Approach to Obtaining Traffic Volume and Speed Based on Video-Extracted Trajectories

Z. Linjie[1] and W. Hao[2]

[1]Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Southeast Univ., Nanjing, Jiangsu, China. Email: 220193107@seu.edu.cn
[2]Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Southeast Univ., Nanjing, Jiangsu, China. Email: haowang@seu.edu.cn

ABSTRACT

Intersections are key nodes of urban roads and constraints of traffic network capacity. And traffic data is the basis for intersection design, analysis, and management. The current measurement of vehicle speed is to use radar guns which could only measure point speed and volume measurement is manual counting which are time-and-energy-consuming. This paper introduces a method that collects the vehicle trajectories from videos taken by an unmanned aerial vehicle (UAV), with further processing of the trajectories to obtain useful traffic data such as intersection flow and vehicle speed. The program uses background subtraction and morphological operation for vehicle detection and uses KCF trackers to track vehicle targets. After extracting the trajectories of each vehicle in the field of view, it performs turning identification, counts the traffic flow of each entrance, and extracts more data such as vehicle speed and time headway. In order to assess the feasibility of the video processing method, case studies were performed based on the field data collected by a UAV at urban intersections in Nanjing, China. The results of these experiments proved that the approach can be applied to the efficient analysis of intersection volume, speed and other parameters based on UAV videos.

INTRODUCTION

Intersections are key nodes of city roads, which are also the bottleneck of limiting urban traffic network capacity. And it is of great importance to pay attention to intersection design, analysis, monitor and management. Obtaining traffic volume and vehicle speed of intersections is vital and fundamental to achieve the goals above.

Traditional way of traffic volume, by counting manually, is time-consuming and energy-consuming. In the above method, an intersection requires multiple people to cooperate and investigate at the same time. When the traffic is at high level, counting volume of different turnings of one intersection entrance may need more than one person, which requires a lot of effort.

As for vehicle speed, the traditional methods of measuring it are using microwave radar, laser, ultrasonic and ground coil. Intersections usually have multiple lanes, of which the vehicle density is high, and the vehicles' driving directions are various and uncertain. Also, pedestrians and non-motor vehicles would interfere with speed measurement. And most importantly, they all have the disadvantage that they only measure the vehicle speed at a certain point and cannot continuously measure the constantly changing speed over a period of time.

This article introduces a new method that extracts data like intersection volume and vehicle speed from aerial videos. Using unmanned aerial vehicle (UAV) to shoot over the intersection, clear and stable intersection video data of the study range and period is taken. Vehicle target recognition is performed for each frame of the video, supplementing new vehicles once they enter the video shooting range, and tracking each target until it is out of the range. Record the trajectory of each vehicle, then use algorithm to deduce the turning of each vehicle and calculate the speed and other data at each track point.

The benefits of the method can be viewed from different perspectives. First, the method saves money and human efforts to gain intersection data. Digital cameras can provide higher quality videos, and video-based method is becoming a lower-cost and highly efficient alternative for non-intrusive speed measurement. Second, image processing has been studied for a long time and predecessors have laid a solid foundation. And mature algorithms have been written that can be utilized. Third, it's convenient to obtain multiple detailed as well as accurate vehicle data using vehicle trajectories by this method.

The data obtained using this method can be very helpful. Vehicle speed at each trajectory point could show its changing trend, range, and fluctuation status when vehicles passing the intersection, thus can be used for speed analysis. Also, vehicle speed and volume provide information needed for traffic control and intersection design. And combining speed and position in the intersection, traffic safety analysis could be performed, and potential conflict points could be inferred.

RELATED WORK

Vehicle Volume Measurement

There are some methods of extracting traffic volume of intersections. Ghanim and Shaaban used an artificial neural networks (ANN) model trained to analyze the relationship between the approach volumes and the corresponding turning movements (Ghanim and Shaaban 2019), but it's an estimation method. Dey and Kundu presented a video-based traffic volume and direction estimation using transfer learning (Dey and Kundu 2019), but the accuracy is lower than our proposed one. Andrew and Schrock used Bluetooth data loggers to obtain origin-to-destination leg data, then count turning movements (Andrew and Schrock 2013), but it's not likely to equip all vehicles in the roads with Bluetooth equipment. Chen, A. et al. used path flow estimator to derive complete link flows and turning movements for the whole road network (Chen et al. 2012), but it's also an estimation method.

Vehicle Speed Measurement

Several video-based approaches for measuring the speed of vehicles in roadways were proposed.

Martucci et al. use calibrated GNSS-based vehicle speed meters for speed measurement (Martucci et al. 2020). Wellbrock et al. suggested a method using fiber carrying high speed data transmission to detect vehicle speed, but it's rather expensive (Wellbrock et al. 2019). Julina et al. suggested a vehicle speed detection system to estimate vehicle speed, but the target is segments of roads instead of intersections (Julina et al. 2019). Chen et al. suggested an adaptive framework for multi-vehicle ground speed estimation in airborne videos (Chen et al. 2019). Luvizon et al. suggested a method using videos captured near a roadway to estimate vehicle speeds by detecting and tracking license plates (Luvizon et al. 2016).

Overall, all the speed measuring approaches focus on a segment of roadway or a point of roadway, not intersections, because they use the videos taken by a fixed cameras overhead near the roadway. Because the video was shot obliquely, rectification for perspective distortion is needed in most methods, and may cause inaccuracy to some extent. Also, the accuracy of methods may not be as high as used in segments for the following reasons: i) intersections have more uncertainties and disturbance; ii) their ways to set moving direction of vehicles may not be available for intersections; iii) methods using license plates are not useful for intersections.

Moreover, the proposed method could collect various traffic data using single program and single source-videos.

METHODOLOGY

In this section, the major algorithms of the method are described.

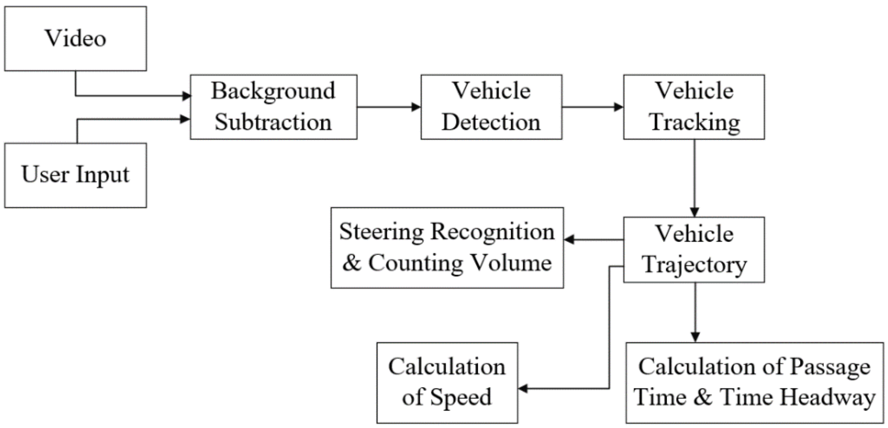

Figure 1. System architecture

A. User input

Users will need to help the system to locate each entrance/exit road for further locating vehicles. By simply draw four lines in the interactive interface, the intersection is divided into five parts (Figure.2).

Also, the pixel coordinates in the image are converted into practical coordinates according to a certain ratio. The system obtains the conversion scale by letting users draw a line segment on the image and inputting the corresponding actual length.

Figure 2. Division of the intersection

Background subtraction

Before extracting foreground, the video is processed to minimize the movement of the video. The program lets users to select three corner points, and then uses the goodFeaturesToTrack function to find the optimal corner point near the input point. Then Lucas-Kanade algorithm is used for debounce processing.

The program uses a mixed Gaussian method to create a background model and uses the background subtraction method to obtain the foreground image. After grayscale processing and removing the vehicle shadow using an HSV color space-based shadow detection, the contrast effect is as follows.

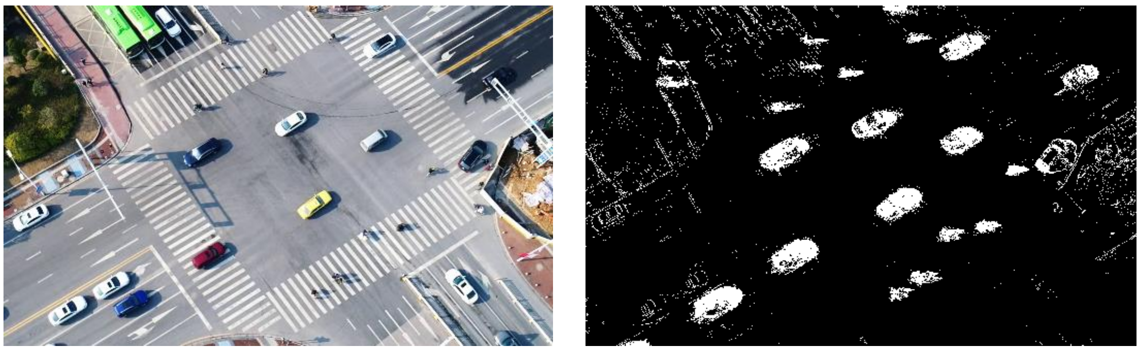

Figure 3. Comparison before / after the background subtraction

For different intersections and video with different resolution, they usually require different combinations of optimal morphological operations. The difficulty lies in ensuring that the vehicle remains intact while do not stick to each other. In the meantime, minimize the interference of environment.

In this example, considering the image size and the size of the vehicle, a convolution kernel having a size of 2 pixels * 2 pixels is used for the morphological operation, and after the combined operation of opening and closing operation, the effect is as follows.

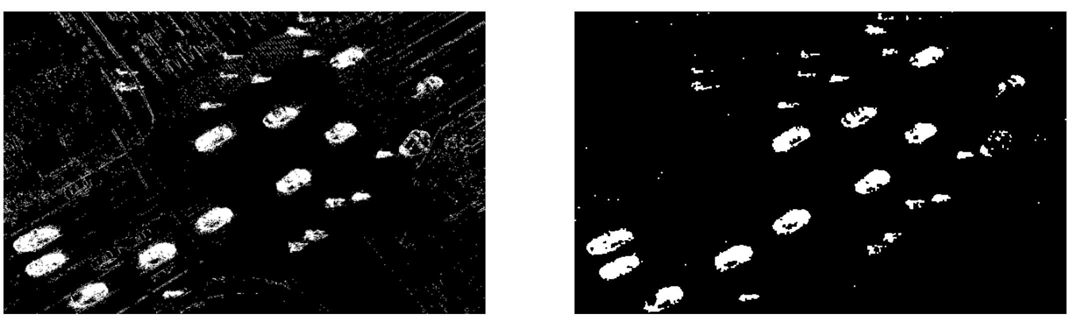

Figure 4. Comparison before / after the morphological operation

B. Vehicle Detection

Detect contours in morphologically processed images and further process the contours: remove contours that are outside the boundary and those that are too small or too large. This step is for eliminating small noise points or objects that are not motor vehicle, such as pedestrians and non-motor vehicles. Then create a bounding rectangle for each contour, which is the initial target frames of the subsequent tracking vehicle phase. And the rectangular center point of the search box is considered to be the position point of the vehicle, that is, the trajectory point.

C. Vehicle Tracking

Track the search box and get each vehicle's trajectory. The process includes the following steps: search and filter new objects, add new objects, and track object targets.

a) Search and filter new objects

New vehicles enter the video range and need to be added as new tracking objects. As described in the previous section, contour detection is performed within the image for each subsequent frame. Then, a logical judgment is added: detecting whether the newly generated search box overlaps with the search frame of the previous frame, and the overlapping area ratio is greater than a certain threshold. If yes, it means that the same car is recognized twice, and one of them needs to be deleted. If the above conditions are not met, it is considered that a new car appears, and a new target search box is added.

b) Add new objects

Add the new search box generated in a) and create a tracker for the new object and initialize the tracing.

Due to factors such as environmental interference, the vehicle is prone to tracking loss during the tracking process. There are two types of lost vehicles: vehicles are outside the monitoring range; vehicles are within the detection range, but the tractor loses its objects due to various factors (which are described in the ERROR INVESTIGATION chapter). The latter needs to be retrieved, otherwise it will be considered as a new vehicle and cause the trajectory to be interrupted.

To avoid interrupted vehicle trajectory of the latter case, add anti-lost code in the tracking process: if the vehicle is found to be lost, use vectors to record its centroid position, vehicle number and lost time.

At this stage, if it is detected that the added new object is near the position of one lost vehicle, which is within a certain pixel threshold area (such as in a nearby 20 pixels * 20 pixels size rectangle), as shown in Figure 5. And the loss time is within a certain threshold (like 10 frames), this new object is considered to be the previous lost vehicle. These two thresholds should also be tested to determine different optimal values for different procedures to ensure maximum accuracy.

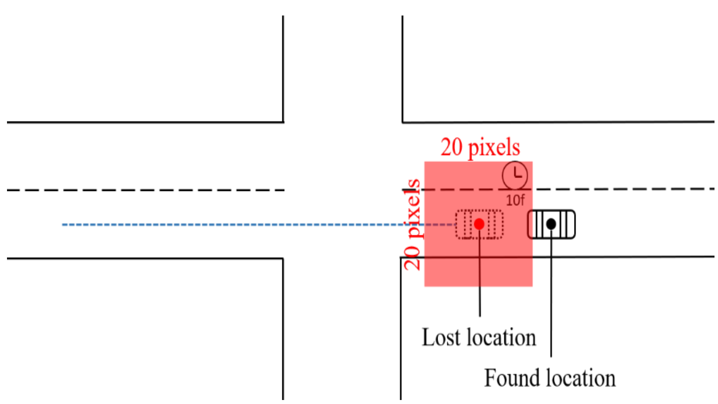

Figure 5. Anti-lost code explanation

c) Track object targets

Track the targets using KCF trackers and update the position of each car's tracking box. Two situations could be met: 1) tracking succeeds and target new tracking box is returned. 2) tracking fails, the vehicle's centroid position, vehicle number, and lost time are recorded separately in the aforementioned vectors. Delete this lost car from tracking targets to avoid further tracking.

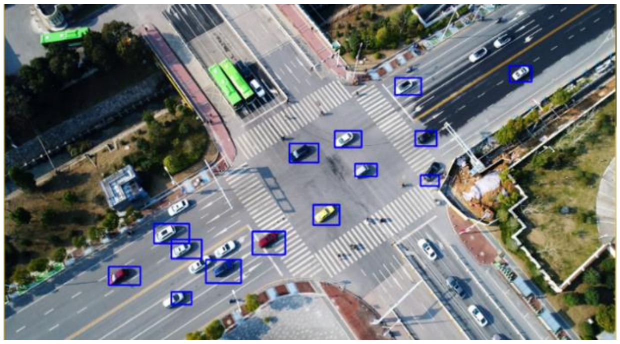

Figure 6. Tracking Box

Figure 7. Tracking status

In either situation, the position of vehicles will be recorded. When displaying each frame, draw vehicle trajectory using different colors. The users could vividly observe the tracking situation (Figure 7).

D. Calculation of vehicle speed

The two-dimensional position of each car of each frame is stored in a vector. Since the frame rate of the video of this project is 25 frames/s and the interval time between each frame is 0.04 s, it can be considered that the speed of the vehicle in this short time interval is constant, that is, the average speed is regarded as instantaneous speed at this point.

For each car, take the position of the adjacent two frames of the vehicle, calculate the distance between them, and convert it into the actual distance according to the scale. Dividing it by the interval time, the instantaneous speed of each track point is calculated.

E. Steering Recognition and Counting Volume

After getting the vehicle trajectory, we can make a logical judgment on the trajectory of each car from the appearance to the end. Knowing five intersection areas defined by the user and the vehicle coordinates, we could infer each car's starting point and ending point thus turning. Accumulated in time, traffic volume can be obtained during this period (excluding vehicles with incomplete tracks).

One the other hand, to obtain separate volumes for cars and trucks, it's necessary to judge the size of the vehicle. If the average size of a tracking box is greater than a certain threshold, it is considered as a large vehicle. The threshold is related to the resolution of the video and the shooting height of the UAV and should be determined by experiments to ensure the accuracy of the judgment.

F. Calculation of intersection passage time and time headway

Further, by using the trajectory and its corresponding number of frames, intersection passage time could be calculated as follows: judge and record the last frame when the car is in the initial area; judge and record the first frame when the car is entering the last area; the time lag between these two frames is the passage time.

Vehicles’ time headway could be calculated as: record vehicles’ lane which they are initially at, and the frame number when they cross the stop line. For each lane, time between two adjacent vehicles is their time headway.

G. Output

The output results include these parts:

a) Volumes of each vehicle movement from different approaches.
b) Excel file result_track, which records information of each vehicle of each frame, including vehicle number, total number of frames, the frame number when the vehicle appears, x and y coordinates, width and height of the tracking box (pixels), x-direction velocity (m/s), y-direction velocity (m/s), total velocity (m/s), velocity direction, etc..
c) Excel file result_veh, which records the summary information of each vehicle, including its vehicle number, movement, number of frames it appears, tracking box size (pixels), whether it is small car, the time duration it passes (s) the intersection and average speed (m/s).

CASE STUDY AND RESULTS

This method is evaluated on videos captured by a UAV equipped with a video camera. The objects are urban intersections in Nanjing, China. Each of them represents two typical type of intersection: a four-leg intersection and a three-leg intersection. The video was captured with the resolution of 4096 px * 2160 px at 25 Hz, and the camera was at a stable position with little interference. The duration includes at least two full cycles of intersection signal control to ensure cars run full course of their routes.

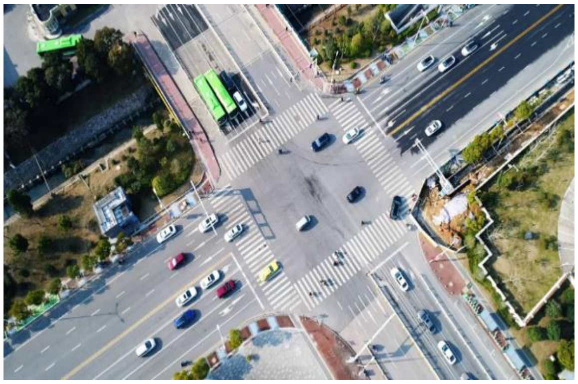

Figure 8. Four-leg intersection

Input the video and get the data progressed by the program and compare the estimated volumes with the true values which were counted manually. Figure 9 is a presentation of all the vehicle trajectories after a signal cycle’s processing. Each vehicle’s trajectory is marked by different color and a unique number.

There are two kinds of errors. One is missed vehicles, which is caused by either vehicles not being detected, or vehicles’ course being interrupted. The other one is false values, which is caused by detecting same vehicles repeatedly or detecting non-vehicle objects as vehicles.

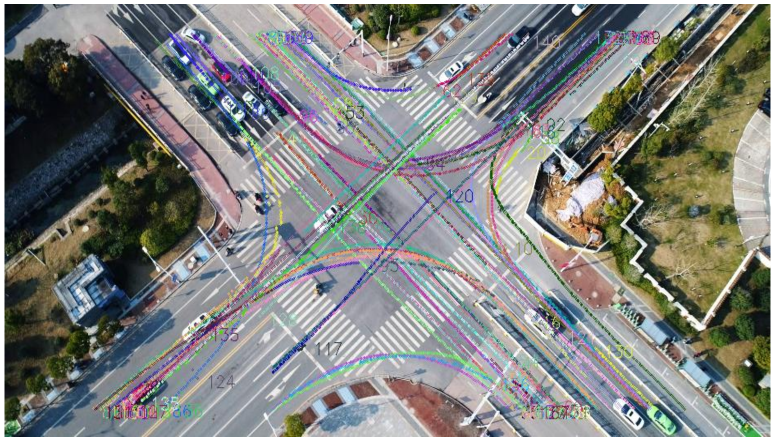

Figure 9. Vehicle trajectories presentation

When the program is running, we could clearly observe its operation status, for the vehicles being tracked and their course would be clearly displayed. In this way, we could find out the missed, the false values and the correct values.

Measured value + missed value – false value = true value

We use relative number of missed targets $\mathrm{MR}_{i,j} = \frac{MV_{i,j}}{N}$, where $\mathrm{MR}_{i,j}$ represents Missed Ratio for approach i and movement j, $MV_{i,j}$ represents missed value for approach i and movement j, and N is the total true vehicle numbers. And relative number of false tracks $\mathrm{FR}_{i,j} = \frac{FV_{i,j}}{N}$, where $\mathrm{FR}_{i,j}$ represents false ratio for approach i and movement j, $FV_{i,j}$ represents false value for approach i and movement j. So, the evaluation of detection accuracy would be:

$$DA = 100\% - \sum_{i,j} \mathrm{MR}_{i,j} - \sum_{i,j} \mathrm{FR}_{i,j}$$

where DA stands for detection accuracy.

Input the intersection video of two signal cycles, compare the measured vehicle numbers of each approach and for each movement are measured by the program with the true values (the numbers for comparison include both car and truck vehicle types). The results and accuracy are in the Table 1.

Table 1. Errors of the case

Sum	Estimated: 151	Missed: 20	False: 6	True: 165
Ratio	91.5%	$\sum_{i,j} \mathrm{MR}_{i,j} = 12.1\%$	$\sum_{i,j} \mathrm{FR}_{i,j} = 3.6\%$	100%
Detection Accuracy = 100% - (Missed + False) = 100% - (12.1%+3.6%) = **84.3%**				

The accuracy of the proposed method of this case is 84.3%. And the accuracy of a T-intersection case is 86.7%.

Speed

Draw trajectory-velocity bar charts of the vehicle in a three-dimensional space, where the x-axis shows the x-coordinate of the vehicle, the y-axis shows the y-coordinate of the vehicle, and the z-axis is the speed of the vehicle. Colors changing from blue to orange indicates the velocity changing from low to high. The following Figure 10 and Figure 11 shows the trajectories of an east-west through vehicle and a south-to-west left-turn vehicle. The changing pattern of the vehicle speed as they travel can be vividly seen from the figures.

In order to get the accuracy of the software's speed measurement, we did another experiment. We installed a GPS logger on the vehicle (Figure 12) and drove it in a T-junction back and forth for 15 times to collect data. And at the same time, we used a drone to take videos over the intersection. Compare the vehicle speed measured by the program with the speed measured by the GPS data.

Here is one instance. Figure 13 shows the comparison between true speed calculated using GPS data and the estimated value calculated by the program.

RMSD is the square root of the average of squared errors. The RMSD of an estimator $\hat{\theta}$ with respect to an estimated parameter θ is defined as the square root of the mean square error:

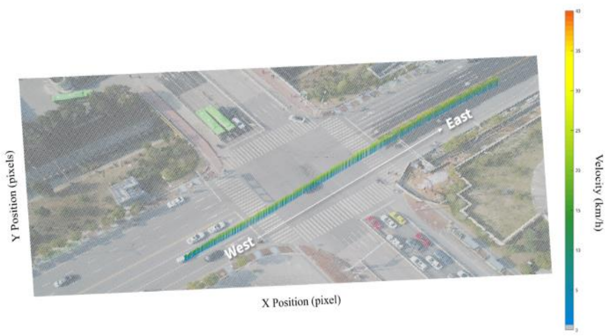

Figure 10. East-west through vehicle

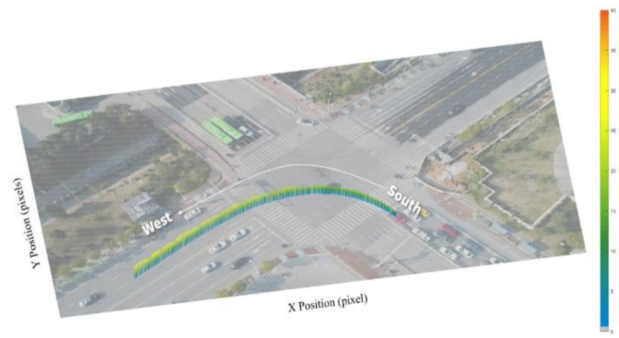

Figure 11. South-west left-turn vehicle

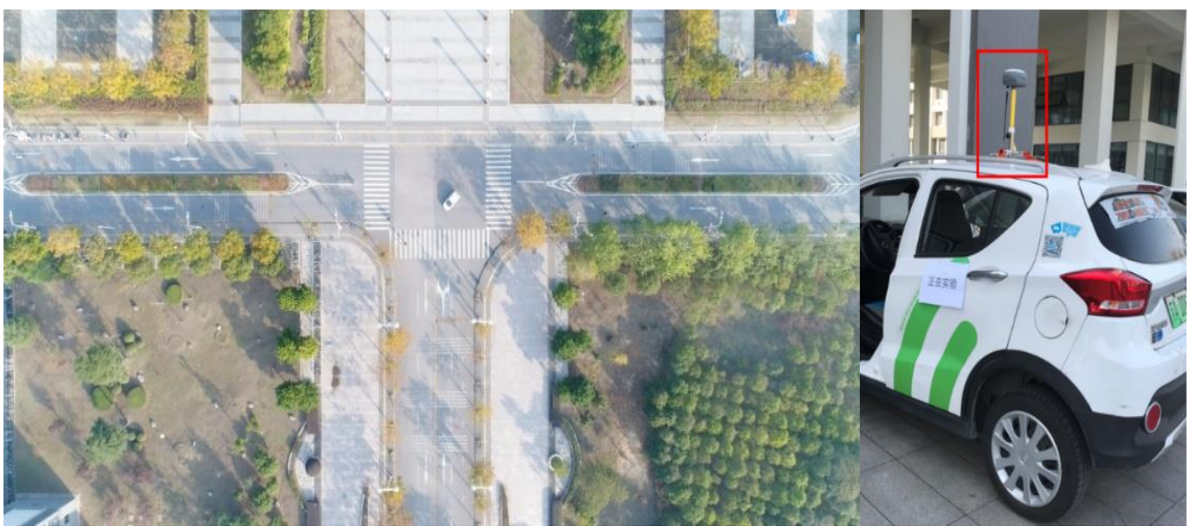

Figure 12. Experiment field and the GPS logger

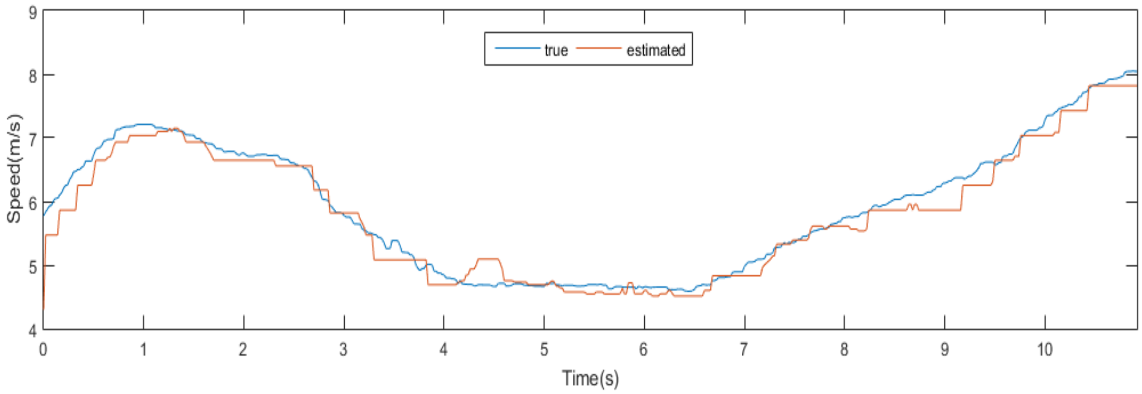

Figure 13. Comparison between real and estimated speed

$$\mathrm{RMSD}(\hat{\theta}) = \sqrt{\mathrm{MSE}(\hat{\theta})} = \sqrt{E((\hat{\theta} - \theta)^2)}$$

Table 2 shows the root mean square errors (RMSEs) of 5 comparative cases, each measures the difference between estimated speed by the program and the real value. RMSE is the standard deviation of the residuals (prediction errors) and the smaller it is, the more accurate the predicted values are.

The mean of the true speeds is approximately 6 m/s, so the normalized root-mean-square deviations range from 0.0015 to 0.0035 (RMSD/mean), which means the standard deviations of the residuals (measurement errors) in this case range from 0.15% to 0.35%, which is quite small.

Table 2. Medium error of estimated speed of vehicles

Vehicle No.	1	2	3	4	5
Turning	E-S, left	S-E, right	E-W, through	S-W, left	W-E, through
RMSE (m/s)	0.009	0.015	0.012	0.016	0.021

ERROR INVESTIGATION

When the program measures the number of vehicles passing through the intersection, it will cause errors due to the background model, the morphological operation after the background subtraction, the interference of the environment in the video to the target, etc., resulting in missed and false detection of the vehicle. There are many algorithms used to prevent tracking errors, but because of these irresistible factors, the measurement results will still produce certain errors, including track record error and speed measurement error. From the observation of tracking status while the program is running, following causes of the vehicle detection error are summarized.

a) When the background color is close to the vehicle target, the area of the vehicle after the background subtraction is too small, which is considered as a non-vehicle object, and thus missed. As shown in Figure 14, the color of the vehicle shown in the red rectangle is similar to the background, so it was not detected.

Figure 14. Instance 1

Figure 15. Instance 2

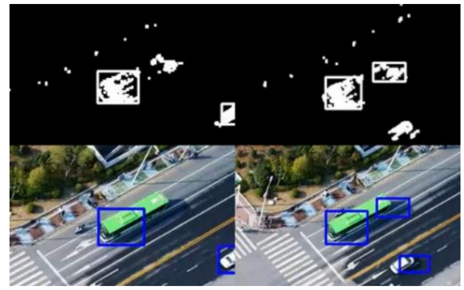

Figure 16. Instance 3

b) After the morphological operation of the two cars in the Figure 15, the target areas of the two cars cling together, causing the system to recognize it as only one car.

c) When the vehicle is large or its partial color is similar to the background color, after the background is subtracted, the target area of a vehicle would be broken into two or even more parts, so that the vehicle is repeatedly identified, as shown in Figure 16.

d) Environmental interference. The signal light railing in the Figure 17 extends to the center of the road, and when the car passes under the railing, the tracker loses track of the vehicle and stays at the railing. The moving vehicle is recognized as a new target, resulting in interruption of the trajectory. It isn't detected by the anti-losing code because the tracker doesn't lose track, but only stays at a non-vehicle object. Fortunately, case like this rarely happens.

Due to the loss of tracking and background interference, the trajectory of the car was interrupted into 3 segments, numbered 9, 13, 14 respectively, and its trajectory is shown in Figure 18.

e) For the speed measurement, the method is to calculate the distance of the same cars in adjacent frames, and the position unit is pixel. Therefore, the distance value is mostly an integer pixel value from 1 to 5, which leads to the measurement being approximated to the nearest integer value, and that causes measurement distortion and errors.

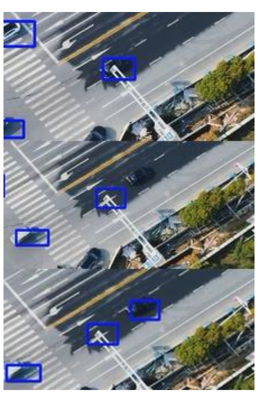

Figure 17. Instance 4

Figure 18. Instance 5

SUMMARY AND CONCLUSIONS

This paper presented a technique for extracting intersection volume and vehicle speed from aerial videos. By using Background subtraction and vehicle tracking, trajectories of vehicles are extracted, and then, by using certain algorithms, volume and speed are gained. The case studies show the accuracy of the program estimating volumes from four-leg intersection and three-leg intersection, which is 84.3% and 86.7%. Another case study presents the accuracy of speed estimation. The medium errors of estimated speed of vehicles are relatively low, which are around 0.01-0.02 m/s. The paper aims to show the method of counting traffic volumes and estimating vehicle speed in intersections. To further improve the accuracy, neural networks could be used for vehicle detection in the future.

ACKNOWLEDGEMENTS

This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1600200.

REFERENCE

Andrew, R. and Schrock, S. D. (2013). “Estimating turning movements at roundabouts using bluetooth technology,” TRB 92nd Annual Meeting Compendium of Papers.

Chen, A., Chootinan, Ryu, P., Lee, S., M., and Recker, W. (2012). “An intersection turning movement estimation procedure based on path flow estimator,” *J. Adv. Transp.*, vol. 46, no. 2, pp. 161–176.

Dey, B. and Kundu, M. K. (2019). “Turning video into traffic data – an application to urban intersection analysis using transfer learning.” *IET Image Process.*, Vol. 13 Iss. 4, pp. 673-679.

Ghanim, M. S. and Shaaban, K. (2019). “Estimating Turning Movements at Signalized Intersections Using Artificial Neural Networks.” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 5.

Julina, J. K. J., Sharmila, T. S. and Gladwin, S. J. (2019). “Vehicle Speed Detection System using Motion Vector Interpolation.” Global Conference for Advancement in Technology.

Li, J., Chen, S., Zhang, F. et al. (2019) “An Adaptive Framework for Multi-Vehicle Ground Speed Estimation in Airborne Videos.” *Remote Sens*, 11, 1241.

Luvizon, D. C., Nassu, B. T. and Minetto R. (2016). “A Video-Based System for Vehicle Speed Measurement in UrbanRoadways.” *IEEE Transactions on Intelligent Transportation Systems.*

Madasu, V. and Hanmandlu, M. (2010). “Estimation of vehicle speed by motion tracking on image sequences.” IEEE Intelligent Vehicles Symposium, pp. 185–190.

Martucci, A., Cerasuolo, G., Petrella, O. et al. (2020) “On the Calibration of GNSS-Based Vehicle Speed Meters.” *Sensors*, 20, 591.

Rad, A. G., Dehghani, A. and Karim, M. R. (2010). “Vehicle speed detection in video image sequences using CVS method.” *International Journal of the Physical Sciences*, Vol. 5, No. 17, pp. 2555–2563.

Wellbrock, G. A., Xia, T. J., Huang, M. et al. (2019) “First Field Trial of Sensing Vehicle Speed, Density, and Road Conditions by using Fiber Carrying High Speed Data.” Optical Fiber Communications Conference and Exhibition, pp. 1–3.