Background Subtraction: Experiments and Improvements for ViBe

M. Van Droogenbroeck and O. Paquot
University of Liège
Institut Montefiore, Grande Traverse 10, B-4000 Liège, Belgium
M.VanDroogenbroeck@ulg.ac.be

Abstract

Motion detection plays an important role in most video based applications. One of the many possible ways to detect motion consists in background subtraction.

This paper discusses experiments led for a particular background subtraction technique called ViBe. This technique models the background with a set of samples for each pixel and compares new frames, pixel by pixel, to determine if a pixel belongs to the background or to the foreground.

In its original version, the scope of ViBe is limited to background modeling. In this paper, we introduce a series of modifications that alter the working of ViBe, like the inhibition of propagation around internal borders or the distinction between the updating and segmentation masks, or process the output, for example by some operations on the connected components. Experimental results obtained for video sequences provided on the workshop site validate the improvements of the proposed modifications.

1. Introduction

Many families of tools related to motion detection in videos are described in literature. Some of them focus on tracking, other on motion analysis or interpretation. In video-surveillance, techniques concentrate on change detection (the user just wants to know if there is some motion in the scene) and on motion segmentation (an exact delineation of objects is desired). For both usages and for fixed cameras, background subtraction techniques are very popular. The principle of these techniques consists in building a background model for each pixel and then to compare the model to the current value of a pixel. Several papers review background subtraction techniques [3, 4, 6, 8, 15]. Except for the exact form of their model, background subtraction techniques also differentiate in the way they update the model, and how they compare a pixel value to the model. These kinds of considerations explain why some popular models, like that of the Mixture of Gaussians (see for example [11, 17, 20]) are declined in several ways. But the technical details of a background subtraction technique are not limited to the core of the algorithm; authors often tend to characterize its behavior with respect to ghosts, shadows, temporary occlusion, camera shaking, camouflage, etc. It appears that to perform well, pre- or post-processing steps are then required, or possibly that the model has to be adapted to meet some specific requirements (for example to handle camera movements or shadows).

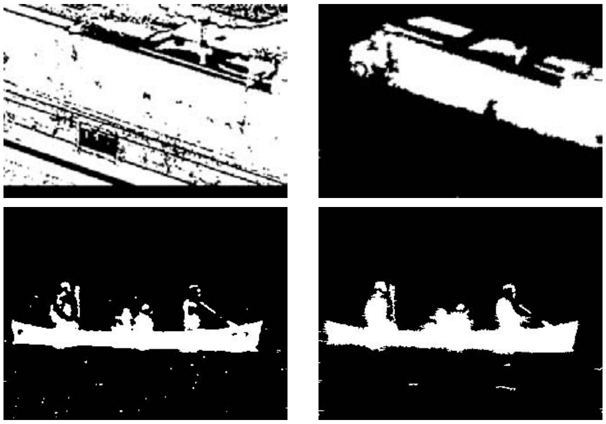


Figure 1. Segmentation masks obtained with the original version of ViBe (left column) and after modifications proposed in this paper (right column).

In this paper, we discuss the performance of a particular background subtraction technique called ViBe and described by Barnich and Van Droogenbroeck [1, 2]. We propose several modifications of the original algorithm and post-processing operations at the blob level. Figure 1 shows some results of our modified version of ViBe, named ViBe+ hereafter.

The remainder of is paper is organized as follows. Section 2 describes the principles of the ViBe algorithm. In Section 3, we present several modifications of the algorithm and some post-processing operations that improve the performance of ViBe. Experiments and discussions are provided in Section 4, and Section 5 concludes this paper.

978-1-4673-1612-5/12/$31.00 ©2012 IEEE

2. Some principles of the ViBe algorithm

ViBe is a technique that collects background samples to build background models. Some key points of ViBe are:

- background models are made of 20 background samples for each pixel.
- background samples are selected randomly to update the model; other samples are discarded.
- there is a spatial propagation mechanism that inserts background values in the models of neighboring pixels. Once the random policy decides to substitute a value of the model, it also inserts that value in the model of one of the neighboring pixels. Only a very few background subtraction techniques use of spatial mechanism (see the paper by Maddalena and Petrosino for another example [12]).
- there is no notion of time in ViBe. Old and recent values are considered equally when there are replaced. In [2], it is shown that the expected remaining lifespan of any sample value of the model decays exponentially.
- there is a simple decision process to determine if a pixel belongs to the background (it is sufficient to find at least two samples close enough, in terms of the euclidean distance, to classify the pixel in the background).

Generally speaking, there are several criteria to classify background subtraction models:

- **Background model based on an underlying model for the probability density function or on a set of samples**. One common approach to background modeling consists to assume that background samples are generated by a random variable and therefore fit a given probability density function. Then it is sufficient to estimate the parameters of the density function to determine if a new sample belongs to the same distribution. Alternatively, it is also possible to collect samples for a background model and to store them instead of computing the parameters of the underlying probability density function of background pixels. The technique proposed by Wang and Suter [19] memorizes the last 100 background samples for each pixel. ViBe has a similar approach except that the amount of stored values is limited to 20, thanks to a random selection policy. Authors sometimes describe their technique as being unimodal or multimodal, but this distinction is difficult to clarify for techniques that have no underlying probability density model.
- **Parametric versus nonparametric**. Parametric models require to optimize parameter values. Nonparametric models are more flexible but also more sensitive to data. In [2], the authors claim that ViBe is nonparametric. In fact, it appears that fixed parameters are adequate for many video sequences, to the exception of the updating factor that refers to the probability that a background value is used to update its model. The original value of 20, which means that only 1 out of 20 background values is selected (randomly), is not the best for rapidly changing backgrounds. Therefore, we use an updating factor of 5, and even 1 when we detect that there is jitter on the camera. To detect that the camera moves, we track a set of features detected in the first frame with the Kanade-Lucas-Tomasi optical flow algorithm, and detect frame by frame if most features remain static or not. Then there is a majority vote over the first 100 frames to decide if there is a global motion of the camera. More details are provided in Section 4.2.
- **Conservative or nonconservative updating policy**. In a conservative background model, only values of pixel classified as background are inserted in the model. ViBe applies this policy. This is important to ensure the background consistency. On the other hand, there is a risk that new objects are never incorporated into the scene and remain forever. This risk is partly dealt with by means of the spatial propagation mechanism explained previously. Note however that for change detection (as opposed to object segmentation), it is appropriate and simpler to incorporate objects progressively instead of maintaining them in the foreground.

In this paper, we do not propose some modifications specific to shadows. Although there are many techniques to address problems caused by shadows (see [14, 16] for surveys), we believe that the question of how to properly handle shadows is subject to controversy, because of the diversity of the physical origins of shadows. As mentioned in [16], shadows have physical, geometrical, and temporal characteristics. In probability based background models, it is possible to compare a value to the mean value of the model for shadow analysis; this is less straightforward for sample based models. Despite that, comparing values is only one method to deal with some physical aspects of shadows. An efficient method should also considered geometrical and temporal characteristics. Ultimately, we decided to ignore shadows and consider shadows as foreground pixels.

A last important consideration is that of the analysis level. Motion can be addressed at the pixel level or at the blob level. It is amazing to see that techniques that ignore the notion of objects, like Vibe, perform well even at the object level. Most pixel based subtraction techniques are real time nowadays, which makes them attractive. However, humans do interpret motion mainly at the object level. To some extent, we can see filtering operations on connected components, as proposed hereafter, as a first step towards the integration between the pixel level and the object level.

3. Modifications of ViBe

In comparison to ViBe, one of the minor modifications introduced in our algorithm is an updating factor reduced to 5 (or 1) as described in Section 2. But there are many more changes.

3.1. Distinction between the segmentation mask and the updating mask

The purpose of a background subtraction technique is to produce a binary mask with background and foreground pixels. Most of the time, it is the segmentation mask that users are looking for. In a conservative approach, the segmentation mask is used to determine which values are allowed to enter the background model. In other words, the segmentation mask plays the role of an updating mask. But this is not a requirement. Therefore, we process the segmentation mask and the updating mask differently. As unique constrain, we impose that foreground pixels should never be used to update the model.

3.2. Filtering connected components

In our algorithm, we apply several area openings [18] on both the segmentation and updating masks:

- Segmentation mask: remove foreground blobs whose area is smaller or equal to 10 (pixels) and fill holes in the foreground whose area is smaller or equal to 20. Blobs that touch the border are kept regardless of their size.
- Updating mask: fill holes in the foreground whose area is smaller or equal to 50. This operation is applied to limit the appearance of erroneous background seeds inside foreground objects. Note that for the updating mask, we keep all the foreground blobs. This is coherent with the conservative nature of the updating process (foreground values should not be inserted in background models).

3.3. Inhibition of propagation

In addition to operations on foreground and background blobs, we introduce a mechanism to inhibit the spatial propagation. The spatial propagation consists in inserting a background value in the model of a 8-connected neighboring pixel taken randomly. This propagation mechanism, which is part of the innovations introduced with ViBe, diffuses values in the background and contributes to suppress ghosts and static objects over time. However, it is not always suitable to suppress static objects; this might better be decided at the blob level depending on the application. As a compromise, we compute the gradient on the inner border of background blobs and inhibit the propagation process when the gradient (rescaled to the $[0, 255]$ interval) is larger than 50. This avoids that background values cross object borders.

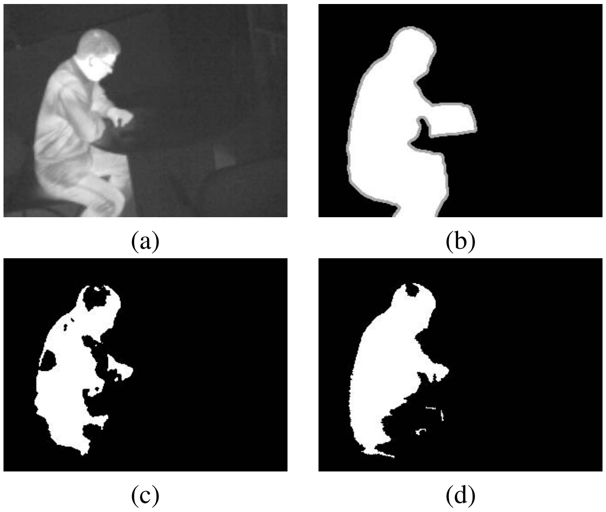


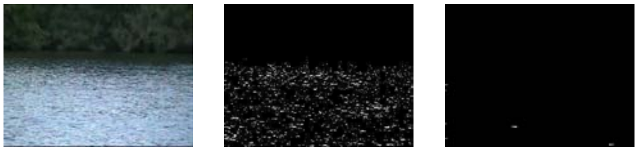
Figure 2. Comparison of the effects of the original version of ViBe and our modified algorithm, ViBe+: (a) infrared input image, (b) groundtruth, (c) segmentation mask of ViBe, (d) segmentation mask obtained with ViBe+.

The effects of this inhibition technique are illustrated in Figure 2. One of the strengths of ViBe consists to gradually suppress ghosts. Some background "seeds" are randomly inserted in neighboring models and once two of these seeds appear in the model of a pixel, this foreground pixel switches to the background. While this approach is meaningful for ghosts, it is not appropriate for static objects when users want to keep static objects over time. In Figure 2, one can see, by comparing (c) and (d), that the inhibition process slows down the propagation process of background seeds in the foreground object.

3.4. Adapted distance measure and thresholding

In [2], the authors of ViBe claim that the method is almost parameterless. This has to be understood as insensitive to a slight modification of the threshold. For simplicity, the authors use an euclidean distance to measure the matching. While it proved efficient on many video sequences, it can be improved. A different approach is used by several authors that distinguish between color matching and luminance matching.

Our distance metric is inspired by the one of Kim *et al.* [10]. The distance for this codebook based background technique compares the intensities and computes some color distortion. Our color distortion is exactly the $colordist()$ defined by equation (2) of [10]. This color dis-



Input frame ViBe ViBe+

Figure 3. Effects of the detection of blinking pixels. Less false positives are detected with ViBe+.

tortion measure can be interpreted as a brightness-weighted version in the normalized color space. In ViBe+, a required condition for two values to match is that the color distortion is lower than 20. In addition, there is a second condition on the intensity values. Originally, ViBe considers that two intensities are close if their difference is lower than 20. In [5], Brutzer *et al.* suggest for ViBe to use a threshold in relation to the samples in the model for a better handling of camouflaged foreground. Therefore, we compute the standard deviation σ_m of the samples of a model and define a matching threshold as $0.5 \times \sigma_m$ bounded to the $[20, 40]$ interval. We observed that both a color distortion metric and an adaptive threshold improve the performance of our algorithm.

3.5. A heuristic to detect blinking pixels

One of the major difficulties related to the use of sample-based models is the handling of multimodal background distributions because there is no explicit mechanism to adapt to them. As an alternative, we propose a method to detect if a pixel often switches between the background and the foreground (this pixel is then called a *blinking* pixel).

For each pixel, we store the previous updating mask (prior to any modification) and a map with the blinking level. This level is determined as follows. If a pixel belongs to the inner border of the background and the current updating label is different from the previous updating label, then the blinking level is increased by 15 (the blinking level being kept within the $[0, 150]$ interval), otherwise the level is decreased by 1.

This process is similar to the known $\Sigma - \Delta$ technique (see for example [13] for a use of it in the context background subtraction). A pixel is considered as blinking if its level is larger or equal to 30, and if so, the pixel is removed from the updating mask. In other words, we allow the blinking level to be increased only at the frontier of the background mask but suppress all blinking pixels from the updating mask. This technique enhances the behavior of our algorithm for multimodal background distributions. Note that the detection of blinking pixels is deactivated when the camera is shaking. An illustration of the benefits of using a heuristic for detection of blinking pixels is shown in Figure 3. With ViBe+, there are less false positives in the water area.

4. Experiments

4.1. Methodology

The modified algorithm was compared to other techniques, including the original version of ViBe, using the public dataset provided on the `http://www.changedetection.net` web site. The dataset contains 31 video sequences, grouped in 6 categories: baseline, dynamic background, camera jitter, intermittent object motion, shadow, and thermal. The names of the categories are quite explicit, so we don't detail their content.

For our experiments, we use a unique set of parameters (given in the next section), including for thermal images. All videos sequences are processed and then binary masks (where a 0 value represents the background) are compared to groundtruth masks. While the groundtruth data contain 5 labels, we only target the detection of static pixels (the background) and pixels of moving objects (the foreground).

If background subtraction is seen as a binary classification problem, where one wants to distinguish foreground (usually considered as *positive*) from background (*negative*), then we can use the common terminology of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). In the following, results are evaluated in terms of two metrics: the percentage of bad classifications (PBC) and the precision. The PBC is expressed as

$$PBC = 100 \times \frac{FN + FP}{TP + FN + FP + TN}, \qquad (1)$$

and the precision is

$$\text{Precision} = \frac{TP}{TP + FP}. \qquad (2)$$

The change detection website also proposes and evaluates other metrics, not considered here.

4.2. Parameters

The list of all the parameters and their value used in our implementation of ViBe+ is given hereafter:

- Initialization
  - updating factor = 1, for the 100 first frames
- Parameters of ViBe
  - updating factor (after initialization frames) = 5
  - number of samples per pixel = 20
  - number of required matches = 2
- Parameters of modifications of ViBe
  - distance metric
    - amplitude multiplicative factor = 0.5
    - amplitude matching threshold range = $[20, 40]$
    - color distortion threshold = 20
  - edge inhibition
    - threshold for edge inhibition = 50
- Detection of blinking pixels

  - blinking value range = $[0, 150]$
  - blinking increment = 15
  - blinking decrement = 1
  - blinking threshold = 30
- Connected components filtering
  - updating mask
    - $*$ minimum size of holes in the foreground = 50 (pixels)
  - segmentation (output) mask
    - $*$ minimum size of foreground blobs = 10 (pixels)
    - $*$ minimum size of holes in the foreground = 20 (pixels)

For camera jitter detection, we use an implementation of the Kanade-Lucas-Tomasi feature tracker provided by Stan Birchfield, available at `http://www.ces.clemson.edu/~stb/klt`. The algorithm selects the 100 best features in the first frame and tracks them over 100 frames. A tracked feature is considered as static if its horizontal and vertical displacements are less than 1 pixel, and as dynamic otherwise. A frame is considered as static if at least half of the tracked features are static. The test is run over the 100 first frames of the video sequence and we operate a majority vote to decide whether there is jitter on the camera. If there is jitter on the camera, then the updating factor is reduced to 1 for the remainder of the sequence. With this simple process, we observed that all the sequences of the "camera jitter" category are detected as resulting from a moving camera. All other videos are rightly assessed as static.

4.3. Results and discussion

In this paper, we propose many changes to the original algorithm. From a practical point of view, it is very difficult to isolate the effects of each change separately either because the changes interact or because their behavior depends on the video sequence. Therefore, we only present the global results.

Values of the average PBC and precision are given, per category and overall, in Tables 1 and 2 respectively. In order to compare the results, we provide the values of the best ranked technique for each category and mention its reference (as available at the beginning of April 2012). We also mention the best at the end of April 2012. For the precision, the technique of KaewTraKulPong and Bowden [9] always performed best at the time of the first ranking. Please note that we hope the PBC to be as small as possible, to the contrary of precision.

For each raw, the best result is mentioned in bold. It appears that our algorithm improves the average percentage of bad classification for several categories; the overall averaged PBC is also close to best. The acronyms or names *PSP-MRF*, *Chebyshev probability*, *PBAS*, and *Integrated spatio-temporal features* relates to new techniques.

Our algorithm also outperforms the previous best precision for some categories and the overall average precision is larger than that of [9].

Similar observations can be done for other metrics, like the specificity but not for the recall. The recall, defined as the ratio between TP and $TP + FN$, is improved for 19 out 31 sequences but the overall average recall is 0.6840 to be compared to the recall of the original ViBe, that is 0.6758. This means that the amount of True Positives and False Negatives is similar for both the original and modified versions of ViBe.

It also appears that ViBe+ is slightly less efficient for the "baseline" category. This is not surprising as modifications were introduced primarily to enhance the behavior of ViBe for specific problems like multimodal backgrounds, camera jitter, or intermittent object motion.

At the time of writing the final version of this paper, other techniques have been ranked. ViBe+ appears second in the "Average ranking across categories" column but first in the "Average ranking" and "Average precision" columns.

5. Conclusions

In this paper, we present several modifications of the original ViBe algorithm. The modifications are mainly: a different distance function and thresholding criterion, a separation between updating and output masks, with proper filtering operations on them, an inhibition of propagation for some pixels in the updating mask, the detection of blinking pixels, and an increased updating factor, especially when there is jitter on the camera.

A comparison shows that the modified version of ViBe is preferable to the original version of ViBe for the majority of video sequences. In addition, for some categories and some metrics, our new algorithm outperforms many known techniques.

References

[1] O. Barnich and M. Van Droogenbroeck. ViBe: a powerful random technique to estimate the background in video sequences. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 945–948, April 2009.

[2] O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, June 2011.

[3] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19(3):033003, 2010.

[4] T. Bouwmans, F. El Baf, and B. Vachon. Statistical background modeling for foreground detection: A survey. In *Handbook of Pattern Recognition and Computer Vision (volume 4)*, chapter 3, pages 181–199. World Scientific Publishing, January 2010.

	Previous best	Best in April 2012	ViBe	ViBe+
baseline	0.4332 [12]	**0.4127** [PSP-MRF]	0.8869	0.9631
dynamic background	0.5405 [9]	**0.3436** [Chebyshev probability]	1.2796	0.3838
camera jitter	2.7479 [12]	**1.8473** [ViBe+]	4.0150	**1.8473**
intermittent object motion	5.1955 [17]	**4.4069** [PBAS]	7.7432	5.4281
shadow	1.6547 [2]	**1.5115** [PBAS]	1.6547	1.6565
thermal	1.6795 [7]	**1.3285** [Chebyshev probability]	3.1271	2.8201
overall	2.7049 [12]	**2.1066** [PBAS]	3.2035	2.1824

Table 1. Average percentage of bad classifications (PBC).

	Previous best	Best in April 2012	ViBe	ViBe+
baseline	**0.9532** [9]	**0.9532** [9]	0.9288	0.9262
dynamic background	**0.7700** [9]	**0.7700** [9]	0.5346	0.7291
camera jitter	0.6897 [9]	**0.8064** [ViBe+]	0.5289	**0.8064**
intermittent object motion	0.6953 [9]	**0.8166** [Integrated spatio-temporal features]	0.6515	0.7512
shadow	**0.8577** [9]	**0.8577** [9]	0.8342	0.8302
thermal	**0.9709** [9]	**0.9709** [9]	0.9363	0.9476
overall	0.8182[9]	**0.8318** [ViBe+]	0.7301	**0.8318**

Table 2. Average precision.

[5] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1937–1944, Colorado Spring, USA, June 2011.

[6] M. Cristani, M. Farenzena, D. Bloisi, and V. Murino. Background subtraction for automated multisensor surveillance: A comprehensive review. *EURASIP Journal on Advances in Signal Processing*, 2010:24 pages, 2010.

[7] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision-Part II*, volume 1843 of *Lecture Notes in Computer Science*, pages 751–767, London, UK, June-July 2000. Springer.

[8] S. Elhabian, K. El-Sayed, and S. Ahmed. Moving object detection in spatial domain using background removal techniques – State-of-art. *Recent Patents on Computer Science*, 1:32–54, January 2008.

[9] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *European Workshop on Advanced Video Based Surveillance Systems*, London, UK, September 2001.

[10] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, June 2005.

[11] H.-H. Li, J.-H. Chuang, and T.-L. Liu. Regularized background adaptation: A novel learning rate control scheme for gaussian mixture modeling. *IEEE Transactions on Image Processing*, 3(20):822–836, March 2011.

[12] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, July 2008.

[13] A. Manzanera and J. Richefeu. A new motion detection algorithm based on Σ-Δ background estimation. *Pattern Recognition Letters*, 28(3):320–328, February 2007.

[14] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. Detecting moving shadows: algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, July 2003.

[15] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, March 2005.

[16] A. Sanin, C. Sanderson, and B. Lovell. Shadow detection: A survey and comparative evaluation of recent methods. *Pattern Recognition*, 45(4):1684–1695, April 2012.

[17] C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 246–252, Ft. Collins, USA, June 1999.

[18] L. Vincent. Morphological area openings and closings for greyscale images. In *NATO Shape in Picture Workshop*, pages 197–208, Driebergen, The Netherlands, September 1992. Springer-Verlag.

[19] H. Wang and D. Suter. A consensus-based method for tracking: Modelling background scenario and foreground appearance. *Pattern Recognition*, 40(3):1091–1105, March 2007.

[20] Z. Zivkovic. Improved adaptive gausian mixture model for background subtraction. In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 2, pages 28–31, Cambridge, UK, August 2004.