



# Multiple Objects Tracking Based Vehicle Speed Analysis with Gaussian Filter from Drone Video

Yue Liu, Zhichao Lian<sup>(✉)</sup>, Junjie Ding, and Tangyi Guo

Nanjing University of Science and Technology, Nanjing, China  
1294384195@qq.com, lzcts@163.com

**Abstract.** Vehicle speed analysis based on the video is a challenging task in the field of traffic safety, which has high requirements for accuracy and computational burden. The drone's video is taken from a top-down perspective, providing more complete view comparing to the common surveillance cameras in poles. In this paper, we introduce a Gaussian Filter to deal with the estimated speed data which are extracted by a multiple objects tracking method composed of You Only Look Once (YOLOv3) and Kalman Filter. We exploit the capability of Gaussian Filter to suppress data noise appearing in the process of tracking vehicles from drone videos, and thus use the filter to solve the case where the estimated vehicle speed is fluctuated along the ongoing direction. On the other hand, we built a vehicle dataset from the drone's videos we mentioned above which additionally contains vehicle's real speed information. Experimental results showed that our method is effective to improve the accuracy of vehicle speed estimated by our tracking module. It can improve Mean Squared error (MSE) accuracy 80.5% on experimental data.

**Keywords:** Gaussian Filter · Data analysis · Multiple object tracking · Traffic safety

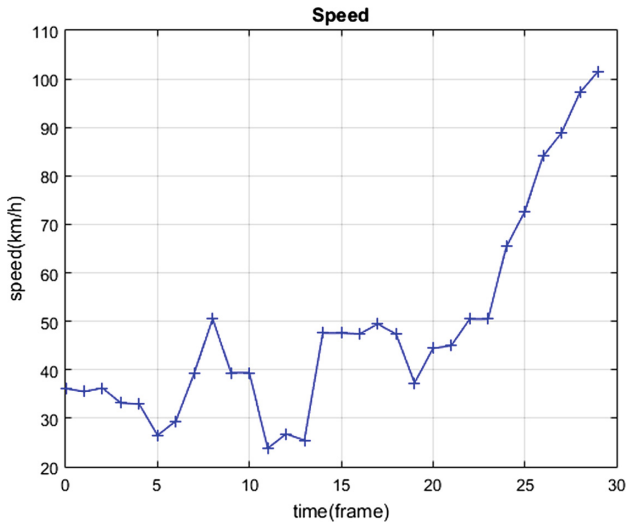
## 1 Introduction

Quickly and accurately extracting vehicle related data from video is a fundamental topic in unmanned driving [11] and road monitoring. In the transportation field, the vehicle speed and density are two important measurements in the vehicle travel. Speeding and other illegal acts will be detected, which has a great impact on traffic safety. This topic is of broad interest for potential applications of traffic supervision, data analysis and so on.

There are many ways to obtain relevant data on the vehicle's travel, which mainly be categorized in two classes. The first one is to analyze the complex signal generated by electronic devices such as radar [8] and vehicle's own attached sensors [9, 10].

The other direction is based on cameras or the like to capture video for object detection and tracking [4]. Road monitoring can cover a certain range of vehicles and the vehicle's data can be measured by the objection detection and tracking method in computer vision [13]. However, there will be a large deviation in the speed measurement due to different monitoring camera's pitch angles to the near and far objects.

We show this phenomenon in Fig. 1. Since the position of the surveillance camera is generally low, the vehicle will have occlusion and overlap in the road, which raises many problems for analysis algorithms.



**Fig. 1.** When some devices measure that the vehicle is approaching a uniform speed, the phenomenon shown in the figure will appear: the speed in the first half is fluctuating up and down, and the speed in the second half is suddenly rising.

Nowadays, it is possible to extract motion information from video sequences thanks to the advancement of multiple object tracking. The pioneer work is to improve the accuracy of object detection like YOLOv3 trained on PASCAL VOC and Microsoft COCO datasets include more labels, such as person, boat and vehicle, etc. On the other hand, more excellent matching mechanism make multiple object tracking better.

State-of-the-art multiple object tracking are based on public datasets which contains multi-classes include oblique perspective vehicle. The model perform not well in vehicle tracking tasks from a special top-down perspective. These problems are due to less information of the vehicle in this view, like the vehicle in drone's video.

The development of drone equipment makes it possible to capture highways at a fixes altitude. This greatly ensures that the scene in the video are more realistically restored and then accurate vehicle data through certain technologies can be obtained. The drone has a panoramic view, which makes the vehicle in the video scene more complete and has no larger scale changes. High-altitude overhead allows for no occlusion between vehicles, simplifying the scene and simultaneously monitoring all vehicles in the scene.

Using video captured by drones, we design a vehicle speed analysis framework that tracks the vehicles in real time, calculates the speed by traces and corrects the speed of vehicles. Through experimental result, it has been proven that the framework can

effectively calculate the running speed of vehicle and has a good performance in suppressing the noise occurring in the measurement process. Our main contributions are threefold.

- We implement a real-time multiple object tracking framework based on the YOLOv3 detection system and Kalman Filter for UAV video.
- We propose a Gaussian Filter [5] to remove the noise from the vehicle's data and refine the calculation of vehicle speed.
- We build a vehicle dataset from a large number of drone videos that contains the actual speed of the vehicle at the moment.

## 2 Related Work

In the following, we review known public traffic scene datasets and drone datasets. Almost all traffic scene datasets consist of images from in-vehicle devices and surveillance videos. In most samples, the scales of the vehicle vary widely, and many vehicles have occlusion from each other like KITTI. For some drone datasets, the proportion of vehicle samples is small and the most critical is that the actual speed of the vehicle is not included in the datasets like the Stanford Drone Dataset.

In order to obtain information such as speed and density in a video sequence, a real-time, stable and accurate tracking framework is often required. This framework is usually combined with the detector and motion prediction model. Driven by YOLO which relies on darknet-53 network, higher accuracy vehicle predictions and real-time detection results are achieved. Prediction of objects in the next frame which utilizing time and space information of vehicles in a video sequence by add Kalman Filter to the tracking framework. To correct the vehicle speed and other data generated by the tracking framework, we proposed a kind of Filter.

Filter selection is roughly divided into two categories. One line is based on some transformation. The data is transformed from the spatial domain to the frequency domain by Fourier Transform [14], processed in the frequency domain, and then processed to the spatial domain by inverse transform [15].

The other method is spatial domain filter. Since they process data or signals directly without any transformation, like operating directly on the pixel in the image. This approach takes advantage of the distribution of data. In the experiment, we notice that the vehicle's data approximates to a certain distribution, and spatial domain filtering is more intuitive and simpler than frequency domain filtering with less computational complexity.

We start with our observation and analysis of the large flutter of the bounding boxes which occurs in object tracking process. The large flutter mentioned above means the coordinates of the bounding boxes are not accurate enough. This is because the tracking framework is not stable enough [1], but actually this is inevitable. So we perform a statistical analysis of the results of the tracking and try to fix it.

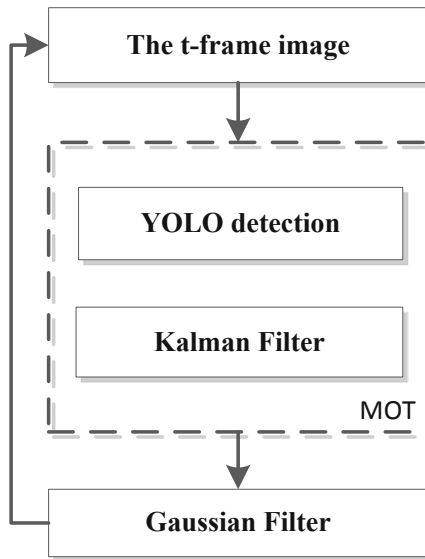
To make a good performance in correcting vehicle data, we design a Gaussian filter based on video characteristics and vehicle data distribution rules. This module takes

advantage of the fact that the vehicle data closes to the Gaussian distribution [7] and then remove noise by the filter.

Our experiments show that the Gaussian filter has a good effect when dealing with the speed data of moving vehicles in drone video. We use MSE to evaluate our effect on vehicle speed correction. In the testing dataset, data fluctuations in vehicle speed were significantly reduced and closer to truth values. Yet there is still much room to exploit in data analysis and data processing.

### 3 Our Proposed System

With above analysis, we design a real-time detection system of vehicle based on the YOLO model from the UAV video [16]. In addition, a Kalman Filter has been combined in the system to implement the tracking module [17, 18]. Finally, a Gaussian Filter has been involved to suppress the noise from the result obtained by the tracking module, and more accurate vehicle speed estimations are achieved by the proposed system in Fig. 2.



**Fig. 2.** The proposed system has a Gaussian Filter and a multiple object tracking module consists of YOLO detection and Kalman Filter

#### 3.1 YOLO Detection System

The YOLOv3 detection system [1] is a state-of-the-art framework in real-time object detection, here we use YOLO to refer to it. YOLO creatively uses anchor boxes in network design to get direct location prediction. The image is divided into a  $7 \times 7$  grid and each grid cell has 9 anchor boxes [2, 6] which are obtained by clustering and have

fixed dimension to predict the possible bounding boxes and locations of the object. Although this makes the model predicts more than thousand, the objects are more detectable. In this improvement, the 9 anchor boxes have less coverage on a small scale and  $7 \times 7$  cells to predict the objects which YOLO used make the  $(x, y)$  locations of bounding boxes has some instability.

### 3.2 Kalman Filter

Kalman Filter [3] is a linear filter which can be descrambled by linear stochastic difference equation. It keeps track of the estimated state of the system and the variance or uncertainty of the estimate.



**Fig. 3.** Since the drone has a panoramic view, the vehicle is crowded but there is no occlusion in the UAV video.

In our drone video, all vehicles are in the lanes and always heading in the same direction. The most important point is that the division between the vehicles is very clear and there is no occlusion at all and the contour of the vehicle remains the same. Some examples are shown in Fig. 3. So Kalman Filter is satisfactory for removing noise and predict the location of vehicle changes in that case.

### 3.3 Gaussian Filter

To calculate more accurate vehicle speed, we introduce the Gaussian filter, which proves that it has a significant effect on correcting tracking data [19, 20].

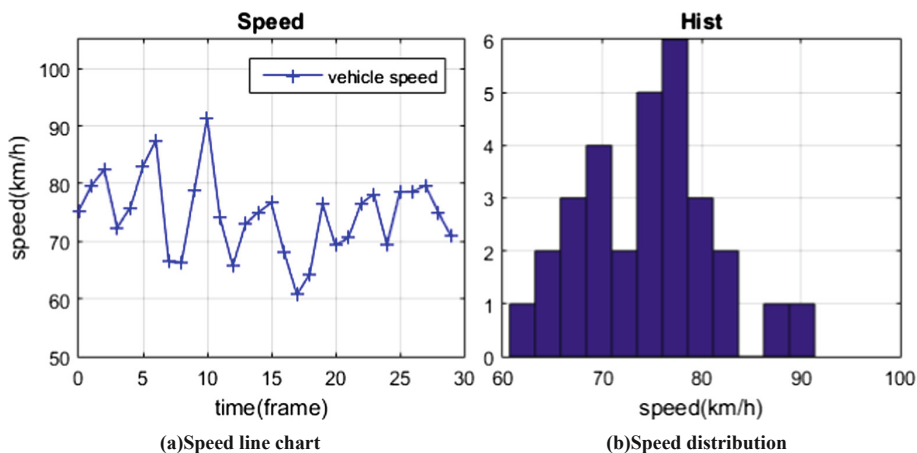
In any object detection framework and scientific calculation, errors are inevitable. If the error is within the allowable range, the data results are also acceptable. Although YOLO has taken some strategies to improve the accuracy of location, the prediction of the locations still has some instability [2]. In particular, anchor boxes which YOLO used to help CNN [22] locate targets make an unstable error between the coordinates of bounding boxes and the ground truth [21]. We address this issue by using a suitable Gaussian filter.

The Gaussian filter is a signal processing method as a filter whose impulse response is a Gaussian function, which is commonly used in eliminating Gaussian noise. The tracking data points recorded by the vehicle just tracked or tracked at the end tend to fluctuate greatly, while more stable during the tracking. Sudden acceleration or

deceleration of the vehicle can also cause large deviations in the data calculated by our tracking module. This means that tracking sometimes lags behind the actual displacement of the vehicle. Gaussian filter is helpful in this regard to modify those mutation data.

Taking vehicle speed as an example, actually, the speed of the vehicle can be regarded as a constant value in one second. In the actual experiment, we use “frame” as the speed measurement unit instead of seconds for more accurate calculation data.

When measured the speed of the vehicle which can be calculated by  $\frac{|x_p - x_n|}{1/fps}$  in every frame, where  $fps, x_p, x_n$  denote frames per second,  $x$  coordinate of current frame,  $x$  coordinate of previous frame respectively. We find that the calculation speed per frame is quite different but regular. We call this phenomenon “data flutter” in Fig. 4, the tracking data float around the true value and satisfy a certain distribution law. According to the characteristics of the data distribution, the correction of vehicle speed can be achieved by utilizing suitable Gaussian filter.



**Fig. 4.** Illustration of data flutter. (a) In one second (assuming the video frame rate is 30), the phenomenon that the forward speed of the uniform moving vehicle in each frame of the image is fluctuated greatly. (b) The vehicle speed calculated by tracking framework is close to the normal distribution.

To analyze these observations, errors are partially related to the precision of the multi-target detection algorithm framework and the video itself has a higher resolution. Since the data has a certain distribution law, a filter with suitable parameters can greatly improve the tracking data results.

We propose a Gaussian Filter to apply to vehicle speed data. We perform confirmatory experiments to show the difference in Sect. 4.

## 4 Experiments

Our improvement works on vehicle tracking data in drone video. The method we used effectively corrects vehicle tracking data and improves detection accuracy. We notice that although there are many public tracking datasets, they didn't provide motion information for targets in the real world such as speed. So we carry our experiments on our drone video dataset, which contains 45 car tracking data groups, real speed of the vehicle, and 4834 usable data records for training and testing.

For a tracking system, components selected in the framework play an extremely important role in terms of real-time and accuracy. These often determine the efficiency of the tracking and the accuracy of the detection. Our implementation is based on excellent algorithm YOLO which has a good performance in real-time and accuracy as well as a classic prediction method Kalman Filter, then the most important one is that we add a suitable Gaussian filter to smooth the tracking data about vehicle speed.

### 4.1 Implementation Details

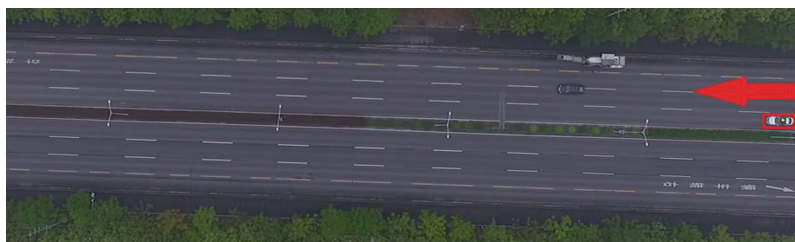
We train 2750 pictures of vehicles taken from drone at a certain height with YOLOv3. We use a high resolution  $2704 * 1520$  or  $1920 * 1080$  picture. The vehicle is usually small in the UAV view. We normalize the image size to  $416 * 416$  and set learning rate and momentum to 0.001 and 0.9. We take the steps strategy and reduce the learning rate by 10 times when the number of training is 5000, 8000, 12000. The maximum iteration number to stop training is 15000. For data augmentation, the value of the saturation and exposure parameters we using are both 1.5 and the hue is 0.1. The part of data augmentation increases the amount of data to some extent.

Kalman Filter is a classic approach to linear filtering and prediction problem. For Kalman Filter, we set delatime to 0.2 to make target more "massive". Since the acceleration is not clear, it is assumed to be a process noise. We set the value of `Accel_noise_mag` to 0.1.

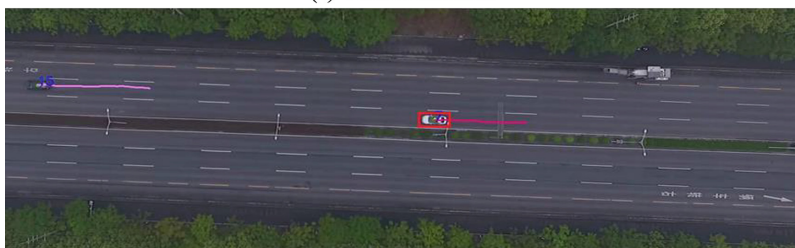
Whether the detected objects in the video sequence are regarded as the same object adopts the IOU judgment strategy.

The observation of bounding box within a small range flutter made we think about how to stable the detection boxes. By analyzed the tracking data of every car, the error between predicted value and truth value is close to the certain normal distribution in Fig. 4. A Gaussian filter with suitable parameters can yield great performance in the tracking data from analyzing the video.

**Datasets.** Existing available public tracking datasets lack object motion information while this is part of our system's main contribution. Therefore we built our own vehicle dataset from the drone video which contains its speed information recorded by speedometer. Some examples in Fig. 3. We collect the data points with  $N$  (e.g., 30) frames and group them in each second of each video. The data in the Table 1 comes from the video with a frame rate of 30 fps, so every second has 30 data points for analysis. The well grouped data is divided into 810/270 data points for training and testing.



(a) Vehicle direction



(b) Tracking target vehicle

**Fig. 5.** The vehicle selected by the red rectangle is the target vehicle. (a) The red arrow indicates the direction of motion moves from the right to the left and (b) the red thin line is the trajectory tracked by the vehicle. (Color figure online)

**Performance Measure.** The process of data analysis is considered as a regression task, so mean square error (MSE) is used as the main performance measure.

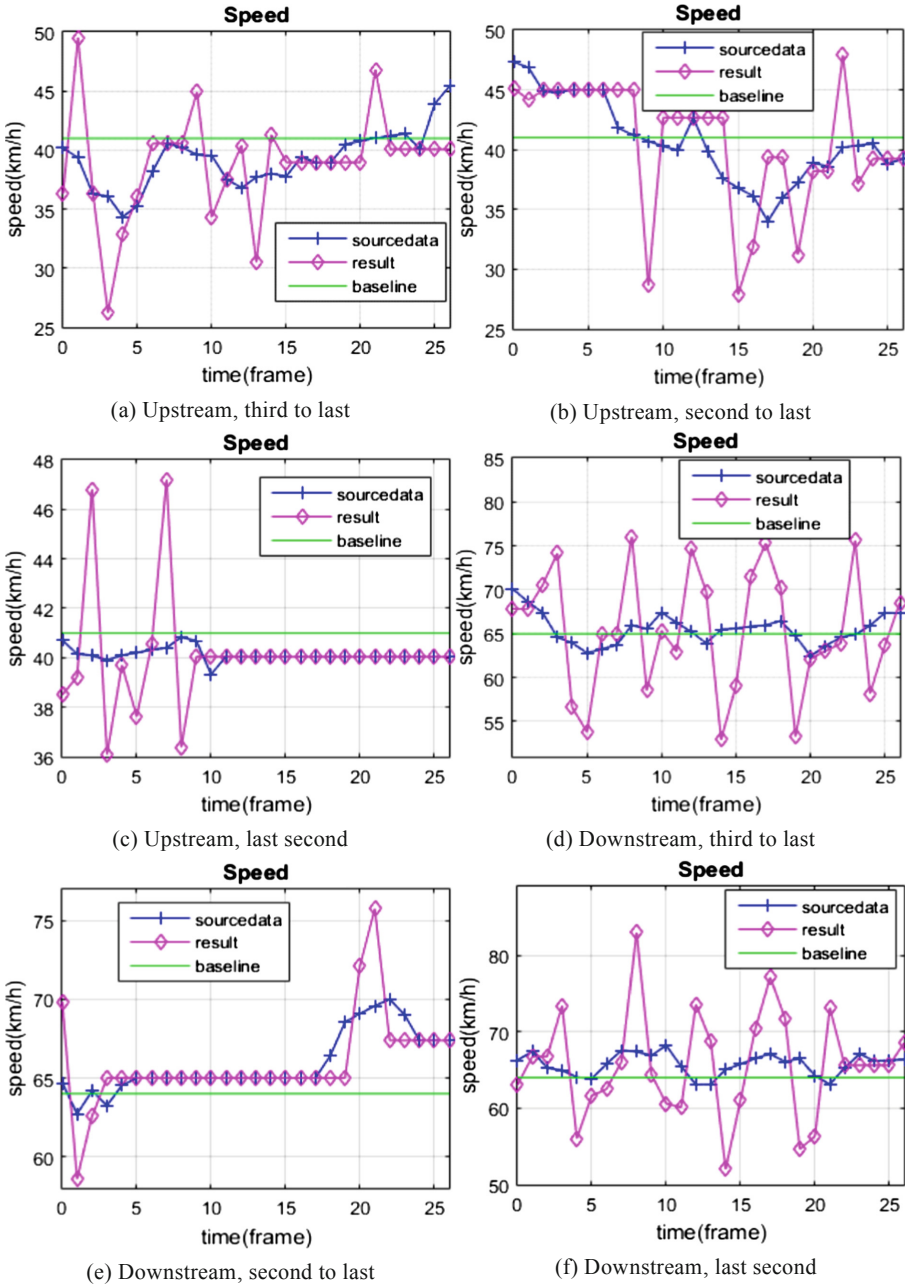
We choose the same car in the same section of the road just for justice, and we pick some videos both in the upstream and downstream directions to test.

**Upstream Vehicle Video.** In the video, “upstream” means that the car drives from right to left in the scene of the video. As shown in Fig. 5, our tracking framework completes the tracking well. We use the tracking data of the previous time that the target vehicle appears in the video scene as the training data to calculate the parameter estimation of the corresponding Gaussian filter, and apply it to the estimation and correction of the vehicle speed for a later period of time.

For this video, we use about 270 data points about the speed of the target vehicle to analysis  $\mu$  and  $\sigma$  which makes us get a suitable Gaussian filter. Then using the filter to correct the vehicle speed data for the last three seconds of the video.

As shown in the first row and the first column of the second row in Fig. 6, our work is excellent in tracking data. The red lines with diamonds are drawn from unprocessed vehicle speeds. The blue lines is composed of vehicle speed data processed by a Gaussian filter. After the last three seconds of the tracking data of the video is processed by a Gaussian filter, it can be clearly seen that the speed of the vehicle with blue lines is largely stabilized and the jitter is significantly reduced and is closer to real vehicle speed as shown with green line. Our performance measure MSE improves about 71.73%, and another set of data improved about 98.6%. The result as shown in the first three rows of the Table 1.

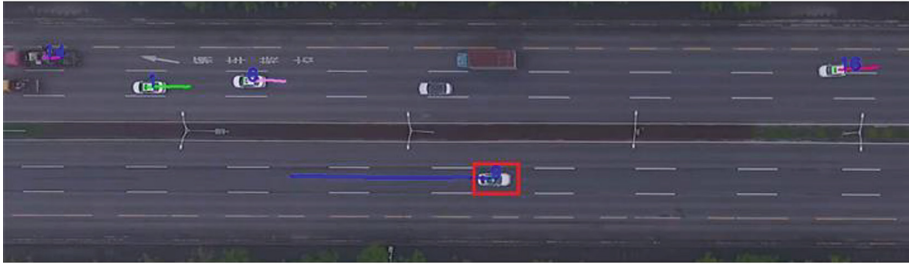




**Fig. 6.** Overview of our data result graph. Speed line chart of the upstream vehicle obtained in the last third second of the video with (a). The chart of second and last second as shown in (b) and (c). Speed line chart of the downstream vehicle obtained in the last third second of the video with (d). The chart of second and last second as shown in (e) and (f). (Color figure online)



(a) Vehicle direction



(b) Tracking target vehicle

**Fig. 7.** The vehicle selected by the red rectangle is the target vehicle. (a) The red arrow indicates the direction of motion move from left to right and (b) the blue thin line with the target vehicle is the trajectory tracked by the vehicle.

**Table 1.** Evaluate vehicle speed with MSE on test dataset. We choose three cars called “Car 1, Car 2, Car 3” in the video count down three seconds denoted by “Time 1, Time 2, Time 3”. The vehicle’s label is from the chosen cars that appears in different videos. The data in bold is the better group. Their MSE performance is better. Their mean and the output error is smaller.

Vehicle	Raw data MSE	Result MSE	Improve (%)	AVERAGE
Car1_Time1	<b>20.20234</b>	<b>5.715011</b>	<b>71.711143</b>	<b>2.813135</b>
Car1_Time2	27.54991	11.71515	57.476631	2.923800
Car1_Time3	<b>4.999145</b>	<b>0.077600</b>	<b>98.447735</b>	<b>1.091221</b>
Car2_Time1	47.59822	3.123054	93.438717	5.439575
Car2_Time2	8.936934	3.644574	59.218967	1.169905
Car2_Time3	47.67669	1.981989	95.842855	5.487424
Car3_Time1	44.39706	13.19164	70.287132	4.698314
Car3_Time2	<b>42.83019</b>	<b>3.381573</b>	<b>92.104698</b>	<b>4.874536</b>
Car3_Time3	<b>6.792341</b>	<b>1.213208</b>	<b>82.138588</b>	<b>1.567593</b>

**Downstream Vehicle Video.** In the video, “downstream” means that the car drives from left to right in the scene of the video, as shown in Fig. 7. This set of data contains 210 tracking data points about the speed of the same vehicle which appearing in the “downstream vehicle video” for training and 90 tracking data points for testing. The testing data points are divided into 3 groups. They are the last three seconds of the



**Fig. 8.** Illustration of vehicle tracking in drone video. The line following the vehicle is the trajectory of the vehicle.

vehicle's speed data. Methods trained the 210 tracking data to get an available Gaussian filter for testing. The result as shown in the middle three rows of the Table 1.

As shown in Table 1, with the suitable Gaussian filter, the three sets of tracking data grouped by vehicle after processing are well corrected, so that the data is distributed near the real data baseline with less jitter and error. The method can improve the MSE of the data about 59.21%–95.84%. We also use the “AVERAGE” about the errors to measure the noise suppression. The MSE from the table is significantly smaller which proved that vehicle speed at different times is closer to the true value by using our method. The “AVERAGE” indicates system output error is less than before. Finally, the system will output these processed data like vehicle speed which can be applied in the analysis process of traffic.

Statistics in Table 1 show that the Gaussian Filter has notable advantage in improving accuracy of the vehicle speed and the like obtained by tracking framework. The above experiments are all in vehicle dataset which contains the real speed of vehicles.

## 5 Conclusion

After constructing a vehicle dataset containing motion information of vehicle, we introduce YOLO and Kalman Filter to create a tracking module which for motion information such as vehicle speed. This module can better track the vehicle appearing in the drone video in real-time in Fig. 8. By statistically analyzing on the vehicle speed extracted by the tracking module, we propose a kind of Filter to refine the vehicle speed which gets an excellent improvement with MSE evaluation on vehicle dataset we built. The Gaussian Filter we used provides a nice feature in removing noise or present in vehicle speed. In the last part of the system, we obtain the corrected more accurate vehicle speed.

We hope the implementation details publicly available can help the community adopt these useful strategies for dealing with the tracking data and advance related techniques.

## References

1. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement (2018)
2. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: IEEE Conference on Computer Vision & Pattern Recognition (2017)
3. Kalman, R.E.: A new approach to linear filtering and prediction problems. *J. Basic Eng. Trans.* **82**, 35–45 (1960)
4. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: Computer Vision & Pattern Recognition (2013)
5. Ito, K.: Gaussian filter for nonlinear filtering problems. In: IEEE Conference on Decision & Control (2002)
6. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. arXiv preprint [arXiv:1506.01497](https://arxiv.org/abs/1506.01497) (2015)
7. Perret-Gentil, C.: Gaussian distribution of short sums of trace functions over finite fields. *Math. Proc. Camb. Philos. Soc.* **163**(3), 38 (2017)
8. Eaves, J.L., Reedy, E.K.: Principles of Modern Radar. SciTech Publishing, Chennai (2013)
9. Yun, D.S., et al.: The system integration of unmanned vehicle and driving simulator with sensor fusion system. In: International Conference on Multisensor Fusion & Integration for Intelligent Systems (2002)
10. Im, D.Y., et al.: Development of magnetic position sensor for unmanned driving of robotic vehicle. In: Sensors (2009)
11. Zhang, X., Gao, H., Mu, G., et al.: A study on key technologies of unmanned driving. *CAAI Trans. Intell. Technol.* **1**(1), 4–13 (2016)
12. Setchell, C., Dagless, E.L.: Vision-based road-traffic monitoring sensor. *IEE Proc. – Vis. Image Signal Process.* **148**(1), 78–84 (2002)
13. Li, C., Dai, B., Wang, R., et al.: Multi-lane detection based on omnidirectional camera using anisotropic steerable filters. *IET Intell. Transp. Syst.* **10**(5), 298–307 (2016)
14. Zhe, Y., et al.: Filter design for linear frequency modulation signal based on fractional Fourier transform. In: IEEE International Conference on Signal Processing (2010)
15. Soo, J.S., Pang, K.K.: Multidelay block frequency domain adaptive filter. *IEEE Trans. Acoust. Speech Signal Process.* **38**(2), 373–376 (1990)
16. Alt, N., Claus, C., Stechele, W.: Hardware/software architecture of an algorithm for vision-based real-time vehicle detection in dark environments. In: Design, Automation & Test in Europe (2008)
17. Zhao, Z., Ping, F., Guo, J., et al.: A hybrid tracking framework based on kernel correlation filtering and particle filtering. *Neurocomputing* **297**, 40–49 (2018)
18. Wei, C., Zhang, K., Liu, Q.: Robust visual tracking via patch based kernel correlation filters with adaptive multiple feature ensemble. *Neurocomputing* **214**, 607–617 (2016)
19. Strait, J.C., Jenkins, W.: Filter architectures and adaptive algorithms for 2-D adaptive digital signal processing. In: International Conference on Acoustics (1989)
20. Reid, D.B., Bryson, R.G.: A non-Gaussian filter for tracking targets moving over terrain. In: Asilomar Conference on Circuits (1979)
21. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E.: SSD: single shot multibox detector. *CoRR*, abs/1512.02325 (2015)
22. Szegedy, C., et al.: Going deeper with convolutions. *CoRR*, abs/1409.4842 (2014)