Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I
IMECS 2008, 19-21 March, 2008, Hong Kong

Real-time Background Modeling/Subtraction using Two-Layer Codebook Model

Mohamad Hoseyn Sigari, Mahmood Fathy

***Abstract*—In this paper, a real-time method for foreground-background segmentation is proposed. This method extracts structure of background and models it in layered codebook. Layered codebook is a simple data structure containing two codebooks that is defined per pixel. The first layer is main codebook, other is cache codebook, and both contain some codewords relative to a pixel. Main codebook models the current background images and cache codebook is used to model new background images during input sequence.**

Initially main and cache codebooks are empty. Main codebook is constructed during train sequence. During input sequence, foreground-background is segmented and two-layered codebook is updated. Proposed algorithm can model moving backgrounds, multi backgrounds and illumination changes and this is efficient in both memory and computational complexity.

***Index Terms*—Background Modeling, Background Subtraction, Codebook Model, Illumination Invariant.**

I. Introduction

Background extraction is an important part of moving object detection algorithms that are very useful in surveillance systems. Moving object detection algorithm will be simple by background subtraction when a clean background image is available. The method of extraction the background during training sequence and updating it during the input frame sequence is called background modeling. The main challenges in moving object detection is extraction a clean background and its updating.

The simplest background modeling is to use a single unimodal distribution [1,2]. This method cannot model multiple backgrounds. Therefore, a more complex model was proposed that is Mixture of Gaussian (MOG) [3]. MOG has some disadvantages. MOG could not model fast background variations with few of Gaussians. In addition, there is a trade-off between modeling and detection in MOG. If the learning rate of MOG is low, it cannot detect sudden changes as background. High learning rates cause to quick adaptation, but foreground that move very slow, absorb to background. To eliminate disadvantages of MOG, a non-parametric method was proposed that known as Kernel [4]. Kernel method uses kernel density method to estimate probability density function of each pixel. The main disadvantage of Kernel method is that need to high memory capacity for long time period backgrounds.

Kim et al [5] introduced layered codebook (CB) to model multi background, moving background and scene illumination variations in a compact data structure. Their proposed algorithm is a color model. Basic codebook model is very fast and usually more efficient in memory relative to MOG and Kernel [5]. They shown that basic CB model using less computational complexity and less memory capacity has better result in foreground-background segmentation with respect to MOG and Kernel. To improve the basic CB model, Kim et al proposed layered modeling and adaptive CB updating in new structure called as adaptive layered CB. Adaptive layered CB can model new background during input sequence and it is robust against illumination variations. However, they claimed that adaptive layered CB is robust against illumination variations, but it cannot model very slow scene illumination variations in long time.

In proposed algorithm, a two-layer CB is constructed by sampling of scene in long time. For each pixel, the CBs contain some codewords (CW) that are a compact representation of background. Despite of adaptive layered CB proposed in [5], this method is robust against very slow scene illumination variation in long time.

II. Basic Codebook Model

Basic CB model is a pixel-based approach; therefore, a CB exists for each pixel. Each CB contains one or more CWs and the number of CWs in each CB is different with others. Indeed, each CW models a cluster of samples that construct a part of background. In background detection, if gray level of input pixel is in range of any CW, the input pixel will be known as a background pixel; else the input pixel is foreground.

A. *Initial Codebook Construction*

Suppose $X = \{I_1, I_2, ..., I_N\}$ is the training sequence for a gray level pixel and $C = \{c_1, c_2, ..., c_L\}$ is the CB containing L CWs. I_t is the input gray level pixel at frame t in training sequence. Data structure of i -th CW contains $c_i = (\breve{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i)$ as a 6-tuple vector. c_i represents the gray level bounds and temporal variables about CW. $\breve{I}_i$ and $\hat{I}_i$ are the minimum and maximum of pixel gray level respectively and f_i shows the frequency of this CW in

Manuscript received January 7, 2008. This work was supported in part by the Bina Pardaz Shargh Ltd., Mashad, Iran.

Mohamad Hoseyn Sigari was with Computer Engineering Department, Ferdowsi University of Mashad, Mashad, Iran. He is now with Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran. (E-mail: hoseyn_sigari@engineer.com).

Mahmoud Fathy is with Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran. (E-mail: mahfathy@iust.ac.ir).

training sequence. The longest interval that this CW does not occur in sequence, is defined as Maximum Negative Run Length (MNRL) and this is denoted by λ_i . p_i and q_i are the first and last time of occurrence in sequence, respectively. The algorithm of initial CB construction has been described below:

---

1. Set $L \leftarrow 0$ and $C \leftarrow \varnothing$.
2. For $t = 1$ to N repeat following operations:
   2.1. Find a c_m form C that I_t is in range $\left[\breve{I}, \hat{I}\right]$.
   2.2. If C is empty or there was not any CW matching with I_t do:
      2.2.1. $L \leftarrow L + 1$ and create new CW named as c_L .
      2.2.2. Create new CW as follow:
      $c_L \leftarrow \left(\max\{0, I_t - \alpha\}, \min\{255, I_t + \alpha\}, 1, t-1, t, t\right)$
   2.3. Otherwise update c_m as follow:
   $$c_m \leftarrow \left(\frac{I_t - \alpha + f_m.\breve{I}}{f_m + 1}, \frac{I_t + \alpha + f_m.\hat{I}}{f_m + 1}, f_m + 1,\right.$$
   $$\left.\max\{\lambda_m, t - q_m\}, p_m, t\right)$$
3. For all c_i ($i = 1,2,...,L$) change the value of λ_i to $\lambda_i \leftarrow \max\{\lambda_i, N - q_i + p_i - 1\}$.

α is a positive real parameter and the best values for α is about 10. In this algorithm, when f_i is little, c_i will represent a small cluster of few samples and gray level range ($[\breve{I}_i, \hat{I}_i]$) of CW will be updated with higher speed. But when f_i is large, c_i will represent a big cluster of many samples and gray level range ($[\breve{I}_i, \hat{I}_i]$) of CW will be updated with lower speed. Experiments show one pass training is sufficient and more training will have negligible effects on model.

Adaptive layered CB model in [5] uses min and max operations instead of averaging in part 2.3 of this algorithm to update values of $\breve{I}_i$ and $\hat{I}_i$. Adaptive layered CB replaces old value of $\breve{I}_i$ with minimum of $\breve{I}_i$ and I_t . Also, in this algorithm old value of $\hat{I}_i$ is replaced by maximum of $\hat{I}_i$ and I_t .

Very slow and large scene illumination variations causes to extend $[\breve{I}_i, \hat{I}_i]$ continuously and after a long time, very extensive gray scales will be modeled as background.

B. Codebook Refinement

After construction of initial CB, moving objects and noise are coded in some CWs of initial CB. Therefore, the initial CB is fat and it must be refined by temporal filtering. Fixed and moving backgrounds usually occur in sequence with short period, but foreground has long period. This distinction between foreground objects and background was used to refine CB. MNRL of each CW shows the long interval not occurring in training sequence. CWs that have large λ_i are background and others are foreground. So CB is refinement as follow:

$$M = \{c_m \mid c_m \in C, \lambda_m < T_M\} \quad (1)$$

M is the refined CB that models background and T_M is a threshold on λ_i to distinguish foreground objects from background. Usually T_M is set to half number of training sequence. Note that this filtering is only based on NMRL criterion. c_i will be removed if λ_i is less than T_M , although the c_i related to CW with large f_i .

C. Background Subtraction Algorithm

Background subtraction algorithm uses for an input pixel I_t to determine it as foreground/background pixel. Background subtraction algorithm is as below:

---

1. Find a c_m form CB that I_t is in range $\left[\breve{I}, \hat{I}\right]$.
2. If there was not any CW matching with I_t then $Output = \text{foreground}$.
3. Otherwise $Output = \mathit{back}\text{ground}$ and update c_m as follow:
$$c_m \leftarrow \left((1-\beta)(I_t - \alpha) + \beta.\breve{I}, (1-\beta)(I_t + \alpha) + \beta.\hat{I}, f_m + 1,\right.$$
$$\left.\max\{\lambda_m, t - q_m\}, p_m, t\right)$$

β is a positive real parameter and must be lower than 1. The best values for β is higher than 0.9. Adaptive layered CB model in [5] uses min and max operations instead of running average method in part 3 of this algorithm to update values of $\breve{I}_i$ and $\hat{I}_i$.

III. Two-Layer Codebook Model

During input sequence, a new background may appear after background training. The main disadvantage of basic CB model is that cannot model new background scene during input sequence. To overcome this problem, a two-layer CB model is proposed. So the background model contains two CB: M (main) and H (hidden or cache). M and H model permanent and non-permanent backgrounds respectively. Data structure of H is like to M .

In training phase, such as basic CB model, only main CB is constructed and cache CB is empty. During input sequence, foreground-background is segmented and layered CB model is updated. For layered CB model three threshold are defined: T_H , T_{add} and T_{delete} . These thresholds are used to refine main and cache CBs. If λ of a CW in H is more than T_H , this CW will be deleted from H . If a CW stays in H longer than a certain time (T_{add}), then it will be moved to M . If a CW of M does not appear longer than a certain time (T_{delete}), then it will be remove from M .

Two-layer background modeling algorithm is as follow:

---

1. Background training obtains M . Create an empty model H .
2. For an input pixel I_t , find a matching CW in M . If a CW was found in M , input pixel is background and update corresponding CW.
3. Otherwise:
   3.1. Input pixel is foreground.

3.2. Find a matching CW in H. If a CW was found in H, update it. If there is no matching CW, create a new CW in H and put this pixel in it.
4. Remove all CWs from H having λ more than T_H.
5. Move all CWs from H to M staying longer than T_{add} in H.
6. Remove all CWs from M not appearing longer than T_{delete}.
7. Go to step 2.

IV. Experimental Result

To evaluate our proposed method, we compared our method with adaptive layered CB model in [5] on PETS dataset [6]. PETS is a standard dataset of some sequences with moving background or illumination variant scene for performance evaluation of tracking and surveillance systems. For example, *'waving trees'* and *'water surface'* are two sequence with moving backgrounds. *'time of day'* is a sequence with large illumination variations in long time. Frame size in this dataset is 160x120.

Proposed algorithm and adaptive layered CB model were implemented in Microsoft C# .Net 2005 on an AMD Athlon 4200+ (2.2 GHz, dual core) with 1 GB RAM. Adaptive layered CB runs on PETS sequences at 27 frames per second, whiles our proposed method runs at 41 frames per second. Both algorithms are real-time, but our proposed method is more 51% faster than adaptive layered CB.

Regarding memory usage, usually our proposed two-layer codebook model and adaptive layered codebook model need equal number of codeword. Each CW in proposed two-layer codebook model has 4 integer (f_i , λ_i , p_i , q_i) and 2 float point ($\breve{I}_i$, $\hat{I}_i$) variables, but each CW in adaptive layered codebook model has 4 integer (f_i , λ_i , p_i , q_i) and 5 float point ($\breve{I}_i$, $\hat{I}_i$, $\overline{R}_i$, $\overline{G}_i$, $\overline{B}_i$) variables. So, if each integer and float point variable are 4 bytes, CWs in adaptive layered codebook model and proposed two-layer codebook model will need 24 bytes and 36 bytes respectively. Therefore, proposed two-layer codebook is 50% more efficient than adaptive layered codebook model in memory usage.

Table I shows comparison between adaptive layered codebook model and proposed two-layer codebook model in background subtraction processing speed and memory usage.

Table I: Comparison between adaptive layered codebook model and proposed two-layer codebook model.

Method	Mode	Processing Speed (frames per second)	CW size (bytes)
Adaptive layered codebook	RGB Color	27	36
Proposed two-layer codebook	Gray level	41	24

Experiments show that proposed algorithm and adaptive layered CB model have very similar results in sequences with moving background. However, adaptive layered CB is a color model and our proposed method is a gray level model. Fig. 1 shows a frame of sequence *'waving trees'*. Fig. 2 and Fig. 3 show result of background subtraction using adaptive layered CB and our proposed method respectively.

Adaptive layered CB model has very weak result on *'time of day'* sequence. Adaptive layered CB model extends range of $[\breve{I}_i, \hat{I}_i]$ sequentially during input sequence, because of using min and max operations. Therefore, it cannot detect moving objects properly after very slow and large illumination changes in long time. Our proposed method updates and moves the range $[\breve{I}_i, \hat{I}_i]$ using running average instead of extending it. Therefore, after large and very slow illumination variations in long time, it can detect moving objects. Fig. 4 shows a frame of sequence *'time of day'*. Fig. 5 and Fig. 6 show result of background subtraction using adaptive layered CB and our proposed method respectively.

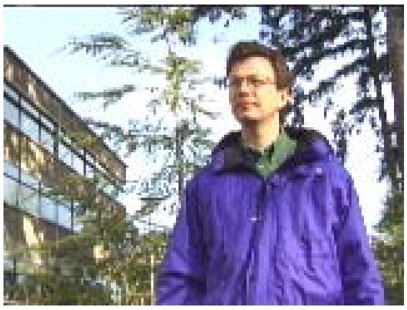

Fig. 1. A frame of *'waving trees'* sequence.

Fig. 2. Result of background subtraction on *'waving trees'* using adaptive layered codebook model

Fig. 3. Result of background subtraction on *'waving trees'* using proposed two-layer codebook model

Fig. 4. A frame of *'time of day'* sequence.

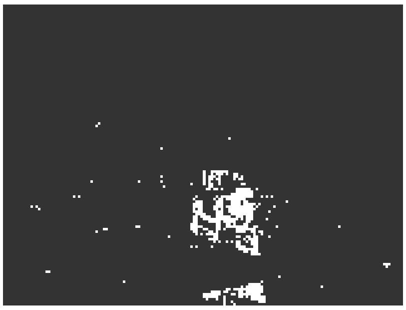

Fig. 5. Result of background subtraction on *'time of day'* using adaptive layered codebook model

Fig. 6. Result of background subtraction on *'time of day'* using proposed two-layer codebook model

V. Conclusions

In this paper a two-layer CB model was proposed to model multi background, moving background and illumination variations. Our proposed method models permanent and non-permanent backgrounds in two CB. Multi background and moving background are coded in some CWs. CWs are updated continuously to overcome scene illumination changes. Proposed algorithm is efficient in both memory and computational complexity.

In experiments, our two-layer CB model was compared with adaptive layered CB model that proposed in [5]. Both methods have very good results on sequences with moving background. But, adaptive layered CB model is weak to detect moving objects in sequences with large and very slow illumination variations in long time. Contrary, our proposed two-layer codebook model is very efficient to segment foreground-background.

Both algorithms are real-time, however our proposed method is more 51% faster than adaptive layered CB, because of using gray level model instead of color model. In addition, our proposed method is 50% more efficient than adaptive layered codebook in memory usage.

Some problems may be occurred in using of adaptive layered CB model in real time applications. Usually a mere background modeling and subtraction is not useful. In many applications, other image processing algorithms are required to extract useful information form background or foreground objects. The speed of adaptive layered CB model on an AMD Athlon 4200+ (2.2 GHz dual core) is about 27 frames per second. Therefore, there will not enough time for other processing if the processing deadline is 40 milli seconds per frame (at 25 frames per second). Our proposed method segments background-foreground in less than 25 milli seconds and there is more than 15 milli seconds for other processing on image (at 25 frames per second).

References

[1] C.R. Wren, A. Azarbayejani, T. Darrell, A. Pentland. "Pfinder: realtime tracking of the human body". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 780–785. 1997.

[2] T. Horprasert, D. Harwood, L.S. Davis. "A statistical approach for real-time robust background subtraction and shadow detection". IEEE Frame-Rate Applications Workshop. Kerkyra, Greece. 1999.

[3] C. Stauffer C, W.E.L. Grimson. "Adaptive background mixture models for real-time tracking". IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 246–252. 1999.

[4] A. Elgammal, D. Harwood, L.S. Davis. "Non-parametric model for background subtraction". European Conference on Computer Vision, Vol. 2, pp. 751–767. 2000.

[5] K. Kim, T.H. Chalidabhongse, D. Harwood, L. Davis. "Real-time foreground-background segmentation using codebook model". Elsevier Real-Time Imaging, Vol. 11, pp. 172-185. 2005.

[6] PETS dataset. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. 2001. Available at http://www.visualsurveillance.org/PETS2001.