International Journal of Innovative Computing, Information and Control
Volume **9**, Number **4**, April **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 1373–1388

LEFT-OBJECT DETECTION THROUGH BACKGROUND MODELING

Chih-Yang Lin[1], Chi-Shiang Chan[2], Li-Wei Kang[3] and Kahlil Muchtar[1]

[1]Department of Computer Science and Information Engineering
[2]Department of Information Science and Applications
Asia University
No. 500, Lioufeng Road, Wufeng, Taichung 41354, Taiwan
{ andrewlin; CSChan }@asia.edu.tw

[3]Institute of Information Science
Academia Sinica
No. 128, Academia Road, Sec. 2, Nankang, Taipei 115, Taiwan
lwkang@iis.sinica.edu.tw

Received December 2011; revised April 2012

Abstract. *Video surveillance systems are becoming extensively deployed in many environments due to the increasing needs of public security and crime prevention. In this paper, we propose a comprehensive solution for managing abandoned objects, which means that the system can deal with objects that are abandoned, removed, or partially occluded. The system contains two adaptive abandoned object detection (AOD) methods that are both based on the proposed texture modeling method associated with a mixture of Gaussians for a real environment. The first method is more efficient than the second one, but the latter is more robust than the former. The proposed methods have been proved to be characterized with prominent efficiency and robustness according to mathematic analyses and experimental results. The designed automatic detection system helps human operators not only to ease tedious monitoring work but also to focus only on suspicious abnormal events.*

Keywords: Object detection, Background subtraction, Video surveillance

1. **Introduction.** Recent years have seen an increase in the number of severe terrorist attacks on public places such as airports, subways, train stations, town centers, shopping malls and financial institutions. The increased need for public security and the need to prevent these attacks have initiated the wide scale deployment of surveillance tools. However, conventional surveillance systems controlled are labor-intensive since many cameras are involved, and these need to be continuously monitored and controlled by human operators. An automatic visual surveillance system is, therefore, urgently needed to provide continuous and proactive prevention and detection for public security. Thanks to the advances in digital camera technology, the design of automatic visual surveillance systems has become feasible, and can be applied to many applications [3].

The problem of detecting abandoned objects (also referred to as static, left, or immobile objects in this paper) is currently one of the most intensive research topics for public security and surveillance services. An object is abandoned [2] if it is static and unattended at the scene, and it was not there earlier; in other words, an object that was carried by a person initially becomes abandoned if the owner left without taking the object away and the object was unattended for a period of time. One of the challenges of solving this problem is that long-time detection is required, so lighting changes must be considered. In addition, the crowds walking near the object may increase the difficulty of detection,

and the removed object, which means the left object is taken away later, should also be taken into account in order to maintain the correctness of detection results.

Some algorithms have been proposed to solve the abandoned object problem. These methods can be categorized into two approaches: one is based on tracking methodology [1,2,8,11,13], and the other is based on the detection approach [14,18,19]. In the tracking-based methods, Auvinet et al. [1] detected spatio-temporal forks to deal with the merge and split activities when two objects meet or separate. Then, the static object is characterized by a foreground blob that remains constant over time. Beynon et al. [2] applied the Kalman filter to track foreground objects and used a Bayesian classifier to search for candidate static objects. The candidate static objects are then verified by a finite state machine. A similar concept to the finite state machine was also applied to Li et al.'s method [11]. Guler and Farrow [8] focused on drop-off events detection to obtain the candidate static objects. The abandoned objects are extracted according to a stationary object confidence image, in which each pixel value indicates the confidence representing whether the pixel belongs to a static object. Lv et al. [13] employed blob tracking and Bayesian inference to verify abandoned objects. Each blob contains information related to size, location, and a histogram of the blob. These tracking-based methods encounter the problems of merging, splitting, entering, corresponding, leaving and occlusion. These problems are not easy to solve in many cases.

On the other hand, in detection-based methods, moving objects in the scene can be neglected, and only immobile objects that were not initially present should be of concern. The main advantage of the detection-based methods is that they do not need to handle the complicated problems associated with the tracking-based methods, such as merging, splitting, and corresponding, since finding the owners of abandoned objects is not our ultimate purpose. Wang and Ooi [19] subtracted the current frame from the background image to obtain foreground objects. Then these objects are compared with the objects extracted from previous frames to identify abandoned objects. This method results in generating too many candidate static objects and is time-consuming due to the matching process. Furthermore, a counter is required for each foreground object to determine the number of frames in which the object has not been moved. Stringa and Regazzoni [18] further improved the temporal occlusion problem (i.e., people crossing between the camera and the static object) that hampered Wang and Ooi's method. Stringa and Regazzoni utilized two simultaneous differences for each pixel to find static objects. The first difference D_1 is between the current frame and the background image, and the second difference D_2 is between the current frame and the previous frame. Then, a shift register is constructed, as shown in Table 1. If the number of couples $(1,0)$ is greater than a threshold, the pixel is recognized as part of a static object. However, Stringa and Regazzoni's method is not robust to slowly-removed objects and is sensitive to noise. Martinez-del-Rincon et al. [14] adopted Jaraba et al.'s method [10] that uses three simultaneous differences for each pixel to find static objects. In their method, short-term and long-term background images should be prepared in advance. Let I_k be the current image, I_{k-1} be the previous image, S be the short-term background image, and L be the long-term background image. For each incoming frame, the operation $\tilde{}(I_k - I_{k-1}) \cap (I_k - S)$ is performed to find the candidate static objects, where $\tilde{}(I_k - I_{k-1})$ means the complement of the difference image $(I_k - I_{k-1})$. The candidate static objects are then accumulated to update S and are verified by the difference image $(S - L)$. Although Martinez-del-Rincon et al.'s method does not have to build shift registers, it still encounters several problems. First, it requires a counter for each pixel of the foreground objects. This fact increases the probability of false detection.

TABLE 1. The content of shift registers

Time	t_i	t_{i+1}	t_{i+2}	t_{i+3}	t_{i+4}	t_{i+5}	...	t_j
D_1	1	1	0	1	1	1	...	1
D_2	1	0	0	0	0	1	...	0

Second, the subtraction results of this method are very sensitive to noise and unstable. In addition, the above methods apply simple background construction methods such as time averaging [5], which may not be suitable for real environments. A similar idea, using long-term and short-term backgrounds, can be found in [15]. These methods also scarcely focus on the problem of removed objects. Therefore, if an object is left for a period of time and then taken away, the above methods will consider there are two abandoned objects appearing and thus cause inconsistency between real results and detection results.

In this paper, we propose an automatic management system for abandoned objects detection (AOD). The main contributions of this paper are stated as follows. First, we propose a comprehensive solution to judge the status of objects, including abandoned, removed, or partially occluded objects. Second, we apply a new background modeling, combining texture and color features [12] based on mixture of Gaussians (GMM) [16,17], to adapt to the changes in the real environment. Such a background modeling method is more robust than previous methods. Third, since GMM has been widely used in the modern surveillance system, the proposed AOD scheme is also based on the GMM model, which can detect abandoned objects without extra computations. Fourth, in the traditional AOD systems, they always pay attention to abandoned objects, but for removed objects, which are originally still but removed later, are seldom discussed. In this paper, the case of removed objects will be sufficiently addressed. Finally, since there always has to be a trade-off between efficiency and robustness, the proposed scheme will provide two options (i.e., two adaptive detection methods). In the experiments, different degrees of complexity of test cases will be considered to verify the system's performance. The remainder of this paper is organized as follows. First, we briefly review the Gaussian mixture model in Section 2. Then, in Section 3, we elaborate on the proposed background construction and AOD. The issues related to the design of the management system for abandoned objects are also explored. Our experimental results are presented and discussed in Section 4. Finally, the paper ends with conclusions in Section 5.

2. **Gaussian Mixture Model.** Stauffer and Grimson [17] proposed the Gaussian mixture model (GMM) to adapt the background in a real environment. The main idea is to use more than one Gaussian distribution to describe the statistics of each pixel, since a pixel may change its value over time due to lighting changes. Gao et al. [6] have proved that the mixture of Gaussians performed better than the single Gaussian and is practical for the real environment and long-term monitoring systems. Assume that a histogram of a pixel p in a sequence of video frames is defined as $\{x_1, x_2, \ldots, x_t\}$, where x_i is the intensity value of p at time instant i. The GMM utilizes multiple (usually 3-5) Gaussian distributions to model the histogram of a pixel to observe the current pixel value x_t, which is defined as

$$P(x_t) = \sum_{i=1}^{k} w_{i,t} \times \eta(x_t, \mu_{i,t}, \Sigma_{i,t}), \tag{1}$$

where k is the number of distributions, $w_{i,t}$ is the weight of the i^{th} Gaussian distribution to describe the portion of the data resolved by this distribution, $\mu_{i,t}$ and $\Sigma_{i,t}$ are, respectively, the mean value and covariance matrix of the i^{th} Gaussian distribution, and η is a Gaussian probability density function, which is defined as

$$\eta(x_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} \left|\Sigma\right|^{1/2}} e^{-\frac{1}{2}(x_t-\mu_t)^T \Sigma^{-1}(x_t-\mu_t)} \tag{2}$$

When a new sequence of frames comes in, the parameters of the GMM should be updated. A 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, and then this distribution is updated by

$$\mu_{i,t} = (1-\rho)\mu_{i,t-1} + \rho x_t, \tag{3}$$

$$\sigma_{i,t}^2 = (1-\rho)\sigma_{i,t-1}^2 + \rho(x_t - \mu_{i,t})^T (x_t - \mu_{i,t}), \tag{4}$$

$$\rho = \alpha\eta(x_t|\mu_{i,k}, \sigma_{i,k}), \tag{5}$$

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha M_{k,t}, \tag{6}$$

where α is the learning rate, and $M_{k,t}$ is 1 for the matched model and 0 for the unmatched remaining models.

If none of the k distributions matches the current pixel, the distribution in GMM with the smallest weight is replaced with a new distribution, where its mean value is set to the value of the current pixel, the variance is set to an initial high variance, and the weight is set to a low prior weight.

In the GMM background model, only B out of k distributions are selected to describe the background. The choice of B distributions is according to the factor w/σ in non-increasing order; that is, the distribution with a larger weight and a smaller standard deviation has a higher priority to be selected. The number of B indicating the minimum portion of the data that should be considered by the background is decided by the following equation, where T is a user-defined threshold.

$$B = \arg\min_b \left(\sum_{k=1}^{b} w_k > T \right) \tag{7}$$

When a pixel comes in, it is considered to be a background pixel if it is matched with one of the B distributions; otherwise, it is recognized as a foreground pixel.

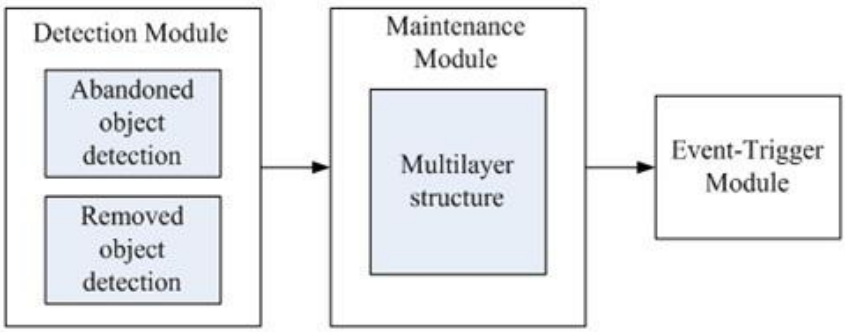

FIGURE 1. The block diagram of the proposed system

3. **The Proposed Method.** The proposed system is comprised of three major modules, i.e., the detection module, the maintenance module, and the event-trigger module. The block diagram is shown in Figure 1. With the detection module, both the cases of left objects and removed objects can be detected; then, the maintenance module updates the statuses of static objects in the scene. Finally, the event-trigger module reports the interesting events that have occurred. Since the event-trigger module is straightforward according to the results from the maintenance module, we only focus on the former two modules in this paper.

3.1. **Background construction using color and texture information.** Since the proposed left object detection is based on the background construction method, the quality of the background model determines the robustness of left object detection. In the following, we describe how to extract the texture information and then combine with the color information as our background model. When a camera captures an image, the frame is first divided into non-overlapping blocks with a size of $n \times n$ pixels. For each block, the mean value m is calculated and defined as follows:

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

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

The output of each image block is a binary bitmap BM with a size equal to the block. The bitmap is generated by Equation (9), where bit "1" in a BM denotes that the corresponding pixel value of the block is greater than m; otherwise, the bit is set to 0. Finally, the set of BM's is called the texture descriptor for an input frame.

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

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

- Texture-based Background Modeling

Initially, each input frame is divided into non-overlapping blocks, and each block is transformed into a bitmap according to the above texture descriptor. Note that since the pixels in a smooth block are sensitive to their mean value, the corresponding texture description may not be robust. In order to solve this problem, we change the bitmap generation equation slightly from Equation (9) to Equation (10). The value TH_{smooth} is usually set to 8 according to our experimental results.

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

In addition, the captured image in a real surveillance system is a color image, so each block should have three masks, one for each of the red, green and blue channels. For convenience, one block is represented by only one mask. It is straightforward to extend the idea to three masks for each block. Figure 2 shows the image generated by the proposed texture descriptor. The fact that the texture of the image in Figure 2 is clearly presented proves the validity of this descriptor.

We now consider how to use the feature vector 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_{new} with size $n \times n$ comes in, BM_{new} is compared with the K bitmaps by the following similarity equation, where m is

(a) Original image (b) Texture description

FIGURE 2. The proposed texture descriptor

in the range of $[1, K]$:

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

If $\max_m Sim(BM_{new}, BM_m)$ is greater than a predefined threshold, the block BM_{new} is considered to match BM_m in the background model, and the update process will be invoked; otherwise, BM_{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 only bit operations are required.

The weight w_i update process is similar to Stauffer and Grimson's method [17]. As to each bit b_{ij}^m of the best-matched bitmap BM_m, the update rules are given in Equations (12) and (13). In Equation (12), when t is greater than a predefined constant $T_{f,t}$ should be set to T_f to meet the self-adaptation requirement.

$$p_{ij}^{m'} = \left(1 - \frac{1}{t}\right) p_{ij}^{m} + \frac{1}{t} b_{ij}^{new}, \tag{12}$$

where t represents the t_{th} frame and $p_{ij}^m = 0$ in the initial stage.

$$b_{ij}^{m'} = \begin{cases} 0, & \text{if } p_{ij}^{m'} \leq 0.5, \\ 1, & \text{otherwise.} \end{cases} \tag{13}$$

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

- Joint Color and Texture Background Model

In this paper, the background model is called a texture model if the above modeling approach is used, while the color model used in this paper comes from Stauffer and Grimson's algorithm [17]. For convenience, the two types of models, color and texture models, are denoted as CM and TM, respectively. If only the texture model is used for left object detection, a false negative may occur when the texture description of the left object is similar to that of the background. This situation also occurs when only the color model is applied. Such a defective case can be greatly alleviated with the help of combining the two models.

3.2. **Abandoned object detection.** In this section, the two modes of AOD are described, with Mode I having a higher efficiency and Mode II being more robust.

3.2.1. *Mode I AOD.* In this mode, each pixel is characterized by a mixture of Gaussians model (GMM) with k Gaussian distributions that are sorted by w/σ into non-increasing order. When a static object occurs, each pixel of the static region would change the order of Gaussians (i.e., the distribution with the smallest weight being replaced). The repeated occurrences of the static region help to increase the weight and decrease the standard deviation of the matched distribution. However, the increasing rate of the weight is usually faster than the decreasing rate of the standard deviation in the initial runs. In other words, when the weight w_{new} of the new distribution is larger than that w of the original first Gaussian distribution (i.e., the distribution with largest w/σ), the rank of this distribution may be still unchanged (i.e., $w/\sigma > w_{new}/\sigma_{new}$) due to the fact that $\sigma < \sigma_{new}$. The proof is described as follows. Assume that the learning rate of Equation (4) for updating the standard deviation is the same as that of Equation (6) for updating the weight. This assumption is reasonable because the two learning rates are quite similar and usually set to equal in the practical system. The increasing rate of the weight of the new distribution is

$$\begin{aligned} &(w'_{new} - w_{new})/w_{new} \\ &= [(1-\alpha)w_{new} + \alpha - w_{new}]/w_{new} \\ &= (\alpha/w_{new}) - \alpha. \end{aligned} \tag{14}$$

If x denotes the incoming pixel value and μ is the corresponding mean value, the decreasing rate of the variance of the new distribution can be defined as

$$\begin{aligned} &(\sigma^2_{new} - \sigma'^2_{new})/\sigma^2_{new} \\ &= [\sigma^2_{new} - ((1-\alpha)\sigma^2_{new} + \alpha(x-\mu)^2)]/\sigma^2_{new} \\ &= [\alpha\sigma^2_{new} - \alpha(x-\mu)^2]/\sigma^2_{new}. \end{aligned} \tag{15}$$

Therefore, the decreasing rate of standard deviation will be

$$\begin{aligned} &([\alpha\sigma^2_{new} - \alpha(x-\mu)^2]/\sigma^2_{new})^{1/2} \\ &\le (\alpha\sigma^2_{new}/\sigma^2_{new})^{1/2} = \alpha^{1/2}. \end{aligned} \tag{16}$$

Now, compare the rate of Equation (14) with that of Equation (16). In the initial stage of the occurrence of the static object, the weight w_{new} in Equation (14) is quite small and approximates to α (usually set lower than 0.05), so Equation (14) can be further simplified to $(1-\alpha)$. Therefore, comparing $(1-\alpha)$ with $\alpha^{1/2}$, the relationship clearly concludes that for the new distribution, the increasing rate of weight is actually faster than the decreasing rate of the standard deviation before the Gaussians become stable.

According to the above description, the object detection scheme can be designed as follows.

Mode I AOD Algorithm

Input: A pixel of the incoming frame.

Output: Determine whether the input pixel belongs to an abandoned object.

Step 1: Extract (μ_1, w_1, σ_1) from the 1st and (μ_2, w_2, σ_2) from the 2nd Gaussian distributions of the input pixel from CM according to the order of w/σ.

Step 2: If $w_2 > w_1$, $(\mu_1 - \mu_2)^2 > TH_1$, $\sigma_1 < TH_2$ and $\sigma_2 < TH_2$, the pixel is classified as a candidate pixel of an abandoned object. Note that at this time, $\sigma_1 < \sigma_2$.

Step 3: Get the corresponding bitmap BM of the input pixel. If BM is recognized as an unmatched block, and the weight of BM is finally increased to the largest one in TM after several runs, the pixel is classified as a candidate pixel of an abandoned object.

Step 4: If the frequency of the appearance of the candidate pixel is above P times, the pixel is considered to be a pixel of an abandoned object.

In the above algorithm, TH_1 and TH_2 are predefined thresholds, and the constraints $\sigma_1 < TH_2$, and $\sigma_2 < TH_2$ are used to avoid noise interference. In this algorithm, Step 3 can be optional if false negative cases concerning texture seldom occur.

As a matter of fact, the Mode I AOD method is quite efficient since in general, GMM is the most basic component in many surveillance systems. In addition, TM is a lightweight computation model and enhances the robustness of GMM [12]. Therefore, Mode I AOD can be implemented with very little effort to fit the existed surveillance systems.

However, this method still cannot well resist severe lighting changes or irregular camera vibration, so it may produce counterfeit candidate abandoned objects. Thus, we proposed an alternative approach (Mode II, described in the next subsection) to further decrease the probability of false detection.

3.2.2. *Mode II AOD.* Instead of using only one GMM model in Mode I, each pixel in Mode II is modeled by two GMMs (called models A and B, respectively) with different learning rates (i.e., the high learning rate and the low learning rate). When an abandoned object is detected, the GMM model A with the high learning rate would show the static object faster than the GMM model B with the low learning rate. At the moment that the static object is already stable in model A but still unstable in model B, the object can be discovered by subtracting A from B. The detailed algorithm is described as follows:

Mode II AOD Algorithm

Input: A pixel of the incoming frame.

Output: Determine whether the input pixel belongs to an abandoned object.

Step 1: Extract the first i ($i \leq k$) Gaussian distributions from the GMM model A with the large learning rate for the input pixel and sort them according to their weights in non-increasing order to obtain $(\mu_{Aj'}, w_{Aj'}, \sigma_{Aj'})$, where $j = 1$ to i. Similarly, extract the first i Gaussian distributions from the GMM model B with the small learning rate to obtain $(\mu_{Bj'}, w_{Bj'}, \sigma_{Bj'})$.

Step 2: Normalize $w_{Ai'}$ and $w_{Bi'}$.

Step 3: If $\sum_{j=1}^{i} [\max(w_{Aj}, w_{Bj}) \times (\mu_{Aj} - \mu_{Bj})] > TH_1$, and $\sigma_{Aj'} < TH_2$, and $\sigma_{Bj'} < TH_2$, the pixel is classified as a candidate pixel of an abandoned object.

Step 4: Get the corresponding bitmap BM of the input pixel. If BM is recognized as an unmatched block, and the weight of BM is finally increased to the largest one in TM after several runs, the pixel is classified as a candidate pixel of an abandoned object.

Step 5: If the frequency of the appearance of the candidate pixel is above P times, the pixel is considered to be a pixel of an abandoned object.

Similarly, Step 4 can be optional. Mode II is more robust than Mode I owing to the fact that Model II considers the differences between multiple distributions and uses different models' standard deviations to constrain noise-production. The detection results of Mode I tend to be significantly affected by the learning rate; furthermore, selecting a proper learning rate in Mode I involves some trial and error. This phenomenon can be greatly alleviated in Mode II, and it is easier to choose a pair of learning rates.

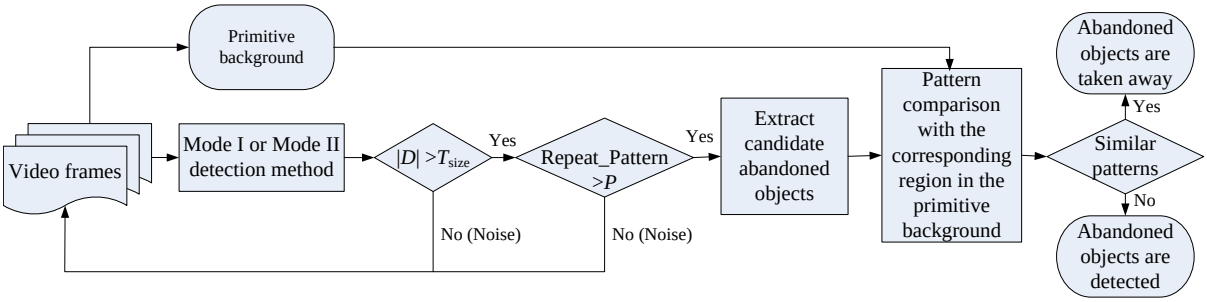

FIGURE 3. The flowchart of the abandoned objects management system

3.3. **Removed object detection.** If someone were to carry the left object away, Mode I and Mode II would produce a false detection because the system cannot be aware of the removal of the static object. To distinguish the ambiguous cases, we construct another primitive background using another GMM model, called PBGMM. Initially, PBGMM is unstable, but the update process is the same as that of original GMM. However, when PBGMM becomes stable, if the new incoming pixel does not match the PBGMM, the least significant distribution of PBGMM would not be replaced, neither is the update procedure required. This is unlike the previous GMM model. Therefore, when the primitive background is stable, it only allows slight lighting changes to fit long-term monitoring, and new incoming objects would not be incorporated.

When Mode I or Mode II detects a suspicious abandoned object, the color histogram of the object is extracted to compare with that of the corresponding region in the primitive background. The status of the static object would be judged as "REMOVED" if the two histograms are similar; otherwise, it is considered that an abandoned object shows up.

The verification of pattern matching also facilitates noise removal and reduces false detections. For example, if noise occurs (such as severe lighting changes or someone putting a bag down but then removing it without much delay), the system may detect a pseudo candidate static object placed on the ground. Nevertheless, in the verification phase, the two matching patterns must be similar so the false detection can be avoided.

Figure 3 shows the flowchart of the detection system, where D is an object of the detection result, $|D|$ denotes the size of D, and T_{size} is a size threshold for D. When the size of D is smaller than or equal to a predefined threshold, D is regarded as noise and is neglected. In addition, for better shape of D, some image processing techniques such as size filter, morphology, and connected components labeling can be applied [4,7,9].

3.4. **Sensitivity of detection.** The sensitivity of the proposed detection method depends mainly on the learning rate. The higher the learning rate is, the more sensitive the detection has. In Mode I and Mode II detection methods, once the order of weights of the distributions is changing, the statuses of static objects are starting to be monitored. Therefore, the weight-updating process can be used to analyze the sensitivity. Recall that in Equation (6) a new weight is updated by the previous weight. This equation can be changed to Equation (17) to represent the new weight after being updated t times. For example, assume that $w_{k,0}$ is 0.05, α is 0.04, and $w_{k,t}$ is 0.5. According to Equation (17), there must be at least 16 updates from $w_{k,0}$ to $w_{k,t}$, which means that an abandoned object can be detected through at least 16 frames. However, if the learning rate is changed to 0.08, the number of detection frames can be reduced to 8, which will give greater sensitivity.

$$w_{k,t} = (1-\alpha)^t w_{k,0} + (1-(1-\alpha)^t) M_{k,t}. \tag{17}$$

3.5. **Abandoned objects maintenance.** After the system performs detecting activities for a period of time, the scene may contain several abandoned objects, and some of them may be occluded by other static objects. Once the occluded object is taken away, the system should observe this event correctly. In the proposed method, we use a multilayer layout to maintain abandoned objects. Each layer records the status of an abandoned object and the layers are ordered according to the time the abandoned object occurs. When a suspect abandoned object is detected, the verification process checks all existing layers and helps to decide the status of the object (incoming or removing). If the status is "incoming", a new layer is added to record the object; otherwise, in the "removing" case, the object and the corresponding layer are removed. The proposed multilayer method with a single camera cannot solve all possible cases of abandoned objects. For example, when the abandoned object could be placed behind some object (totally occluded), multi-camera deployment should be considered.

4. **Experiments.** The performance of the proposed method is verified using five video sequences that were acquired from real outdoor environments. Figure 4 shows some frames from the video data, where video 1 demonstrates that one static object is left, video 2 demonstrates that a static object is left but is carried away later, video 3 gives an example of leaving multiple abandoned objects, video 4 illustrates the case of occluded abandoned objects, and video 5 presents a synthetic case of multiple static objects, an occluded object, crowds walking, and a removed object. The simulation environment for the experiments is equipped with a 1.8 GHz Core 2 Intel processor and 2 GB of memory. The image resolution was set to 320×240 pixels. All algorithms were implemented in Microsoft Visual C++. The number of Gaussian distributions used in each GMM is set to 3. In addition, the learning rate of Mode I is set to 0.05, and the high and low learning rates of Mode II are set to 0.05 and 0.025, respectively.

The results of finding candidate static objects of Mode I, Mode II, and Martinez-del-Rincon et al.'s method [14] are shown in Figures 5-7, respectively. From these figures we can clearly observe that Mode II has the best detection results in terms of detection correctness, shapes of static objects, and noise resistance. In practice, each foreground object (moving object) of Martinez-del-Rincon et al.'s method has a high probability of becoming a candidate static object. The reason for this is that Martinez-del-Rincon et al. use $\tilde{}(I_k - I_{k-1}) \cap (I_k - S)$ to detect abandoned objects, but the moving objects usually exist in $\tilde{}(I_k - I_{k-1})$ and $(I_k - S)$, as shown in Figures 8(d) and 8(e). Therefore, Martinez-del-Rincon et al.'s method completely depends on the counter for each pixel, leading to noise sensitive and increasing false detections. On the other hand, no matter whether Mode I or Mode II method is employed, the moving objects can hardly become candidate static objects, and the shapes of the candidate static objects can be easily refined by the connected components labeling method and the closing operation of morphology, which can be seen the results in Figures 8(g) and 8(h).

The summary of the detection results using different methods is presented in Table 2. The row of "number of static objects" and the row of "number of removed objects" indicate the actual number of objects in the scene. The value x/y presented in this table means that the total numbers of x static objects and y removed objects are detected. The table shows that the Mode II method can provide the most accurate result among these methods. Since Martinez-del-Rincon et al.'s method does not offer the function of detecting removed objects, the number of removed objects in this method is always 0.

The noise generated by the compared methods (excluding the moving objects) is presented in Table 3. The number, counted by hand, in this table shows how many pseudo-blobs are generated during detection. It is clear that Martinez-del-Rincon et al.'s method

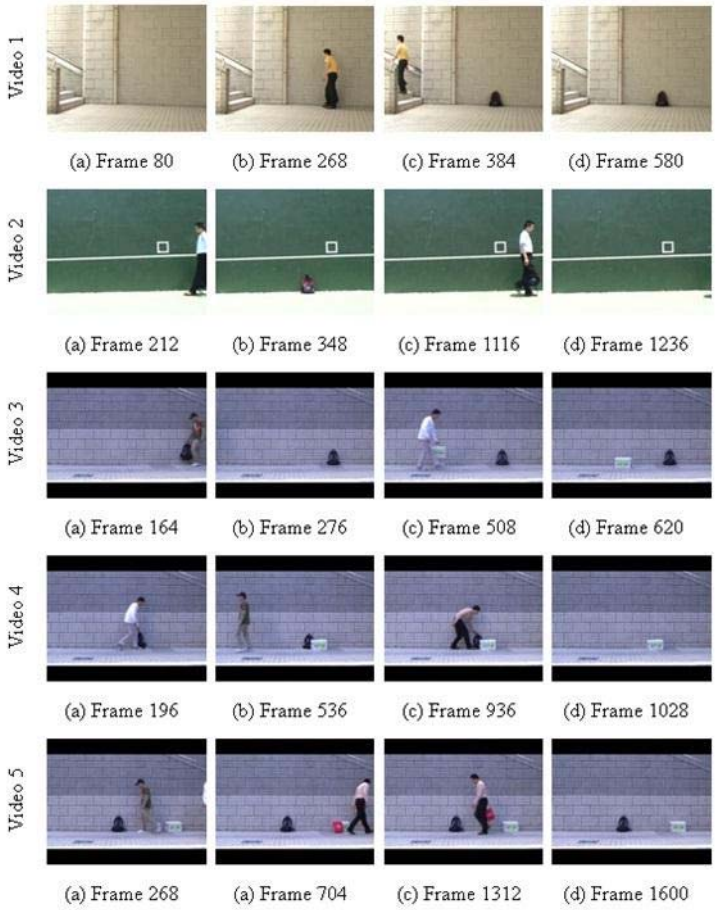

FIGURE 4. Original video images

TABLE 2. The detection results using different methods

Scenario	Video 1	Video 2	Video 3	Video 4	Video 5
Number of static objects	1	1	2	2	3
Number of removed objects	0	1	0	1	1
Mode I	2	1/1	3	2/1	3/1
Mode II	1	1/1	2	2/1	3/1
Martinez-del-Rincon et al.'s method	2	1/0	3	2/0	4/0

produces the most noise because of the simple background construction method and the imperfect subtraction results. In Mode I, the noise is generated due to lighting changes and irregular camera vibrations, and in Mode II, most of the noise comes from camera vibrations.

Table 4 compares the sensitivity of detecting candidate abandoned objects. Since Martinez-del-Rincon et al.'s method acquires the candidate objects by direct subtraction,

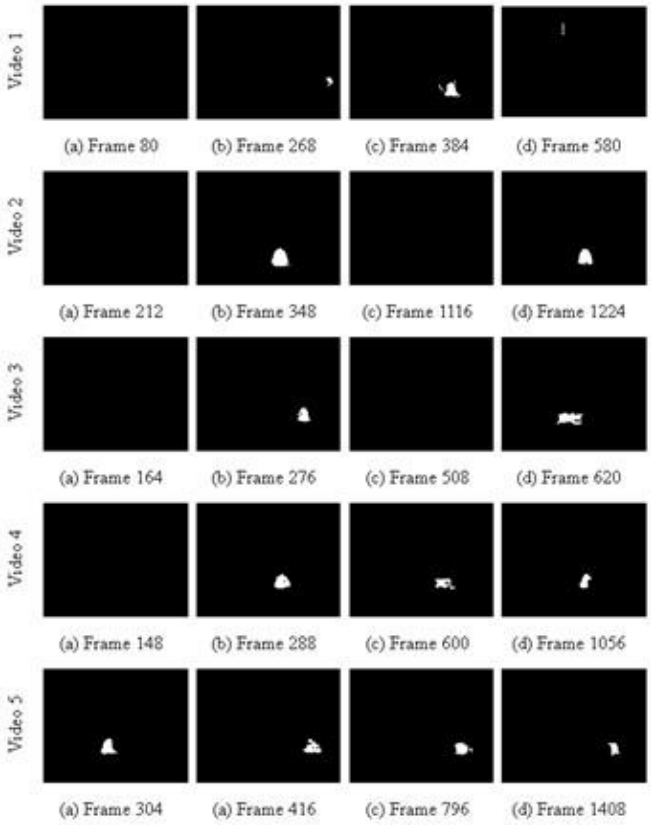

FIGURE 5. Resultant images of Mode I

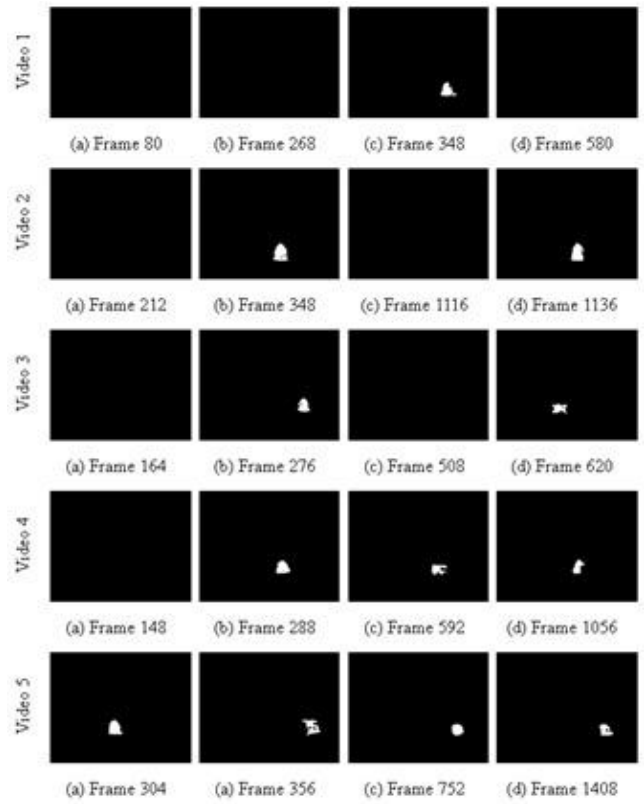

FIGURE 6. Resultant images of Mode II

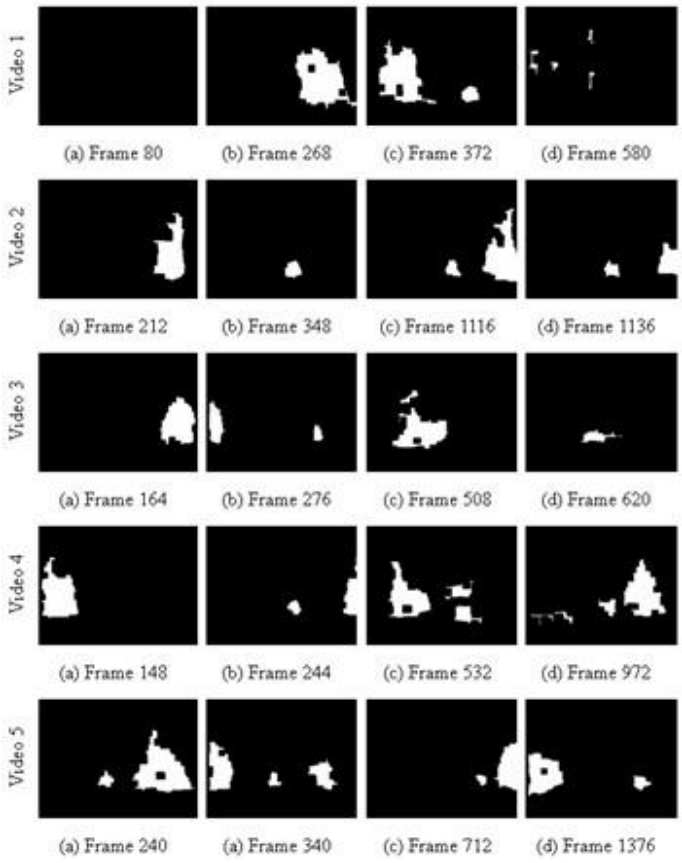

FIGURE 7. Resultant images of Martinez-del-Rincon et al.'s method

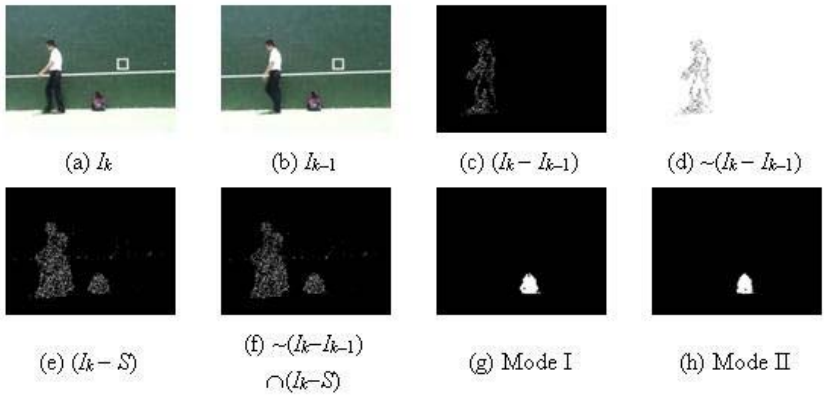

FIGURE 8. The resultant images

it can detect the static object immediately. However, the proposed method is determined by the learning rate, so the candidate blobs can be obtained only when the static object actually becomes stable. This also explains why the proposed method is more robust than simple subtraction methods.

The comparison of efficiency is shown in Table 5. The proposed method uses texture and color GMM-based models to detect static objects, but Martinez-del-Rincon et al. utilize only the single Gaussian model and median method to construct short-term and long-term backgrounds respectively. Therefore, the frame rate of the proposed method is lower than that of Martinez-del-Rincon et al.'s method. (Note that the frame rate

TABLE 3. The number of unrelated blobs generated by the compared methods

Scenario	Video 1	Video 2	Video 3	Video 4	Video 5
Mode I	7	0	2	0	1
Mode II	2	0	0	0	0
Martinez-del-Rincon et al.'s method	34	3	5	7	6

TABLE 4. The sensitivity comparison

Method	Mode I	Mode II	Martinez-del-Rincon et al.'s method
Number of frames	18	14	2

TABLE 5. The frame rate of the compared methods

Method	Mode I	Mode II	Martnez-del-Rincn et al.'s method
Frame rate	8.7	6.1	13.7

shown here involves all the required image processing operations such as morphology and filtering.) Since Gao et al. [6] have proved that the GMM model is better than single Gaussian for real-time and long-term monitoring systems, building the abandoned object detection system based on the present GMM model can create a higher value for current surveillance systems. Table 6 shows the comparisons with different methods developed by companies and research groups. Beynon et al.'s method [2] is developed by the MIT laboratory, Wang and Ooi's method [19] is proposed by Cornell's research group, Guler and Farrow's method [8] is designed by IntuVision Inc., and dual background models [14,15] are proposed by Martinez-del-Rincon et al. and Singh et al. and use a similar basis to the proposed method. In the first column, the "Detection approach" means that the abandoned objects are detected by subtraction or tracking; "Learning template in advance" means that the template of abandoned objects should be learned in advance or not; "Number of candidates" means that what kind of foreground objects will be regarded as candidates of abandoned objects; "Detect removed objects" means whether the method can detect the removed objects. In the proposed method we apply a subtraction-based method without learning any template in advance, thus regarding only the foreground static objects as candidates for abandoned objects, and discuss how to detect removed objects and resist interference. With these features, the proposed method can outperform other methods in terms of efficiency and effectiveness.

TABLE 6. Comparisons of different methods

Items	Proposed method	Beynon et al.'s method [2]	Wang and Ooi's method [19]	Guler and Farrow's method [8]	Dual background models [14,15]
Detecting approach	Subtraction base	Tracking base	Subtraction base	Tracking base	Subtraction base
Learning template in advance	No	Yes	No	No	No
Number of candidates	Only foreground static objects	All foreground objects	All foreground objects	All foreground objects	Only foreground static objects
Detect removed objects	Yes	No	No	No	No
Resist noise & illumination changes	Good	Fair	Fair	Fair	Fair
Efficiency	High	Low	Low	Low	High

5. **Conclusions.** In this paper, an automatic management system has been proposed for supervising the status of abandoned objects in real environments. According to the previous research, a robust abandoned objects detection method is still required, and the case of removed objects is scarcely discussed. Therefore, the proposed system provides a comprehensive solution that can deal with abandoned, removed, and partially occluded objects. Without using complicated tracking methods, two detection methods (Mode I and Mode II) are proposed based on the Gaussian mixture model, which can resist noise generation and is also commonly used in current video surveillance systems. The experiments consider different degrees of complexity of the test cases to verify that the proposed management system can successfully accomplish the task of detecting abandoned objects. From the experimental results, Mode II is more robust than either Mode I or Martinez-del-Rincon et al.'s method in terms of noise resistance and detecting results. In addition, the proposed system is suitable for long-time detection while maintaining the accuracy of detection results, which is rarely achieved by conventional methods.

Acknowledgement. This work was supported by National Science Council, Taiwan, under Grant NSC 100-2221-E-468-021.

REFERENCES

[1] E. Auvinet, E. Grossman, C. Rougier, M. Dahmane and J. Meunier, Left-luggage detection using homographies and simple heuristics, *Proc. of the 9th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, New York, USA, pp.51-58, 2006.

[2] M. D. Beynon, D. J. V. Hook, M. Seibert, A. Peacock and D. Dudgeon, Detecting abandoned packages in a multi-camera video surveillance system, *Proc. of IEEE Conference on Advanced Video and Signal Based Surveillance*, Miami, FL, USA, pp.221-228, 2003.

[3] L. Cao, In-depth behavior understanding and use: The behavior informatics approach, *Information Sciences*, vol.180, no.17, pp.3067-3085, 2010.

[4] F. Chang, C. J. Chen and C. J. Lu, A linear-time component-labeling algorithm using contour tracing technique, *Computer Vision and Image Understanding*, vol.93, no.2, pp.206-220, 2004.

[5] N. Friedman and S. Russell, Image segmentation in video sequences: A probabilistic approach, *Proc. of the 13th Conference on Uncertainty in Artificial Intelligence*, Providence, Rhode Island, pp.175-181, 1997.

[6] X. Gao, T. Boult, F. Coetzee and V. Ramesh, Error analysis of background adaption, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, pp.503-510, 2000.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ, USA, 2002.

[8] S. Guler and M. Farrow, Abandoned object detection in crowded places, *Proc. of the 9th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, New York, USA, pp.99-106, 2006.

[9] R. Jain, R. Kasturi and B. G. Schunck, *Machine Vision*, McGraw-Hill, Singapore, 1995.

[10] E. H. Jaraba, C. O. Urunuela and J. Senar, Detected motion classification with a double-background and a neighborhood-based difference, *Pattern Recognition Letters*, vol.24, no.12, pp.2079-2092, 2003.

[11] L. Li, R. Luo, R. Ma, W. Huang and K. Leman, Evaluation of an IVS system for abandoned object detection on PETS2006 datasets, *Proc. of the 9th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, New York, USA, pp.91-98, 2006.

[12] C. Y. Lin, C. C. Chang, W. W. Chang, M. H. Chen and L. W. Kang, Real-time robust background modeling based on joint color and texture descriptions, *Proc. of the 4th International Conference on Genetic and Evolutionary Computing*, Shenzhen, China, pp.622-625, 2010.

[13] F. Lv, X. Song, B. Wu, V. K. Singh and R. Nevatia, Left luggage detection using Bayesian inference, *Proc. of the 9th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, New York, pp.83-90, 2006.

[14] J. Martinez-del-Rincon, J. E. Herrero-Jaraba, J. R. Gomez and C. Orrite-Urunuela, Automatic left luggage detection and tracking using multi-camera UKF, *Proc. of the 9th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, New York, USA, pp.59-66, 2006.

[15] A. Singh, S. Sawan, M. Hanmandlu, V. K. Madasu and B. C. Lovell, An abandoned object detection system based on dual background segmentation, *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Genova, pp.352-357, 2009.

[16] C. Stauffer and W. E. L. Grimson, Adaptive background mixture models for real-time tracking, *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, USA, pp.246-252, 1999.

[17] C. Stauffer and W. E. L. Grimson, Learning patterns of activity using real-time tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, no.8, pp.747-757, 2000.

[18] E. Stringa and C. S. Regazzoni, Real-time video-shot detection for scene surveillance applications, *IEEE Transactions on Image Processing*, vol.9, no.1, pp.69-79, 2000.

[19] J. Wang and W. Ooi, Detecting static objects in busy scenes, *Technical Report TR99-1730*, Department of Computer Science, Cornell University, 1999.