Real-Time Speed Estimation of Vehicles from Uncalibrated View-Independent Traffic Cameras

Anil Rao Y G, Sujith Kumar N, Amaresh H S, Chirag H V
Electronics and Communication Engineering
R V College of Engineering
Bangalore, India

***Abstract*—This paper presents a comprehensive solution for implementing a processing module on traffic cameras that is capable of tracking every vehicle in the camera frame and estimating its speed in real-time. A hybrid framework for multiple vehicle tracking is employed that utilizes Kalman filter and Hungarian Algorithm to resolve occlusions. A speed estimation methodology is described that is robust enough to work with camera feed from any angle without calibration and camera mounted at a minimum height of 7m. The system has been tested on computer-generated sequences as well as real-life scenarios and speed estimates have been obtained with maximum error of less than 3kmph.**

***Index Terms*—Region of Interest, Kalman filter, Hungarian algorithm, multiple tracking, camera view, top view, occlusion, homography, perspective transform, frames per second (FPS).**

I. Introduction

Vehicular density and speed are the main parameters required in intelligent traffic management systems that also include traffic violation detection. Hence, several researchers have proposed wide-ranging solutions for the same. In [1], Wireless Magnetic Sensor Networks are utilized to estimate the speeds of passing cars. The authors have designed the system as hierarchy architecture to reduce power consumption even with the use of 3 WSN nodes compared to standard 2 nodes model. The major drawbacks are that the system is not scalable to simultaneous multiple vehicle speeds estimation spanning the entire road width and detecting high speed vehicles. In [2], the authors describe a novel vehicle detection and speed estimation method that detects traffic density by leveraging common wireless networks such as Wi-Fi or cellular. The advantage of this solution is that it requires no specialized hardware such as GPS-enabled cars or smart phones. On the other hand, the speed estimation module is only able to provide a single estimate for the entire lifetime of a vehicle in the frame. Also, the estimation has an accuracy of only 90% making it infeasible for speed violation detection.

Due to the ubiquity of traffic cameras, speed estimation research has been steadily moving towards video-based methods. This is because of the cost-effectiveness with no or less extra hardware requirements, scalability and immense key traffic parameters obtainable by traffic video processing. A simple 1D vehicle speed assessment has been carried out in [3] which is not compliant with the real scenarios since most of the cases are 2D and not top view measurements. In [4] the authors have proposed a method of vehicular speed estimation based on spherical projection and in [5] speed estimation using optical flow [6] for a side view. While the algorithms work with un-calibrated cameras, they are limited to vehicles travelling in 1D and the camera must be placed in-line with the vehicle motion. Also, the methods feature a weak background subtraction technique that yields poor results in the presence of camera noise. In [7] only average speeds are estimated restricted to 1D motion. Also, the main drawback is that calibration method proposed depends significantly on detected vehicle length which results in less accurate results due to poor vehicle length extraction from low resolution and noisy camera frames. The work proposed in [8] was estimation of Mean Vehicle Speed using Calibrated cameras. The main disadvantage in this method is that changes in camera parameters like camera angle with respect to road etc. have to be constantly monitored and updated manually which is a tedious work. In [9], the authors have developed a camera calibration model similar to [8] for speed estimation with lesser calibrations and no markers. But the disadvantage is that the technique is limited by the camera setup angle. Authors in [10] develop a speed estimation method suppressing camera vibrations using background compensation. But due to the Interframe difference (IFD) technique they adopted the error rate in speed estimations at low speeds are high. This is a major drawback in high vehicular density cases where most of the vehicles have low speeds. Also the previous work requires manual updating of scale factor for speed determination. [7–10] are limited to single road lane and work only in case of vehicles moving either towards or away from the cameras. In [11] simple and novel approach has been proposed to estimate speeds of vehicles at night times using headlights with the help of low speed shutter cameras. But the proposal requires a top oblique view of the road lane which is not the case in most of the situations. A computationally economical technique was proposed by authors in [12] capable of carrying out at 60FPS. But the proposal mainly relies on standard markers to develop the Homography matrix for speed estimation which is a disadvantage when the cameras field of view change and the speeds computed at 95th percentile of tracking is taken for comparison against ground truth. Also the proposal efficiency greatly decreases at high speeds since it takes into account the speeds near the end of the tracking. [13] is based on tracking with the help of license plates of vehicles. This method is

978-1-4799-8641-5/15/$31.00

limited to only camera is closer to the vehicles and at a particular angle and not applicable to multiple lanes. Also during occlusion when one vehicle is behind another this method fails in tracking. All the above speed estimation methods based on image processing discussed are mainly limited in tracking vehicles at particular camera angles, adapting to the changing camera parameters and failure during small camera tremors.

In this paper, we develop a smart module to track multiple vehicles and estimate their speed in real-time which utilizes the traffic cameras to generate and transfer information to the server. A combination of Kalman filter and Hungarian assignment solver is used to improve tracking performance significantly by correctly mapping centroids obtained from foreground detection algorithm to the corresponding vehicle objects. The strength of the speed estimation lies in its invariance to the camera setup and very less calibrations. Only 4 points on a road lane have to be chosen during initial setup of the module. These 4 points provide ROI on which the module processes which decrease the computational complexity and focuses only on moving vehicles. This is extended to multiple lanes which can be tracked simultaneously.

II. System Overview

The proposed system consisting of 4 modules as shown in Fig 1 has been implemented on BeagleBoard- xM at 10 FPS and on Lenovo T440 ThinkPad at 30 FPS. The first stage consists of background subtraction wherein the foreground objects, i.e. vehicles are extracted as binary images. The output of this stage is used in the centroid identification block to obtain the vehicle centroids which is fed into the multiple vehicle tracking module. This module employs Kalman filter along with Hungarian assignment solver. The final stage is the speed estimation module that is assisted by a homography-based transformer which transmutes the vehicle tracking coordinates into another space. Each of these blocks is explained in detail in the following sections.

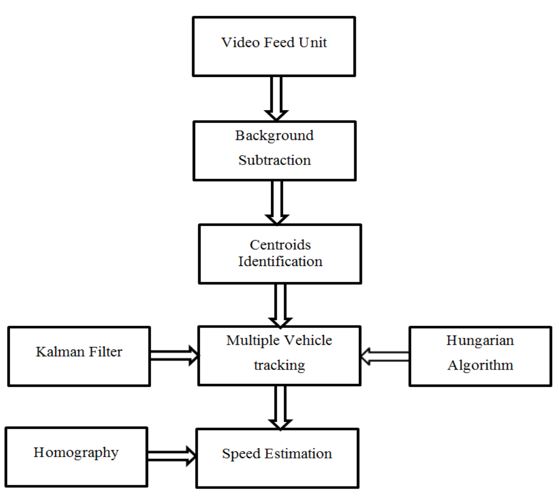
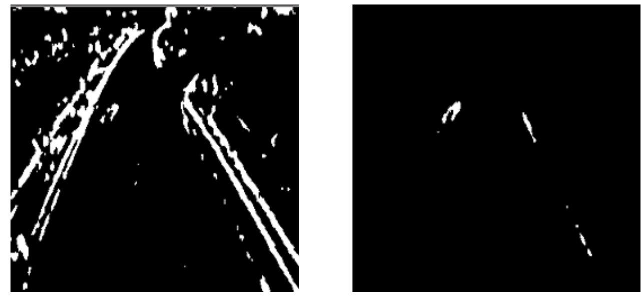


Fig. 1. Processing Module Block Diagram



(a) (b)

Fig. 2. Effect of camera tremor on Background Subtraction (a) Without ROI (b) With ROI

III. Foreground Object Identification

Most of the computer vision and image processing applications that involve some form of detection or tracking require background subtraction as the preliminary step. It provides binary output frames when used on traffic videos. Each blob that is detected corresponds to a vehicle whose centroid has to be identified for tracking.

A. Background Subtraction

Background subtraction refers to the image processing technique where foreground objects are extracted from dynamic backgrounds in the presence of shadows and camouflaging.

In the proposed solution, a ROI is defined initially by specifying four points on road and all processing is performed only on the ROI. This has a threefold benefit firstly, it ensures that the background is mostly static except for gradual changes of illumination; secondly, the tracking of unnecessary objects (such as pedestrians) is avoided; lastly, the execution speed is improved as the computation has to be performed on a reduced set of pixels.

ROI extraction enables the use of background subtraction algorithms such as Mixture of Gaussians that can be implemented on real-time embedded systems. With the use of ROI camera tremor effects can be eliminated easily and the results are shown in Fig 2. Since the effectiveness of background subtraction decides the accuracy of the system to a large extent, a more sophisticated algorithm known as Multi-Cue Background Subtraction has been utilized [14]. In this algorithm, a diverse set of cues - pixel texture, pixel color and region appearance is employed to provide a robust performance in complicated environments. The output of the background subtraction produces a binary frame with moving vehicles in the form of white blobs.

B. Centroid Identification

Centroid identification is performed using connected component analysis [15]. Due to errors in background subtraction certain false blobs of smaller size get scattered around the frame. Also, sudden camera tremors lead to larger blobs being formed. These false blobs are discounted by adapting the parameters - minimum and maximum blob area for detection. Since the camera angle can vary, a novel training process has been incorporated in the proposed solution that generates the variation of minimum blob size over x and y coordinates.

A critical issue with camera-based object tracking is blob-splitting due to the presence of poles or cables in the camera frame of view. This leads to the initialization of false tracking objects. To resolve this issue, a parameter called minimum distance between blobs is calculated for the applicable regions of the camera feed and if blobs are separated by a distance less than this parameter, they are automatically considered as belonging to a single object. While this could lead to occlusion cases occasionally the subsequent subsystem is capable of handling such situations.

IV. Multiple Object Tracking

The centroids obtained from the preceding stage are converted from camera view to top view using inverse perspective transform which is discussed in the later section. But these centroids vary in their order from frame to frame which leads to difficulty in assigning the centroids to the objects. The combination of Kalman Filter and Hungarian Assignment Solver is implemented to track multiple vehicles.

A. Kalman Filter

Kalman filter is an optimal estimator i.e. it finds the best estimate from noisy data filtering out the noise. It is used to achieve three main objectives - tracking multiple objects, identity resolution and to manage occlusion scenarios.

The filter algorithm involves two stages:

1) Prediction Stage: The state of a system x_t at a time t is assumed to evolve from the state x_{t-1} using state transition matrix F_t and w_t matrix according to the equation (1).

$$x_t = F_t \; x_{t-1} + w_t \tag{1}$$

The state model devised assumes the motion of the object to be a non-accelerating one with 4 state variables s_x^t, s_y^t, v_x, v_y modeled as in (2).

$$x_t = \begin{pmatrix} s_x^t \\ s_y^t \\ v_x \\ v_y \end{pmatrix}, F_t = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

Since the perceived vehicle acceleration acc is low in the video frames it is modelled as process noise captured in w_t with acc = 0.1 and the covariance matrix given in (3)

$$w_t = \begin{pmatrix} \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} \\ \Delta t \\ \Delta t \end{pmatrix} acc, E_z = \begin{pmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{pmatrix} acc^2 \tag{3}$$

2) Measurement Update: The centroids obtained are assumed to be noisy data with zero-mean Gaussian noise having covariance matrix R. Since only position measurements are obtained, measurement update is applied only on s_x^t and s_y^t. The detailed explanation of measurement update can be found in [16]. The noise covariance values in noise matrices are assumed to be 0.05.

B. Hungarian Assignment Solver

The Hungarian Assignment Solver is used to find the minimal matching of a given set of data that is used to map centroids to objects. The Assignment Solver is applied on the cost matrix. The matrix is formed by taking number of centroids detected in the frame for the number of columns and the tracker objects as the number of rows. The matrix is populated with the value that is the difference between the Kalman centroid prediction and the actual centroid that is obtained. The assignment is obtained by applying the Hungarian Solver on the cost matrix. The technique used is explained in depth in [17]. Once the assignment matrix is solved, it is used as an index to assign the centroids to the tracker object and thus help in object identification and tracking. However, when there is an occlusion the two cells combine and then have a single centroid for both of them. In such a situation, the centroids predicted using Kaman Filter are used in the place of actual centroid and used for object identification and tracking.

V. Speed Estimation

In most of the video based speed estimations the number of pixels travelled by the objects per unit time is estimated and scaled by a certain factor to obtain speed in $kmph$ or mph. But the centroids that are in the camera view have to be converted to the top view for accurate speed estimations. This is done using the concept of Homography.

A. Homography

The projective mapping of an image from one plane to another plane in computer vision is called Planar homography. A detailed explanation of Homographic transformation is given in [18]. An example for planar homography is the mapping of a two-dimensional planar points to the camera imager. In the proposed work four points X_{src} which were chosen in the background subtraction section for ROI are used to estimate the Homography matrix H for mapping from camera view to top view given by (4). The top view is an image of resolution 256×256 and X_{dst} are the 4 corner points of the image.

$$\begin{pmatrix} x_{dst} \\ y_{dst} \\ z_{dst} \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} x_{src} \\ y_{src} \\ 1 \end{pmatrix} \tag{4}$$

The matrix H is computed assuming the coordinates are homogenous. But the coordinates of interest is given by (5)

$$\begin{pmatrix} x'_{dst}, & y'_{dst} \end{pmatrix} = \begin{pmatrix} \frac{x_{dst}}{z_{dst}}, & \frac{y_{dst}}{z_{dst}} \end{pmatrix} \tag{5}$$

(a) (b)

Fig. 3. (a) Oblique Road Image (b) Road image after Inverse Perspective Transform

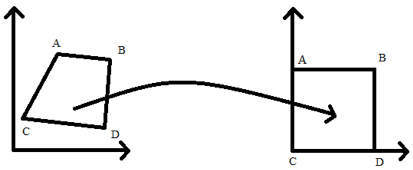


Fig. 4. Action of Inverse Perspective Transform

Details regarding computation of H from (4) can be found in [19].

B. Inverse Perspective Transform

The points on a source image in camera view plane are transformed to the points in the top view plane with the use of homography matrix H and (5). This transformation is depicted in Fig 3. The principle action of the inverse perspective transform is pictorially captured in Fig 4. H can be computed without knowledge of the camera parameters.

C. Perspective Transform

An 8×8 grid generated, shown in Fig 5a is used in order to show that homography works successfully. To display it on the road lane in camera view the grid has to be transformed to fit on the road exactly. Hence in effect an opposite transform has to be used compared to one explained in previous section. This is called perspective transform. Given the 4 points on each road the transformation can be carried out on any number of road lanes. As an example a 2 lane road system is chosen to illustrate homography. Fig 5b shows the grid effectively transformed to be overlaid on the road.

(a) (b)

Fig. 5. (a) Grid (b) Grid Overlaid on Road Lanes

D. Distance and Speed estimation by pixel proportions

This simple method of distance and width estimation takes advantage of the standard white road lines as reference to calculate the length and width scale factors in $pixels/m$. Only $1/3^{rd}$ of the ROI is chosen of estimation since nearer lines yield more accurate results. These scale factors can then be used to calculate the road length and width chosen in ROI.

The major advantage of this method is the flexibility it introduces in determining the region of interest. But a small error in measurement of the marker dimensions can lead to significant error in the overall road dimensions that can decrease the accuracy of the speed estimation module. Hence it is assumed that the dimension estimations are carried under good lighting conditions. The above method was applied to NICE road in Bangalore, India, shown in Fig 3 having actual dimensions of $48.6m \times 8m$. The standard road marker dimensions was $3m \times 0.15m$. The road dimensions were estimated to be $48m \times 7.68m$ with an error of 1.23% and 4% respectively which are tolerable.

The distance between the end points of the road estimated using the above method is used here with the grid width and length. Since the mapping is done on a 256×256 image, the GRID_WIDTH and GRID_LEN are equal to 256. Then the speed in kmph uses v_x and v_y which are provided by Kalman filter in $pixels/m$ and is estimated by formula given by (6).

$$Speed = \frac{\sqrt{(Road\_length * V_y)^2 + (Road\_width * V_x)^2} * FPS * 3.6}{GRID\_WIDTH} \tag{6}$$

VI. Results

A series of test videos were taken to validate instantaneous speeds at a resolution of 768×576. Fig 6 captures a few of the diverse camera setups on which the system was tested. For any change in the camera view due to pan, zoom or tilt, the user should only update the 4 corner points and the system will adapt its parameters accordingly.

Fig 7 shows the representation of multiple tracking with speeds on one such frame having a grid overlaid. The ground truth speeds were obtained using GPS speedometer (resolution

Fig. 6. Various Camera Setups

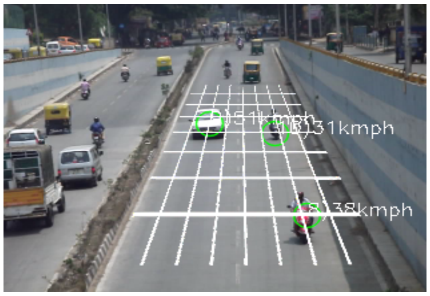


Fig. 7. Multiple Tracking

$1kmph$) with an accuracy of $0.5kmph$. The road was traversed at fairly constant speeds in increments of $5kmph$ starting from $20kmph$. The corresponding speed estimates with centroid speed initialization and resulting error are tabulated in Table I. The errors are not monotonic due to the accelerating and noise model of the Kalman Filter used. Here after 2 frames of tracking we initialize Kalman filter with centroid speeds estimated in the first 2 frames. The main advantage is that Kalman filter converges to the actual speed very quickly and accurate results are obtained leading to a max error of $\pm 3kmph$. Also, very high speeds of around $150kmph$ can be tracked accurately. Fig 8 shows the absolute percentage error in speed estimations with or without centroid speed initializations. Speeds up to $60kmph$ were from real scenarios but after $60kmph$ test videos were generated taking into consideration the real case parameters of the road dimensions and scale factors. It can be observed that without initialization the error increases with the increase in the speed. One major problem in multiple tracking is when two objects are close enough to be detected as a single blob as shown in Fig 9a. This scenario is tackled by using only Kalman predicted path without carrying measurement update. The result of applying this strategy is shown in Fig 9b demonstrating occlusion avoidance.

A special acceleration case from $30Kmph$ to $50Kmph$ in 2 seconds was validated as shown in Fig 11. The graph in Fig 10 compares the actual and estimated speeds. Due to speed initialization and the proposed acceleration model the error is within limits.

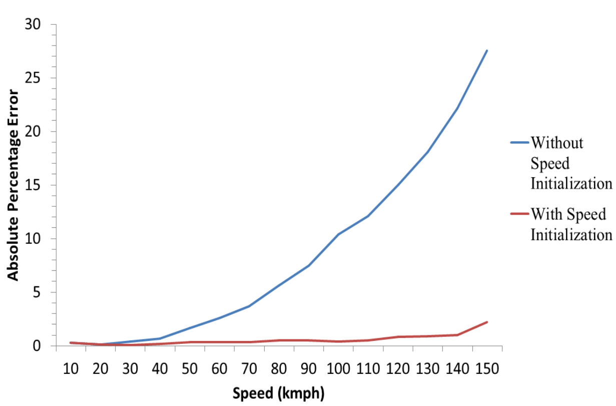


Fig. 8. Speed Error graph

(a) (b)

Fig. 9. (a) Occlusion resulting in single entity (b) Avoidance

TABLE I
SPEED ESTIMATION RESULTS

Sl No	Ground Truth (kmph)	Estimated Speed (kmph)	Error (kmph)	Error (%)
1	20	20.52	0.52	2.60
2	25	24.34	-0.66	-2.64
3	30	28.54	1.46	-4.86
4	35	33.61	-1.39	-3.97
5	40	39.05	-0.95	-2.38
6	45	46.36	1.36	3.02
7	50	51.94	1.94	3.88
8	55	53.49	-1.51	-2.75
9	60	58.27	-1.73	-2.88

The instantaneous and the average speed error range obtained in [8–13] are more compared to the proposed system. Hence this paper is an improvement to the previous approaches. The main source of error in the current approach is the estimations of dimensions of the road. This is avoided by assuming they are estimated in good lighting conditions and only $1/3^{rd}$ of ROI is chosen for good contrast of road lines. The algorithms were executed on a Lenovo laptop utilizing a video capture card to obtain live feed from a $Sony\,DCR-SX45E$ at $30FPS$ and a resolution of 768×576. Table II shows the average processing time in ms for the module which tracked on an average of 10 vehicles per frame. The total time taken is well within the budget of $33ms$ for real-time processing.

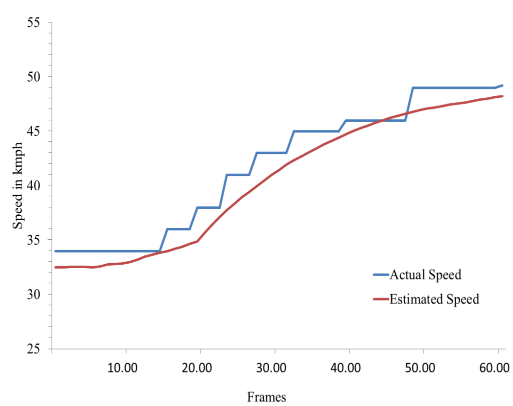


Fig. 10. Speed Estimation graph

(a) (b)

Fig. 11. Speed Estimation

TABLE II
MODULE-WISE PROCESSING TIME

Module	Processing Time in ms
Background Subtraction	21.246
Centroid Identification	3.012
Multiple-Vehicle Tracking	0.023
Speed Estimatio	0.003
Total	24.284

VII. CONCLUSION

Previous techniques for Speed estimations mostly were limited to only certain angles of the cameras to obtain good features for tracking, required manual tuning of speed factors due to the changing camera parameters and not adapting to the slight camera tremors. The USP of the proposed system is its capability to estimate vehicular speeds independent of camera intrinsics, setup parameters and its dynamic adaptation to pan, tilt and zoom. A simple distance estimation technique was developed which can synchronize easily with the changing camera parameters. A Kalman filter model adept of handling accelerating scenarios and a blend with Hungarian assignment solver was able to track multiple vehicles in occluded cases effectively. The algorithms were successfully implemented in real-time at 10 FPS on Beagle Board-xM and 30 FPS on a Lenovo laptop. The results obtained were satisfactory and maximum errors were within the tolerable range of $\pm$ 3kmph.

In future, as an enhancement to the proposed system, we plan to integrate a high resolution camera for license plate imaging with the traffic IP cameras for automatic traffic violation detection and enforcement. Additionally we would like to develop a smart signal module that is capable of vehicle-actuated timing which uses the data provided by the cameras to adapt signal timings.

REFERENCES

[1] D. Nan, T. Guozhen, M. Honglian, L. Mingwen, and S. Yao, "Low-power vehicle speed estimation algorithm based on wsn," in *Proc. 11th IEEE International Conference on Intelligent Transportation Systems*, Beijing, 12-15 Oct 2008, pp. 1015–1020.

[2] N. Kassem, A. E. Kosba, and M. Youssef, "Rf-based vehicle detection and speed estimation," in *Proc. 75th IEEE Conference on Vehicular Technology Conference*, Yokohama, 6-9 May 2012, pp. 1–5.

[3] C. Pornpanomchai and K. Kongkittisan, "Vehicle speed detection system," in *Proc. 2009 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Kuala Lampur, 18-19 Nov 2009, pp. 135–139.

[4] M. V. K and H. M, "Estimation vehicle speed by motion tracking on image sequences," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, Istanbul, 21-24 Jun 2010, pp. 185–190.

[5] S. Doğan, M. S. Temiz, and S. Külür, "Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera," *Sensors*, vol. 10, no. 5, pp. 4805–4824, 2010.

[6] F. D. J and W. Y, *Optical flow estimation*, ser. Handbook of Mathematical Models in Computer Vision, 2006, vol. 2, ch. 15, pp. 239–258.

[7] D. J. Dailey, F. W. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 98–107, Jun 2000.

[8] S. T. N and D. D. J, "Algorithms for calibrating roadside traffic cameras and estimating mean vehicle speed," in *Proc. IEEE International Conference on Intelligent Transportation Systems*, Seattle, 30 Sept 2007, pp. 277–283.

[9] H. M. *et al*, "Viewpoint independent vehicle speed estimation from uncalibrated traffic surveillance cameras," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, 11-14 Oct 2009, pp. 4920–4925.

[10] T. T. N. *et al*, "Compensating background for noise due to camera vibration in uncalibrated-camera-based vehicle speed measurement system," *IEEE Trans. Vech. Tech.*, vol. 60, no. 1, Jan 2011.

[11] G. Y. *et al*, "Proposal a vehicle speed measuring system using image processing," in *Proc. International Symposium on Computer, Consumer and Control (IS3C)*, Taichung, 10-12 Jun 2014, pp. 541–543.

[12] B. E. A. *et al*, "Monocular vision-based vehicular speed estimation from compressed video streams," in *Proc. IEEE The 16th International Conference on Intelligent Transportation Systems*, The Hague, 2013, pp. 1155–1160.

[13] D. C. Luvizon, B. T. Nassu, and R. Minetto, "Vehicle speed estimation by license plate detection and tracking," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, 4-9 May 2014, pp. 6563–6567.

[14] S. J. Noh and M. Jeon, "A new framework for background subtraction using multiple cues," in *Proc. the 11th Asian Conference on Computer Vision, Verlag*, Daejeon, 2013, pp. 493–506.

[15] S. S and A. K, "Topological structural analysis of digitized binary images by border following," in *CVGIP 30 1*, 1985, pp. 32–46.

[16] F. R, "Understanding the basis of the kalman filter via a simple and intuitive derivation," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, 2012.

[17] F. Lütteke, X. Zhang, and J. Franke, "Implementation of the hungarian method for object tracking on a camera monitored transportation system," in *Proc 7th German Conference on Robotics*, Munich, 21-22 May 2012, pp. 1–6.

[18] Technobit, "The homography transformation," http://www.corrmap.com/features/homography_transformation.php, 1 Jan 2013.

[19] "Computing the plane to plane homography," http://www.robots.ox.ac.uk/~vgg/presentations/bmvc97/criminispaper/node3.html, 7 Nov 2014.