Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 197875, 11 pages
doi:10.1155/2008/197875

*Research Article*

# Robust Abandoned Object Detection Using Dual Foregrounds

**Fatih Porikli,[1] Yuri Ivanov,[1] and Tetsuji Haga[2]**

[1] *Mitsubishi Electric Research Labs (MERL), 201 Broadway, Cambridge, MA 02139, USA*
[2] *Mitsubishi Electric Corp. Advanced Technology R&D Center, Amagasaki, 661-8661 Hyogo, Japan*

Correspondence should be addressed to Fatih Porikli, fatih@merl.com

Received 25 January 2007; Accepted 28 August 2007

Recommended by Enis Ahmet Çetin

As an alternative to the tracking-based approaches that heavily depend on accurate detection of moving objects, which often fail for crowded scenarios, we present a pixelwise method that employs dual foregrounds to extract temporally static image regions. Depending on the application, these regions indicate objects that do not constitute the original background but were brought into the scene at a subsequent time, such as abandoned and removed items, illegally parked vehicles. We construct separate long- and short-term backgrounds that are implemented as pixelwise multivariate Gaussian models. Background parameters are adapted online using a Bayesian update mechanism imposed at different learning rates. By comparing each frame with these models, we estimate two foregrounds. We infer an evidence score at each pixel by applying a set of hypotheses on the foreground responses, and then aggregate the evidence in time to provide temporal consistency. Unlike optical flow-based approaches that smear boundaries, our method can accurately segment out objects even if they are fully occluded. It does not require on-site training to compensate for particular imaging conditions. While having a low-computational load, it readily lends itself to parallelization if further speed improvement is necessary.

## 1. INTRODUCTION

Conventional approaches on abandoned item detection can be grouped as motion detectors [1–3], object classifiers [4], and tracking-based analytics approaches [5–10].

In [2], a dense optical flow map is estimated to infer the foreground objects moving in opposite directions, moving in a group, and staying stationary by predetermined rules. In [3], a pixel-based method for characterizing objects introduced into the static scene by comparing the background image estimated from the current frame with the previous ones is described. This approach requires storing as many backgrounds as the minimum detection duration in the memory and causes ghost detections even after the abandoned item is removed from the scene.

Recently, an online classifier [4] that incorporates a boosting-based feature selection to label image blocks as background, valid objects, and unidentified regions is presented. This method adapts itself to the depicted scene, however, fails short of discriminating moving objects from stationary ones. Classifier-based methods face with the challenge of dealing with unknown object type as such objects can vary from small luggage to ski bags.

A considerable amount of effort has been devoted to hypothesize abandoned items by analyzing object trajectories [5–7, 9, 10] in multicamera setups. In principle, these methods require solving a harder problem of object initialization and tracking as an intermediate step in order to identify the parts of the video frames corresponding to an abandoned object. It is often assumed that the background scene is nearly static or periodically varying, while the foreground comprises groups of pixels that are different from the background. However, object detection in crowded scenes, especially for uncontrolled real-life situations, is problematic due to the partial occlusions, heavy shadows, people entering the scene together, and so forth. Moreover, object appearance is often indiscriminative as people tend to dress in similar colors, which leads inaccurate tracking results.

For static camera setups, background subtraction provides strong cues for apparent motion statistics. Various background generation methods have been employed in a quest for a system that is robust to changing illumination conditions, appearance variations, shadows, camera jitter, and severe noise. Parametric mixture models are employed to handle such variations. Stauffer and Grimson [11] propose an expectation maximization- (EM-) based adaptation

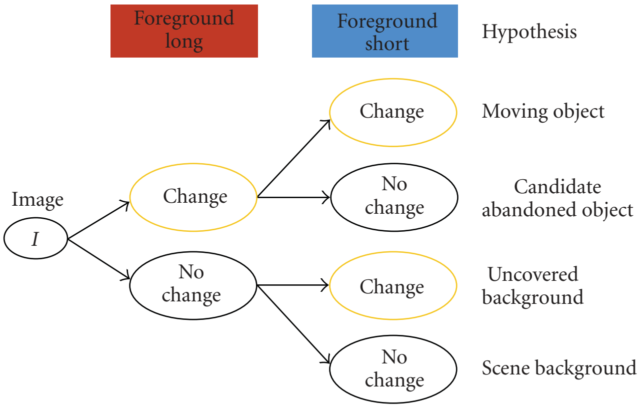


Figure 1: Hypotheses on long- and short-term foregrounds.

method to learn a mixture of Gaussians with predetermined number of models at each pixel using fixed learning parameters. The online EM update causes a weak model, which has a larger variance, to be dissolved into a dominant model, which has a smaller variance in case the mean value of the weak model is close to the mean of the dominant one. To address this issue, Porikli and Tuzel [12] develop an online Bayesian update mechanism for adaptation multivariate Gaussian distributions. This method estimates the number of necessary layers for each pixel and the posterior distributions of mean and covariance of each layer by assuming the data to be normally distributed with mean and covariance as random variables.

There are other variants of the mixture of models that use modified feature spaces, image gradients, optical flow, and region segmentation [13–15]. Instead of iteratively updating models as mixture methods, nonparametric kernel density estimation [16] stores a large number of previous frames and estimates weights of multiple kernel functions. Since both memory and computational complexity proportionally increases with the number of stored frames, kernel methods are usually impractical for real-time applications.

There exists a class of problems that cannot be solved by the traditional foreground-background detection methods. For instance, objects deliberately abandoned in public places, such as suitcases, packages, do not fall into either of these two categories. They are static; therefore, they should be labeled as background. On the other hand, they should not be ignored as they do not belong to the original scene background. Depending on the learning rate, the pixels corresponding to the temporary static objects can be mistaken as a part of the scene background (in case of a high-learning rate), or grouped with the moving regions (low-learning rate). A single background is not sufficient to separate the temporarily static pixels from the scene background.

In this paper, we propose a pixel-based method that employs dual foregrounds. Our motivation is that by changing the background learning rate, we can adjust how soon a static object should be blended into the background. Therefore, temporarily static image regions can be distinguished from the longer term background and moving regions by analyzing multiple foregrounds of different learning rates. This simple idea is wrapped into our adaptive background estimation algorithm, where the slowly adapting background and the fast adapting foreground are aggregated into an evidence image. We impose different learning rates by processing video at different temporal resolutions. The background models have identical initial parameters, thus they require minimal fine tuning in the setup stage. The evidence statistics are used to extract temporarily static image areas, which may correspond to abandoned items, illegally parked vehicles, objects removed from the scene, and so forth, depending on the application.

Our method does not require object initialization, tracking, or offline training. It accurately segments objects even if they are fully occluded. It has a very low-computational load and readily lends itself to parallelization if further speed improvements are necessary. In the subsequent sections, we give details of the dual foregrounds, show Bayesian adaptation method, and present results on real-world data.

## 2. DUAL FOREGROUNDS

To detect an abandoned item (or an illegally parked vehicle, removed article, etc.), we need to know how it alters the temporal and spatial statistics of the video data. We built our method on the fact that an abandoned item is not a part of the original scene, it was brought into the scene not that long ago, and it remained still after it has been left. In other words, it is a temporarily static object which was not there before. This means that by learning the prolonged static scene and the moving foreground regions, we can hypothesize on whether a pixel corresponds to an abandoned item or not.

A scene background can be determined by maintaining a statistical model that captures the most consistent modes of the color distribution of each pixel in extended durations of time. From this background, the changed pixels that do not fit into the statistical models are obtained. However, depending on the learning rate, the pixels corresponding to the temporary static objects can be mistaken as a part of the scene background (higher-learning rates), or grouped with the moving regions (lower-learning rates). A single background is not sufficient to separate the temporarily static pixels from the scene background.

As opposed to single background approaches, we use two backgrounds to obtain both the prolonged (long-term) background $B_L$ and the temporarily static (short-term) background $B_S$. Note that it is possible to improve the temporal granularity by employing more than two backgrounds at different learning rates. Each of these backgrounds is defined as a mixture of Gaussian models. We represent a pixel as layers of 3D multivariate Gaussians where each dimension corresponds to a color channel. Each layer models to a different appearance of the pixel. We perform our operations on the RGB color space. We apply a Bayesian update mechanism. At each update, at most one layer is updated with the current observation. This assures the minimum overlap over the layers. We also determine how many layers are necessary for each pixel and use only those layers during the foreground segmentation phase. This is performed with

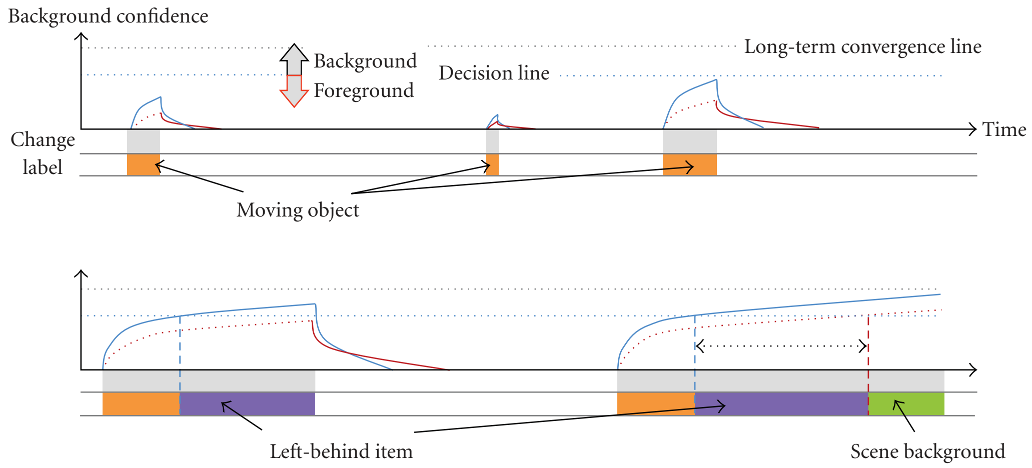


Figure 2: The confidence of the long-term and short-term background models (vertical axis) changes differently for ordinary objects (moving or temporarily stationary ones), abandoned items, and scene background.

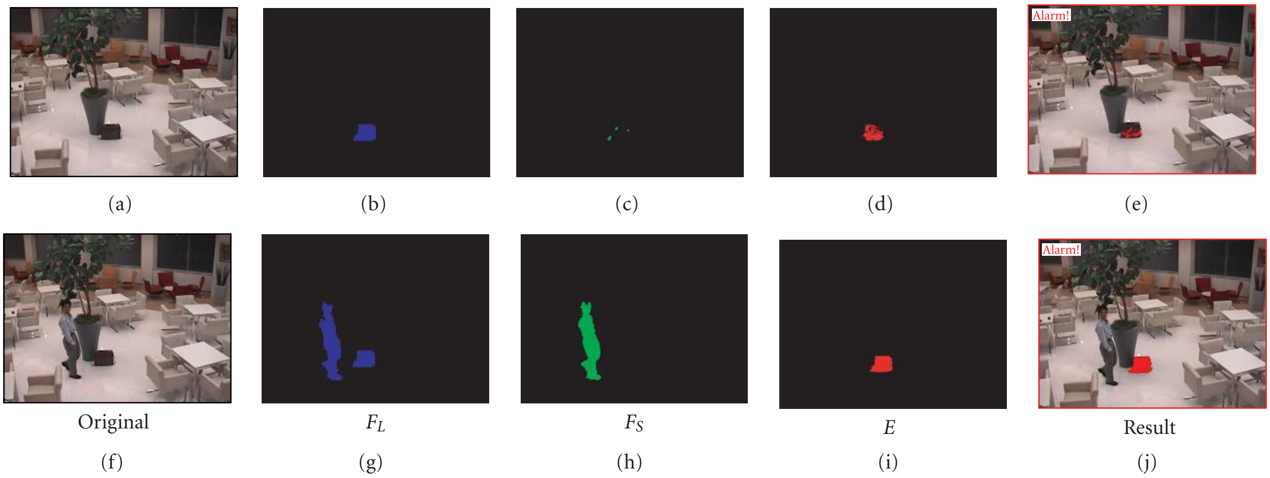


Figure 3: First row: $t = 350$. Second row: $t = 630$. The long-term foreground $F_L$ captures moving objects and temporarily static regions. The short-term foreground $F_S$ captures only moving objects. The evidence $E$ gets greater as the object stays longer.

an embedded confidence score. Both of the backgrounds have identical initial parameters, such as the initial mean and variance of the marginal posterior distribution, the degrees of freedom, and the scale matrix, except the number of the prior measurements, which is used as a learning parameter.

At every frame, we estimate the long and short term foregrounds by comparing the current frame $I$ by the background models $B_L$ and $B_S$. We obtain two binary foreground masks $F_L$ and $F_S$, where $F(x, y) = 1$ indicates that the pixel $(x, y)$ is changed. The long term foreground mask $F_L$ shows the color variations in the scene that were not there before including moving objects, temporarily static objects, as well as moving cast shadows and illumination changes that the background models fail to adapt. The short-term foreground mask $F_S$ contains the moving objects, noise, and so forth. Depending on the foreground mask values, we postulate the following hypotheses as shown in Figure 1.

(1) $F_L(x, y) = 1$ and $F_S(x, y) = 1$, where $(x, y)$ is a pixel that may correspond to a moving object since $I(x, y)$ does not fit any backgrounds.

(2) $F_L(x, y) = 1$ and $F_S(x, y) = 0$, where $(x, y)$ is a pixel that may correspond to a temporarily static object.

(3) $F_L(x, y) = 0$ and $F_S(x, y) = 1$, where $(x, y)$ is a scene background pixel that was occluded before.

(4) $F_L(x, y) = 0$ and $F_S(x, y) = 0$, where $(x, y)$ is a scene background pixel since its value $I(x, y)$ fits both backgrounds $B_L$ and $B_S$.

The short term background is updated at a higher-learning rate than the long-term background. Thus, the short-term background adapts to the underlying distribution faster and the changes in the scene are blended more rapidly. In contrast, the long-term background is more resistant against the changes.

```
Given: New sample $\mathbf{x}$, background layers $\{(\theta_{t-1,i}, \mathbf{\Lambda}_{t-1,i}, \kappa_{t-1,i}, \upsilon_{t-1,i})\}_{i=1,\dots,k}$
    Sort layers according to confidence measure defined in (11). $i \leftarrow 1$.
    While $i < k$
        Measure Mahalanobis distance:
        $d_i \leftarrow (\mathbf{x} - \mu_{t-1,i})^T \mathbf{\Sigma}_{t-1,i}^{-1} (\mathbf{x} - \mu_{t-1,i})$.
        If sample $\mathbf{x}$ is in 99% confidence interval,
            then update model parameters according to (6), and stop.
            else update model parameters according to (13).
        $i \leftarrow i + 1$
    Delete layer $k$, initialize a new layer having parameters defined in (7).
```

Algorithm 1

In case a scene background pixel changes temporarily then sets back to its original value, the long-term foreground mask will be zero; $F_L(x, y) = 0$. The short term background is pliant and adapts itself during this time, which causes $F_S(x, y) = 1$. We assume it takes more time to adapt the long-term background to the newly observed color than the change period. A changed pixel will be blended into the short-term background, that is, $F_S(x, y) = 0$, if it keeps its new color long enough. If this duration is not prolonged enough to blend it, the long term-foreground mask will be one; $F_L(x, y) = 1$. This is the common case for the abandoned items. If no change is observed in neither of the backgrounds $F_L(x, y) = 0$ and $F_S(x, y) = 0$, the pixel is considered as a part of the static scene background as the pixel has the same value for much longer periods of time.

The dual foreground mechanism is illustrated in Figure 2. In this simplified drawing, the horizontal axis corresponds to time and the vertical axis to the confidence of the background model. *Action* indicates that the pixel color has significantly changed. *Label* represents the result of the above hypotheses. For pixels with relatively short duration of change, the confidences of the long- or short-term models do not increase enough to make them valid backgrounds. Thus, such pixels are labeled as moving object. Whenever the short-term model blends the pixel in the background but the long-term model still marks it as foreground, the pixel is considered to belong to the abandoned item. Finally, if the pixel change takes even longer, the pixel is labeled as a scene background. Sample foregrounds that show these cases are given in Figure 3.

We aggregate the framewise detection results into an evidence image $E(x, y)$ by updating the pixelwise values at each frame as

$$E(x,y) = \begin{cases} E(x,y) + 1 & F_L(x,y) = 1 \wedge F_S(x,y) = 0, \\ E(x,y) - k & F_L(x,y) \neq 1 \vee F_S(x,y) \neq 0, \\ \max_e, & E(x,y) > \max_e, \\ 0, & E(x,y) < 0, \end{cases} \tag{1}$$

where $\max_e$ and $k$ are positive numbers. The evidence image enables removing noise in the detection process. It also controls the minimum time required to assign a static pixel as an abandoned item. For each pixel, the evidence image collects the motion statistics. Whenever it elevates up to a preset level $E(x, y) > \max_e$, we mark the pixel as an abandoned item pixel and raise an alarm flag. The evidence threshold $\max_e$ is defined in term of the number of frames and it can be chosen depending on the desired responsiveness and noise characteristics of the system. In case the foreground detection process produces noisy results, higher values of $\max_e$ should be preferred. High values of $\max_e$ lower the false alarm rate. On the other hand, the higher the preset level gets, the longer the minimum duration a pixel takes to be classified as a part of an abandoned item. A typical range of the evidence threshold $\max_e$ is 300 frames.

The decay constant $k$ determines how fast the evidence should decrease. In other words, it decides what should happen in case a pixel that is marked as an abandoned item is blended into the scene background or gets its original value before the marking. To set the alarm flag off immediately after the removal of object, the value of decay should be large, for example, $k = \max_e$. This means that there is only a single parameter to set for the likelihood image. In our experiments, we observed that the larger values of decay constant generate satisfying results.

In the following section, we describe the adaptation of the long- and short-term background models by a Bayesian update mechanism.

## 3. BAYESIAN UPDATE

Our background model [12] is similar to adaptive mixture models [11] but instead of mixture of Gaussian distributions, we define each pixel as layers of 3D multivariate Gaussians. Each layer corresponds to a different appearance of the pixel. Using Bayesian approach, we are not estimating the mean and variance of the layer, but the probability distributions of mean and variance. We can extract statistical information regarding these parameters from the distribution functions. For now, we are using expectations of mean and variance for change detection, and variance of the mean for confidence.

### *3.1. Layer model*

Data is assumed to be normally distributed with mean $\mu$ and covariance $\mathbf{\Sigma}$. Mean and variance are assumed unknown and

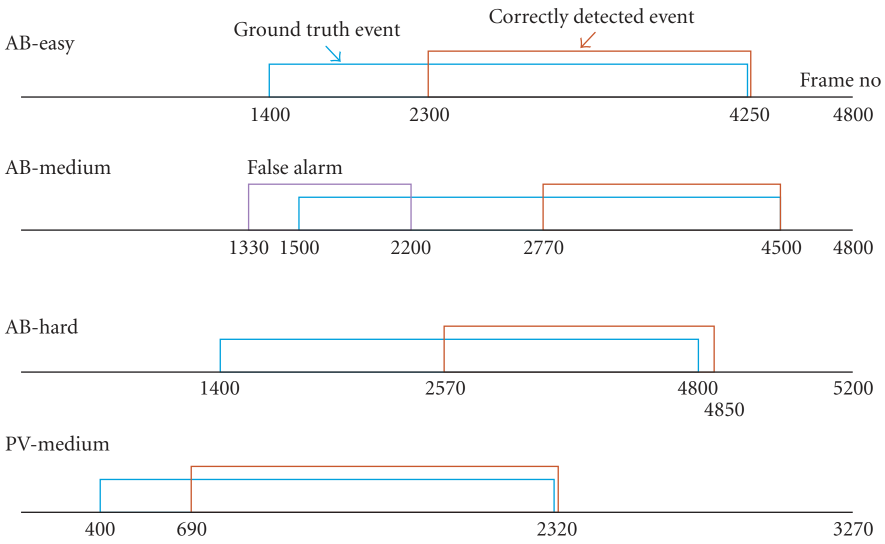


Figure 4: Detected events for i-LIDS datasets.

modeled as random variables. Using Bayesian theorem, joint posterior density can be written as

$$p(\mu, \mathbf{\Sigma} \mid \mathbf{X}) \propto p(\mathbf{X} \mid \mu, \mathbf{\Sigma}) p(\mu, \mathbf{\Sigma}). \tag{2}$$

To perform recursive Bayesian estimation with the new observations, joint prior density $p(\mu, \mathbf{\Sigma})$ should have the same form with the joint posterior density $p(\mu, \mathbf{\Sigma} \mid \mathbf{X})$. Conditioning on the variance, joint prior density is written as

$$p(\mu, \mathbf{\Sigma}) = p(\mu \mid \mathbf{\Sigma}) p(\mathbf{\Sigma}). \tag{3}$$

The above condition is realized if we assume inverse Wishart distribution for the covariance and, conditioned on the covariance, multivariate normal distribution for the mean. Inverse Wishart distribution is a multivariate generalization of scaled inverse $\chi^2$-distribution. The parametrization is

$$\begin{aligned} \mathbf{\Sigma} &\sim \text{Inv-Wishart}_{\upsilon_{t-1}}\left(\mathbf{\Lambda}_{t-1}^{-1}\right), \\ \mu \mid \mathbf{\Sigma} &\sim \mathbf{N}\left(\theta_{t-1}, \frac{\mathbf{\Sigma}}{\kappa_{t-1}}\right), \end{aligned} \tag{4}$$

where $\upsilon_{t-1}$ and $\mathbf{\Lambda}_{t-1}$ are the degrees of freedom and scale matrix for inverse Wishart distribution, $\theta_{t-1}$ is the prior mean, and $\kappa_{t-1}$ is the number of prior measurements. With these assumptions, joint prior density becomes

$$\begin{aligned} p(\mu, \mathbf{\Sigma}) \propto & \left|\mathbf{\Sigma}\right|^{-((\upsilon_{t-1}+3)/2+1)} \\ & \times e^{(-(1/2)\mathrm{tr}(\mathbf{\Lambda}_{t-1}\mathbf{\Sigma}^{-1})-(\kappa_{t-1})/2(\mu-\theta_{t-1})^T\mathbf{\Sigma}^{-1}(\mu-\theta_{t-1}))} \end{aligned} \tag{5}$$

for three-dimensional feature space. Let this density be labeled as normal inverse Wishart $(\theta_{t-1}, \mathbf{\Lambda}_{t-1}/\kappa_{t-1}; \upsilon_{t-1}, \mathbf{\Lambda}_{t-1})$. Multiplying prior density with the normal likelihood and arranging the terms, joint posterior density becomes normal inverse Wishart $(\theta_t, \mathbf{\Lambda}_t/\kappa_t; \upsilon_t, \mathbf{\Lambda}_t)$ with the parameters updated:

$$\begin{aligned} \upsilon_t &= \upsilon_{t-1} + n \qquad \kappa_n = \kappa_{t-1} + n, \\ \theta_t &= \theta_{t-1}\frac{\kappa_{t-1}}{\kappa_{t-1}+n} + \overline{\mathbf{x}}\frac{n}{\kappa_{t-1}+n}, \\ \mathbf{\Lambda}_t &= \mathbf{\Lambda}_{t-1} + \sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T \\ &\quad + n\frac{\kappa_{t-1}}{\kappa_t}(\overline{\mathbf{x}} - \theta_{t-1})(\overline{\mathbf{x}} - \theta_{t-1})^T, \end{aligned} \tag{6}$$

where $\overline{\mathbf{x}}$ is the mean of new samples and $n$ is the number of samples used to update the model. If update is performed at each time frame, $n$ becomes one. To speed up the system, update can be performed at regular time intervals by storing the observed samples. During our tests, we update one quarter of the background at each time frame, therefore $n$ becomes four. The new parameters combine the prior information with the observed samples. Posterior mean $\theta_t$ is a weighted average of the prior mean and the sample mean. The posterior degrees of freedom is equal to prior degrees of freedom plus the sample size. System is started with the following initial parameters:

$$\kappa_0 = 10, \quad \upsilon_0 = 10, \quad \theta_0 = \mathbf{x}_0, \quad \mathbf{\Lambda}_0 = (\upsilon_0 - 4)16^2\mathbf{I}, \tag{7}$$

where $\mathbf{I}$ is the three-dimensional identity matrix.

Integrating joint posterior density with respect to $\mathbf{\Sigma}$, we get the marginal posterior density for the mean

$$p(\mu \mid \mathbf{X}) \propto t_{\upsilon_t - 2}\left(\mu \mid \theta_t, \frac{\mathbf{\Lambda}_t}{\kappa_t(\upsilon_t - 2)}\right), \tag{8}$$

where $t_{\upsilon_t-2}$ is a multivariate $t$-distribution with $\upsilon_t - 2$ degrees of freedom.

We use the expectations of marginal posterior distributions for mean and covariance as our model parameters at

Table 1: Detection results.

| Sets | $T_{\text{all}}$ | $T_{\text{event}}$ | *Events* | TD | FA | $T_{\text{true}}$ | $T_{\text{miss}}$ | $T_{\text{false}}$ |
|---|---|---|---|---|---|---|---|---|
| AB-easy | 4850 | 2850 | 1 | 1 | 0 | 2220 | 630 | 0 |
| AB-medium | 4800 | 3000 | 1 | 1 | 1 | 1730 | 1270 | 970 |
| AB-hard | 5200 | 3400 | 1 | 1 | 1 | 2230 | 1170 | 350 |
| PV-medium | 3270 | 1920 | 1 | 1 | 0 | 1630 | 290 | 20 |
| PETS | 3000 | 1200 | 1 | 1 | 0 | 950 | 250 | 10 |
| ATC-1 | 6600 | 3400 | 6 | 6 | 0 | 2350 | 1100 | 50 |
| ATC-2 | 13500 | 6500 | 18 | 18 | 0 | 4740 | 1850 | 40 |
| ATC-3 | 5700 | 2400 | 5 | 5 | 0 | 1390 | 1010 | 0 |
| ATC-4 | 3700 | 2000 | 6 | 6 | 1 | 1300 | 700 | 350 |
| ATC-5 | 9500 | 5350 | 11 | 10 | 2 | 3160 | 2150 | 420 |

time $t$. Expectation for marginal posterior mean (expectation of multivariate $t$-distribution) becomes

$$\mu_t = E(\mu \mid \mathbf{X}) = \theta_t, \tag{9}$$

whereas expectation of marginal posterior covariance (expectation of inverse Wishart distribution) becomes

$$\mathbf{\Sigma}_t = E(\mathbf{\Sigma} \mid \mathbf{X}) = (v_t - 4)^{-1}\mathbf{\Lambda}_t. \tag{10}$$

Our confidence measure for the layer is equal to one over determinant of covariance of $\mu \mid \mathbf{X}$:

$$C = \frac{1}{|\mathbf{\Sigma}_{\mu|\mathbf{X}}|} = \frac{\kappa_t^3 (v_t - 2)^4}{(v_t - 4)\,|\mathbf{\Lambda}_t|}. \tag{11}$$

If our marginal posterior mean has larger variance, our model becomes less confident. Note that variance of multivariate $t$-distribution with scale matrix $\mathbf{\Sigma}$ and degrees of freedom $v$ are equal to $v/(v-2)\mathbf{\Sigma}$ for $v > 2$.

System can be further speeded up by making independence assumption on color channels. Update of full covariance matrix requires computation of nine parameters. Moreover, during distance computation, we need to invert the full covariance matrix. To speed up the system, we use three univariate Gaussians corresponding to each color channel. After updating each color channel independently, we join the variances and create a diagonal covariance matrix

$$\mathbf{\Sigma}_t = \begin{pmatrix} \sigma_{t,r}^2 & 0 & 0 \\ 0 & \sigma_{t,g}^2 & 0 \\ 0 & 0 & \sigma_{t,b}^2 \end{pmatrix}. \tag{12}$$

In this case, for each univariate Gaussian, we assume scaled inverse $\chi^2$-distribution for the variance and conditioned on the variance univariate normal distribution for the mean.

### 3.2. Background update

We initialize our system with $k$-layers for each pixel. Usually, we select three-five layers. In more dynamic scenes, more layers are required. As we observe new samples for each pixel, we update the parameters for our background model. We start our update mechanism from the most confident layer in our model. If the observed sample is inside the 99% confidence interval of the current model, parameters of the model are updated as explained in (6). Lower confidence models are not updated.

For background modeling, it is useful to have a forgetting mechanism so that the earlier observations have less effect on the model. Forgetting is performed by reducing the number of prior observation parameter of unmatched model. If current sample is not inside the confidence interval, we update the number of prior measurements parameter,

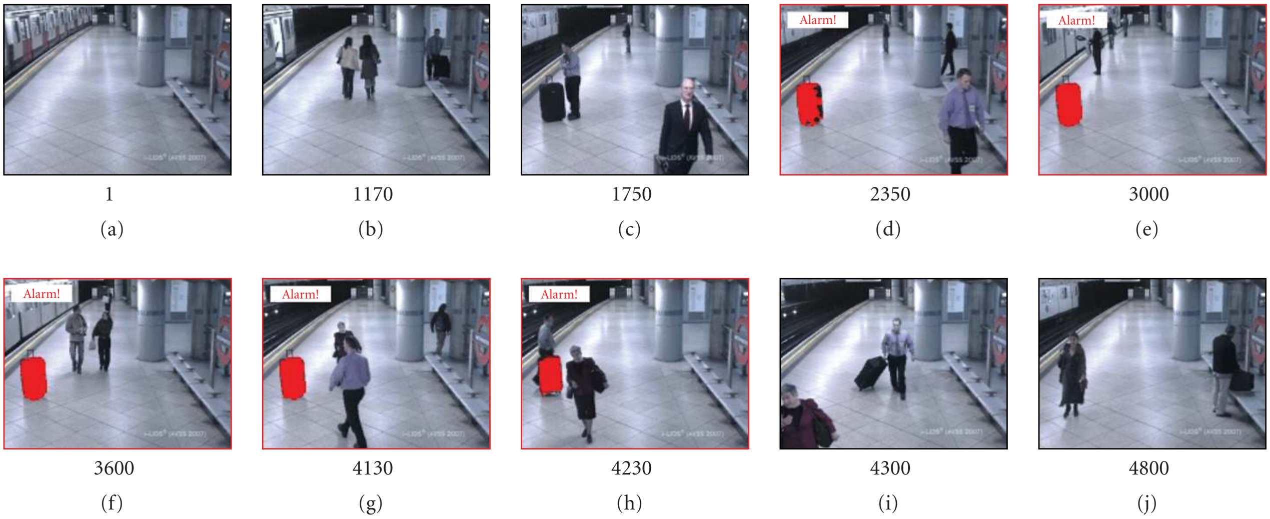
$$\kappa_t = \kappa_{t-1} - n, \tag{13}$$

and proceed with the update of next confident layer. We do not let $\kappa_t$ become less than initial value 10. If none of the models is updated, we delete the least confident layer and initialize a new model having current sample as the mean and an initial variance (7). The update algorithm for a single pixel can be summarized as shown in Algorithm 1

With this mechanism, we do not deform our models with noise or foreground pixels, but easily adapt to smooth intensity changes like lighting effects. Embedded confidence score determines the number of layers to be used and prevents unnecessary layers. During our tests, usually secondary layers correspond to shadowed form of the background pixel or different colors of the moving regions of the scene. If the scene is unimodal, confidence scores of layers other than first layer become very low.

### 3.3. Foreground segmentation

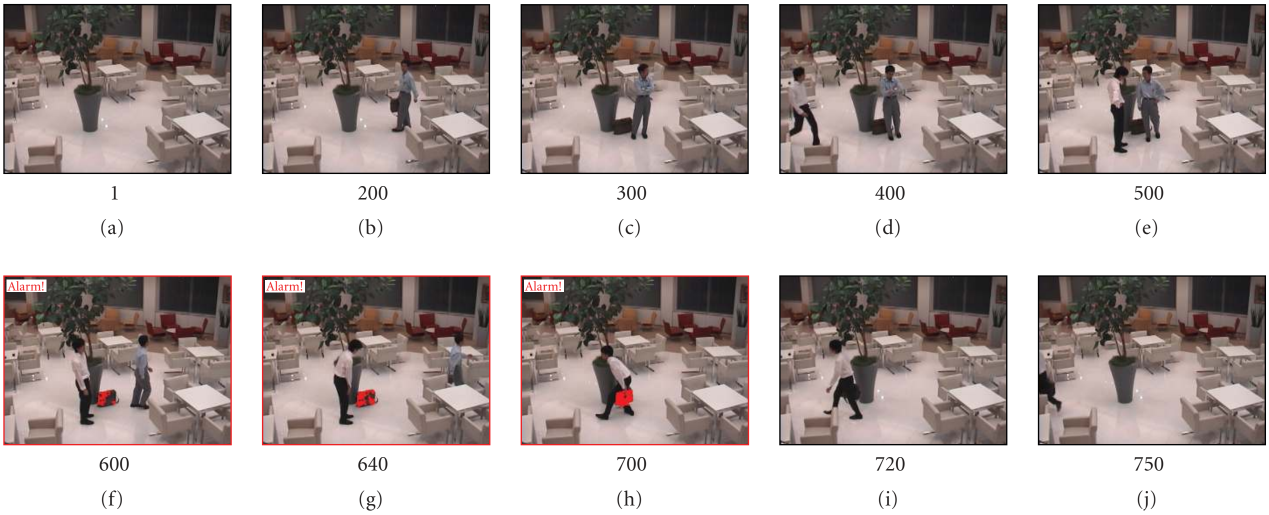
Learned background statistics are used to detect the changed regions of the scene. We determine how many layers are necessary for each pixel and use only those layers during foreground segmentation phase. The number of layers required to represent a pixel is not known beforehand, so background is initialized with more layers than needed. Usually, we select three to five layers. In more dynamic scenes, more layers are required. Using the confidence scores, we determine how many layers are significant for each pixel. As we observe new samples for each pixel, we update the parameters for our background model. At each update, at most one layer is updated with the current observation. This assures the minimum overlap over layers. We order the layers according to



1 (a) 1170 (b) 1750 (c) 2350 (d) 3000 (e)

3600 (f) 4130 (g) 4230 (h) 4300 (i) 4800 (j)

Figure 5: Test sequence AB-easy (*Courtesy of i-LIDS*). The alarm sets off immediately when the item is removed even though the luggage was stationary 2000 frames (image size is 180 × 144).



1 (a) 200 (b) 300 (c) 400 (d) 500 (e)

600 (f) 640 (g) 700 (h) 720 (i) 750 (j)

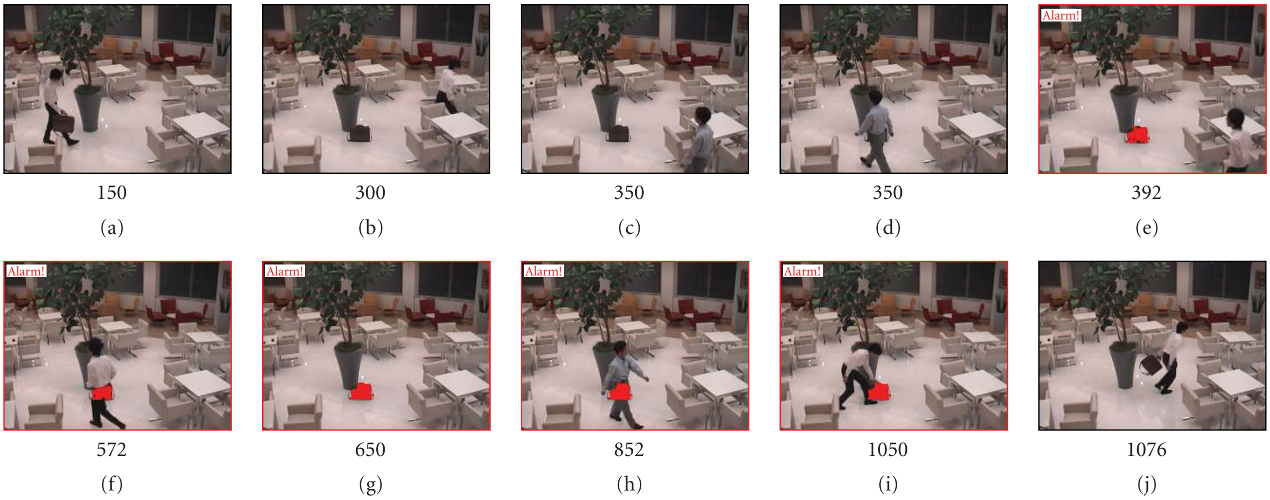
Figure 6: In sequence ATC-2.2 (*Courtesy of Advanced Technology Center, Amagasaki*), one person brings a bag, puts it on the ground, another person comes and picks it up. As visible, the object is detected accurately, and the alarm immediately sets off when the bag is removed.

confidence score and select the layers having confidence value greater than the layer threshold. We refer to these layers as confident layers. We start the update mechanism from the most confident layer. If the observed sample is inside the $2.5\sigma$ of the layer mean, which corresponds to 99% confidence interval of the current model, parameters of the model are updated. Lower confidence models are not updated.

## 4. EXPERIMENTAL RESULTS

To evaluate the dual foreground method, we used several public datasets from PETS 2006, i-LIDS 2007, and Advanced Technology Center. We tested a total of 32 sequences grouped into 10 sets. The videos have assorted resolutions; 180 × 144, 320 × 240, and 640 × 480. The scenarios ranged from lunch rooms to underground train stations. Half of these sequences depict scenes that are not crowded. Other sequences contain complex scenarios with multiple people sitting, standing, and walking at variable speeds. Some sequences show vehicles parked. The abandoned items are left in different durations from 10 seconds to 2 minutes. Some sequences contained small abandoned items. A few sequences have multiple abandoned items.

The sets AB-Easy, AB-Medium, and AB-Hard, which are included in i-LIDS challenge, are recorded in an underground train station. Set PETS is a large closed space platform with restaurants. Sets ATC-1 and ATC-2 are recorded from a wide angle camera of a cafeteria. Sets ATC-3 and ATC-4 are different cameras from a lunch room. Set ATC-5 is a waiting lounge. Since the proposed method is a pixelwise scheme, it is not difficult to set detection areas in the initialization time. We manually marked the platform in



150 (a)

300 (b)

350 (c)

350 (d)

392 (e)

572 (f)

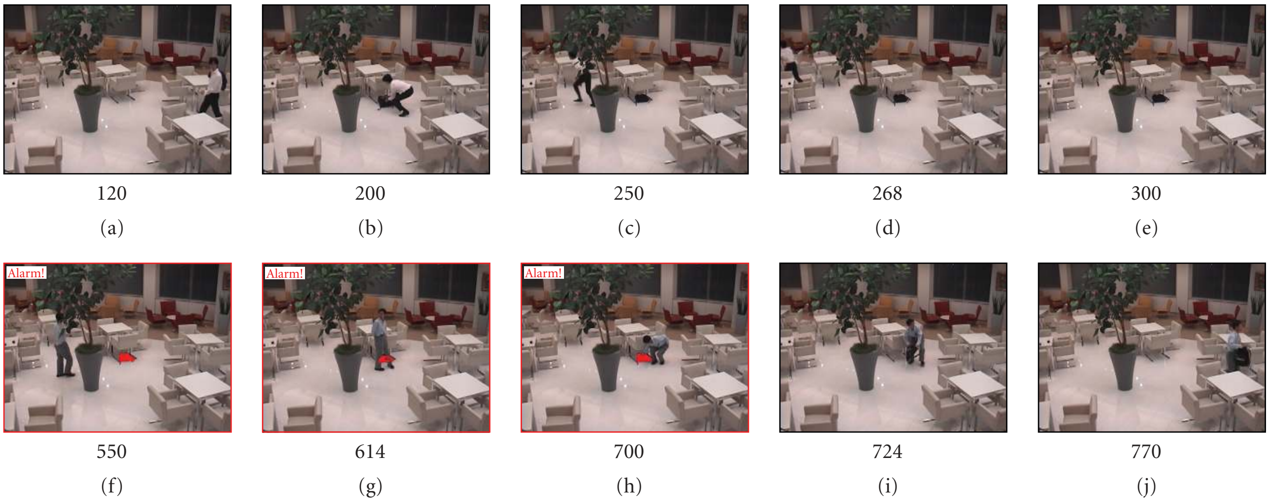
650 (g)

852 (h)

1050 (i)

1076 (j)

Figure 7: In sequence ATC-2.3 (*Courtesy of Advanced Technology Center, Amagasaki*), one person bring a bag, leaves it on the floor. As visible, after it was detected as an abandoned item, temporary occlusions due to the moving people do not cause the system to fail.



120 (a)

200 (b)

250 (c)

268 (d)

300 (e)

550 (f)

614 (g)

700 (h)

724 (i)

770 (j)

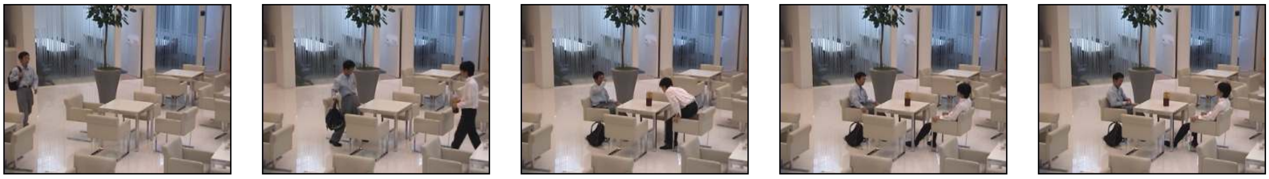
Figure 8: In sequence ATC-2.6 (*Courtesy of Advanced Technology Center, Amagasaki*), one person hides the bag under a shadowed area of the table and runs away. Another person comes, wanders around, takes the bag and leaves the scene.

AB-easy, AB-medium, and AB-hard sets, the waiting area in PETS 2006 set, and the illegal parking spots in PV-easy, PV-medium, and PV-hard sets. For the ATC sets, all of the image area is used as the detection area. For i-LIDS sets, we replaced the beginning parts of the video sequences with 4 frames of the empty platform.

For all results, we set the learning rate of the short-term background at 30 times the learning rate of the long-term background. We assigned the evidence threshold $\max_e$ in the range $[50, 500]$ depending on the desired responsiveness time that controls how soon an abandoned item is detected as an alarm. We used $k = 1$ as the decay parameter.

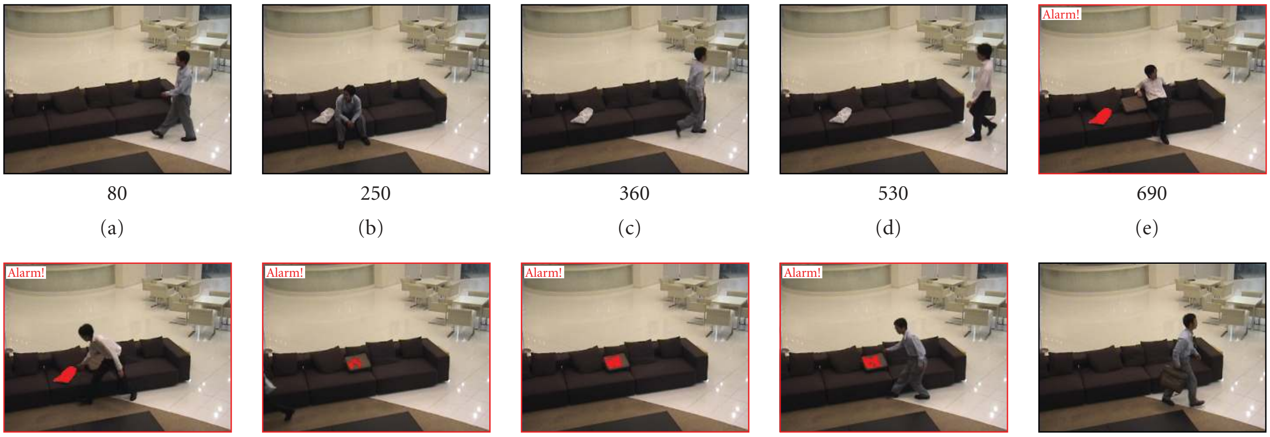
Figure 4 shows the detection results for the i-LIDS datasets. We reported the performance scores of all sets in Table 1, where $T_{\text{all}}$ is the total number of frames in a set and $T_{\text{event}}$ is the duration of the event in terms of the number of frames. We measure the duration right after an item has been left behind. It is also possible to measure the duration after the person moved away or after some preset waiting time in case additional tracking information is incorporated. *Events* indicates the number of abandoned objects (for PV-medium, the number of the illegally parked vehicles). TD means the correctly detected objects. A detection event is considered to be both spatially and temporally continuous. In other words, there might be multiple detections for a frame if the objects are spatially disconnected. FA shows the falsely detected objects. $T_{\text{true}}$ and $T_{\text{false}}$ are the duration of the correct and false detections. $T_{\text{miss}}$ is the duration that an abandoned item could not be detected. Since we start an event as soon as an object is left, this score does not consider any waiting time. This means that we overestimate our miss rate.

As our results show, we successfully detected almost all abandoned items while achieving a very low false alarm rate. Our method performed satisfactory when the initial frame


100 (a) 166 (b) 250 (c) 300 (d) 400 (e)

500 (f) 600 (g) 650 (h) 700 (i) 750 (j)

FIGURE 9: In sequence ATC-3.1 (*Courtesy of Advanced Technology Center, Amagasaki*), two people sit on a table. One person leaves a back bag, another a bottle. They leave both items behind when they depart.


80 (a) 250 (b) 360 (c) 530 (d) 690 (e)

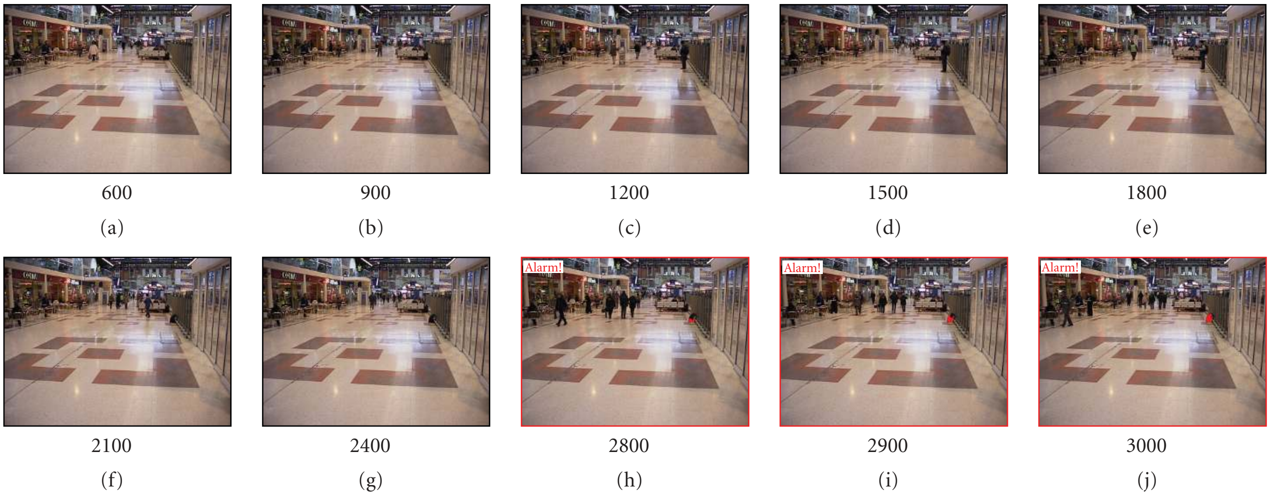
820 (f) 860 (g) 1000 (h) 1064 (i) 1118 (j)

FIGURE 10: In sequence ATC-5.3 (*Courtesy of Advanced Technology Center, Amagasaki*), one person sits on a couch and puts a bag next to him. After a while, he leaves but the bag stays on the couch. Another person comes, sits on the couch, puts his briefcase next to him, and takes away the bag. The briefcase is also removed later.

showed the actual static background. The detection areas have not included any people at the initialization time in the ATC sets, thus the uncontaminated backgrounds are easily learned. This is also true for the PV and AB-easy sets. However, the AB-medium and AB-hard sets contained several stationary people in the initial frames. This resulted in false detections when those people moved away. Since the background models eventually learn the statistically dominant color values, such false alarms should not occur in the long run due to the fact that the background will be more visible than the people. In other words, the ratio of the false alarms should decrease in time. We do not learn the color distribution of the abandoned items (or parked vehicles), thus the proposed method can detect them even if they are occluded. As long as the occluding object, for example, a passing by person, has different color than the long-term background, our method still shows the boundary of the abandoned item.

Representative detection results are given in Figures 5–12. As visible, none of the moving objects, moving shadows, people that are stationary in shorter durations was falsely detected. Besides, there are no *ghost* false detections due the inaccurate blending of the abandoned items in the long-term background. Thanks to the Bayesian update, the changing illumination conditions as in PV-medium are properly adapted in the backgrounds.

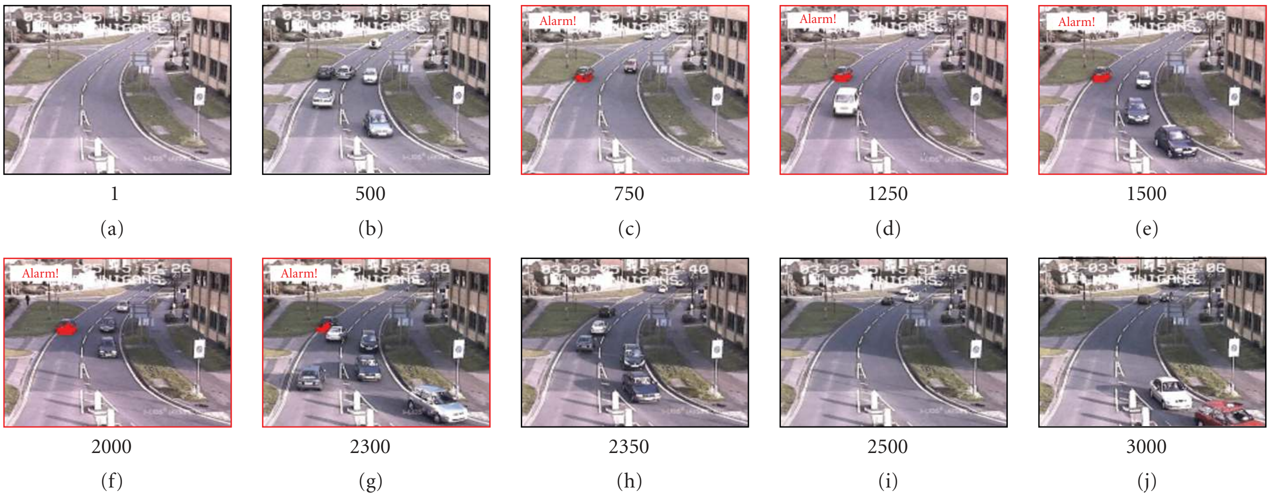
Another advantage of this method is that the alarm is immediately set of as soon as the abandoned item is removed from its previous position. Although we do not know whether the person who left the object is moved away from the object or not, we consider this property as a superiority over the tracking-based approaches that require a decision



600 (a) 900 (b) 1200 (c) 1500 (d) 1800 (e)

2100 (f) 2400 (g) 2800 (h) 2900 (i) 3000 (j)

Figure 11: A test sequence from PETS 2006 datasets (*Courtesy of PETS*). There is significant motion all around the scene. To make things more challenging, the person who leaves his back bag after stays still for an extended period of time.



1 (a) 500 (b) 750 (c) 1250 (d) 1500 (e)

2000 (f) 2300 (g) 2350 (h) 2500 (i) 3000 (j)

Figure 12: Test sequence PV-medium from AVSS 2007 (*Courtesy of i-LIDS*). A challenge in this video is the rapidly changing illumination conditions that cause dark shadows.

net of heuristic rules and context-depended priors to detect such event.

One shortcoming is that it cannot discriminate the different types of objects, for example, a person who is stationary for a long time can be detected as an abandoned item. This can be, however, an indication of another suspicious behavior as it is not common. To determine object types and reduce the false alarm rate, object classifiers, that is, a human or a vehicle detector, can be used. Since such classifiers are only for verification purposes, their computation time should be negligible. Since no tracking is integrated, trajectory-based semantics, for example, who left the item or how long the item left before the person moves away can not be extracted. Still, our method can be used as a preprocessing stage to improve the tracking-based video analytics.

The computational load of the proposed method is low. Since we only employ pixelwise operations and make pixelwise decisions, we can take advantage of the parallel processing architectures. By assigning each image pixel to a processor on the GPU using CUDA programming, since each processor can execute in parallel, the speed improves more than 14 × in comparison to the corresponding CPU implementation. For instance, full background update for 360 × 288 images takes 74.32 milliseceonds on CPU (P4 DualCore 3 GHz), however on CUDA, it only needs 6.38 milliseceonds. We observed that the detection can be comfortably employed in quarter spatial resolution by processing the short-term background at 5 fps while updating the long term at every 5 seconds (0.2 fps) with the same learning rates.

## 5. CONCLUSIONS

We present a robust method that uses dual foregrounds to find abandoned items, stopped objects, and illegally parked vehicles in static camera setups. At every frame, we adapt the dual background models using Bayesian update, and aggregate evidence obtained from dual foregrounds to achieve temporal consistency.

This method does not depend on object initialization and tracking of every single object, hence its performance is not upper bounded to these error prone tasks that usually fail for crowded scenes. It accurately outlines the boundary of items even if they are fully occluded. Since it executes pixelwise operations, it can be implemented on parallel processors.

## ACKNOWLEDGMENT

The authors thank their colleagues Jay Thornton and Keisuke Kojima for their constructive comments.

## REFERENCES

[1] J. D. Courtney, "Automatic video indexing via object motion analysis," *Pattern Recognition*, vol. 30, no. 4, pp. 607–625, 1997.

[2] S. Velastin and A. Davies, "Intelligent CCTV surveillance: advances and limitations," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, Wageningen, The Netherlands, August-September 2005.

[3] A. E. Cetin, M. B. Akhan, B. U. Toreyin, and A. Aksay, "Characterization of motion of moving objects in video," US patent no. 20040223652, 2004.

[4] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 260–267, New York, NY, USA, June 2006.

[5] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, and J. Meunier, "Left-luggage detection using homographies and simple heuristics," in *Proceedings of the 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS '06)*, pp. 51–58, New York, NY, USA, June 2006.

[6] J. Martínez-del-Rincón, J. E. Herrero-Jaraba, J. R. Gómez, and C. Orrite-Uruñuela, "Automatic left luggage detection and tracking using multi-camera UKF," in *Proceedings of the 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS '06)*, pp. 59–66, New York, NY, USA, June 2006.

[7] N. Krahnstoever, P. Tu, T. Sebastian, A. Perera, and R. Collins, "Multi-view detection and tracking of travelers and luggage in mass transit environments," in *Proceedings of the 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS '06)*, pp. 67–74, New York, NY, USA, June 2006.

[8] F. Lv, X. Song, B. Wu, V. K. Singh, and R. Nevatia, "Left luggage detection using bayesian inference," in *Proceedings of the 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS '06)*, pp. 83–90, New York, NY, USA, June 2006.

[9] K. Smith, P. Quelhas, and D. Gatica-Perez, "Detecting abandoned luggage items in a public space," in *Proceedings of the 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS '06)*, pp. 75–82, New York, NY, USA, June 2006.

[10] S. Guler and M. K. Farrow, "Abandoned object detection in crowded places," in *Proceedings of the 9th IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS '06)*, pp. 99–106, New York, NY, USA, June 2006.

[11] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, vol. 2, pp. 246–252, Fort Collins, Colo, USA, June 1999.

[12] F. Porikli and O. Tuzel, "Bayesian background modeling for foreground detection," in *Proceedings of the 3rd ACM International Workshop on Video Surveillance & Sensor Networks (VSSN '05)*, pp. 55–58, Singapore, November 2005.

[13] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proceedings of the 17th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 1, pp. 255–261, Kerkyra, Greece, September 1999.

[14] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proceedings of the Workshop on Motion and Video Computing (MOTION '02)*, pp. 22–27, Orlando, Fla, USA, December 2002.

[15] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 302–309, Washington, DC, USA, June-July 2004.

[16] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th European Conference on Computer Vision-Part II (ECCV '00)*, vol. 2, pp. 751–767, Dublin, Ireland, June-July 2000.