Real-Time Hysteresis Foreground Detection in Video Captured by Moving Cameras

Hadi Ghahremannezhad and Chengjun Liu
Department of Computer Science, New Jersey Institute of Technology
Newark, NJ 07102, USA
Email: hg255@njit.edu, cliu@njit.edu

Hang Shi
Innovative AI Technologies
Newark, NJ 07201, USA
Email: hs328@njit.edu

***Abstract*—Foreground detection is an important first step in video analytics. While the stationary cameras facilitate the foreground detection due to the apparent motion between the moving foreground and the still background, the moving cameras make such a task more challenging because both the foreground and the background appear in motion in the video. To tackle this challenging problem, an innovative real-time foreground detection method is presented, that models the foreground and the background simultaneously and works for both moving and stationary cameras. In particular, first, each input video frame is partitioned into a number of blocks. Then, assuming the background takes the majority of each video frame, the iterative pyramidal implementation of the Lucas-Kanade optical flow approach is applied on the centers of the background blocks in order to estimate the global motion and compensate for the camera movements. Subsequently, each block in the background is modeled by a mixture of Gaussian distributions and a separate Gaussian mixture model is constructed for the foreground in order to enhance the classification. However, the errors in motion compensation can contaminate the foreground model with background values. The novel idea of the proposed method matches a set of background samples to their corresponding block for the most recent frames in order to avoid contaminating the foreground model with background samples. The input values that do not fit into either the statistical or the sample-based background models are used to update the foreground model. Finally, the foreground is detected by applying the Bayes classification technique to the major components in the background and foreground models, which removes the false positives caused by the hysteresis effect.. Experimental evaluations demonstrate the feasibility of the proposed method in the foreground segmentation when applied to videos in public datasets.**

I. Introduction

Detecting the location of interesting objects has been intensively studied in the field of computer vision. Generally speaking, the current techniques for locating objects of interest can be categorized into two groups of appearance-based and motion-based methods. Motion-based methods are applicable to video data and tend to perform a binary classification on the pixel locations at each video frame. In many applications of video analytics systems the objects of interest (aka the foreground) have a dynamic pattern different from the rest of the scene, namely background. This difference has been exploited by many studies in order to segment the foreground from the background and subsequently locate the objects of interest.

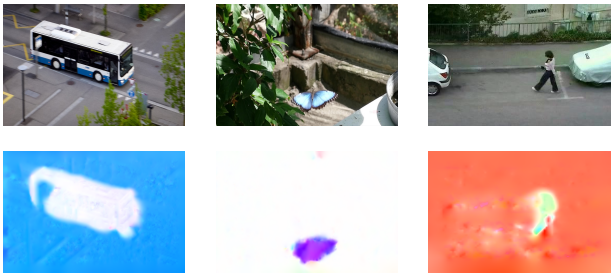


Fig. 1. Optical flow field calculated by applying the UnFlow method [25]. The direction is indicated by hue and the velocity is represented by saturation.

Foreground segmentation has specifically been applied to intelligent surveillance systems [3], [5], traffic monitoring [2], [10], [13]–[17], [22], [30]–[33], gesture recognition [18], [19], and robot vision [24], [35]. The input video data used in the majority of these applications are captured by stationary cameras which causes the foreground to have significant motion compared to the background. A large number of studies have attempted various approaches to subtract the relatively static background from the changing foreground in order to detect the location of the moving objects [12]. The strong presumption that the camera is stationary or only has jittering movements is common among all these studies and substantially affects their strategies to the point that they become ineffective in case the camera has considerable movements. However, in real-world applications camera movements are common and can happen in restricted forms, such as pan, tilt, or zoom in case of PTZ cameras used in auto tracking and video surveillance, and freely moving cameras, such as handheld cameras, smartphones, drones, or dashcams, in which case the camera is mounted on a moving platform. In all these scenarios the camera is non-stationary with regards to the the captured scene and therefore, everything seems to be moving in reference to the camera. Consequently, there is a need to implement foreground segmentation methods that are capable of dealing with camera motion and quickly adapt to the changes in the background.

When relying solely on motion information to segment the foreground from the background in video frames captured by non-stationary cameras the only heuristic lies in the differences

between the dynamic pattern of the moving objects and the background (Figure 1). Many approaches have been proposed to take these differences into account and locate the objects of interest in videos captured by non-stationary cameras [6], [37].

The real-world applicability of the current methods suffers from high requirements in computational resources and/or low performance in classifying foreground and background. Here we apply spatial and temporal features for statistical modeling of the background and the foreground separately in order to classify them in real-time. Each block of the background is modeled using a mixture of Gaussian distributions (MOG) and a set of values sampled randomly in spatial and temporal domains. At each video frame the Lucas-Kanade optical flow method is applied on the block centers in order to estimate the camera motion and find the corresponding locations between two adjacent frames. The global motion is then compensated by updating the background models of each block according to the values of its corresponding location in the previous frame. On the other hand, the foreground is modeled by another MOG which is updated by the input values that do not fit into the background models. The final classification is performed by comparing the input super-pixel intensity values with the major components in the statistical background and foreground models. The remainder of this paper is organized as follows: In section II the main steps of the proposed framework are described in order. Section III contains experimental evaluations of the method's performance and the conclusions are summarized in section IV.

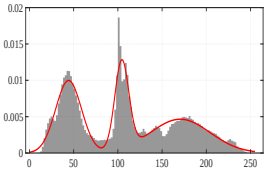
II. The Proposed Foreground Segmentation Method

First observation in videos obtained by moving cameras is that the entire captured scene appears to be moving from the camera's perspective. However, by assuming the background to occupy the majority of the scene compared to the objects of interest we can estimate the motion of the camera relative to the background. Afterwards, the estimated camera motion can be compensated by using the corresponding values in the previous frame for updating background models. After motion compensation the foreground can be segmented using approaches similar to the methods used for the applications of stationary cameras. Here, we apply an MOG to model the entire foreground using the values that are not absorbed by the background models. The major components of the Gaussian mixture distributions in the background and foreground models are utilized for final binary classification. The details of each step are described in this section.

A. Global motion estimation

The main purpose behind moving the camera in most applications of video analytics is to focus on the interesting objects and trying to keep them in view field of the camera. In many scenarios the objects of interest occupy a portion each video frame and the remaining majority is considered to be background. Therefore, the majority of point displacements among video frames is caused by the camera motion which can be estimated by calculating the global motion. For the sake of computational efficiency and accounting for spatial relationships, a similar approach to [27], [40] is applied where the input image is converted to grayscale and divided into a number of grids with equal sizes. The Kanade–Lucas–Tomasi feature tracking approach [34] is applied on the centers of the grid cells from the previous frame. Then a homography matrix is obtained that warps the image pixels at frame t to pixels at frame $t-1$ through a perspective transform. If we denote the intensity of the grayscale image at time t by $I^{(t)}$ and assume consistent intensity between consecutive frames, the corresponding location of each point in the new frame can be used to calculate the global velocity vector as follows:

(a) The foreground object



(b) MOG model

Fig. 2. The foreground is modeled by a mixture of Gaussian distribution.

$$I^{(t)}(x_i + u_i, y_i + v_i) = I^{(t-1)}(x_i, y_i) \tag{1}$$

where (u_i, v_i) is the velocity vector of the center point of the i-th block located at (x_i, y_i). Three-dimensional vectors X_i can be constructed as:

$$X_i^{(t-1)} = (x_i, y_i, 1)^T, \quad X_i^{(t)} = (x_i + u_i, y_i + v_i, 1)^T \tag{2}$$

and a reverse transformation matrix $H_{t:t-1}$ is obtained that satisfies eq. (1) for the largest possible number of samples:

$$\left[X_1^{(t)}, X_2^{(t)}, ...\right] = \mathbf{H}_{t:t-1} \left[X_1^{(t-1)}, X_2^{(t-1)}, ...\right] \tag{3}$$

which is solved by applying the by RANSAC algorithm [11] in order to remove outliers from the further calculations. Also the center points of the blocks classified as foreground in the previous frame are excluded from this calculation as they do not contribute to the camera motion.

B. Background and foreground modeling

Each block of the image is modeled by a mixture of Gaussian distributions and the model is updated at each video frame. In order to update the background models at each frame we have to calculate the corresponding values in the warped background image of the previous frame. The mean and variance of the warped background model is calculated as a weighted sum of the neighboring models where each weight is proportional to a rectangular area as a bilinear interpolation:

$$\begin{aligned} \tilde{\mu}_i^{(t-1)} &= \sum_{k \in \mathcal{R}_i} \omega_k \mu_k^{(t-1)} \\ \tilde{\sigma}_i^{(t-1)} &= \sum_{k \in \mathcal{R}_i} \omega_k \sigma_k^{(t-1)} \end{aligned} \tag{4}$$

(a) (b) (c)

Fig. 3. Improving classification results with the aid of foreground modeling. (a) Original frame, (b) False-positives caused by hysteresis effect in background modeling, (c) False-positives are avoided after foreground modeling.

where $\mathcal{R}$ is a set of block indices falling in a rectangular region centered at the corresponding point location calculated by the homography matrix in eq. (3), ω_k is the weight calculated by the overlapping area, and μ and σ represent the mean and variance of the Gaussian distributions, respectively.

Since the camera might have movements in the form of pan there can be slight variations in the illumination due to the changes in the angle of view and light direction. Also, even after motion compensation the pan motion of the camera can cause a part of the background to move out of the scene which results in a block to represent another part. The Gaussian modeling keeps the information of previous frames and might be slow in catching up with the pace of changing values at the borders of the video frames. In order to make the model parameters adapt to these changes a global variation factor g is calculated by subtracting the mean intensities in the background model and the current frame:

$$g^{(t)} = \frac{1}{N}\sum_{j=1}^{N} I_j^{(t)} - \frac{1}{B}\sum_{i=1}^{B} \tilde{\mu}_i^{(t-1)} \tag{5}$$

with B being the number of blocks and N being the number of pixels. At each frame the parameters of the Gaussian mixture model for each block are updated as follows:

$$\begin{aligned} \mu_k^{(t)} &= \left(n_k^{(t-1)}\left(\tilde{\mu}_k^{(t-1)} + g^{(t)}\right) + M^{(t)}\right)/(n_k^{(t-1)}+1) \\ \sigma_k^{(t)} &= \left(n_k^{(t-1)}\tilde{\sigma}_k^{(t-1)} + V^{(t-1)}\right)/(n_k^{(t-1)}+1) \\ n_k^{(t)} &= n_k^{(t-1)} + 1 \\ \alpha_k^{(t)} &= n_k^{(t)} / \sum_{k=1}^{K} n_k^{(t)} \end{aligned} \tag{6}$$

where n_k is a counter representing the number of times an input value has been used to update component k, α_k is the weight of the kth component, M and V stand for the mean intensity and the variance of the block, respectively. The component with the largest weight of each Gaussian mixture model is considered to be the background value of the block.

In case of moving cameras the objects of interest are usually present in the scene for a longer time as the camera is focused on them. Therefore, it is reasonable to model the values of the foreground objects throughout the video. A similar approach to background modeling is applied for modeling the foreground except only one mixture of Gaussian distributions is used for the entire foreground pixels. Also, instead of a single component, a number of components from the foreground model that have the largest weights are considered to represent the foreground objects. This is because the foreground objects have multiple parts with different intensity values and each major component in the foreground model is used to represent one part of the foreground. Figure 2 illustrates an example of a foreground object modeled by an MOG with three components.

Algorithm 1: Acquiring the foreground mask

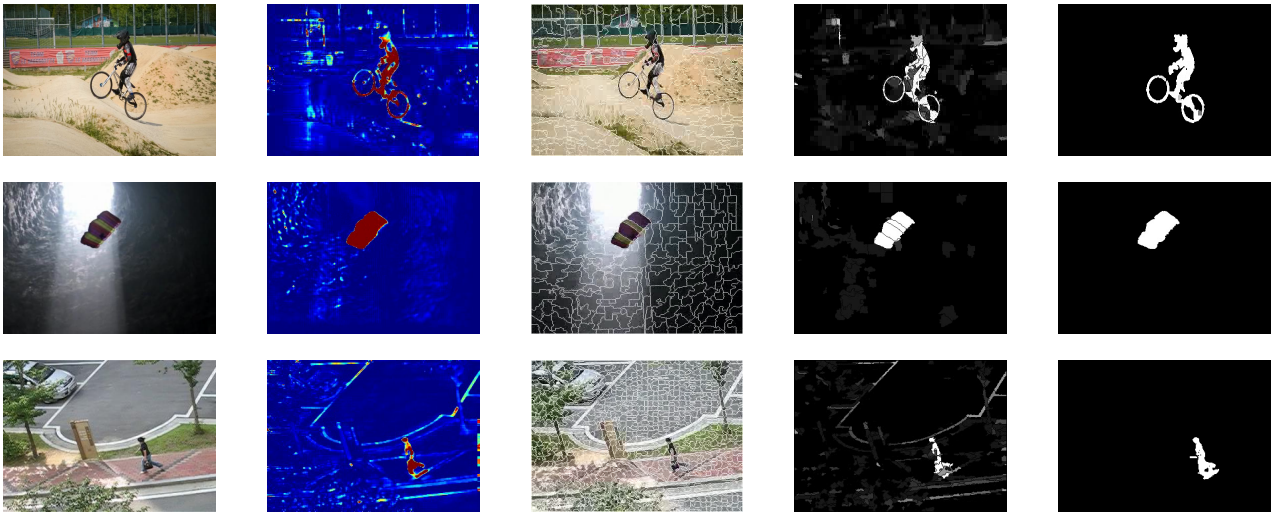
Input:
The input video frame in gray-scale $I^{(t)}$
A set of predefined thresholds
Output:
The foreground mask $\mathcal{H}$ of the same size as the video frame

initialize $\mathcal{F}$ with 0;
foreach *pixel* $p \in I^{(t)}$ **do**
  if *p fits into the MOG model of block i* **then**
    update the ith MOG;
  end
  else if *p doesn't fit the ith sample-based model* **then**
    update the foreground MOG;
  end
end
apply watershed segmentation to obtain $\mathbb{P}$;
$\mathcal{H} = 0$;
foreach *super-pixel* $P_i \in \mathbb{P}$ **do**
  calculate $\mathcal{F}(P_i)$;
  foreach *component m in foreground model* **do**
    calculate $\mathcal{B}_m(P_i)$;
    if $\mathcal{F}(P_i) > \mathcal{B}_m(P_i)$ **then**
      $\mathcal{H}(P_i) = 1$;
      break;
    end
  end
end

In addition to the statistical modeling and inspired by the ViBe method [1], we keep a set of sample values as a secondary non-parametric model for each block. This set is initialized by the mean value of the block and its neighboring blocks at the first frame. At each of the consecutive frames one of the values in the set is selected randomly and replaced with the new mean value. We can denote the collection of background sample values for the block i as $\mathcal{S}_i$ as follows:

$$\mathcal{S}_i = \{s_i^1, s_i^2, ..., s_i^K\} \tag{7}$$

where s_i^k is the kth sampled mean intensity of block i. The sample-based model is kept and updated mainly to avoid contaminating the foreground model by the background values that do not fit into any of the Gaussian components of the corresponding block model. This problem occurs mostly because of motion compensation errors or new background values being introduced into the scene due to the camera



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

Fig. 4. The steps in the final classification process. (a) Original frame, (b) Heat-map of the foreground probability, (c) Super-pixels obtained by applying watershed segmentation, (d) Foreground confidence map, (e) Final foreground mask.

motion. If an input value does not fit into any of the Gaussian components of a background model the Euclidean distance between the pixel value and each background sample in the set of the corresponding block is calculated. If the number of samples in the set of block i that are closer than a distance threshold to the input value is less than a counting threshold, the foreground model is updated by that value. Representing this number of samples by C_i it can be calculated as follows:

$$C_i = \sum_{j=1}^{|\mathcal{S}_i|} \mathbb{1}\left(D(\mathbf{x}, \tilde{s}_j^{(s)}) < \theta_d\right) \tag{8}$$

with $\mathbf{x}$ being the input pixel intensity value, D representing the Euclidean distance, θ_d being a predefined threshold, $\mathbb{1}$ denoting an indicator function, $\tilde{s}_j^{(s)}$ representing the corresponding value of $s_i^{(k)}$ after motion compensation, and $\mathcal{N}_i$ denoting the set of neighboring blocks.

Since the camera is in motion the parameters in the background models can lag behind the sudden changes caused by motion compensation errors, sudden illumination changes, or new samples appearing at the borders of the frame. Consequently, the distance between the new samples and the mean values may exceed the threshold defined based on the standard deviations which in turn causes the new samples to falsely be classified as foreground. By keeping a set of values containing a number of recent background samples we can compensate for the hysteresis effect of Gaussian models representing the older samples. We calculate the Euclidean distances between the new values and the samples in the set and only classify the new values as foreground if they match with less than a few samples in the set. The foreground model is only updated with values that belong to the foreground class with a high certainty and therefore, the majority of false positive cases are avoided. An example of the classification is illustrated in Figure 3. As seen in the fig. 3b, some of the input values do not fit into their corresponding background model due to the camera movements and the motion compensation errors. In fig. 3c these values are removed from the foreground mask as they do not fit into any of the major components of the foreground model.

C. Background and foreground classification

For the final classification, at first the foreground likelihood values are calculated for each pixel at an input image as follows:

$$L_{fg}(x,y) = \frac{(I(x,y) - \mu_k)^2}{\sigma_k} \tag{9}$$

where $I(x,y)$ and $L_{fg}(x,y)$ are the intensity and foreground likelihood values of the pixel at location (x,y), and μ_k and σ_k are the mean and variance of the corresponding background block, respectively. Afterwards, the watershed segmentation algorithm [26] is applied to each input image in order to extract a set of super-pixels, notated by $\mathbb{P} = \{P_1, P_2, ..., P_k\}$.

For final classification the mean value of each super-pixel is compared against the major component in the background model of the corresponding block as well as each component in the foreground model. The foreground confidence map $\mathcal{F}$ is obtained by calculating the mean of confidence values in each super-pixel as follows:

$$\mathcal{F}(P_i) = \frac{1}{|P_i|} \sum_{x,y \in P_i} L_{fg}(x,y) \tag{10}$$

where $|P_i|$ is the number of pixels at super-pixel P_i. Assuming there are M major components in the global foreground model, a background confidence map $\mathcal{B}_m, m \in \{1, ..., M\}$ is similarly obtained based on each component. The Gaussian Naive Bayes (GNB) classifier is applied for each super-pixel in order to calculate the z-score distance between the input value

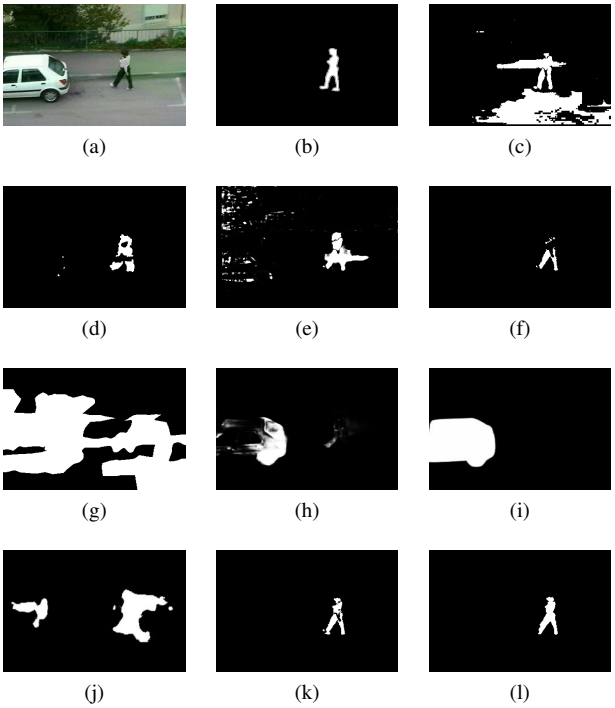


Fig. 5. Comparison of the qualitative results of other methods applied on the "Woman" sequence. (a) Original frame, (b) Ground truth, (c) MCD [27], (d) MCD NP [20], (e) Stochastic approx [23], (f) SC MCD [40], (g) uNLC [9], (h) OSVOS [4], (i) BASNet [29], (j) CIS [36], (k) uMOD [39], (l) Proposed method.

and each class-mean and classify the super-pixel accordingly in order to obtain the final foreground mask $\mathcal{H}$:

$$\mathcal{H}(P_i) = \begin{cases} 1, \text{if } \mathcal{F}(P_i) > \mathcal{B}_m(P_i) \\ 0, \text{otherwise} \end{cases} \tag{11}$$

where $\mathcal{B}_m$ is the background confidence map corresponding to the m-th foreground model and $\mathcal{H}(P_i) = 1$ indicates that the super-pixel at location P_i belongs to the moving objects and $\mathcal{H}(P_i) = 0$ means it belongs to the background. The process of segmenting the foreground is detailed in Algorithm 1.

The different stages in the classification process can be seen in Figure 4 . From top to bottom, each row in the figure represents a sample video frame from the DAVIS [28], Segment Pool Tracking [21], and SCBU [40] datasets, respectively. The second column represents heatmaps where the pixels with a higher probability of belonging to the foreground are represented by red colors. The third column is the results of the watershed segmentation algorithm applied to each video frame with the markers chosen uniformly across the image at the same locations as the background block centers. The fourth column illustrates the foreground confidence maps calculated based on eq. (10) and the last column is the final results of foreground detection after morphological dilation.

III. Experiments

The performance of the proposed method is evaluated using video data collected from the publicly available SCBU dataset [40] that consists of nine video sequences captured by moving cameras. The videos in the dataset impose various challenges in the way of foreground segmentation, such as fast or slow moving objects, objects with different sizes, illumination changes, and the similarities in intensity values between the background and foreground. Figure 5 represents the foreground masks detected by various methods. Similar to [39], in addition to background modeling methods [1], [7], [8], [20], [23], [27], [38], [40], the detection results are compared with a number of object-centric methods, such as uNLC [9], which is the unsupervised version of the NLC [9] approach, OSVOS [4] without the fine-tuning step, CIS [36], and BASNet [29]. In terms of time and space complexity, the statistical methods are more efficient as the methods based on deep neural networks require more resources. Therefore, our method is more practical in applications with real-time requirements and edge devices that have a lower hardware capacity.

Figure 6 represents the foreground detection results in a number of video sequences compared with other background modeling methods. It can be seen that our proposed method is able to detect the foreground in various challenging scenarios. One of the limitations in the proposed method is the ability of the foreground model to adapt well to sudden illumination changes caused by the pan movements of the camera. Also, despite applying the Bayes classification approach the camouflage problem still exists in the cases where the foreground is very similar to the corresponding background block (part of the person's head is not detected in fig. 5l). This problem can be solved by introducing more discriminating features to the statistical modeling process.

The hardware specification used for the experiments is a 3.4 GHz processor and 16 GB RAM. The average processing speed for video frames of size 320×240 pixels was about $\sim$ 143 frames per second, which is feasible for real-time applications of video analytics. The average running speed of the proposed method is reported in Table II for each video frame of size 320×240 pixels. The run-time calculations show that the method is feasible to be used as a pre-processing step in real-time traffic video analysis tasks.

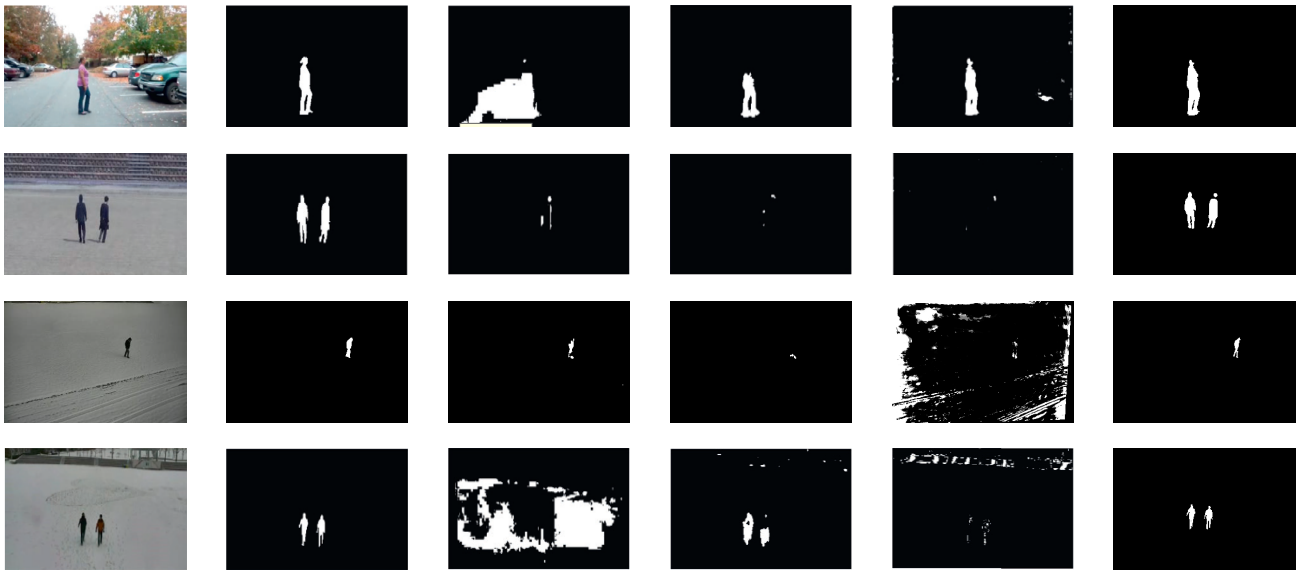
The f-score metric is used in order to evaluate the quantitative results:

$$\begin{cases} PRE = T_P/(T_P + F_P) \\ REC = T_P/(T_P + F_N) \\ F_1 = 2 \times (PRE \times REC)/(PRE + REC) \end{cases} \tag{12}$$

where T_P, F_P are the number of pixels correctly and incorrectly reported as road regions, and T_N and F_N are the number of pixels that are correctly and incorrectly reported as non-road regions, respectively. PRE, REC, and F_1 refer to precision, recall, and F1-score, respectively. The F1-scores are listed in Table I in comparison with other popular methods. The quantitative results demonstrate the robustness of our method in detecting the foreground mask in different videos.

TABLE I
THE FOREGROUND SEGMENTATION RESULTS COMPARED TO OTHER METHODS.

Methods	Walking	Skating	Woman	Woman2	Ground1	Ground2	Ground3	Ground4	Ground5	Average
ViBe [1]	0.0375	0.2229	0.0375	0.0929	0.5656	0.4733	0.4118	0.0299	0.1309	0.2107
FIC [8]	0.0613	0.2373	0.0361	0.1345	0.4543	0.4108	0.1538	0.0453	0.1319	0.1761
BMRI-ViBE [7]	0.0438	0.2402	0.0400	0.0921	0.4249	0.3868	0.2161	0.0383	0.1377	0.1730
MCD NP [20]	0.4351	0.4164	0.4935	0.5791	0.2773	0.3750	0.1222	0.1969	0.3540	0.3519
FP Sampling [38]	0.7058	0.8539	0.7268	0.5828	0.7977	0.8306	0.1396	0.4226	0.8212	0.6646
MCD [27]	0.7349	0.2447	0.3395	0.3448	0.6573	0.7177	0.1531	0.5274	0.0678	0.4523
SC MCD [40]	0.7496	0.8560	0.6650	0.6311	0.8965	0.9118	0.8843	0.8824	0.9326	0.8173
Stochastic approx [23]	**0.8335**	0.6543	0.3986	**0.8783**	0.2221	0.2792	0.0181	0.0111	0.2181	0.4392
uNLC [9]	0.0158	0.1419	0.0178	0.0487	0.0570	0.0342	0.0216	0.0031	0.0143	0.0389
OSVOS [4]	0.3397	0.5344	0.0121	0.1260	0.7697	0.5447	**0.9696**	0.0050	0.1224	0.4127
CIS [36]	0.0538	0.3036	0.1522	0.4681	0.1545	0.0862	0.0581	0.0046	0.0184	0.1418
BASNet [29]	0.3433	0.9379	0.0205	0.2289	0.6039	**0.9564**	0.9586	**0.9439**	**0.9829**	0.6188
uMOD [39]	0.7809	0.9600	0.7269	0.7065	0.9037	0.9032	0.8700	0.9080	0.9793	0.8546
Proposed method	0.8144	**0.9710**	**0.7874**	0.7136	**0.9112**	0.9113	0.8946	0.8812	0.9686	**0.8725**



(a) (b) (c) (d) (e) (f)

Fig. 6. Comparison of the qualitative results of background modeling methods. From top to bottom, the rows represent the *Woman2*, *Ground3*, *Ground4*, and *Ground5* sequences. Each subfigure at the first column illustrates one video frame of each sequence with the corresponding ground-truth represented at the second column. The remaining columns are the classification results of (c) MCD [27] method, (d) MCD NP [20] method, (e) Stochastic approx [23] method, and (f) our proposed method, respectively.

TABLE II
THE AVERAGE RUNNING TIME (IN MILLISECONDS) FOR EACH VIDEO FRAME IN DIFFERENT METHODS

Methods	Run time (ms)	FPS
ViBe [1]	14.6	68.5
MCD [27]	**7.46**	**134**
MCD NP [20]	20.9	47.85
SC MCD [40]	9.56	104.6
uMOD [39]	29.23	34.2
Proposed method	9.4	106.38

IV. CONCLUSION

In this study, a new real-time method is proposed for locating the moving objects in videos captured by non-stationary cameras, which is one the challenging problems in computer vision. The global motion is estimated and used to compensate for background variations caused by the camera movements. Each block is modeled by a mixture of Gaussian distributions which is updated by the values at the corresponding locations in the warped image after motion compensation. Additionally, the mean values of each block are modeled along with the mean values of its neighboring blocks as a set of samples which is in turn updated by random selection. The foreground on the other hand is modeled by a separate MOG which is updated by values that do not fit into either of the statistical or sample-based background models. For classification, each input value is compared against both the background and foreground models to obtain the definite and the candidate foreground locations, respectively. The watershed segmentation algorithm is then applied to detect the final foreground mask. Experimental results demonstrate the feasibility of the proposed method in real-time video analytics systems.

REFERENCES

[1] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image processing*, vol. 20, no. 6, pp. 1709–1724, 2010.

[2] R. Bhardwaj, A. Dhull, and M. Sharma, "A computationally efficient real-time vehicle and speed detection system for video traffic surveillance," in *Proceedings of International Conference on Artificial Intelligence and Applications*. Springer, 2021, pp. 583–594.

[3] T. Bouwmans and B. García-García, "Visual surveillance of human activities: Background subtraction challenges and methods," in *From Visual Surveillance to Internet of Things*. Chapman and Hall/CRC, 2019, pp. 43–66.

[4] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221–230.

[5] Y.-T. Chan, "Comprehensive comparative evaluation of background subtraction algorithms in open sea environments," *Computer Vision and Image Understanding*, vol. 202, p. 103101, 2021.

[6] M.-N. Chapel and T. Bouwmans, "Moving objects detection with a moving camera: A comprehensive review," *Computer Science Review*, vol. 38, p. 100310, 2020.

[7] F.-C. Cheng, B.-H. Chen, and S.-C. Huang, "A background model re-initialization method based on sudden luminance change detection," *Engineering Applications of Artificial Intelligence*, vol. 38, pp. 138–146, 2015.

[8] J. Choi, H. J. Chang, Y. J. Yoo, and J. Y. Choi, "Robust moving object detection against fast illumination change," *Computer Vision and Image Understanding*, vol. 116, no. 2, pp. 179–193, 2012.

[9] A. Faktor and M. Irani, "Video segmentation by non-local consensus voting." in *BMVC*, vol. 2, no. 7, 2014, p. 8.

[10] M. O. Faruque, H. Ghahremannezhad, and C. Liu, "Vehicle classification in video using deep learning," in *Machine Learning and Data Mining in Pattern Recognition, MLDM*. ibai publishing, Leipzig, 2019, pp. 117–131.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[12] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Computer Science Review*, vol. 35, p. 100204, 2020.

[13] H. Ghahremannezhad, H. Shi, and C. Liu, "Robust road region extraction in video under various illumination and weather conditions," in *2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)*. IEEE, 2020, pp. 186–191.

[14] ——, "Automatic road detection in traffic videos," in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE, 2020, pp. 777–784.

[15] ——, "A new adaptive bidirectional region-of-interest detection method for intelligent traffic video analysis," in *2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. IEEE, 2020, pp. 17–24.

[16] ——, "A real time accident detection framework for traffic video analysis," in *Machine Learning and Data Mining in Pattern Recognition, MLDM*. ibai publishing, Leipzig, 2020, pp. 77–92.

[17] ——, "A new online approach for moving cast shadow suppression in traffic videos," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3034–3039.

[18] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak, "Hand gesture recognition for human computer interaction," *Procedia computer science*, vol. 115, pp. 367–374, 2017.

[19] M. M. Islam, M. R. Islam, and M. S. Islam, "An efficient human computer interaction through hand gesture using deep convolutional neural network," *SN Computer Science*, vol. 1, no. 4, pp. 1–9, 2020.

[20] S. W. Kim, K. Yun, K. M. Yi, S. J. Kim, and J. Y. Choi, "Detection of moving objects with a moving camera using non-panoramic background model," *Machine vision and applications*, vol. 24, no. 5, pp. 1015–1028, 2013.

[21] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *ICCV*, 2013.

[22] G. Liu, H. Shi, A. Kiani, A. Khreishah, J. Lee, N. Ansari, C. Liu, and M. M. Yousef, "Smart traffic monitoring system using computer vision and edge computing," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[23] F. J. López-Rubio and E. López-Rubio, "Foreground detection for moving cameras with stochastic approximation," *Pattern Recognition Letters*, vol. 68, pp. 161–168, 2015.

[24] E. Martinez-Martin and A. P. Del Pobil, "Robot vision for manipulation: a trip to real-world applications," *IEEE Access*, vol. 9, pp. 3471–3481, 2020.

[25] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[26] F. Meyer, "Topographic distance and watershed lines," *Signal processing*, vol. 38, no. 1, pp. 113–125, 1994.

[27] K. Moo Yi, K. Yun, S. Wan Kim, H. Jin Chang, and J. Young Choi, "Detection of moving objects with non-stationary cameras in 5.8 ms: Bringing motion detection to your mobile device," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 27–34.

[28] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[29] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "Basnet: Boundary-aware salient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7479–7489.

[30] H. Shi, H. Ghahremannezhad, and C. Liu, "Anomalous driving detection for traffic surveillance video analysis," in *2021 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, 2021, pp. 1–6.

[31] H. Shi, H. Ghahremannezhadand, and C. Liu, "A statistical modeling method for road recognition in traffic video analytics," in *2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. IEEE, 2020, pp. 000 097–000 102.

[32] H. Shi and C. Liu, "A new foreground segmentation method for video analysis in different color spaces," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2899–2904.

[33] ——, "A new global foreground modeling and local background modeling method for video analysis," in *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2018, pp. 49–63.

[34] C. Tomasi and T. Kanade, "Detection and tracking of point," *Int J Comput Vis*, vol. 9, pp. 137–154, 1991.

[35] S.-H. Wang and X.-X. Li, "A real-time monocular vision-based obstacle detection," in *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2020, pp. 695–699.

[36] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto, "Unsupervised moving object detection via contextual information separation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 879–888.

[37] M. Yazdi and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: A survey," *Computer Science Review*, vol. 28, pp. 157–177, 2018.

[38] K. Yun and J. Y. Choi, "Robust and fast moving object detection in a non-stationary camera via foreground probability based sampling," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 4897–4901.

[39] K. Yun, H. Kim, K. Bae, and J. Park, "Unsupervised moving object detection through background models for ptz camera," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 3201–3208.

[40] K. Yun, J. Lim, and J. Y. Choi, "Scene conditional background update for moving object detection in a moving camera," *Pattern Recognition Letters*, vol. 88, pp. 57–63, 2017.