



**HAL**  
open science

# INBaSA: An Improved Non-parametric background subtraction algorithm improving pixel segmentation

Lionel Landry Sop Deffo, Elie Fute Tagne

► **To cite this version:**

Lionel Landry Sop Deffo, Elie Fute Tagne. INBaSA: An Improved Non-parametric background subtraction algorithm improving pixel segmentation. 2022. hal-03697944v1

**HAL Id: hal-03697944**

**<https://hal.science/hal-03697944v1>**

Preprint submitted on 17 Jun 2022 (v1), last revised 7 Nov 2022 (v2)

**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: An Improved Non-parametric background subtraction algorithm improving pixel 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, pedestrians' detection and tracking, person identification, etc. All these operations need to be performed following a certain number of steps in which one of the most important is background subtraction used for motion detection. Despite improvements in proposed approaches, we still face demand for greater performances from users. In this line, ViBE algorithm has captured our attention through the original version and one of its improvement called EFF-ViBE. While the first suffers from two main drawbacks which are the apparition of a fictive object in movement called ghost phenomenon as well as poor quality of results in case of complex background. The second on the other hand while trying to solve those drawbacks introduce a great number of false positive and false negative and therefore affects the results. This paper presents an approach called INBaSA to tackle down the previous mentioned problems. We have introduced new factors such as the mean of temporal gradients, adaptive radius with offset value, 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 computer vision applications, the first step is to detect the moving object in a scene in consecutive video frames. In the case of pedestrians' detection for example, we need to know the position of the pedestrian in the scene by using an average of images over time (Shaikh et al., 2014). 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 read clearly and intuitively (Dong et al., 2013). It can also be seen as a situation where clutter has been introduced in the background (Fanfeng et al., 2013) and therefore leads to an influence on the performance of detection algorithm. This study made us notice that 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 (Barnich and Droogenbroeck, 2009).

That is why we will focus our study on sample-based techniques which build their models from observed pixel values (Barnich and Droogenbroeck, 2009).

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

## II PRESENTATION OF INBASA BACKGROUND SUBTRACTION TECHNIQUE

We present in this section the details of INBaSA approach. The major modifications are the introduction of mean of temporal gradients (mTG), which leads to the modification of parameters such as the adaptive radius as well as the number of frames to initialize the background. Since it is still a ViBE based approach, it has the 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 (Murari et al., 2018) which enables to locate 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 using Equation 1.

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

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

As it can be seen on 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, the *change* parameter will still be used. Let us now look at those steps in details.

### 2.1 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

$$M = \{v_0(x), v_1, v_2, \dots, v_N\} \quad (2)$$

where

$$v_0(x) = \frac{1}{n} \sum_{t=1}^n v^t(x) \quad (3)$$

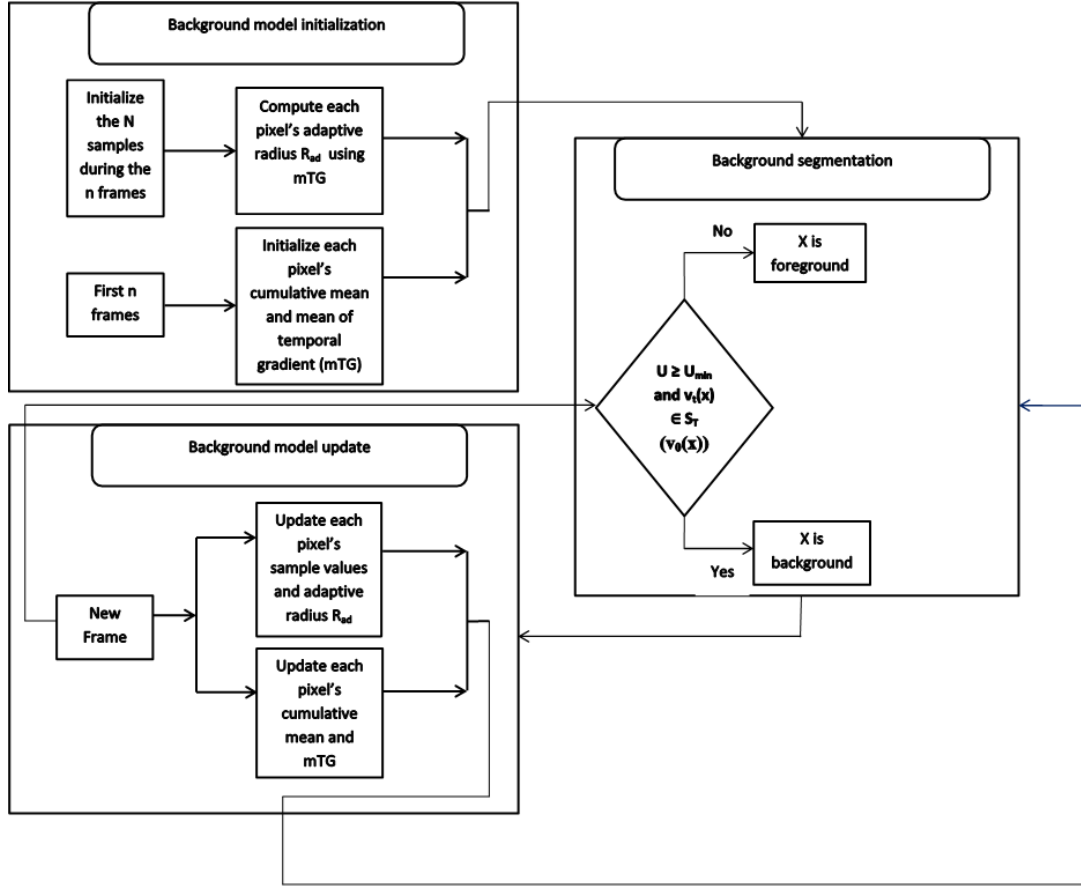


Figure 1: General architecture INBaSA approach

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  $x$ .

$$R_{ad}(x) = mTG(x) + \alpha \quad (4)$$

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 1.

## 2.2 INBaSA Background segmentation

The segmentation process follows the same steps as it is in ViBE with additional parameters. Therefore, using some notations that is  $B(x)$ , being 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 :

$$B(x) = \begin{cases} 0, & \text{if } U \geq U_{min} \text{ and } v^t(x) \in S_T(v_0(x)) \\ 255, & \text{else} \end{cases} \quad (5)$$

$$U = |S_{R_{ad}}(v^t(x)) \cap \{v_1, v_2, \dots, v_N\}| \quad (6)$$

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

$$T = \begin{cases} R_{ad}, & \text{if } \sigma \leq \beta * R_{ad} \\ \sigma/\beta, & \text{else} \end{cases} \quad (7)$$

$$\sigma = \sqrt{\sum_{t=1}^n (v^t(x) - v_0(x))^2} \quad (8)$$

The values of  $U_{min}$  and  $\beta$  are maintained to 2 and 3 respectively.

### 2.3 INBaSA Background model update

Using the same assumptions of ViBE algorithm, we update our model. The first one considers that background pixels in the neighborhood share the same temporal distribution and that a newly 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 foreground within  $K$  consecutive frames as background pixel is used leading to:

$$B(x) = \begin{cases} 0, & \text{if } count(x) > counter_{max} \\ B(x), & \text{otherwise} \end{cases} \quad (9)$$

$$count(x) = \begin{cases} count(x) + 1, & \text{if } B(x) = 255 \\ & \text{and } count(x) \leq counter_{max} \\ 0, & \text{if } count(x) > counter_{max} \end{cases} \quad (10)$$

Finally  $\sigma$ ,  $v_0(x)$  and  $d_{mean}$  are updated using the following equations.

$$\sum_{t=2}^{n+1} v^t(x) = \sum_{t=1}^n v^t(x) + v^{n+1}(x) - v^1(x) \quad (11)$$

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

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

$$v_0^{t+1}(x) = \frac{1}{n} \sum_{t=2}^{n+1} v^t(x) \quad (13)$$

$$\sigma^{t+1} = \sqrt{\left( \sum_{t=2}^{n+1} (v^t(x))^2 \right) - (v_0^{t+1}(x))^2} \quad (14)$$

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  $x$ :

$$M_{t+1}(x) = M_t(x) - |I_2(x) - I_1(x)| + |I_{t+1}(x) - I_t(x)| \quad (15)$$

Where

$$M_t(x) = \sum_{k=2}^n |I_k(x) - I_{k-1}(x)| \quad (16)$$

Table 1: Parameters used in our implementation

| Parameters               | values |
|--------------------------|--------|
| Number of frames ( $n$ ) | 300    |
| $U_{min}$                | 2      |
| $\beta$                  | 3      |
| Time sampling $\phi$     | 16     |
| $counter_{max}$          | 50     |
| $\varepsilon_c$          | 0.08   |
| $\varepsilon_d$          | 0.35   |
| $\delta$                 | 6      |

### III IMPLEMENTATION

We have used C, C++ languages for the implementation of the proposed algorithm. Once the source code 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 [changedetection.net](http://changedetection.net) dataset (Yi et al., 2014), and the parameters used for our simulation are gathered in Table 1

Since it is ViBE-based approach, we have adapted the code (Laugraud, 2014) to implement INBaSA approach. The algorithms implemented and tested were respectively original ViBE(only test) (Barnich and Droogenbroeck, 2009), EFF-ViBE (Fute et al., 2019) and INBaSA. We have also compared the results with popular background subtraction algorithms such as Euclidean distance (Kamal et al, 2017), Mixture of Gaussian (MoG) (Stauffer and Grimson, 1999), Kernel density estimation (KDE) (Elgammal et al., 2002), self organized background subtraction (SOBS) (Kamal et al, 2017) and SUBSENSE (St-Charles et al., 2014).

To measure the performance parameters, we have used the source code available on [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.

#### 3.1 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 (Fute et al., 2019) to use the  $n$  first frames to initialize the background and it has shown to give better results with a possibility of improvements. It is in this view that we come with this improvement by introducing the mean of temporal gradient and use 300 frames rather than 50 to initialize the background model. The general remark is that the shadow if 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 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.

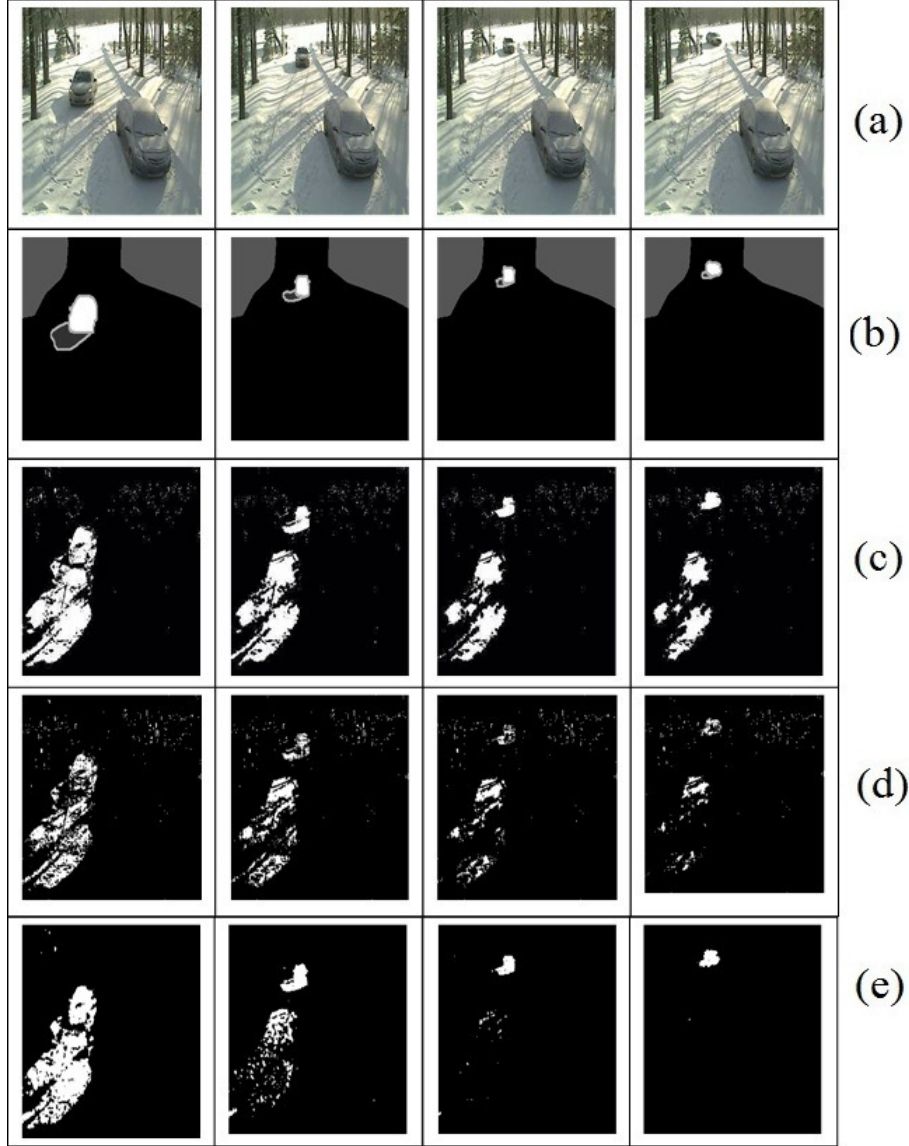


Figure 2: A target moves from a stationary position (winter Drive way). (a) the 1900<sup>th</sup>, 2000<sup>th</sup>, 2100<sup>th</sup>, 2200<sup>th</sup> frame; (b) ground truth images provided by the dataset; (c) the detection results of the original ViBE; (d) the detection results of EEF-ViBE approach;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

### 3.2 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 **ref74** 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.

### 3.3 Measure of performances

Here, we have recorded F-measure of eight algorithms: the original ViBE algorithm (Barnich and Droogenbroeck, 2009), EEF-ViBE (Fute et al., 2019) algorithm, the proposed INBaSA method and some challenging algorithms such as Euclidean distance (Kamal et al., 2017), MOG (Stauffer and Grimson, 1999), KDE (Elgammal et al., 2002), SOBS (Kamal et al., 2017) and

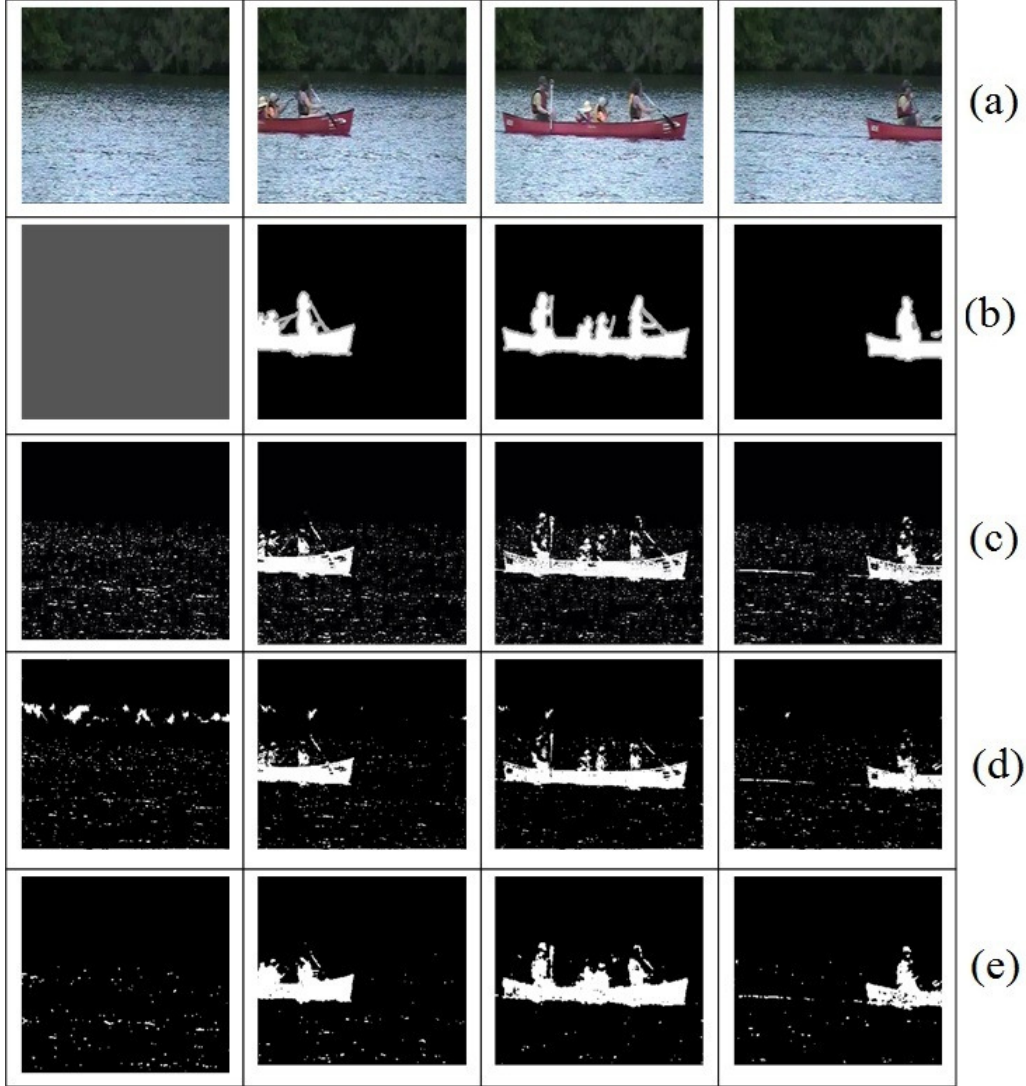


Figure 3: Effect of the adaptive radius (canoe). (a) the 600<sup>th</sup>, 900<sup>th</sup>, 960<sup>th</sup>, 1040<sup>th</sup> frame; (b) the real background corresponding with (a); (c) the detection results of the original ViBE; (d) the detection results of improved version EEF-ViBE approach; (e)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

SUBSENSE (St-Charles et al., 2014). The results are shown in Table 2. 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 utilities code available on 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 changedetection platform. Mathematically, the precision  $Pr$ , recall  $R$  and the F-measure  $F$  are computed using the formula

$$precision : (Pr) = \frac{TP}{TP + FP} \quad (17)$$

$$Recall : (R) = \frac{TP}{TP + FN} \quad (18)$$



Table 2: 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        |

$$F - measure : (F) = \frac{2 * Pr * R}{Pr + R} \quad (19)$$

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 2, INBaSA outperforms EFF-ViBE with a percentage of 90.90% that is 10 categories on 11 except PTZ. For SUBSENSE, this outperforming percentage will be 9.09% that is 1 category (turbulence) on 11. The comparison with the rest of the algorithms gives a percentage of 54.54% that is 6 categories on 11. It is important to mention that, even though SUBSENSE has good results it is 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. At the end, the total execution time of each algorithm is divided by the total number of processed frames.

$$fps_{algorithm\ x} = \text{execution time of } x / \text{total number of processed frames} \quad (20)$$

## IV CONCLUSION

In this study, we have demonstrated that ViBE background subtraction algorithm can be better improved. For this purpose, we introduced three main mechanisms which are the mean of temporal gradient (mTG), the adaptive radius with 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 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 in 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 ViBE algorithm such as texture, RGB + depth as well as some feature schemes like pool of features calculated using algorithms such as KDE.

## REFERENCES

- [Barnich and Droogenbroeck, 2009] OLIVIER BARNICH MARC VAN DROOGENBROECK, (2009), ViBE: A powerful random technique to estimate the background in video sequences, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.
- [Dong et al., 2013] LI DONG LI YALI HE FEI WANG SHENGJIN, (2013), Object Detection in Image with Complex Background, *Third International Conference on Multimedia Technology*.
- [Elgammal et al., 2002] AHMED ELGAMMAL RAMANI DURAISWAMI DAVID HARWOOD LARRY DAVIS, (2002), Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, *Proceedings of the IEEE*, 90, 7.
- [Fute et al., 2019] ELIE FUTE T. LIONEL L. SOP DEFFO EMMANUEL TONYE, (2019), EFF-ViBE: An Efficient and Improved Background Subtraction Approach based on ViBE, *International Journal of Image, Graphics and Signal Processing(IJIGSP)*, 11, 2.
- [Kamal et al., 2017] KAMAL SEHAIRI FATIMA CHOUIREB JEAN MEUNIER, (2017), Comparative study of motion detection methods for video surveillance systems, *Journal of Electronic Imaging*, 26, 2.
- [Laugraud, 2014] BENJAMIN LAUGRAUD, (2014), `vibe-sources`, Source code in C/C++! Original implementation. Example for OpenCV, `url =orbi.ulg.ac.be/bitstream/22681458535/vibe-sources.zi`.
- [Murari et al., 2018] MURARI MANDAL PRAFULLA SAXENA SANTOSH KUMAR VIPPARTH SUBRAHMANYAM MURALA, (2018), CANDID: Robust Change Dynamics and Deterministic Update Policy for Dynamic Background Subtraction, *24th International Conference on Pattern Recognition (ICPR)*.
- [Shaikh et al., 2014] Shaikh Soharab Hossain SAAED KHALID CHAKI NABENDU, (2014), Moving Object Detection Using Background Subtraction, *Springer Briefs in Computer Sciences*.
- [Stauffer and Grimson, 1999] CHRIS STAUFFER ERIC GRIMSON, (1999), Adaptive background mixture models for real-time tracking, *Computer Vision and Pattern Recognition*, 2.
- [St-Charles et al., 2014] PIERRE LUC ST-CHARLES GUILLAUME ALEXANDRE BILODEAU ROBERT BERGEVIN, (2014), Flexible Background Subtraction With Self-Balanced Local Sensitivity, *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*.

- [Wang et al., 2014] YI WANG PIERRE MARC JODOIN FATIH PORIKLI JANUSZ KONRAD YANNICK BENEZETH PRAKASH ISHWAR, (2014), CDnet 2014: An expanded Change detection Benchmark Dataset, *IEEE Conference on computer Vision and Pattern Recognition Workshops*.
- [Zeng et al., 2013] FANFENG ZENG GUOFENG ZHANG JIN JIANG, (2013), Text Image with Complex Background Filtering Method Based on Harris Corner-point Detection, *journal of software*, 8, 8.