DETERMINISTIC ALGORITHM FOR TRAFFIC DETECTION IN FREE-FLOW AND CONGESTION USING VIDEO SENSOR

Shah Md Muniruzzaman
Military Institute of Science & Technology
Email: head@ce.mist.ac.bd, Tel: 88-02-9013166

Nazmul Haque
Bangladesh University of Engineering and Technology
Email: nhaque20@gmail.com, Tel: 88-02-55167100

Fahmida Rahman
Bangladesh University of Engineering and Technology
Email: famibuet10@gmail.com, Tel: 88-02-55167100

Mohammad Rayeedul Kalam Siam
Bangladesh University of Engineering and Technology
Email: rayeedulkalam@gmail.com, Tel: 88-02-55167100

Rafee Musabbir
Bangladesh University of Engineering and Technology
Email: rafeemusabbir_buet@yahoo.com, Tel: 88-02-55167100

Md. Hadiuzzaman
Bangladesh University of Engineering and Technology
Email: mhadiuzzaman@ce.buet.ac.bd, Tel: 88-02-55167100

Sanjana Hossain
Bangladesh University of Engineering and Technology
Email: sanjana@ce.buet.ac.bd, Tel: 88-02-55167100

ABSTRACT

This paper presents a novel algorithm for accurate traffic detection through video sensor. The algorithm involves deterministic parameters, which aid in defining detection challenges and developing solutions. The algorithm detects traffic in three conditions: (1) stationary, (2) slowly moving, and (3) free-flow. The innovation involves the use of transparency effect for frame correction to deal with three challenges: (1) illumination variation, (2) shadow, and (3) camera displacement. To test the applicability of the proposed algorithm, both qualitative and quantitative analysis are performed with respect to four BMC (Background Model Challenges) algorithms such as Kernel Density Estimation (KDE), Gaussian Mixture Model (GMM), Pixel-Based Adaptive Segmenter (PBAS) and Sigma-Delta. The evaluation tasks are performed using synthetic and real-time video dataset. The developed algorithm achieves overall classification accuracy (>95%) with small computational effort (500 frame per second). A set of equations are developed to estimate speed, flow and density which can be readily adopted with other detection algorithms. Finally, correlations between traffic parameters and algorithm evaluation parameters are established to determine the applicability of any image-processing algorithm for traffic state estimation. The increment or decrement of evaluation parameters marks the characteristics of an algorithm, which dictates whether the algorithm is suitable for traffic state estimation.

Keywords: Traffic detection; Video sensor; Traffic state estimation; Deterministic algorithm; Image-processing

1. Introduction

Accurate detection of traffic employing various spatial and temporal sensors contributes significantly in describing and improving traffic condition. For this, video-based sensors are most appropriate over other alternatives as they provide high quality traffic information over spatial and temporal dimensions. Video sensors are centered on image processing algorithms. These algorithms are involved in image extraction, processing and recording specific information. The ideal requirement of these detection algorithms is dictated by the technique and the parameters involved in analysis. In real-time context, a detection algorithm should be able to detect traffic in a variety of situations. This actually recommends that the algorithms must be robust to various challenges such as shadow, illumination variation, camouflage and noise associated with the video dataset. These challenges affect the accuracy of traffic detection resulting in overestimation or underestimation. For instance, illumination variation causes increase or decrease in the intensity of pixels, resulting in false positive or overestimation. Likewise, lower intensity of illumination causes camouflage, generating false negative or underestimation. Moreover, presence of shadow also disrupts vehicle detection and causes overestimation. Similarly, presence of camera movement induces pixel displacement from

its designated position that results in overestimation. All these challenges are generated by natural phenomenon. For example, illumination variation is a natural phenomenon, which occurs gradually and sometimes suddenly due to weather variation. On the contrary, camouflage depends on the color of the pavement and the vehicle. In any case, when the vehicle color is equivalent or deeper than the pavement, the video sensor faces difficulties in distinguishing traffic. Another practical requirement of the algorithm is that it should be sensitive enough to detect minor changes of traffic flow and stop-and-go situation. In fact, the algorithm should be capable of detecting fast moving, slowly moving, and stationary traffic over the roadway.

Scope of accurate and reliable detection information is that it can be adjusted to enhance incident detection framework, while distinguishing stationary vehicles, complex maneuvers (lane change, right or left turn) and acceleration or deceleration patterns that are indicative of incidents beyond the sensor's field of view (FOV) (Mandellos et al. 2011). Besides, these information can be used to conduct traffic studies, such as examining vehicle maneuvers in weaving sections or bottlenecks, detecting pedestrians (Guo et al. 2012)**,** detecting vehicles in complex traffic scenarios (Mandellos et al. 2011) and extracting individual vehicle data (e.g. spacing, headway, velocity, acceleration), which in due course lead to better traffic flow modeling and an improved understanding of driver behavior.

Coifman et al. (1998) investigated vehicle detection and tracking methods and estimated traffic flow parameters from surveillance video. They developed a feature-based tracking algorithm for detecting vehicles under challenging conditions. The algorithms are robust to partial occlusion by tracking certain features of a vehicle instead of the entire vehicle. More recent vehicle detecting approaches are model-based methods that use prior knowledge to detect the desired targets (Lai et al. 2010; Shen, 2008), or deformable templates which are used when targets are matched against known vehicle models in the video frames (Takeuchi et al. 2010), or involving camera calibration and vehicle tracking (Wan et al. 2014). Pletzer et al. (2012) proposed a robust traffic state estimation technique**,** where the algorithm is integrated on an embedded smart camera and tested in various road sections as well as illumination conditions. The experimental result shows that it performs exceedingly well for stationary traffic. Some researchers also investigated individual flow parameters and exposure such as speed (Mao et al. 2009), flow (Pan et al. 2010) and incident detection (Zou et al. 2009) employing video sensors.

Normally, for vehicle detection, most methods (Gupte et al. 2002; Hsu et al. 2004) are based on the assumption that the camera is static. Given such assumption, foreground vehicles can be detected by image differencing between current frame and estimated background. For example, Gupte et al. (2002) presented an algorithm where vehicles in monocular image sequences of traffic scenes are recorded by a stationary camera based on a three level processing approach such as raw images, region level, and vehicle level. For vehicle detection in urban traffic scenes, Vargas et al. (2010) focused on background subtraction algorithm based on the sigma-delta filter. In the developed model, a reference image is considered to draw a comparison with each video frame. Consequently, this operation helps to eliminate all the stationary objects and extract foreground image of the frame. However, this method cannot cope up with shadow problems and the detection is limited to moving vehicles. Mandellos et al. (2011) proposed an algorithm to reconstruct the actual background color map without the need of any human intervention even in harsh traffic conditions, such as stop-and-go traffic flow, stationary vehicles (i.e. accident) and rain or snow. In a survey approach, Buch et al. (2011) conducted a comprehensive review on the computer vision techniques employed in urban condition to mark the potential of different algorithms for traffic analysis.

It should be noted that to achieve the primary goal of detection, several type of image processing techniques are available in literature. However, background subtraction (BGS) technique is the simplest and most computationally efficient among the multitude of techniques (Bouwman, 2014). General application of BGS based video sensors for detecting vehicles commence by separating the static portion of the scene (background) from the non-static portion (foreground) while only identifying moving objects of interest (i.e. moving vehicles). However, in order to detect and measure accurate traffic flow parameters more emphasis is required to detect the presence of all the significant elements over the carriageway whether static or moving. This specific consideration of static object has not been made valid for many of the detection approaches, which consider object as a moving entity and absorbs long duration static object while in congestion into the background model.

Even with the recent advances in traffic detection technique and technology, still there are challenging issues in vehicle detection, especially when there are: (1) moving objects that merge into the background due to a temporary stop and then become foreground again, after resuming its motion; (2) rapid changes in background lighting; (3) irregular camera displacement; and (4) shadow interference. The goal of this study is to estimate traffic parameters in urban traffic environments, where complex conditions such as dense traffic flow, traffic congestions, or vehicle queues are likely to appear. For this purpose, a novel algorithm for accurate traffic detection is devised. Moreover, a set of equations are developed to estimate traffic parameters that can be readily adopted with other detection algorithms. Interestingly, most of the available methods build on addressing challenges from the image-processing point of view, making it difficult to select a particular method to be useful in traffic application. Thus, correlations between traffic parameters and algorithm evaluation parameters are established to determine the applicability of any image-processing algorithm for traffic state estimation.

2. Defining Traffic Detection Challenges

Before approaching towards a solution, it is important to define the problem and the underlying principles that are active in the backdrop. For instance, traffic detection operation faces real-time challenges that affect the analysis and output of an algorithm in particular. In this regard, several mechanisms are required to address specific challenges in the algorithm. Another important consideration is the computational complexity. It is plausible to solve most or all challenges at the expense of computational or automation effort. So there lies a tradeoff between an accurate detection algorithms and the computational complexity, best

described by the involved parameters. Actually, the number and type of parameters used in a particular technique affect the rigor or intensity of the analysis procedure. The smaller the number of parameters involved in a methodology the simpler it becomes for analysis. For instance, the interactions of different parameters require rigorous calibration efforts since an optimized set of parameter values are required for a particular set of solution. From this elaboration, it is apparent that a more balanced algorithm is warranted where a small number of parameters are considered and all challenges are addressed accordingly. In short, an algorithm should be capable to generate highly accurate output with least computational requirement.

Table 1 represents a list of parameters with a goal to describe and generate solution for the relevant challenges. It is important to focus on the main parameters for the detection algorithm, which are static background and threshold value. The other parameters have been utilized to define the problem and derive a solution. Parameters used for the application of the detection algorithms in determining the traffic state parameters are explained in detail in the application section.

Table 1: Parameters involved in analysis and evaluation

Symbols	*Parameter Name*	*Unit*
B^*	Static background	Pixel
$B(t)$	Actual Background at time t	Pixel
$I(t)$	Frame at time t	Pixel
$d(t)$	Differential Image at time t	Pixel
$k_1(t)$	Intensity variation due to illumination change at time t	Pixel
$k_2(t)$	Intensity variation due to camera movement at time t	Pixel
$F(t)$	Intensity of vehicle at time t	Pixel
$\tau(t)$	Threshold requirement at time t	Pixel
e	Error term	Square Pixel
α	Transparency factor	N/A
$P(t)$	Vehicle Present at time t	Vehicle
$n(t)$	Number of entered vehicles at time t	Vehicle
$N(t)$	Count at time t	Vehicle
$q(t)$	Flow at time t	Veh./hour
w	Width of counting strip	Pixel
C_w	Width of a car	Pixel
A	Area of car	Sq. Pixel
L	Length of the strip	Pixel
s_i	Change in co-ordinate of vehicle	Pixel
N_j	Number of vehicles present at frame j	Vehicle
t'	Time interval between consecutive frame	Second
m	Number of frames	-

As mentioned previously, in defining the detection problem, several operations and challenges in the detection algorithms have to be outlined and defined systematically. Since complete understanding of the characteristics aids to generate a solution for individual challenges, such terms are depicted below.

i) Static background B^*: Static background relates to a specific frame, static in nature and temporal dimension. It is selected from a sequence of frames or from the mean of several frames. However, consideration of a static background is faces three challenges, such as: (1) Illumination variation $k_1(t)$; (2) Camera Displacement $k_2(t)$; (3) Movement of permanent object. For simplicity in analysis procedure, these challenges can globally be termed as noise.

ii) Illumination variation $k_1(t)$: Illumination variation challenge is dynamic in nature and varies temporally. It is directly linked to daily sunlight variation and climatic condition. It attempts to increase or decrease the intensity of a pixel at a certain position with sunlight variation, causing gradual or sudden change in illumination intensity.

iii) Camera displacement $k_2(t)$: The complexity imparted by camera displacement is also dynamic in nature and the occurrence is more uncertain than illumination variation. It is caused due to wind action over the video sensor and other unwanted vibrations. This particular phenomenon displaces image details from its actual position, inducing intensity variation on both pixels origination and destination.

iv) Frame $I(t)$: Frame is a component of the video, which is dynamic in nature and contains all of the information with in a field of vision.

v) Actual background $B(t)$: Dynamic background contains all information of the current frame except traffic.

Since detection approach focuses on traffic, permanent object movement within the study area can be neglected. Otherwise, the permanent object will act as traffic and the distinction between traffic and permanent object will become invalid.
Considering illumination variation and camera displacement, actual background $B(t)$ can be expressed as $B(t) = B^* + k_1(t) + k_2(t)$. Any frame at time t, $I(t)$ consists of the actual background model at time t, $B(t)$ and intensity of traffic at time t, $F(t)$. The parameter $I(t) = B^* + k_1(t) + k_2(t) + F(t)$, represents a pixel in a frame or the entire pixels of the frame. For the latter one, the parameters act as a 2D matrix for single color channel and 3D matrix for RGB or CMYK color channel.

The differential image $d(t) = I(t) - B(t)$ is obtained from the simple subtraction operation. However, large uncertainty associated with obtaining actual background may limit the availability of $B(t)$. For this, B^* can be adopted instead of $B(t)$ resulting in the expression $d(t) = I(t) - B^*$. Then the traffic extracted can be expressed as, $F(t) = d(t) - k_1(t) - k_2(t)$. The magnitude of $F(t)$ can be both positive and negative, whereas the other parameters are non-negative integers varying from 0 to 255. When the background will have a higher intensity than the respective frame, the traffic intensity will become negative. Such phenomenon is common when a black colored car travels over the pavement.

In case when the noises due to $k_1(t)$ and $k_2(t)$ become zero, the differential image would result in traffic image. However, this condition will never prevail because of the static background. The resulting traffic image promotes the need of a threshold value $\tau(t)$, the baseline intensity above which the differential image will be considered as traffic. This threshold $\tau(t)$ filters the noises from $d(t)$ and extracts the traffic from the differential image. The threshold requirement depends upon the magnitude of noise; where, the higher intensity of noise requires the increased threshold value. Therefore $\tau(t)$ can be expressed as a function of noises, such as $\tau(t) = k_1(t) + k_2(t) = f(k_1, k_2)$. Likewise, object in the field of view will be detected as, $F(t) = d(t) - \tau(t)$ when $d(t) \geq \tau(t)$; and $F(t) = 0$ for other considerations. The intensity of the traffic $F(t)$ largely depends upon the selection of the threshold value $\tau(t)$. It varies from 0 to 255. In fact, a higher threshold value can eliminate the object. On the contrary, the underestimation of threshold value may allow ingression of noise within the detected traffic. For example, when the threshold value $\tau(t)$ increases, severe noises from the image are eliminated. Consequently, the possibility of eliminating the object escalates, especially for $\tau(t) > d(t)$. Therefore the threshold level should be adjusted as minimum as possible towards zero. Understandably, to minimize the effect of noise $d(t)$ must be equal to $F(t)$ and this condition will be fulfilled when $\tau(t)$ becomes zero.

3. What Kind of Technique Should be Appropriate?

A multitude of detection technique involving background subtraction can be found where detection is achieved through methods, such as parametric probabilistic background model (e.g. Gaussian Mixtures Models (GMM)) for each background pixel (Stauffer and Grimson, 1999; Zivkovic and Van der Hijden, 2006), by background reconstruction (Mandellos et al., 2011), by determining the optimal threshold for foreground–background segmentation and object detection (Karasulu and Korukoglu, 2012). Among the stated methods, GMM methodology models each pixel history as a cluster of Gaussian type distributions and uses an on-line approximation to update its parameters. Accordingly, the background is found as the expected value of the distribution corresponding to the most populated cluster (Stauffer and Grimson, 1999). This method is able to tackle low illumination variations whereas rapid variations of illumination and shadows still remain difficult. Furthermore, the learning stage can be inefficient if it is realigned with noisy video frames (Sobral and Vacavant, 2014). Addressing such difficulties, previous GMM methodology is greatly improved on grounds of performance by considering recursive equations to adaptively update the parameters of the Gaussian model (Zivkovic and Van der Hijden, 2006).

In effect, parametric models are tightly coupled with underlying assumptions. They occasionally deviate from conforming to the real data. Moreover, the choice of parameters can be cumbersome, increasing research effort and reducing automation. On the contrary, nonparametric models are more flexible while intensely data oriented. In this regard, Pixel Based Adaptive Segmenter (PBAS) provides a better solution being a nonparametric model. In this technique, multitude of tunable parameters is adaptively adjusted for each pixel separately during runtime for optimal algorithms performance. In this approach, two per-pixel thresholds such as decision threshold and learning parameter are dynamically changed based on an estimate of the background dynamics. However, the method requires extensive tuning while detecting moving objects stopped suddenly, sudden light changes, cast shadows, and moving trees or dynamic background. Another established nonparametric algorithm is Kernel Density Estimation (KDE). It attempts to solve the problem relating to the presence of a dynamic scene with fast variations or

non-stationary properties, where the background cannot be modeled accurately with a set of Gaussians. It tackles the problem by estimating background probabilities at each pixel from many recent samples using KDE (Elgammal, 2000).
Another simple yet powerful non-linear background subtraction technique is sigma-delta background estimation (Manzanera and Richefeu, 2007). It consists of incrementing or decrementing the current background estimate by a constant value if it is smaller or greater than the current analyzing frame. The limitations remain in its uni-modal nature which are: (1) one single mode in the density model is inefficient to discriminate moving objects over a complicated background, and (2) one single dispersion estimate, related to one time constant, is not sufficient for certain kind of motion such as remote objects with radial velocity with respect to the optical centre.

An algorithm is best described as: (a) variables with known fixed values and connected by a known equation that is deterministic; or (b) variables with random values and connected by a known or unknown equation that is probabilistic. In fact, Deterministic approaches develop systems that are perfectly predictable. Such approaches follow established laws, equations or fixed procedures so that the state of each component and of the entire algorithms can be determined at any time for any time in the past and future. For statistical approach, rules can be derived from a large number of similar events based on experience where directly applicable observations can be transferred to the algorithms or to the event level. On the contrary, probabilistic algorithms involve a degree of uncertainty in predicting behavior and require "random variables" to describe a particular system's components and their interactions. Moreover, there is no general consensus on the actual meaning of "randomness" of a particular algorithm. For example, randomness could mean generated by chance mechanism, being unpredictable, showing a lack of an apparent order, etc. Any engineered intervention or an innovative solution should adhere to deterministic approach prior to the path of probabilistic solution since deterministic solution grows over the potential formulations and measures the actual condition to determine the performance of an algorithm. While probabilistic method is apart from deterministic solution, it offers flexibility to address different challenges during analysis such as occlusion, shadow, noise, and camouflage. Generally, probabilistic models (parametric or nonparametric) are robust against rain, snow, sleet, hail, and overcast and handles sensitive motion from swaying branches, rippling water, noise. They also perform under day and night condition. However, they face difficulty in detecting long shadow, object overlapping (object closely spaced from one another) and lighting changes. Moreover, they suffer from errors such as ghost phenomenon that arises when the object remains static for a long duration over time-space dimension, resulting in overestimation. This particular phenomenon governs while the signal timing is more than usual or in congested scenario. Generally probabilistic background models based on greatest occurrence of background pixels in the image sequence, fail to detect such stationary vehicle. Because the stationary vehicle is stored as a part of the image sequence rather than a distinct object, the pixels belonging to an object become the part of the background that consequently eliminates the detected object in the binary image regardless of the opted threshold value.

This paper presents a new background subtraction algorithm for detecting vehicles in both moving and stop-and-go situation. In this algorithm, deterministic analytical approach has been adopted instead of conventional probabilistic approach. From the benchmark provided by BMC (Background Models Challenge), a set of four methods are extracted that clearly overtake the existing ones (Sobral and Vacavant, 2014) to test the performance. Quantitative and qualitative analysis of the proposed algorithm are performed with respect to Kernel Density Estimation (KDE), Gaussian Mixture Model (GMM), Pixel-Based Adaptive Segmenter (PBAS) and Sigma-Delta.

4. Solving the Detection Problem: Analytical Approach

In order to derive an analytical solution to the defined challenges of traffic detection, $\tau(t)$ can be taken as τ. Because $\tau = f(k_1, k_2)$ can be written as $\tau = k(t)$ considering the influence of noises (k_1, k_2) upon threshold as they cannot be separated easily. Accordingly, to ensure $\tau = 0$, $k(t)$ corresponds to 0. At this instant, $k(t)$ can only be determined if the actual background $B(t)$ at time t is known,

$$k(t) = B(t) - B^* \quad (1)$$

$k(t)$ represents the difference between a static frame and a dynamic frame. The subtracted value can be both positive and negative. For example, when a comparison is drawn between day background and night background, the value of $k(t)$ will be negative. Similarly, when camera displacement causes a pixel to be replaced by a lower intensity value after background selection, $k(t)$ will be negative. On the contrary, when camera displacement causes a pixel to be replaced by a higher intensity value after background selection, $k(t)$ will be positive.

Since acquisition of $B(t)$ involves controlled environment, which may not be feasible in many urban scenarios $I(t)$ can be preferred instead of $B(t)$ ($B(t) \in I(t)$; $B(t) \subset I(t)$). The relationship between $I(t)$ and (B^*), considering $k(t) = 0$ and $k(t) = B(t) - B^*$ is outlined as follows.

$$I(t)^2 + (B^*)^2 = 2I(t)B^* \quad (2)$$

When the equation is true, noise will be eliminated or vice-versa. Because, $B^* = I(t) - k(t) - F(t)$, the Equation *(2)* can be rewritten as follows.

$$I(t)^2 + I(t)^2 + k(t)^2 + F(t)^2 - 2k(t)I(t) - 2F(t)I(t) + 2k(t)F(t) = 2I(t)B^* \quad (3)$$

In Equation *(3)*, the product has been expressed using $I(t)$, $k(t)$ and $F(t)$ eliminating the user-defined parameter B^*. Considering Equation (3), an error term e can be outlined from Equation *(2)* such that,

$e = e(t) = k(t)^2 + F(t)^2 - 2k(t)I(t) - 2F(t)I(t) + 2k(t)F(t)$. Depending upon the interlaced relationship of the parameters ($F(t)$, $k(t)$ and $I(t)$) in the error term, it can be defined as e where, $F(t)$ and $k(t)$ is far less than $I(t)$; $k(t)I(t)$ and $F(t)I(t)$ are greater than $k(t)^2 + F(t)^2 + 2k(t)F(t)$.

$$e = \begin{cases} 0, if\ I(t)=B^* \\ <0, if\ I(t)\neq B^* \end{cases}$$

Therefore, Equation *(3)* can be rewritten as,

$$\sqrt{I(t)B^*} = I(t)\sqrt{1+\frac{e}{2I(t)^2}} = I(t)\sqrt{1-\frac{|e|}{2I(t)^2}} \quad (4)$$

Considering a factor α, ($\alpha = \sqrt{1-\frac{1}{2}\frac{|e|}{I(t)^2}}$) termed as transparency factor, Equation *(4)* is transformed as

$$\sqrt{I(t)B^*} = \alpha I(t) \quad (5)$$

Experimental tests show that, the value of α for an entire 2-dimensional frame varies from 1 to 1.154. So the value of α can be taken as 1. Thus the Equation becomes $I(t) \approx \sqrt{I(t)B^*}$, which gives rise to a new frame equivalent to the actual frame with lower noise intensity. Consequently, when this equivalent frame is used instead of the actual frame, noise is minimized. Subsequently, minimum threshold value is achieved. In terms of operation and outcome, Equation *(5)* is referred to as current frame correction since it only applies correction over the frame while recording all the traffic details. Thus, the induced transparency of vehicles in the field of view is termed as 'Transparency Effect' as detailed in Equation *(5)*.

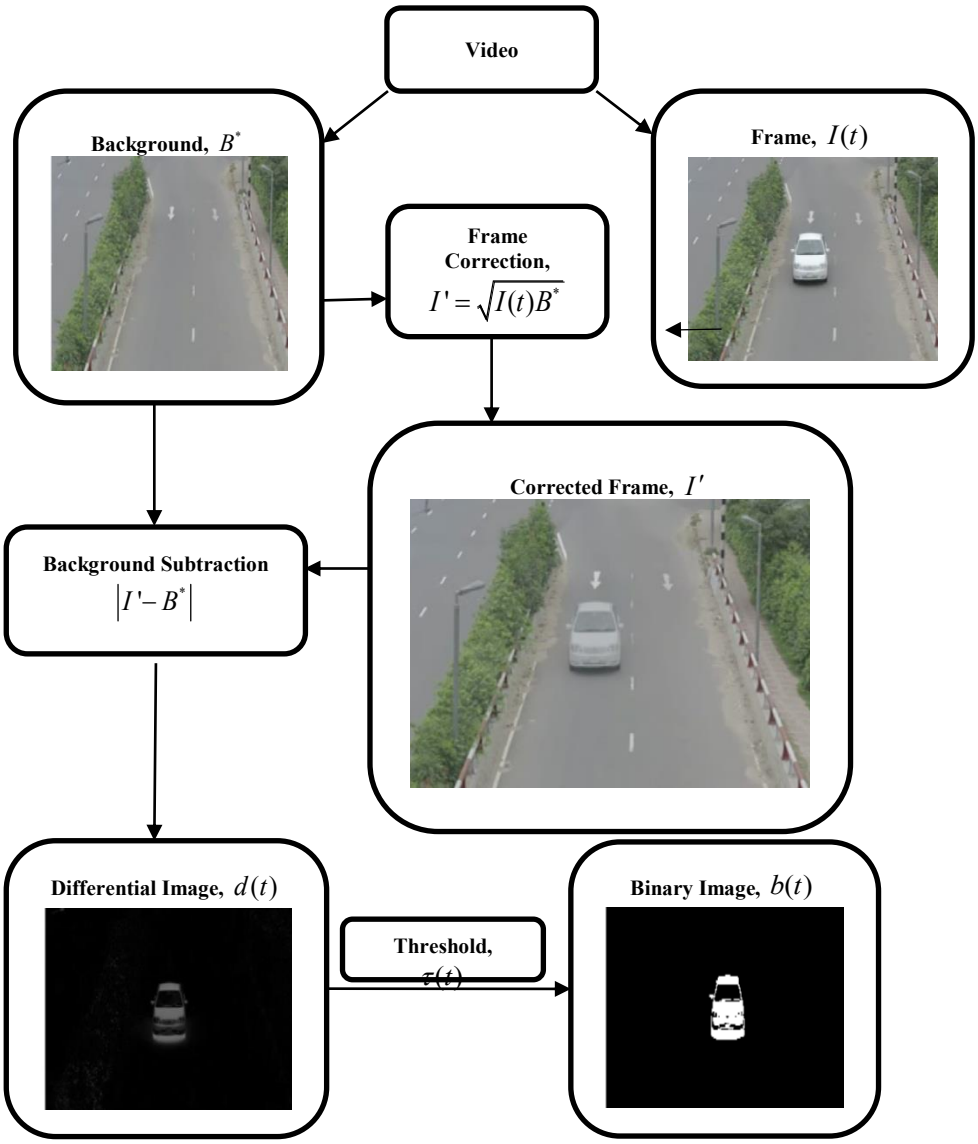
5. Methodology

Particular methodology of this study is outlined in Figure 1, which contains the important steps for the developed algorithm. Firstly, a background is selected from the video sequence without any traffic. Then, each video frame is passed through the frame correction technique according to Equation *(5)* and the differential image is obtained by subtracting the background from the corrected frame. After this, a vehicle image is obtained using a threshold. This image is then converted to a binary image using the same threshold to obtain a machine-readable content. Binary image, $b(t)$ means that if the pixel intensity in differential image is greater than a particular threshold, then the value in the binary image corresponding to that pixel is one and otherwise zero.

$$\left.\begin{aligned} b(t) &= 1, if\ d(t) \geq \tau(t) \\ &= 0, otherwise \end{aligned}\right\} \quad (6)$$

For the proposed algorithm, there are two parameter inputs: (1) static background; and (2) threshold value. Static background represents a frame without any vehicle in the field of view (FOV). Threshold value represents the intensity above which the difference is to be considered as traffic and has a range between 0 to 1. Otherwise, vehicles in the FOV will be eliminated. For this method, a threshold value of 0 to 0.15 is recommended.

Figure 1: Synthesis of frame correction, static background and threshold steps of the proposed algorithm



6. Data Collection

In order to evaluate the developed detection algorithms for free-flow and stop-and-go situation, two types of video sequence is considered: (1) synthetic, and (2) real-time.

- The synthetic video sequence is generated considering illumination variation, and camera displacement analogous to urban condition. For illumination variation, gradual and irregular sudden variation is adopted to simulate daylight variation and cloudy/rainy weather, respectively. Camera displacement is simulated by contrasting the pixel positions by 10% of the frame size in horizontal direction. It is done by replacing the coordinates of the pixel (x_t, y_t) with the $(x_t + \frac{B}{10}, y_t)$ at a certain time t, where B indicates the width of the frame.
- The real-time video sequence represents actual detection challenges, such as shadow interference, illumination variation and camera displacement. Moreover, high quality video dataset is collected to determine the accuracy of different techniques in actual condition.

In this study, ground truth relates to actual data measured from both type of video sequences. Particularly for the synthetic video, one ground-truth frame for each frame is considered, which corresponds to a sampling rate of 25-frames/second. Following this step, 1400 frames are labeled manually for ground truth extraction task. This task involves two important operations: (1) removal of noise, and (2) filling up of voids. For instance, in the synthetic video, background is considered as a frame with no object, a complete black frame to be particular. Then, this background is subtracted from the synthetic video. The noises and the voids generated from this subtraction process are removed and filled up, respectively.

7. Effect of Frame Correction

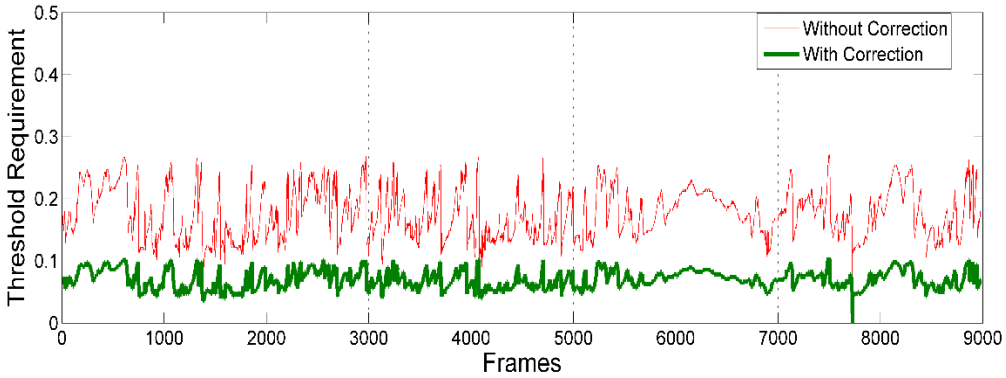
In this section, the effect of frame correction is outlined in detail. Frame correction is applied for noise reduction. It should be noted that a large set of frames is a prerequisite to observe the effect of the frame correction. For this reason, the analysis is performed over a sample containing 9000 frames from the real video dataset. The actual condition is represented in Figure 2((a)-(b)) where, (a) represents an arbitrary frame from the sample and (b) shows the effect after the application of frame correction. Interestingly, the vehicles over the carriageway became transparent and other features such as lane markings achieved visibility through the vehicles. Such condition emphasize that noises have also gone through the same effect since they have smaller pixel intensity than the vehicles.

The effect of such correction is explained through a qualitative comparison between the application of frame correction and without frame correction. Figure 2(c) illustrates the threshold requirement in a static background with and without frame correction. This shows that the sample frames go through a scale down effect. Both curves are analogous in shape; curve has lower amplitude than the former one and shifts downward. This shift eliminates noises and the amplitude reduction makes the impact of noise smaller in the differential image.

Figure 2(a)-(c): Frame correction technique; (a) real-time test frame; (b) real-time transparency effect; (c) threshold requirement with and without frame correction.

(a)

(b)



(c)

8. Experimental Analysis

This section outlines the comparison of state of art algorithms namely KDE, GMM, PBAS and Sigma-Delta with our proposed algorithm in various test situations. The experimental analysis for the proposed algorithm are carried out in two ways: (1) qualitative analysis; and (b) quantitative analysis. Different parameter values involved in evaluation measures are elaborated in Table 2.

Table 2: Different parameter values involved in evaluation measures (Sobral 2014)

Algorithm	
Parameter	Value
KDE	
Sequence Length	50

Time Window Size	100
SD Estimation Flag	Yes
Normalized RGB	Yes
Probability threshold	10e-8
suppress shadow rate	0.3
Binary threshold	0.15
GMM	
fading factor	0.05
Binary threshold	0.15
PBAS	
(default Parameter values)	
Sigma Delta	
(default Parameter values)	
Proposed Algorithm	
Static background	Located at 1st frame
Binary threshold	0

7.1 Qualitative Analysis

In this subsection, the visual quality or perception of a binary image of the developed algorithm in comparison to other techniques is illustrated for different challenges such as illumination variation, camera displacement, and shadow.

7.1.1 Illumination Variation

In this experimental analysis a synthetic video dataset is used to simulate the gradual and sudden illumination variation conditions. The actual frame of the synthetic dataset contains three yellow colored rectangular vehicles. The analysis illustrates that the proposed algorithm and PBAS outperforms KDE, GMM and Sigma-Delta. This particular result of KDE, GMM and Sigma-Delta can be attributed to noise surrounding the vehicle.

7.1.2 Camera Displacement

The second experimental analysis sequence is based on the performance of several techniques in camera displacement condition. Interestingly, the road marking in the synthetic sequence is detected as an object due to camera displacement. Analysis of different algorithms shows that the lane markings are identified as individual objects, where the proposed algorithm performs better than others.

7.1.3 Shadow Interference

In this experimental analysis, a synthetic video sequence is used to simulate the shadow condition. The qualitative analysis of different techniques shows the performance on eliminating shadow. For better representation, frame from real video dataset is used, where only the sigma delta algorithm failed to eliminate the shadow.

7.1.4 Stationary Vehicle Detection

In this experimental analysis, the ‘stationary vehicle’ sequence has been used to see the effect of stationary vehicle over the different algorithm. Figure 3 illustrates the binary image produced by different algorithms for a stationary vehicle. The result shows that only KDE and proposed algorithm have successfully detected the vehicle.

7.1.5 Real-time Video Dataset

The previous analysis was done over the synthetic videos to see the performance in various controlled situations. This subsection outlines the performance analysis of different algorithms over the real-time video dataset. To see the performance, three types of situation are selected: ‘Empty Road’, ‘Slowly moving traffic’ and ‘Congestion’.

Illumination Variation (Synthetic Video Sequence)

Actual Frame	KDE	GMM	PBAS	Sigma-Delta	Proposed Algorithm

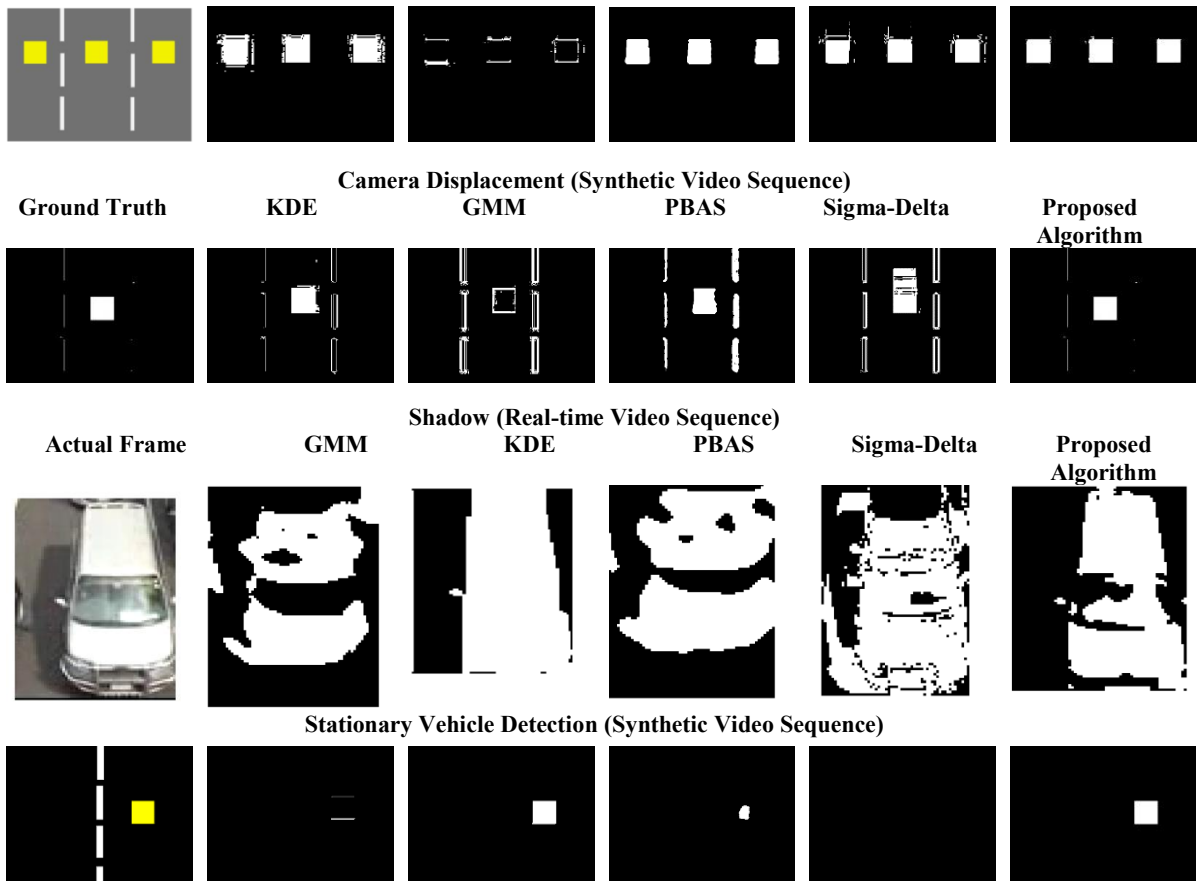


Figure 3: Qualitative comparison of video frames in illumination variation, camera displacement, shadow, and stationary vehicle detection.

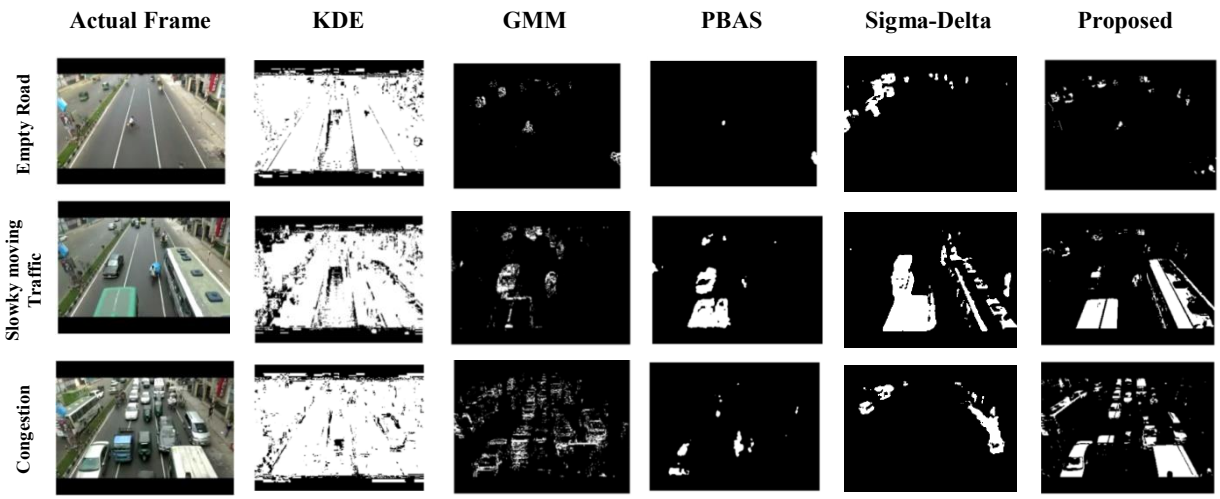


Figure 4: Qualitative comparison of real video frames in empty road, slowly moving traffic and congestion situation.

Figure 4 illustrates qualitative comparison of real video Frames for different algorithms. Table 3 explains the performance of different algorithms over these situations. The result shows that the proposed algorithm outperforms other methods.

Table 3: Description of qualitative evaluation of real video frames

Method	Empty Road	Slowly Moving Vehicle	Congestion

KDE	Frame is filled with FP pixels resulting in overestimation.	Accuracy is highly reduced as the frame is filled with FP pixels causing overestimation.	Accuracy is highly reduced for FP pixels in the frame, causing overestimation.
GMM	Can detect small vehicles and there is no impression of FP. FN is higher as the algorithm is unable to detect full details of a vehicle.	Can detect slowly moving traffic and there is no impression of FP. However, the detection accuracy is highly affected by FN since the vehicle is absorbed into the background for stop-and-go situation.	Can detect moving vehicles. However, unable to detect stationary vehicles.
PBAS	Cannot detect small vehicles and there is no impression of FP. However, FN is higher as the algorithm cannot detect the full details of the vehicle accurately.	Can detect slowly moving traffic and there is no impression of FP. However, the detection accuracy is highly affected by FN since vehicles are absorbed in the background for stop-and-go situation.	Cannot detect stationary vehicles as well as moving vehicles.
Sigma-Delta	Can detect small vehicles and suffers least amount of FP and FN. Whereas the detected vehicles are well-defined binary object.	Can detect slowly moving traffic and there is no impression of FP. However, the detection accuracy is highly affected by FN since vehicles are absorbed into the background for stop-and-go situation.	Cannot detect stationary vehicles as well as moving vehicles.
Proposed Algorithm	Can detect large as well as small vehicles and suffers least amount of FP and FN. The detected traffics are well-defined binary object.	Can detect slowly moving vehicles.	Can detect stationary vehicles. However, few vehicles are absorbed into the background due to presence of FN.

FP – False Positive (Overestimation)
FN – False Negative (Underestimation)
GMM – Gaussian Mixture Model

7.2 Quantitative Analysis

In this subsection, the quantitative analysis of different algorithms (GMM, PBAS, KDE and Sigma-Delta) with respect to the proposed algorithm is outlined in detail. Generally, an algorithm labels sample frame as either positive or negative in a binary decision problem. In this study, samples are considered as pixel values in binary image, where "positive" represents foreground object pixel, and "negative" represents background pixel. In this study two classes of image (foreground and background) are used for analysis. For a given frame in a video sequence a comparison can be drawn between the resultant image and the ground truth image. A pixel is denoted as white when it is a part of an object in the foreground, and black when it actually belongs to the background. To quantify the classification performance, with respect to ground-truth, four basic measures are used, such as (1) true positives (TP): correctly classified foreground pixels; (2) true negatives (TN): correctly classified background pixels; (3) false positives (FP): incorrectly classified foreground pixels; and (4) false negatives (FN): incorrectly classified background pixels.

ROC curves represent the variation of the number of incorrectly classified examples with the number of correctly classified positive examples (TPR versus FPR) as given by,

$$TPR = \frac{TP}{TP+FN}\ ;\ FPR = \frac{FP}{FP+TN}\ ;\ \text{and}\ FNR = \frac{FN}{FN+TP}$$

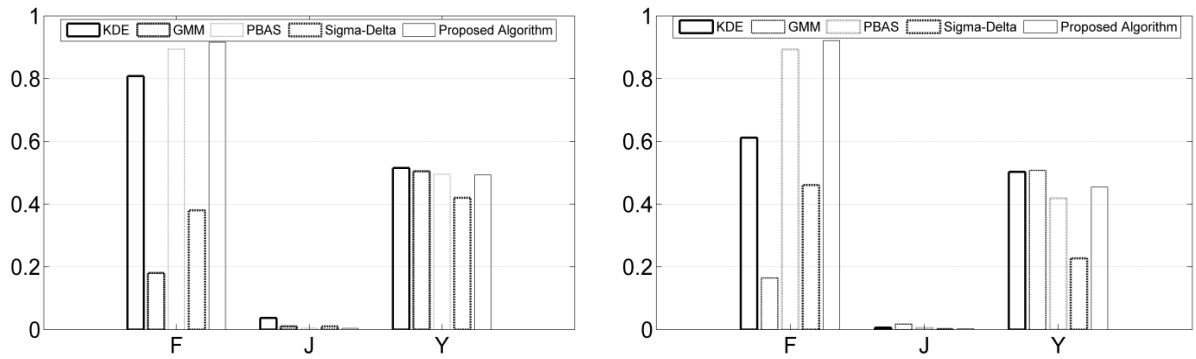
However, PR curves are a good alternative since ROC curves have the risk of providing an optimistic appraisal of the classifier's performance when there is a significant skew in the class distribution (Davis and Goadrich, 2006). Moreover, PR curves may display differences between the algorithms behavior that are not visible in the ROC curves. PR curves are assembled from the following metrics:

$$\text{Precision} = \frac{TP}{TP+FP}\ ;\ \text{Recall} = \frac{TP}{TP+FN}\ ;\ \text{and Specificity} = \frac{TN}{FP+TN}$$

With a goal to obtain more handy metrics, particularly in the form of scalar measures, the area under the ROC or PR curves can be utilized. Other measures for fitness quantification, in the context of background subtraction techniques, have been proposed in

the literature (Rosin and Ioanndis, 2003; White and Shah, 2007; Ilyaset et al. 2009) such as F-measure (*F*), Percentage of Correct Classification (*PCC*), Jaccard Coefficient (*J*), etc.

$$\left.\begin{aligned} F &= 2\left(\frac{PR.RE}{PR+RE}\right),\ (0 \le F \le 1) \\ PCC &= \frac{TP+TN}{TP+FN+FP+TN} \\ J &= \frac{TP}{TP+FN+FP};\ (0 \le J \le 1) \\ Y &= \frac{TP}{TP+FP} + \frac{TN}{TN+FN} - 1;\quad (-1 \le Y \le 1) \end{aligned}\right\} \qquad (7)$$



(a) (b)

Figure 5((a)-(b)): Evaluation measures for moving and stationary vehicle sequence.

F-measure (*F*) combines precision and recall in the form of their harmonic mean, providing an index that is more representative than the pure PR and RE measures themselves. PCC is the percentage of correct classification alone, commonly used for assessing a classifier's performance. However, it can also give misleading estimates for a significant skew in the class distribution (Rosin and Ioanndis, 2003).

Yule coefficient (*Y*) is particularly defined on the interval [-1, 1]. The lower limit of this interval occurs when there are no matching pixels, whereas the coefficient would hit the upper bound for a perfect match. The following index set is considered as a valuable quantification of relative performance of each algorithm: $S = \{F, J, 0.5\,(1+Y)\}$. The set includes fitness coefficients with an ideal value equal to 1. Evaluation measures for moving and stationary vehicle sequence are presented in Figure 5((a)-(b)). In Table 4, quantitative result for moving and stationary vehicle sequence are discussed in detail.

Table 4: Quantitative result for moving and stationary vehicle sequence

Evaluation Measures		KDE	GMM	PBAS	Sigma-Delta	Proposed
			Moving Vehicle Sequence			
Precision	μ	0.673	0.584	0.924	0.719	0.935
	σ	0.016	0.014	0.021	0.016	0.01
Recall	μ	0.966	0.113	0.86	0.885	0.906
	σ	0.001	0.007	0.005	0.001	0.004
Specificity	μ	0.973	0.99	0.996	0.983	0.995
	σ	0.0005	0.0002	0.0001	0.0003	0.0001
PCC	μ	0.974	0.94	0.988	0.978	0.992
	σ	0.0005	0.0013	0.0002	0.0005	0.0001
F	μ	0.789	0.18	0.893	0.38	0.916
	σ	0.007	0.006	0.009	0.013	0.008
J	μ	0.027	0.01	0.004	0.01	0.004
	σ	0.0001	0.0002	0.0001	0.016	0
Y	μ	-0.056	0.009	-0.009	0.0.42	-0.014

	σ	0.022	0.0004	0	0.017	0
			Stationary Vehicle Sequence			
Precision	μ	0.539	0.571	0.85	0.526	0.884
	σ	0.017	0.013	0.022	0.015	0.023
Recall	μ	0.08	0.51	0.949	0.61	0.952
	σ	0.01	0.004	0.0003	0.0063	0.001
Specificity	μ	0.991	0.993	0.995	0.989	0.998
	σ	0.0002	0.0002	0.0001	0.0003	0.00005
PCC	μ	0.975	0.978	0.995	0.983	0.997
	σ	0.0006	0.0005	0.0001	0.0004	0.00007
F	μ	0.164	0.597	0.893	0.583	0.921
	σ	0.01	0.003	0.003	0.01	0.002
J	μ	0.017	0.006	0.006	0.0015	0.003
	σ	0.0008	0.0001	0.001	0.0004	0
Y	μ	0.014	0.007	-0.163	-0.006	-0.092
	σ	0.0006	0.0002	0.0005	0.0003	0.024

μ = mean

σ = standard deviation

The main concept of evaluating algorithms using a constant threshold is to illustrate the stability of each algorithm. In this context, the clustering of points in one position represents more stability whereas scattered points represent that a particular algorithm fails to maintain its stability for a constant threshold. Furthermore, clustering of points at the upper right corner of Precision-Recall plot represent that algorithm is efficient in detection. Precision-recall relationship is illustrated in the Figure 6((a)-(e)). Results of false negative rate against time are presented in Figure 7((a)-(e)). Figure 7(f) illustrates the applied speed variation in the synthetic video to see the behavior of various algorithms in stop-and-go situation. The upward shift of the FNR curve represents the inability of traffic detection at stationary condition. Table 5 outlines the description of quantitative analysis of different algorithms.

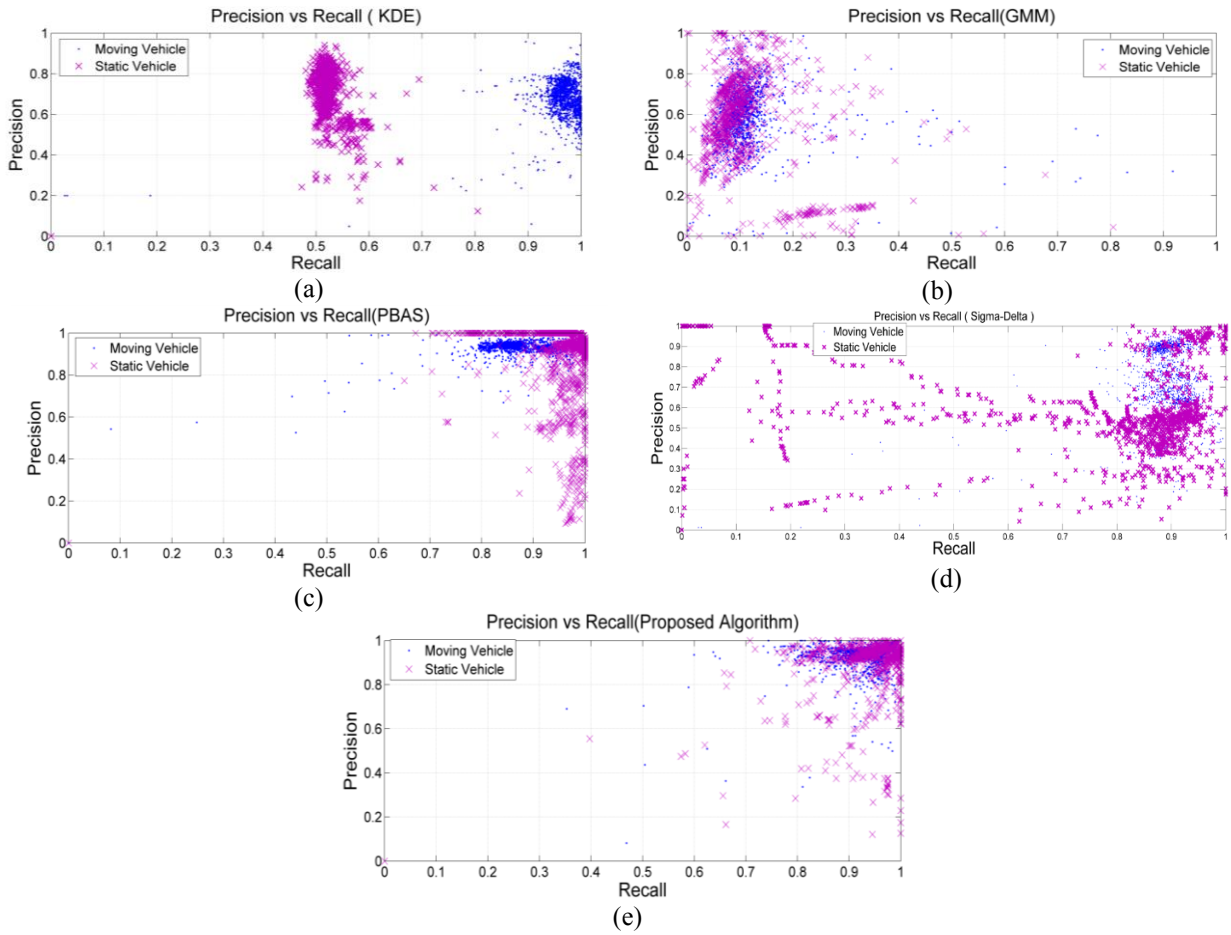


Figure 6((a)-(e)): (a)-(e) Precision-Recall Measure for different algorithms.

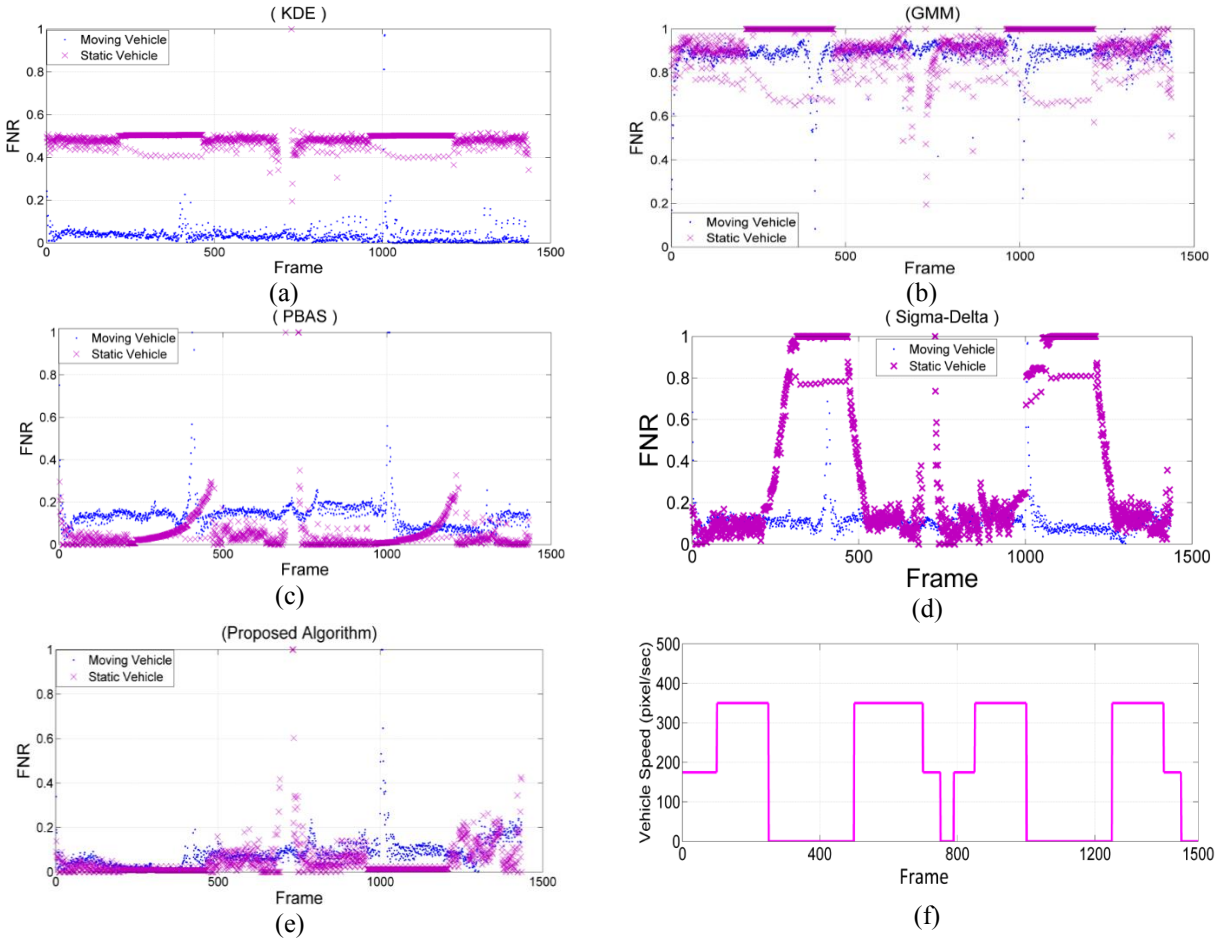


Figure 7 ((a)-(f)): (a)-(e) Detection of moving and stationary vehicles by different algorithms; and (f) applied speed variation in 'static vehicle' sequence.

Table 5: Description of quantitative analysis of different algorithms

Method	Precision-Recall Curve	FNR vs. Time Curve
KDE	• Shows good Precision-Recall relationship for 'Moving Vehicle' video sequences. • Recall decreases for 'Stationary Vehicle' sequence due to increase in FN values. • Shows good stability in both moving and stationary sequence as the points are clustered in a spot. It represents that the algorithm gives stable result for a single threshold value.	• Shows less amount of FNR for 'Moving Vehicle' sequence. • Shows increased amount of FNR for 'Static Vehicle' sequence. • The FNR remains almost constant with the time. Whereas for stationary condition it shifts upward a little. The speed of the vehicle in different frames is illustrated in Figure 7(f).
GMM	• Shows higher precision value and lower recall value for both 'Moving Vehicle' and 'Stationary Vehicle' video sequences. • Shows poor stability in PR relationship. The PR points are scattered, which means that the precision can vary for attaining a certain recall and vice versa. Therefore, for a single threshold value the algorithm will not give a stable result.	• It shows similar high FNR for both video sequences. • The FNR increases to maximum when the object comes to a rest condition.
PBAS	• Shows good Precision-Recall relationship for both 'Moving Vehicle' and 'Stationary Vehicle' video sequences. • Shows good stability in relationship.	• For both moving and stationary sequence the FNR is lower. • The FNR increases in a parabolic shape for the frames where the vehicles are static condition. Zero vehicle speed in Figure 7(f) represents the stationary condition.

Sigma-Delta	• Shows good precision-recall relationship for 'Moving Vehicle' sequence. • The precision value decreases for 'Static Vehicle' condition. • Shows poor stability in PR relationship as the points are scattered.	• For both condition the FNR is lower. • However, the FNR increase exponentially with time when the vehicle remains in stationary condition (Figure 7(f)).
Proposed Algorithm	• Shows good PR Relationship for both 'Moving Vehicle' and 'Stationary Vehicle' video sequences. • Shows stable Precision-Recall relationship as the points are dense in a certain area.	• FNR is low for both the video sequence. • No increase in FNR when the vehicle remains in stationary condition (Figure 7(f)).

7.2.1 Computational Complexity

The computational speed is observed on 320 x 232 size of frame having 3 color channels. Table 6 shows the performance of several algorithms on a similar platform (3.60 GHz Core i7 CPU, 8 GB of RAM, C++ implementation).

Table 6: Computational complexity of different algorithms

Method	Average Time/frame	FPS	CPU (%)
GMM (RGB)	4ms, 8ms at congestion	125	6.27
KDE (RGB)	13ms, 30ms at congestion	33	6.3
PBAS (RGB)	14ms (also constant in congestion)	70	6.2
Sigma-Delta (RGB)	2.5 ms	400	0.5
Proposed Algorithm (RGB)	2 ms (constant in congestion)	500	1.1

ms – Millisecond
RGB – Red, Green, Blue

Figure 8 shows that the average CPU time for the proposed algorithm is less than KDE, GMM, and PBAS, whereas a bit more than Sigma-Delta. For private memory, PBAS requirement is the highest. However, the proposed method consumes memory at a rate equivalent to Sigma-Delta, and GMM. The performance on average execution time shows that the proposed algorithm requires the least amount of time and the PBAS method requires the highest time.

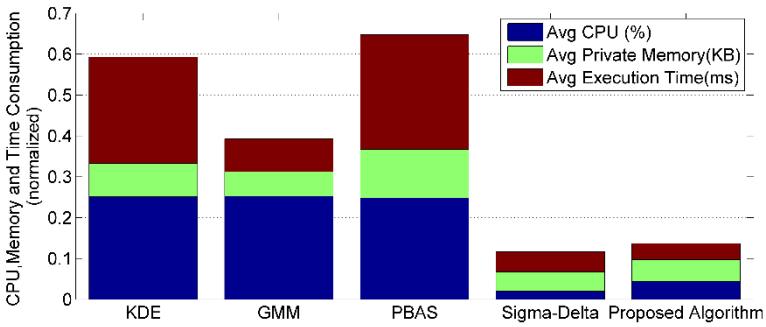


Figure 8: Computational complexity of different algorithms

7.3 Estimation of Traffic Parameters

7.3.1 Flow Estimation

Dimension of the analysis frame is an important element for fast processing as reduction in this feature reduces the processing effort significantly. In this regard, strip analysis technique is more fathomable and favorable over frame based or regional approach. Moreover, magnitude of error or noise arising from camera vibration is considerably lower for smaller strips in contrast to entire frame. As a result, careful selection of the strip in the field of view contains vehicle pixel data and acts as a catalyst to accelerate the process. In this study, a single strip is selected in the frame and pixel by pixel processing is adopted for vehicle counting or traffic flow measurement. This method provides a count of one for a vehicle present in the strip, whereas zero without a vehicle, thereby forming a strip matrix. Summation of binary ones in the strip matrix represents the total width of an object occupying the strip. Considering the width of a representative vehicle (e.g. passenger car), vehicles present in the frame at time t, $P(t)$ is outlined where, w relates to strip width, C_w expresses car width, and $b(t)$ is the binary image which is extracted by the proposed algorithm.

$$\left.\begin{aligned} P(t) &= \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} b(t)}{w \times C_w} \\ n(t) &= \begin{cases} \left(\frac{dP}{dt}\right)_t, & if\ dP>0 \\ 0 \end{cases} \\ N(t) &= \int_0^t n(t)dt \\ q(t) &= \left(\frac{dN}{dt}\right)_t \end{aligned}\right\} \quad (8)$$

The value of $P(t)$ largely depends on the vehicle width *w*. Large vehicle width tends to underestimate small vehicle counts, whereas small vehicle width when selected tends to overestimate. For this, the weighted mean of the observed composition is more suitable. The number of vehicle within the frame at time *t*, is represented as $n(t)$. When a vehicle enters the FOV, $n(t)$ will be non-negative, whereas $n(t)$ will return zero for a vehicle leaving the frame. In fact, the function responds to the increment of vehicles. Vehicle count is represented as $N(t)$, sum of the all vehicles that passed the field of view at up to time *t*. The flow $q(t)$ is obtained, differentiating the counting function in Equation *(8)*.

7.3.2 Density Estimation

Density is estimated employing the area of standard vehicle in the current frame or foreground. The technique operates through facilitating a larger and wider strip, positioned on the frame over which the density is analyzed rather than the entire frame region. To avoid overestimation or underestimation, mid-section is preferred over upstream and downstream location of the strip. Density $k(t)$ is calculated using the Equation *(9)* where, *A* relates to standard vehicle area, L represents the strip length.

$$k(t) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} b(t)}{A \times L} \quad (9)$$

The Equation is also sensitive to vehicle area A. The area of a vehicle changes in the field of view. On the contrary, the area also changes with the vehicle type. As a result, for practical purpose the midrange weighted average area is more suitable for this computation. Another parameter L also affects the result significantly. Small length vehicles when selected as reference, tends to overestimate density over a link. Moreover, L depends on the camera height and angle. An aerial view of a link can provide a conducive view for density computation.

7.3.3 Speed Estimation

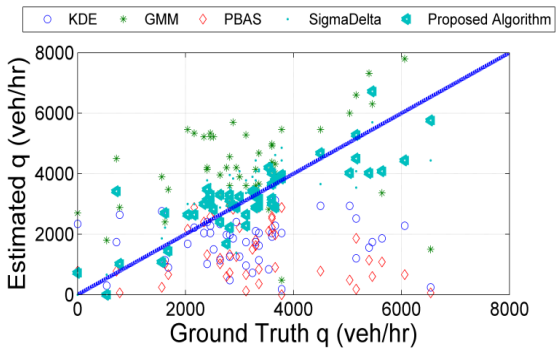
The speed of the vehicles in the field of view is computed at blob level. The frame is divided into several segments to determine the aggregated speed. This method is adopted to avoid excess utilization of computational resources like pattern recognition within consecutive frames. The instantaneous and mean speed of all the vehicles present in a frame can be determined by Equation *(10)*, where, v_j denotes to instantaneous speed at frame j, $\bar{v}$ relates to time mean speed at frame j; s_i relates to the change in position of object at segment i within time t' from frame $(j-1)$ to frame j; N expresses number of vehicles present in frame; n represents the number of user defined segments; N_j expresses number of objects present in frame j; and *m* corresponds to number of frames. The value of n is selected carefully depending upon the size of vehicles within the binary image, not in the field.

$$\left.\begin{aligned} v_j &= \frac{\sum_{i=1}^{n} s_i}{N \times t'} \\ \bar{v} &= \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} s_i}{N_j \times t' \times m} \end{aligned}\right\} \quad (10)$$

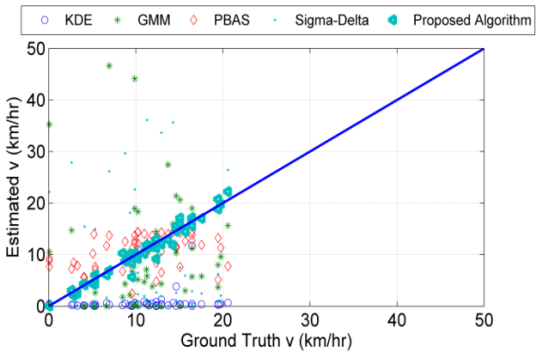
7.3.4 Estimated Traffic Parameters

Using Equation (8)-(10) traffic state parameters are estimated for different detection algorithms. Figure 9((a)-(b)) compares the estimated parameters with the actual one, which shows that the algorithm performs with much accuracy in flow and speed. Figure 9(a)-(b) shows scatter plots of the algorithm output versus manually extracted ground truth data for velocity and Flow. Figure 9(c) summarizes the RMSE for speed, flow and density. Output of the detection algorithms show that the estimated speed is accurate. Even if the vehicles suffer from occlusion or splitting, the imperfect blobs still move at the normal speed. The errors in flow and density are primarily due to missed or divided vehicles at their windshield. Often, an error of two or three vehicles in one sample can be very significant. For example, one missed vehicle in a 1 min sample can cause 60 veh/hour differences. At 2500 veh/h less error is projected in the output whereas for 1500 veh/h and 4500 veh/h, the error per missed vehicle is slightly higher. The greatest source of error appears to be from the selection of appropriate background that ultimately yields noise to the frames. As a result, the detection algorithm underestimates some vehicles, having same color as the pavement. The results are promising nonetheless and the error shown in Figure 9(c) (RMSE) should be taken in context.

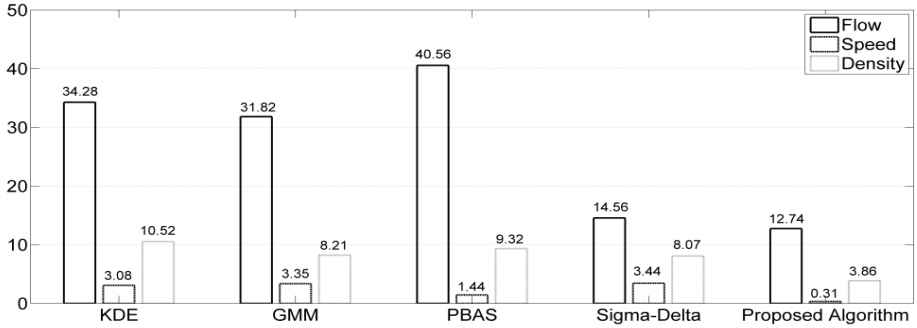
These experiments underscore difficult conditions where earlier many image processing algorithms ceases: congestion, shadows linking vehicles together, and camera movement. The real video sequence includes shadows, both from vehicles and from wayside objects like building. Approximately 25% of the samples included shadow and 15% of the segments included camera movement due to high winds. Performance does not show any significant changes under different conditions. The performance of the algorithms in traffic state estimation is featured in Figure 8, under two challenging conditions: (1) illumination variation with long shadows during daytime and (2) camera displacement. There are 60 sample video datasets, each being 1 minute in time, containing 6800 vehicles. Note that the evening peak starts during the video sequence and approximately 50% of data are under light to severe congestion, and thus, frequent occlusions. Occlusion results from the variety of traffic composition or from the position of the video sensor. For traffic composition, smaller vehicles remain undetected in the FOV due to the presence of large truck or other vehicles. For solution, a minimum height of 40 feet and an angle around 45 degree with the vertical would resolve this problem for the video sensor. Table 7 elaborates on the performance of different algorithms in traffic state measurement.



(a)



(b)



(c)

Figure 9 ((a)-(c)): (a)-(b) Scatter plot of the algorithm output versus manually extracted ground truth data for all the algorithms for speed and flow; and (c) (%) RMSE value for speed, flow and density

Table 7: Performance description for traffic state measurement

Method	Flow	Speed	Density
KDE	Exhibits high RMSE value	Exhibits better performance compared to Sigma-Delta	Exhibits very high RMSE value
GMM	Exhibits better performance compared to KDE	Exhibits very high RMSE value compared to other methods	Exhibits better performance compared to KDE

PBAS	Exhibits very high RMSE value compared to other methods	Shows comparatively better result than KDE	Exhibits better performance compared to KDE
Sigma-Delta	Exhibits better performance compared to GMM	Exhibits very high RMSE value	Exhibits better performance compared to KDE, GMM, and PBAS
Proposed Algorithm	Exhibits better performance compared to all other methods	Exhibits better performance compared to all other methods	Exhibits better performance compared to all other methods

7.3.4 Relationship of Traffic State Parameters and Evaluation Parameters

The relationship between traffic state parameters and evaluation parameters is established to provide a guideline to nominate a particular image-processing algorithm suitable for traffic application. Two particular parameters such as False Positive Rate (FPR), and False Negative Rate (FNR) are selected to establish the relationship because they form the basic measure of evaluation for any image processing algorithm. FPR and FNR both affects flow measurement. When FPR increases, the estimated flow will also increase from the actual flow. Whereas, the flow will decrease from the actual flow when FNR increases. Table 8 underscores the relationship between traffic state parameters and evaluation parameters.

Table 8: Relationship between Traffic State Parameters and Evaluation Parameters

Traffic State Parameter	Evaluation Parameter	
	FPR	**FNR**
Flow	+	–
Density	+	–
Speed	–	–

+ = Increase
– = Decrease

The FPR and FNR have the same effect over density as flow. However, FNR and FPR have substantial effect on speed because the FNR rate increases non-linearly at stationary condition (Figure 7(c)-(d)). This increase marks a phenomenon in which stationary traffic is absorbed into the background over time. Thus, the presence of traffic decreases with time. Conversely, when FPR increases the aggregate speed of the vehicles decreases. From the derived relationship between traffic state parameters and evaluation parameters it can be stated that an algorithm for which FPR and FNR are minimum, should be adopted for traffic detection.

9. Conclusion

This research emphasizes on a novel traffic detection algorithm. The developed algorithm follows deterministic approach rather being probabilistic. The deterministic relationship among the algorithm parameters aids in defining the problem and developing the exact solution. Moreover, such approach also reduces the number of parameters involved in the application process, thus making it simple to deal with variety of challenges. The developed algorithm uses only two different parameters: (1) static background; and (2) threshold parameter. The first one initializes the background for background subtraction model and applies correction over the frames to be analyzed. The latter parameter decides whether the pixel belongs to traffic.

The algorithm achieves high accuracy of detection through small computational effort. Specifically, it achieves overall classification accuracy (>95%) with very small computational resource (500 frames per second) as compared to the most efficient algorithms identified in the recent study by Sobral and Vacavant (2014). Another important feature of the algorithm is that it detects traffic in three prevalent conditions such as stationary, slowly moving, and fast moving, having very little setbacks compared to Vargas et al. (2010).

The innovation of the algorithm involves the use of transparency effect for frame correction to deal with challenges affecting the accuracy of traffic detection such as illumination variation, shadow, and camera displacement. The qualitative analysis shows that compared to others the proposed algorithm can detect vehicles accurately and suffers least amount of false positive and false negative, under different traffic conditions. The quantitative analysis of the algorithm shows stable Precision-Recall Relationship for both moving and stationary vehicle. The precision, recall, and percent correct classification values are 0.935, 0.906, and 0.992 respectively.

The RMSE values computed from ground truth and estimated traffic parameters show that the proposed algorithm outperforms others. While estimating the traffic parameters, a set of equations were developed which can be readily adopted with other detection algorithms. Most of the algorithms elaborated in literature focuses only on the image processing for object detection, making it difficult to select a particular algorithm to be useful in traffic application. Such application although in highlighted in a survey approach by Buch et al. (2011), does not provide a generic guideline based on which an algorithm can be selected. Thus, correlations between traffic parameters and algorithm evaluation parameters (False Positive Rate, and False Negative Rate) are established to determine the applicability of any image-processing algorithm for traffic state estimation. The increment or

decrement of evaluation parameters marks the characteristics of an algorithm, which dictates whether the algorithm is suitable for traffic state estimation.

For background changing with time, the proposed algorithm can be adjusted for day to night transition sequence by replacing the static background parameter. Moreover, there is a certain research scope to estimate the traffic parameters at nighttime situation by extending state estimation equations developed in this study.

10. Acknowledgements

This research work is supported by the Committee for Advanced Studies and Research (CASR) (Grant No. 69) of Bangladesh University of Engineering and Technology (BUET). The contents of this paper reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein.

References

Bouwmans, T., 2014. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, *11*, 31-66.

Buch, N., Velastin, S., Orwell, J., 2011. A review of computer vision techniques for the analysis of urban traffic. *Intelligent Transportation Systems, IEEE Transactions on*, *12*(3), 920-939.

Coifman, B., Beymer, D., McLauchlan, P., Malik, J., 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. Transportation Research Part C: Emerging Technologies, 6(4), 271-288.

Davis, J., Goadrich, M., 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240). ACM.

Elgammal, A., Harwood, D., Davis, L., 2000. Non-parametric model for background subtraction. In *Computer Vision—ECCV 2000* (pp. 751-767). Springer Berlin Heidelberg.

Guo, L., Ge, P. S., Zhang, M. H., Li, L. H., Zhao, Y. B., 2012. Pedestrian detection for intelligent transportation systems combining AdaBoost algorithm and support vector machine. *Expert Systems with Applications*, *39*(4), 4274-4286.

Gupte, S., Masoud, O., Martin, R. F., Papanikolopoulos, N. P., 2002. Detection and classification of vehicles. *Intelligent Transportation Systems, IEEE Transactions on*, *3*(1), 37-47.

Hsu, W. L., Liao, H. Y., Jeng, B. S., Fan, K. C. (2004). Real-time traffic parameter extraction using entropy. *IEE Proceedings-Vision, Image and Signal Processing*, *151*(3), 194-202.

Ilyas, A., Scuturici, M., Miguet, S., 2009. Real time foreground-background segmentation using a modified codebook model. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on* (pp. 454-459). IEEE.

Karasulu, B., Korukoglu, S., 2012. Moving object detection and tracking by using annealed background subtraction method in videos: Performance optimization. *Expert systems with applications*, *39*(1), 33-43.

Lai, J. C., Huang, S. S., Tseng, C. C., 2010. Image-based vehicle tracking and classification on the highway. In *Green Circuits and Systems (ICGCS), 2010 International Conference on* (pp. 666-670). IEEE.

Mandellos, N. A., Keramitsoglou, I., Kiranoudis, C. T., 2011. A background subtraction algorithm for detecting and tracking vehicles. *Expert Systems with Applications*, *38*(3), 1619-1631.

Manzanera, A., Richefeu, J. C., 2007. A new motion detection algorithm based on Σ–Δ background estimation. *Pattern Recognition Letters*, *28*(3), 320-328.

Mao, H., Ye, C., Song, M., Bu, J., Li, N. (2009, October). Viewpoint independent vehicle speed estimation from uncalibrated traffic surveillance cameras. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* (pp. 4920-4925). IEEE.

Pan, X., Guo, Y., Men, A., 2010. Traffic surveillance system for vehicle flow detection. In *Computer Modeling and Simulation, 2010. ICCMS'10. Second International Conference on* (Vol. 1, pp. 314-318). IEEE.

Pletzer, F., Tusch, R., Böszörmenyi, L., Rinner, B., 2012. Robust traffic state estimation on smart cameras. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on* (pp. 434-439). IEEE.

Rosin, P. L., Ioannidis, E., 2003. Evaluation of global image thresholding for change detection. *Pattern Recognition Letters*, *24*(14), 2345-2356.

Shen, Y., 2008. The research and applications of visual tracking based on mean shift (Ph.D. thesis). Jiangsu University of Science and Technology.

Sobral, A., 2014. BGS Library. Available at: https://github.com/andrewssobral/bgslibrary. Accessed June., 2015.

Sobral, A., Vacavant, A., 2014. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, *122*, 4-21.

Stauffer, C., Grimson, W. E. L., 1999. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* (Vol. 2). IEEE.

Takeuchi, A., Mita, S., McAllester, D. (2010, June). On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods. In *Intelligent Vehicles Symposium (IV), 2010 IEEE* (pp. 1014-1021). IEEE.

Vargas, M., Milla, J. M., Toral, S. L., Barrero, F., (2010). An enhanced background estimation algorithm for vehicle detection in urban traffic scenes. *Vehicular Technology, IEEE Transactions on*, *59*(8), 3694-3709.

Wan, Y., Huang, Y., Buckles, B., 2014. Camera calibration and vehicle tracking: Highway traffic video analytics. Transportation Research Part C: Emerging Technologies, 44, 202-213.

White, B., Shah, M., 2007. Automatically tuning background subtraction parameters using particle swarm optimization. In *Multimedia and Expo, 2007 IEEE International Conference on* (pp. 1826-1829). IEEE.

Zivkovic, Z., van der Heijden, F., 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, *27*(7), 773-780.

Zou, Y., Shi, G., Shi, H., Wang, Y., 2009. Image sequences based traffic incident detection for signaled intersections using HMM. In Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on (Vol. 1, pp. 257-261). IEEE.