

Pixel-wise Background Segmentation with Moving Camera

Neeraj Mishra¹, Manas Kamal Bhuyan¹, T. Malathi¹, Yuji Iwahori²,
and Robert J. Woodham³

¹ Department of Electronics and Electrical Engineering, IIT Guwahati, India
neeraj1928@gmail.com, {mkb,malathi}@iitg.ernet.in

² Department of Computer Science, Chubu University, Japan
iwahori@cs.chubu.ac.jp

³ Department of Computer Science, University of British Columbia, Canada
woodham@cs.ubc.ca

Abstract. This paper proposes a novel approach for background extraction of a scene captured by a moving camera. Proposed method uses a codebook, which is a compression technique used to store data from a long sequence of video frames. This technique has been used to construct a model which can segment out the foreground and that with using only few initial video frames as a training sequence. It is a dynamic model which keeps on learning from new video frames throughout its lifetime and simultaneously produces the output. It uses a pixel-wise approach, and the codebooks for each pixel are made independently. Special emphasis has been laid on the intensity of an image as the human eye is more sensitive to intensity variations. A two layer modelling has been performed where codebooks are passed from the cache to the background model after satisfying the frequency and negative run length conditions. Experimental results show the efficacy of the proposed method.

Keywords: Video Surveillance, Codebook, Background Modelling.

1 Introduction

Separation of foreground objects from a video sequence is the foremost task in video surveillance systems, tracking systems, etc. The tracking system gets boosted up significantly if we have a good reliable dynamic real-time background model. The most important aspect for a background model to be successful is that it should be able to handle a dynamic background. Additionally, a background model should efficiently handle a dynamic scene. The typical cases of moving tree leaves, water ripples, illumination change, etc must be handled by the model in a computationally efficient manner. Besides this, there may be chances of movement of the camera. In such cases, the model must possess a dynamic learning capability.

Earlier, unimodal background modelling was done for all the pixels [1]. Then, this approach was changed to mixture of Gaussians on all the pixels [2]. Models

using Mahalanobis distance and image vectors were proposed in [3]. However, all these approaches fail for the scenarios involving fast variations in the background. Background modelling using codebook method was proposed in [4]. In this method, initially a long video sequence was used as a training data, which was stored in the compressed form at each pixel as codebooks. So, a static background model was formed. This model was able to handle the case of shadows and illumination change. But, it fails when any objects in the scene which was stationary during the training period moves afterwards. This happens because the model was totally dependent on the initial scenes. An improvement of it was given in [5], where after developing the model based on the training samples, another cache model was made. If any new object comes in front of the camera for a longer duration, then it will be included in the background. But a drawback here is that one has to wait till the training period to get over before getting the output, since the system is in the learning stage during that period of time. Also, it can not handle motion of the camera. A hybrid codebook model was proposed in [6] which uses the mixture of Gaussians. This model gives good results, however, for a setup involving a moving camera, it is unable to give the desired results. A model which considered the case of movable camera has been proposed in [7]. This model makes use of a homography matrix, however, it is computationally very complex.

To address some of these issues, a novel background modeling algorithm is proposed which can handle small motions of the camera with a lesser computational cost. The proposed technique does not require any dedicated training period as it is based on dynamic learning. The proposed algorithm is also seen to easily adapt to new scenes as the camera moves.

2 Proposed Method

In order to successfully deal with the situations wherein the camera shows some kind of movements, the proposed method handles each of the pixels separately. Initially the entire scene is taken as the foreground, and from that the model starts forming the codebook. Additionally, no dedicated time is allotted for the training period. The proposed method only makes use of 2-tuples $\langle \lambda_i, f_i \rangle$. Also, unlike the existing algorithms [4], [5], Maximum Negative Run Length (MNRL) is not used in the proposed method. Instead, the proposed method makes use of only the latest negative run length λ of the codebook.

Initially the codebook is formed by considering two parameters $\lambda_i = 0$ and the frequency $f_i = 1$. This codebook is included in the background model only when the following conditions are satisfied:

$$\lambda_i < \epsilon_1, \quad f_i > \epsilon_2 \quad (1)$$

The first condition in (1) indicates that for a codebook to be considered as the background, its negative run length must be less than some constant ϵ_1 . The second condition implies that the object should remain in front of the camera for a certain minimum number of frames ϵ_2 to be considered as the background.

The proposed technique uses the YCbCr colour space. As Y space accounts more for the luma component of the image, more tolerance has been given to it than the other two Cb and Cr channels which account for the chroma component of the image. This is done in order to counter the effects of sudden shadow or highlights due to illumination change. In the case of intensity change, the value of Y component mostly changes. Hence, although the object was supposed to be in the background comes in the foreground. Hence more tolerance has been given in the Y axis for learning the codebook than the Cb and Cr in YCbCr colour space.

Let $C = c_1, c_2, \dots, c_i$ represent the codebook for the pixel consisting of L codewords. Each pixel has a unique codebook size based on its sample variations. Each codeword consist of YCbCr colour details and 2-tuples $aux_i = \langle \lambda_i, f_i \rangle$. For any pixel x_t at time t , $x_t = \langle Y_t, Cb_t, Cr_t \rangle$. The flowchart of the proposed algorithm is shown in Fig. 1.

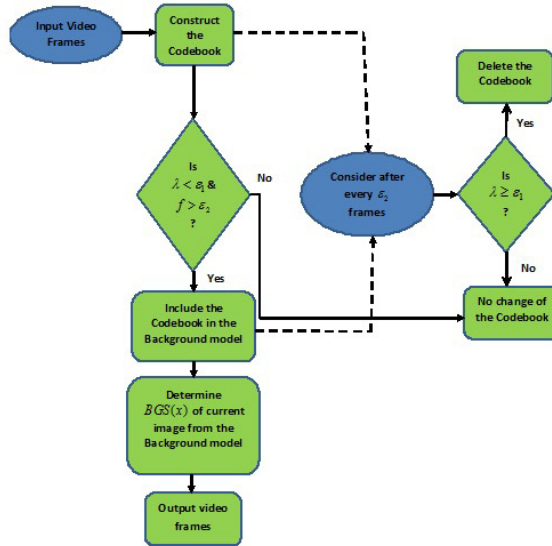


Fig. 1. Proposed Algorithm

Handling of Camera Motion: When the camera moves, the values at each pixel starts changing. As the pixel starts getting new values, it will be considered as the foreground. The background is defined as something which should be in front of a camera for at least ϵ_2 frames. Also at the same time, its negative run length should be less than ϵ_1 *i.e.*, the codebook should not be absent at that pixel for that number of frames. Generally at a particular pixel, we have 2 to 6 codebooks. For each frame, at each pixel, the λ and f value of all the codebooks are modified which are present at that pixel. First, the match with the value of the current pixel is checked, if the match is found then the λ is

assigned as zero and the f is incremented by 1. For the remaining codebooks, λ is increased by 1 and f remains the same. If there is no match, then a new codebook is formed. Additionally, the λ value of all the earlier codebooks will be incremented by 1, while the f value remains the same. This is the structure of the proposed cache model. Then in this model, at each pixel, the codebooks which satisfy the condition of the background are searched. Those which can satisfy this requirement are inducted in the background model B codebooks. Subsequently, the background subtraction of the current frame is implemented from this B model. Also after every ϵ_2 frames, the λ values of all the codebooks at each pixel in both the models are checked and those having values greater than ϵ_1 are deleted as there is no need of storing the values which are not repeating itself at the pixel. This method can efficiently handle very slow camera motions in any directions. The proposed method for the construction of codebook is detailed as follows:

Proposed Algorithm for Foreground Extraction

- I $L \leftarrow 0$, $\mathbf{C} \leftarrow \phi$ (empty set)
 - II **for** $t \geq 1$ **do**
 - i. $x_t = (Y_t, Cb_t, Cr_t)$
 - ii. Find the codeword \mathbf{c}_m in $\mathbf{C} = [\mathbf{c}_i | 1 \leq i \leq L]$ matching to x_t based on following two conditions:
 - (a) colordistance: $|Cb_t - Cb_m| \leq \delta_1$, $|Cr_t - Cr_m| \leq \delta_1$
 - (b) brightness: $|Y_t - Y_m| \leq \delta_2$
 - iii. If $\mathbf{C} = \phi$ or there is no match then $L \leftarrow L + 1$.
Create a new codeword \mathbf{c}_L by setting:
 $v_L \leftarrow (Y, Cb, Cr)$
 $aux_L \leftarrow (0, 1)$
 - iv. Else update the matched codeword \mathbf{c}_m consisting of
 $v_m = (Y, Cb, Cr)$ and $aux_m = (\lambda_m, f_m)$ by setting:
 $v_m \leftarrow \left(\frac{f_m \bar{Y} + Y}{f_m + 1}, \frac{f_m \bar{Cb} + Cb}{f_m + 1}, \frac{f_m \bar{Cr} + Cr}{f_m + 1} \right)$
 $aux_m \leftarrow (0, f_m + 1)$
 - III For each codeword $\mathbf{c}_i, i = 1, \dots, L$
If $f_m > \epsilon_1$ & $\lambda < \epsilon_2$ then move that codeword in the Background Model \mathbf{B}
 - IV Background Subtraction
 - i. $\mathbf{x} = (Y, Cb, Cr)$
 - ii. For all codewords in Background Model \mathbf{B} , find the matching codewords based on two conditions:
 - (a) colordistance: $|Cb_t - Cb_m| \leq \delta_3$, $|Cr_t - Cr_m| \leq \delta_3$
 - (b) brightness: $|Y_t - Y_m| \leq \delta_4$
 Update the matched codeword as in Step II iv.
 - iii. $BGS(x) = \begin{cases} \text{Background,} & \text{if match found} \\ \text{Foreground,} & \text{otherwise} \end{cases}$
 - V After every ϵ_2 frames, all codebooks having $\lambda \geq \epsilon_1$ will be deleted.
-

In this, the thresholds $\delta_1, \delta_2, \delta_3$ and δ_4 are judiciously selected to measure colour and brightness distortions for a particular video sequence.

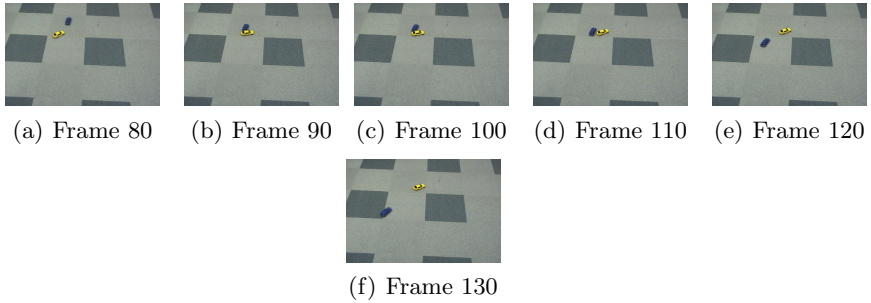


Fig. 2. Input video frames

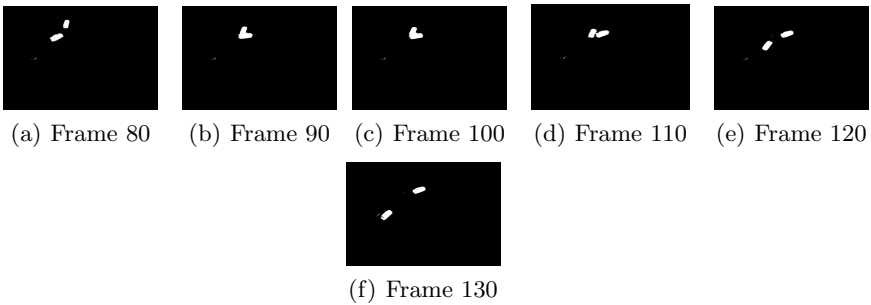


Fig. 3. Detection performance of the proposed method for input images in Fig. 2

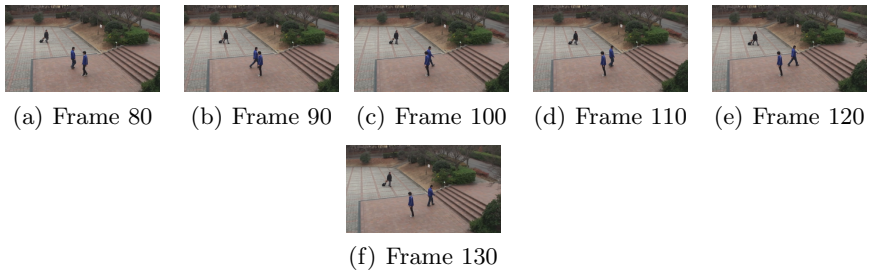


Fig. 4. Input video frames

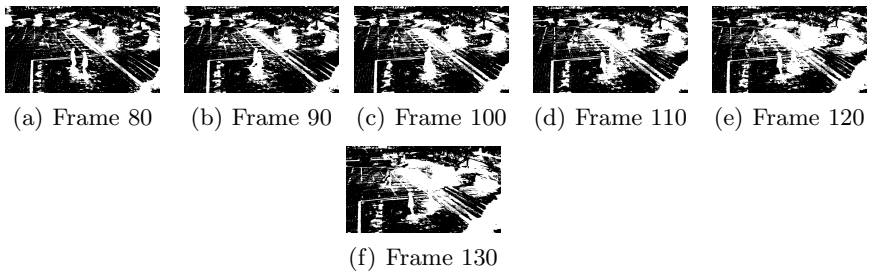


Fig. 5. Detection performance of the existing method for input images in Fig. 4

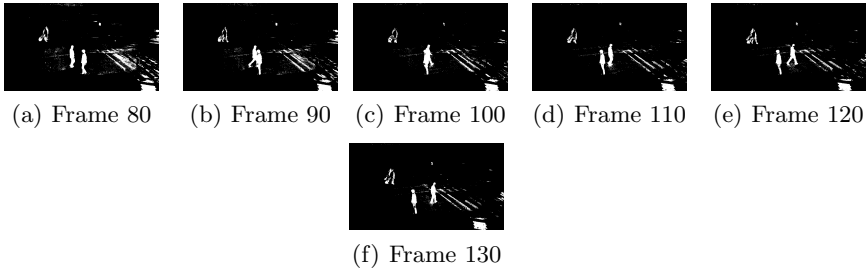


Fig. 6. Detection performance of the proposed method for input images in Fig. 4

3 Experimental Results

The proposed algorithm has been made to run on the Core 2 Duo E8400 3.00GHz processor with 4.00 GB RAM. The size of the video frames is 720 x 480 pixels. The proposed method has been tested for two different scenarios involving a static and a moving camera. The results obtained have been detailed under Case Study 1 and Case Study 2 respectively. With static camera (Case 1), the speed of proposed algorithm is around 37ms/frame while with the moving camera (Case 2) it is around 47ms/frame. Values of ϵ_1 and ϵ_2 are selected as 25 and 50 respectively.

Case Study 1 (Static Camera): In this, Fig. 2 shows the input images, and Fig. 3 demonstrates the foreground detection performance using the proposed algorithm. It is observed that the proposed approach does not have unwanted small noises, unlike the results from earlier codebook methods in [4], [5]. Usually, the earlier methods make use of post-processing techniques like the 8-connected component labeling on the output in order to remove these unwanted small noises.

Case Study 2 (Moving Camera): For this, a moving camera is taken to capture the videos and the camera moves very slowly in any directions. The input images of the video sequence are shown in Fig. 4. The detection performance using the existing methods [4], [5] is given in Fig. 5. Finally, Fig. 6 demonstrates the detection performance of the proposed algorithm. The proposed method clearly outperforms the existing methods.

4 Conclusions

The proposed foreground background segmentation method is able to detect the foreground moving objects without making use of long sequences of video frames as a training data. Unlike previous methods, the parameters being passed to the codebook element are only the frequency f , and the negative run length λ which has resulted in a significant reduction in the computation time and better memory utilization, thereby enabling effective real-time implementation. Also, by giving extra allowance to the luminance component than chrominance,

the shadow effects has been countered efficiently. The efficacy of the proposed method for real-time background segmentation in the scenarios involving a moving camera has also been established.

Acknowledgement. Iwahori's research is supported by JSPS Grant-in-Aid for Scientific Research (C) (23500228) and Chubu University Grant. Woodham's research is supported by the Natural Sciences and Engineering Research Council (NSERC).

References

1. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfunder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7), 780–785 (1997)
2. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 246–252 (1999)
3. Seki, M., Fujiwara, H., Sumi, K.: A robust background subtraction method for changing background. In: *5th IEEE Workshop on Applications of Computer Vision*, pp. 207–213 (2000)
4. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Background modeling and subtraction by codebook construction. In: *Proc. of the IEEE Conf. on Image Processing*, pp. 3061–3064 (2004)
5. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. *Real-time Imaging* 11(3), 172–185 (2005)
6. Sun, I.-T., Hsu, S.-C., Huang, C.-L.: A hybrid codebook background model for background subtraction. In: *IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 96–101 (2011)
7. Jin, Y., Tao, L., Di, H., Rao, N.I., Xu, G.: Background modeling from a free-moving camera by multi-layer homography algorithm. In: *Proc. of the IEEE Conf. on Image Processing*, pp. 1572–1575 (2008)