Computers and Electrical Engineering 76 (2019) 238–248

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Vehicle speed measurement model for video-based systems[☆]

Saleh Javadi[a,*], Mattias Dahl[b], Mats I. Pettersson[b]

[a] *Department of Mathematics and Natural Sciences, Blekinge Institute of Technology (BTH), Karlshamn 37435, Sweden*
[b] *Department of Mathematics and Natural Sciences, Blekinge Institute of Technology (BTH), Karlskrona 37179, Sweden*

ARTICLE INFO

Article history:
Received 18 July 2018
Revised 4 April 2019
Accepted 4 April 2019
Available online 10 April 2019

Keywords:
Speed measurement system
Motion analysis
Machine vision
Pattern recognition
Intelligent transportation systems

ABSTRACT

Advanced analysis of road traffic data is an essential component of today's intelligent transportation systems. This paper presents a video-based vehicle speed measurement system based on a proposed mathematical model using a movement pattern vector as an input variable. The system uses the intrusion line technique to measure the movement pattern vector with low computational complexity. Further, the mathematical model introduced to generate the pdf (probability density function) of a vehicle's speed that improves the speed estimate. As a result, the presented model provides a reliable framework with which to optically measure the speeds of passing vehicles with high accuracy. As a proof of concept, the proposed method was tested on a busy highway under realistic circumstances. The results were validated by a GPS (Global Positioning System)-equipped car and the traffic regulations at the measurement site. The experimental results are promising, with an average error of 1.77 % in challenging scenarios.

1. Introduction

Vehicle speed measurement is one of the key components of traffic surveillance systems. The provided data can be used for traffic management and law enforcement. Speed measurement systems can be divided into two main categories based on their methodology being either active or passive. An active system measures how transmitted signals are affected by a passing vehicle in order to estimate the vehicles speed, e.g., a sensor system based on an induction-coil loop, a radar or a laser. However, some of these systems are expensive and require extensive calibration or maintenance [1]. Passive and active sensors often complement each other well, and, in this study, we concentrate on passive optical video systems. Video-based systems have become more popular due to the technological advancements in cameras and computer devices, which have become more efficient, available and reliable [2,3]. In addition, the reasonable cost-effectiveness of these systems makes them highly competitive with conventional methods [4,5].

Video-based speed measurement systems analyse consecutive video frames in order to track a vehicle and thereby measure its speed. Various image processing and pattern recognition approaches have been employed to realise this task [6]. The main assumptions are that the speed of the vehicle during detection is constant and that the camera is facing down

---

[☆] This paper is for regular issues of CAEE. Reviews processed and recommended for publication to the Editor-in-Chief by Area Editor Dr. E. Cabal-Yepez.

[*] Corresponding author.

E-mail addresses: saleh.javadi@bth.se (S. Javadi), mattias.dahl@bth.se (M. Dahl), mats.pettersson@bth.se (M.I. Pettersson).

https://doi.org/10.1016/j.compeleceng.2019.04.001

towards the scene. A commonly employed technique is foreground/background segmentation, which extracts the moving vehicle as a blob in order to estimate its speed in consecutive frames [4,7–12]. The segmentation is mainly pixel-based, employing Gaussian [13,14], GMM (Gaussian Mixture Model) and kernel densities or a region-based method,such as Statistical SCSM (Circular Shift Moments), which considers spatial dependencies to determine the foreground in the video frame. The direct blob analysing methods using the blobs area or displacement are sensitive to shadow and sudden illumination and require a sparse area for segmentation in several consecutive frames [6].

Many speed measurement methods have been proposed to track the features of a moving vehicle directly such as edges and corners [1,9,15,16]. The main challenges here are to find and track discriminative features representing the vehicle and to achieve more accurate results by detecting the features on the same road plane. Then the displacement in pixels is turned into the correct real- world geometric displacement in order to calculate the speed. Optical flow algorithms have been used in many applications for motion estimation of a subset of points in a scene. The LK (Lucas–Kanade) algorithm, Horn–Schunck algorithm and Farnebäck polynomial expansion algorithm are among the most common algorithms employed to yield the optical flow [17]. Feature-based methods are more robust and effective when it comes to shadows, i.e., tracking vehicles in challenging lighting conditions. It should be noted that feature extraction for geometric measurements, as opposed to recognition, usually requires images with high resolution [18].

Furthermore, there are approaches that have been introduced to track objects based on kernels and local functions, such as density distributions and image patches [19]. Mean-Shift and Camshift (Continuously Adaptive Mean-Shift) algorithms [20] are among the popular techniques used for tracking objects whose sizes may change in consecutive frames. The process of convergence to a local maximum of the distribution under a window containing the moving object continues in each successive frame in order to track the object in the video [21]. These methods are suitable for tracking vehicles that are relatively far from the camera with high contrast; otherwise, it may be difficult to obtain a geometrically correct displacement [12].

There are also methods which combine several of the above techniques in order to estimate the speed of the vehicle along a road, e.g., by tracking the license plate region [6,22]. These methods require at least two images containing the moving vehicle. The license plate region is then detected based on various approaches such as adaptive thresholds, a priori boundaries for size and shape and so forth. Finally, the speed is estimated based on a motion vector obtained by tracking the license plate region or its distinctive features in at least two frames. When using a calibrated camera, a projective transformation (inverse perspective mapping) can be used to achieve more accurate results. These methods require specific camera placement and images with high resolution in order to be able to detect the license plate region properly. In addition, the license plates for all vehicles are assumed to be at the same height from the ground [6].

It can be seen from the review of related works above that as a result of the perspective distortion, the pixel displacement in the image varies from the-real word geometric displacement. Consequently, for feature-based tracking methods, the inverse perspective mapping should be applied to the plane containing the features (and not necessarily to the ground). Therefore, without a 3D model of the whole system, there could be an inevitable source of error due to the perspective distortion [4,6]. In addition, any model must consider the temporal sampling of the camera, the placement of the intrusion lines, the vehicle's speed and other uncertainties dealing with these measurements.

This paper presents a speed estimation method that utilizes the camera frame rate, the placement of the intrusion lines and a movement pattern vector in order to provide a pdf (probability density function) model of the vehicle's speed. This model does not rely on the camera's parameters, and it requires only a few reference points and their relative positioning to obtain a precise relation to the real-world geometric displacement. A proof of concept of this method was evaluated on several hours of video of a major highway recorded by an off-the-shelf handheld camera and a smartphone using different frame rates under various illuminations and traffic conditions. The ground truth dataset containing the vehicle speeds was obtained using a GPS (Global Positioning System)-equipped car. Furthermore, the results are discussed and evaluated based on the traffic circumstances and regulations of the respective highway for different categories of vehicles.

This work's main motivation is to propose a vision-based speed measurement system that can cover a large area or a multi-lane road with minimal installation and maintenance costs in comparison with active sensors, such as radar. As another motivation, the proposed model is able to measure the speed of a vehicle precisely by utilizing only a few parameters and with low computational complexity.

The main contributions of this paper are listed below:

- Addressing the parameters affecting peed measurements in video-based systems analytically, and
- Developing a comprehensive mathematical model to obtain the probability density function of the vehicles speed in the video.

The rest of the paper is organised as follows. The proposed model and the implementation methodology of the speed measurement system are described in Section 2. In Section 3, experimental results are provided and discussed. Finally, the paper is concluded in Section 4.

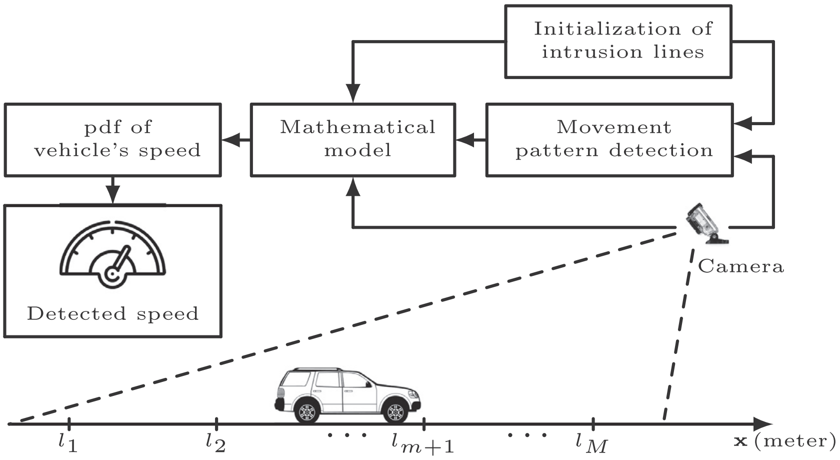

Fig. 1. A detection scene containing several intrusion lines (l_{m+1}, where $m \in \{0, \dots, M-1\}$) and a passing vehicle observed by a camera facing the road.

2. Proposed methodology

In this section, a model is introduced in order to consider the main parameters contributing to uncertainties of a speed measurement system. The detection system is based on a scenario with several intrusion lines, knowledge of their relative positioning in the real world and a passing vehicle (see Fig. 1). The vehicle's speed is assumed to be constant when crossing the lines since the length of the measurement scene is very short (few meters). The proposed model then yields a pdf of the vehicle's speed based on a detected movement pattern vector. The implementation of such a system is described later in this section. Using the intrusion lines as well as storing the incidents in a pattern vector is incredibly effective regarding both memory and computational complexity.

2.1. Speed estimation model

Intrusion detection is a method used to determine when an object crosses a virtual line and enters a region of interest. For video-based observations, the object's position is acquired at discrete locations due to the temporal sampling rate. The sampling interval between the detected discrete locations is equal to the camera's sampling time. Thus, the intrusion cannot be detected exactly at the line but instead within a detection distance Δ (m). The detection distance is related directly to the camera's sampling time T (s) and the hypothetical speed of the vehicle v (m/s),

$$\Delta = Tv. \tag{1}$$

In other words, the detection distance is directly proportional to the vehicle's speed and inversely proportional to the camera's frame rate. The initial position (starting point) x_1 (m) where the vehicle appears after the first intrusion line is assumed to be random. In the case of several intrusion lines, for successful detection, the vehicle should fall within the detection distances associated with each consecutive intrusion line.

In this paper, a model is proposed for calculating a pdf of a passing vehicle's speed based on a movement pattern vector. Hence, M intrusion lines ($M \geq 2$) should be placed on the receiving video frames of a stationary camera such that the relative distances between them in the real world along the road are known. Accordingly, the intruding vehicle is detected as soon as it crosses each line at a specific frame index. The main inputs of the model are the movement pattern vector $\boldsymbol{n} = [n_0, \dots, n_{M-1}] \in \mathbb{N}$ and the distance vector of the intrusion lines $\boldsymbol{d} = [d_0, \dots, d_{M-1}] \in \mathbb{R}_0^+$, which are defined as,

$$n_m = f_{m+1} - f_1, \tag{2}$$

$$d_m = l_{m+1} - l_1, \tag{3}$$

where $m \in \{0, \dots, M-1\}$, M is the number of intrusion lines, f_{m+1} is the video frame index number at which detection occurs, and l_{m+1} (m) is the position of the intrusion line in the real- world coordinate system (see Fig. 2). As mentioned earlier, given the relative placements of the intrusion lines in the distance vector $\boldsymbol{d}$ and the camera frame rate $1/T$, a vehicle can appear at any point after the first intrusion line within Δ. Thus, the starting points distance δ from the first intrusion line is unknown (random) and not greater than the possible maximum detection distance Δ, where $0 \leq \delta \leq \Delta$. However, the vehicle's position x_m within the following detection distance can be computed knowing the starting point x_1, the frame rate, the distance vector of the intrusion lines and the vehicle's speed.

At this stage, we introduce two sets of possible positions that demonstrate the places where a vehicle with the hypothetical speed v observed by a camera with the frame rate $1/T$ can appear along the path travelled:

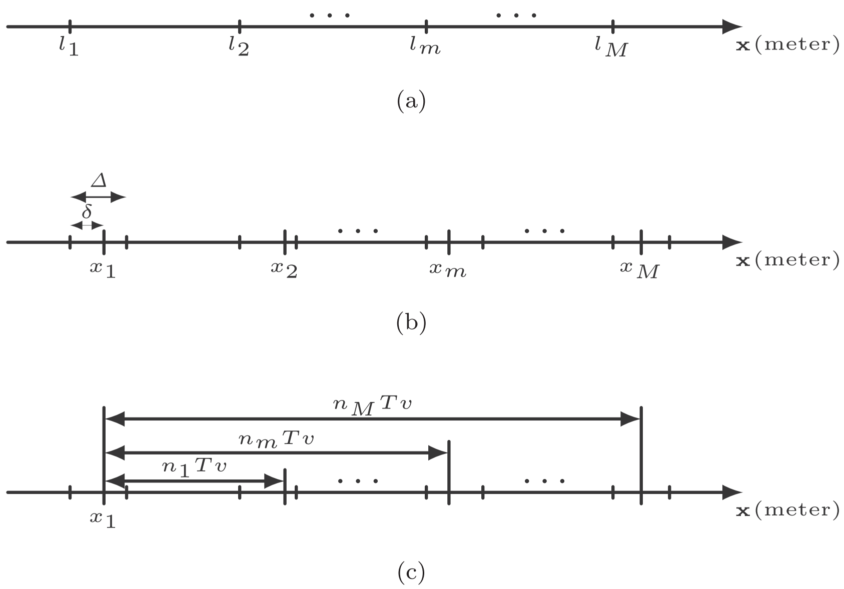

Fig. 2. (a) The intrusion lines l_m, (b) The hypothetical vehicle's positions x_m within the maximum detection distance Δ for a hypothetical speed v, (c) One illustration of vehicle's positions and the associated n_ms for a hypothetical speed v.

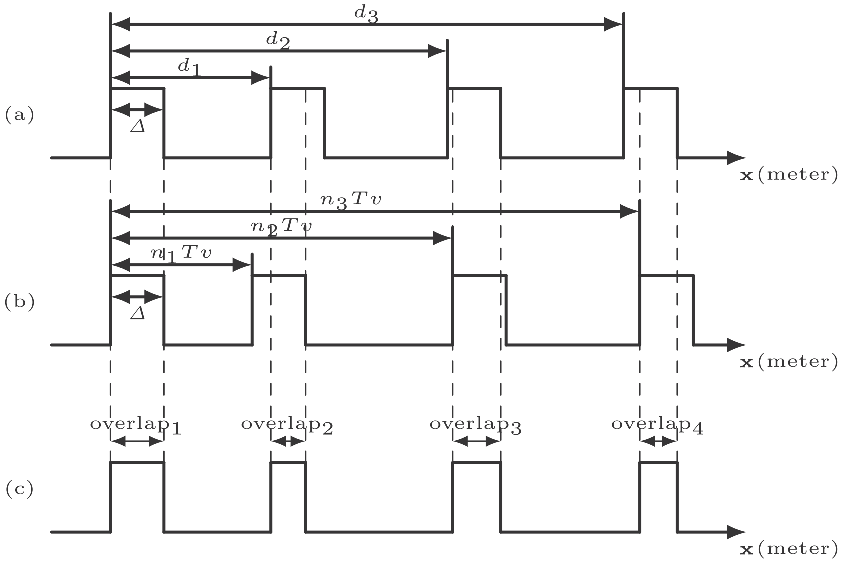

Fig. 3. One example of possible positions based on: (a) the distance vector of the intrusion lines $\boldsymbol{d}$, (b) the movement pattern vector $\boldsymbol{n}$, and (c) the overlap intervals for (a) and (b) for a hypothetical speed v.

- First, a set of possible positions is computed based on the distance vector of the intrusion lines $\boldsymbol{d}$. These positions indicate the detection distances within which a vehicle traveling at a hypothetical speed should appear in order to satisfy the real-world distance vector of the intrusion lines (see Fig. 3 (a)).
- Second, a set of possible positions is calculated based on the detected movement pattern vector $\boldsymbol{n}$. These positions indicate the detection distances within which a vehicle traveling at a hypothetical speed should appear in order to satisfy the movement pattern vector (see Fig. 3 (b)).

The necessary (but not sufficient) condition for a valid detection is that the two sets of possible positions above overlap for all the detection distances for each intrusion line. For example, the method is outlined in a simple scenario with four intrusion lines in Fig. 3. The first row shows the set of possible positions in accordance with the distance vector of the intrusion lines $\boldsymbol{d}$, followed by the detection distances for a hypothetical speed v. In the second row, the set of possible positions is provided based on the movement pattern vector $\boldsymbol{n}$ and the same hypothetical speed v. The δ is the same for both sets since their initial detection conditions are identical. Thus, the first distance overlap of the two sets of possible

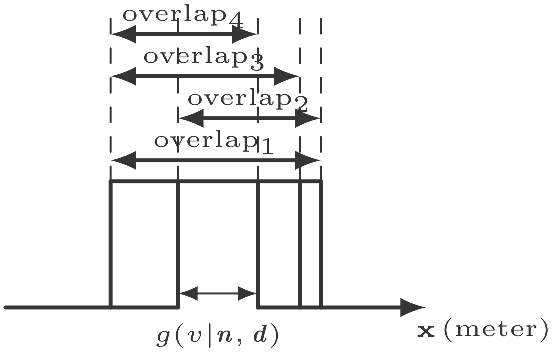

Fig. 4. The common region g of all the distance overlaps based on the scenario in Fig. 3 associated with the probability of the hypothetical speed v.

positions is a complete Δ. The remaining distance overlaps are the intersections of the possible position intervals for (a) and (b) (see Fig. 3 (c)).

In addition, there should be a common region between all the distance overlaps themselves (see Fig. 4) since the set of possible positions are generated by repetition according to the movement pattern vector, the camera frame rate and the vehicle's hypothetical constant speed. In other words, the detection distances at intrusion lines are not, in fact, continuous but rather discrete positions at which the vehicle can appear after each line provided the movement pattern vector $\boldsymbol{n}$ is satisfied. Therefore, the distance overlaps are valid if the intervals between them do not violate the movement pattern vector.

The common region of all distance overlaps is found via

$$g(v|\boldsymbol{n}, \boldsymbol{d}) = \max\Big(0, \min_m(d_m + (1 - n_m)Tv) \\ - \max_m(d_m - n_m Tv)\Big), \tag{4}$$

where $m \in \{0, 1, \ldots, M-1\}$, and $g(v|\boldsymbol{n}, \boldsymbol{d})$ is the common region of the distance overlaps. The pdf is obtained with

$$h_V(v|\boldsymbol{n}, \boldsymbol{d}) = \frac{g(v|\boldsymbol{n}, \boldsymbol{d})}{\int_{v_{\text{lower}}}^{v_{\text{upper}}} g(v|\boldsymbol{n}, \boldsymbol{d})dv}, \tag{5}$$

where v_{lower} and v_{upper} are the lower and upper bounds, respectively, of the hypothetical speeds with $g(v|\boldsymbol{n}, \boldsymbol{d})$ greater than zero, and $h_V(v|\boldsymbol{n}, \boldsymbol{d})$ is the pdf for the stochastic speed V. Accordingly, the expected speed can be determined as

$$\mu_V = E[V] = \int_{v_{\text{lower}}}^{v_{\text{upper}}} v h_V(v|\boldsymbol{n}, \boldsymbol{d})dv, \tag{6}$$

where $E[V]$ is the expected speed of the vehicle given the movement pattern vector $\boldsymbol{n}$ and the intrusion lines distance vector $\boldsymbol{d}$.

To sum up, the mathematical model that has been introduced is able to provide the pdf of the vehicles speed based on the input parameters. The input parameters are the camera sampling time T, the intrusion lines distance vector $\boldsymbol{d}$ and the movement pattern vector $\boldsymbol{n}$. The learning phase for the system is based mainly on T and $\boldsymbol{d}$, which are computed only once in the beginning. The remaining parameter $\boldsymbol{n}$ is computed upon the passage of each vehicle in real-time; therefore, the model computes the speed of the vehicle instantly.

2.2. Implementation of the speed estimation system

As described earlier, the movement pattern vector $\boldsymbol{n}$ is a set of video frame index differences, i.e., the number of frames between detections at successive intrusion lines. Therefore, a detection system is required for each intrusion line in order to determine the vehicles entrance into that specific detection distance. First, the distinctive features of the road corresponding to the measurement area on the receiving video frames are selected and the corresponding keypoints extracted. Then, the extracted keypoints are tracked in consecutive frames, so that any motion or displacement in the region is detected.

2.2.1. Motion detector

The ideal output of a motion detector algorithm would be some speed or displacement estimation vectors that indicate the movement of any pixel in a pair of frames [18].

In this paper, we use a motion estimation approach based on the polynomial expansion algorithm [17]. The polynomial expansion algorithm computes an optical flow using an analytical model that approximates the image as a function for which, at every pixel, a polynomial is locally fitted. Using this method, when the image is considered as a continuous function, a small displacement of a portion of the image can be determined by equating the coefficients of the polynomial expansions at the same point. However, other methods can be used as well.

This algorithm provides a motion detection component with which to identify the exact frame during which a vehicle crosses an intrusion line. Thus, the computed displacement is used only to determine the significance of the movement by using a desired distance threshold to detect the intrusion of the vehicle while ignoring noise and other unwanted sources.

In the proposed system, the region of interest for detection after each intrusion line is employed for a maximum possible speed ($v_{\max}$) around 40 m/s. Therefore, the system is able to detect any vehicle travelling below that speed, i.e., the maximum detection distance Δ (1) is within the region of interest for any hypothetical speed less than 40 m/s. The pdf of the vehicle's speed is then determined based on the detected movement pattern vector $\boldsymbol{n}$.

2.2.2. Practical considerations

In this section, some practical considerations are addressed in order to achieve a robust measurement in a noisy environment. The first practical consideration is that the detection distances associated with their respective intrusion lines are not activated all at the same time but only one at a time. Therefore, the system is initially expecting the intrusion to be detected at the first detection distance based on the direction of the movement. Once motion is estimated at the current detection distance, then it would be deactivated and the next distance activated. Such a system will ultimately provide more robust measurements since its vulnerability is limited to only one intrusion line and its detection distance at a time.

The second practical consideration is avoiding noisy measurements by considering a threshold for motion detection (motion filter), meaning that if the estimated motion within the detection distance is less than the minimum displacement, then it would not be considered to be an intrusion since the noise and vibration of the camera (due to wind or heavy vehicles) can cause small movements in the detection distances.

Another practical consideration involves improving measurement accuracy and robustness via the waiting time (time filter). The idea is that after each detection at an intrusion line, there is a waiting time during which the system waits for the next intrusion at the next intrusion line. If, within this waiting time, no intrusion occurs, then the initial detection is ignored and the system returns to its initial condition. The waiting time is calculated based on the prior knowledge of the minimum speed of the vehicles on the road (vehicles traveling at speeds lower than 11 m/s ($\sim$40 km/h) are not allowed on highways). Doing so helps the system to avoid false detections due to noise or targets of no interest, such as bicycles. Other types of video noise caused by external factors, such as flying birds or shadows, are also filtered out.

It should be noted that the proposed model is still valid for heavy traffic on multi-lane roads since it relies only on the movement pattern vector $\boldsymbol{n}$. Therefore, only an additional tracking component would be required to track every single vehicle passing through the intrusion lines (even if they are changing lanes) in order to obtain their individual movement pattern vectors. Then the pdfs of the passing vehicles' speeds would be generated separately according to the proposed method.

3. Experimental results and discussion

This system focuses on employing multiple intrusion lines although the evaluation of the parameters under these circumstances can be quite complicated. When there are more than two intrusion lines, the system's error e would be reduced since the number of measurements would increase. The error e is considered as the absolute difference between the actual and expected speeds for each measurement. However, intrusion lines cannot be added indefinitely. The primary restriction is that the minimum distance between two intrusion lines should be greater than $Tv_{\max}$. Further restrictions include the geometrical properties of the scene, the size of the vehicles and the camera's resolution.

In this experiment, we used approximately two hours of data collected by cameras monitoring a major highway with normal to heavy traffic. The data contains information on trucks, trailers, cars and buses passing by at various speeds. The recordings were acquired simultaneously by one handheld camera and one smartphone with frame rates of $1/T = 50$ fps and $1/T = 30$ fps, respectively, in order to investigate the proposed model. The size of the input frame was 960×540 pixels. The distance between the first and last intrusion lines was 8.97 m, which was in accordance with the cameras' resolutions and the detection scene. Four intrusion lines were implemented, and each intrusion line was aligned to a guardrail pole as a reference point, see Fig. 5. The distance vector of the intrusion lines was then defined as $\boldsymbol{d} = [0, 2.87, 5.95, 8.97]$ (m). The proposed system was realized on a common notebook platform[1] and was able to operate in real time.

For every passing vehicle, a movement pattern vector was generated that indicated the frame number differences $\boldsymbol{n}$ between the four intrusion lines. According to the model derived in Section 2. A, a speed pdf was associated with the movement pattern vector of each vehicle. The pdf provided a range of possible speeds (lower and upper bounds) and also the probability of each speed to occur given the movement pattern vector $\boldsymbol{n}$. In order to evaluate the proposed method more accurately, we drove through the four intrusion lines with a GPS-equipped car and compared the detected speeds with the ones obtained via the GPS. Several runs were made at different constant speeds, and the motion data extracted from the vehicle included the time, location and speed, see Fig. 6. The speeds extracted from the navigator device were considered to be ground truth for our measurements since the error range of a GPS used under the same atmospheric conditions is very low [23].

As explained earlier, a higher frame rate provides a higher sampling rate, generating a smaller Δ. It was also observed during the experiment that the relation between the sampling rate and the measurement error for various speeds

---

[1] Windows notebook platform using Visual C++ and OpenCV libraries.

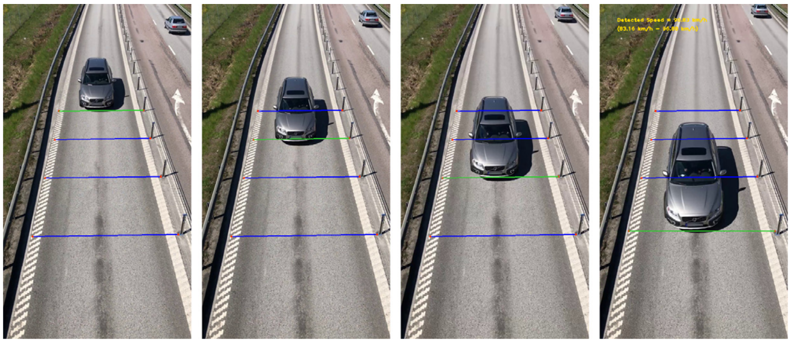

Fig. 5. Speed measurements for a vehicle travelling through four intrusion lines, including the estimation of possible lower and upper bounds.

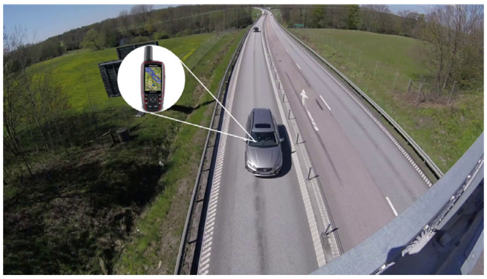

Fig. 6. An overview of the measurement site being traversed by the GPS-equipped car.

is not strictly proportional (see Tables 1 and 2). The error rate for the proposed method at 50 fps is 1.77%, while it is 2.17% at 30 fps. Thus, in general, the accuracy is improved with an increase in the frame rate. However, when comparing the two measurement setups with different frame rates, the error difference is small and not proportional. This result can be explained by the low number of measurements, and possible errors in estimating the speed via the GPS device must be considered. Table 3 presents the average errors obtained via four different methods. It can be seen that the performance of our proposed method at 50 fps is superior to the results presented in [24] and [25]. Although, Lan et al. [1] has lower average error rate, it does not perform well at low speeds due to being based on the frame difference algorithm. Furthermore, the error rate in our method can be reduced further by varying the distances between intrusion lines as well as using cameras with higher frame rates.

Another important aspect of the results obtained using the proposed method is that the actual speeds are within the range of the estimated pdf regardless of the sampling rate. This finding suggests that the model is reliable in terms of detecting a range that contains the true speed. This property is crucial when taking legal action based on the results. Further, a higher frame rate often provides a narrower range of possible speeds, see Fig. 7.

Table 1
The performance evaluation of the proposed method at a frame rate of $\frac{1}{T} =$ 30 fps with $\boldsymbol{d} = [0, 2.87, 5.95, 8.97]$ (m).

No.	Actual speed (m/s)	Movement pattern vector	Detected speed (m/s) $(v_{\text{lower}} - v_{\text{upper}})\ \mu_V$	Error rate (%)
1	20.5	$\boldsymbol{n} = [0, 5, 9, 13]$	(20.3 – 21.5) 20.8	1.46
2	26.2	$\boldsymbol{n} = [0, 4, 7, 11]$	(23.1 – 26.9) 25.2	3.82
3	20.0	$\boldsymbol{n} = [0, 4, 9, 13]$	(19.2 – 22.3) 20.6	3.00
4	25.3	$\boldsymbol{n} = [0, 4, 7, 11]$	(23.1 – 26.9) 25.2	0.39

Table 2
The performance evaluation of the proposed method at a frame rate of $\frac{1}{T} =$ 50 fps with $\boldsymbol{d} = [0, 2.87, 5.95, 8.97]$ (m).

No.	Actual speed (m/s)	Movement pattern vector	Detected speed (m/s) $(v_{\text{lower}} - v_{\text{upper}})\ \mu_V$	Error rate (%)
1	20.5	$\boldsymbol{n} = [0, 7, 14, 21]$	(20.4 – 22.4) 21.4	4.39
2	26.2	$\boldsymbol{n} = [0, 6, 12, 17]$	(25.4 – 27.0) 26.2	0.00
3	20.0	$\boldsymbol{n} = [0, 7, 15, 22]$	(19.5 – 21.2) 20.3	1.50
4	25.3	$\boldsymbol{n} = [0, 6, 12, 18]$	(23.6 – 26.4) 25.0	1.18

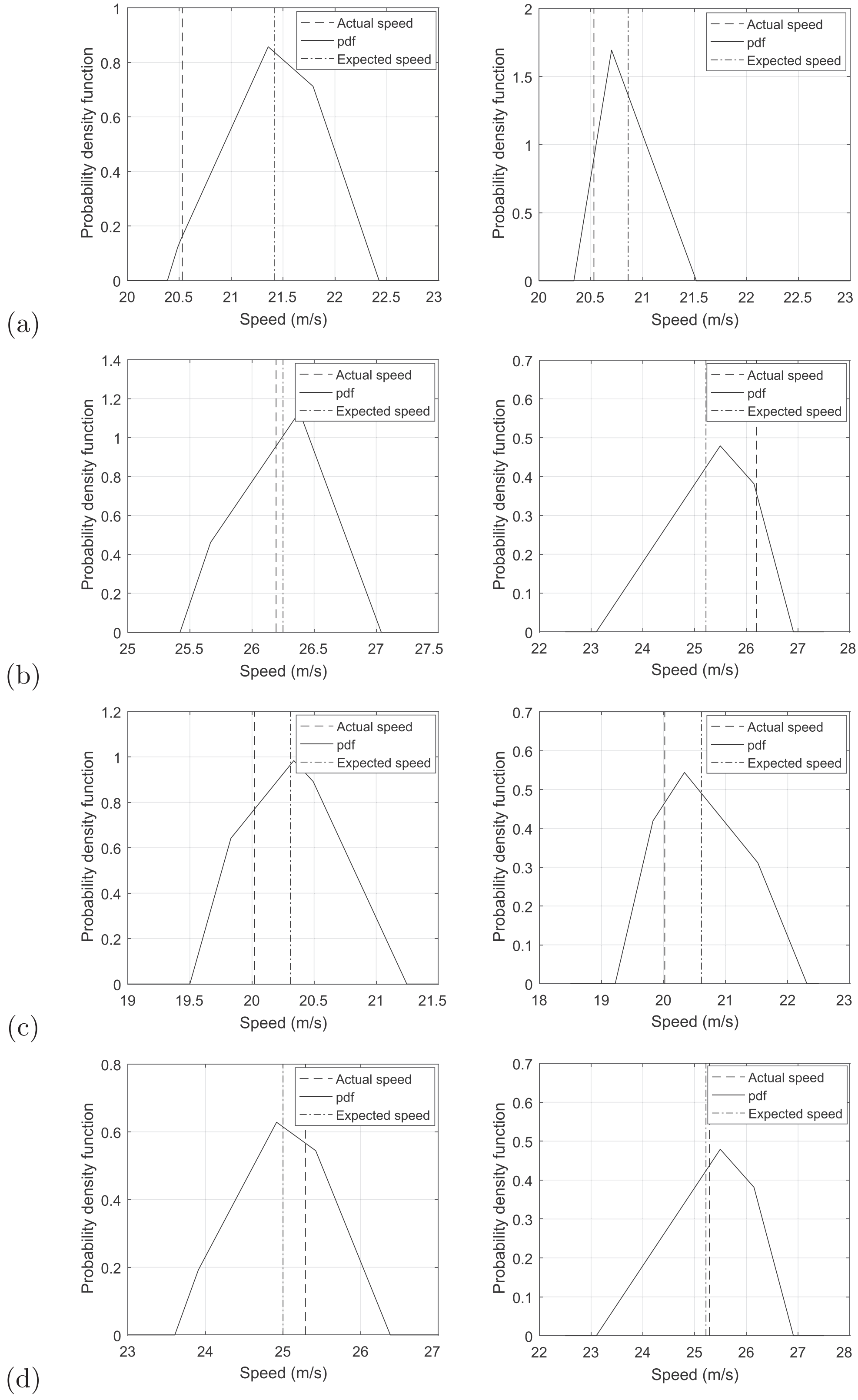

Fig. 7. The estimated pdfs of the experiments described in Table 1 at 30 fps (right column) and Table 2 at 50 fps (left column).

Table 3
The average errors for different methods.

	Method [24]	Method [25]	Method [1]	Our method	
				$\frac{1}{T} = 30$ fps	$\frac{1}{T} = 50$ fps
Average error (%)	6.1	1.9	1.12	2.17	1.77

Table 4
The speed limits and detected speeds of the passing vehicles.

Speed range (km/h)	Passenger car & van	Passenger car & van with trailer	Truck & bus	Truck with trailer
$50 \le v < 60$	2	1	0	1
$60 \le v < 70$	6	0	1	2
$70 \le v < 80$	56	7	4	38
$80 \le v < 90$	200	10	16	41
$90 \le v < 100$	106	2	0	0
$100 \le v < 110$	154	0	0	0
$110 \le v < 120$	24	0	0	0
$120 \le v < 130$	7	0	0	0
Average speed	92.9	81.4	81.7	78.9

Furthermore, we have measured the pdfs and expected values for 678 passing vehicles on three different days and compared them with the posted speed limits for the measurement site. A correlation between the detected speeds and the speed limit can also validate the proposed model considering that vehicles probably travel within the speed limits. Table 4 shows the distribution of the detected speeds. The speed limits for the measurement site were 27.8 m/s (100 km/h) for passenger cars and vans, 25 m/s (90 km/h) for heavy vehicles (e.g., trucks and buses) and 22.2 m/s (80 km/h) for any vehicles with trailer(s). The experimental results show a high correlation between the detected speeds and the expected ranges for different categories of vehicles.

As shown in Fig. 8, the greatest number of vehicles fall within the range 22.2–24.7 m/s (80–89 km/h). The heavy vehicles and vehicles with trailers are supposed to travel within this range and around 22.2 m/s (80 km/h), respectively. In addition, the passenger cars that were following these slower-moving vehicles without being able to overtake them had to travel at

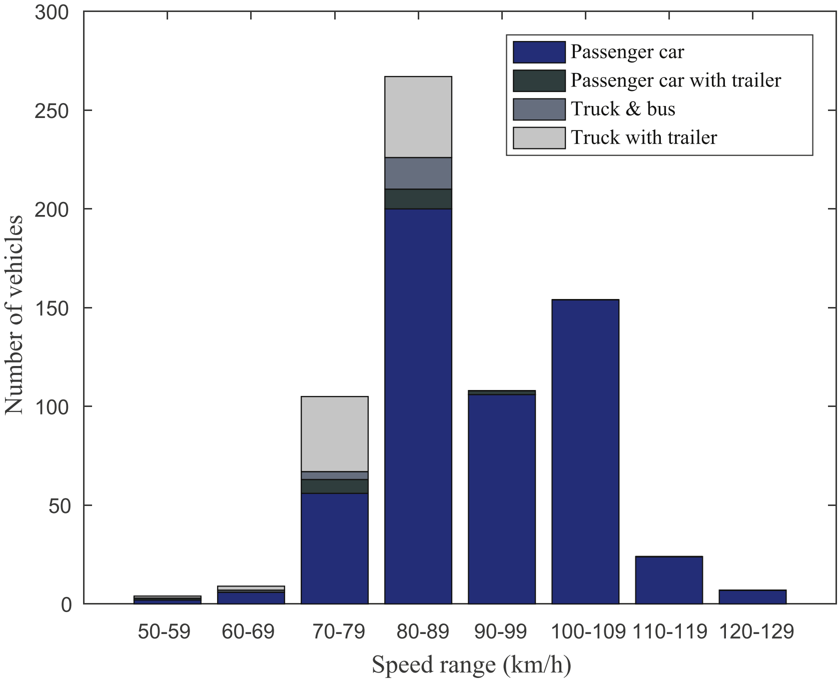

Fig. 8. The detected speeds of passing vehicles categorized according to class.

almost at the same speeds. The second highest number of vehicles were detected travelling in the range of 27.8–30.3 m/s (100–109 km/h), which is as expected since that was the speed limit for passenger cars.

More importantly, no heavy vehicle was probably travelling at a speed above 25 m/s (90 km/h) since the majority of them are controlled automatically according to the speed limit. The experimental results demonstrate that the proposed model is reliable and can achieve accurate measurements.

4. Conclusion

The presented video-based speed measurement model provides the probability density function of a passing vehicle's speed based on its movement pattern vector. In this work, the proposed method was implemented using four intrusion lines, and the input frames were captured simultaneously by an off-the-shelf handheld camera and a smartphone device with frame rates of 50 fps and 30 fps, respectively. According to the experimental results, the average error rates of the speed measurement system at 50 fps and 30 fps were 1.77% and 2.17%, respectively, for vehicles travelling in the range of 70 km/h–100 km/h. The error rate decreased noticeably with an increase in the frame rate, as expected. In addition, the actual speeds of the vehicles were within the range of the estimated pdfs, increasing the confidence in the model. It was also observed that the ranges of obtained pdfs often decreased with the increase in the camera frame rate. Furthermore, the proposed model was employed to measure the speed of more than 670 vehicles, and the results were compared with the speed limits with respect to the vehicles categories passing through the measurement site. The experimental results demonstrated that the detected speeds and the speed limits were highly correlated. Theses correlations validate the proposed model since the majority of heavy vehicles (e.g., buses, trucks and trailers) are equipped with speed regulators forcing them to follow the speed limits. It can be concluded that the proposed video-based model performs with high precision and robustness when measuring the speeds of passing vehicles. Future work should focus on utilizing a depth map in order to initiate the intrusion lines automatically.

Conflict of interest

None.

Acknowledgments

The authors would like to thank the Swedish Road Administration, the Swedish Transport Agency, Netport Science Park, and the Municipality of Karlshamn Sweden for their support in this work.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compeleceng.2019.04.001.

References

[1] Lan J, Li J, Hu G, Ran B, Wang L. Vehicle speed measurement based on gray constraint optical flow algorithm. Optik - Int J Light Electron Opt 2014;125(1):289–95. doi:10.1016/j.ijleo.2013.06.036.

[2] Gomes SL, de S Rebouças E, Neto EC, Papa JP, de Albuquerque VHC, Filho PPR, et al. Embedded real-time speed limit sign recognition using image processing and machine learning techniques. Neural Comput Appl 2017;28(1):573–84. doi:10.1007/s00521-016-2388-3.

[3] Neto EC, Gomes SL, Filho PPR, de Albuquerque VHC. Brazilian vehicle identification using a new embedded plate recognition system. Measurement 2015;70:36–46. doi:10.1016/j.measurement.2015.03.039.

[4] Buch N, Velastin SA, Orwell J. A review of computer vision techniques for the analysis of urban traffic. IEEE Trans Intell Transp Syst 2011;12(3):920–39. doi:10.1109/TITS.2011.2119372.

[5] Kaya S, Kilic N, Kocak T, Gungor C. A battery-friendly data acquisition model for vehicular speed estimation. Comput Electr Eng 2016;50:79–90. doi:10.1016/j.compeleceng.2016.01.017.

[6] Luvizon DC, Nassu BT, Minetto R. A video-based system for vehicle speed measurement in urban roadways. IEEE Trans Intell Transp Syst 2017;18(6):1393–404. doi:10.1109/TITS.2016.2606369.

[7] Banerjee S, Dhar M, Sen S. A novel technique to detect the number of ground vehicles along with respective speed of each vehicle from a given video. In: 2018 emerging trends in electronic devices and computational techniques (EDCT); 2018. p. 1–6. doi:10.1109/EDCT.2018.8405057.

[8] Zhiwei H, Yuanyuan L, Xueyi Y. Models of vehicle speeds measurement with a single camera. In: Proceedings 2007 international conference on computational intelligence and security workshops; 2007. p. 283–6. doi:10.1109/CISW.2007.4425492.

[9] Madasu VK, Hanmandlu M. Estimation of vehicle speed by motion tracking on image sequences. In: Proceedings 2010 IEEE intelligent vehicles symposium; 2010. p. 185–90. doi:10.1109/IVS.2010.5548051.

[10] He XC, Yung NHC. A novel algorithm for estimating vehicle speed from two consecutive images. In: Proceedings 2007 IEEE workshop on applications of computer vision; 2007. 12–12

[11] Maduro C, Batista K, Peixoto P, Batista J. Estimation of vehicle velocity and traffic intensity using rectified images. In: Proceedings 2008 15th IEEE international conference on image processing; 2008. p. 777–80. doi:10.1109/ICIP.2008.4711870.

[12] Jeyabharathi D, Dejey D. Vehicle tracking and speed measurement system (vtsm) based on novel feature descriptor: Diagonal hexadecimal pattern (dhp). J Visual Commun Image Represent 2016;40:816–30. doi:10.1016/j.jvcir.2016.08.011.

[13] Cheung S-CS, Kamath C. Robust background subtraction with foreground validation for urban traffic video. EURASIP J Appl Signal Process 2005;2005(14):2330–40. doi:10.1155/ASP.2005.2330.

[14] Brutzer S, Hferlin B, Heidemann G. Evaluation of background subtraction techniques for video surveillance. In: Proceedings 2011 IEEE conference on computer vision and pattern recognition; 2011. p. 1937–44. doi:10.1109/CVPR.2011.5995508.

[15] Hua S, Kapoor M, Anastasiu DC. Vehicle tracking and speed estimation from traffic videos. In: 2018 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW); 2018. p. 153–1537. doi:10.1109/CVPRW.2018.00028.

[16] Doan S, Temiz MS, Klr S. Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera. Sensors 2010;10(5):4805–24. doi:10.3390/s100504805.

[17] Farnebäck G. Two-frame motion estimation based on polynomial expansion. In: Proceedings 13th Scandinavian conference on image analysis; 2003. p. 363–70. doi:10.1007/3-540-45103-X_50.

[18] Kaehler A, Bradski G. Learning OpenCV 3: computer vision in C++ with the OpenCV library. 1. O'Reilly Media, Inc.; 2016. ISBN 1491937998, 9781491937990.

[19] Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 2003;25(5):564–77. doi:10.1109/TPAMI.2003.1195991.

[20] Wang N, Du H, Liu Y, Tang Z, Hwang J. Self-calibration of traffic surveillance cameras based on moving vehicle appearance and 3-d vehicle modeling. In: 2018 25th IEEE international conference on image processing (ICIP); 2018. p. 3064–8. doi:10.1109/ICIP.2018.8451478.

[21] Comaniciu D. Nonparametric information fusion for motion estimation. In: Proceedings 2003 IEEE computer society conference on computer vision and pattern recognition, 1; 2003. p.I-59-I-66 doi: 10.1109/CVPR.2003.1211338

[22] Czajewski W, Iwanowski M. Vision-based vehicle speed measurement method. In: Proceedings 2010 international conference computer vision and graphics, Part I; 2010. p. 308–15. doi:10.1007/978-3-642-15910-7_35.

[23] Tougaw D. Finding your way with the garmin gps v. Comput Sci Eng 2002;4(3):10–13. doi:10.1109/MCISE.2002.998635.

[24] Nguyen TT, Pham XD, Song JH, Jin S, Kim D, Jeon JW. Compensating background for noise due to camera vibration in uncalibrated-camera-based vehicle speed measurement system. IEEE Trans Vehicular Technol 2011;60(1):30–43. doi:10.1109/TVT.2010.2096832.

[25] Celik T, Kusetogullari H. Solar-powered automated road surveillance system for speed violation detection. IEEE Trans Ind Electron 2010;57(9):3216–27. doi:10.1109/TIE.2009.2038395.

Saleh Javadi received B.Sc. in Electrical-Control Engineering from Amirkabir University of Technology in 2009, M.Sc. in Electrical, Electronic and Systems Engineering from The National University of Malaysia in 2013 and the Licentiate Degree in Systems Engineering from Blekinge Institute of Technology (BTH). He is currently pursuing a Ph.D. at BTH and his research interests include Computer Vision and Machine Learning.

Mattias Dahl is a Professor of Systems Engineering at Blekinge Institute of Technology. He received a B.Sc. in Electrical Engineering from Chalmers University of Technology, a M.Sc. in Computer Science from Lulea Institute of Technology and a Lic./Ph.D. in Applied Signal Processing from Lund University and Blekinge Institute of Technology. Currently, his research focuses on Autonomous Vehicles and Transport Technologies.

Mats Pettersson received a M.Sc. in Engineering Physics and a Ph.D. in Signal Processing from Chalmers University of Technology. He has worked at Ericsson and spent 10 years at the Swedish Defence Research Agency. He has been at Blekinge Institute of Technology since 2005 as a Full Professor. His work is related to Remote Sensing and SAR Processing.