the Ph.D. degree from the School of Transportation, Southeast University, Nanjing, China, 2014. From 2010 to 2012, he was a visiting student with University of California at Berkeley, Berkeley, CA, USA. He is currently a Postdoctoral Researcher with the Smart Transportation Applications and Research Laboratory (STAR Lab), University of Washington, Seattle, WA, USA. He is the author or coauthor of more than 80 technical papers and conference proceedings. His research interests include intelligent transportation systems, big data, traffic safety, and traffic operation.

Dr. Li received the China National Scholarship in 2012 and 2013 and won the Best Doctoral Dissertation Award by the China Intelligent Transportation Systems Association in 2015.

Sung Kim received the B.S. degree in electrical engineering from University of Washington, Seattle, WA, USA, in 2015, where he is currently working toward the Ph.D. degree, with his time divided between the VLSI Systems Research Laboratory (Department of Electrical Engineering) and Sampa (Department of Computer Science and Engineering). His research interests include hardware accelerators for computer vision applications, low-power VLSI design, and techniques for approximate computing.

John Ash received the B.S. degree in civil engineering and mathematics and the M.S. degree in civil and environmental engineering from University of Wisconsin—Madison, Madison, WI, USA, in 2012 and 2014, respectively. He is currently working toward the Ph.D. degree in civil and environmental engineering at University of Washington, Seattle, WA, USA. He has been a Research Assistant for the duration of his graduate education at both University of Wisconsin—Madison and University of Washington. His research interests include traffic safety, traffic operations, and intelligent transportation systems.

Mr. Ash was selected to be a Valle Scholar at University of Washington in 2014.
Zhiyong Cui received the B.S. degree in software engineering from Beihang University, Beijing, China, in 2012 and the M.S. degree in software engineering from Peking University, Beijing, in 2015. He is currently working toward the Ph.D. degree in civil and environmental engineering at University of Washington, Seattle, WA, USA. Since 2015, he has been a Research Assistant with the Smart Transportation Applications and Research Laboratory (STAR Lab), University of Washington. His research interests include traffic big data mining and computer-vision-based traffic detection.

Yinhai Wang received the Master’s degree in computer science from University of Washington (UW), Seattle, WA, USA, and the Ph.D. degree in transportation engineering from University of Tokyo, Tokyo, Japan, in 1998. He is a Professor in transportation engineering and the Founding Director of the Smart Transportation Applications and Research Laboratory (STAR Lab), UW. He also serves as the Director for the Pacific Northwest Transportation Consortium (PacTrans), U.S. Department of Transportation University Transportation Center for Federal Region 10. He has published over 100 peer-reviewed journal articles and delivered more than 110 invited talks and nearly 200 other academic presentations. His active research fields include traffic sensing, e-science of transportation, and transportation safety.

Dr. Wang is a member of the Transportation Information Systems and Technology Committee and the Highway Capacity and Quality of Service Committee of the Transportation Research Board. He is currently a member of the Steering Committee for the IEEE Smart Cities and chaired the First IEEE International Smart Cities Conference in 2015. He was an elected member of the Board of Governors for the IEEE Intelligent Transportation Systems Society from 2010 to 2013 and served on the Board of Governors for the ASCE Transportation and Development Institute during 2013–2015. He is an Associate Editor of Journal of Intelligent Transportation Systems, Journal of Computing in Civil Engineering, and Journal of Transportation Engineering. He was the winner of the ASCE Journal of Transportation Engineering Best Paper Award for 2003.