Multi-Layer Multi-Feature Background Subtraction Using Codebook Model Framework

Yun-Tao Zhang, Jong-Yeop Bae, Whoi-Yul Kim

***Abstract*—**Background modeling and subtraction in video analysis has been widely used as an effective method for moving objects detection in many computer vision applications. Recently, a large number of approaches have been developed to tackle different types of challenges in this field. However, the dynamic background and illumination variations are the most frequently occurred problems in the practical situation. This paper presents a favorable two-layer model based on codebook algorithm incorporated with local binary pattern (LBP) texture measure, targeted for handling dynamic background and illumination variation problems. More specifically, the first layer is designed by block-based codebook combining with LBP histogram and mean value of each RGB color channel. Because of the invariance of the LBP features with respect to monotonic gray-scale changes, this layer can produce block wise detection results with considerable tolerance of illumination variations. The pixel-based codebook is employed to reinforce the precision from the output of the first layer which is to eliminate false positives further. As a result, the proposed approach can greatly promote the accuracy under the circumstances of dynamic background and illumination changes. Experimental results on several popular background subtraction datasets demonstrate very competitive performance compared to previous models.

***Keywords*—**Background subtraction, codebook model, local binary pattern, dynamic background, illumination changes.

I. Introduction

DETECTION of moving objects in a video sequence is a fundamental task to analyze and understand significant information in many computer vision applications. Commonly, accurate foreground objects detection is referred as background subtraction [1]. Generally, background is viewed as stationary or quasi-periodic areas of the visual scene, whereas, foreground is regarded as moving objects in the scene [2]. The main idea behind the background subtraction is the separation of moving objects from the background, inside of the video frame. In recent years, although a great number of background subtraction approaches [3], [4] have been proposed, there still remains many challenges in this task such as dynamic background, illumination changes, bootstrapping, camouflage, and camera jitter. To solve these problems, one of the most popular background subtraction methods is the Mixture-of-Gaussian (MOG) [5], which utilizes several weighted Gaussian distributions to represent each pixel in the background modeling. This approach can offer great robustness under the circumstances of dynamic backgrounds. However, the parameters of Gaussians are quite sensitive to the background. A cluster method which constructs a codebook that consists of one or more codewords for each pixel was proposed in [1]. This model clusters samples at each pixel into a set of codewords in terms of a color distortion and brightness bounds. Although, this method can also handle dynamic background problem well, it will produce unstable results with high computational cost under illumination changes due to expanded generation of codewords in dark and noisy regions [3]. In addition to this, local binary pattern (LBP) has been used to model the background with a group of weighted adaptive LBP histograms in order to obtain the background statistics of each image block [6]. In terms of monotonic gray-scale changes of LBP features, the algorithm in [6] can obtain robust results with considerable tolerance of gray-scale variations together with fast processing merit. In another approach, samples based algorithm called Visual Background Extractor (ViBe) [7] is introduced. This approach builds the background model by aggregating previously observed values for each pixel location.

The prime goal of the proposed work is to solve both the dynamic background and the illumination change problems together. In this paper, an advantage of codebook model [1] is taken that is capable of solving dynamic background problem and the merits of LBP texture. In particular, LBP texture has the ability of handling illumination changes, and it is also able to compute results fast as well, due to the simple thresholding scheme. Therefore, a two-layer two-feature background modeling approach is presented for moving objects detection. The proposed approach uses the LBP texture histograms to build the first layer to obtain block wise detection result, and then employs codebook algorithm over the output of the first layer to achieve pixel wise detection result.

The remaining of this paper is organized as follows. The original codebook algorithm is first introduced in Section II. Section III describes the proposed algorithm which contains three parts: the block-based codebook established by LBP texture histogram, pixel-based codebook, and foreground detection process. The experimental results on several popular complex video datasets are presented in Section IV. The paper conclusion is in Section V.

Yun-Tao Zhang is with the Electronics, Communications and Computer Engineering Department, Hanyang University, Seoul, South Korea (corresponding author, phone: +82-02-2220-0561, fax: +82-2-2292-6316, e-mail: yuntao.zhang@vision.hanyang.ac.kr).

Jong-Yeop Bae is with the Electronics, Communications and Computer Engineering Department, Hanyang University, Seoul, South Korea (e-mail: jybae@vision.hanyang.ac.kr).

Whoi-Yul Kim is with the Electronic Engineering Department, Hanyang University, Seoul, South Korea (e-mail: wykim@hanyang.ac.kr).

II. Previous Work

A. Original Codebook Algorithm

The classic codebook algorithm is proposed in [1]. The whole process of this method consists of two phases: training phase, and detection phase.

Considering N training sequences for a single pixel to include N RGB-vectors: $\chi = \{x_1, x_2, \ldots, x_N\}$. Each single pixel is represented by a codebook $c = \{c_1,\ c_2, \ldots, c_L\}$ with length L. The length of codebook on each pixel can be different from other codebooks at different positions. Every codeword c_i is represented by a RGB vector $v_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ together with a 6-tuple $aux_i = \{\check{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i\}$, where $\check{I}$ and $\hat{I}$ are the min and max brightness of all pixels appointed to this codeword, correspondingly; f is the frequency with which the codeword has occurred; λ is the maximum negative run-length (MNRL) defined as the longest interval during the training period that the codeword has not recurred; p and q are the first and last access times that the codeword has occurred, respectively. The entire training process of codebook is described as:

I. Initialize $L \leftarrow 0, c \leftarrow \phi$

II. For $t = 1$ to N

- $x_t = (R, G, B),\ \ I \leftarrow \sqrt{R^2 + G^2 + B^2}$
- Find the codeword c_m in $c = \{c_1, c_2, \ldots, c_L\}$ matching to x_t based on two conditions (i) and (ii).

  i. $colordist(x_t, v_m) \le \varepsilon$

  ii. $brightness(I, \langle \check{I}_m, \hat{I}_m \rangle) = true$
- If $c = \phi$ or there is no match, then $L \leftarrow L + 1$. Create a new code word c_L by setting

  i. $v_L \leftarrow (R, G, B)$

  ii. $aux_L \leftarrow \{I, I, 1, t-1, t, t\}$.
- Otherwise, update the matched code word by setting

  i. $v_m \leftarrow \left(\dfrac{f_m \bar{R}_m + R}{f_m + 1}, \dfrac{f_m \bar{G}_m + G}{f_m + 1}, \dfrac{f_m \bar{B}_m + B}{f_m + 1} \right)$

  ii. $aux_m \leftarrow \{\min(I, \check{I}_m), \max(I, \hat{I}_m), f_m + 1, \max(\lambda_m, t - q_m), p_m, t\}$.

End for

III. For each codeword c_i, $i = 1, \ldots, L$, wrap around λ_i by setting $\lambda_i \leftarrow \max\{\lambda_i, (N - q_i + p_i + 1)\}$.

For each input pixel, if it can be matched with any codeword at the corresponding position, according to matching criteria (color distortion and brightness bounds), then the algorithm updates matched codeword, otherwise it adds a new codeword. Similarly, the input pixel is categorized into background or foreground depending on two judging criteria in the detection phase.

The color distortion is defined as:

$$colordist(x_t, v_i) = \delta = \sqrt{(R^2 + G^2 + B^2) - \frac{(\bar{R}_i R + \bar{G}_i G + \bar{B}_i B)^2}{\bar{R}_i^2 + \bar{G}_i^2 + \bar{B}_i^2}}, \quad (1)$$

where, (R, G, B) represents the RGB vector of incoming pixels, whereas, $(\bar{R}_i, \bar{G}_i, \bar{B}_i)$ represents the RGB vector of corresponding codeword.

The brightness function is defined as:

$$brightness(I, \langle \check{I}, \hat{I} \rangle) = \begin{cases} \text{true} & \text{if } \alpha \hat{I} \le \|x_t\| \le \min\left\{ \beta \hat{I}, \dfrac{\check{I}}{\alpha} \right\} \\ \text{false} & \text{otherwise} \end{cases}, \quad (2)$$

where, α and β are the user-defined constant parameters.

III. Proposed Method

The main contribution of this paper is the combination of the LBP texture information and the codebook algorithm framework. The goal of this combination is to improve the detection accuracy of moving objects under the circumstances of dynamic background and illumination changes. The proposed method consists of a two-layer model which the block-based codebook is built to get block wise detection result in the first layer, the second layer borrowed the idea of original codebook algorithm to obtain pixel wise detection result. The original pixel-based codebook is introduced in Part A. The block-based codebook is described in Part B. Part C briefly introduces the pixel-based codebook. Part D explains the foreground detection process.

B. Construction of Block-Based Codebook

Suppose input frames of size $W \times H$ pixels, after calculating the circular LBP ([8]) code for every pixel on the gray scale image, the LBP texture frame is divided into non-overlapped blocks of size $M \times N$ and process each block independently. In both training phase and detection phase, the difference measurement between incoming LBP histogram H_t and existing L model histograms at the corresponding position is evaluated as:

$$D(H_1, H_2) = \sum_i |H_1[i] - H_2[i]|, \quad (3)$$

where i is the bin index of each histogram. The user defines the threshold value T_H for the histogram distance as a method parameter. Throughout this paper, all of the histograms are normalized.

Unfortunately, LBP histogram is not robust, particularly, when background and foreground have similar structure due to the simple thresholding mechanism of LBP. This drawback results in black hole in the output frame. As shown in Fig. 1, total different pixel values may have identical binary number, so it is likely to construct a similar histogram over a region at the same position. As a result, the moving foreground is merged into background. In order to solve this problem, this paper takes advantage of color distance with respect to each mean value of color channels between foreground and background to provide better discrimination result. The mean values are obtained over each block. This technique is similar with [9]; however, the approach in [9] calculates many more mean values to provide a good discrimination capability resulting in higher computation time. As this manner is just used as a complementary approach for the LBP block histogram, it is unnecessary to calculate many high-mean and low-mean values. The distance measurement of each block on three color channels is given by:

$$D(\vec{d_1},\vec{d_2}) = \begin{cases} true & if \ \dfrac{\vec{d}^T \vec{d}}{\dim(\vec{d})} < T_D \\ false & otherwise \end{cases}, \tag{4}$$

where $\vec{d} = \vec{d_1} - \vec{d_2}$. The user defines the threshold value T_D for the color distance as a method parameter.

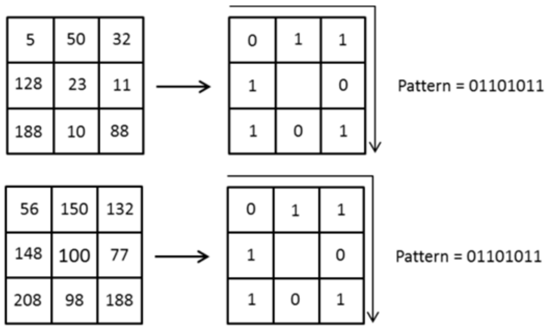

Fig. 1 An example of drawback of LBP

Considering N training sequences, a single block is represented by a codebook $c = \{c_1, c_2, \ldots, c_L\}$ with length L. Each codeword c_i is represented by a LBP histogram H_i , mean value vector of RGB three channels $\overline{I}_i = (\overline{R}_i, \overline{G}_i, \overline{B}_i)$ and together with a 4-tuple $aux_i = \{f_i, \lambda_i, p_i, q_i\}$. The meaning of each auxiliary member in the tuple is equivalent to the original codebook model. The training phase of LBP histogram based block codebook algorithm is given below:

I. Initialize $L \leftarrow 0, c \leftarrow \phi$

II. For $t = 1$ to N

- H_t, $\overline{I}_i \leftarrow (\overline{R}_i, \overline{G}_i, \overline{B}_i)$
- Find a codeword c_m in $c = \{c_1, c_2, \ldots, c_L\}$ matching to x_t based on two conditions (i) and (ii).
  i. $histogram\_dist(H_t, H_m) \le T_H$
  ii. $color\_dist(\overline{I}_t, \overline{I}_m) \le T_D$
- If $c = \phi$ or there is no match, then $L \leftarrow L + 1$. Create a new codeword c_L by setting
  i. $H_L \leftarrow H_t$
  ii. $\overline{I}_L \leftarrow (\overline{R}_t, \overline{G}_t, \overline{B}_t)$
  iii. $aux_L \leftarrow \{1, t-1, t, t\}$
- Otherwise, update the matched codeword by setting
  i. $H_m[\mathrm{i}] = \alpha_b H_t[\mathrm{i}] + (1-\alpha_b) H_{m-1}[\mathrm{i}]$
  ii. $\overline{I}_{m,k} = \beta_b \overline{I}_{t,k} + (1-\beta_b)\overline{I}_{m-1,k}$
  iii. $aux_m \leftarrow \{f_m + 1, \max(\lambda_m, t - q_m), p_m, t\}$.

End for

where α_b and β_b are user defined learning rates and k is the index of color channel for each block.

C. Pixel-Based Codebook

The proposed multi-layer multi-feature model is designed to train the block-based codebook layer and the pixel-based layer, simultaneously. The idea of pixel-based model is borrowed from the original Codebook algorithm [1]. One of the advantages of original Codebook algorithm is that it does not assume the potential parametric distributions of the background. This merit makes it has enough ability to handle multiple backgrounds situation.

D. Foreground Detection

After the training process of two background models, the constructed block-based and pixel-based codebooks are utilized to do the foreground detection. During the detection process, block-based codebook and pixel-based codebook are both updated to conquer dynamic background problem. The whole detection process is given below:

I. Construct LBP histogram for each block of every incoming video frame;

II. Match against LBP block-based codebook based on histogram difference and block color distance:

- If there is a match, update corresponding codeword and output this block as the background;
- Otherwise, input the block to the pixel-based codebook model to check every pixel based on the color distortion and brightness bounds:
  i. If there is a match, output the pixel as background and update corresponding matched codeword in the pixel-based codebook model;
  ii. Otherwise, output the pixel as foreground.

IV. Experimental Results

The proposed model has been evaluated on selected benchmarking datasets from [10], [11]. As shown in Fig. 2, they are “People in Shade”, “Waving Trees” and “Canoe” datasets. The experimental setup includes Intel Core i5-3570 CPU@3.40Hz with 4GB RAM memory. The parameters setting for pixel-based codebook are $\alpha = 0.5$ and $\beta = 1.3$. Color distortion threshold is set as 20. For the LBP block-based codebook, the number of circular neighborhood for LBP pattern calculation is selected as 8 with respect to the radius of 3. The block size is set to 8×8 , and the learning rate for histogram updating (α_b) and color channel mean value updating (β_b) are both 0.01. The histogram difference thresholds are set as 0.35 and 1.4, for training phase and detection phase, respectively.

The block color distance threshold T_D was setting by 100. The first 50 frames of each dataset are used to train the background model.

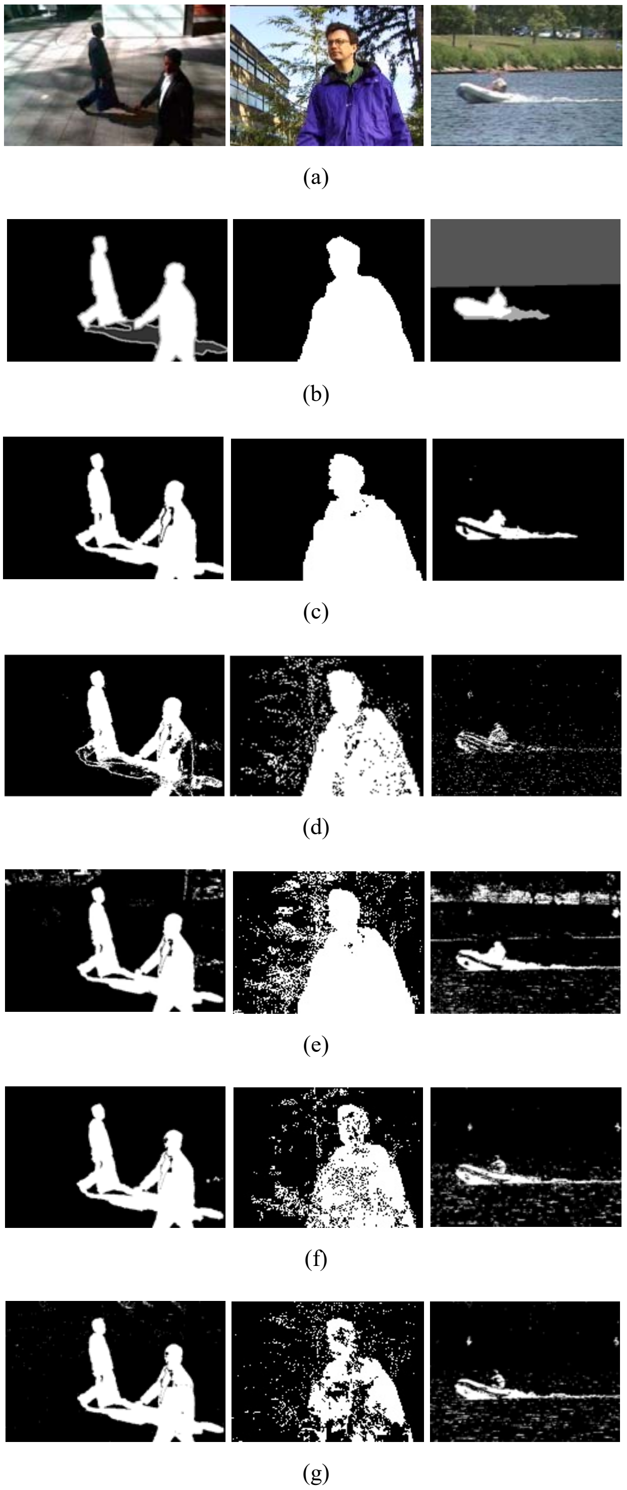

Fig. 2 (a) Input frames, (b) Ground truth, (c) Proposed approach, (d) MOG [5], (e) CB [1], (f) Hierarchical CB [9], (g) ViBe [7]

Fig. 2 shows the results. The performance of proposed approach, MOG (implemented by [12]), Codebook, Hierarchical Codebook, and ViBe are tested under the same hardware condition. The first dataset is the illumination changes dataset, the second and the third datasets are the dynamic background datasets.

In terms of the illumination changes dataset, the proposed method can provide very clean background and complete foreground masks without any post-processing manners compared to the previous models. However, there are still some jags in the boundary of the foreground masks because of the block effect. MOG can also work very well, but it generates black hole in the foreground mask because of the sensitive parameters. ViBe and Hierarchical Codebook perform better than original Codebook. Unlike MOG and Hierarchical Codebook models which can eliminate shadows, the performance of the proposed approach is pretty good, but it cannot greatly detect the tiny moving objects especially when they are appearing in the shadows.

For dynamic background datasets, the output of original Codebook method is better in detecting foreground than Hierarchical Codebook and ViBe, but having many false results. To some extent, statistical model MOG can work better than Codebook and ViBe with dynamic background. The proposed approach can work robustly in dynamic background without many noises. However, sometimes the proposed model fails if the color of moving objects is very close to the background, as it is the intrinsic drawback of the codebook algorithm due to its use of color similarity measurements between foreground and background. This is clear from the third column of Fig. 2. There is a black gap split in the canoe body because the color of that area is very close to the color of the water surface. The similar phenomenon can also be found in the related Codebook algorithms.

V. Conclusion

In this paper, a favorable multi-layer multi-feature model is proposed, based on codebook algorithm associated with local binary pattern. The prime goal of this paper is to solve dynamic background and illumination change problems. Due to the great merit of handling illumination change problem of LBP and the powerful capability of handling multiple background problem of codebook algorithm, the proposed algorithm can work well. The performance can be seen from the experimental part.

References

[1] K. Kim, H. C. Thanarat, H. David, and L. Davis. "Real-time foreground–background segmentation using codebook model." Real-time imaging 11, no. 3 (2005): 172-185. 15–64.

[2] M. Shah, J. D. Deng, and B. J. Woodford. "A Self-adaptive CodeBook (SACB) model for real-time background subtraction." Image and Vision Computing 38 (2015): 52-64.

[3] S. Brutzer, B. Höferlin, and G. Heidemann. "Evaluation of background subtraction techniques for video surveillance." In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 1937-1944. IEEE, 2011.

[4] T. Bouwmans, F. El Baf, and B. Vachon. "Background modeling using mixture of gaussians for foreground detection-a survey." Recent Patents on Computer Science 1, no. 3 (2008): 219-237.

[5] C. Stauffer, and W. E. L. Grimson. "Adaptive background mixture models for real-time tracking." In Computer Vision and Pattern

Recognition, 1999. IEEE Computer Society Conference on., vol. 2. IEEE, 1999.

[6] M. Heikkilä, M. Pietikäinen, and J. Heikkilä. "A texture-based method for detecting moving objects." In BMVC, pp. 1-10. 2004.

[7] O. Barnich, and M. Van Droogenbroeck. "ViBe: a powerful random technique to estimate the background in video sequences." In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pp. 945-948. IEEE, 2009.

[8] T. Ojala, M. Pietikäinen, and T. Mäenpää. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." Pattern Analysis and Machine Intelligence, IEEE Transactions on 24, no. 7 (2002): 971-987.

[9] J. M. Guo, Y. F. Liu, C. H. Hsia, M. H. Shih, and C. S. Hsu. "Hierarchical method for foreground detection using codebook model." Circuits and Systems for Video Technology, IEEE Transactions on 21, no. 6 (2011): 804-815.

[10] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. "Changedetection. net: A new change detection benchmark dataset." InComputer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pp. 1-8. IEEE, 2012.

[11] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. "Wallflower: Principles and practice of background maintenance." In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 1, pp. 255-261. IEEE, 1999.

[12] A. Sobral. "BGSLibrary: An opencv c++ background subtraction library." In IX Workshop de Visao Computacional (WVC'2013), Rio de Janeiro, Brazil. 2013.