Background Subtraction: Experiments and Improvements for ViBe

Marc Van Droogenbroeck and Olivier Paquot

INTELSIG Laboratory, Montefiore Institute, University of Liège, Belgium

June 2012

Outline

Background subtraction

Common tasks:

- background subtraction $\Rightarrow$ segmentation
- change detection $\Rightarrow$ generate alarms (and reduce rate of false alarms!)

Different contexts:

- Pixel based
- Region based
- Tracking

Plethora of methods

- 'motion detection' on IEEE Xplore: 4,400 papers
- 'background subtraction video' on IEEE Xplore: 740 papers

Common problems in background subtraction or modelling

- Shadows
- Noise
- Ghosts
- Camouflage
- Moving background
- Intermittent motion
- Light changes
- "Universal" vs "Application specific"

Improving ViBe?!

There is a challenge...

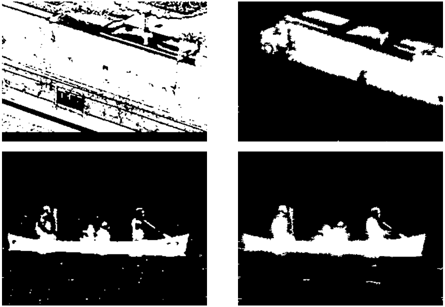


Figure: Segmentation masks obtained with the original version of ViBe (left column) and after modifications proposed for ViBe+ (right column).

History of ViBe

ViBe = Video Background Extractor

- M. Van Droogenbroeck and O. Barnich. Visual background extractor. World Intellectual Property Organization, WO 2009/007198, 47 pages, January 2009 (introduced in July 2007).
- O. Barnich and M. Van Droogenbroeck. ViBe: a powerful random technique to estimate the background in video sequences. In International Conference on Acoustics, Speech, and Signal Processing, pages 945-948, April 2009.
- 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.

Keys of a background subtraction technique

1. **Model**. What is a good model for the background?
2. How to **classify** pixels in the background/foreground? Need for a classification criterion.
3. How to **update** the model?
4. **Initialization**?

Requirements while designing ViBe:

1. No pre- or post-processing!
2. In real-time.
3. A unique set of parameters.
4. Working as soon as you get the second frame.

Model

- Model each background pixel with a set of samples instead of with an explicit pixel model (like for the method named SACON)
  - No estimation of pdfs.
  - No statistical notion. Assume a binary image, what is the meaning of the mean? It is even a value that might never be observed...
  - need to link a model to values of its immediate neighborhood.

Pixel classification

- Each background pixel x is modeled by a collection of N background sample values:
  $M(x) = \{v_1, v_2, \ldots, v_N\}$
- A pixel belongs to the background if there are two matches:
  $\#\{S_R(v(x)) \cap \{v_1, v_2, \ldots, v_N\}\} \geq \#_{\text{min}}$
- $N = 20$ and $\#_{\text{min}} = 2$

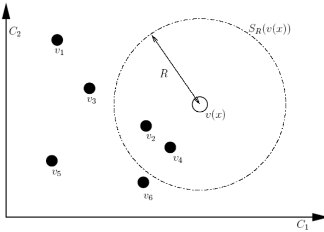


Fig. 1. Comparison of a pixel value with a set of samples in a dimensional Euclidean color space (C_1, C_2). To classify $v(x)$, we c the number of samples of $\mathcal{M}(x)$ intersecting the sphere of radius R cent on $v(x)$.

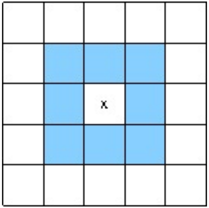
Updating the model over time

- **Conservative update**: update the model only if a pixel value is declared as a background. If it is foreground, do not update at all.
- **Spatial diffusion**: neighboring pixels can update the background model of a given pixel.
- Background model of a pixel is **updated randomly**!
- Random **time subsampling**.

Model Initialization

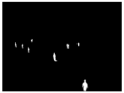
From a single frame, populate the pixel models with values found in the spatial neighborhood of each pixel.

- Values randomly taken in their neighborhood
- Values chosen in the close 8-connected neighborhood



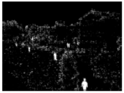
Results

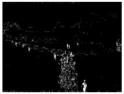

(a) Input image


(b) Ground-truth


(c) ViBe (RGB)

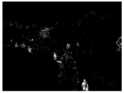

(d) ViBe (gray)


(e) Bayesian histogram

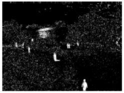

(f) Codebook


(g) EGMM [Zivkovic]


(h) GMM [Li et al.]


(i) Gaussian model


(j) 1^{st} order filter


(k) Sigma-Delta Z.

Characteristics of ViBe (I)

- There is no notion of time in ViBe.
- Random update of the model. ViBe proposes a nondeterministic background model. Running ViBe twice on the same sequence produces different results (but statistically similar).
- Diffusion in the neighborhood.
- Random time subsampling for updating.
- No pre- or post-processing.
- Less than 100 lines of C code !

Characteristics of ViBe (II)

For the first-order model $B_t = \alpha I_t + (1-\alpha)B_{t-1}$, we have

- 2 multiplications and 2 additions on doubles per pixel

For ViBe (average per pixel, assuming that most pixels belong to the background):

- 4 subtractions on bytes.
- $\frac{3}{16}$ addition on memory addresses.

$\Rightarrow$ ViBe is very fast.

Characteristics of ViBe (III)

Downsized version of ViBe for $N = 1$ and $\#_{\text{min}} = 1$

Input Video

ViBe Segmentation

Improving ViBe?

Already proposed:

- Dynamic thresholds.
- Introduce the notion of object.
- Deal with shadows explicitly.

Or

- Better color matching criterion?
- Adapt probabilities for keeping samples in the model?

Distinction between the segmentation mask and the updating mask

- The *segmentation mask* is the usual binary output mask.
- The *updating mask* only serves to determine which pixel might randomly be updated. The constraint of conservative update only impacts on the updating masks.

Filtering connected components

- Segmentation mask:
  - *remove foreground blobs* whose area is smaller or equal to 10 (pixels)
  - *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.
  - keep all the foreground blobs (conservative update).

Inhibition of propagation

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.

infrared input image

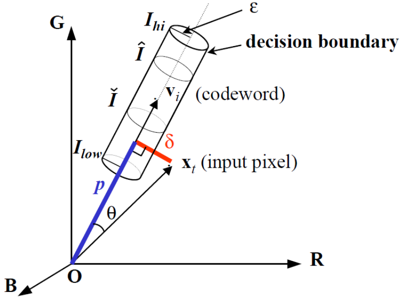
groundtruth

ViBe

ViBe+

Matching criterion: modified distance measure and adaptive thresholding

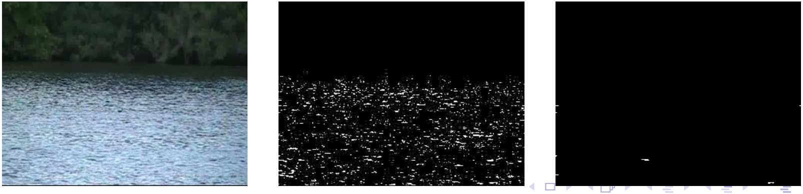
- K. Kim *et al.*, "Real-time foreground-background segmentation using codebook model", Real-Time Imaging 11(3):172-185, 2005.



- Adaptive thresholds (a portion of the variance)

A heuristic to detect blinking pixels

- Estimate the blinking level for each pixel:
  - 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.
- If the blinking level of the pixel is superior to 30, remove the pixel from the updating mask.



Evaluation

- Use the public dataset provided on the `http://www.changedetection.net` web site.
- 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`.
  Tracked features belonging to the background are used to estimate the global motion of the camera. If there is motion, the updating factor is increased.
- A unique set of parameters for all the sequences.

Average Percentage of bad classifications (PBC)

$$PBC = 100 \times \frac{FN + FP}{TP + FN + FP + TN}$$

	Prev. best	Best in April 2012	ViBe	ViBe+
baseline	0.4332	**0.4127** [PSP-MRF]	0.8869	0.9631
dynamic bkg	0.5405	**0.3436** [Chebyshev probability]	1.2796	0.3838
camera jitter	2.7479	**1.8473** [ViBe+]	4.0150	**1.8473**
int. object	5.1955	**4.4069** [PBAS]	7.7432	5.4281
shadow	1.6547	**1.5115** [PBAS]	1.6547	1.6565
thermal	1.6795	**1.3285** [Chebyshev probability]	3.1271	2.8201
overall	2.7049	**2.1066** [PBAS]	3.2035	2.1824

Conclusions

Compared to ViBe, ViBe+ introduces

- 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.