Moving Objects Detection Based on Hysteresis Thresholding*

Hsiang-Erh Lai[1], Chih-Yang Lin[2,**], Ming-Kai Chen[1], Li-Wei Kang[3], and Chia-Hung Yeh[1]

[1] Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan
B983012011@student.nsysu.edu.tw, e123642261@gmail.com
yeh@mail.ee.nsysu.edu.tw
[2] Dept. of Computer Science & Information Engineering, Asia University, Taichung, Taiwan
andrewlin@asia.edu.tw
[3] Institute of Information Science, Academia Sinica, Taipei, Taiwan
lwkang@iis.sinica.edu.tw

Abstract. Background modeling is the core of event detection in surveillance systems. The traditional Gaussian mixture model has some defects when encountering some situations like shadow interferences, lighting changes, and other problems causing foreground image broken. All of these cases will result in deficiencies of event detection. In this paper, we propose a new background modeling method to solve these problems. The model features of our method are the combination of texture and color characteristics, hysteresis thresholding, and the motion estimation to recover broken foreground objects.

Keywords: background modeling, moving objects detection, hysteresis thresholding.

1 Introduction

In these days, people are more and more concerned about the importance of environment security. Not only working space but also residences are commonly equipped with surveillance systems. In addition, foreground detection is also a milestone of the surveillance system. By this way, we can save lots of time of focusing on the monitor. In fact, background modeling is used to distinguish foreground and background. Thus, a robust background modeling method is needed when detecting moving objects.

There are a number of methods for moving objects detection but most of them are based on color information. For instance, a statistical approach based on color information [1] built a background model and reduce the shadow interference. In addition, Wren et al. [2] proposed a one-Gaussian method to strengthen the background flexibility. However, one-Gaussian method has a defect for a dynamic background, such

* This work was supported in part by the National Science Council, Taiwan, under Grants NSC101-2221-E-468-021.

** Corresponding author.

J.-S. Pan et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 21, pp. 289–298.
DOI: 10.1007/978-3-642-35473-1_29

as swaying trees, ripples, and the blink of a screen. Thus, Stauffer and Grimson proposed the Gaussian mixture model (GMM) method [3]. For each pixel, GMM used more than one Gaussian to model the background [3, 4, 5]. If the pixel does not match the GMM model, it will be regarded as a foreground pixel. There is an example of GMM method for traffic monitoring [6] and others can be found in [7] and [8].

Though GMM improves Wren et al.'s method a lot, it still suffers from shadow interference and illumination changes. Therefore, Heikkilä and Pietikäinen proposed the texture-based background model with local binary patterns (LBPs) [9, 10]. This method has the tolerance to illumination changes. However, LBPs are not robust. When noises or swaying trees strike the central pixel value, the corresponding LBPs histogram would be interfered and increase the possibilities of false positive and false negative cases.

In this paper, we proposed a new background modeling method. Our main contributions are as follows: (1) Our proposed method is based on hysteresis thresholding. Hysteresis thresholding has never been used for background modeling and it can greatly alleviate the cavity problem in foreground objects. (2) We proposed a new texture descriptor derived from our previous work [11]. With this descriptor, we can enhance the tolerance to illumination changes and shadow interference. (3) Our method combines the information of both texture and color to reduce shadow problems while improving the shapes of foreground objects. (4) The motion estimation technique is also applied in our method to recover broken foreground objects caused by motion problems, such as moving too slow or walking toward the camera.

The following parts of this paper are organized as follows: in Section 2, we take a brief review on the Gaussian mixture model. Our proposed method is presented in Section 3. The results of our experiments are discussed in Section 4. Finally, conclusions are provided in Section 5.

2 Preparation Work

In this section, we will explain how a mixture of Gaussians model works. This method was first proposed by Grimson and Stauffer [1, 2]. They model each background pixel into a K-Gaussians mixture model (GMM), where K is between 3 and 5. The weight of each Gaussian distribution represents the portion of the data accounted for that Gaussian.

First, each pixel is modeled by a mixture of K Gaussian distributions. The probability of observing the current pixel value is:

$$P(X_t) = \sum_{j=1}^{K} \omega_{j,t} * \eta(X_t, \mu_{j,t}, \sum{}_{j,t}), \tag{1}$$

where X_t is the current pixel value at time t, K is the number of Gaussian distributions, $\omega_{j,t}$ is the weight estimation of the j^{th} Gaussian in the mixture at time t, $\mu_{j,t}$ and $\sum_{j,t}$ are the mean value and covariance matrix respectively, of the j^{th} Gaussian in the mixture at time t, and η is a Gaussian probability density function (*pdf*).

After the model is built, each incoming pixel of following frames is compared with the existing model components. In the case that the input pixel fits one of the weighted Gaussian distributions, it means that its pixel value is within 2.5 standard deviations of the matched distribution. Once the pixel is matched, the update process will be invoked to fine-tune the corresponding model; otherwise, we will replace the distribution, which has the lowest weight, with a new distribution using the current incoming pixel as its mean value, an initial high variance, and a low prior weight.

In order to select the best Gaussians for each pixel, the K distributions are sorted based upon the value ω/σ. Only the first B distributions are selected as the background model of a pixel for the scene and denoted as:

$$B = \arg\min_b \left(\sum_{k=1}^{b} \omega_k > T_B\right), \tag{2}$$

where T_B is a predefined threshold and usually set to about 90%, ω_k is the weight parameter of the k^{th} model component and b indicates the number of background distributions.

At last, the update process will change the weights of K Gaussian distributions as follows:

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}), \tag{3}$$

where α is the learning rate and $M_{k,t}$ is 1 for the matched distribution and 0 for the unmatched distributions. In addition, weights of distributions should be renormalized. If the new pixel matches a Gaussian distribution, the values of mean and variance of this distribution are updated as follows:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t, \tag{4}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t), \tag{5}$$

where

$$\rho = \alpha\eta(X_t \mid \mu_k, \sigma_k). \tag{6}$$

3 Proposed Method

In this section, we describe the proposed method with texture descriptor, texture-based background modeling, hysteresis thresholding, and motion estimation for foreground recovery.

3.1 Texture Descriptor

In the beginning, we divide the input frame into several non-overlapping blocks with a size of $n \times n$ pixels. For each block, mean value m of the block is calculated by:

$$m = \frac{1}{n \times n} \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}, \tag{7}$$

where x_{ij} indicates the pixel value in the position (i, j) of the block.

With the mean value, we can build a binary map (BM) for the block by the following equation.

$$b_{ij} = \begin{cases} 1, & \text{if } x_{ij} \geq m, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where bit "1" denotes that the pixel value is greater than the mean value m of that block; Otherwise, the bit is "0".

Here is an example for the texture descriptor. In this case, the block is set to 3×3 as shown in Fig. 1. In Fig.1, the mean of this block is 51.56 and the bitmap is built accordingly.

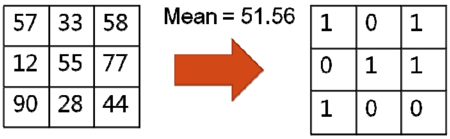


Fig. 1. The process of building a binary bitmap

3.2 Texture-Based Background Modeling

In the previous section, each block has been transformed into a binary bitmap. However, if there is a smooth block, where the pixels are either barely larger or barely smaller than the block mean, the corresponding bitmap would result in an interlaced 0/1 pattern. This situation gives rise to an unstable background model since smooth blocks and non-smooth blocks cannot be distinguished. To solve this problem, Eq.(8) is changed to Eq.(9) with a threshold TH_{smooth} to solve this problem, and TH_{smooth} is set as 8 according to our experiments.

$$b_{ij} = \begin{cases} 0, & \text{if } x_{ij} < m + TH_{smooth}, \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

Note that in the process of the bitmap generation, the input frame captured by a camera is transformed into a grayscale image by Eq.(10)in order to improve the efficiency. Fig. 2 illustrates the result of the bitmap generation, which shows the validity of the proposed texture descriptor.

$$\text{Gray} = 0.299\text{R} + 0.587\text{G} + 0.114\text{B} \tag{10}$$

(a) Original image (b) Texture description

Fig. 2. The proposed texture descriptor

The proposed background modeling is a pixel-based model. Therefore, each pixel has its own texture description, *i.e.*, the corresponding BM. The background model based on the proposed texture descriptor consists of K weighted bitmaps, $\{BM_1, BM_2, \ldots, BM_K\}$, where each weight is between 0 and 1, and the summation of the weights is 1. The weight of the k^{th} bitmap is denoted as w_k. When a new BM_{new} is captured, it is compared with the K bitmaps by the following Hamming distance equation, where m is in the range of $[1, K]$:

$$Dist(BM_{new}, BM_m) = \sum_{i=1}^{n}\sum_{j=1}^{n}(b_{ij}^{new} \oplus b_{ij}^{m}) \tag{11}$$

If $\min_m Dist(BM_{new}, BM_m)$ is smaller than a threshold predefined, the BM_{new} matches the background model, and then the update process will be invoked; otherwise, BM_{new} is regarded as a foreground pattern, and the unmatched process will be applied. The process of the model maintenance can be referred to our previous work [11].

3.3 Hysteresis Thresholding

In the traditional GMM, whether a pixel is a background pixel or not is determined by a single threshold TH(*i.e.*, 2.5 standard deviations of a Gaussian) as follows:

$$\text{input} = \begin{cases} \text{foreground, if } \min Dist \geq TH, \\ \text{background, otherwise.} \end{cases} \tag{12}$$

This approach brings about a serious problem. If TH is too small, the output image will contain lots of noises; on the contrary, when TH is too large, the foreground objects may contain a lot of cavities.

In this paper, hysteresis thresholding is proposed to solve this problem, where double thresholds TH_{high} and TH_{low} are used to enhance the foreground estimation. The threshold TH_{high} is responsible for generating "strong" information and TH_{low} is responsible for gathering "weak" information. The strong information means these generated foreground pixels are very robust but may result in breaks or cavities in the foreground objects. On the other hand, the weak information will generate more complete shapes of foreground objects but involve more noise as well.

We apply hysteresis thresholding on both of the original color GMM and the proposed texture-based background modeling to generate four binary maps, called strong color, weak color, strong texture, and weak texture maps. The pixels in the strong texture map are called real foreground pixels and the pixels in the remaining maps are called pseudo foreground pixels. The foreground objects generation starts from the strong texture map, and traces weak texture, strong color, and weak color maps to gradually compensate or mend the shapes of the foreground objects. More specifically, if a pixel belongs to one of the strong color, weak color, and weak texture maps, and the pixel is connected with a strong texture pixel, then the pixel is identified as a real foreground pixel and will be treated as a new strong texture pixel in the next iteration. The process will continue until all the pseudo foreground pixels have been tested.

The combination of the color-based GMM and the proposed texture-based model has higher tolerance to shadow interference and illumination changes. In addition, using hysteresis thresholding can fix the cavities problem caused in the strong texture map and get more complete shapes of foreground objects. The noise in the proposed scheme can be nearly removed because noise is usually not connected with strong maps. Fig. 3 shows an example of the four maps.

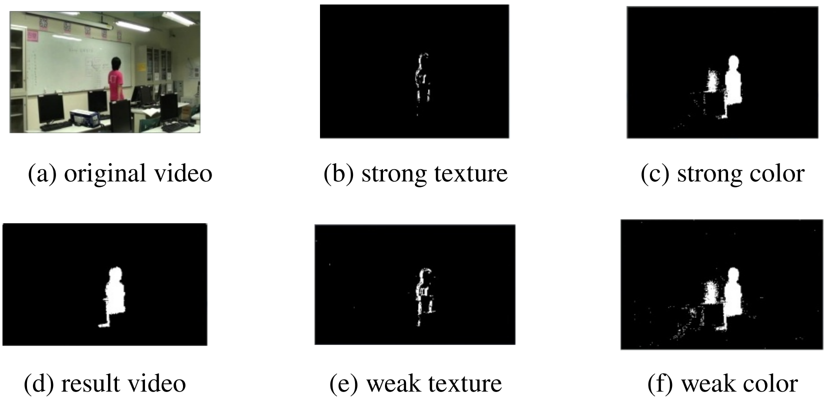


Fig. 3. The results of four maps based on hysteresis thresholding

3.4 Motion Estimation

In addition to the above four maps discussed in the previous section, in this section, another binary map, called motion map, is generated based on motion estimation. When there is an object that moves too slowly or moves toward the camera, some pixels of this object will be gradually becoming background due to the effects of Eqs. (3) to (6), resulting in the foreground object fractured. With the help of the motion map, this problem can be greatly alleviated.

The motion map is generated as follows. Set the binary image of the i^{th} frame using the four maps mentioned in Section 3.3 be denoted as R_i, and set P_{i-1} be the result after applying the four maps and the motion map of the $(i-1)^{th}$ frame. The motion map for the i^{th} frame is the difference between R_i and P_{i-1}, denoted as D_i. To mend cavities in R_i, each pixel in R_i will find its 5×5 neighbors. If some of these 5×5 neighbors belong to D_i, these pixels will be included in the foreground objects. The flow chart of generating the motion map is shown in Fig. 4 and an example of applying the motion map is presented in Fig. 5.

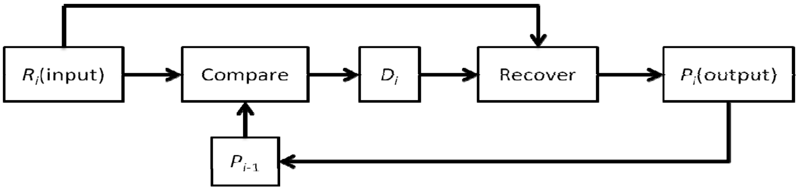
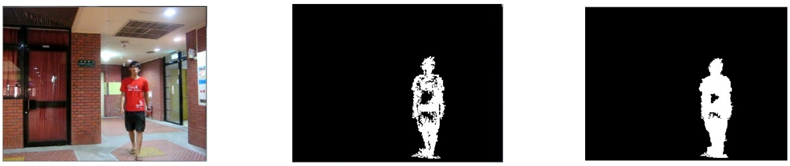


Fig. 4. Flow chart of motion map



(a) Original video (b) Result video (R_i) (c) Final result video(P_i)

Fig. 5. Final result of using the motion map

4 Experimental Results

4.1 Detecting Results

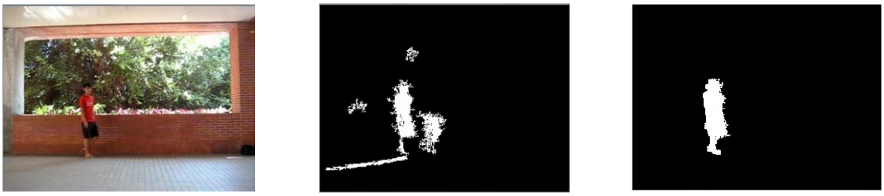
The following are our experimental results. Figs. 6 and 7 show the indoor and outdoor scenes with shadow interference. The results show that the proposed method has higher tolerance to the shadow interference than the original GMM.

Fig. 8 shows an outdoor scene that a person walks toward the camera. In this video, our proposed method shows the repair ability of broken image, which cannot be achieved in the traditional GMM.

Fig. 9 shows a person walking toward an indoor camera. This figure clearly reveals that the proposed method not only removes the shadow but also successfully mends most of the cavities in the body.

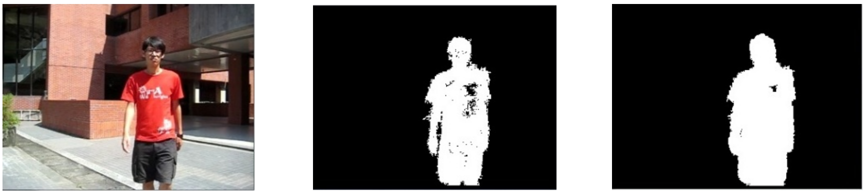
(a) Original image (b) GMM method (c) Proposed method

Fig. 6. Experimental results of indoor video 1



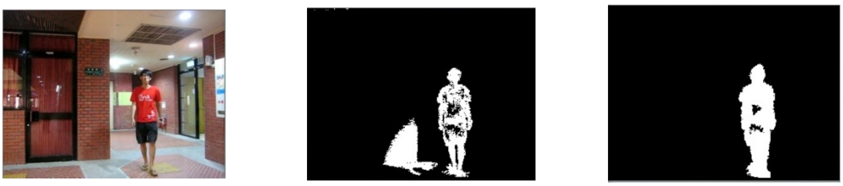
(a) Original image (b) GMM method (c) Proposed method

Fig. 7. Experimental results of outdoor video 1



(a) Original image (b) GMM method (c) Proposed method

Fig. 8. Experimental results of outdoor video 2



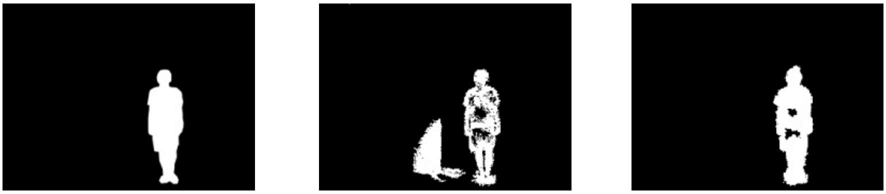
(a) Original image (b) GMM method (c) Proposed method

Fig. 9. Experimental results of indoor video 2

4.2 Quantitative Results

We compare our method with the traditional GMM in a quantitative way. The simulation environment is equipped with a 2.93 GHz Core 2 Duo Intel processor and 4 GB of memory. All algorithms were implemented in C++. The parameters, α and K, used in the experiments are set to 0.005 and 3, respectively.

Fig. 10 shows the accuracy comparison of the proposed method, GMM method and the ground truth. There are three items in this evaluation: False positive (FP) is the number of background pixels which are mistaken for foreground; False negative is the number of foreground pixels which are mistaken for background; Total Error (TE) is the sum of FP and FN.



(a) Ground truth (b) GMM method (c) Proposed method

Fig. 10. Comparison on indoor video

From Table 1, it clearly reveals that the proposed method has much lower FP than that in the GMM due to the shadow removing, and FN in the proposed scheme is greatly reduced for the reason of our mending technique.

Table 1. Accuracy comparison

Items	GMM	Proposed method
FP	2376	556
FN	749	238
TE	3125	794

5 Conclusions

In this paper, we proposed a new background modeling method based on hysteresis thresholding. The proposed method has the following advantages: (1) tolerance to shadow interference and illumination change due to the texture characteristic; (2) resistance to noise and shape fracturing because of hysteresis thresolding; (3) repairing the foreground objects with the help of the mostion estimation technique. The expirement results show that FP and FN of the proposed method are much better than those in the original GMM method. Our future work will improve our efficency for real time surveillance applications.

References

1. Wixson, L.: Detecting Salient Motion by Accumulating Directionally-Consistent Flow. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 774–780 (2000)
2. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: Real-Time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 780–785 (1997)
3. Stauffer, C., Grimson, W.E.L.: Adaptive Background Mixture Models for Real-Time Tracking. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA, pp. 246–252 (1999)
4. Stauffer, C., Grimson, W.E.L.: Learning Patterns of Activity Using Real-Time Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 747–757 (2000)
5. Zivkovic, Z.: Improved Adaptive Gaussian Mixture Model for Background Subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, pp. 28–31 (2004)
6. Friedman, N., Russell, S.: Image Segmentation in Video Sequences: A Probabilistic Approach. In: Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence, San Francisco, pp. 175–181 (1997)
7. Dedeoğlu, Y., Töreyin, B.U., Güdükbay, U., Çetin, A.E.: Silhouette-Based Method for Object Classification and Human Action Recognition in Video. In: Huang, T.S., Sebe, N., Lew, M., Pavlović, V., Kölsch, M., Galata, A., Kisačanin, B. (eds.) ECCV 2006 Workshop on HCI. LNCS, vol. 3979, pp. 64–77. Springer, Heidelberg (2006)
8. Power, O.W.: Understanding Background Mixture Models for Foreground Segmentation. In: Proceedings of Image and Vision Computing, Auckland, New Zealand, pp. 267–271 (2002)
9. Heikkilä, M., Pietikäinen, M.: A Texture-Based Method for Modeling the Background and Detecting Moving Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4), 657–662 (2006)
10. Heikkilä, M., Pietikäinen, M., Heikkilä, J.: A Texture-Based Method for Detecting Moving Objects. In: Proceedings of British Machine Vision Conference, British, pp. 187–196 (2004)
11. Lin, C.Y., Chang, C.C., Chang, W.W., Chen, M.H., Kang, L.W.: Real-Time Robust Background Modeling Based on Joint Color and Texture Descriptions. In: Proceedings of The Fourth International Conference on Genetic and Evolutionary Computing (ICGEC 2010), Shenzhen, China, December 13-15, pp. 622–625 (2010)
12. Fisher, R., Santos-Victor, J., Crowley, J.: EC Funded CAVIAR project/IST 2001 37540, `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`
13. Lumia, R., Shapiro, L., Zuniga, O.: A New Connected Components Algorithm for Virtual Memory Computers. Computer Vision, Graphics, and Image Processing 22(2), 287–300 (1983)
14. Chang, F., Chen, C.J., Lu, C.J.: A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique. Computer Vision and Image Understanding 93(2), 206–220 (2004)