Proceedings of the 4th
International Conference on Robotics and Mechatronics
October 26-28, 2016, Tehran, Iran

An Adaptive Algorithm for Fast Moving Object Detection from Dynamic Background based on Cobebook

Mikaël A. Mousse*†, Cina Motamed* and Eugène C. Ezin†
*Laboratoire d'Informatique Signal et Image de la Côte d'Opale - EA 4491
Université Littoral Côte d'Opale
F-62228 Calais, France
†Unité de Recherche en Informatique et Sciences Appliquées
Institut de Mathématiques et de Sciences Physiques
Université d'Abomey-Calavi, Bénin
BP 613 Porto-Novo
Email: mousse@lisic.univ-littoral.fr

***Abstract*—In automatic videosurveillance system, the first step is the extraction of moving objects. In this work, we investigate a new approach for foreground-background segmentation by using a modified Codebook algorithm. This approach exploit the perceptual information to optimize the detection. It is an adaptive strategy which is proposed to to reduce the computational complexity of the foreground detection algorithm while maintaining its global accuracy. The algorithm uses superpixels which model the spatial dependencies between pixels and controls actively the distribution of these superpixels by focusing them on possible awaited location of foreground objects. The performance evaluation is performed on public sequences that propose sequences with a dynamic background and the experimental results demonstrate an increase in the detection speed.**

I. Introduction

Video sequences filmed by static cameras contain moving objects on a fixed or dynamic background. Moving objects extraction from background is an important task. It is the first step for any intelligent video surveillance system. Background modeling and subtraction are the widely used techniques to extract objects from background. Background modeling techniques model the background using previous frames history. Every image pixel is matched with its background model. Background modeling techniques are considered as a good choice for their better performance for indoor and outdoor scene. One of these methods is the Codebook based approach prposed by Kim et al. [6]. Indeed, they propose a real-time foreground background segmentation using codebook model. In this method, each pixel p_t is represented by a codebook $\mathcal{C} = \{c_1, c_2, ..., c_L\}$ and each codeword $c_i,\ i = 1, ..., L$ by a RGB vector v_i and a 6-tuples aux_i $=\{\check{I}_i,\ \hat{I}_i,\ f_i,\ p_i,\ \lambda_i,\ q_i\}$ where Ǐ and Î are the minimum and maximum brightness of all pixels assigned to this codeword c_i, f_i is the frequency at which the codeword has occurred, λ_i is the maximum negative run length defined as the longest interval during the training period that the codeword has not recurred, p_i and q_i are the first and last access times, respectively, that the codeword has occurred. The codebook model is created or updated using two criteria. The first criterion is based on color distortion (3) whereas the second is based on brightness distortion (4). Color distortion ($Colordist$) and brightness distortion (or pixel intensity I) are respectively computed by using expressions (1) and (2).

$$Colordist(p_t, c_i) = \sqrt{||p_t||^2 - C_p^2} \tag{1}$$

$$I = \sqrt{R^2 + G^2 + B^2} \tag{2}$$

$$Colordist(p_t, c_i) \leq \varepsilon_1 \tag{3}$$

$$I_{low} \leq I \leq I_{hi} \tag{4}$$

In (3), the autocorrelation value C_p^2 is given by expression (5) and $||p_t||^2$ is given by expression (6).

$$C_p^2 = \frac{(R_i R + G_i G + B_i B)^2}{R_i^2 + G_i^2 + B_i^2} \tag{5}$$

$$||p_t||^2 = R^2 + G^2 + B^2 \tag{6}$$

In relation (4), $I_{low} = \alpha \hat{I}_i$, $I_{hi} = min\{\beta \hat{I},\ \frac{\check{I}}{\alpha}\}$.

After the training period, if an incoming pixel matches to a codeword in the codebook, then this codeword is updated. If the pixel doesn't match, its information is put in cache word and this pixel is treated as a foreground pixel. The matched codeword is searched by using a condition based on color distortion (7) and brightness distortion (4).

$$Colordist(p_t, c_i) \leq \varepsilon_2 \tag{7}$$

Discussion about the choice of parameters $\varepsilon_1, \varepsilon_2, \alpha, \beta$ is made by Kim et al. [6]. Due to the performance of this method, several researchers studies it in deeply [5], [10], [2], [3]. In a past work [9], we propose to convert pixels from RGB to CIE L*a*b* color space and exploiting this color space specification and use L component as brightness value to reduce amount of calculation. We also use the improved simple linear iterative clustering (SLIC) algorithm to model the spatial dependencies between pixels. Mousse et al. [9] also use SLIC algorithm to model the spatial dependencies between pixels but they convert pixel from RGB to CIE L*a*b* color space and exploiting this color space specification. This strategy achieve a competitive performance and is use in some videosurveillance application [8]. Due to the use of superpixels, Fang et al. [3] and Mousse et al. [9] are the fastest methods among all the improvements of codebook.

The main objective of this work is to propose a foreground pixel extraction method which is based on Codebook and has competitive performance with less computational complexity. For this, we exploit the work presented in Mousse et al. [9] because it has an acceptable moving object detection accuracy. So we reduce the complexity of this algorithm while keeping its accuracy. Then, unlike to the algorithm proposed in [9], we use an adaptive approach to manage the superpixels. This strategy exploits perceptual image characteristic to reduce the number of superpixels that will be treated in order to extract the foreground objects. The purpose of the adaptive strategy is then to manage efficiently clusters and thereby reduce the complexity of the detection algorithm while maintaining the performance of the previous algorithm.

II. MODIFIED CODEBOOK MODEL FOR OBJECT DETECTION

In this section we present our proposed algorithm based on codebook for moving object detection. Like algorithm proposed in [9], our algorithm also works in two phases: the learning phase and the moving objects extraction phase. The section consists of two subsections. The first subsection presents our background modeling method and the second subsection presents our moving pixels extraction strategy.

A. Background Modeling

After the extraction of superpixels, we build a codebook background model based on these superpixels. Let $P = \{s_1, s_2, ..., s_k\}$ represent the K superpixels obtained after superpixels segmentation. Each superpixel $s_j, j \in \{1, 2, ..., k\}$ is composed approximately by m pixels. With each superpixel we built a codebook $C = \{c_1, c_2, ..., c_L\}$ which contains L codewords $c_i, i \in \{1, 2, ..., L\}$. Each codewords c_i consists on an vector $v_i = (\bar{a}_i, \bar{b}_i)$ and 6-tuples $aux_i = \{\check{L}_i, \hat{L}_i, f_i, p_i, \lambda_i, q_i\}$ in which $\check{L}_i$, $\hat{L}_i$ are the minimum and maximum of luminance value, f_i is the frequency at which the codeword has occurred, λ_i is the maximum negative run length defined as the longest interval during the training period that the codeword has not recurred, p_i and q_i are the first and last access times, respectively, that the codeword has occurred. $\bar{L}$, $\bar{a}$, $\bar{b}$ are respectively the average value of component L*, a* and b* of the pixels in a superpixel. We compute the color distortion by replacing (8) and (9) into (1). For the brightness distortion degree (2) we use $\bar{L}$ value as the intensity of the superpixel.

$$p_t = \bar{a}^2 + \bar{b}^2 \tag{8}$$

$$C_p^2 = \frac{(\bar{a}_i\bar{a} + \bar{b}_i\bar{b})^2}{\bar{a}_i^2 + \bar{b}_i^2} \tag{9}$$

For each superpixel, if we find a codeword c_i which respect these two criteria (brightness distortion criterion and brightness distorsion criterion) then we update this codeword by setting v_i to $(\frac{f_i\bar{a}_i+\bar{a}}{f_i+1}, \frac{f_i\bar{b}_i+\bar{b}}{f_i+1})$ and aux_L to $\{min(\bar{L}, \check{L}_i)$, $max(\bar{L}, \hat{L}_i)$, f_i+1, $max(\lambda_i, t-q_i)$, p_i, $t\}$. If we don't find a matched codeword, we create a new codeword c_K. In this case, v_K is equal to $(\bar{a}, \bar{b})$ and aux_K is equal to $\{\bar{L}, \bar{L}, 1, t-1, t, t\}$. The detailed algorithm is given by Algorithm 1. In this algorithm N is the number of frame that we used to model background.

Algorithm 1: Background modeling

$l \leftarrow 0$
for $t = 1$ *to* N **do**
  for *each frame* F_t **do**
    Segment frame F_t into superpixels
    for *each superpixels* Su_k *of frame* F_t **do**
      $p_t(\bar{L}, \bar{a}, \bar{b})$
      Find the matched codeword c_i in codebook matching to Su_k based on two conditions (a) and (b).
        (a) colordist $(p_t, v_i) <= \epsilon_1$
        (b) (brightness $(\bar{L}, \check{L}_i, \hat{L}_i)$) = true
      if $l = 0$ *or there is no match* **then**
        $l \leftarrow l + 1$
        create codeword c_L by setting parameter $v_L \leftarrow (\bar{a}, \bar{b})$ and $aux_L \leftarrow \{\bar{L}, \bar{L}, 1, t-1, t, t\}$
      else
        update codeword c_i by setting $v_i \leftarrow (\frac{f_i\bar{a}_i+\bar{a}}{f_i+1}, \frac{f_i\bar{b}_i+\bar{b}}{f_i+1})$ and $aux_i \leftarrow \{min(\bar{L}, \check{L}_i)$, $max(\bar{L}, \hat{L}_i)$, f_i+1, $max(\lambda_i, t-q_i)$, p_i, $t\}$
for *each codeword* c_i **do**
  $\lambda_i \leftarrow max\{\lambda_i, ((m \times n \times t) - q_i + p_i - 1)\}$

B. Foreground pixel extraction

After the learning phase, we detect foreground pixel. This operation is also based on the superpixels. Unlike other approaches of state of the art which perform a native subtraction method, this paper proposes an adaptive method to extract foreground maps.

With the first incoming frame, we perform the native subtraction algorithm. It's defined by Algorithm 2. In this algorithm, ϵ_2 is the detection threshold. The superpixel is detected as foreground if no acceptable matching codeword exists. Otherwise, it is classified as background and the corresponding codeword is updated.

Algorithm 2: Foreground Pixel Extraction

$p_t(\bar{L}, \bar{a}, \bar{b})$
for *all codewords* **do**
    find the codeword c_m matching to Su_k based on :
        (a) colordist $(p_t, v_m) <= \epsilon_2$
        (b) (brightness ($\bar{L}$, $\check{L}_m$, $\hat{L}_m$)) = true
        Update the matched codeword as in Step 12 in the algorithm of background modeling (Algorithm 2).

$$BGS(Su_k) = \begin{cases} \textbf{foreground} & \text{if there is no match} \\ \textbf{background} & \text{otherwise} \end{cases}$$

With the following frames, after the extraction of the superpixels, we distinguish two groups of superpixels as shown in figure 1. The superpixels which belong to the first

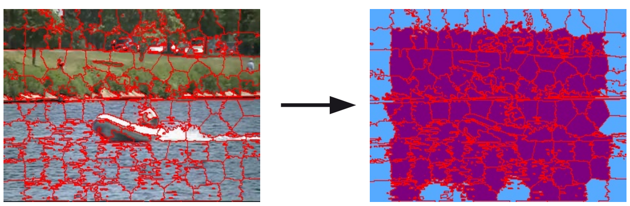

Fig. 1: Superpixels classification.

group are those that are on the border of the frame. For these superpixels, we perform the foreground/background segmentation by subtracting them from the corresponding superpixels in the background model by using algorithm 2. This part allows our algorithm to detect objects that enter the scene.

After that, we focus on the superpixels of the second group. The second group of superpixels is composed by the superpixels which are not on the border of the frame. We also use algorithm 2 to check if the superpixels that are adjacent to the background superpixels which belong to the first group are foreground superpixels or not. This part allows our algorithm to tolerate misdetection at the entrance of the scene.
At the end, the test using algorithm 2 is done on the rest of superpixels from the second group that are considered as foreground superpixels in the previous frame or adjacent to foreground superpixels foreground in the previous frame. With this result,if a superpixel is detected as foreground superpixel in the current frame then the superpixels which are adjacent to its and are not evaluated will be checked in order to verify if they are also foreground superpixels or not. This operation is repeated until all superpixels that are adjacent to foreground superpixels became background superpixels.

Using this strategy, we do not use necessarily all superpixels but we are only interested by significant superpixels depending on the evolution of the foreground information.

III. Experimental Results

For the validation of our single view algorithm, We have selected public benchmarking datasets (available from http://www.changedetection.net) which are covered under the work done by Goyette et al. [4]. They are “fall”, “boats”, “canoe”, “fountain01”, “fountain02” and “overpass” datasets. These datasets present a sequence with dynamic background and are made available to researchers for the evaluation of moving object detection algorithms.

The experiment environment is Intel Core i5 CPU L 640 @ $2.13GHz \times 4$ processor with 4GB memory and the programming language is C++. The parameters of superpixels segmentation algorithm is given by Schick et al.[11]. In our implementation if the size of input frame is $(M \times N)$ then we construct $\frac{M \times N}{50}$ superpixels as suggested by Mousse et al. [9]. Some segmentation results are presented in Figure 2 and Figure 3.

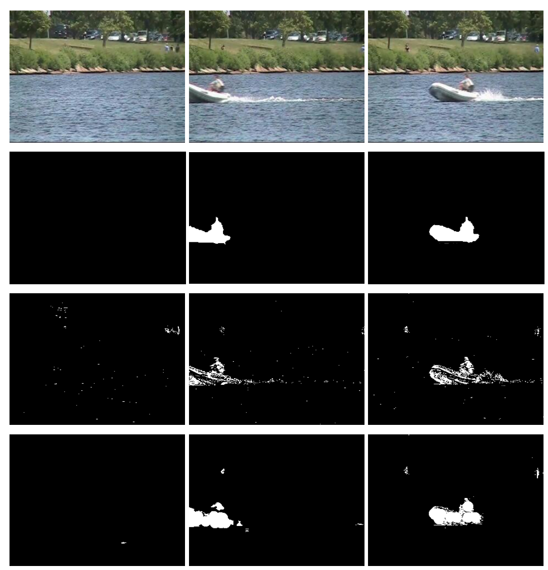

Fig. 2: “boats” dataset segmentation results. The first row shows the original images. The second row shows the ground truth. The third row shows the detected results by [6], and the last row shows the detected results by our proposed algorithm.

IV. Performance evaluation and discussion

In this part, we show the performance of the proposed approach by comparing with standard codebook algorithm proposed by [6], and with other extensions of codebook algorithm based on extension on pixels which are proposed in [3], [1], [2]. For the performance evaluation of our moving

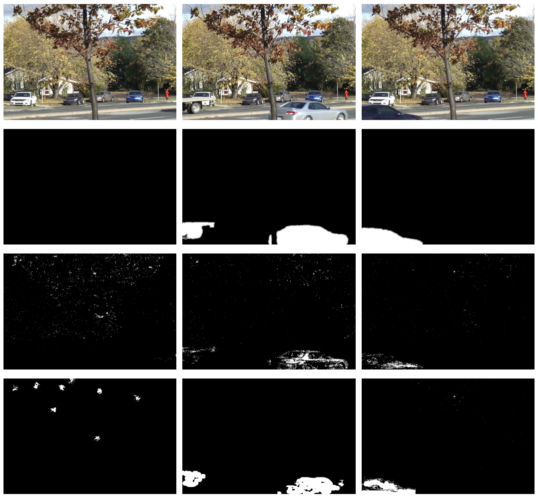

Fig. 3: “fall” dataset segmentation results. The first row shows the original images. The second row shows the ground truth. The third row shows the detected results by [6], and the last row shows the detected results by our proposed algorithm.

objects detection approach we used some quantitative metrics. For this, some frame based metrics were been used. The basic frame based metrics are : true negative (TN), true positive (TP), false negative (FN) and false positive (FP). A pixel is a true negative pixel when both ground truth and system result agree on the absence of object. A pixel is a true positive pixel when ground truth and system agree on the presence of objects. A pixel is a false negative (FN) when system result agree of absence of object whereas ground truth agree of the presence of object. A pixel is a false positive (FP) when the system result agrees with the presence of object whereas ground truth agree with the absence of object. With these metrics, we calculate :

- **False positive rate** (FPR) using formula (10);

$$FPR = 1 - \frac{TN}{TN + FP} \tag{10}$$

- **True positive rate** (TPR) using formula (11);

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

- **F-measure** (FM) using formula (12);

$$FM = \frac{2 \times PR \times TPR}{PR + TPR} \tag{12}$$

with

$$PR = \frac{TP}{TP + FP} \tag{13}$$

FPR allows us to quantify the number of false alarms which is sent by the algorithm. With TPR, we show the percentage of true positive pixel which have been detected. PR allows us to quantify the precision of object detection and FM represent a harmonic mean of precision and true positive rate. It measures the accuracy of video segmentation algorithm. According to

TABLE I: Different metrics according to experiments with “boats” dataset.

Metrics	CB	CB_HSV	CB_HSL	CB_YUV	CB_LAB
FPR	0.23	0.25	0.21	0.27	**0.21**
PR	0.87	0.86	0.89	0.80	**0.91**
FM	0.60	0.62	0.64	0.65	**0.66**

TABLE II: Different metrics according to experiments with “fall” dataset

Metrics	CB	CB_HSV	CB_HSL	CB_YUV	CB_LAB
FPR	0.31	0.33	0.25	0.38	**0.23**
PR	0.56	0.60	0.63	0.41	**0.67**
FM	0.41	0.47	0.51	0.43	**0.54**

TABLE III: Different metrics according to experiments with “canoe” dataset

Metrics	CB	CB_HSV	CB_HSL	CB_YUV	CB_LAB
FPR	0.16	0.18	0.15	0.21	**0.13**
PR	0.41	0.46	0.46	**0.52**	0.48
FM	0.35	0.38	0.37	**0.41**	0.39

TABLE IV: Speed comparison

Datasets	Mousse et al. [9]	Proposed approach
overpass	75.82	**92.15**
fall	63.15	**75.78**
fountain02	73.15	**91.44**
fountain01	70.52	**95.91**

our experiment, we conclude that our algorithm has the same performance with the algorithm proposed in [7] and according to the results presented in Table I, Table II and Table III, we prove that the accuracy of the proposed system is close to the accuracy of other codebook based algorithm enhancements. In these tables, CB refers to the method proposed by Kim et al., and CB_HSV refers to the method suggested by Doshi and Trivedi, CB_HSL refers to the algorithm proposed by Fang et al., CB_YUV refers to the algorithm suggested by Cheng et al., and CB_LAB refers to our proposed algorithm.

To assess the contribution of our approach, we calculate our execution time (in frame per second) and compare it to the execution time of Mousse et al. [9] (Fang et al. [3] and Mousse et al. [9] are the fastest methods among all the improvements of codebook with practically the same execution time). The results are reported in Table IV. In this table, the speed is expressed in frames per second. According to values presented in Table IV, our approach improves processing time at least by 20%. The gain is more substantial when the size of the foreground objects is small. This is evidenced by experiments based on the sequence fountain01 where the speed was increase by 36%.

V. Conclusion

This paper presents a new method to detect moving objects in video sequence based on codebook background model. It suggests an adaptive method to reduce the amount of calculation. The perceptual visual information is used to actively focus the attention on important parts of the image. Experiments in dynamic scene verify that our proposed approach outperforms other algorithms in calculation cost and has an accuracy which is close to the state of art algorithms.

REFERENCES

[1] X. Cheng, T. Zheng, and L. Renfa. A fast motion detection method based on improved codebook model. *Journal of Computer and Development*, 47:2149–2156, 2010.

[2] A. Doshi and M. M. Trivedi. Hybrid cone-cylinder codebook model for foreground detection with shadow and highlight suppression. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 121–133, 2006.

[3] X. Fang, C. Liu, S. Gong, and Y. Ji. Object detection in dynamic scenes based on codebook with superpixels. In *Proceedings of the Asian Conference on Pattern Recognition*, pages 430–434, 2013.

[4] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2012.

[5] J.-M. Guo, Y.-F. Liu, C.-H. Hsia, M.-H. Shih, and C.-S. Hsu. Hierarchical method for foreground detection using codebook model. *IEEE Transactions on Circuits and Systems for Video Technology*, volume 21(6):294–301, 2011.

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

[7] M. A. Mousse, E. C. Ezin, and C. Motamed. Foreground-background segmentation based on codebook and edge detector. In *Proceedings of the 10th International Conference on Signal Image Technology & Internet Based Systems*, 2014.

[8] M. A. Mousse, C. Motamed, and E. C.Ezin. Percentage of human-occupied areas for fall detection from two views. *The Visual Computer*, 2016.

[9] M. A. Mousse, C. Motamed, and E. C. Ezin. Fast moving object detection from overlapping cameras. In *Proceedings of the In International Conference on Informatics in Control, Automation and Robotics*, pages 296–303, 2015.

[10] A. Pal, G. Schaefer, and M. E. Celebi. Robust codebook-based video background subtraction. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1146–1149, 2010.

[11] A. Schick, M. Fischer, and R. Stiefelhagen. Measuring and evaluating the compactness of superpixels. In *Proceedings International Conference on Pattern Recognition*, pages 930–934, 2012.