Abstract

This paper presents a statistical adaptive real-time background subtraction algorithm that is very robust to moving shadows and dynamic scene environment. The algorithm enhances the previously developed method reported in [4] by adding adaptation of modeling correspond to dynamic background using adaptive brightness and color distortion. In addition, we propose a novel “vivacity factor” to measure activities of foreground objects. It is used to delay the adaptation rate for the area of often-occurred moving foregrounds. Our method provides a solution to real-time moving object and shadow detection in dynamic background scene from video stream. We also develop the learning-rate control mechanism that was not addressed by most background subtraction algorithms.

1. Introduction

In many years of successes in object detection, segmentation and tracking, background subtraction technique seems to be the major solution for automated video surveillance applications. The advantages of low complexity and high precision bear the technique working in high speed and satisfactory accuracy. Typically, algorithms start by building reference background parameters. This process can be done using statistical or deterministic approaches and is known as background model synthesizing. Background modeling process is considered to be one of the most important parts in the system. Model that is the closest approximation to “real-background” gives the best reference for subtraction process. Unfortunately, background in real-world scenario can be changed over time in various ways. The major problems in background modeling include [1,2,3,4]:

- Illumination changes in scene, that drastically increases the deviation of color vector in background model, causes classification process works erroneously.
- Change in location of background objects usually alters the model at pixels in the region of original and new locations.
- Movement of foreground objects brings their shadows move and causes system to classify them as foreground layer.
- Waving motion in background scene (e.g., swaying trees, refreshing monitor) may generate periodic characteristic in input frames.

The first two problems can be solved by real-time adaptation of background as addressed in [1,2,3,5]. The instant matching of background model in Mixture of Gaussain Models moderately resolves the fourth problem [6]. Object’s shadow is also an important issue in background subtraction. Greffenhagen et al. [7] proposed the normalized color to get rid of illumination component; while Jabri et al. [8] used edge information to eliminate the influence of illumination. Our proposed algorithm is most closely related to the work in [4], which has shown very efficiently computed, accurate, and robust to illumination changes such as shadows and highlight [9,10]. Unluckily, this algorithm cannot cope with changes in background scene. We take distinctive advantages of their method and add capability of adaptation into it. This makes our real-time algorithm is very robust to both illumination changes and moving scene problems.

2. Building Background Model

Modeling of background plays a vital role in background subtraction algorithms. The accuracy of the background model directly affects the effectiveness of the detection. In our framework, we perform pixel-based background modeling in two steps. First, we construct an initial background model from a stationary background scene. Then, while the subtraction is being performed, the background model is updated by the proposed “on-line modeling” method. The details are described below.
2.1 Initial Background Model

We consider color value from real-time camera in RGB color space. A color vector at pixel \((i,j)\) of the \(n\)th frame is depicted as in eq. (1).

\[
X_{i,j}[n] = (X_{i,j}^R[n], X_{i,j}^G[n], X_{i,j}^B[n])
\]  

(1)

where \(X_{i,j}^R\), \(X_{i,j}^G\), \(X_{i,j}^B\) are red, green, blue color intensity at pixel \((i,j)\). This process is a stationary background modeling process in which we collect \(N\) frames of “empty” scene. So we obtain \(N\) color vectors for each pixel. The sophisticated and reduction of sensitive detection in fast dynamic variation scene [10] of Mixture of Gaussian Models [MOG] lead us back to consider background model as Single Gaussian. Naturally, we obtain two significant parameters automatically. The first one is “Expected Color Vector”, as in eq. (2),

\[
E_{i,j} = E\{X_{i,j}[n]\} : 1 \leq n \leq N
\]  

(2)

where \(E_{i,j}\) is expectation operation. So, \(E_{i,j}(E_{i,j}^R, E_{i,j}^G, E_{i,j}^B)\) represents mean of color vectors at pixel \((i,j)\) over \(N\) frames. The latter is “Color Covariance Matrix”. The covariance matrix, \(C_{i,j}\) is assumed to be diagonal to reduce computational cost, and can be written as in eq. (3).

\[
C_{i,j} = \text{I}((\alpha_{i,j}^R) \times (\alpha_{i,j}^G) \times (\alpha_{i,j}^B))^T
\]  

(3)

Next, we compute the distortion of \(X_{i,j}[n]\) from its mean, \(E_{i,j}\), by considering two orthogonal distortion parameters, “Brightness Distortion” \((\alpha_{i,j}[n]\)) and “Color Distortion” \((\lambda_{i,j}[n]\)).

Brightness distortion implies the brightness intensity of input color vector, \(X_{i,j}[n]\), respect to the expected color vector, \(E_{i,j}\), and can be obtained as in eq. (4).

\[
\alpha_{i,j}[n] = \arg\min_{\psi} \min((X_{i,j}[n] - \psi E_{i,j})^2)
\]  

(4)

On the other hand, color distortion is defined as the orthogonal distance between input color vector and the reference expected color vector, and is given in eq. (5).

\[
\lambda_{i,j}[n] = \|X_{i,j}[n] - \alpha_{i,j}[n] E_{i,j}\|
\]  

(5)

As shown in [4], there are variations of \(\alpha_{i,j}[n]\) and \(\lambda_{i,j}[n]\) ; and their values are different for different pixels. Thus, to optimize the detection process, we compute two variation parameters: one represents the variation of brightness distortion \((a_{i,j})\) and another one represents the variation of color distortion \((b_{i,j})\), as defined respectively in eqs. (6) and (7).

\[
a_{i,j} = \text{RMS}(\alpha_{i,j}[n]) = \sqrt{\frac{\sum_{n=1}^{N} (\alpha_{i,j}[n] - 1)^2}{N}}
\]  

(6)

\[
b_{i,j} = \text{RMS}(\lambda_{i,j}[n]) = \sqrt{\frac{\sum_{n=1}^{N} (\lambda_{i,j}[n])^2}{N}}
\]  

(7)

Then, the initial background model is represented by a “four-tuple” statistical parameters \(\{E_{i,j}, C_{i,j}, a_{i,j}, b_{i,j}\}\) for each pixel \((i,j)\).

This typical stationary background modeling approach has been shown that it yields an outstanding effective performance in terms of both quantitative performance, as shown by detection rate and false alarm rate; and qualitative performance, defined in terms of robustness to noise, flexibility to shadow and computational load [9]. However, the approach cannot cope with the problem of dynamic background scene. The erroneous classification might occur in cases of change of global scene illumination and movement of background objects. Adaptive background modeling is a solution for this problem.

2.2 On-line Background Model

To adapt to changes in the dynamic scene, we update the background model continuously while performing the subtraction. We use the initial model as a seed of adaptation \((n=0)\). The “on-line background model” \(\{E_{i,j}[n], C_{i,j}[n], a_{i,j}[n], b_{i,j}[n]\}\) is given, as in eqs. (8)-(11),

\[
E_{i,j}[n] = (1 - \gamma) E_{i,j}[n-1] + \gamma X_{i,j}[n]
\]  

(8)

\[
C_{i,j}[n] = (1 - \gamma) C_{i,j}[n-1] + \gamma (X_{i,j}[n] - E_{i,j}[n])(X_{i,j}[n] - E_{i,j}[n])^T
\]  

(9)

\[
a_{i,j}[n] = \sqrt{(1 - \gamma) a^2_{i,j}[n-1] + \gamma (\alpha_{i,j}[n]-1)^2}
\]  

(10)

\[
b_{i,j}[n] = \sqrt{(1 - \gamma) b^2_{i,j}[n-1] + \gamma (\lambda_{i,j}[n])^2}
\]  

(11)

, where parameter \(\gamma\) can be interpreted as a “learning rate of adaptation”. Thus, \(1/\gamma\) effectively defines the time constant: implies speed of the model change or update.

3. On-line Vivacity Factor (\(\nu\))

The learning rate of adaptation \((\gamma)\) mentioned in the previous section indicates speed of background model adaptation. If the value of \(\gamma\) is large, the effect of the relocation of background objects (such as moving chair in the office scene) will be updated quickly. At the same time, the true background model might be rapidly lost in
the area that has high frequency of moving foreground objects appearance as well as in the case of moving foreground objects become stationary for a period of time. As a result, we have to reduce speed of adaptation at pixels that represent high activity of foreground objects.

First, we define “vivacity” as a value that determines the activity of foreground objects in terms of temporal change of color vector at each pixel. Consider N frames of 24-bits RGB video sequence, Vivacity of each pixel (i,j) at frame n can be represented by “Vivacity Factor” (υ) which is defined as in eq. (12).

\[ υ_{i,j}[n] = \frac{\sum_{j} \sum_{n} (X_{i,j}[n] - X_{i,j}[n-1])^2}{255^2 N} \]  
(12)

In case of “On-line Vivacity Factor”, this factor will be updated overtime as in eq. (13),

\[ υ_{i,j}[n] = (1-\zeta) (υ_{i,j}[n-1])^2 + \zeta (X_{i,j}[n] - X_{i,j}[n-1])^2 \]  
(13)

where ζ is vivacity learning rate that maintains vivacity factor for the low-textural moving objects in the scene. We initialize υ_{i,j}[1] = 0. Now, the speed of adaptation or leaning rate γ in equations (8) - (11) must be substituted by factor (1 - υ_{i,j}[n]) as given in eq. (14).

\[ γ_{i,j}'[n] = (1 - υ_{i,j}[n]) \gamma \]  
(14)

In additions, to prevent the false updating (e.g., slow-moving or low-textural objects are added into the background model too quickly), we introduce update frame interval (I_U) parameter. Instead of update the model every frame, we update every I_U^{th} frame and the learning rate of adaptation is set to I_U \gamma. With this, the real moving objects will be updated at the rate of γ while the new deposited background objects (that stay still longer than I_U frames) will be updated at the rate of I_U \gamma.

### 4. Online Subtraction and Classification

This section describes real-time subtraction process and pixel classification. We start by initializing the online background model by the background model (its seed). For each input nth frame, we compute α_{i,j}[n] and λ_{i,j}[n] using Eqs. (4) and (5), and normalize them by on-line background parameters as in eqs. (15) and (16).

\[ \hat{α}_{i,j}[n] = \frac{\hat{α}_{i,j}[n]-1}{\hat{α}_{i,j}[n]} \]  
(15)

\[ \hat{λ}_{i,j}[n] = \frac{\hat{λ}_{i,j}[n]}{b_{i,j}[n]} \]  
(16)

Then, pixel mask M_{i,j}(n) can be classified into 4 classes: B: Background, F: Foreground, S: Shadow, H: Highlight by these conditions, as in eq. (17).

\[
M_{i,j}(n) = \begin{cases} 
F: & \hat{α}_{i,j}[n] > \tau_δ \text{ or } \hat{α}_{i,j}[n] < \tau_{obo}, \\
B: & \hat{α}_{i,j}[n] < \tau_{a1} \text{ and } \hat{α}_{i,j}[n] > \tau_{a2}, \\
S: & \alpha_{i,j}[n] < 0, \text{ else} \\
H: & \text{otherwise}
\end{cases}
\]  
(17)

Where τ_δ, τ_{a1} and τ_{a2} are computed given detection error-rate (r). τ_{obo} is user defined threshold to limit degree of shadow in case of dark objects. Classification model is shown in Fig. 1.

![Figure 1 Background and classification model.](image)

### 5. Experimental Results

In this section, we demonstrate the performance of the proposed algorithm on image sequences of two different scenarios shown in Figs. 2 and 3, where (a) is the input image, (b) is the image of online background model, (c) is the pixel masks result from non adaptive algorithm in [4], (d) is the pixel masks result from our algorithm, (e) is the final segmentation result from non adaptive algorithm[4], and (f) is the final segmentation result of our algorithm. The colors of pixel masks in images (c) and (d) are denoted as follows: F = cyan, B = original color, S = red, H = green (see eq. 17).

The first sequence is a video of a person moving in a room; at the middle of the sequence, half of the fluorescence lamps that illuminate the room were turned off. This causes global illumination changed condition. The result in Figure 2 demonstrates that our algorithm can adapt to the change quickly and be able to detect the target successfully. The second sequence is another video of moving person in a room; initially, the person came in, placed a box on the table, and left the room. Then the person came back with a handbag and put the handbag in front of the box with respect to the camera view. This is a
challenge problem. Since the box was placed and stationary for a period of time, it should be included as a new background objects. Most algorithms such as [4] will detect both box and handbag as foregrounds (Fig. 3(e)). However, our algorithm can learn the new background object and adapt the model to include it. The segmentation result shown in Fig. 3(f) shows only the person and the handbag as foregrounds.

6. Conclusions

In this paper, we propose a statistical adaptive real-time background subtraction algorithm. The proposed algorithm adds capability of background modeling adaptation using adaptive brightness and color distortion into original background subtraction algorithm proposed in [4]. The algorithm proposed also measure activities of foreground object to adaptively control the learning rate of the model. Experimental results indicate that our real-time algorithm is very robust to both illumination changes and moving scene problems.

Acknowledgements

The authors are very grateful to Kyungnam Kim for providing benchmark video sequences and Cooperative Project between Department of Electrical Engineering and Private Sector for Research and Development, Year 2003, for in part supporting this research.

References


Figure 2 Test the algorithm on a sequence of global illumination change with these parameters: $r = 0.002$, $a_0 = 0.5$, $\gamma = 0.1$, $\zeta = 0.05$, and $I_U = 5$.

Figure 3 Test the algorithm on a sequence of moving or new background object with these parameters: $r = 0.001$, $a_0 = 0.4$, $\gamma = 0.05$, $\zeta = 0.01$, and $I_U = 8$. 