Abandoned Object Detection via Temporal Consistency Modeling and Back-Tracing Verification for Visual Surveillance

Kevin Lin, Shen-Chi Chen, Chu-Song Chen, Daw-Tung Lin, *Senior Member, IEEE*, and Yi-Ping Hung

***Abstract*— This paper presents an effective approach for detecting abandoned luggage in surveillance videos. We combine short- and long-term background models to extract foreground objects, where each pixel in an input image is classified as a 2-bit code. Subsequently, we introduce a framework to identify static foreground regions based on the temporal transition of code patterns, and to determine whether the candidate regions contain abandoned objects by analyzing the back-traced trajectories of luggage owners. The experimental results obtained based on video images from 2006 Performance Evaluation of Tracking and Surveillance and 2007 Advanced Video and Signal-based Surveillance databases show that the proposed approach is effective for detecting abandoned luggage, and that it outperforms previous methods.**

***Index Terms*— Abandoned luggage detection, abandoned object detection, short-term background model, long-term background model, object detection and tracking, visual surveillance.**

I. Introduction

IN THE visual surveillance research, detecting abandoned luggage is referred to as the problem of abandoned-object or left-luggage detection. It is a crucial task for public security, particularly for identifying suspicious stationary items. Because there is no object type of category that can be assumed as having been abandoned, common object detection methods such as training an object detector for a particular category of objects are inappropriate for performing this task. To address this problem, foreground/background extraction techniques are suitable for identifying static foregrounds regions (i.e., objects that remain static for a long time) as left-luggage candidates.

Manuscript received May 30, 2014; revised December 23, 2014; accepted February 12, 2015. Date of publication March 2, 2015; date of current version May 15, 2015. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 103-2221-E-305-008-MY2 and Grant MOST 103-2221-E-001-010 and in part by Taiwan Secom Company, Ltd. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Liang Wang.

K. Lin is with the Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan (e-mail: r01944012@csie.ntu.edu.tw).

S.-C. Chen is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: d98922030@csie.ntu.edu.tw).

C.-S. Chen is with the Research Center for Information Technology Innovation & Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan (e-mail: song@iis.sinica.edu.tw).

D.-T. Lin is with the Department of Computer Science and Information Engineering, National Taipei University, Taipei 23741, Taiwan (e-mail: dalton@mail.ntpu.edu.tw).

Y.-P. Hung is with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan (e-mail: hung@csie.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2015.2408263

A. Related Works

The algorithms for identifying a static foreground or abandoned object can be classified into three categories. The first category involves constructing double-background models for detecting a static foreground [1]–[3]. The double-background models are constructed using fast and slow learning rates. Subsequently, the static foreground is localized by differentiating between the two obtained foregrounds. A weakness of these methods is the high false alarm rate, which is typically caused by imperfect background subtraction resulting from a ghost effect, stationary people, and crowded scenes. In addition, these methods involve using only the foreground information per single image to locate regions of interest (ROIs) of abandoned-object candidates. Consequently, temporally-consistent information that may be useful for identifying sequential patterns of ROIs may be overlooked.

The second category of methods for extracting static foreground regions involves using a specialized mixture of Gaussian (MOG) background model. In previous researches [4]–[6], three Gaussian mixtures were used to classify foreground objects as moving foreground, abandoned objects, and removed objects by performing background subtraction. In addition, the approach proposed in [6] uses visual attributes and a ranking function to characterize various types of alarm events.

The third category involves accumulating a period of binary foreground images or tracking foreground regions to identify a static foreground. The methods proposed in [7] and [8] involved localizing the static foreground based on the pixels with the maximal accumulated values, which were subsequently considered the candidate regions of stationary objects. However, this category of methods fails in complex scenes.

LV *et al.* [9] used a blob tracker to track foreground objects based on their size, aspect ratio, and location. Left luggage is identified when a moving foreground blob stops moving for a long period. Li *et al.* [10] tracked moving objects by incorporating principle color representation (PCR) into a template-matching scheme, and also by estimating the status (e.g., occluded or removed) of a stationary object.

Rather than using a single camera, some approaches use multiple cameras for detecting abandoned luggage. Auvinet *et al.* [11] employed two cameras for detecting abandoned objects, and the planer homography between

two cameras was used to regulate the foreground tracking results.

To fulfill the semantic requirement of abandoned luggage events where a person drops their luggage and then leaves, some of the aforementioned methods combine a tracker to track the involved person(s) for further verification. Liao *et al.* [7] tracked luggage owners based on skin color information and by performing contour matching with a Hough transform. In [1], Kalman filter (KF) and unscented KF (UKF) were used to track foreground objects (including people and carried luggage) based on low-level features, such as color, contour, and trajectory. Tian *et al.* [4] integrated a human detector and blob tracker to track the owner of abandoned luggage, and the corresponding trajectory was recorded for further analysis. Fan *et al.* [6] used a blob tracker to track moving people close to the left-luggage. The obtained movement information was used as an input for their attribute-based alert ranking function.

B. Our Approach

In this paper, we propose a temporal dual-rate foreground integration method for static-foreground estimation for single-camera video images. Our approach involves constructing both short- and long-term background models learned from an input surveillance video on-line. Subsequently, we introduce a simple pixel-based finite-state machine (PFSM) model that uses temporal transition information to identify the static foreground based on the sequence pattern of each object pixel.

Because the proposed approach involves using temporal transition information, we can reduce the influence of imperfect foreground extractions in the double-background models, thereby improving the accuracy of the constructed static foreground inference. An owner-tracking procedure is also employed in our method to semantically verify the abandoned-object event. Contributoins of the proposed method over previous methods are summarized as follows.

1) We introduce a dual-rate background modeling framework with temporal consistency. It performs considerably better than the single-image-based double background models in [1]–[3].
2) We develop a simple spatial-temporal tracking method for back-tracing verification. Compared to the frame-by-frame tracking approaches such as the KF- or UKF employed in [1], our approach is superior in handling temporary occlusions and is still highly efficient to implement.
3) Experimental results on benchmark datasets (PETS2006 and AVSS2007) show that our method performs more favorably against all of the compared methods [1]–[8].

The remainder of this paper is organized as follows. Section II details the proposed algorithm, Section III shows the experimental results, and finally, our conclusion and discussion are offered in Section IV.

II. Temporal Dual-Rates Foreground Integration Method

The proposed abandoned-object detection method is based on background modeling and subtraction. The following subsection provides a conceptual review of background-subtraction and the associated learning rates for updating a background model. Subsequently, the remaining subsections introduce our algorithm for identifying static foreground regions.

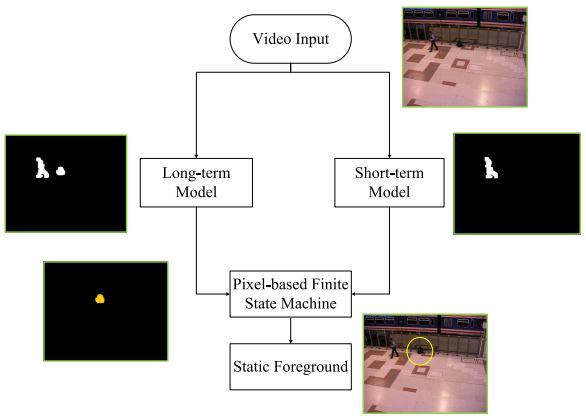


Fig. 1. Flowchart of static foreground detection.

A. Review of Background Modeling and Learning Rates

Background subtraction is an essential technique for detecting moving objects in surveillance systems. To apply this technique, a pixel-based background model is typically learned from preceding images. The learned background model is used to identify whether each pixel of the incoming image is a background pixel. When a pixel in an incoming image is identified as a background pixel, the associated features (e.g., pixel color) can subsequently be used to update the background model to more suitably represent the recently observed pixel values. Given a sequence of images I_t $(t \in N)$ of size $m \times n$, the principle of a general background modeling and updating procedure can be summarized as follows:

1) Initialize a background model $\mathbf{B}(x, y)$ for each pixel (x, y), $0 \leq x \leq m-1$, and $0 \leq y \leq n-1$.
2) For every pixel (x, y) of the incoming image I_t, if $I_t(x, y) \in \mathbf{B}(x, y)$, then (x, y) is classified as a background pixel, otherwise it is considered a foreground pixel.
3) For every newly identified background pixel (x, y), update $\mathbf{B}(x, y)$ by considering the new training sample, $I_t(x, y)$.
4) $t \leftarrow t+1$, go to Step 2).

In this procedure, a learning rate $\lambda \in [0, 1]$ is typically applied to update the background in Step 3). The learning rate provides a tradeoff between $\lambda\mathbf{B}$ and $(1-\lambda)I_t$, and thus the preceding model $\mathbf{B}$ is tuned toward the new training data I_t faster when λ is smaller in the incremental updating. For example, in the MOG method proposed in [12], the background model $\mathbf{B}(x, y)$ is recorded as an mixture-of-Gaussian distribution in RGB color space. The learning rate λ is applied to update the mixture-distribution model when the new color $I_t(x, y)$ is observed and (x, y) is identified as a background pixel. Similar updating mechanisms exist in other methods such as Codebook [13], enhanced Gaussian mixture model (EGMM) algorithm [14], and coarse-to-fine approach [15].

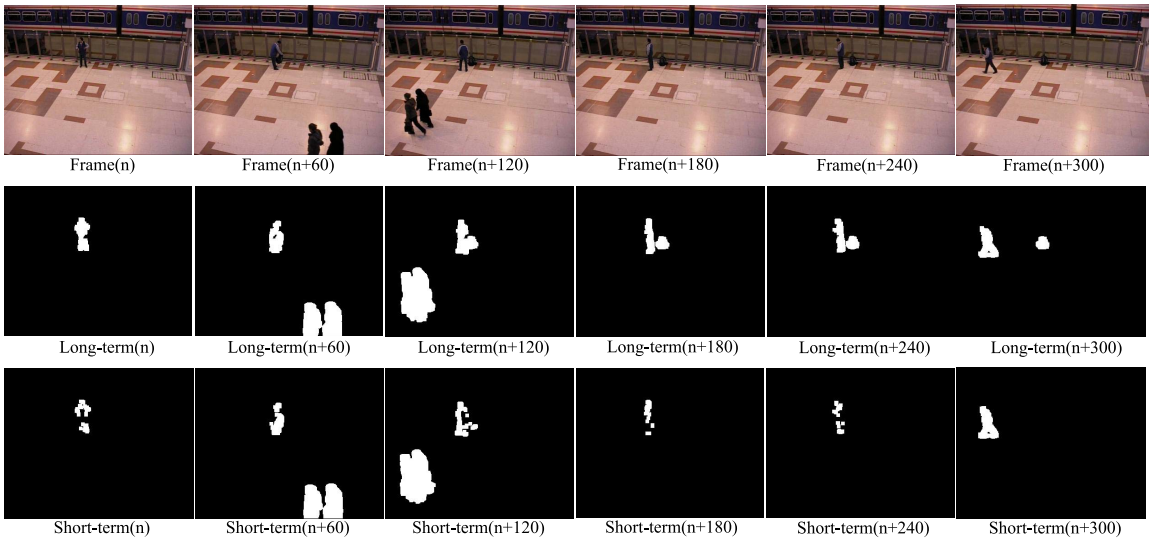


Fig. 2. Background subtraction results of PETS2006-S1 video sequence.

B. Long-Term and Short-Term Integration Background Modeling

Figure 1 shows an overview of the integrated background modeling method proposed in this study. First, we describe the long- and short-term models built in our approach for static foreground detection. The proposed algorithm starts from a generic background modeling method operated at two learning rates. Without loss of generality, we select the MOG method in [12] as our background modeling method; however, other methods equipped with learning-rate mechanisms for updating background models can be used in our framework as well.

As aforementioned, a small learning rate λ_S updates the background model at a faster speed. The model that learns at this small rate is called the *short-term background model* $\mathbf{B}_S$, where F_S denotes the binary foreground image obtained via the short-term model. By contrast, a large learning rate λ_L yields the model that is updated at a slower speed. Similarly, the model that learns at this rate is referred to as the *long-term background model* $\mathbf{B}_L$, where F_L denotes the binary foreground image obtained using the long-term model. Figure 2 shows an example of the foreground regions obtained using the long- and short-term background models.

The assembly of long- and short-term background models is suitable for detecting stationary objects. Figure 3 shows an example of an abandoned-object event. Whenever luggage is left by an owner, the long-term model detects it as a foreground object, as shown in Figure 3(c). Moreover, because of the faster updating rate, the left-luggage would be classified as a background object by the short-term model, as shown in Figure 3(d). Accordingly, a pixel is represented as a two-bit code S_i by concatenating the detected long- and short-term foregrounds, as follows:

$$S_i = F_L(i)F_S(i), \tag{1}$$

where $F_L(i)$, and $F_S(i) \in \{0, 1\}$ represent the binary values of pixel i of the foreground images.

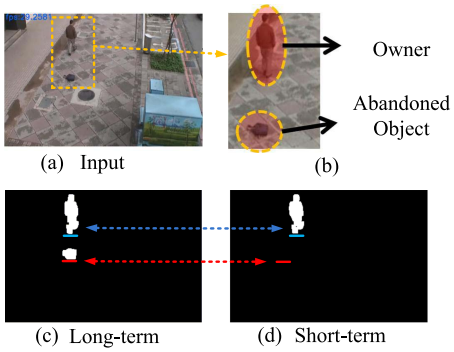


Fig. 3. An example of object abandoned event, where the combination of long-term and short-term foreground results is well suited for abandoned luggage detection.

TABLE I
PIXEL CLASSIFICATION FROM LONG-TERM AND SHORT-TERM BACKGROUND MODEL [2]

S_i	Hypotheses of the pixel i
00	Background
01	Uncovered background
10	Candidate static foreground
11	Moving foreground

Therefore, there are four states represented by the two-bit code S_i, as shown in Table I, and they are expressed as follows:

- $S_i = 00$ indicates that pixel i is a background pixel because it is classified as background by both $\mathbf{B}_L$ and $\mathbf{B}_S$.
- $S_i = 01$ implies that pixel i is an uncovered background pixel that has been temporarily occluded by an object and then exposed in a recent image.
- $S_i = 10$ indicates that pixel i is likely to be a static foreground pixel.
- $S_i = 11$ indicates that pixel i corresponds to a moving object.

When detecting abandoned objects, we are primarily concerned which pixels exhibiting a state value of 10, because

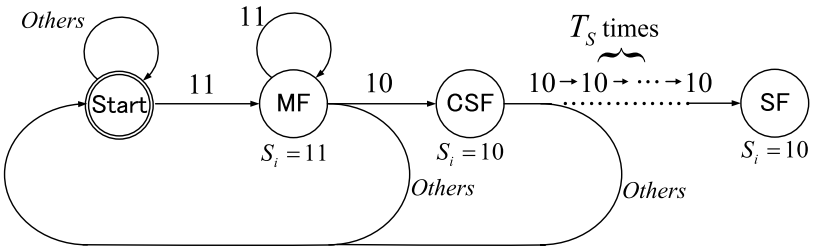


Fig. 4. PFSM for static foreground detection. MF is moving foreground, CSF denotes candidate static foreground and SF represents static foreground. T_s is the transition time for changing state from 10 to 10.

1) these are foreground pixels that have existed for a long time, as indicated by their long-term presence under the long-term model, and 2) they have not moved or vibrated for a considerable period of time; thus, the short-term model is expected to reject it soon. These properties confine the aforementioned static foreground pixel and make the state codes suitable for identifying abandoned object candidates.

However, these codes are defined for a single image only. Because noise could result from imperfect background modeling, these codes could be temporary or imprecise. Hence, the pixel classifications in Table I for single images are typically insufficient for identifying abandoned objects in an uncertain environments, which is why methods for a single or isolated images, such as that proposed in [2], are unreliable and frequently fail in practical cases.

In this paper, we propose using temporal-continuity information to improve the performance. We assert that the code pattern in an image sequence should primarily follow a temporal rule, and that the rule is representable by a very simple finite-state machine (FSM) model. Details are given in the next section.

C. Pixel-Based Finite State Machine (PFSM)

Instead of recognizing the status of each pixel based on only a single frame, we use temporal transition information to identify the stationary objects based on the sequential pattern of each pixel. A pixel is associated with only one state at a time. Based on long- and short-term background models, the state of pixel i can be changed from one state at time t to another state at time $t + 1$. Accordingly, we construct a simple FSM model to describe the behavior of each pixel. We detect the static foreground by identifying a specific pattern of transitions. Figure 4 illustrates the particular transition for identifying the static foreground.

As shown in Figure 4, the transition pattern describes the static foreground in an object-abandoned event. Starting from an initial state, the system is triggered by $S_i = 11$, indicating that pixel i is currently occluded by a foreground region. Hereafter, when a person abandons their luggage, the short-term method soon updates the luggage into its background model, whereas the long-term method does not; thus, the status of this site is changes to $S_i = 10$. Finally, when the status of $S_i = 10$ persists for a certain duration of time (i.e., for T_s times), we then conjecture that pixel i has become a part of the static foreground. During this procedure, only those pixels associated with this particular transition pattern are considered static foreground pixels. Otherwise, the state of pixel i would return to the initial state and restart until the initial state $S_i = 11$. The PFSM model thus describes the following rule: given a two-bit code sequence, if there is consecutive subsequence starting by a series of 11 and followed by a sufficiently long series of 10, then this subsequence is a detection of the static foreground.

For each frame, those pixels accepted by the PFSM model are collected. Subsequently, we perform a connected component analysis to group those pixels and remove the small components. If no pixel is accepted by the PFSM model, or if all of the components in the current frame are too small, no further verification is performed. Otherwise, the preserved components (i.e., the static foreground pixels) are considered the abandoned luggage candidates in the current frame, and they are sent to the subsequent stage for further verification by using the back-tracing algorithm, as detailed in the following section.

D. Back-Tracing Verification

Next, we verify whether the luggage is abandoned or simply placed on the ground for a short time by using the back-tracing verification procedure. Accordingly, our system first verifies whether the luggage owner is close to the luggage. If the owner does not return to his or her luggage, the object is considered abandoned. To perform the aforementioned semantic analysis of the object-abandoned event, the back-tracing verification is performed as follows.

The static foregrounds found in Section II-C are subsequently considered luggage candidates. When a static foreground is deemed a left-luggage candidate at time t and no other moving foreground objects are within its neighbor region of radius D, we then return from the current frame t to the preceding frame $t_0 = t - T_s$, which denotes the moment that the owner has likely put down the luggage, where T_s is the transition-time constant employed in our PFSM model (Figure 4). Let the image position of the left luggage candidate be p at time t_0. Centered at p, we create a spatial-temporal window $\boldsymbol{W_0}$ of size (r^2, δ), where r specifies the radius of a circle centered at p, and δ denotes the time interval $[t_0, t_0 + \delta]$.

Subsequently, for window $\boldsymbol{W_0}$, we consider all foreground blobs identified using the background subtraction algorithm. From these blobs, we then select the one that is approximate to the shape of human by using the height/width estimator

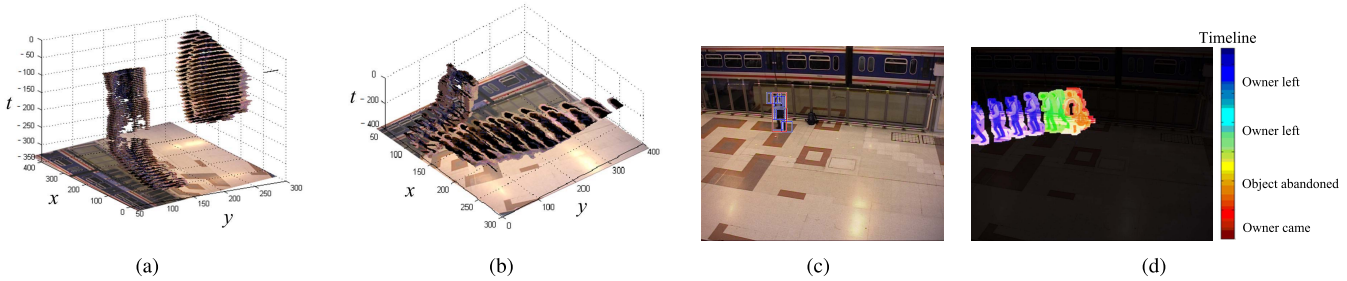


Fig. 5. (a) and (b) The trajectory construction of PETS2006-S1 in two different views. (c) Pedestrian detection result. (d) Result of Back-tracing verification.

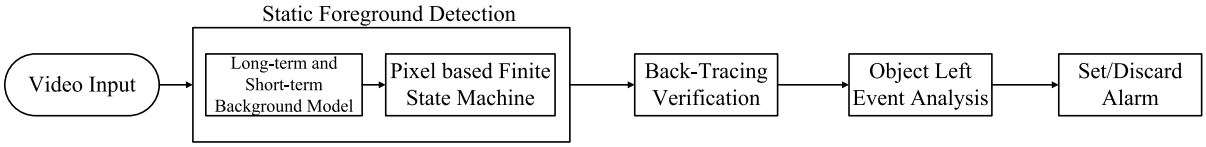


Fig. 6. The proposed system diagram.

in [1] and the human detector in [16] and [17], which filter the static foreground objects that could be human.

We give a brief review of the human detection method below. The deformable part-based model (DPM) detector [16] is one of the state-of-the-art human detection algorithms, which employs the sliding window technique with multiple filter kernels to detect the object in an image. The object to be detected is represented using a root filter and several part filters. The root filter describes the overall appearance of the object while the part filters depict partial regions of the object. The object is located when the region is voted with the highest scores by root and part filters. However, due to the number of filters adopted, the computation cost becomes extremely high. To overcome this difficulty, Dubout and Fleuret [17] approximate the sliding window technique as a convolutional procedure. According to the theorem that time-domain convolution is equivalent to frequency-domain multiplication, the part-based human detector is speeded up by fast Fourier transform and is employed in this work.

The foreground region containing human is then treated as the owner blob for further tracking, and we denote its image position as p_1. If there are more than one humans detected, we simply choose the blob closest to p as the most-fit blob position p_1.

We extract the color distribution as a feature representation of the foreground blobs. Next, centered at p_1, we creat a new spatial-temporal window $\boldsymbol{W_1}$ of the size (r^2, δ). We then employ the Bhattacharyya coefficient to identify the blob with the color distribution most similar to that of the owner in $\boldsymbol{W_1}$, and then create a window $\boldsymbol{W_2}$ centered at the newly identified blob. The aforementioned procedure is then used to track the blob representing the owner until the time exceed the original time t or until the tracked blob is outside of the neighbor region (i.e., within radius D) centered at the candidate luggage.

An advantage of the aforementioned procedure is that the time interval δ is used in the spatial-temporal domain, and hence it can track the target when occlusion occurs within δ. Thus, unlike the frame-by-frame tracking approaches, such as the KF- or UKF-based approaches employed in previous studies of left-luggage detection [1], our approach is more powerful for handling temporary occlusions, and it is still highly efficient to implement because only the foreground blobs within a limited number of spatial-temporal windows are considered.

Figure 5 demonstrates our back-tracing result of the first sequence. Figure 5(a) and Figure 5(b) show the 3D trajectory constructed based on our spatial-temporal structure. The back-tracing algorithm initiates the search for the owner from the location of the luggage, and then proceeds examining similar foreground patches. Figure 5(c) shows the pedestrian detection result. Figure 5(d) shows a summary of the object-abandoned event in the first sequence. The regions denoting the owner are displayed sequentially with rainbow colors depicting various time stamps of the event.

Our tracking procedure is extendable for preserving multiple hypotheses of blobs tracked simultaneously when employing a probabilistic framework such as particle filtering (PF) to represent the multiple hypotheses for dynamic tracking. However, PF is slow and cannot fulfill the real-time verification requirements of most visual surveillance applications. Hence, we use the aforementioned single-hypothesis approach, which can be generalized for more effective tracking when necessary.

E. Abandoned Object Event Analysis

Figure 6 shows the proposed system architecture. Once the trajectory of owner is obtained, a warning is issued that the luggage has been abandoned in accordance with the following two rules, as defined by PETS2006 [18].

1) Temporal rule: The luggage is declared an unattended object when it is left by its owner, and the luggage is not reattended within time $\boldsymbol{T}=$ 30 seconds.
2) Spatial rule: The unattended luggage is declared an abandoned object when it is left by its owner. When the distance between the owner and luggage is greater than a predefined distance $\boldsymbol{D}=$ 3 m, then an alarm event is triggered.

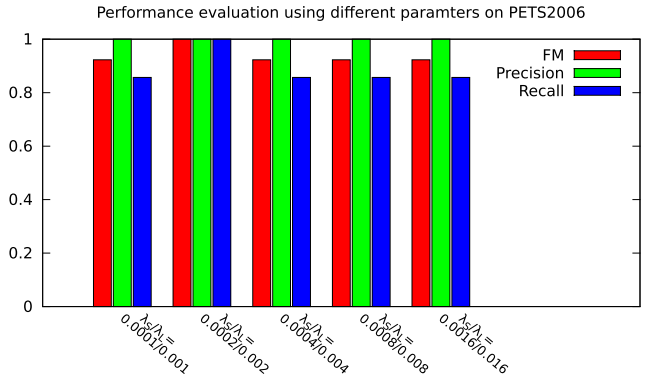


Fig. 7. Performance evaluation using different parameters on PETS2006. We restricted the learning rates λ_S and λ_L in a fixed ratio $\lambda_S/\lambda_L = 1/10$. The configuration, $\lambda_S = 0.0002$ and $\lambda_L = 0.002$, demonstrates a more favorable performance. Red, green, and blue bar represent F-measure, precision, and recall, respectively.

According to the PFSM, the temporal rule is satisfied by letting $T_s = 30f$ frames, where f is the frames per second (fps) at which the video is captured. The spatial rule is verified by examining the trajectory of owner. We create a luggage-centered ROI with a radius of $D = 3\mu$ pixels (where μ denotes the scaling factor to convert pixels into real-world distances), and investigate whether the owner is within and then left the ROI. An alarm is raised when both the spatial and temporal rules are satisfied.

III. Experimental Results

A. Implementation Details

The proposed system was developed using the programming language C/C++. The overall computation speed is 29 fps when testing the video of 360×240 pixel video by using a general purpose laptop with a 2.4 GHz Intel Core-i7 processor.

Various previous studies have proposed background subtraction algorithms, including MOG [12], Codebook [13], EGMM [14], and coarse-to-fine approach [15]. EGMM, which is available in the OpenCV library, is used in this work because of its high performance.

In this study, the long- and short-term background models is constructed using EGMM, which is similar to MOG with an additional mechanism for adapting the number of Gaussian components for each pixel, instead of using a fixed number of Gaussian components for every computation. To satisfy the characteristics of dual-background models, the learning rate of each background model should differ significantly. Based on our empirical study, we restricted the learning rates λ_S and λ_L in a fixed ratio $\lambda_S/\lambda_L = 1/10$, and find that the short- and long-term models can be distinguished well in practice.

First, we perform a preliminary experiment on the PETS2006 dataset by varying λ_S and λ_L, and evaluate the performance of the abandoned object detection. Figure 7 shows that when λ_S varies from 0.0001 to 0.0016 (and λ_L varies from 0.001 to 0.016, respectively), the precision value remains the same while the recall value becomes different, where FM is the F-measure values [19] that can be expressed as a harmonic mean between precision P and recall R with $FM = \frac{2\times P\times R}{P+R}$.

TABLE II
Performance Comparison on PETS2006 Video Dataset

Video	Difficulty	[2]	[1]	[7]	[10]	[4]	[11]	Ours
S1	*	T	T	T	T	T	T	T
S2	***	N/A	T	T	F	T	T	T
S3	*	N/A	T	N/A	T	F	T	T
S4	****	N/A	N/A	T	T	T	T	T
S5	**	N/A	N/A	T	F	T	T	T
S6	***	N/A	N/A	T	T	T	T	T
S7	*****	N/A	T	T	T	T	T	T

This reveals that our method is highly stable in the precision, i.e., the abandoned objects detected are correct, but could miss to detect some of the abandoned objects because of the non-perfect recall rate for the PETS2006 dataset when different parameters are selected. Among them, $\lambda_S = 0.0002$ and $\lambda_L = 0.002$ perform more favorable against the others. Hence, we choose this setting and use the identical parameter values for all of the experiments conducted by this study, including the experiments for the datasets of AVSS2007 and ABODA (Section III.D).

In addition, as the goal is to detect the abandoned object, considering only the region-of-interest area is a natural way to reduce imperfect background initialization. We follow the previous studies (such as [2]) that manually marked the train station platform in AVSS2007 and the waiting area in PETS2006 for abandoned object detection.

B. Results on PETS2006 and AVSS2007

We conducted experiments using the public datasets PETS2006 [18] and AVSS2007 [20].

1) PETS2006: The PETS2006 dataset comprises seven sequences of various scenarios. Each sequence includes an abandoning event except the third one. In the third sequence, a person puts down his bag for a short time; because the owner dose not abandon the luggage, no alarm should be triggered. Table II compares the results of our approach with those obtained by several other state-of-the-art studies [1], [2], [4], [7], [8], [10], [11]. Some previous studies have evaluated their methods by selecting a limited number of sequences and showing that their methods achieve high accuracy for those sequences only. By contrast, we have evaluated all seven sequences, and our method successfully detects the luggage-left events for the entire dataset of PETS2006 without triggering any false alarms. Figures 8 and 9 show the results of the 7th and 5th sequences of PETS2006, respectively.

Table III shows further evaluations of the precision-recall of the proposed algorithm. The compared approaches are sorted in order of their corresponding F-measure values. The method in [11] accurately detects all abandoned objects, as shown in Table II. However, their method results in several false alarms in Sequence 5 and 7; consequently, their F-measures are lower than those of the other methods. Sequence 5 and 7 are challenging to solve because of the problems with crowded scenes and occlusion. However, our temporal consistency model robustly and successfully localizes the abandoned luggage. Furthermore, our back-tracing

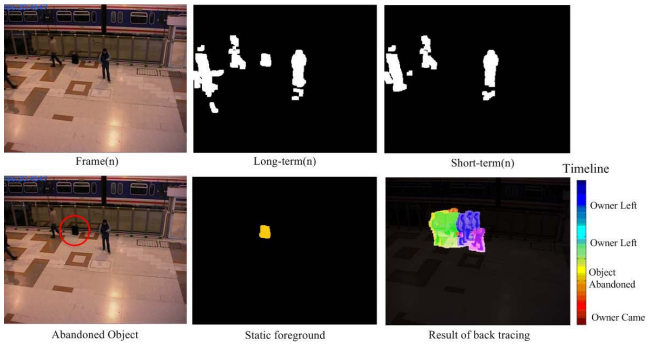


Fig. 8. Detection results of the 7th sequence of PETS2006.

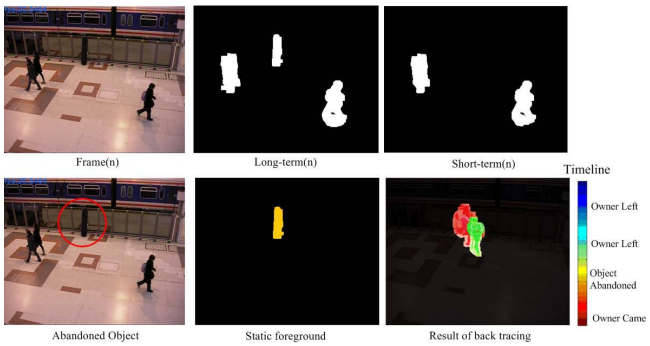


Fig. 9. Detection results of the 5th sequence of PETS2006.

TABLE III
COMPARISON OF DIFFERENT METHODS ON PETS2006 VIDEO DATASET

-	[2]	[11]	[10]	[7]	[6]	[4]	Ours
Precision	0.03	0.58	1.0	0.75	0.95	0.85	1.0
Recall	1.0	1.0	0.71	1.0	0.80	1.0	1.0
F-measure	0.05	0.73	0.83	0.86	0.87	0.92	1.0

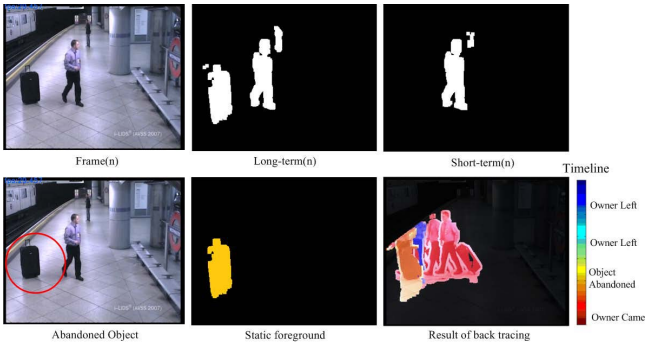


Fig. 10. Detection results of the sequence AB-Easy of AVSS2007.

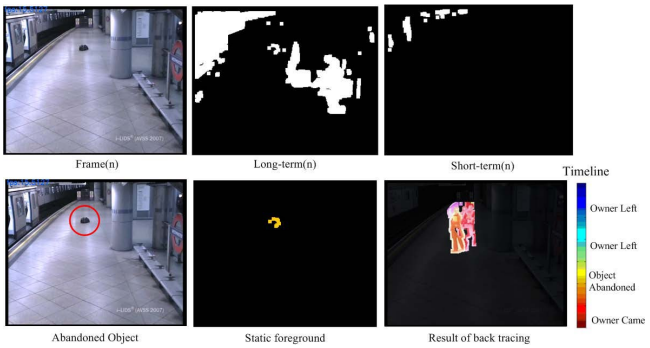


Fig. 11. Detection results of the sequence AB-Medium of AVSS2007.

TABLE IV
COMPARISON OF DIFFERENT METHODS ON AVSS2007 VIDEO DATASET

-	[2]	[3]	[5]	[4]	[6]	[7]	[8]	Ours
Precision	0.05	0.21	0.40	0.35	0.97	1.0	1.0	1.0
Recall	1.0	1.0	0.67	1.0	1.0	1.0	1.0	1.0
F-measure	0.09	0.35	0.50	0.52	0.98	1.0	1.0	1.0

method performs adequately and raises the alarm in a timely manner.

2) AVSS2007: We also tested our system by using the AVSS2007 dataset. The dataset was obtained from the i-LIDS video library, which includes several scenarios, such as abandoned luggage and parked vehicles. We evaluated the left-baggage scenario to fit the scope of this study. The abandoned-luggage dataset comprises three sequences (AB-EASY, AB-MEDIUM and AB-HARD) that are labeled with various difficulty levels according to the luggage size and crowd density. Each sequence contains only one abandoned-luggage event, similar to the PETS2006 dataset. We followed the detection rules provided by i-LIDS, which stipulates that the detection area is restricted to the platform of the train station. Some detection results of AVSS2007 are shown in Figures 10 and 11.

For the sake of comparison, Table IV shows the precision-recall of our method and those reported by other state-of-the-art studies [2]–[8]. The luggage-left event is easily detected because of the large size of luggage and less occlusion in AB-EASY. By contrast, AB-MEDIUM and AB-HARD are more difficult because they involve scenes with small pieces of luggage and dense crowds. Because of the luggage was temporarily occluded, several methods yielded false alarms, and they were thus considered less promising. Noteworthily, our method localize the abandoned objects in all three sequences, as shown in Table IV. The table also shows that [7] and [8] outperform several related works, as dose our method. However, [7] results in several false alarms when testing the PETS2006 dataset, and the method in [8] is evaluated using the AVSS2007 dataset only, which is considered limited in the context of comparative research. In the context of evaluating both the PETS2006 and AVSS2007 datasets, the proposed method is more effective than the previous studies for detecting abandoned objects, and achieves the best performance in general.

C. Effectiveness of PFSM and Back-Tracing Verification

This section validates the effectiveness of the proposed PFSM model and back-tracing procedure in improving the performance of abandoned-luggage events. Hereafter, we define ***DualBG-only*** as the method that uses only the pixel classifications from the dual-background models shown in Table I in each single image to detect the abandoned objects. In addition, we define ***PFSM-only*** as the method that removes the back-tracing module in our algorithm.

TABLE V
PERFORMANCE COMPARISON ON PETS2006

Method	Precision	Recall
DualBG-only	0.03	1.0
PFSM-only	0.50	1.0
PFSM with back-tracing (Ours)	1.0	1.0

TABLE VI
PERFORMANCE COMPARISON ON AVSS2007

Method	Precision	Recall
DualBG-only	0.05	1.0
PFSM-only	0.43	1.0
PFSM with back-tracing (Ours)	1.0	1.0

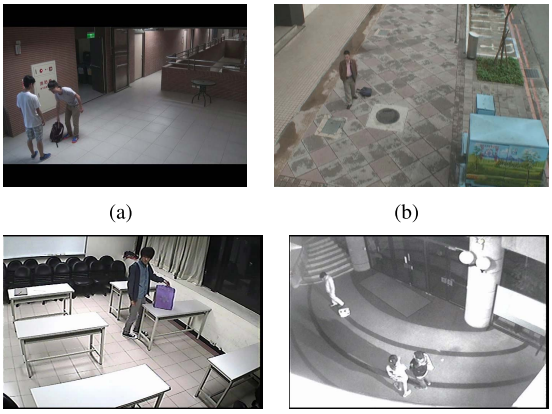


Fig. 12. Examples in ABandoned Objects DAtaset (ABODA). The dataset consists of different scenarios include (a) and (b) outdoor environment, (c) indoor environment, and (d) sudden light changes condition.

For comparison, Table V and Table VI presents the performance under various configurations. All methods attain high recall values; thus, reducing the occurrence of false alarms (i.e., improving the precision) is a critical problem. ***DualBG-only*** provides unstable prediction caused of noisy and imperfect background subtraction processes. Compared to the ***PFSM-only*** method, the temporal transition pattern analysis is critical for detecting the abandoned objects. The PFSM effectively reduces the occurrence of false alarms, and it improves the overall precision to 50% on the PETS2006 dataset and 43% on the AVSS2007 dataset. Most of the false alarms generated by the PFSM are associated with cases of a person remaining temporarily still; for example, in Sequence 3 of the PETS2006 dataset, a person sets down his luggage and rests for a short period. Hence there should be no abandoned event in this case; however, the PFSM issues an alarm because it could not verify whether the owner had left. Therefore, combining the ***back-tracing*** function is assisted in correctly identifying the alarm event. Although tracking remains challenging in crowded scenes, however, we only need to trace the owner in the luggage-centroid ROI, in accordance with the temporal and spatial rules stipulated by PETS2006. It is adequately efficient when tracking the owner by using a simple blob tracker with a human-detector verification method in a spatial-temporal window search. The overall computation speed of our system is 29 fps.

TABLE VII
DETECTION RESULTS ON OUR OWN DATASET ABODA

Sequence	Ground-truth	TP	FP	Scenario
Video1	1	1	0	Outdoor
Video2	1	1	0	Outdoor
Video3	1	1	0	Outdoor
Video4	1	1	0	Outdoor
Video5	1	1	1	Detection in night
Video6	2	2	0	Light switching
Video7	1	1	1	Light switching
Video8	1	1	1	Light switching
Video9	1	1	0	Indoor
Video10	1	1	0	Indoor
Video11	1	1	3	Crowded scene

∗ TP, FP denote true positive, and false positive, respectively.

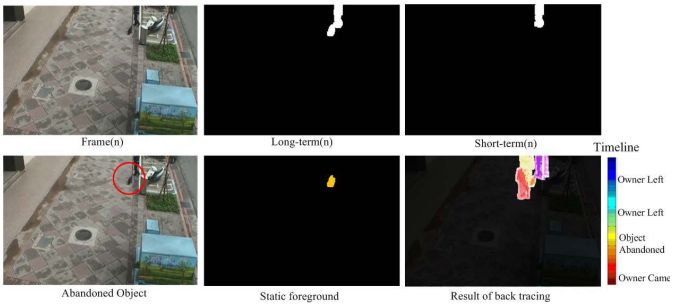


Fig. 13. Detection results of the sequence Video2.

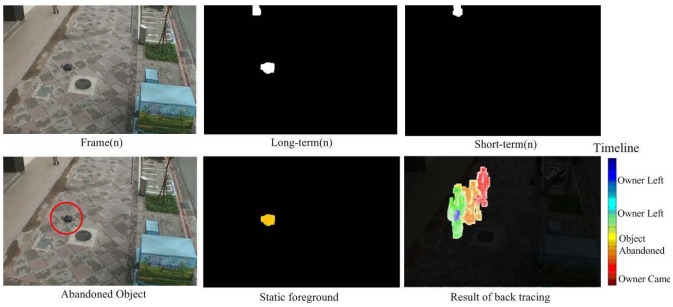


Fig. 14. Detection results of the sequence Video3.

D. Realistic Environment Detection in Our Own Sequence

In this study, we constructed the ABandoned Objects DAtaset (ABODA) for further reliability evaluation.[1] ABODA comprises 11 sequences labeled with various real-application scenarios that are challenging for abandoned-object detection. The situations include crowded scenes, marked changes in lighting condition, night-time detection, as well as indoor and outdoor environments. Figure 12 shows some sequences from the ABODA dataset. Figure 12(a) shows a scenario of a luggage-left event. The owner places his bag down and converses with another person before leaving the scene without his bag (also shown in Figure 15). Figure 12(d) shows a night-time scene. The stationary people stops beside the

[1] ABODA is publicly available for scientific studies and can be downloaded from http://imp.iis.sinica.edu.tw/ABODA/index.html

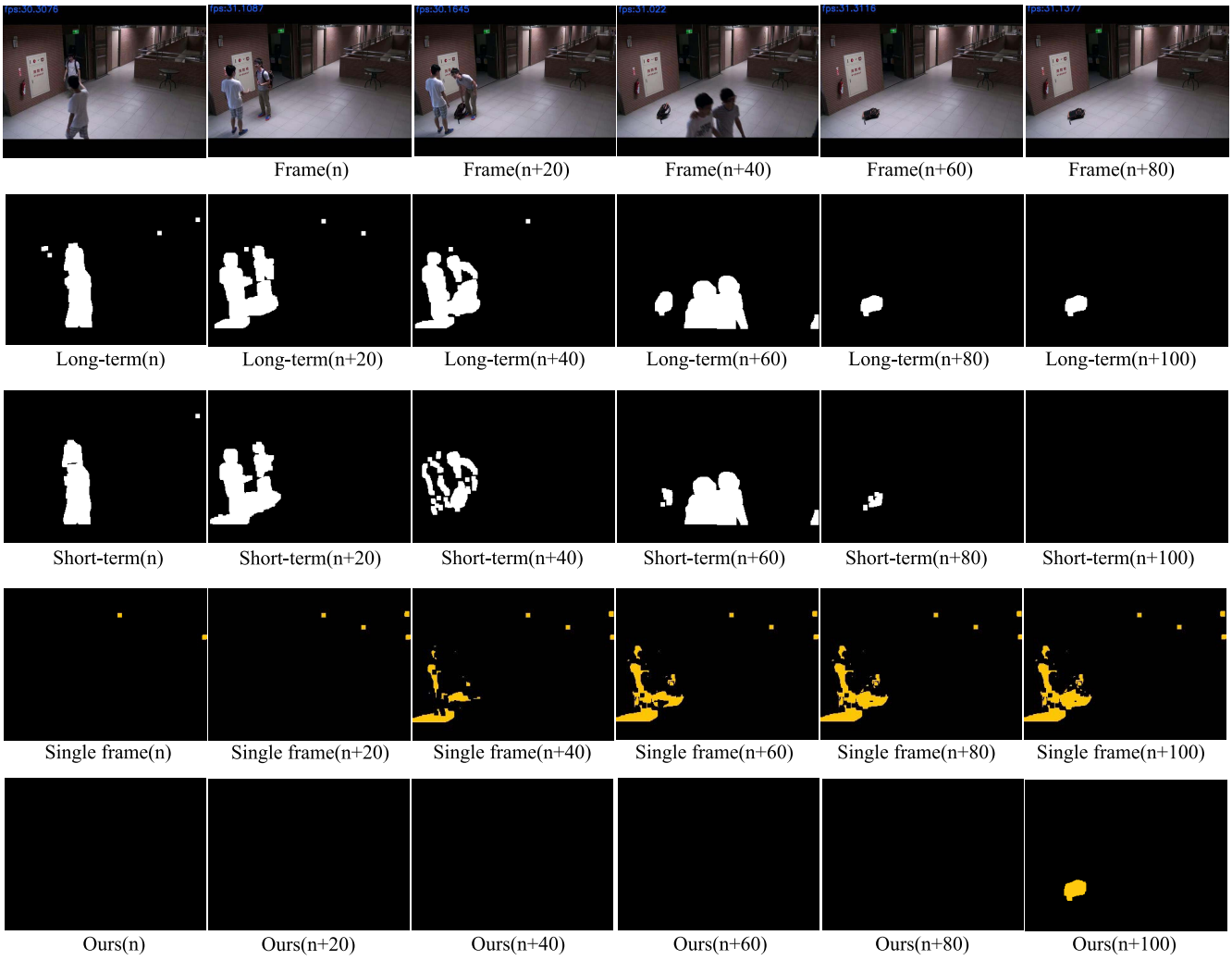


Fig. 15. Static foreground detection of our own dataset. Compare to the single-frame-based method [2], the proposed PFSM precisely localizes the static foreground region of the left luggage, and effectively prevents false alarms generated by ghost and still people.

light rays, and shadows are thus produced behind them. In this case, the shadows are similar to an abandoned object that is "dropped" near the people, which is also a difficult situation for abandoned-object detection algorithms.

The detection results in Table VII show that the overall precision P and recall R are 66.67% and 100%, respectively. The proposed method successfully recalls all of the abandoned objects for both outdoor and indoor environments. Figure 13 and Figure 14 show the ABODA detection results.

In the experiments, few false alarms are raised, which were primarily caused by sudden changes in illumination conditions and crowded scenes. In general, our static foreground detection is based on long- and short-term background modeling. Whenever the light was suddenly turned off, the detection scene became completely dark. Because of its fast learning rate, the short-term background model adapted this condition quickly; however, the long-term background model could not work well, and it extracted several inaccurate foregrounds. These inaccurate foregrounds retained a state $S = 10$ for a while; consequently, the state transition of these foregrounds was similar to that of the static foreground. Figure 16 shows that this condition may have affected the FSM analysis, thereby causing several false alarms. An intuitive solution would be speeding up the learning rate of the long-term background model when the illumination conditions change suddenly. However, this method may be unreliable because both background models would treat the abandoned object as a background object. Therefore, addressing considerable changes illumination remains a challenging issue in our framework.

The challenge of the 11th video is caused by the crowded scene and partial occlusion problem of small objects. In this video, there are a crowd of people waiting in line at the information desk, and the videos were taken by a distant camera. The crowded people (together with specular lightings) cause unperfect background subtraction, as shown in Figure 17. The small object size and partial occlusion also makes the owner identification and tracking highly demanding. Hence, handling more complex crowded scenes with intensively partial occlusions still remains a challenging problem.

E. Performance Comparison With Different Background Subtraction Methods

Background modeling plays an important role in the proposed system. The performance of employing different

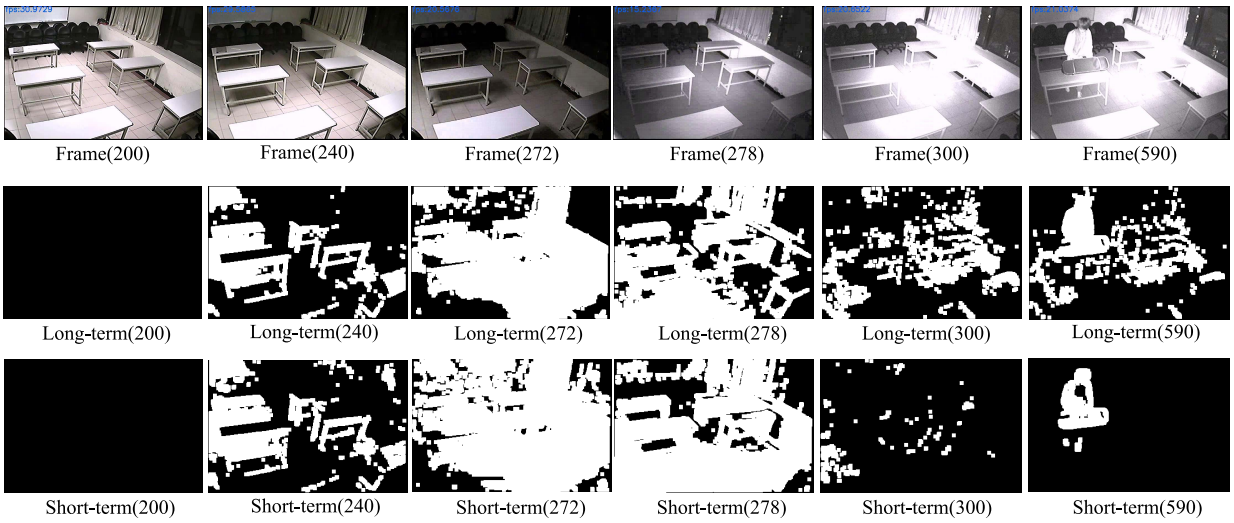


Fig. 16. Results of background subtraction in the sequence Video8 with the scenario of sudden light switching. The digital camera converts to IR mode starts from frame 278.

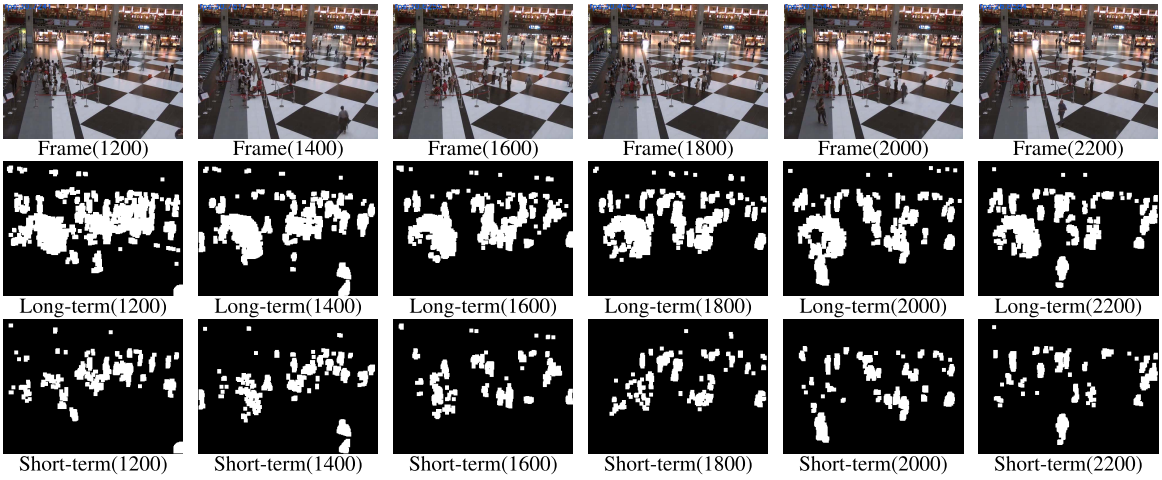


Fig. 17. Background subtraction results of the sequence Video11 with the scenario of crowded scene, which remains a challenge to our approach.

background models in our framework are shown as follows. We have implemented another popular background-modeling method, Codebook [13], for performance comparison. Both EGMM and Codebook gather a series of colors for each pixel and then employ on-line mixture-distribution or clustering to find the candidate colors for per-pixel model building. Codebook has also the parameters controlling the background updating speed. Hence, it can be used for generating long- and short-term background models as well.

Unlike EGMM that treats the colors inside a spherical region centered at a candidate color C as the background colors, Codebook treats the colors inside a cylindrical region centered at C as the background colors. Because the shadow (or lighting) could cause a pixel's color to shift toward (or far away from) the origin in the RGB color space, Codebook claims that a cylindrical region with its axis passing through the origin can avoid generating false foreground pixels caused by shadow or lighting.

However, a drawback of Codebook is that it tends to generate fragmented foregrounds because neighbor pixels are apt to be inconsistent in the background subtraction results. We conduct the experiment on PETS2006 dataset for performance comparison of EGMM and Codebook. We follow the preliminary parameter evaluation as mentioned in Section III.A, and select the best parameters for each background model. Table VIII indicates that EGMM demonstrates a more favorable performance than Codebook. Figure 18 illustrates that the foreground regions extracted from Codebook have many fragmented regions. The fragments are getting increased when the learning rate is slower, particularly for the long-term model. The noisy foreground regions generated from Codebook make our method fail to infer the

TABLE VIII
PERFORMANCE COMPARISON WITH DIFFERENT BACKGROUND SUBTRACTION METHODS ON PETS2006

Background Model	Precision	Recall
EGMM	1.0	1.0
Codebook	0.60	0.43

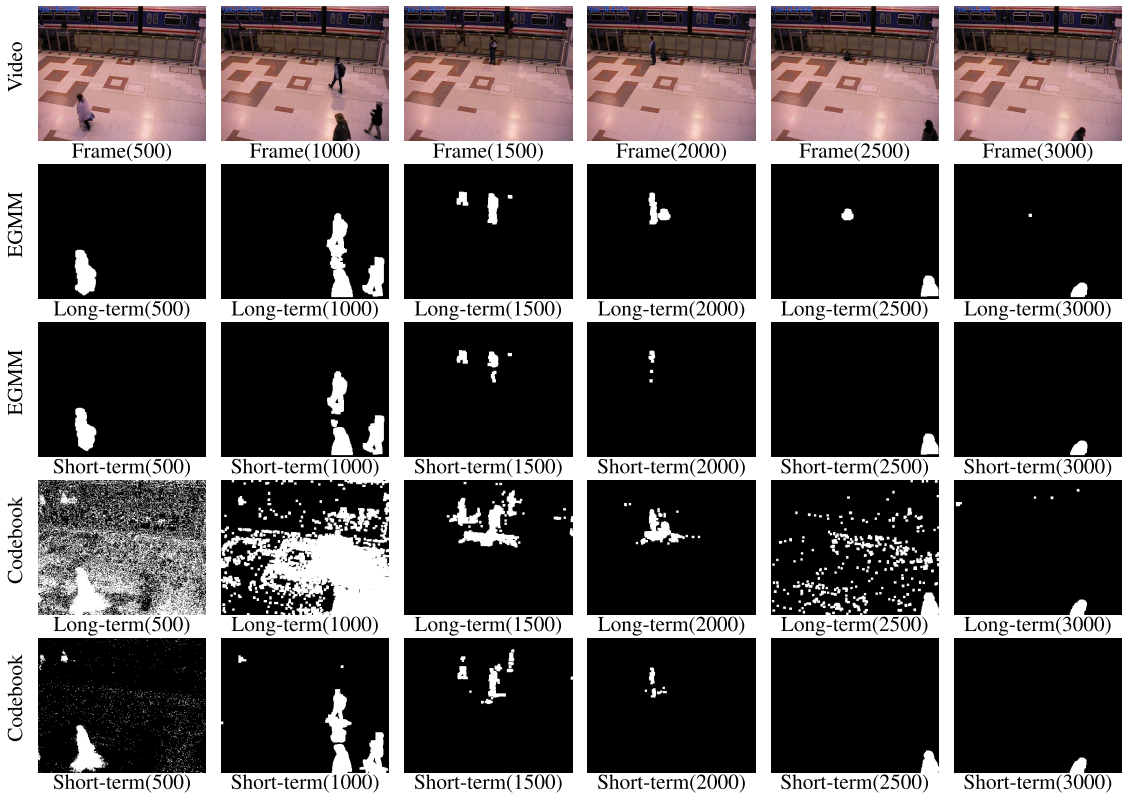


Fig. 18. Performance comparison of background modeling with EGMM and Codebook. First row shows several sample frames in PETS2006. Second and third rows show the long- and short-term model results of EGMM, respectively. Fourth and fifth rows show the long- and short-term model results of Codebook, respectively.

static foreground pixels. Therefore, we recommend to employ EGMM due to its better performance for dual-rate background modeling.

IV. Conclusion

This paper presents a temporal consistency model combining a back-tracing algorithm for abandoned object detection. Characteristics of the proposed approach are summarized as follows:

1) The temporal consistency model is described by a very simple FSM. It exploits the temporal transition pattern generated by short- and long-term background models, which can accurately identify static foreground objects.
2) Our back-tracing algorithm iteratively tracks the luggage owner by using spatial-temporal windows to efficiently verify left-luggage events.
3) The experimental results show that our approach outperforms previous approaches using the PETS2006 and AVSS2007 datasets.
4) In addition, we constructed a novel publicly available dataset, entitled ABODA, comprising plentiful abandoned-object detection situations to assist validating the effectiveness of various approaches for this research direction.

In the future, we plan to enhance our method for handling more challenging situations such as sudden changes in lighting and overly crowded scenes.

References

[1] J. Martínez-del-Rincón, J. E. Herrero-Jaraba, J. R. Gómez, and C. Orrite-Urunuela, "Automatic left luggage detection and tracking using multi-camera UKF," in *Proc. IEEE 9th IEEE Int. Workshop PETS*, Jun. 2006, pp. 59–66.

[2] F. Porikli, Y. Ivanov, and T. Haga, "Robust abandoned object detection using dual foregrounds," *EURASIP J. Adv. Signal Process.*, vol. 2008, Jan. 2008, Art. ID 30.

[3] R. H. Evangelio, T. Senst, and T. Sikora, "Detection of static objects for the task of video surveillance," in *Proc. IEEE WACV*, Jan. 2011, pp. 534–540.

[4] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M.-T. Sun, "Robust detection of abandoned and removed objects in complex surveillance videos," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 5, pp. 565–576, Sep. 2011.

[5] Q. Fan and S. Pankanti, "Modeling of temporarily static objects for robust abandoned object detection in urban surveillance," in *Proc. 8th IEEE Int. Conf. AVSS*, Aug./Sep. 2011, pp. 36–41.

[6] Q. Fan, P. Gabbur, and S. Pankanti, "Relative attributes for large-scale abandoned object detection," in *Proc. IEEE ICCV*, Dec. 2013, pp. 2736–2743.

[7] H.-H. Liao, J.-Y. Chang, and L.-G. Chen, "A localized approach to abandoned luggage detection with foreground-mask sampling," in *Proc. IEEE 5th Int. Conf. AVSS*, Sep. 2008, pp. 132–139.

[8] J. Pan, Q. Fan, and S. Pankanti, "Robust abandoned object detection using region-level analysis," in *Proc. 18th IEEE ICIP*, Sep. 2011, pp. 3597–3600.

[9] F. Lv, X. Song, B. Wu, V. K. Singh, and R. Nevatia, "Left-luggage detection using Bayesian inference," in *Proc. IEEE Int. Workshop PETS*, 2006, pp. 83–90.

[10] L. Li, R. Luo, R. Ma, W. Huang, and K. Leman, "Evaluation of an IVS system for abandoned object detection on PETS 2006 datasets," in *Proc. IEEE Workshop PETS*, 2006, pp. 91–98.

[11] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, and J. Meunier, "Left-luggage detection using homographies and simple heuristics," in *Proc. 9th IEEE Int. Workshop PETS*, 2006, pp. 51–58.

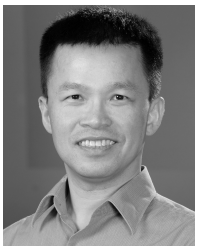
[12] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 2. Jun. 1999, pp. 246–252.
[13] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imag.*, vol. 11, no. 3, pp. 172–185, 2005.
[14] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th ICPR*, 2004, pp. 28–31.
[15] Y.-T. Chen, C.-S. Chen, C.-R. Huang, and Y.-P. Hung, "Efficient hierarchical method for background subtraction," *Pattern Recognit.*, vol. 40, no. 10, pp. 2706–2715, 2007.
[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[17] C. Dubout and F. Fleuret, "Exact acceleration of linear object detectors," in *Proc. 12th ECCV*, 2012, pp. 301–311.
[18] *PETS 2006 Dataset*. [Online]. Available: http://www.cvg.reading.ac.uk/PETS2006/data.html, accessed Mar. 17, 2015.
[19] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1475–1490, Nov. 2004.
[20] *AVSS 2007 Dataset*. [Online]. Available: http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html, accessed Mar. 17, 2015

Kevin Lin received the B.S. degree in electronics engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2012, and the M.S. degree from the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, in 2014. He is currently a Research Assistant with the Institute of Information Science, Academia Sinica, Taipei. His research interests include computer vision, pattern recognition, and machine learning.

Shen-Chi Chen received the B.S. degree in computer science from National Cheng Chi University, in 2007, and the M.S. degree in biomedical engineering from the College of Computer Science, National Chiao Tung University, Taiwan, in 2009. He is currently pursuing the Ph.D. degree in computer science and information engineering with National Taiwan University. His research interests include computer vision, pattern recognition, surveillance system, and intelligent transportation.

Chu-Song Chen is currently a Research Fellow with the Institute of Information Science and the Research Center for IT Innovation, Academia Sinica, Taiwan. He is an Adjunct Professor with the Graduate Institute of Networking and Multimedia, National Taiwan University. His research interests include computer vision, signal/image processing, and pattern recognition. He is on the Governing Board of the Image Processing and Pattern Recognition Society, Taiwan. He served as an Area Chair of ACCV'10 and NBiS'10, the Program Chair of IMV'12 and IMV'13, the Tutorial Chair of ACCV'14, and the General Chair of IMEV'14, and will be the Workshop Chair of ACCV'16. He is on the Editorial Board of the *Journal of Multimedia* (Academy Publisher), *Machine Vision and Applications* (Springer), and the *Journal of Information Science and Engineering*.

Daw-Tung Lin (SM'12) received the B.S. degree in control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1985, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, USA, in 1990 and 1994, respectively. From 1995 to 2004, he was an Associate Professor with the Department of Computer Science and Information Engineering, Chung Hua University, Taiwan. He served as the Director of the Computer Center with Chung Hua University from 2001 to 2003, the Dean of the College of Engineering with Chung Hua University from 2003 to 2005, the Chair of the Department of Computer Science and Information Engineering with National Taipei University from 2006 to 2009, the Director of the Graduate Institute of Communication Engineering with National Taipei University from 2010 to 2011, and the Dean of Academic Affairs with National Taipei University from 2011 to 2015. He is currently the Dean of Academic Affairs, and a Professor with the Department of Computer Science and Information Engineering, National Taipei University, Taipei, Taiwan. He has been with National Taipei University since 2005, where he became a Tenured Professor of Computer Science and Information Engineering in 2009. He has been a regular contributor to the literature in computer vision and image processing. His research interests include image processing, computer vision, pattern recognition, and intelligent surveillance.



Yi-Ping Hung received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1982, the M.S. degree from the Division of Engineering, Brown University, Providence, RI, in 1987, the M.S. degree from the Division of Applied Mathematics, Brown University, in 1988, and the Ph.D. degree from the Division of Engineering, Brown University, in 1990. From 1990 to 2002, he was with the Institute of Information Science, Academia Sinica, Taipei, where he became a Tenured Research Fellow in 1997, and is currently a Joint Research Fellow. He served as the Deputy Director of the Institute of Information Science from 1996 to 1997, and the Director of the Graduate Institute of Networking and Multimedia with National Taiwan University from 2007 to 2013. He is currently a Professor with the Graduate Institute of Networking and Multimedia, and the Department of Computer Science and Information Engineering, National Taiwan University. His current research interests include computer vision, pattern recognition, image processing, virtual reality, multimedia, and human–computer interaction. He was the Program Cochair of ACCV'00 and ICAT'00, and the Workshop Cochair of ICCV'03. He has been an Editorial Board Member of the *International Journal of Computer Vision* since 2004. He will be the General Chair of ACCV'16.