Bidirectional Background Modeling for Video Surveillance

Chih-Yang Lin, Wei-Wen Chang, and Yung-Chen Chou

Abstract—Traditional background model methods often require complicated computations, and are sensitive to illumination and shadow. In this paper, we propose a block-based background modeling method, and use our proposed method to combine color and texture characteristics. Suppression and relaxation are the two key strategies to resist illumination changes and shadow disturbance. The proposed method is quite efficient and is capable of resisting illumination changes. Experimental results show that our method is suitable for real-world scenes and real-time applications.

Index Terms—Background modeling, Gaussian mixture modeling, motion detection.

1. Introduction

Object detection is a very important task in video surveillance. In general, background modeling is utilized for distinguishing between foreground and background. With a robust background model, the objects can then be successfully extracted from the background.

In literatures, a number of methods for detecting moving objects have been proposed. Many different features are employed for background modeling. The most frequently used features are based on color information. For example, a color statistical approach[1] accomplished background subtraction without being affected by shadow; furthermore, the algorithm was also implemented by a DM270 iMX subsystem[2] for digital video (DV) applications. In the statistic model, an one-Gaussian adaptive modeling method is a popular approach that can be found in [3]. However, the one-Gaussian model cannot perfect interpretation of the dynamic background model. Therefore, the Gaussian mixture model (GMM) method[4]–[6] was presented after the one-Gaussian model. This method is based on each pixel using more than the one-Gaussian model to maintain the background structure. GMM can model the background in a more detail way. When a new pixel comes in, if this pixel value cannot meet the requirements of the corresponding background models, it will belong to the foreground area. In addition, GMM involves an adaptive scheme to update the current background model. One of the examples using GMM was developed (three Gaussian) for traffic monitoring[7]. Other discussions on implementation using GMM can be found in [8] and [9], which provide details of Stauffer and Grimson’s algorithm[6]. However, although the Gaussian mixture model has adequate ability to record the changing of scenes, the main disadvantage of GMM is the highly computational cost due to the pixel-based operations. The inefficient computations would cause a heavy burden for real-time video surveillance systems.

In addition to GMM, some background modeling methods for moving objects detection are motion-based or edge-based methods. The motion-based method[1] uses optical flow to detect salient motion, and the edge-based method[10] is based on edge information, such as the edge histogram. However, these methods usually have complicated computations.

In 2006, Heikkilä and Pietikäinen[11], [12] proposed a texture-based method to construct the background model. This method is using local binary patterns (LBPs). LBPs can reduce computations and have the property of tolerance for illumination changes. However, the central pixel value in LBPs method is easily affected by noise or swinging trees, due to the unstable histogram when LBP is applied. This would increase the possibilities of false positive and false negative cases.

In this paper, we propose a bidirectional background modeling method based on the concepts of suppression and relaxation. In addition, color and texture information will be involved in the background modeling process. Instead of using LBPs, we propose a new texture descriptor, the idea of which is inspired by block truncation coding (BTC)[13], to enhance the tolerance of illumination changes and shadow interference. With the help of suppression and relaxation, the proposed method can greatly reduce the possibilities of false positive and false negative cases. In addition, the proposed method is a block-based method, so the performance would be much more efficient than that of the conventional GMM, which adopts the pixel-based
2. Related Work

2.1 Mixture of Gaussian Models

Here, we will present how a mixture of Gaussians model works, which was proposed by Stauffer and Grimson\cite{4,6}. The authors proposed a new method to model every background pixel into a $K$ GMM. Typically, $K$ is a small number ranged between 3 and 5. The weight parameters represent the portion of the data accounted for by Gaussians.

The probable background colors should stay longer and more invariant than others. In other words, its Gaussian distributions have the most supporting evidence and the least variance. This model was widely used in a real-time mode accompanied by an update process. When a new pixel comes in, it is checked against existing model components. The new pixel is said to match one of the weighted Gaussian distributions if its pixel value is within 2.5 standard deviations of the matched distribution. If any of the models is matched, the matched distribution will be updated. Otherwise, (i.e., none of the $K$ Gaussian distributions can be matched to the current pixel value), the distribution that has the minimum weight is replaced with a distribution using the current value as its mean value, an initially high variance, and a low prior weight.

Each pixel in the scene is modeled by a mixture of $K$ Gaussian distributions. The probability of observing the current pixel value is

$$P(X_t) = \sum_k^K w_k \eta(X_t, \mu_k, \Sigma_k)$$

where $K$ is the number of Gaussian distributions, $w_k$ is the weight estimation of the $k$th Gaussian in the mixture at time $t$, $\mu_k$ and $\Sigma_k$ are the mean value and covariance matrix respectively, of the $k$th Gaussian in the mixture at time $t$, and $\eta$ is a Gaussian probability density function (PDF) defined in (2):

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)}.$$  

(2)

For computation efficiency, $\Sigma_k$ is defined as $\sigma_k^2 I$ to represent the covariance of the $k$th model component.

The $K$ distributions are sorted based upon the value $\omega_k = \alpha$\text{Var} of the current pixel and denoted as $B = \arg\min_b \left( \sum_{k=1}^b w_k > T \right)$

where $T$ is a predefined threshold that representing the minimum quantity of the data that must be accounted for the background model. $T$ is usually set to be about 90% in many applications.

After the incoming pixel has been determined whether it will match to the existing Gaussian distributions, the prior weights of $K$ Gaussian distributions are changed as follows:

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha$$

where $\alpha$ is the learning rate and $M_k,t$ is 1 for the matched distribution and 0 for the unmatched distribution. Subsequently, weights of distributions are renormalized. If the new pixel matches to a Gaussian distribution, the values of mean and variance of this distribution are renormalized and 0 for the unmatched distribution.

$$\mu_{k,t} = (1-\rho)\mu_{k,t-1} + \rho X_t$$

$$\sigma_{k,t}^2 = (1-\rho)\sigma_{k,t-1}^2 + \rho(X_t - \mu_{k,t})^T(X_t - \mu_{k,t})$$

(6)

$$\rho = \alpha \eta(X_t | \mu_k, \sigma_k).$$

(7)

2.2 Heikkilä and Pietikäinen’s Method

The texture-based method proposed by Heikkilä and Pietikäinen\cite{11,12} first partitions each image frame into overlapping blocks so that the extracted shape of the moving object can be more accurately described. Then, the pixels in each block produce a histogram according to their LBP values. For example in Fig. 1, if the current pixel value is 6, and the surrounding pixel values are 5, 9, 3, 1, when the surrounding pixel value is bigger than the current pixel, this surrounding pixel is marked as 1, otherwise is marked as 0. Then, the marks of the surrounding pixels transform into a binary value in accordance with the sequence. As the example shown in Fig. 1 (b), the binary value is 0100, and after transformation, the decimal value is 2, which is called the LBP value of the current pixel.

The background modeling is supported by histograms of each block. A histogram is composed of LBP values in a block. For a block, it can contain many histograms, which are coming from the consecutive frames. Then, these histograms are modeled by $K$ weighted histograms for the

![Fig. 1. Example of generating LBP: (a) neighbors of the central pixel and (b) binary pattern.](image)
purpose of multi-model backgrounds. When a new block histogram comes in, the histogram compares with the $K$ weighted histograms. If the new block histogram is similar to the background model, the new block will be classified as background and performs background updating; otherwise, it is recognized as a foreground block. The update process is similar to Grimson’s method\(^6\). In the updating process, only $B(\leq K)$ histograms are selected as the background model.

3. New Background Modeling Method

In this section, we describe the proposed texture descriptor, texture-based background modeling, and finally the proposed motion detection method based on joint color and texture descriptors.

The main defect of the traditional background modeling approach is the heavy computations because of the pixel-based approach. In order to speed up the performance, we use the block-based method to build our background model. Although the block-based method can improve the efficiency problem in pixel-based methods, it may produce many broken regions or hollows on the foreground area. In order to solve this problem, in this paper, the concepts of suppression and relaxation are proposed. In addition, color-based and texture-based background models are combined to get more accurate foreground areas. The following describes these details.

In order to reduce the complexity, when the camera receives the frame, the color image is transformed to the gray image, and the gray image is partitioned into non-overlapping blocks with a size of $n$ pixels$\times n$ pixels. For each block, we must calculate the corresponding mean value $m$, which is defined as follows:

$$m = \frac{1}{n \times n} \sum_{i=1}^{n} \sum_{j=1}^{n} x_{i, j} \tag{8}$$

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

3.1 Color-Based Background Modeling

The traditional Gaussian model is based on pixel values, but that may be quite inefficient. Therefore, we use each block mean $m$ to implement the GMM model. The details of this part can refer to \[4\] and \[6\].

3.2 Texture-Based Background Modeling

In this approach, each pixel $x_{i, j}$ in a block is transformed into a binary bit according to the mean value of the block. The transformed block is a binary mask (BM), and the transformation process is defined as follows:

$$b_{i, j} = \begin{cases} 1, & \text{if } x_{i, j} > m \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $b_{i, j}$ means the bit in the position $(i, j)$ of a BM.

However, we may encounter a problem in such comparisons. The problem is that when two pixel values in a block are similar but not the same, if (9) is applied and the block mean value is between these two pixel values, the two pixels would be separated into different groups. This result is not reasonable and it results in unstable BM especially for a smooth block. In order to solve this problem, we modify (9) by adding a threshold $TH_{\text{smooth}}$ as shown in (10). In general, $TH$ is set 8 from our observations.

$$b_{i, j} = \begin{cases} 0, & \text{if } x_{i, j} < m + TH_{\text{smooth}} \\ 1, & \text{otherwise}. \end{cases} \tag{10}$$

We now consider how to use the feature vector BM to construct the background model. The background model for each block consists of $K$ weighted bitmaps, $\{BM_1, BM_2, \ldots, BM_K\}$, where each weight is between 0 and 1, and the $K$ weights have a sum of 1. The weight of the $k$th bitmap is denoted as $w_k$. When a new block $BM_{\text{new}}$ comes in, $BM_{\text{new}}$ is compared with the $K$ bitmaps by the following similarity equation, where $m$ is in the range of $[1, K]$:

$$\text{Sim}(BM_{\text{new}} \cdot BM_k) = \sum_{i=1}^{n} \sum_{j=1}^{n} (h_{\text{new}}^{ij} \cap h_{k}^{ij}). \tag{11}$$

If the maximum similarity $\text{Sim}(BM_{\text{new}}, BM_k)$ is greater than a predefined threshold $TH_{\text{dist}}$, the block $BM_{\text{new}}$ matches $BM_k$ in the background model, and the update process will be invoked; otherwise, $BM_{\text{new}}$ is regarded as a foreground block, and the unmatched process will be launched. The complexity of the above distance calculation is quite low since it requires only bit operations.

The update and unmatched processes are derived from Stauffer and Grimson’s method\(^6\). When an incoming block is considered as a background block, the weights of the background model are updated by:

$$w'_k = \alpha M_k + (1 - \alpha) w_k \tag{12}$$

where $\alpha$ is the learning rate and $M_k$ is 1 for the best-matched bitmap and 0 for the others.

The learning rate determines the speed of adaptation. That is, larger learning rates result in faster adaptions. As for each bit $h_{ij}^{\text{new}}$ of the best-matched bitmap $BM_k$, the update rules are given in (13) and (14). In (13), when $t$ is greater than a predefined threshold $T$, $t$ should be set to $T$ to meet the self-adaptation requirement.

$$p_{i, j}^{w} = \frac{1}{t} p_{i, j}^{w} + \frac{1}{t} b_{i, j}^{\text{new}} \tag{13}$$

where $t$ represents the $r$th frame and $p_{i, j}^{w} = 0$ in the initial stage.
\[ b_{ij}^{m'} = \begin{cases} 0, & \text{if } p_{ij}^{m'} \leq 0.5, \\ 1, & \text{otherwise.} \end{cases} \] (14)

If the incoming block is a foreground block, the unmatched process replaces the bitmap that has the lowest weight in the background model with the incoming block. Then, the weight of the new block is set to a low initial weight of 0.01 in our experiments. Finally, the weights of the background model are renormalized in order to have a sum of one.

In the above description, the incoming block may match the bitmap with a low weight and is regarded as a background block. However, the low weight means that the corresponding bitmap has a low probability of being a background block. To solve this problem, the weights of the background model are sorted in a decreasing order, and only the first \( B \) bitmaps are selected as the background model, such that:

\[ \sum_{i=1}^{B} w_i > \text{TH}_B \] (15)

where \( \text{TH}_B \) is a predefined threshold.

### 3.3 Suppression & Relaxation

In this subsection, we will describe how the concepts of suppression and relaxation can be applied to construct a better background model. The suppression means that the block will be regarded as a foreground block only when the probability of foreground is high. On the contrary, the relaxation means that more foreground blocks would be generated since the threshold is relaxed to a lower level.

In this paper, each block will be maintained by four background models, called strong color model, weak color model, strong texture model, and weak texture model, respectively. The first two models are maintained by GMM, and the remaining two models are maintained by our proposed texture modeling. In the conventional GMM, the threshold of 2.5 standard deviations is applied to judge whether the incoming pixel belongs to background or not. However, for the strong color model, the threshold for standard deviations is reduced to 2. On the contrary, for the weak color model, the threshold for standard deviations is increased to 3. Similarly, two thresholds \( \text{TH}_{s\_\text{suppress}} \) and \( \text{TH}_{r\_\text{relax}} \) should be set for a strong texture model and a weak texture model, respectively, where \( \text{TH}_{s\_\text{suppress}} > \text{TH}_{r\_\text{relax}} \). The two thresholds can be applied to (11) to obtain the corresponding foreground images.

After the four background models are constructed, when a new frame comes in, we can get two motion detection result images, called foreground images, for the two color models, respectively. Then, a difference image can be generated by subtracting the strong color foreground image from the weak color foreground image. Similarly, two texture-based models can also generate a texture difference image. Finally, we combine color difference image, texture difference image, and strong texture image to find the most appropriate foreground region. The framework is shown in Fig. 2.

### 4. Experimental Results

Our propose method is constructed on Windows 7 and the simulated environment for the experiments is equipped with a 3.1 GHz Core i5-2400 Intel processor and 16 GB of memory. The algorithms were implemented in C++. The image resolution in the experiments was set to 320 pixels×240 pixels.

For the sake of labeling and segmenting the foreground pixels, the connected components algorithm\(^{[14]}\) is applied to each background modeling method. The parameters used in the experiments are listed in Table 1, where \( \alpha \) is the learning rate, \( \text{TH}_B \) is used in (15), \( K \) denotes the number of Gaussians, \( X \) represents the fact that the parameter is not required in this method, and BS means the block size.

The performance comparisons of these two methods are presented in Table 2, where the last row denotes the connected components labeling (CCL) method that is also involved in the background construction. In Table 2, we can find our proposed method is much faster than the original GMM, because our proposed method is based on the non-overlapping block that the operator is less than the original GMM.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>( \alpha )</th>
<th>( \text{TH}_B )</th>
<th>( K )</th>
<th>BS</th>
<th>( \text{TH}_{\text{mean}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stauffer and Grimson’s method</td>
<td>0.005</td>
<td>0.9</td>
<td>3</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Proposed method</td>
<td>0.005</td>
<td>0.9</td>
<td>3</td>
<td>8</td>
<td>0.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Stauffer and Grimson’s method</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame rate</td>
<td>12.7</td>
<td>27.8</td>
</tr>
<tr>
<td>Frame rate with CCL</td>
<td>11.1</td>
<td>19.9</td>
</tr>
</tbody>
</table>
Fig. 3 shows the indoor scenes with some illumination changes, where people were walking towards the camera. The detection results can be observed that Stauffer and Grimson’s method is very sensitive to illumination changes. On the contrary, the proposed method has a better ability to resist illumination changes because we simultaneously consider the color and texture information, and combine multiple foreground data to repair the foreground objects. Since the non-overlapping block approach is adopted, the contour of the proposed method is coarser than those of the compared methods. More experimental results of the outdoor scenes compared with Stauffer and Grimson’s method are shown in Fig. 4 and Fig. 5. Although the proposed method may result in more holes of the inner areas of the moving object, we can use the morphology operations to solve this problem.

5. Conclusions

In this paper, we proposed a joint color- and texture-based method for background modeling, and combined suppression and relaxation concepts to find out more accurate foreground objects. The suppression and relaxation can be regarded as a bidirectional approach, where the suppression is at the top and relaxation is at the bottom, and the best result is between them. The proposed method not only can resist illumination changes but also requires low computations. The experimental results show that our method is quite efficient. Since the proposed method possesses a very high frame rate, it is quite suitable for real-time applications or low-power computation systems such as cell phones and personal digital assistants (PDAs).

References


Chih-Yang Lin received his Ph.D. degree in computer science and information engineering from National Chung-Cheng University, Chiayi in 2006. After graduated, he joined in Advanced Technology Center of Industrial Technology Research Institute of Taiwan (ITRI) from 2007 to 2009. Then, he joined the Institute of Information Science (IIS), Academia Sinica, as a postdoctoral fellow. Currently, he is an assistant professor with the Department of Computer Science and Information Engineering, Asia University, Taichung. Professor Lin’s research interests are in areas of computer vision, digital rights management, image processing, and data mining. He has published over 45 papers and several patents in these fields, and got two best paper awards in PCM 2008 (Pacific-Rim Conference on Multimedia) and CVGIP 2009 (Computer Vision, Graphics and Image Processing), respectively. Recently, he devotes himself to the design of surveillance systems and privacy-preserving related topics.

Wei-Wen Chang received the B.S. degree in computer science and information engineering from Nanhua University, Chiayi in 2011. Now he is pursuing his M.S. degree with the Department of Computer Science and Information Engineering, Asia University, Taichung. His current research interests include image processing, computer vision, and video processing.

Yung-Chen Chou received the B.S. degree in management information systems from National National Pingtung University of Science & Technology, Pingtung in 1998, the M.S. degree in information management from Chao Yang University of Technology in 2002, and the Ph.D degree in computer science and information engineering from the National Chung Cheng University, Chiayi in 2008. Since February 2009, he has been an assistant professor with Asia University, Taichung. His current research interests include steganography, watermarking, and image processing.