



**HAL**  
open science

# INBaSA: Novel Non-parametric background subtraction algorithm based on and improved pixel based segmentation

Lionel Landry Sop Deffo, Elie Fute Tagne

► **To cite this version:**

Lionel Landry Sop Deffo, Elie Fute Tagne. INBaSA: Novel Non-parametric background subtraction algorithm based on and improved pixel based segmentation. CARI, Oct 2022, Dschang, Cameroon. hal-03816276

**HAL Id: hal-03816276**

**<https://hal.science/hal-03816276v1>**

Submitted on 25 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# INBaSA: Novel Non-parametric background subtraction algorithm based on and improved pixel based segmentation

LIONEL LANDRY SOP DEFFO\*<sup>1</sup> and ELIE FUTE TAGNE<sup>1,2</sup>

<sup>1</sup>University of Buea, Cameroon

<sup>2</sup>University of Dschang, Cameroon

\*E-mail : [lsopdeffo@yahoo.com](mailto:lsopdeffo@yahoo.com)

---

## Abstract

Applications dealing with images have increased recently. These include video surveillance, person identification, pedestrian detection, tracking, etc. All these operations need to be performed following a certain number of steps of which one of the most important is background subtraction used for motion detection. Despite improvements in several proposed approaches, there is still demand for better performance from users. It is for this reason that, many algorithms have captured our attention among which is the ViBE-based algorithm called EFF-ViBE. While the first (ViBE) suffers from two major drawbacks which are; the apparition of ghost phenomenon as well as poor quality of results in case of complex background. The second (EFF-ViBE) on the other hand while trying to solve those drawbacks introduces a large number of false positives and false negatives and therefore affects the results. This paper presents an approach called INBaSA to address the problems mentioned above. We have introduced new factors such as the mean of temporal gradients, and adaptive radius with offset values, change the pixel counting threshold, and modified the segmentation process.

## Keywords

Background model ; Background subtraction ; Mean of temporal gradients ; Motion detection ; Offset ; Segmentation

---

## I INTRODUCTION

In most video processing applications, the first step is to detect the moving object in a scene in consecutive video frames. In the case of pedestrian detection, for example, we need to know the position of the pedestrian in the scene by using an average of images over time [1]. The idea is that we separate the object in movement from the static one and this is done using a technique called background subtraction. This gave rise to the proposition of different background subtraction algorithms that can be grouped into many categories.

A complex background appears in a situation where the targeted object cannot be detected clearly and intuitively [2]. It can also be seen as a situation where clutter has been introduced in the background [3] and therefore leads to an influence on the performance of the detection algorithm. This study made us notice that a more complex background model leads to more parameters to handle, but in some cases, the estimation of the parameters of the model can

become problematic for noisy images . That is why we will focus our study on sample-based techniques which build their models from observed pixel values [4].

The rest of the paper is organized as follows, previous studies are first of all made in Section II, and in Section III we give details on INBaSA background subtraction method. An implementation is done in Section IV and results are interpreted. We finish with Section V that concludes the work by reminding the global objectives, and work done and gives some reflections which could be explored in the future.

## II RELATED WORK

Basically, the optimization in background segmentation tries to solve three main categories of problems namely; ghost phenomenon, shadow removal, and hardware acceleration for real-time applications. This is intended to ameliorate the accuracy of results.

### 2.1 Ghost elimination and Background interference elimination approaches

They consist of approaches that try to eliminate the ghost phenomenon which is, the apparition of fictive moving objects due to the complexity of the background and sometimes we also have situations where interferences are introduced in the background due to factors such as the movement of trees, waves at a sea surface, etc. Amongst them, we have the fusion method. This method tries to bring together information about the depth and color to ensure foreground segmentation using algorithms based on color [5]. In [6], the authors have proposed a method that accelerates the elimination of the ghost area by detecting and reinitializing the region of the ghost area and have shown by experiments that frame processing time can still meet the demand of real-time performance. Moreover, a salient region detection-based ViBE algorithm has been proposed [7]. Its main goal was to quickly suppress ghost areas which is why the foreground region is firstly removed from the initial frame. This is assured by an algorithm based on salient region detection. Based on the obtained results, the background is kept separate from the foreground. Consequently, pixels belonging to the foreground area The foreground pixels will see their position changed in the sample set. This is intending to uniquely use background pixels to initialize the background model. In [8], a motion detection method called optical flow ViBE (OF-ViBE) is proposed. It combines an improved ViBe algorithm and motion information to remove the ghost and noise points more accurately and to reduce the dynamic background interferences using direction information. In addition, a method to remove noise in the image in marine surveillance has been proposed in [9] to reduce the interferences introduced in the background model due to waves at the sea surface. Another update mechanism consists of a first-in-first-out model update policy such as those employed in [10] or [11]. A certain number of samples (300 in this case) is used to handle a window of time 10 seconds at a frame rate of 30 frames per second. As a consequence, a pixel moving slowly over a period greater than 10 seconds will see itself still inserted in the background model. More recent techniques such as [12] suggest that one frame alone cannot distinguish whether pixels belong to the foreground or background. Therefore, they propose to use the first  $n$  (with  $n=20$  in the paper) frames of the video sequence to complete the initialization of the background model. These frames are used to compute the mean value of each pixel we added to the sample values to have  $N + 1$  sample values. Consequently each pixel is modeled by

$$M = \{v_0(x), v_1, v_2, \dots, v_N\} \quad (1)$$

where

$$v_0(x) = \frac{1}{n} \sum_{t=1}^n v^t(x) \quad (2)$$

## 2.2 Shadow removal approaches

Here the problem of the shadow is tackled by trying to remove it as much as possible. In [13] a method for robust motion segmentation has been proposed. Its main composition includes two models which are related. One of them called N-model is a normal random model while the other one called E-model is also a random model which has been enhanced. These two models are obtained and maintained to cope with changes in illumination and non-static background. The information used is Spatio-temporal and the strategy used is Ada-Boost-like. This method called DMSTAB is a Dual-model Motion segmentation approach employing Spatio-Temporal information based on AdaBoost-like strategy and ViBE. Other authors such as [14], believe that for complex dynamic backgrounds, the radius  $R$  should be increased appropriately so that the background cannot be easily detected as the foreground. On the other hand, they also believe that for simple static background, the radius  $R$  should see its value reduced to detect negligible variations in the foreground. That is why they propose to adapt the radius  $R$  of the ViBE algorithm using the formulas: If  $R_{adaptive}$  denote the adaptive radius

$$R_{adaptive} = \begin{cases} R*(1 + \varepsilon_c), & \text{if } R \leq d_{mean} * \delta \\ R*(1 - \varepsilon_d), & \text{else} \end{cases} \quad (3)$$

where  $\varepsilon_c = 0.06$ ,  $\varepsilon_d = 0.4$ ,  $\delta = 5$  and

$$d_{mean} = \sum_{i=1}^N D^i(x) \quad (4)$$

where  $D^i(x)$  is the distance between pixel  $x$  intensity value  $v(x)$  and sample value  $v_i$ .

## 2.3 Hardware optimization approaches

Here hardware optimization components such as GPU are used to enhance and foster the execution. In this view, a CUDA implementation of the ViBE (Visual Background Extraction) background subtraction algorithm on NVIDIA Jetson TX1 / TX2 modules has been proposed in [15]. The optimizations proposed enables a real-time execution on Jetson TX1 / TX2 modules with 720p.

We can see in this classification that each time an approach tries to solve one or many challenges leads to a cascade. At times, trying to solve one challenge has as consequence the resolution of another one in a moderate manner. Table 1 gives a summary comparison on the challenges that each approach tries to solve. Apart from this classification, further reading can be made in [16], [17], and [18].

## III PRESENTATION OF INBASA BACKGROUND SUBTRACTION TECHNIQUE

We present in this section the details of the INBaSA approach. It is derived from a ViBE-based approach named EFF-ViBE that uses the first  $n$  frames to construct the background model. The major modifications are the introduction of the mean of temporal gradients (mTG), which leads

Table 1: Comparison table of approaches related to ViBE

Algorithm	Background interferences elimination	Shadow elimination	Ghost elimination	Hardware optimization
(X. Zhou et al., [19])	Moderate	High	High	Moderate
(K. Cheng et al., [6])	Moderate	No	High	Moderate
(Y. Zhang et al., [7])	Moderate	No	High	Moderate
(L. Zhao et al., [8])	High	No	High	Moderate
(D. Adiguzel et al., [9])	High	No	Low	Moderate
(A. Elgammal et al., [20])	High	High	High	Moderate
(H. Wang et al., [11])	High	No	Low	Moderate
(H. Botao et al., [12])	Moderate	No	High	Moderate
(L. Chang et al., [14])	High	Moderate	Moderate	Moderate
(Z. Fan et al., [13])	Moderate	Moderate	Moderate	Moderate
(M. Karagoz et al., [15])	Moderate	No	Low	High
(E. Fute et al., [21])	High	Moderate	High	Moderate

to the modification of parameters such as the adaptive radius as well as the number of frames used to initialize the construction of the background model. Since it is still a ViBE-based approach, it has three well-known phases which are the initialization phase, the segmentation phase, and the update phase. The introduction of the mean of temporal gradients was inspired by the work of [22] which allows the location of stable and unstable regions in a video. We are going to detail that in the following sections.

Since INBaSA is a ViBE-based algorithm, its overall functioning follows the ViBE schema which consists of three parts namely background model initialization, background segmentation, and background model update. Here we try to introduce and modify the adaptive radius according to the stable and unstable parts in the scene determined using the mean of temporal gradients which is computed as follows. Let  $I_k$  be a video frame of size  $P \times Q$  at time  $k$  in a video stream  $\{I_k\}_{k=1}^V$ . The pixel coordinates of image  $I_k$  is represented as  $I_k(a, b) \forall a \in [1, P]$  and  $\forall b \in [1, Q]$  and  $V$  is the length of the video. Then mTG can be computed as Equation 5.

$$mTG(a, b) = \frac{1}{F_s - 1} \sum_{k=2}^{F_s} |I_k(a, b) - I_{k-1}(a, b)| \quad (5)$$

where  $F_s$  is the total number initial frames selected for parameter initialization.

As it can be seen in Figure 1, the steps are the same as it is in EFF-ViBE with the difference that rather than using  $d_{mean}$ ,  $\varepsilon_c$ ,  $\varepsilon_d$ , and  $\sigma$  to compute the adaptive radius, we instead use the  $mTG$  value for each pixel because it helps us to determine the stable and unstable region in the scene. The other parameters such as the number of samples, the counter parameter, and the *change* parameter will still be used. Let us now look at those steps in detail.

### 3.1 INBaSA Background model initialization

The following three techniques were used. Firstly, the ViBE initialization process that models each background pixel with  $N$  sample values taken from their direct neighborhood. Secondly, the first  $n$  frames were used to compute the mean value of each pixel and this value added to the sample values to have  $N + 1$  sample values so as to have a pixel modeled by

$$M = \{v_0(x), v_1, v_2, \dots, v_N\} \quad (6)$$

where

$$v_0(x) = \frac{1}{n} \sum_{t=1}^n v^t(x) \quad (7)$$

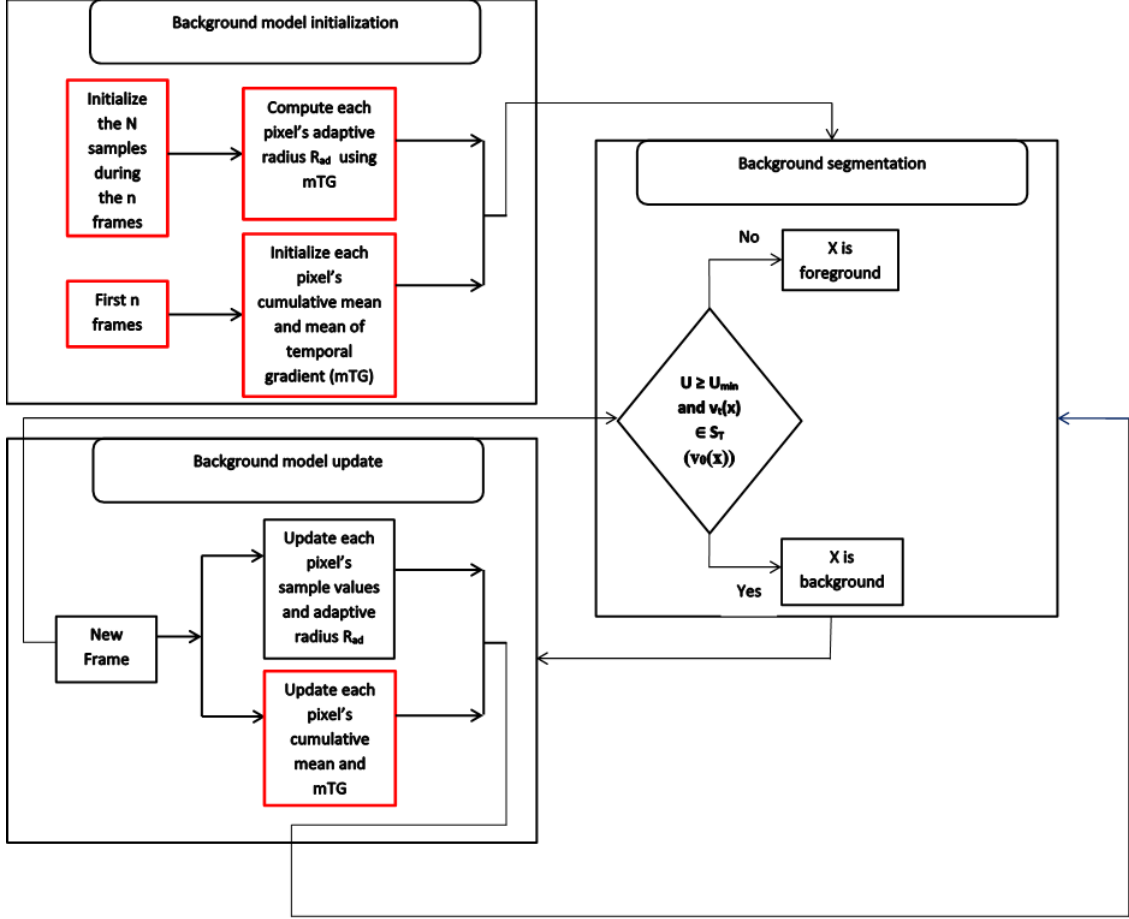


Figure 1: General architecture INBaSA approach based on EFF-ViBE [21]

However the value of  $n$  has been set to 300 rather than 30 as in EFF-ViBE. The following equation is used for any given pixel  $x$ .

$$R_{ad}(x) = mTG(x) + \alpha \quad (8)$$

where  $\alpha$  represents the matching threshold value when comparing pixel values to have the number of matching pixels in the circle of radius  $R$  and in the case of ViBE algorithms the value in practice is equal to 20. Also  $mTG$ , is computed using Equation 5.

Note that the choice of 20 as the value of  $R$  is guided by previous ViBE-Based approaches and empirical studies. Also, it can be noted that the newly inserted  $R_{ad}$  differs from that of EFF-ViBE because it uses the mean of temporal gradient rather than shifting parameters. The same goes for other parameters such as  $\sigma$ ,  $counter_{max}$  and  $n$

### 3.2 INBaSA Background segmentation

The segmentation process follows the same steps as it is in ViBE with additional parameters. These steps include the computation of an adaptive radius and the intersection of the control sphere with sample values. Added to that the following notation  $B(x)$  is used, which is the value of the pixel  $x$  in matrix  $B$  where it is equal to 0 if  $x$  is a background pixel and 255 if  $x$  is a foreground pixel, the following rules are applied for classification :

$$B(x) = \begin{cases} 0, & \text{if } U \geq U_{min} \text{ and } v^t(x) \in S_T(v_0(x)) \\ 255, & \text{else} \end{cases} \quad (9)$$

$$U = |S_{R_{ad}}(v^t(x)) \cap \{v_1, v_2, \dots, v_N\}| \quad (10)$$

Where  $v^t(x)$  is the value of pixel  $x$  at time  $t$

$$T = \begin{cases} R_{ad}, & \text{if } \sigma \leq \beta * R_{ad} \\ \sigma/\beta, & \text{else} \end{cases} \quad (11)$$

$$\sigma = \sqrt{\sum_{t=1}^n (v^t(x) - v_0(x))^2} \quad (12)$$

Where parameters  $T$  and  $\beta$  are values that reflect the historical scene change intensity, while value of  $U_{min}$  is maintained to 2, that of  $\beta$  is assigned a value of 3.

### 3.3 INBaSA Background model update

Using the same assumptions of the ViBE algorithm, we update our model. The first one considers that background pixels in the neighborhood share the same temporal distribution and that a new sample value of pixel added to the background model should also update the pixels models in their respective neighborhood leading to the use of the same memory-less update and the time sampling policy.

Secondly, the same pixel counting mechanism that classifies a pixel that stays in the foreground within  $K$  consecutive frames as a background pixel is used leading to:

$$B(x) = \begin{cases} 0, & \text{if } count(x) > counter_{max} \\ B(x), & \text{otherwise} \end{cases} \quad (13)$$

$$count(x) = \begin{cases} count(x) + 1, & \text{if } B(x) = 255 \\ & \text{and } count(x) \leq counter_{max} \\ 0, & \text{if } count(x) > counter_{max} \end{cases} \quad (14)$$

Finally  $\sigma$  and  $v_0(x)$  are updated using the following equations.

$$\sum_{t=2}^{n+1} v^t(x) = \sum_{t=1}^n v^t(x) + v^{n+1}(x) - v^1(x) \quad (15)$$

$$\sum_{t=2}^{n+1} (v^t(x))^2 = \sum_{t=1}^n (v^t(x))^2 + (v^{n+1}(x))^2 - (v^1(x))^2 \quad (16)$$

Denoting by  $v_0^t(x)$ ,  $\sigma^t$ ,  $v_0^{t+1}(x)$  and  $\sigma^{t+1}$  the value of the cumulative mean and standard deviation at time  $t$  and  $t + 1$  respectively, its update will be done following

$$v_0^{t+1}(x) = \frac{1}{n} \sum_{t=2}^{n+1} v^t(x) \quad (17)$$

Table 2: Parameters used in our implementation

Parameters	values
Number of frames ( $n$ )	300
$U_{min}$	2
$\alpha$	20
$\beta$	3
Time sampling $\phi$	16
$counter_{max}$	50

$$\sigma^{t+1} = \sqrt{\left(\sum_{t=2}^{n+1} (v^t(x))^2\right) - (v_0^{t+1}(x))^2} \quad (18)$$

In addition, to compute at any moment the value of  $mTG$ , we need to keep track of pixel differences, that why a matrix  $M$ , containing the sum of differences used to compute  $mTG$  values. So to update this matrix value for each pixel we need to remove the first difference and add the new one that is for a given pixel  $x$ :

$$M_{t+1}(x) = M_t(x) - |I_2(x) - I_1(x)| + |I_{t+1}(x) - I_t(x)| \quad (19)$$

Where

$$M_t(x) = \sum_{k=2}^n |I_k(x) - I_{k-1}(x)| \quad (20)$$

#### IV RESULTS AND DISCUSSION

We used C and C++ languages for the implementation of the proposed algorithm. Once the source code was compiled we used a wrapper to create an executable in python for all algorithms used in the test phase. This was to benefit from powerful functions to handle images found in libraries such as OpenCV, NumPy, Images, etc. This was done on a Linux operating system, more specifically on Fedora 31. The dataset used to measure the performance of the algorithms is the one provided by the [changedetection.net](http://changedetection.net) dataset [23] and the parameters used for our simulation are gathered in Table 2

Since it is a ViBE-based approach, we have adapted the code [24] to implement the INBaSA approach. The algorithms implemented and tested were respectively original ViBE(only test) [4], EFF-ViBE [21], and INBaSA. We have also compared the results with popular background subtraction algorithms such as Euclidean distance [25], Mixture of Gaussian (MoG) [26], Kernel density estimation (KDE) [20], self-organized background subtraction (SOBS) [25] and SUBSENSE [27].

To measure the performance parameters, we have used the source code available on the [changeDetection.net](http://changeDetection.net) platform. Since the objective was to highlight the differences in results obtained by the proposed INBaSA and other ViBE-related algorithms and we have used the results published on their platform for the other algorithms. The following results were obtained.



#### 4.1 Fast elimination of ghost phenomenon

The first drawback of the original ViBE algorithm was the presence or the apparition of the ghost phenomenon when the initial frame used to construct the background model was containing a moving object. To initiate a solution for this problem we proposed in [21] to use the  $n$  first frames to initialize the background and it has been shown to give better results with a possibility of improvements. It is in this view that we come up with this improvement by introducing the mean of temporal gradient and using 300 frames rather than 50 to initialize the background model. The general remark is that the shadow is first of all globally reduced, this can be seen on different 1900<sup>th</sup> and 2000<sup>th</sup> frames where the quantity of shadow is significantly reduced for the INBaSA approach compared to the other ViBE based approaches namely ViBE and EFF-ViBE. Secondly, INBaSA surpasses the previously improved EFF-ViBE in the sense that at frame 2100 while we still have some noise created by the ghost phenomenon in EFF-ViBE (even if it is significantly reduced already), it has completely been removed in INBaSA which can be seen as an improvement.

#### 4.2 Effect of adaptive radius

The second challenge was to adapt the radius to the variability of the background model in order to have a more efficient segmentation mechanism. For that, we have proposed an adaptive radius principle in [21] that adapts the radius according to certain factors  $\varepsilon_c$ ,  $\varepsilon_d$  and  $\delta$  notably. The same philosophy is adopted in INBaSA using now the introduced mean of temporal gradient and has proven to be more efficient. This can be observed in Figure 3 where the impact of the adaptive radius is already visible on both EFF-ViBE and INBaSA approaches compared to the original ViBE algorithm. In addition, the details given by INBaSA in terms of moving object detection are better than the one given by EFF-ViBE. These details include the shape of the boat, and the people within the boat, while in EFF-ViBE there are still some black holes in EFF-ViBE, but this was significantly removed in INBaSA (last line of Figure 3)

#### 4.3 Measure of performances

Here, we have recorded the F-measure of eight algorithms: the original ViBE algorithm [4], EFF-ViBE [21] algorithm, the proposed INBaSA method, and some challenging algorithms such as Euclidean distance [25], MOG [26], KDE [20], SOBS [25] and SUBSENSE [27]. The results are shown in Table 3. In each case, the best value is highlighted on a single row using bold and underlined font style while the second-best value is highlighted using underlined font only.

The datasets used are those of CDNet 2014 and to have the results we have used the utility code available on the change detection platform to compute those values. The process is as follows, we use the python file ProcessFolder.py to execute the algorithm chosen. Note that this process was used to compute values for ViBE-based approaches namely ViBE, EFF-ViBE, and INBaSA. For the other algorithms, we have just taken the values presented on the change-detection platform. Mathematically, the precision  $Pr$ , recall  $R$  and the F-measure  $F$  are computed using the formula

$$precision : (Pr) = \frac{TP}{TP + FP} \quad (21)$$

$$Recall : (R) = \frac{TP}{TP + FN} \quad (22)$$

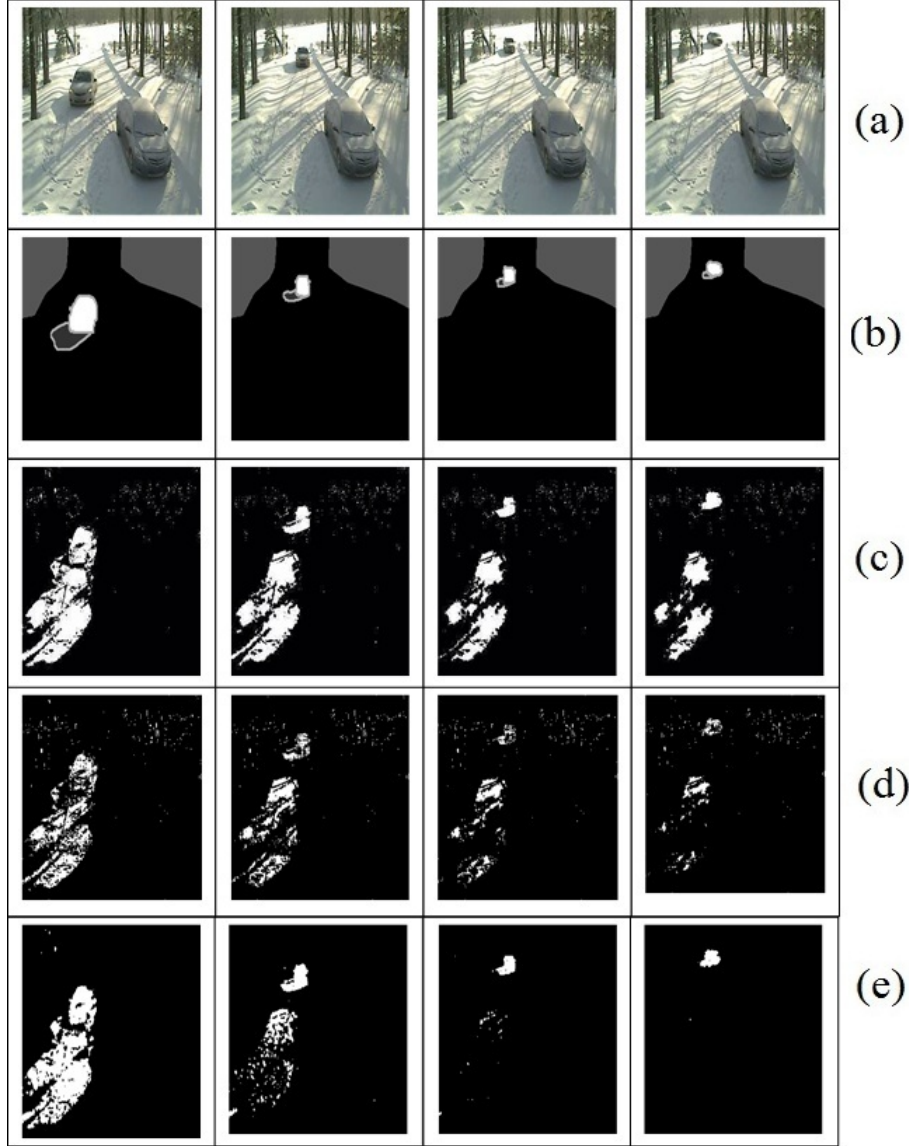


Figure 2: A target moves from a stationary position (winter Driveway). (a) the 1900<sup>th</sup>, 2000<sup>th</sup>, 2100<sup>th</sup>, 2200<sup>th</sup> frame; (b) ground truth images provided by the dataset; (c) the detection results of the original ViBE; (d) the detection results of EFF-ViBE approach; (e) the detection results of INBaSA proposed method.

$$F - measure : (F) = \frac{2 * Pr * R}{Pr + R} \quad (23)$$

where  $TP$  denotes the number of foreground pixels truly classified as foreground,  $FP$  denotes the number of background pixels wrongly classified as foreground and  $FN$  the number of foreground pixels wrongly classified as background.

Observing Table 3, INBaSA outperforms EFF-ViBE with a percentage of 90.90% is 10 categories on 11 except PTZ which was the initial objective. For the specific case of SUBSENSE, this outperforming percentage is only 9.09% which is 1 category (turbulence) out of 11. The comparison with the rest of the algorithms gives an overall percentage of 54.54% which is 6 categories on 11. It is important to mention that, even though SUBSENSE has good results it is

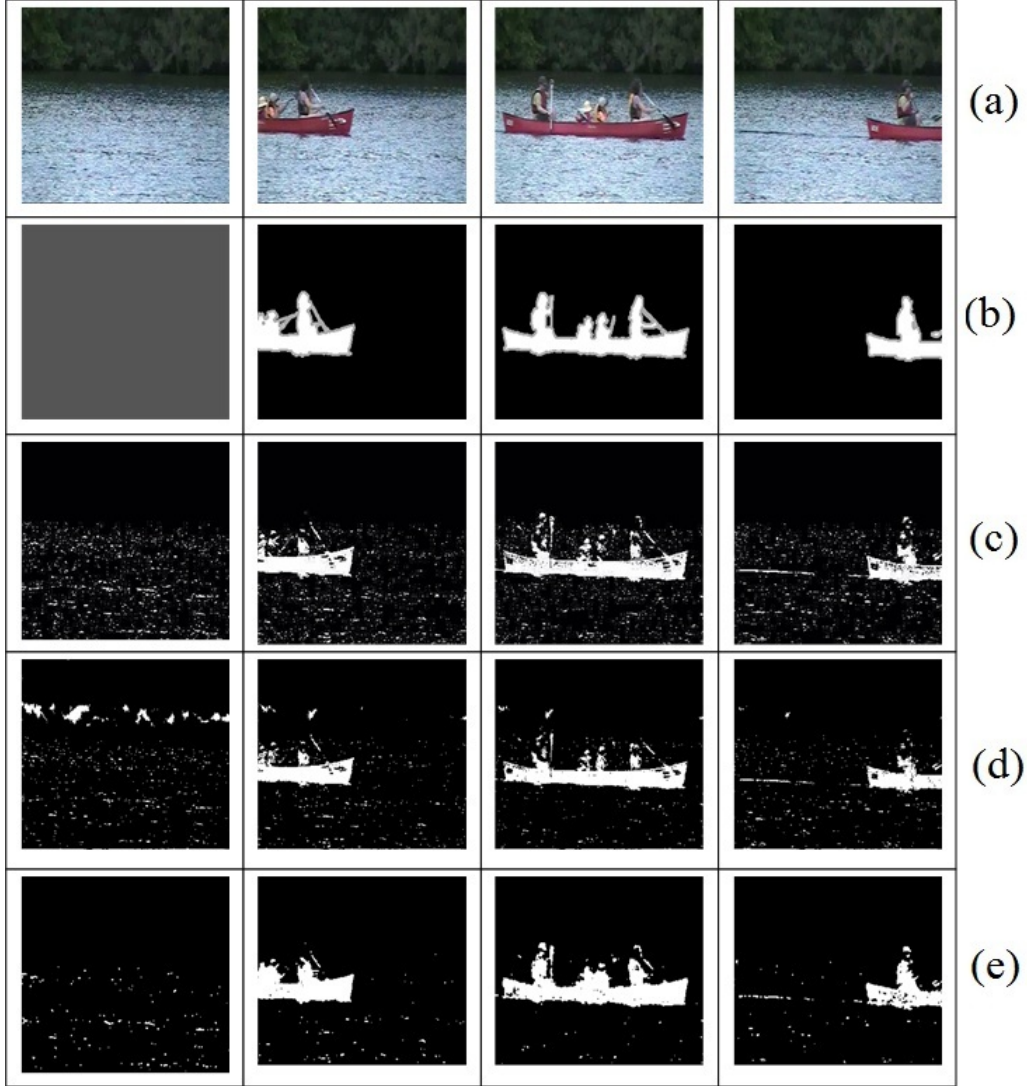


Figure 3: Effect of the adaptive radius (canoe). (a) the 600<sup>th</sup>, 900<sup>th</sup>, 960<sup>th</sup>, 1040<sup>th</sup> frame; (b) the real background corresponding with (a); (c) the detection results of the original ViBE; (d) the detection results of improved version EEf-ViBE approach; (e) the detection results of INBaSA proposed method.

Table 3: Recapitulating table of F-Measure

Category	Euclidian	MoG	KDE	SOBS	SUBSENSE	ViBE	EFF-ViBE	INBaSA
Bad weather	0.670	0.740	0.757	0.684	<b>0.861</b>	0.620	0.768	<u>0.802</u>
Low F.R.	0.501	0.506	0.547	0.464	<b>0.644</b>	0.451	0.556	<u>0.563</u>
Night V	0.385	0.396	0.436	<u>0.446</u>	<b>0.559</b>	0.383	0.3400	0.400
PTZ	0.039	0.104	0.036	0.138	<b>0.347</b>	0.102	<u>0.247</u>	0.217
Turbulence	0.413	0.416	0.447	0.158	0.779	<b>0.796</b>	0.527	<u>0.794</u>
Baseline	0.872	0.838	<u>0.909</u>	0.608	<b>0.950</b>	0.786	0.766	0.828
Dynamic B.	0.508	0.632	0.596	0.161	<b>0.817</b>	0.617	0.576	<u>0.720</u>
Camera J.	0.487	0.567	0.572	0.414	<b>0.815</b>	0.607	0.635	<u>0.680</u>
Intermittent O.M.	0.489	<u>0.532</u>	0.408	0.302	<b>0.656</b>	0.243	0.425	0.491
Shadow	0.611	0.723	<u>0.766</u>	0.752	<b>0.864</b>	0.724	0.731	0.740
Thermal.	0.631	0.654	<u>0.742</u>	0.409	<b>0.817</b>	0.509	0.511	0.618

not yet suitable for real-time applications in the sense that it takes greater time for its execution. If we take the example in our experiment we notice that SUBSENSE has an average speed of

16fps while INBaSA has an average speed of 58fps. It, therefore, means that INBaSA is more suitable than SUBSENSE as far as real-time applications are concerned. The average speeds were determined as followed; under the same experimental conditions, different algorithms are executed on the same number of frames. In the end, the total execution time of each algorithm is divided by the total number of processed frames.

$$fps_{algorithm\ x} = \text{execution time of } x / \text{total number of processed frames} \quad (24)$$

## V CONCLUSION

In this study, we have demonstrated that the ViBE background subtraction algorithm can be further improved. For this purpose, we introduced three main mechanisms which are the mean of temporal gradient(mTG), the adaptive radius with an offset value, and the modification of pixel counting threshold and called the new algorithm INBaSA. This led to a modification of the initialization process as well as the segmentation process. We did experiments to verify the theory stated followed by a quantitative as well as a qualitative analysis of this result. This analysis has proven that INBaSA outperforms the EFF-ViBE with an overall percentage of 90.90%. In addition, the INBaSA algorithm has been compared to some popular algorithms and outperforms them as well with an overall percentage of 54.54%. So it can be said with evidence that the objective has been attained. However, the overall percentages are not yet of 100% so the approach needs to be further improved. So for further studies, more investigations will be made into this view with the possibility of integrating machine learning/ deep learning approaches since it has been proven to gain interest in the field of background subtraction. Also, we intend to explore other features that can be taken into consideration to improve the ViBE algorithm such as texture, RGB + depth as well as some feature schemes like a pool of features calculated using algorithms such as KDE.

## REFERENCES

- [1] S. S. Hossain, S. Khalid, and C. Nabendu. *Moving Object Detection Using Background Subtraction*. Springer, Cham: Springer Briefs in Computer Sciences, 2014.
- [2] L. Dong, L. Yali, H. Fei, and W. Shengjin. “Object Detection in Image with Complex Background”. In: *Third International Conference on Multimedia Technology* (), (2013).
- [3] F. Zeng, G. Zhang, and J. Jiang. “Text Image with Complex Background Filtering Method Based on Harris Corner-point Detection”. In: *journal of software* 8.8 (), (2013).
- [4] O. Barnich and M. V. Droogenbroeck. “ViBe : A universal background subtraction algorithm for video sequences.” In: *IEEE Transactions on Image Processing* (June 2011), pages 1709–1724.
- [5] L. Xin, Y. Jiawen, H. Xiaopeng, Z. Ziheng, Q. Chun, and Z. Guoying. “Background Subtraction using Spatio-Temporal Group Sparsity Recovery”. In: *European Conference on Computer Vision* (2008), pages 155–168.
- [6] K. Cheng, H. Kaifa, Y. Zhan, and M. Qi. “A novel improved ViBe algorithm to accelerate the ghost suppression”. In: *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD* (2016), pages 1692–1698.
- [7] Y. Zhang and B. Guo. “An Improved ViBe Algorithm Based on Salient Region Detection”. In: *Advances in Intelligent Information Hiding and Multimedia Signal Processing* (Mar. 2017), pages 333–339.

- [8] L. Zhao, Y. Chen, Y. Zou, and Q. Ye. “Shooting for Smarter Motion Detection in Cameras: Improvements for the Visual Background Extractor Algorithm Using Optical Flow”. In: *IEEE Consumer Electronics Magazine* 6.4 (Oct. 2017), pages 81–91.
- [9] D. Adiguzel and L. Ozyilmaz. “Improving of ViBE algorithm for moving target detection at maritime surveillance”. In: *Signal Processing and Communications Applications Conference, SIU* (), (2018).
- [10] A. Elgammal, D. Harwood, and L. Davis. “Non-parametric Model for Background Subtraction”. In: *European Conference on Computer Vision* (), (2000).
- [11] H. Wang and D. Suter. “A consensus-based method for tracking: Modeling background scenario and foreground appearance”. In: *Pattern Recognition* 40 (Mar. 2007), pages 1091–1105.
- [12] H. Botao and Y. Shaohua. “An Improved Background Subtraction Method Based on ViBE”. In: *Chinese Conference on Pattern Recognition, Springer* 662 (2016), pages 356–368.
- [13] Z. Fan, Z. Li, P. Li, and H. Wang. “Motion Segmentation Based on Dual Interrelated Models”. In: *University Journal of Natural Sciences* 22.1 (2017), pages 79–84.
- [14] L. Chang, Zhenghua, and Y. Ren. “Improved Adaptive ViBE and the Application for Segmentation of Complex Background”. In: *Hindawi Publishing Corporation, Mathematical Problems in Engineering* (), (2016).
- [15] M. F. Karagoz, M. Aktas, and T. Akgun. “CUDA implementation of ViBe background subtraction algorithm on jetson TX1/TX2 modules”. In: *IEEE Signal Processing and Communications Applications Conference, SIU* (), (2018).
- [16] K. Rudrika and A. Sakshi. “Background subtraction for moving object detection: explorations of recent developments and challenges”. In: *The Visual Computer* (2021), pages 1–28.
- [17] K. Rudrika and A. Sakshi. “A comprehensive survey of video datasets for background subtraction”. In: *IEEE Access* 7 (2019), pages 59143–59171.
- [18] S. K. Choudhury, A. image of Pankaj Kumar Sa, S. Bakshi, and B. Majhi. “An evaluation of background subtraction for object detection vis-a-vis mitigating challenging scenarios”. In: *IEEE Access* 4 (2016), pages 6133–6150.
- [19] X. Zhou, X. Liu, A. Jiang, B. Yan, and C. Yang. “Improving Video Segmentation by Fusing Depth Cues and the ViBe Algorithm”. In: *Preprint* (), (2017).
- [20] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance”. In: *Proceedings of the IEEE* 90 (July 2002), pages 1151–1163.
- [21] E. T. Fute, L. L. S. Deffo, and E. Tonye. “**EFF-ViBE: An Efficient and Improved Background subtraction Approach based on ViBE**”. In: *International Journal of Image, Graphics and Signal Processing(IJIGSP)* 11.2 (2019), pages 1–14.
- [22] M. Mandal, P. Saxena, S. K. Vipparth, and S. Murala. “CANDID: Robust Change Dynamics and Deterministic Update Policy for Dynamic Background Subtraction”. In: *24th International Conference on Pattern Recognition (ICPR)* (), (2018).
- [23] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. “CDnet 2014: An expanded Change detection Benchmark Dataset”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops* (June 2014), pages 16–21.
- [24] B. Laugraud. “vibe-sources, Source code in C/C++! Original implementation. Example for OpenCV”. In: *University of Liege, Belgium, <http://orbi.ulg.ac.be/bitstream/2268/145853/5/vibe-sources.zip>* (), (2014).

- [25] K. Sehairi, F. Chouireb, and J. Meunier. “Comparative study of motion detection methods for video surveillance systems”. In: *Journal of Electronic Imaging* 26.2 (), (2018).
- [26] C. Stauffer and E. Grimson. “Adaptive background mixture models for real-time tracking”. In: *Computer Vision and Pattern Recognition* (1999), pages 246–252.
- [27] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin. “Flexible Background Subtraction With Self-Balanced Local Sensitivity”. In: *In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA (24-27 June 2014)*, pages 414–419.