This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Rapid and Robust Background Modeling Technique for Low-Cost Road Traffic Surveillance Systems

Kratika Garg, Nirmala Ramakrishnan, Alok Prakash, and Thambipillai Srikanthan

***Abstract*— Fast and accurate detection of vehicles on road traffic scenes captured by traffic surveillance cameras, is essential for large-scale deployment of automated traffic surveillance systems. The state-of-the-art techniques typically employ background modeling for low-complexity foreground detection. However, this is a challenging problem as these methods need to be robust to varying road scene conditions (such as illumination changes, camera jitter, stationary vehicles, and heavy traffic) leading to huge computation cost. In this paper, we propose a highly accurate yet low-complexity foreground (i.e., vehicle) detection technique, which can effectively deal with the varying road scene conditions, and generate accurate pixel-level foreground masks in real-time. We propose a novel robust block-based feature suitable for modeling road background and detecting vehicles as foreground, and employ Bayesian probabilistic modeling on these features. The experimental evaluations on widely used traffic datasets demonstrate that the proposed method can achieve comparable accuracy to the existing state-of-the-art techniques but at a much higher processing frame rate (40x speedup over PAWCS). The real-time performance of the proposed system has also been demonstrated by implementing it on a low-cost embedded platform, Odroid XU-4, that still achieves a frame rate of over 80 frames/s, thereby enabling the real-time detection of foreground objects in road scenes.**

***Index Terms*— Traffic surveillance, background modeling, Bayesian framework.**

I. Introduction

MODERN smart cities require traffic surveillance systems to ensure smooth traffic flow and optimized road usage for commuters. Among the various surveillance sensors deployed across cities, data from CCTV cameras provide the richest information that can be used for a myriad of applications, for example, traffic flow estimation, incident detection, law enforcement, and behavior understanding. There are two ways of automating such systems - central server computing and on-board computing. Central server computing systems for distributed traffic surveillance cameras necessitates huge bandwidth requirement for sending high-resolution video data that makes large-scale deployment of these camera-based sensors an expensive, and therefore, unscalable proposition. On the contrary, on-board computing systems only communicate events of interest for traffic surveillance systems to the central command station, thereby significantly alleviating the bandwidth requirements. Additionally, on-board local processing also allows for localized decision making on the road network in real time. For example, smart cameras at road intersections can detect high traffic density in a particular direction and activate the corresponding green traffic signal for a prolonged period, if possible. A set of such sensors could also work collaboratively to provide for a 'green wave' on roads in specific directions. Several attempts have been made to improve these systems, but they are usually too computationally complex to achieve real-time performance on low-cost embedded platforms, which makes them costly for mass deployment. Therefore, affordable on-board processing solutions are required for large-scale automation of video-based traffic surveillance systems. Even after years of research in this field, development of low cost yet robust solutions still remains a challenging problem [1].

Detection of vehicles on road scenes is the first step in a vision-based traffic surveillance system. Background modeling is a common approach to achieve this, in which the moving vehicles are considered the *foreground* and stationary regions of the scene, i.e., road, are considered the *background*. These techniques generate a foreground mask where each pixel is classified as background or foreground. Background modeling has been widely researched over the years, however, it still remains an open problem due to the large number of challenges associated with it [2], e.g. dealing with the variety of changes like illumination changes, weather changes, camera jitter, image noise; stationary foreground objects; similar foreground and background intensities (i.e. camouflage); uniform foreground regions (i.e. foreground aperture), etc.

Among the several techniques that have been proposed for background modeling, most of them are generic in nature and aim to generate perfect pixel level foreground masks for any scene, often sacrificing the computational efficiency of the algorithms to achieve robustness [3], [4]. While the application-agnostic design of these high complexity background modeling techniques allows them to be used for a variety of situations, including traffic-surveillance applications, they are not suitable for implementation on low cost embedded platforms. Existing work has also proposed compute-efficient techniques that limit the area of foreground detection to suit a particular traffic-surveillance application such as traffic density

Manuscript received February 22, 2018; revised August 6, 2018, November 27, 2018 and February 21, 2019; accepted April 14, 2019. This work was supported in part by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) Programme with the Technical University of Munich at TUM-CREATE. The Associate Editor for this paper was H. Huang. *(Corresponding author: Kratika Garg.)*

The authors are with the School of Computer Science and Engineering Nanyang Technological University, Singapore 639798 (e-mail: kratika001@e.ntu.edu.sg; nirmalar@ntu.edu.sg; alok@ntu.edu.sg; astsrikan@ntu.edu.sg).

Digital Object Identifier 10.1109/TITS.2019.2917560

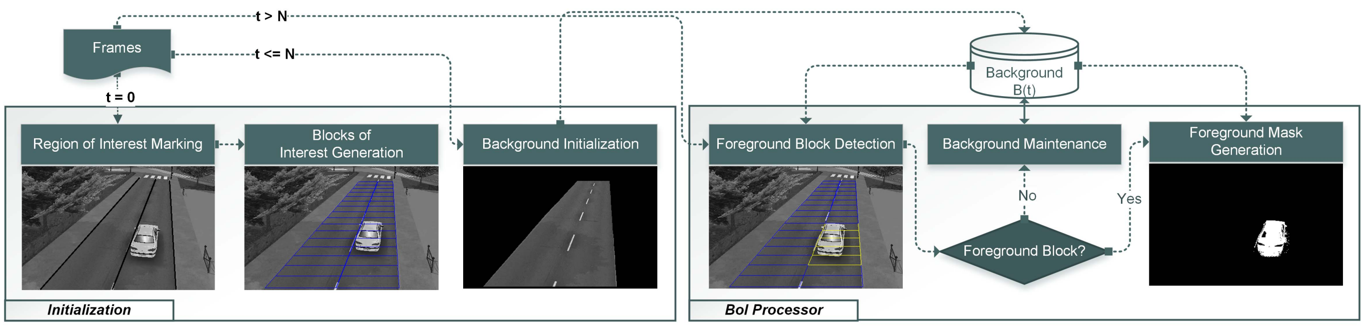


Fig. 1. Overall flow of the proposed approach. Yellow/blue blocks represent foreground/background blocks. (Best viewed in color).

estimation [5], [6] and incident detection [7], [8]. However, the information generated by these techniques is limited to these applications and therefore not usable for higher level tasks like vehicle tracking for behavior analysis.

For a real-world traffic surveillance sensor, a balance between application specific and generalized techniques is necessary. We ideally require a robust yet low-complexity foreground object detection technique that can be deployed to extract the information about traffic flow and traffic incidents in real-time without limiting the amount of information required for the tracking and classification of vehicles needed for long-term/higher level analysis. In this paper, we present such a background modeling technique that can be effectively used to detect foreground masks for road scenes in real-time even on low cost embedded devices. Similar to other application-specific techniques, we also leverage application-specific characteristics to reduce the computational complexity of the proposed technique. However, unlike them, we take overall traffic surveillance as the application at hand, where all vehicles on the road are treated as foreground and need to be detected. We limit our region of interest to road lanes and use the significant size of the foreground objects, i.e., vehicles, to move to a low-complexity block-level processing technique. In addition to that, we achieve higher resilience to environmental changes by harnessing the spatial relationship between pixels compared to pixel-based techniques. In this paper, we employ a block processing method to initialize and maintain a background image, and classify each block as background or foreground, which enables the generation of pixel-level foreground masks. Fig. 1 shows the complete flow of the proposed approach.

The rest of the paper is organized as follows. In Section II, the most popular approaches for background modeling are presented and the main contributions of our approach are discussed. Section III explains the proposed approach in detail. In Section IV, the proposed technique has been evaluated and compared with other state-of-the-art methods. Finally in Section V, we draw our conclusions.

II. Related Work

In the literature, several techniques for processing traffic surveillance videos have been proposed that focus on a specific problem. For example, vehicle counting techniques [5], [6] based on virtual loop detectors detect and count vehicles at the start of each lane, and illegal parking detection techniques [7], [8] monitor the restricted zones (double solid lines at the side of the lanes) for parked vehicles. Although these methods are computationally lean, their output cannot be directly used for other higher level tasks such as vehicle tracking.

On the other hand, generalized background modeling techniques have been employed for detecting foreground objects of interest, i.e., vehicles on road scenes [9], [10]. It involves construction and maintenance of a background model/image, followed by a background or foreground classification resulting in pixel-level foreground masks, which can be used by higher level surveillance tasks. An excellent survey of the techniques is presented in [2].

Most background modeling methods rely on pixel-level modeling with pixel intensity as the feature. Gaussian Mixture Model (GMM) [11] is a classic and widely used parametric statistical model, which employs multiple Gaussian models to represent the variations in background pixel intensities. Several adaptive variants [12], [13] of this technique have been proposed in the literature since then. Non-parametric methods [14], [15] based on kernel density estimation (KDE) [16] do not make any assumptions about the underlying model and rely solely on the observed background intensities. Instead of using explicit models, some pixel-based approaches [17], [18] maintain a cache of the intensity values as the background model. However, all these pixel-based techniques ignore the spatial relationship of the pixels with their neighborhood. This leads to missed detections when the foreground objects have a similar color to the background and false detections in the presence of camera jitter [19].

Recently, deep convolutional neural networks have been employed to learn the spatial features such as color and texture for background modeling, which can generate pixel-level foreground masks [20]–[23]. However as described in [23], [24] the accuracy of these methods relies heavily on the quality of the ground truth segmentation masks in the training datasets and how well they represent all possible variations in the background for that scene. Generating such diverse datasets with accurate foreground masks is non-trivial. Also, they are not able to achieve real-time performance (30 frames/second) even on high-end GPUs [23].

As both the spatial features as well as temporal behavior is crucial for background modeling, recent state-of-the-art techniques that do not employ deep-learning incorporate spatiotemporal awareness in their pixel-level decision-making process to improve robustness, e.g. [3] and [4] model features like Local Binary Similarity Pattern (LBSP) instead of pixel intensities [25], [26]. These techniques achieve better performance than other state-of-the-art techniques; however, this comes at the cost of higher computational complexity.

On analyzing the state-of-the-art techniques, we observed that most of them aim to develop a generalized background modeling method that can work in all scenarios. In order to achieve robustness to all the challenges associated with background modeling, these techniques often end up requiring high-end computational platforms (e.g., GPUs) to achieve real-time performance. However, if we focus only on background modeling for traffic surveillance applications, there are multiple opportunities to achieve robust solutions without sacrificing the run-time performance. *First, for most highway and urban traffic scenes, the main objects of interest are found on the road.* This can help reduce computational complexity by processing only the road regions. Also, the background modeling needs to cater to only the dynamic changes happening to the road surface that are less diverse. *Second, for our application, the size of foreground objects of interest (i.e., the vehicles) is known.* This provides the opportunity to model "larger" regions of the image instead of modeling each pixel, reducing the complexity of background modeling. In addition to that, this allows us to employ a strict background update strategy that never assimilates persistent foreground objects like slow moving/ stationary vehicles, unlike pixel-based methods. *Third, unlike other foreground objects (e.g., pedestrians), the trajectories of vehicles are reasonably uniform and predictable.* This enables the use of temporal continuity of vehicles to achieve better performance. *Fourth, since we are dealing with similar foreground objects (i.e., vehicles) that repeatedly appear in the scene, properties of the foreground objects can also be learned over time.* This helps in improving the robustness to "once off" changes in the background, e.g., a sudden change in illumination due to a cloud cover, which could otherwise lead to false foreground detections.

In this paper, we propose a novel block-based background modeling and foreground detection technique for traffic scenes, that is robust to illumination changes, camera jitter, image noise and deals with heavy traffic and stopped vehicles effectively. The proposed block-level modeling enables the generation of accurate pixel-level foreground masks, without the need for any other post-processing. The main contributions of our proposed approach are:

1) Selection of each road lane as the region of interest (RoI), and its further division into blocks of interest (BoIs), large enough for vehicle detection.
2) A novel spatial feature, *BoI variance*, that is invariant to illumination changes, camera jitter, and noise.
3) Two separate probability models for background and foreground that learn the relative shift in the aggregate/spatial feature (i.e., *BoI variance difference*), and a Bayesian probabilistic framework that combines the two models.
4) Incorporation of prior information about traffic flow into the Bayesian framework for improving robustness and its usage in isolating and dealing with missed detections, leading to improved foreground detection as well as background updates.

III. Proposed Approach

In Fig. 1, we present our proposed block-based fully-adaptive foreground detection method. During *Initialization*, each lane to be monitored on the road scene is marked as a region of interest (RoI). Each lane RoI is then divided into blocks of interest (BoI) that allows the detection of the smallest vehicle on the road. The background is then initialized for each of the BoIs. (Additional details on the initialization phase are provided in our earlier work on traffic density estimation [27] and incident detection [28].) Subsequently, each incoming frame is processed by the *BoI Processor*, which is applied to each BoI, to detect the foreground, update the background and generate the corresponding pixel-wise foreground mask. Significant improvements have been made over our previous work on block-based traffic incident detection technique [28] in which the foreground blocks were detected using the block variance difference and foreground pixel ratio. In this paper, we have moved towards a statistical approach to achieve an adaptive classification framework by modeling the block variance difference for background and foreground.

In this section, we first describe the feature used to represent each BoI, followed by the background initialization strategy. We then present the proposed models for foreground and background classification for each BoI. Next, the strategy for background maintenance is presented. We then provide detailed considerations for the initialization and updates of the proposed models. Finally, the foreground mask generation criterion is presented.

A. BoI Variance as a Feature

For a block-based background modeling technique, the feature representing a BoI should be invariant to the dynamic changes that happen on the road background. At the same time, it should facilitate the detection of vehicle foregrounds. To this end, we propose the block intensity variance as the block-level feature representing each BoI. For each BoI b_i in the frame I_t, the *BoI variance* $\sigma^2(b_i)$ is computed as:

$$\sigma^2(b_i) = \frac{1}{N^{b_i}} \sum_{x_{min}}^{x_{max}} \sum_{y_{min}}^{y_{max}} (I_t(x, y))^2 - (\mu^2(b_i))^2 \qquad (1)$$

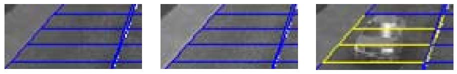
$$\mu(b_i) = \frac{1}{N^{b_i}} \sum_{x_{min}}^{x_{max}} \sum_{y_{min}}^{y_{max}} (I_t(x, y)) \qquad (2)$$

where $I_t(x, y)$ is the pixel intensity at location (x, y), $\mu(b_i)$ is the BoI mean, (x_{min}, x_{max}) and (y_{min}, y_{max}) represent the boundaries of b_i and N^{b_i} is the total number of pixels in the BoI.

We now show the invariance of *BoI variance* to the following dynamic changes:

- *Changes in Illumination:* Fig. 2 shows that individual pixel intensities experience significant change for both

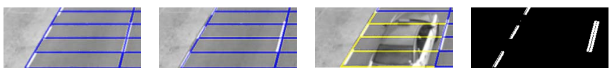
(a) (b) (c)



	(a) Cloudy	(b) Sunny	(c) Cloudy+Vehicle
BoI Mean	88.704	102.93	101.77
BoI Variance	127.88	173.51	1066.5

Fig. 2. Effect of illumination change on *BoI Mean* and *BoI Variance* on BoIs from a (a) cloudy frame (b) sunny frame and (c) cloudy frame with a vehicle.

(a) (b) (c) (d)



	(a) Frame	(b) Frame+Jitter	(c) Frame+Vehicle
BoI Variance	277.89	209.99	1812.60

Fig. 3. Effect of camera jitter on *BoI variance* using a (a) base frame (b) frame after camera jitter. Figure (d) shows the foreground mask generated for (b) by a GMM based technique [13].

illumination changes and vehicle occupancy as shown using their aggregated value, *BoI mean*. On the other hand, *BoI variance* experiences a low variation for illumination changes, as all the pixels in the BoI undergo an identical shift in their intensity. However, in the presence of a vehicle, it does experiences a higher variation.

- *Camera Jitter:* Similarly, Fig. 3 illustrates that pixel-based techniques lead to false detections in the presence if camera jitter as shown in the mask generated using GMM [13] in Fig. 3d. On the other hand, small camera movements only lead to a slight change in the *BoI variance*.
- *Image Noise:* Image noises are largely filtered out by the average during the calculation of variance, while techniques based on pixel intensity have to cater to them explicitly in their modeling process [29].

Additionally, the computational complexity of the *BoI variance* calculation is also low, which can lead to significant savings in comparison to per pixel modeling. In the next subsection, we describe the use of this block feature for initializing a background image.

B. Background Initialization

Initializing a representative background image which does not include any foreground objects is a crucial step in background modeling techniques [30]. In this paper, we propose a block-level background initialization technique for free-flowing traffic conditions. We follow a progressive strategy that initializes background for the BoIs which meet the required criterion and continues until the background is obtained for all BoIs [2].

The criterion used for initializing the background for each BoI is the stability of its variance (σ^2) across frames. When no vehicle passes through a BoI, the variance is expected to remain stable across frames. In order to check the stability of the *BoI variance* across frames, the *variance of the variance values (VoV)* of a BoI from several frames is calculated.

Let the background and current frame for a BoI b_i, at time t, be defined as $B_t^{b_i}$ and $I_t^{b_i}$. For each BoI b_i, a circular buffer is constructed which stores the variance values of N most recent frames. Once the buffer is full, the variance of the variance values stored VoV^{b_i} is calculated as:

$$VoV^{b_i} = \sigma^2(\sigma^2(I_{t-N}^{b_i}) : \sigma^2(I_t^{b_i})) \tag{3}$$

If $VoV^{b_i} < T_B$, where T_B is a pre-defined threshold, the pixel intensities of the BoI from the current frame are used to initialize the background image, i.e. $B_{t+1}^{b_i} = I_t^{b_i}$.

In order to obtain the optimum value of N and T_B, we performed extensive simulations. N was set to 6 for a frame rate of 30 fps, corresponding to a duration of 0.2s in which the stability criterion needs to be met, for the background to be initialized. For free-flowing traffic, this leads to fast background initialization whenever the road is seen. For a lower frame rate, N also has to be reduced (e.g., N = 4 was used for 10-15 fps) to consider the same duration. For slow-moving traffic, using a higher value of N may be beneficial to ensure accurate background initialization. T_B was empirically set to 100 based on the observation that VoV values changed significantly (> 500) in the presence of vehicles in the past N frames. Thus, a threshold of 100 was chosen to ensure that the background was only initialized when variance values experienced low variation across time.

This process is repeated until the background is constructed for all BoIs. An example of the constructed background for the BMC dataset [31] can be seen in Fig. 1. After background initialization, each BoI is sent to the BoI processor in the subsequent frames, where they are classified as background/foreground, their background is updated, and their foreground masks are generated. We first discuss the classification process in detail in the next subsection.

C. Model for Background/Foreground Classification

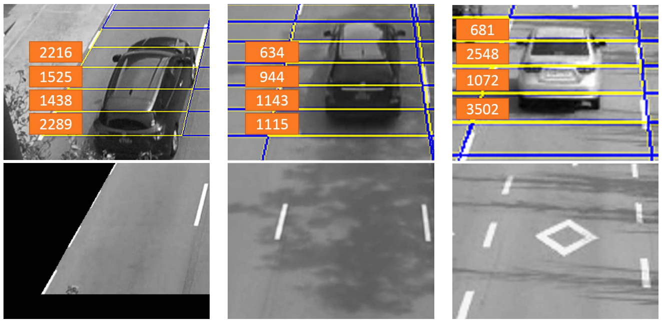
In this section, we present the Bayesian framework used to model the background and foreground.

1) BoI Variance Difference as the feature: As described in Section III-A, the proposed *BoI variance* undergoes significant change when a vehicle is present in the BoI. Thus, we use the change in *BoI variance* relative to the background, i.e., variance difference as the key feature for the classification process. For a BoI b_i, the variance difference ΔV^{b_i} is calculated as follows:

$$\Delta V^{b_i} = abs(\sigma^2(B_t^{b_i}) - \sigma^2(I_t^{b_i})) \tag{4}$$

In Fig. 4, we show that the selected feature *BoI variance* enables robust vehicle detection in the following scenarios:

- *Low-textured road:* Blocks encapsulating low-textured parts of a road experience low variance and therefore, there is a significant change in the *BoI variance* ($> 10^3$) when the block is occupied by a vehicle.
- *Static shadows:* When a block is covered by static shadows of roadside buildings, trees or other structures, its



(a) (b) (c)

Fig. 4. Change in variance with respect to the current background (Row 2) for three scenarios: (a) Vehicle on smooth road (b) Vehicle on road with static shadows (c) Vehicle on road with road markings. (Row 1: Yellow: Foreground, Blue : Background, Orange Box: Variance Difference).

BoI variance increases. However, it still experiences a large value of ΔV^{b_i} when occupied by a vehicle due to the change in texture.

- *Road markings:* Road markings have a similar effect as the static shadows on the *BoI variance* values; thus, it also leads to high values of ΔV^{b_i} in the presence of vehicles.

The above-described cases show the robustness of *BoI variance difference* as a feature for all the common road textures encountered in traffic surveillance applications. In the next section, we present statistical models for foreground and background that employ the *BoI variance difference* ΔV^{b_i} as the feature.

2) Bayesian Modeling Using BoI Variance Difference: Changes in the background for outdoor environments can be characterized as gradual changes (e.g., slow illumination changes), dynamic changes (e.g., swaying trees) or “once-off” changes (e.g., camera movement, cloud cover). Modeling background alone is usually sufficient to deal with the gradual/dynamic changes in the environment. However, dealing with the “once-off” changes is challenging as it deviates significantly from the background model, leading to false positives. This can be alleviated by modeling the foreground in addition to the background. Therefore, we propose two separate statistical models for the background and foreground respectively, that are selectively updated after each classification. We use the Bayesian probabilistic framework to combine the models for the classification process [16], [32], [33].

Applying Bayes rule as described in [34], for a BoI at location b_i, with a variance difference v, the posterior probability of it being background $P^{b_i}(b|v)$ (i.e., having observed the *BoI variance difference* values in the past frames, when a value v is observed for the current frame, the probability that it belongs to background) can be described as:

$$P^{b_i}(b|v) = \frac{P^{b_i}(v|b)P^{b_i}(b)}{P^{b_i}(v)} \tag{5}$$

where b indicates the background, $P^{b_i}(v|b)$ is the probability of the feature v being observed as a background, $P^{b_i}(b)$ is the prior probability of the BoI belonging to the background, and $P^{b_i}(v)$ is the overall probability of the feature v being observed at the position b_i.

Similarly, the posterior probability of the feature v being foreground (f) at location b_i is:

$$P^{b_i}(f|v) = \frac{P^{b_i}(v|f)P^{b_i}(f)}{P^{b_i}(v)} \tag{6}$$

According to Bayes decision theory, in order to classify a BoI as foreground the following condition must be met $P^{b_i}(f|v) > P^{b_i}(b|v)$. Since it is a two-class problem, i.e., a particular BoI can only be classified as either background or foreground, we also know that $P^{b_i}(f|v)+P^{b_i}(b|v) = 1$. Thus, the above equation can be redefined as $P^{b_i}(f|v) > 0.5$, or:

$$\frac{P^{b_i}(v|f)P^{b_i}(f)}{P^{b_i}(v)} > 0.5 \tag{7}$$

On substituting $P^{b_i}(v)$ (*adding Eq. 5 and Eq. 6*), the above equation becomes:

$$\frac{P^{b_i}(v|f)P^{b_i}(f)}{P^{b_i}(v|b)P^{b_i}(b) + P^{b_i}(v|f)P^{b_i}(f)} > 0.5 \tag{8}$$

Assuming that the prior probability of a BoI being foreground or background is equal, we get $P^{b_i}(b) = 0.5$ and $P^{b_i}(f) = 0.5$, thus giving us:

$$\frac{P^{b_i}(v|f)}{P^{b_i}(v|b) + P^{b_i}(v|f)} > 0.5 \tag{9}$$

This simplifies the classification problem to modeling the observed *BoI variance difference* as background $P^{b_i}(v|b)$ and foreground $P^{b_i}(v|f)$ respectively. For our technique, we use a stricter threshold of 0.7 instead of 0.5 for the classification process, so that the borderline cases are not misclassified.

Intuitively, *BoI variance difference* represents the change in the current BoI with respect to the background. Thus, a low (or high) change should signify a high probability of the current BoI being background (or foreground). Building on this intuition, we modeled the background and foreground models as exponentially decreasing and increasing functions respectively. The conditional probability of a *BoI variance difference*, given it is background, was formulated as:

$$P^{b_i}(v|b) = exp(-\Delta V^{b_i}/\lambda_b)) \tag{10}$$

The conditional probability of a *BoI variance difference* given it is foreground was formulated as:

$$P^{b_i}(v|f) = (1 - exp(-\Delta V^{b_i}/\lambda_f)) \tag{11}$$

where λ_b and λ_f represent the rate of decrease and increase of the background and foreground probability distributions respectively. Fig. 5d shows the background and foreground conditional probability models for a BoI from a sunny video Fig. 5a and a foggy video Fig. 5b from the BMC Dataset [31].

We also adapt the model parameters for the probability distributions to incorporate the latest information about the scene using the following equations: If b_i is classified as background,

$$\lambda_b = (1 - \alpha_b) * \lambda_b + \alpha_b * \Delta V^{b_i} \tag{12}$$

else,

$$\lambda_f = (1 - \alpha_f) * \lambda_f + \alpha_f * \Delta V^{b_i} \tag{13}$$

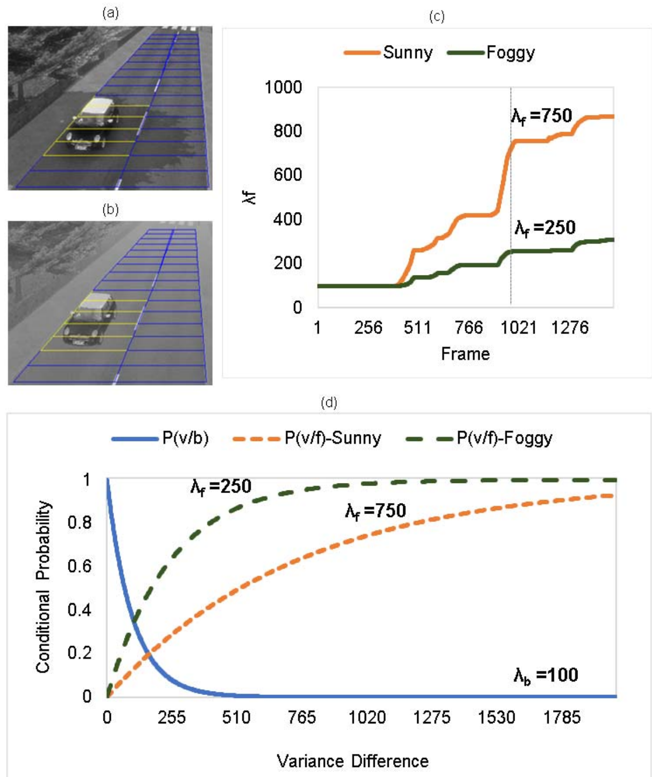


Fig. 5. Model adaptivity in (a) cloudy and (b) foggy situations from BMC dataset: (c) λ_f (rate of increase for foreground probability distribution) automatically adapts to lower/higher for foggy/sunny conditions which leads to (d) Corresponding conditional probability curves (1000th frame) where a lower/higher variance difference will be classified as foreground for foggy/sunny scenarios respectively.

where, α_b and α_f are the learning rates for the background and foreground models respectively. This enables our models to adapt themselves over time according to the current scene. As shown in Fig. 5c, λ_f automatically adapts to a higher value for a sunny scene as compared to a foggy scene. Fig. 5d shows that for a given *BoI variance difference*, under foggy conditions it will have higher probability of being classified as foreground (as compared to a sunny scene). Therefore, effectively dealing with the lower distinguish-ability of vehicles during foggy conditions. This enables the adaptive nature of the proposed Bayesian classification framework. The initialization values and the learning rates for these models are discussed later in Section III-F.

In the next two sections, we discuss the use of prior probability and block mean to further enhance robustness by detecting true foreground cases missed by the above described model.

3) Incorporating Prior Information: To further improve the robustness of our Bayesian classification framework we incorporated the vehicle trajectories as prior information. For highway/urban traffic scenario the following prior assumptions can be made:

- Vehicles enter the frame at the first/last block in a lane for outgoing/incoming traffic respectively.
- The types of trajectories for a vehicle are typically limited i.e., left/right/forward.

In Section III-C2, the prior probabilities for the appearance of background and foreground were assumed to be equal (i.e., 0.5). In this section, the prior probabilities are updated using the foreground or background classification from the previous frame. We use the following rules to set the foreground prior probability $P^{b_i}(f)$ as shown in Fig. 6a.

- *Case 1:* For the blocks that are not in the vicinity of any foreground block in the previous frames, the prior probability is reduced to 0.4.
- *Case 2:* If the block before(or after) for outgoing(or incoming) traffic was detected as foreground in the previous frame, the prior probability is increased to 0.6.
- *Case 3:* If any of the neighboring blocks or the block itself were detected as occupied in the previous frame, the prior probability is maintained at 0.5, as the chances of a vehicle turning in any direction cannot be predicted.
- *Case 4:* The first(or last) block of each lane for incoming(or outgoing) traffic respectively always has a prior probability of 0.5 as there is no prior information available for these blocks.

It is important to note that we only decrease/increase the prior probability values marginally as there are exceptions to traffic rules mentioned above e.g., pedestrians crossing the road, vehicles stopping on the road.

Fig. 6b-c shows that incorporating the prior probability into our Bayesian classification framework helps increase the overall robustness of our proposed approach by:

- Increasing the foreground probability for borderline foreground cases (i.e. $P^{b_i}(f|v)$ is between 0.5 and 0.7) using *Case 2*, thereby reducing false negatives as shown in Fig.6b.
- Decreasing the foreground probability for borderline background cases (i.e. $P^{b_i}(f|v)$ is between 0.7 and 0.8) by using *Case 1*, thereby reducing false positives as shown in Fig. 6c.
- Incorporating the prior probability in the background maintenance procedure to reduce false background updates. This is discussed in the background maintenance section i.e. Section III-D in detail.

4) Mean Model: Fig. 6d shows cases when our *BoI variance difference* based Bayesian probabilistic framework is unable to detect the foreground BoIs. This happens when a BoI is occupied by a smooth part of a vehicle (mostly heavy vehicles), and the edges are not dominant enough to incur a significant change in variance. However, in most cases, there is a change in intensity that can be exploited to detect such parts.

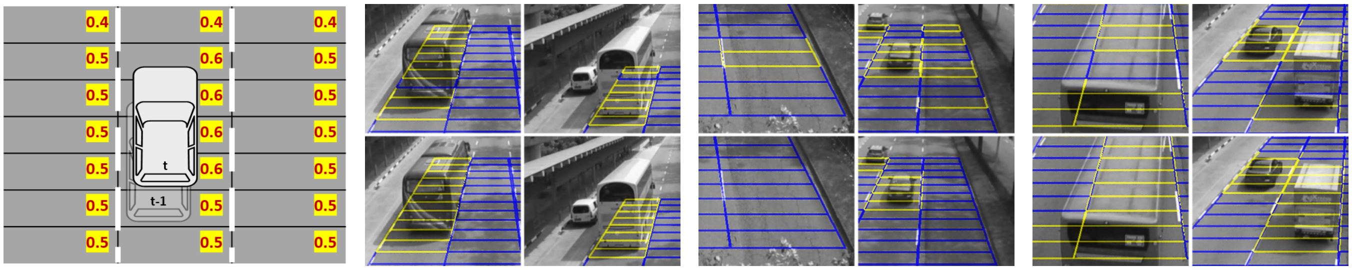
For such cases, we used *BoI mean* to distinguish them from the background. We modeled the *BoI mean* intensities using a single Gaussian model, which is defined as:

$$P_m^{b_i}(x) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp^{-\frac{x-\mu_m}{2\sigma_m^2}} \tag{14}$$

where,

$$\mu_m = (1-\alpha_m) * \mu_m + \alpha_m * \mu_t^{b_i} \tag{15}$$

$$\sigma_m^2 = (1-\alpha_m) * \sigma_m^2 + \alpha_m * (\mu_t^{b_i} - \mu_m)^2 \tag{16}$$



(a) (b) (c) (d)

Fig. 6. (a) Prior probability assignment for the current frame t based on background/foreground classification in the previous frame *(t-1)* of neighboring blocks; (b) False negative and (c) False positive detections in Row 1 corrected in Row 2 after the incorporation of prior information; (d) False negative detections (Row 1) corrected (Row 2) after the incorporation of block mean model (Best viewed in color).

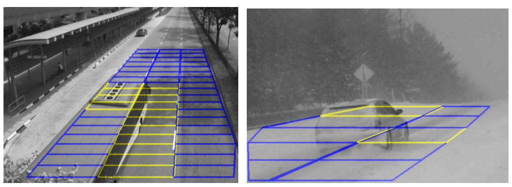


Fig. 7. False negative detections in case of complete camouflage.

$P_m^{b_i}(x)$ represents the probability that x fits the Gaussian model with a mean μ_m and variance σ_m^2 which adapts over time with a learning rate of α_m. However, as highlighted in Fig. 2 and Fig. 3, *BoI mean* is susceptible to illumination changes. Thus, we used the prior probability o isolate likely false negative detections effectively. When the foreground prior probability $P^{b_i}(f)$ of a BoI was higher than 0.5, and Bayesian classifier classified it as background block, we employed the BoI Mean model. The following detection criterion was used:

$$if(P_m^{b_i}(\mu_t^{b_i}) > P_m^{b_i}(\mu_m + 3 * \sigma_m^2)) : I_t^{b_i} = foreground \tag{17}$$

Additionally, we used the classifications from our robust Bayesian classifier to learn and adapt the *BoI mean* Gaussian background model (Details in Section III-F).

Fig. 6d shows cases where *BoI mean* helps to detect false negative cases, thus further improving the robustness of the proposed approach. It should be noted that the proposed method, would not be able to detect parts of vehicle which have similar texture and grayscale intensity as the road surface (i.e., camouflage, shown in Fig. 7) which is a challenging problem for existing state-of-the-art techniques as well. We aim to incorporate color information into our models to detect such cases in the future.

D. Background Maintenance

Background maintenance ensures that the background is kept updated and any changes to the background due to environmental variations are assimilated. Therefore, any errors in this step can lead to prolonged effects on the background or foreground classification. Thus, we follow a strict background maintenance scheme for our proposed method using the prior probability assignment shown in Fig. 6a. First, the prior probability and classification based on the Bayesian model are used to decide the BoIs that should be sent for background update. The BoIs classified as background that have a prior probability less than 0.5 (i.e., BoIs with no neighboring foreground blocks in the previous frame) are sent for background update. This eliminates the chances of a false background update due to a missed foreground detection (i.e., parts of foreground objects wrongly classified as background). After this, a stability check similar to the background initialization step in Section III-B, Eq. 2 is applied to confirm if it is a background block. Thus, for the update procedure, we follow a selective update strategy as follows:

If b_i is classified as background and $P^{b_i}(f) < 0.5$,

$$B_{t+1}^{b_i} = \begin{cases} \alpha * I_t^{b_i} + (1-\alpha) * B_t^{b_i} & VoV^{b_i} < T_B \\ B_t^{b_i} & VoV^{b_i} \geq T_B \end{cases} \tag{18}$$

where $B_{t+1}^{b_i}$ and $B_t^{b_i}$ are the background images for a block b_i at time $t+1$ and t respectively, α is the learning rate for background image update and T_B is the threshold on the stability criterion (described in Section III-B). Additionally the first/last blocks in each lane for outgoing/incoming traffic respectively are also sent for background update if they are classified as background by the Bayesian classifier.

These regular updates to the background, ensure that the proposed technique is adaptive to illumination changes, formation/fading of static shadows on the road, camera movements, etc. Also, this helps to achieve more accurate pixel-level foreground masks. In the next section, we present how these foreground masks are generated.

E. Foreground Mask Generation

Our foreground mask generation strategy follows a simple thresholded difference for the BoIs classified as foreground, which is represented as follows:

If $I_t^{b_i}$ is foreground,

$$M_t^{b_i}(x,y) = \begin{cases} 1 & abs(I_t^{b_i}(x,y) - B_t^{b_i}(x,y)) > T_M \\ 0 & otherwise \end{cases} \tag{19}$$

where $I_t^{b_i}$, $B_t^{b_i}$ and $M_t^{b_i}$ is the current image, background image and the corresponding foreground mask respectively for a block b_i at time t. T_M is the threshold used for the pixel intensities, it is empirically set to 30. An example of the generated foreground mask for the BMC dataset is given in Fig. 1.

F. Model Initialization and Updates

We have created three different models for our proposed approach namely variance difference based background model $P^{b_i}(f|v)$ and foreground model $P^{b_i}(b|v)$ in Section III-C2 and block mean based background model $P_m^{b_i}(x)$ in Section III-C4. In this section, we describe how the parameters for each of these models i.e. $(\lambda_f,\lambda_b,(\mu_m,\sigma_m^2))$ are adapted using $(\alpha_f,\alpha_b,\alpha_m)$.

It is to be noted that the initialization values for the model parameters do not impact the overall performance of the algorithm since the proposed method learns on-line and adapts to the given environment. For our proposed technique, λ_b and λ_f are initialized to 100. μ_m and σ_m^2 are initialized using the BoIs that are used to construct the background. For the variance difference models, there is a risk of over-fitting of the models, when non-dynamic background is seen for a long-time (*for background model*) or very distinct vehicles are observed for a long time (*for foreground model*). Therefore, we also put a minimum and maximum cut-off on the λ_b and λ_f values. The minimum values are set to the initialization values, and the maximum values are empirically set to 500 and 2000 for λ_b and λ_f respectively.

For adapting these models effectively, learning rates play an important role. For our proposed approach, α_f is set to 0.01. This slow learning rate ensures that the foreground model does not get over-fitted to a certain type of vehicle. For α_b, two learning rates are used i.e. 0.01 and 0.1. When ΔV^{b_i} is low, α_b is set to 0.01 as the model is learning no new information. However, when ΔV^{b_i} is high, α_b is set to 0.1. This helps incorporate the infrequent changes that occur in the background due to camera jitter or drastic illumination changes rapidly. This strategy helps the model to quickly adapt to dynamic changes in the background and at the same time retain this for a longer time as the smaller changes are learned at a slower learning rate. For α_m a similar strategy is used. When $\mu_t^{b_i}$ fits the model, α_m is set to 0.01, otherwise it is set to 0.1. We want to highlight that these values were set based on a standard video frame rate, i.e. (25-30 fps). For extreme scenarios, e.g., for a frame rate of less than 10 fps, these values should be increased to ensure that the models adapt effectively.

In Algorithm 1, we summarize our classification models, model updates and background maintenance in detail.

Algorithm 1 Foreground Block Detection

for *each* b_i **do**
  $P^{b_i}(v|b) = exp(-\Delta V^{b_i}/\lambda_b))$;
  $P^{b_i}(v|f) = (1 - exp(-\Delta V^{b_i}/\lambda_f))$;
  $P^{b_i}(f|v) = \frac{P^{b_i}(v|f)P^{b_i}(f)}{P^{b_i}(v|b)(1-P^{b_i}(f))+P^{b_i}(v|f)P^{b_i}(f)}$;
  $P_m^{b_i}(x) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp^{-\frac{x-\mu_m}{2\sigma_m^2}}$;
  if $(P^{b_i}(f|v) > 0.7)$ **then**
    $I_t^{b_i}$ = foreground;
    $\lambda_f = (1-\alpha_f) * \lambda_f + \alpha_f * \Delta V^{b_i}$;
  else if $P^{b_i}(f) < 0.5 ||$ *"first blocks"* **then**
    $I_t^{b_i}$ = background;
    UpdateModel();
    if $(VoV^{b_i} < T_B)$ **then**
      $B_{t+1}^{b_i} = \alpha * I_t^{b_i} + (1-\alpha) * B_t^{b_i}$
  else
    if $P^{b_i}(f) > 0.5$ && $P_m^{b_i}(\mu_t^{b_i}) > P_m^{b_i}(\mu_m + 3 * \sigma_m^2)$ **then**
      $I_t^{b_i}$ = foreground;
    else
      $I_t^{b_i}$ = background;
      UpdateModel();
    end
  end
end
Function *UpdateModel* **is**
  if $\Delta V^{b_i} < \lambda_b$ **then**
    $\alpha_b = 0.01$;
  else
    $\alpha_b = 0.1$;
  end
  $\lambda_b = (1-\alpha_b) * \lambda_b + \alpha_b * \Delta V^{b_i}$;
  if $P_m(\mu_t^{b_i}) > P_m(\mu_m + 3 * \sigma_m^2)$ **then**
    $\alpha_m = 0.01$;
  else
    $\alpha_m = 0.1$;
  end
  $\mu_m = (1-\alpha_m) * \mu_m + \alpha_m * \mu_t^{b_i}$;
  $\sigma_m^2 = (1-\alpha_m) * \sigma_m^2 + \alpha_m * (\mu_t^{b_i} - \mu_m)^2$;
end

IV. Results

In this section, we present details about the datasets used, testing setup, quantitative and qualitative evaluations of the proposed method and lastly, comparison with state-of-the-art methods in terms of accuracy and run-time performance.

A. Experimental Setup

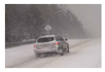
All the algorithms are executed and verified on a state-of-the-art mobile application development platform — Odroid-XU4 [35] from Hardkernel. In our experimental setup, this platform runs on Ubuntu 15.10. We run the algorithms on a single A15 core at 2 GHz. The Exynos 5422 System on Chip(SoC) on this platform is constrained to a maximum of ~10W thermal design power and is representative of a typical SoC used in low-cost, low-power embedded platforms.

B. Dataset and Evaluation Metrics

The proposed technique is more suited for traffic videos captured from a roadside traffic camera installed on an overhead bridge or a pole overlooking a street/highway to minimize occlusion. We therefore select such representative scenarios from publicly available datasets- the Background

TABLE I

DATASET USED FOR EVALUATIONS

	BMC Street [31]	**iLids** [36]	**highway** [19]	**traffic** [19]	**snowfall** [19]	**boulevard** [19]
Frame						
ROI						
Size	640*480	720*576	320*240	320*240	720*480	352*240
Climate	Sunny, Windy, Cloudy, Foggy	Sunny, Cloudy	Sunny	Sunny	Snow	Sunny
Jitter	No	Yes	No	Yes	Yes	Yes
Shadows	Yes	Yes	Yes	Yes	No	No

Models Challenge Dataset [31], iLids Parked Vehicle Detection Dataset [36] and Change Detection dataset [19]. These datasets cover a wide range of challenges such as changes in illuminations and weather, camera jitter, slow-moving traffic, and stationary vehicles. Table I provides a summary of the datasets used for the evaluation of the proposed technique.

For quantitative comparisons, we use the BMC Street Test and four videos (highway, traffic, boulevard, snowFall) from Change Detection dataset. Precision, Recall, and F-Measure are used as the evaluation metrics. The implementation of the state-of-the-art methods has been used from the BGS Library provided by [37]. In addition to accuracy, in order to compare the computational complexity of these algorithms, we measure the run-time performance (i.e., the achieved average frame rate) on the low-cost embedded platform - Odroid-XU4. In order to provide a fair comparison, all the state-of-the-art algorithms from the BGS library are modified to generate the foreground masks only for the RoIs given in Table I.

C. Quantitative Comparison With State-of-the-Art Methods

First, we compare the proposed method with the foreground detection technique from our previous work [28]. The proposed method achieved an average improvement of 1%-4% across the videos of the BMC Street Dataset [31].

We further compare the proposed approach with the state-of-the-art techniques on videos from BMC Street Dataset [31] and Change Detection Dataset [19]. Figure 8 shows that the proposed method outperforms the accuracy of a wide range of state-of-the-art techniques i.e. basic methods (DPAdaptiveMedian [40], DPPratiMediod [38], SigmaDelta [41]), texture-based methods (DPTexture [42], LBP_MRF [43]), single Gaussian models(LBSimpleGaussian [44]), Mixture of Gaussians (LBMixtureofGuassians [12], DPZivkovicAGMM [13]), Non-parametric methods (KDE [15], IndependentMultimodal [45], ViBe [17]), Fuzzy-based methods (T2FGMM [46], LBFuzzyGaussian [47], FuzzySugenoIntegral [48], FuzzyChoquetIntegral [49]) neuro-fuzzy methods (LBFuzzyAdaptiveSOM [50]) and techniques combining multiple features (MultiCue [51], MultiLayer [39]). It is noteworthy that the proposed approach which solely relies on block-level

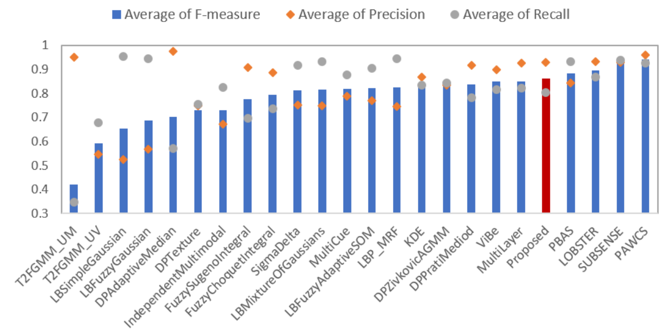


Fig. 8. Quantitative comparison with state-of-the-art techniques.

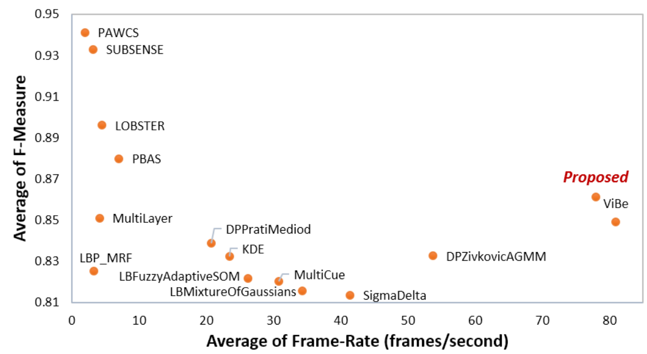


Fig. 9. Accuracy vs Complexity trade-off comparison.

processing achieves better pixel-level accuracy than state-of-the-art pixel-based techniques.

Furthermore, we also compare run-time performance (the achieved frame rate *on Odroid-XU4*) of the proposed method with the other state-of-the-art techniques that achieve higher or comparable accuracy to the proposed method. Fig. 9 presents the trade-off between accuracy (F-Measure) and computational complexity (frame-rate) for the selected state-of-the-art background modeling methods. The proposed algorithm is over 40 times faster than the best performing state-of-the-art technique, PAWCS [3]. It is clear that most of the high-performance state-of-the-art techniques like PAWCS [3], SUBSENSE [4], LOBSTER [52] PBAS [18], and MultiLayer [39] are not able to achieve a processing frame rate of even 10 frames/second on the low-cost embedded platform - Odroid XU4. Hence, these methods are unsuitable for low-cost, scalable and robust traffic surveillance solutions that can be deployed at a large scale as envisioned in this work.

Vibe [17], DPPratiMediod [38], DPZivkovicAGMM [13] and KDE [15] performed relatively well both on accuracy and run-time performance. However, our method achieves better accuracy than all these techniques. Additionally, it has significantly lower computational complexity when compared to the DPPratiMediod [38], DPZivkovicAGMM [13] and KDE [15]. Vibe [17] is the only method that achieves similar computational complexity as the proposed work. However, as shown in the following subsection, this method suffers significantly in challenging scenarios such as camera jitter and slow-moving traffic.

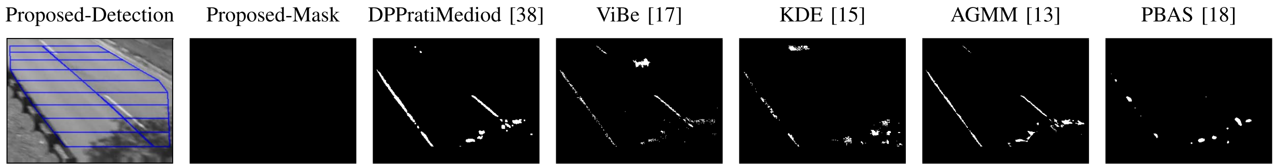


Fig. 10. Effect of camera jitter: A frame from the *traffic* video from CDNet dataset in which the camera jitters has been used.

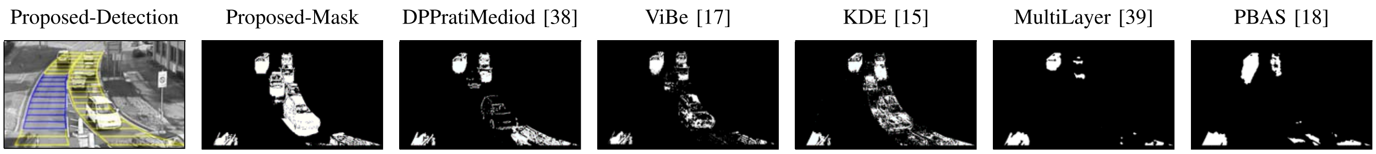


Fig. 11. Effect of slow moving heavy traffic: A frame from iLids Hard video in which vehicles were moving slowly has been used.

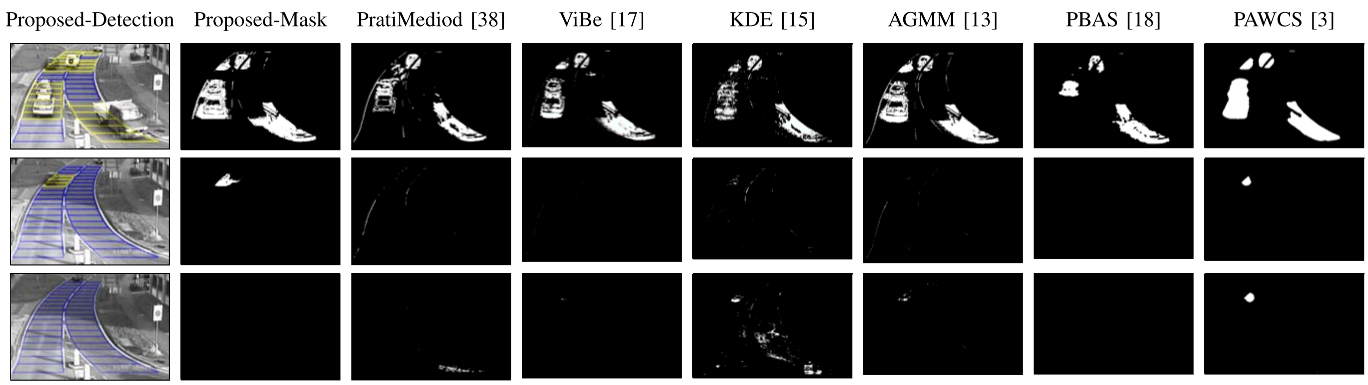


Fig. 12. Effect of stationary foreground objects: Three frames from the *iLids Medium* video have been used. Row 1- Black Vehicle in the left lane became stationary. Row 2- Vehicle remained stationary for 60 seconds, Row 3- Vehicle moved away.

Fig. 9 shows that we achieve a processing frame rate of 80 frames/second that surpasses the requirement for real-time performance. The high frame rate achieved by our method provides the opportunity to perform additional steps like vehicle localization/tracking required for higher-level analysis in real-time, even on a low cost embedded solutions, such as the Odroid XU4 platform, used in this work.

In the next subsection, we additionally present qualitative comparisons of the proposed method in diverse challenging scenarios on real traffic videos.

D. Qualitative Comparison With State-of-the-Art Methods

Camera jitter, slow-moving heavy traffic, and stationary vehicles on the road are common challenges in processing traffic surveillance videos. In this subsection, we provide qualitative comparisons of our proposed method with the existing low complexity state-of-the-art techniques identified in the previous section, in these challenging scenarios. It should be noted that moving shadows that accompany the vehicles are treated as foreground, as in existing background modeling techniques [2].

1) Camera Jitter: Traffic surveillance cameras face camera jitter during windy conditions. In Fig. 10, it can be seen that unlike the proposed method, all other low-complexity methods and even the highly accurate Multilayer [39] technique suffers due to camera movement.

We achieved immunity to camera jitter due to two important reasons: (1) We use *BoI variance* as the feature, which is invariant to slight camera movements as described in Section III-A. (2) We model the foreground variance difference in addition to the background, which provides additional information about how much change should be considered a foreground as described in Section III-C2. This helps us to effectively classify the “once-off” or “intermittent” changes as background, unlike pixel-based techniques.

2) Slow-Moving Heavy Traffic: Slow-moving heavy traffic needs to be detected and reported for traffic surveillance applications. Figure 11 shows that pixel-based techniques like ViBe [17], DPPratiMediod [38], KDE [15] and even PBAS [18], assimilate slow-moving foreground pixels into the background, thus resulting in a large number of false negatives. Unlike these techniques, it can be seen that the proposed method is able to detect slow-moving vehicles effectively.

Pixel-based techniques make local decisions, thus being unable to differentiate between persistent foreground pixels due to slow-moving heavy traffic and sudden “once-off” changes in the background. Therefore, they employ a background maintenance strategy where all persistent changes are assimilated into the background. On the other hand, our robust block-level classification framework ensures that “once-off” background changes are not detected as foreground as highlighted above. Thus, we follow a strict background update strategy which does not assimilate persistent foreground pixels thus enabling effective detection of slow-moving vehicles.

3) Stationary Foreground Objects: For traffic surveillance cameras, stationary foreground objects (like illegally parked vehicles, vehicle breakdowns, accidents) are objects of interest that need to be detected.

In Fig. 12, it can be seen that most of the techniques absorb the stationary foreground objects into their background model due to the same reasons mentioned in the previous section. Most pixel based methods only concentrate on getting rid of 'ghosts' from the stationary foreground objects once they have moved, which is being achieved by the existing techniques. Surprisingly, PAWCS [3], which performs robustly in all scenarios, and detects the stationary foreground object as well, leaves a ghost when the vehicle has moved. Fig. 12 shows that we are able to detect the stationary objects on the road and include them in the foreground mask without leaving any ghost when the vehicle moves.

V. CONCLUSION

In this paper, we introduced a novel low-complexity yet robust block-based technique for detecting foreground objects for traffic scenes. The proposed methods include a BoI based background initialization, maintenance, and background or foreground block classification technique, which generates accurate pixel-level foreground masks. It achieves robustness to changes like *illumination changes, weather conditions, camera jitter, image noise* and effectively deals with traffic situations like *stationary foreground objects, slow-moving heavy traffic*. This can be attributed to our robust block feature, i.e., *BoI variance* and our highly adaptive classification framework that detects foreground effectively. Additionally, block-based processing significantly collapses the computational complexity of the proposed approach. Experimental results on a real, low-cost, embedded Odroid-XU4 platform confirm that the proposed method achieves comparable accuracy in real-time to significantly more complex current state-of-the-art techniques.

REFERENCES

[1] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.
[2] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vols. 11–12, pp. 31–66, May 2014.
[3] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Universal background subtraction using word consensus models," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4768–4781, Oct. 2016.
[4] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.
[5] Y. Xia, X. Shi, G. Song, Q. Geng, and Y. Liu, "Towards improving quality of video-based vehicle counting method for traffic flow estimation," *Signal Process.*, vol. 120, pp. 672–681, Mar. 2016.
[6] J. I. Engel, J. Martín, and R. Barco, "A low-complexity vision-based system for real-time traffic monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1279–1288, May 2017.
[7] J. T. Lee, M. S. Ryoo, M. Riley, and J. K. Aggarwal, "Real-time illegal parking detection in outdoor environments using 1-D transformation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1014–1024, Jul. 2009.
[8] C.-M. Pun and C. Lin, "A real-time detector for parked vehicles based on hybrid background modeling," *J. Vis. Commun. Image Represent.*, vol. 39, pp. 82–92, Aug. 2016.
[9] J. Hao, C. Li, Z. Kim, and Z. Xiong, "Spatio-temporal traffic scene modeling for object motion detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 295–302, Mar. 2013.
[10] Z. Zhong, B. Zhang, G. Lu, Y. Zhao, and Y. Xu, "An adaptive background modeling method for foreground segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1109–1121, May 2017.
[11] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 246–252.
[12] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection—A survey," *Recent Patents Comput. Sci.*, vol. 1, no. 3, pp. 219–237, 2008.
[13] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 28–31.
[14] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, May 2006.
[15] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vis.*, Jun. 2000, pp. 751–767.
[16] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.
[17] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.
[18] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 38–43.
[19] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 393–400.
[20] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. Int. Conf. Syst., Signals, Image Process.*, May 2016, pp. 1–4.
[21] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognit.*, vol. 76, pp. 635–649, Apr. 2018.
[22] S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?" in *Proc. Int. Conf. Image Anal. Process.*, Sep. 2017, pp. 96–107.
[23] L. A. Lim and H. Y. Keles. (2018). "Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding." [Online]. Available: https://arxiv.org/abs/1801.02225
[24] T. Minematsu, A. Shimada, and R.-I. Taniguchi, "Analytics of deep neural network in change detection," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2017, pp. 1–6.
[25] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU-Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, Aug. 2010.
[26] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171–177, Jan. 2010.
[27] K. Garg, S.-K. Lam, T. Srikanthan, and V. Agarwal, "Real-time road traffic density estimation using block variance," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2016, pp. 1–9.
[28] K. Garg, A. Prakash, and T. Srikanthan, "Low complexity techniques for robust real-time traffic incident detection," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2017, pp. 1–8.
[29] B. Zhong, H. Yao, and S. Liu, "Robust background modeling via standard variance feature," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 14–19.
[30] L. Maddalena and A. Petrosino, "Towards benchmarking scene background initialization," in *Proc. Int. Conf. Image Anal. Process.*, Aug. 2015, pp. 469–476.
[31] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequièvre, "A benchmark dataset for outdoor foreground/background extraction," in *Proc. Asian Conf. Comput. Vis.*, Nov. 2012, pp. 291–300.
[32] C. Cuevas and N. Garcia, "Efficient moving object detection for lightweight applications on smart cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 1–14, Jan. 2013.

[33] K. Wang, F.-Y. Wang, C. Gou, and Y. Liu, "A multi-view learning approach to foreground detection for traffic surveillance applications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4144–4158, Jun. 2016.

[34] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[35] *Odroid-XU4*. Accessed: May 21, 2019. [Online]. Available: https://bit.ly/30yeWtF

[36] *i-LIDS Parked Vehicle Dataset*. Accessed: May 21, 2019. [Online]. Available: http://www.eecs.qmul.ac.uk/~andrea/avss2007_ss_challenge.html

[37] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand.*, vol. 122, pp. 4–21, May 2014.

[38] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[39] J. Yao and J.-M. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[40] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Mach. Vis. Appl.*, vol. 8, no. 3, pp. 187–193, May 1995.

[41] A. Manzanera and J. Richefeu, "A new motion detection algorithm based on Σ-Δ background estimation," *Pattern Recognit. Lett.*, vol. 28, no. 3, pp. 420–428, Feb. 2007.

[42] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[43] C. Kertész and V. Oy, "Texture-based foreground detection," *Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 4, no. 4, 2011.

[44] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *Proc. Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[45] D. D. Bloisi and L. Iocchi, "Independent multimodal background subtraction," in *CompIMAGE*, 2012, pp. 39–44.

[46] F. E. Baf, T. Bouwmans, and B. Vachon, "Type-2 fuzzy mixture of Gaussians model: Application to background modeling," in *Advances in Visual Computing*. Berlin, Germany: Springer, 2008.

[47] M. H. Sigari, N. Mozayani, and H. R. Pourreza, "Fuzzy running average and fuzzy background subtraction: Concepts and application," *Int. J. Comput. Sci. Netw. Secur.*, vol. 8, no. 2, pp. 138–143, Feb. 2008.

[48] H. Zhang and D. Xu, "Fusing color and texture features for background model," in *Fuzzy Systems and Knowledge Discovery*. Berlin, Germany: Springer, 2006.

[49] F. E. Baf, T. Bouwmans, and B. Vachon, "Fuzzy integral for moving object detection," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jun. 2008, pp. 1729–1736.

[50] L. Maddalena and A. Petrosino, "A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection," *Neural Comput. Appl.*, vol. 19, no. 2, pp. 179–186, Mar. 2010.

[51] S. Noh and S. Jeon, "A new framework for background subtraction using multiple cues," in *Proc. Eur. Conf. Comput. Vis.*, Nov. 2012, pp. 493–506.

[52] P.-L. St-Charles and G.-A. Bilodeau, "Improving background subtraction using local binary similarity patterns," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 509–515.

Kratika Garg received the B.Tech. degree in electronics and communication engineering from VIT University, Vellore, India, in 2014. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, where she is currently a Project Officer. Her research interests include developing low complexity computer vision algorithms for low-cost smart traffic surveillance systems.

Nirmala Ramakrishnan received the Ph.D. degree from the School of Computer Science and Engineering, Nanyang Technological University (NTU), in 2017. Her interests include embedded computer vision algorithms/systems specifically, feature detection and tracking, and background modeling for low cost/low power smart camera systems. She is currently a Research Fellow with the Hardware and Embedded Systems Laboratory, NTU, focusing on low complexity vision algorithms for traffic surveillance cameras.

Alok Prakash received the Ph.D. degree from the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2014, where he is currently a Research Fellow. Prior to this, he was a Research Fellow at the School of Computing, National University of Singapore. His research interests include developing customized application specific platforms, custom instructions, and processor selection and customization.

Thambipillai Srikanthan joined Nanyang Technological University, Singapore, in 1991. He founded the Centre for High Performance Embedded Systems (CHiPES) in 1998 and elevated it to a University Level Research Centre in 2000. He currently holds a full professor and joint appointments as the Director of CHiPES which is a 100 strong center and also the Director of the Intelligent Devices and Systems (IDeAS) Cluster. He has authored over 250 technical papers. His research interests include design methodologies for complex embedded systems, architectural translations of compute intensive algorithms, computer arithmetic, high-speed techniques for image processing, and dynamic routing.