

## Research Article

# Background Modeling with Extracted Dynamic Pixels for Pumping Unit Surveillance

Tianming Yu, Jianhua Yang , and Wei Lu

*Dalian University of Technology, Faculty of Electronic Information and Electrical Engineering, School of Control, Science and Engineering, No. 2 Linggong Road, Ganjingzi District, Dalian 116024, China*

Correspondence should be addressed to Jianhua Yang; [jianhuay@dlut.edu.cn](mailto:jianhuay@dlut.edu.cn)

Received 2 May 2018; Accepted 25 July 2018; Published 18 September 2018

Academic Editor: Giovanni Falsone

Copyright © 2018 Tianming Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Background modeling plays an important role in the application of intelligent video surveillance. Researchers have presented diverse approaches to support the development of dynamic background modeling. However, in the case of pumping unit surveillance, traditional background modeling methods often mistakenly detect the periodic rotational pumping unit as the foreground object. To address this problem here, we propose a novel background modeling method for foreground segmentation, particularly in dynamic scenes that include a rotational pumping unit. In the proposed method, the ViBe method is employed to extract possible foreground pixels from the sequence frames and then segment the video image into dynamic and static regions. Subsequently, the kernel density estimation (KDE) method is used to build a background model with dynamic samples of each pixel. The bandwidth and threshold of the KDE model are calculated according to the sample distribution and extremum of each dynamic pixel. In addition, the strategy of sample adjustment combines regular and real-time updates. The performance of the proposed method is evaluated against several state-of-the-art methods applied to complex dynamic scenes consisting of a rotational pumping unit. Experimental results show that the proposed method is available for periodic object motion scenario monitoring applications.

## 1. Introduction

Background modeling plays a fundamental and important role for foreground detection in application of video analysis, especially intelligent video surveillance, which includes real-time tracking [1] and event analysis [2–4]. The purpose of background modeling is to remove interference from the background and to detect and extract the foreground moving target, to meet the needs of video surveillance. Classic foreground detection algorithms, such as frame difference [5], optical flow [6], and background subtraction [7], are implemented mainly through extracting interframe motion information, detecting optical flow change or background modeling. However, in the application of a dynamic background scene, which includes dynamic noise such as swaying branch, moving water, and the shaking camera, the accuracy and robustness of foreground modeling is an efficient way to obtain foreground objects. Over the past couple of decades, many researchers have presented diverse background modeling methods to identify foreground objects in a video. The general steps [8] of these background modeling methods are

as follows: First, set up a background model by using the first of first few frames of the video. Second, compare the background model to the current frame to obtain the foreground object. Finally, update the background model. Background modeling methods can be categorized into two groups: parametric and nonparametric methods. The most widely used parametric methods are GMM [9] and its modified version. Nonparametric methods, ViBe [10], KDE [11], SACON [12], PBAS [13], and so on, have good adaptability to dynamic background modeling. These methods have good effects on dynamic background, illumination changing conditions, and jitter induced by the sensor. However, for the scene with periodic recurring changes that have the same scale as the foreground object, periodic recurring background objects are easily detected as foreground objects by mistake when using traditional background modeling methods, which lead to a high alarm rate. As shown in Figure 1, the movement of the pumping unit is a special kind of background interference. In this case, the moving pumping unit should be regarded as part of the background, but the existing background modeling



FIGURE 1: Monitoring scene of the pumping unit.

method will mistakenly detect it as foreground, which will lose the purpose of intelligent video surveillance.

This paper proposes a novel background modeling method for foreground segmentation that is different from all previous works, particularly for dynamic scenes having the same scale movement of the background object as the foreground object. Our algorithm models the background with extracted dynamic pixels (BOWED). In the proposed framework, the ViBe method is employed to extract both dynamic and static pixels. The object of periodic motion in the background is detected as dynamic pixels, and KDE is used to build a new background model for these dynamic pixels after they are extracted. For a current frame, each pixel has two background models: a dynamic background and a static background. This modeling method has high efficiency while ensuring accuracy in detecting objects. To our knowledge, this is the first method that uses both ViBe and KDE for the purpose of pumping unit surveillance.

The exposure of the material is structured as follows. Section 2 provides an overview of the foreground detection and background modeling method. Section 3 presents a detailed account of the BOWED. Subsequently, Section 4 describes profound experiments on surveillance videos of a pumping unit to verify the validity and feasibility of the proposed method. Finally, Section 5 provides conclusions.

## 2. Related Work

Over the past decades, many algorithms were proposed to tackle problems with accuracy in moving object detection [14]. Frame difference is one of the most common methods for moving object detection. It uses subtraction of two or three sequential frames to detect moving objects based on differences in pixels and obtain objects by a threshold value. The method does not require background modeling because the previous frame is used as the background model, which is easily implemented and fast to execute. However, it is sensitive to image noise and not sensitive to illumination changes. Additionally, precision in selecting the threshold value is critical, as an unsuitable threshold will result in an unsatisfactory outcome [15, 16]. Optical flow is also a classic method. The main task of optical flow is to calculate the optical flow field, in the appropriate condition, by estimating

the motion region through spatiotemporal gradient of image sequences and then detecting and segmenting the motion target from background by analyzing changes in the moving region. This method can detect the moving target without having information about the scene beforehand. It can obtain the moving object completely and performs well in cases where the camera is moving. However, this method has shortcomings such as complexity and longtime calculation, which does not fulfill the requirement for real-time [17].

The main idea of the background subtraction method is to build the background model, compare the current frame against the background model, and then detect moving objects according to their differences in value. Over the last few years, various background subtraction methods have been tried to improve upon processing speed, reliability, and real-time performance [18–20]. GMM is one of the most classic methods. Stauffer and Grimson [9] proposed a Gaussian mixture model (GMM) for background modeling in cases of dynamic scenes, illumination changes, slow-moving objects, shaking trees, and so on. In addition, an adaptive mixture of Gaussians can respond better to cases of changes in lighting conditions. This method can cope with the problem of multimodal background objects such as waving trees and shadow [9, 21]. However, each pixel has a fixed Gaussian distribution, which requires a large amount of system resources to process. The temporal median filter (TMF) is a nonlinear digital filter, and it is usually used to remove noises from images and signals. TMF assumes that the values of the background of an image sequence are repeated frequently in a time series, which is useful in image processing, and it can filter noises, background models, and so on. In this method, a fixed scene is required to avoid noises when modeling the background. Compared with linear filters, it has higher computational complexity, and its processing speed is not as fast as linear filters [22–24].

Barnich et al. [10] proposed the ViBe method, which first applies a random aggregation to background extraction. In building a samples-based estimation of the background and updating background models, it uses a novel random selection strategy which indicates that information between neighboring pixels can propagate [25, 26]. In this method, the background model is initialized by the first frame, and the model maintenance strategy uses a propagation method with a random updating method. ViBe is superior to many methods because it uses samples as background models, which can effectively represent changes to the background in recent frames. Moreover, it has good robustness and effectiveness for dynamic backgrounds. However, foreground detection can be easily influenced by noise and illumination variation. In ViBe method, the first frame of video is used to initialize the background model.  $f_t(x)$  denotes the value of the pixel  $x$  at time  $t$ . Therefore,  $t=0$  indexes the first frame, and for each pixel in the first frame, a set of samples  $M(x) = v_1, v_2, \dots, v_N$  is randomly selected from its 8-connected neighborhood. Then, a sphere  $S_R(f_t(x))$  of radius  $R$  centered on  $f_t(x)$  is defined.  $\#$  denotes the number of which the samples and the sphere intersect.  $\#$  is compared to  $\#_{min}$ , as (1) shows [25]:

$$\# \{S_R(v(x)) \cap \{v_1, v_2, \dots, v_N\}\} \quad (1)$$

where  $\#_{min}$  denotes the threshold. If  $\#$  is above  $\#_{min}$ , the pixel is regarded as background; otherwise, it is foreground,  $N=20$ ,  $R=20$ , and  $\#_{min} = 2$  [10]. In this method, the background does not need to update each pixel model for each new frame. When a pixel value is deemed as background, it has a 1 in 16 chance of being selected to update its pixel model.

The pixel based adaptive segmenter (PBAS) was proposed by Hofmann et al. in 2012 [13]. This algorithm, a nonparametric model based on pixels, combines the advantages of SACON and ViBe while making some improvements. In PBAS, the thresholds of each pixel are pixel dependent and are determined by the average value of the minimum values, which are obtained by  $N$  distance values from the recent  $N$  frames. PBAS has realized nonparameter moving object detection, and it is robust to slow illumination variations. However, its large calculation quantity causes slow processing speeds. In recent years, St-Charles et al. [27] proposed a high performing method which introduces a new strategy that focuses on balancing the inner workings of a nonparametric model based on pixel-level feedback loops. Moreover, in the same year, they presented an adaptive background subtraction method that was derived from the low-cost and highly efficient ViBe method, and it includes a spatiotemporal binary similarity metric [28].

In 2000, Elgammal et al. presented a novel method based on kernel density estimation (KDE) [11]. It is a nonparametric method that is used for estimating the probability density function. This method eliminates the step of estimating the parameter because it depends on previously observed pixel values, and it is not necessary to store the complete data. Moreover, KDE can address multimodal, fast-changing backgrounds. It is very common and has often been applied to vision processing, especially in cases where the underlying density is unknown, due to its property of converging to any probability density function (pdf). Because KDE can make histogram smooth and continuous, Elgammal et al. applied KDE to build statistical representations of background and foreground, to address the varying problem of pdf [29]. Tavakkoli et al. trained KDE with a full kernel bandwidth matrix for each pixel to build up a unary classification [30]. Wu Wang et al. applied nonparametric KDE on the MoGLOH descriptor to eliminate feature noise [31]. Jeisung Lee and Mignon Park used the first frame to initialize KDE and subsequently updated it at every frame by controlling the learning rate on the basis of image storage reduction [32]. In KDE, it is difficult to choose the proper bandwidth in order to achieve the best probability density function. Furthermore, it requires high computation costs during foreground detection processing [33]. In this method, the background model is modeled from the last  $M$  samples of each pixel. Therefore,  $M$  is the number of data points of the samples. The estimate of the density function  $f(x_{t+1})$  can be obtained by

$$f(x_{t+1}) = \frac{1}{M} \sum_{i=1}^M K_h(x_{t+1} - x_i) \quad (2)$$

where  $K_h$  is a kernel function and its bandwidth is  $h$ .  $x_i$  denotes the data points of samples,  $i = 1 \dots M$ . Gaussian kernel function is adopted in this paper; hence, (2) becomes

$$f(x_{t+1}) = \frac{1}{M} \sum_{i=1}^M \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{(x_{t+1} - x_i)^2}{2h^2}\right) \quad (3)$$

where  $x_{t+1}$  and  $x_i$  are pixel value and sample value of the observed pixel point, respectively.

$Th$  is the global threshold where the pixel is considered a foreground pixel if

$$f(x_{t+1}) < Th \quad (4)$$

otherwise, the pixel is a background pixel.

### 3. The Proposed Method

In the proposed method, ViBe is used to segment each image into static background and dynamic background regions. So that each pixel point has two background models: static model and dynamic model. The color of each pixel is represented in the YUV space and model only on the luminance (Y) channel. The structure of the framework is illustrated in Figure 2.

In the process of background modeling, ViBe is used to extract the foreground pixels of each input video frame, and then OR operation of pixels is used to segment static background and dynamic background regions. In the dynamic region, sample values of each foreground pixel determined to be a foreground point by ViBe processing are then used in KDE background modeling. This method not only reduces the amount of calculation but can also avoid deadlock caused by false detection in the process of selectively updating the dynamic background model. However, in the proposed method, two background models are adopted; through ViBe method, the pixel point of static background model can be judged whether it is background, thereby solving the deadlock problem. KDE is utilized to segment the target foreground object from the dynamic background region. A dynamic threshold is used to detect moving objects. Since the threshold is computed by a global probability density function, it has higher accuracy and adaptive ability and can also accurately identify moving targets in conditions where a pixel contains complex information. For background updates, the static background model adopts the classic ViBe update strategy and the dynamic background model uses a strategy of combined regular and real-time updates, in that the background is updated periodically under normal conditions, but if a moving object is detected, the background is updated immediately. The process of choosing samples to replace old example is fair and random. To our knowledge, this is the first method to detect a foreground moving object under the disturbance of the same scale movement object.

*3.1. Dynamic Background and Static Background Segmentation.* Figure 3 illustrates the procedure for segmenting dynamic background and static background, which is also the procedure of background modeling.  $I(x, y, t)$  is the pixel of an input image, where  $I$  represents the frame,  $(x, y)$  is the coordinate of the pixel, and  $t$  is the frame number. Static pixel and dynamic pixel can be obtained after each frame is processed by the ViBe method. Then, static background and

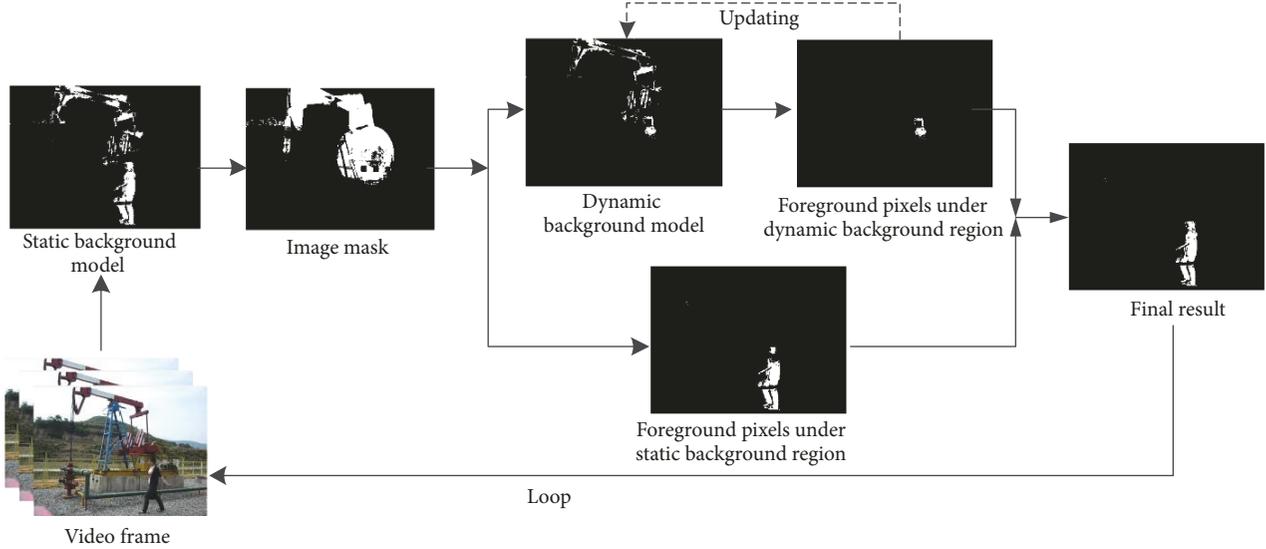


FIGURE 2: Schematic illustration of the proposed method for dynamic background estimation and foreground segmentation.

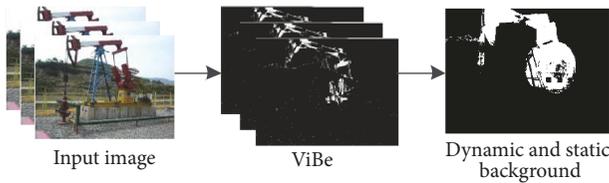


FIGURE 3: Segmentation for dynamic and static background.

dynamic background can be obtained through  $N$  processed images,

$$B(x, y) = \begin{cases} 1, & \text{if } \sum_{t=1}^N I_f(x, y, t) > 0 \\ 0, & \text{if } \sum_{t=1}^N I_f(x, y, t) = 0 \end{cases} \quad (5)$$

where

$$I_f(x, y, t) = \begin{cases} 1, & \text{ViBe}(I(x, y, t)) \text{ is foreground} \\ 0, & \text{ViBe}(I(x, y, t)) \text{ is background} \end{cases} \quad (6) \\ (t = 1, 2, \dots, N)$$

obviously,  $B(x, y)$  is a binary image and is also called the image mask. In the matrix  $B(x, y)$ , the pixel value is 1 for the dynamic background region and 0 for the static background region.

**3.2. Dynamic Background Modeling with Kernel Density Estimation.** The dynamic pixels of video frames have two categories: One is the moving background caused by wind, swaying grass, water ripples, or a regular moving object; these pixels are real background. The other is changes in the dynamic background caused by foreground movement, which is segmented in the proposed method. To achieve segmentation of foreground movement from a dynamic

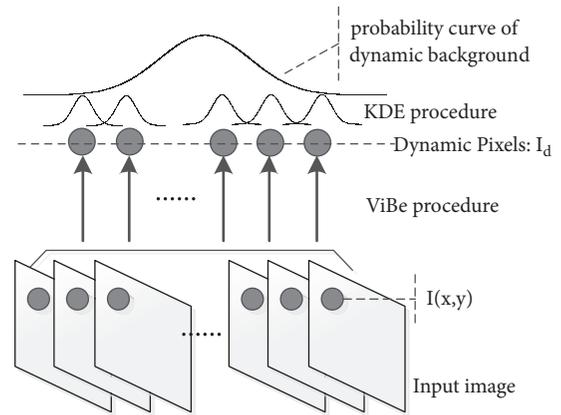


FIGURE 4: Procedure of dynamic background model.

background, the area of dynamic pixels needs to be further processed. In the proposed method, the KDE method is used to model dynamic background in order to extract the real moving foreground region. The procedure of modeling pixels in a dynamic background area based on KDE is shown in Figure 4. For the pixels on  $I(x, y)$  coordinates, the foreground detection method of ViBe is used to extract a collection of its dynamic pixel samples. The set of each pixel is  $S_M = I_d(x, y, k) \mid k = 1, 2, \dots, M$ , where  $M$  is the dynamic pixels number in  $N$  frames training images. Then, the probability density curve of dynamic background samples can be derived by taking (2). The set of dynamic pixel samples is as follows:

$$I_d(x, y, k) = I(x, y, t), \quad \text{if } I_f(x, y, t) = 1 \quad (7)$$

If the probability density of the new dynamic pixel is less than the set threshold and satisfies  $f(x_{t+1}) < Th$ , then the pixel is considered as the foreground target point.

**3.2.1. Determination of Optimal Bandwidth.** Bandwidth refers to the scope in which an egocentric sample can work

in the overall probability density; therefore, it is critical to choose a proper bandwidth [34]. Elgammal et al. proposed a classic method to calculate bandwidth: For a pixel, the median absolute deviation (MAD)  $m$  of its adjacent sample values is used to calculate bandwidth dynamically [11]:

$$h = \frac{m}{0.68\sqrt{2}} \quad (8)$$

However, in this method, the absolute differences of adjacent sample points need to be ranked, which is time-consuming and does not fully utilize each sample point. Therefore, the mean of absolute differences between pixel values of adjacent sample points is used in (8) to calculate the bandwidth. The formula for calculating bandwidth is

$$h = \frac{\sum_{i=1}^{M-1} m_i / (M-1)}{0.68\sqrt{2}} \quad (9)$$

where  $m_i = |I_d(x, y, i) - I_d(x, y, i+1)|$  is the absolute difference between adjacent samples. In addition, the standard deviation of sample points can be regarded as bandwidth, as is shown below:

$$h = \sqrt{\frac{\sum_{i=1}^{M-1} (I_d(i) - \mu)^2}{M-1}} \quad (10)$$

where  $\mu$  is the sample average and  $M$  is the number of sample points. The method has strengths such eliminating the need for sorting, saving on storage space, fast computational speed, higher accuracy, and a combination of information for each sample.

In most cases, distribution of the pixel along the time axis is normal distribution, where (10) is the optimal estimation of bandwidth; otherwise, (9) has stronger adaptability. These two methods are combined in the proposed method: At the initialization stage of the algorithm, the skewness and the kurtosis of pixel point are inspected based on mathematical statistics to estimate whether it matches Gaussian distribution. Equation (9) will be adopted if the distribution of the pixel along the time axis is not normal distribution; otherwise, (10) will be taken, as it increases accuracy and robustness of bandwidth estimation.

**3.2.2. Selection of Dynamic Threshold.** Suppose that dynamic background model of the pixel  $I(x, y)$  contains  $M$  sample points, then the global probability density can be obtained through the following way. The probability of each sample  $I_d(i)$  composes kernel density probability. In the proposed method, the Gaussian function is used as the kernel function in kernel density estimation; so, for the current samples, the farther away from the center point, the less effective it is. In the same way, the whole probability density function which is composed of all samples meets the regulation, too. The probability density curve is shown in Figure 5:

The distribution model of dynamic pixel points can be considered as the weighted sum of all samples, according to (3) means. When a dynamic pixel appears, its probability can be calculated from (3), and (4) is used to determine whether it belongs to the foreground point. The threshold value to be determined can be seemed as a critical value of

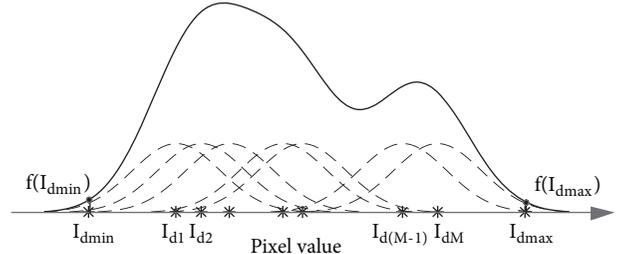


FIGURE 5: Probability fitting curve of samples and the selection of threshold.

kernel density probability function. If such critical value can be found from the total probability density function, then the dynamic threshold value can also be obtained. As shown in Figure 5, at both ends of the probability density curve, the points whose value is small enough can always be found, such as  $f(I_{dmin})$  and  $f(I_{dmax})$  which are the total probability density value of  $I_{dmin}$  and  $I_{dmax}$  in Figure 5. Therefore, the values of these two points can be used as the threshold. Pixel values of  $M$  samples are sorted from small to large, as  $I_{d1}$  to  $I_{dM}$  in the figure, then  $I_{dmin}$  and  $I_{dmax}$  can be obtained as follows:

$$\begin{aligned} I_{dmin} &= \min(I_d) - Dh \\ I_{dmax} &= \max(I_d) + Dh \end{aligned} \quad (11)$$

The total contribution of samples is considered as the Gaussian function, and all Gaussian functions have the same standard deviation  $h$  and the same weight. We set  $D=2.5$ , for a sample  $I_{di}$ ; if the distance between the observed value  $I_{dt}$  and  $I_{di}$  is larger than  $2.5h$ , according to the standard normal table  $P(|I_{dt} - I_{di}| > 2.5h) < 0.0124$ , the sample at  $I_{dt}$  has little effect on total probability density.

The sample values in the background model are sorted from small to large and the weights of  $I_{dmin}$  and  $I_{dmax}$  are the smallest of all the samples. Subsequently the total probability density of point, in which one of  $I_{dmin}$  and  $I_{dmax}$  is extended to 2.5 times bandwidth, is regarded as the threshold.  $I_{dmin}$  or  $I_{dmax}$  is plugged into (3) to calculate dynamic threshold  $Th$ . Note that, in fact, the probability densities of  $I_{dmin}$  and  $I_{dmax}$  may not be equal, because of unknown population distribution, but their difference is not significant, so either  $I_{dmin}$  or  $I_{dmax}$  can be used to calculate the threshold. Either one simply needs to pick out the minimum or the maximum value of the samples, and it is not necessary to sort all of the samples. For example, the dynamic threshold can be obtained by plugging  $I_{dmax}$  into (3),

$$Th = \frac{1}{M} \sum_{i=1}^M \frac{1}{\sqrt{2\pi}h^2} \exp\left(-\frac{(I_{dmax} - I_d(i))^2}{2h^2}\right) \quad (12)$$

**3.3. Updating the Background Model.** In this paper, there are two types of background model updating methods due to each pixel containing two kinds of background models. The static background model is used to segment the dynamic background region from the nondynamic background region. It can follow the changes of environment over time and prevent false detection of a moving object as static,

such as in the case of a background object disappearing. The ViBe update policy is used in updating the static model. The pixels in the dynamic background area are the updated target of the dynamic background model and are processed by the method that combines regular updates and real-time updates.

After initialization, the regular update of dynamic background model begins, and the interval is preestablished. Once moving pixels are detected in the dynamic background model, their values are updated to the sample set of the dynamic background model. At this time, real-time updates can make up for the background model lags of regular updates. Real-time updates can help the system update the

dynamic background model in time and eliminate false detection. Without real-time updates, the robustness of the dynamic background model is poor, and the kernel density algorithm may cause false detection, such as foreground points being detected as background points.

Instead of replacing background model points, a sample is taken randomly from  $M$  sample points and then replaced by the new pixel value of the dynamic background. Each point can be treated fairly in this method, as it eliminates the interference of subjective and time factors.

The method for updating the dynamic background model  $I_d$  in dynamic pixel point  $I_d(x, y)$  is as follows:

$$S_M(x, y, t + 1) = \begin{cases} S_{M-1}(x, y, t) \cup I_d(x, y, t), & \text{if } f(I_d(x, y, t)) < Th \text{ or } T_c > T_s \\ S_M(x, y, t), & \text{others} \end{cases} \quad (13)$$

where  $T_c$  denotes a timer,  $T_s$  is the time interval of regular updates, and  $S_{M-1}$  indicates that a sample is deleted randomly from sample set  $S_M$ . The timer  $T_c$  will be reset to zero after the update is completed.

#### 4. Experiments

In this section, we compare BOWED with the most classic or the newest methods including ViBe [10], GMM [35], KDE [11], PBAS [13], SuBSENSE [27], GMG [36], LOBSTER [28], T2-FGMM with MRF [37], and IMBS [38]. The results of different methods are achieved by running available codes or software. The test video sequences came from a live monitoring video of a pumping unit with complex outdoor environments, including waving trees or crops and video noise, particularly those consisting of the rotational pumping unit in the background scene. The characteristics of the video test sequences are shown in Table 1. Furthermore, the ground truth of each frame in the videos was annotated manually.

The performances of each background modeling methods were evaluated at pixel-level. Therefore, foreground detection can be regarded as binary classification of each pixel. The framework of the CDnet 2014 challenges [39] was applied to quantitatively measure the performance of the proposed method.

Six metrics were used for evaluation: Precision, Recall, F-measure, False positive rate (FPR), False negative rate (FNR), and Percentage of wrong classification (PWC). The metrics used to quantify performance are as follows:

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$F - \text{measure} = 2 \frac{recall \times precision}{recall + precision} \quad (16)$$

$$FPR = \frac{FP}{FN + FP} \quad (17)$$

$$FNR = \frac{FN}{TP + FN} \quad (18)$$

$$PWC = \frac{FN + FP}{TP + FN + FP + TN} \quad (19)$$

where true positive (TP) is the number of correctly detected foreground pixels and true negative (TN) is the number of correctly detected background pixels. In contrast, false positive (FP) is the number of background pixels that were incorrectly marked as the foreground pixel, and false negative (FN) is the number of foreground pixels that were incorrectly marked as the background pixel in the background subtraction method [40, 41].

To evaluate our proposal, a subset of state-of-the-art foreground detection methods was selected for comparison, which included ViBe, GMM, and KDE. For the ViBe method, we used the program which was provided by the author and the parameter setting was also suggested by the author. For other methods, we used the implementation available in the BGSlibrary [42]. The scenes with and without foreground objects contain the same scale dynamic disturbance and were detected separately.

*4.1. Scene Detection without Foreground Object.* In the surveillance of practical mining, live monitoring video is usually without any foreground objects. Therefore, the ability to adapt to the situation without foreground objects plays an important role in a foreground video detection algorithm. Low false alarm rate is one of the important indicators of an actual monitoring system, so FPR is adopted to evaluate the proposed foreground detection algorithm.

Background disturbance of video scenes in this paper is complex. Background modeling not only needs to address the interference of shaking leaves and crops and video noises, but must also address the rotation of the pumping unit, which has the same scale as foreground objects. Background disturbance in Video 1 contains pumping unit rotation and trees swaying. Video 2 has pumping unit rotation and video noises caused by the system device. Video 3 has pumping unit rotation and plants shaking. The results for detecting the dynamic background without foreground objects are presented in Figure 6. It is clear that the proposed algorithm considerably outperformed the others. In the background

TABLE 1: Characteristics of the video test sequences.

	Frame Dimensions	FPS	Number of frames	Objects
video 1	$320 \times 240$	24	1677	person
video 2	$352 \times 288$	24	1708	person and vehicle
video 3	$640 \times 480$	24	1643	persons



FIGURE 6: Scene detection in only dynamic background. (a) Original frame. (b) Segmentation of dynamic background and static background. (c) BOWED. (d) ViBe. (e) GMM. (f) KDE. (g) PBAS. (h) SuBSENSE. (i) GMG. (j) LOBSTER. (k) T2-FGMM with MRF. (l) IMBS.

modeling stage, shaking of trees and crops, noises interference of video system, and rotation of pumping unit should be considered as the background of video and should not be detected as the foreground. The video frames are processed in Figure 6, where the 520th frame was used in Video 1, the 850th frame was used in Video 2, and the 800th frame was used in Video 3. Figure 6(a) is the scene graph which contains only a complex dynamic background. Figure 6(b) expresses the segmentation image of dynamic background region and static background region, which were obtained by the proposed method. Figure 6(c) is the result of the proposed method. Figures 6(d)–6(l) are the results of the other nine methods. Because the proposed method adopts a dynamic pixel background modeling strategy, it can eliminate the influence of the background texture caused by waves on the

trees or plants shaken by the wind, and it can also eliminate the influence of the pumping unit rotation. Although other background modeling methods can eliminate the disturbance of small amplitude motion in the background, it is difficult for them to eliminate background motion interference of the same scale, such as the pumping unit rotation.

The FPR is shown in Table 2. To verify the performance of the algorithm in the detection of the dynamic background without a foreground object, morphology was not used to process the detection results. The FPR of the proposed method is no more than 0.1%, which is almost negligible, but the FPR of other mainstream background modeling methods is higher. When KDE is used for background modeling, FPR sometimes reaches as high as 20%, which indicates that the original KDE can detect dynamic pixels easily, but

TABLE 2: FPR of complex dynamic background without foreground object.

Method	video 1	video 2	video 3
BOWED	<b>0.0008</b>	<b>0.0001</b>	<b>0.0000</b>
ViBe [10]	0.0155	0.0021	0.0378
GMM [35]	0.0049	0.0067	0.0065
KDE [11]	0.2228	0.1964	0.1144
PBAS [13]	0.0058	0.0073	0.0224
SuBSENSE [27]	0.0216	0.0039	0.0371
GMG [36]	0.0554	0.0991	0.0521
LOBSTER [28]	0.0252	0.0012	0.0504
T2-FGMM with MRF [37]	0.0011	<b>0.0001</b>	0.0004
IMBS [38]	0.0519	0.0531	0.0372

TABLE 3: Comparisons of BOWED with nine other methods on the pumping unit monitoring videos.

Method	video	Precision	Recall	F-Measure	FPR	FNR	PWC
BOWED	1	<b>0.9866</b>	0.6507	0.7842	<b>0.0003</b>	0.3493	<b>0.0121</b>
	2	<b>0.9940</b>	0.6354	0.7752	<b>0.0005</b>	0.3646	0.0420
	3	<b>0.9552</b>	0.8366	<b>0.8920</b>	0.0025	0.1634	<b>0.0120</b>
ViBe [10]	1	0.5146	0.6441	0.5721	0.0213	0.3559	0.0327
	2	0.9319	0.6517	0.7670	0.0061	0.3483	0.0451
	3	0.5363	0.7331	0.6195	0.0400	0.2669	0.0535
GMM [35]	1	0.7823	0.6027	0.6808	0.0059	0.3973	0.0192
	2	0.6197	0.1423	0.2315	0.0112	0.8577	0.1077
	3	0.7413	0.3894	0.5106	0.0086	0.6106	0.0443
KDE [11]	1	0.0749	0.9401	0.1387	0.4079	0.0599	0.3961
	2	0.3682	0.8760	0.5184	0.1933	0.1240	0.1854
	3	0.3609	0.9866	0.5285	0.1103	0.0134	0.1045
PBAS [13]	1	0.7390	0.8783	<b>0.8027</b>	0.0109	0.1217	0.0146
	2	0.7939	0.7402	0.7661	0.0247	0.2598	0.0515
	3	0.6261	0.7502	0.6826	0.02883	0.2498	0.0414
SuBSENSE [27]	1	0.5196	0.9370	0.6685	0.0304	0.0630	0.0315
	2	0.8010	0.8019	0.8015	0.0256	0.1981	0.0453
	3	0.5585	0.9782	0.7111	0.0488	0.0218	0.0472
GMG [36]	1	0.3029	<b>0.9846</b>	0.4632	0.0796	<b>0.0154</b>	0.0774
	2	0.4337	<b>0.9517</b>	0.5959	0.1598	<b>0.0483</b>	0.1471
	3	0.5679	<b>0.9907</b>	0.7220	0.0476	<b>0.0093</b>	0.0453
LOBSTER [28]	1	0.4180	0.8802	0.5669	0.0430	0.1198	0.0456
	2	0.9719	0.7194	<b>0.8268</b>	0.0027	0.2806	<b>0.0343</b>
	3	0.5613	0.9472	0.7049	0.0467	0.0528	0.0471
T2-FGMM with MRF [37]	1	0.4638	0.0123	0.0239	0.0005	0.9877	0.0340
	2	0.4340	0.0020	0.0040	0.0005	0.9980	0.1140
	3	0.7001	0.0260	0.0501	<b>0.0007</b>	0.9740	0.0585
IMBS [38]	1	0.4509	0.8154	0.5806	0.0349	0.1846	0.0399
	2	0.8682	0.7479	0.8036	0.0146	0.2521	0.0417
	3	0.5294	0.9302	0.6747	0.0522	0.0698	0.0533

its ability to suppress the interference of dynamic pixels is weak. Obviously, the higher the FPR is, the easier it is to cause misinformation, which loses the original purpose of video anomaly detection. Table 2 presents results that show BOWED outperformed the others.

*4.2. Detection of Foreground Object.* Figure 7 shows the foreground segmentation results of the pumping unit monitoring videos. Figure 7(a) is the scene which has foreground objects entering the complex dynamic background. Figure 7(b) is ground truth. Figure 7(c) presents the result of BOWED method. Figures 7(d)–7(l) present the results of other state-of-the-art foreground detection methods. It is clear that the proposed algorithm outperformed the others. In particular, the ViBe algorithm was used to extract the dynamic pixels

of the video first, so the integrity of foreground object extracted using BOWED was equivalent to that of ViBe. However, BOWED was able to filter the perturbation of the dynamic background better, and the simple ViBe still considered the rotation of the pumping unit as a foreground target. Notable results in the other state-of-the-art algorithms were as follows: Although KDE had the most complete extraction of foreground target, it mistook many of the dynamic background pixels as foreground target pixels, so its FPR was very high. T2-FGMM with MRF had a low FPR but could not detect the object correctly.

The results of the quantitative comparison are listed in Table 3. The larger the Precision, Recall, and F-measure values and the smaller the FPR, FNR, and PWC values, the better the segmentation. The best scores are highlighted in bold. It

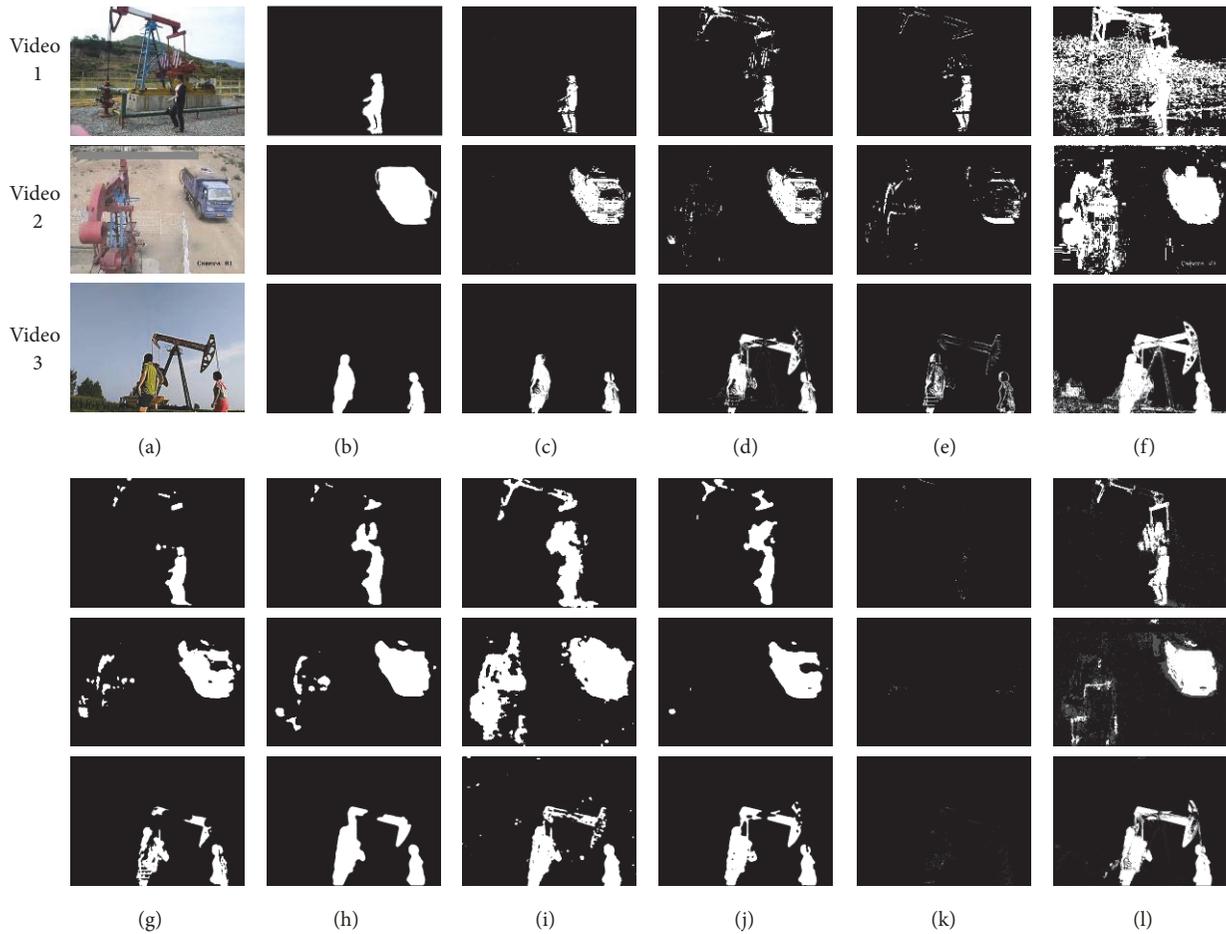


FIGURE 7: Detection of foreground object. (a) Original frame. (b) Ground truth. (c) BOWED. (d) ViBe. (e) GMM. (f) KDE. (g) PBAS. (h) SuBSSENSE. (i) GMG. (j) LOBSTER. (k) T2-FGMM with MRF. (l) IMBS.

can be seen that the proposed method obtained lower FPR and PWC scores and higher Precision scores. It shows that the proposed method detected fewer incorrect pixels and achieved higher accuracy in pixels which were detected as foreground. For the FPR scores, the T2-FGMM with MRF method and BOWED had nearly the same score; however, using the T2-FGMM with MRF method, it was difficult to obtain the foreground object. For the F-measure scores, the proposed method is performed as well as PBAS and LOBSTER methods and better than other methods. These results indicate that BOWED can achieve a better global effect. Because our goal is to reduce false alarm rate as much as possible under the premise of ensuring accuracy in foreground object detection. Although BOWED did not give the best Recall and FNR scores, it inaccurately detected a large quantity of background pixels as foreground pixels. In that case, the pseudo foreground pixels were almost superimposed together, so that even the morphological method could not be used to separate the foreground object from the background. BOWED not only separates the foreground object from the background with ease but also has a very low false alarm rate, which achieves the main purpose of intelligent video surveillance.

## 5. Conclusions

This paper proposed a novel dynamic background modeling approach that was applied to same scale movement object interference for foreground detection. In the proposed method, the background pixels were divided into two types: dynamic background pixels and static background pixels. First, ViBe was used to extract the value of each pixel that was detected as foreground, and then KDE was used to model the detected foreground pixels to obtain the real foreground target pixels. The selection of bandwidth for KDE combines the advantages of the mean of the adjacent sample difference and the standard deviation of the sample, and the random extraction strategy was adopted when replacing the sample. For the strategy that combined timing and real-time updates within a given time interval, only calculation of kernel density was needed, as a calculation of threshold was not necessary due to the background model being updated in real-time when a foreground object was present. The proposed method was evaluated for highly dynamic backgrounds of complex industrial scenes which contained the periodic motion of a rotating pumping unit. The experimental results showed that our method can obtain satisfactory performance.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China 12th Five-Year Plan (no. 2011ZX05039).

## References

- [1] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, 2014.
- [2] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using a nonparametric bayesian model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [3] S. Calderara, R. Cucchiara, and A. Prati, "A distributed outdoor video surveillance system for detection of abnormal people trajectories," in *Proceedings of the 2007 First ACM/IEEE International Conference on Distributed Smart Cameras, ICDS-C-07*, pp. 364–371, Austria, September 2007.
- [4] C.-R. Huang, P.-C. J. Chung, D.-K. Yang, H.-C. Chen, and G.-J. Huang, "Maximum a posteriori probability estimation for online surveillance video synopsis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1417–1429, 2014.
- [5] R. Jain and H. H. Nagel, "On the analysis of accumulative difference pictures from image sequences of real world scenes," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. PAMI, pp. 206–214, 1979.
- [6] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.
- [7] S. D. Babacan and T. N. Pappas, "Spatiotemporal algorithm for background subtraction," in *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '07*, pp. 11065–11068, USA, April 2007.
- [8] T. Bouwmans, "Background Subtraction for Visual Surveillance," in *Handbook on Soft Computing for Video Surveillance*, vol. 20123846 of *Chapman & Hall/CRC Cryptography and Network Security Series*, pp. 103–138, Chapman and Hall/CRC, 2012.
- [9] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 246–252, June 1999.
- [10] O. Barnich and M. Van Droogenbroeck, "ViBe: a powerful random technique to estimate the background in video sequences," in *Acoustics, Speech and Signal Processing*, pp. 945–948, 2009.
- [11] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Computer Vision—ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part II*, vol. 1843 of *Lecture Notes in Computer Science*, pp. 751–767, Springer, Berlin, Germany, 2000.
- [12] H. Wang and D. Suter, "A consensus-based method for tracking: Modelling background scenario and foreground appearance," *Pattern Recognition*, vol. 40, no. 3, pp. 1091–1105, 2007.
- [13] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2012*, pp. 38–43, June 2012.
- [14] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 11–12, pp. 31–66, 2014.
- [15] H. Zhang and K. Wu, "A vehicle detection algorithm based on three-frame differencing and background subtraction," in *Proceedings of the 2012 5th International Symposium on Computational Intelligence and Design, ISCID 2012*, pp. 148–151, China, October 2012.
- [16] N. Singla, "Motion detection based on frame difference method," vol. 4, pp. 1559–1565, 2014.
- [17] J. S. Kulchandani and K. J. Dangarwala, "Moving object detection: Review of recent research trends," in *Proceedings of the 2015 International Conference on Pervasive Computing, ICPC 2015*, India, January 2015.
- [18] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1937–1944, Colorado Springs, Colo, USA, June 2011.
- [19] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV'99)*, vol. 1, pp. 255–261, September 1999.
- [20] M. M. Alan, "Background subtraction techniques," *Proc of Image Vision Computing*, vol. 2, pp. 1135–1140, 2000.
- [21] T. Bouwmans, F. E. Baf, and B. Vachon, "Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey," *Recent Patents on Computer Science*, vol. 1, no. 3, pp. 219–237, 2008.
- [22] T.-S. Le, N.-T. Do, and K. Hamamoto, "Speed up temporal median filter and its application in background estimation," in *Proceedings of the 12th IEEE RIVF International Conference on Computing and Communication Technologies: Research, Innovation, and Vision for the Future, RIVF 2016*, pp. 175–180, Viet Nam, November 2016.
- [23] M. Piccardi, "Background subtraction techniques: a review," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 3099–3104, Hague, The Netherlands, October 2004.
- [24] W. Liu, Y. Cai, M. Zhang, H. Li, and H. Gu, "Scene background estimation based on temporal median filter with Gaussian filtering," in *Proceedings of the 23rd International Conference on Pattern Recognition, ICPR 2016*, pp. 132–136, Mexico, December 2016.
- [25] 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, 2011.
- [26] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for ViBe," in *Proceedings*

- of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2012, pp. 32–37, USA, June 2012.
- [27] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, “Flexible background subtraction with self-balanced local sensitivity,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2014*, pp. 414–419, USA, June 2014.
- [28] P.-L. St-Charles and G.-A. Bilodeau, “Improving background subtraction using Local Binary Similarity Patterns,” in *Proceedings of the 2014 IEEE Winter Conference on Applications of Computer Vision, WACV 2014*, pp. 509–515, March 2014.
- [29] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1162, 2002.
- [30] A. Tavakkoli, M. Nicolescu, and G. Bebis, “Automatic statistical object detection for visual surveillance,” in *Proceedings of the 7th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 144–148, USA, March 2006.
- [31] W. Wang, Y. Cheng, L. Hong et al., “Violent video detection based on MoGLOH feature,” in *Proceedings of the Optical Sensing and Imaging Technology and Applications*, p. 55, Beijing, China, June 2017.
- [32] J. Lee and M. Park, “An adaptive background subtraction method based on kernel density estimation,” *Sensors*, vol. 12, no. 9, pp. 12279–12300, 2012.
- [33] Y. Xu, J. Dong, B. Zhang, and D. Xu, “Background modeling methods in video analysis: A review and comparative evaluation,” *CAAI Transactions on Intelligence Technology*, vol. 1, no. 1, pp. 43–60, 2016.
- [34] V. Katkovnik and I. Shmulevich, “Nonparametric density estimation with adaptive varying window size,” in *Conference on Image and Signal Processing for Remote Sensing VI, European Symposium on Remote Sensing*, pp. 25–29, 2001.
- [35] P. Kaewtrakulpong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” in *Proceedings of the 2nd European Workshop on Advanced Video Based Surveillance System, AVBS01*, pp. 135–144, 2001.
- [36] A. B. Godbehere, A. Matsukawa, and K. Goldberg, “Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation,” in *Proceedings of the 2012 American Control Conference, (ACC '12)*, pp. 4305–4312, IEEE, Montreal, QC, Canada, June 2012.
- [37] Z. Zhao, T. Bouwmans, X. Zhang, and Y. Fang, “A Fuzzy Background Modeling Approach for Motion Detection in Dynamic Backgrounds,” in *Multimedia and Signal Processing*, vol. 346 of *Communications in Computer and Information Science*, pp. 177–185, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [38] D. Bloisi and L. Iocchi, “Independent multimodal background subtraction,” in *Proceedings of the Third International Conference on Computational Modeling of Objects Presented in Images: Fundamentals, Methods and Applications*, vol. 413, pp. 39–44, 2012.
- [39] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, “CDnet 2014: an expanded change detection benchmark dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '14)*, pp. 393–400, IEEE, Columbus, Ohio, USA, June 2014.
- [40] Z. Zhong, B. Zhang, G. Lu, Y. Zhao, and Y. Xu, “An Adaptive Background Modeling Method for Foreground Segmentation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1109–1121, 2017.
- [41] L. Maddalena and A. Petrosino, “A self-organizing approach to background subtraction for visual surveillance applications,” *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168–1177, 2008.
- [42] A. Sobral, “Bgslibrary: An opencv c++ background subtraction library,” in *1x Workshop De Visao Computacional*, pp. 32–37, An opencv c++ background subtraction library, Bgslibrary, 2013.

