International Journal of Computer Applications (0975 – 8887)
Volume 164 – No 10, April 2017

Abandoned Object Detection using Temporal Consistency Modeling

Divya C. Patil
Dept. of Electronics
Engineering, SSVPSBSD COE,
Dhule

Pravin S. Patil
Dept. of Electronics Engineering,
SSVPSBSD COE,
Dhule

ABSTRACT
In this paper an effective approach for detecting the abandoned object/ luggage for video surveillance is present. Here the long-term and short-term background models are combined to extract foreground objects, where each pixel in an input is classified as two bit code. To identify the static foreground regions, a framework is used based on the temporal transition of code pattern and it also determines whether the candidate regions contain the abandoned object by analyzing the back traced trajectories of luggage owner. This paper also introduces the real-time application of proposed method. The real-time application is performed by using raspberry-pi processor and the raspberry-pi camera. The experimental results show that, the proposed approach is effective for detecting abandoned object/ luggage.

Keywords
Abandoned object detection, long-term background model, short-term background model, visual surveillance, pixel based finite state machine, image processing.

1. INTRODUCTION
Detecting abandoned object/abandoned luggage is referred to as a problem of left luggage or abandoned object detection in the visual surveillance research for public security. It is very critical task particularly for identifying suspicious stationary items. To perform this task common detection method such as training an object detector are inappropriate, because there is no object type of category that can be assumed as having been abandoned.

Figure 1: Examples of abandoned(left) and stolen (right) objects

There are some foreground or background subtraction techniques which are suitable for identifying static foreground regions, that means the object that remain static for a long time as a left luggage candidates. Abandoned object is defined as a static region generated by disappearance of an abandoned object which has been there in the scene before.

To deal with this detection problem an effective approach is proposed in which the short-term and long- term background models are combined for detecting abandoned luggage in surveillance videos.

2. LITERATURE
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.

3. APPROACH

In this project a temporal dual rate foreground integration method is proposed. This method is used for static foreground estimation for single camera video images. Here short-term and long-term background models learned from input surveillance video are involved. Here a simple pixel-based finite state machine (PFSM) models introduced, which uses temporal transition information. Based on sequence pattern of each object pixel, it identifies the static foreground.

Due to these models, the influence of imperfect foreground extraction can be reduced and also accuracy of constructed static foreground inference is improved.

This method performs considerably better than single image based double background model in [1]-[3].

4. PROPOSED ALGORITHM

Proposed method is based on background modeling and subtraction. In camera/video surveillance systems, for detecting moving objects the background subtraction is an essential technique.

Background modeling review:
Here a pixel based background model is learned by taking various images. By studying these images one can define the background pixel and the background model is updated. By studying the new image, the background model is compared. There will be sequence I_m ($t \in N$) of images of size a x b.

To update the background model, we will first initialize the background image b_g (x, y) for each pixel (x, y) where value of x will be vary between 0 to a-1 and value of y will be vary between 0 to b-1. Then there will be comparison of background image (b_g) and previous image (I_m). If the background image (b_g [x, y]) and previous image (I_m [x, y]) will be equal then image will be background image and if b_g (x, y) and I_m (x, y) are not equal then image will be defined as foreground pixel.

To update the background model, the learning rate λ will be applied. The background model used here is Mixture of Gaussian (MOG) model.

Next session of this paper introduces the algorithm for identifying static foreground regions. In following figure there is a flowchart for static foreground detection.

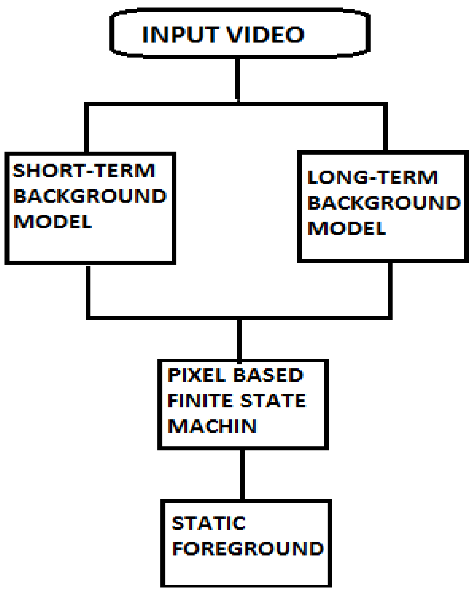


Fig.2: Flowchart for static foreground detectionLong-term and Short-term background model:

Figure 2 shows an overview of flowchart for detecting static foreground regions. There are two types of learning rates small learning rate 'λs' and large learning rate 'λl'. The background model is updated faster with small learning rate 'λs'. This model is called short term model because it is learnt at small learning rates ‘λs’. The background model which is updated at slower speed is called long term model.

Task of finding the stationary objects performed better by using the combination of long-term model and short-term model. The short term model classifies the left-luggage as a background pixel while, the long-term model classifies the foreground pixel.

By concatenating the detected long term and short term foreground, the pixel is represented as two bit code as follows:

$Pi = I_l(i). I_s(i)$

Where,

I_s denotes the binary foreground pixel obtained by short-term model.

I_l denotes the binary foreground pixel obtained by long-term model.

And I_l (i) & I_s (i) ϵ (0, 1) represents the binary values of pixel i of the foreground images.

So, there are 4 states which are represented by two bit code Pi as shown below in table 1 :->

Table 1: classification of pixel

Pi	Hypothesis of the pixel i
00	Background
01	Uncovered background
10	Candidate static foreground
11	Moving foreground

Pi = 00 this condition indicates that pixel I is a background pixel because it is classified as background by both models.

Pi = 01 implies that pixel I is an uncovered background.

Pi = 10 corresponds to static foreground.

Pi = 11 indicates to moving foreground.

From above table we are concerned with state ‘10’ because they are present in the scene from long time and not moved or vibrated for particularly long period of time. So it gives us information about abandoned object.

But this method is not sufficient to detect abandoned object. Therefore we are using pixel based finite state machine (PFSM).

In PFSM, the temporal transition information based on sequential pattern of each pixel is used to identify stationary objects. There are 2 states of each pixel i. e. Pi = 10 and Pi = 11, but the pixel is associated with only one state at a time. The state of the pixel can be changed from one state to another, according to long-term and short term models. So, to understand the behavior of each pixel, a simple Finite State Machine model is described. By identifying the specific pattern of transition we detect the static foreground.

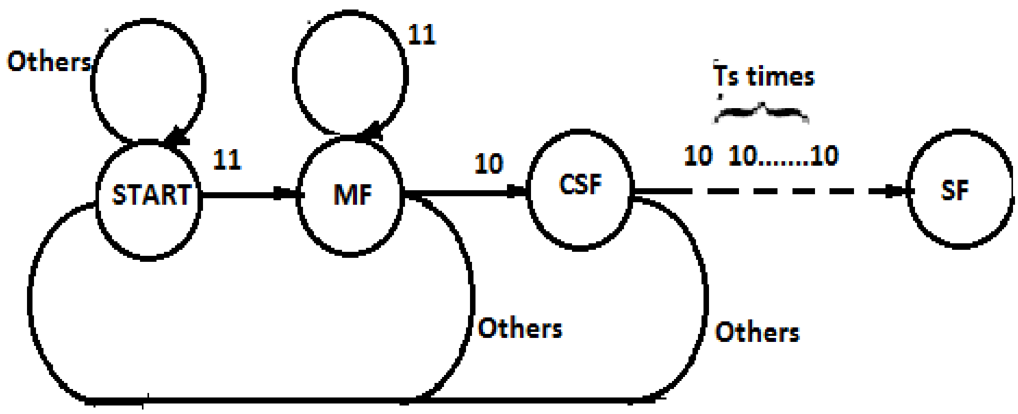


Fig.3 a simple pixel based finite state machine model

Fig shows particular transition for identifying static foreground.

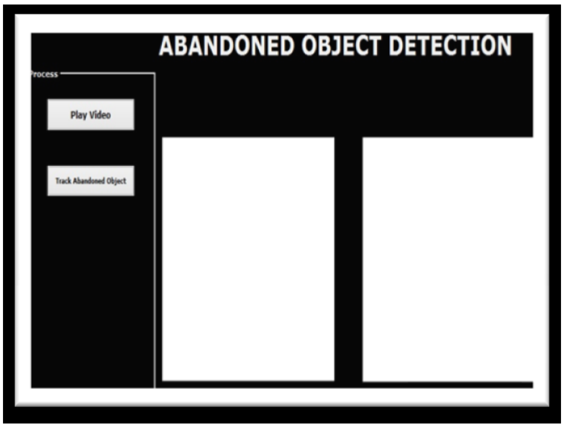
As shown in figure, the system is initially triggered at Pi = 11 which indicates that there is occlusion of foregrounds. After the luggage is abandoned by owner, the short-term model updates the luggage as background, but not long term model; so Pi changes from 11 to 10 i. e. Pi = 10 when this state appears for long time, i. e. t = Ts times, we then decide pixel I is candidate pixel. Thus when initially system starts @ Pi = 11 and followed by long series of 10, then this will be foreground detection.

Results:

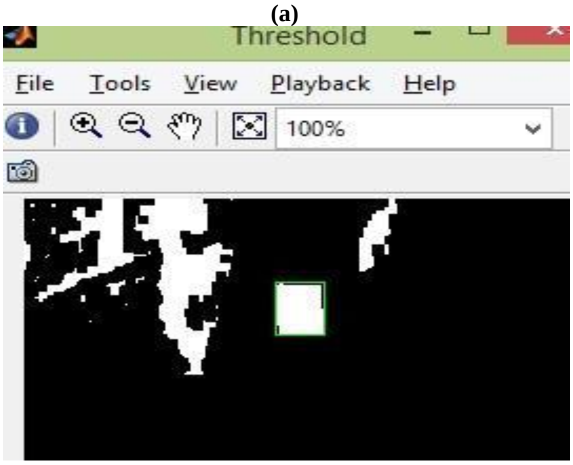
The hardware and software is implemented to get the simulation results.

Software implementation results: we have used the programing language MATLAB (2010b version) to implement the software. The speed of computation while testing video of 360 x 240 pixels was 30 frames per second. The testing was done on a general purpose laptop of 2.00 GHz Intel core-i3 processor.

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. Here, we play a video and then track the object. Then All Objects window marks the region of interest (ROI) with a yellow box and all detected objects with green boxes and Abandoned Object get detected as shown in fig 4. The threshold window shows the threshold as shown in fig



(a)



(b)

Fig.3.(a), (b) Threshold

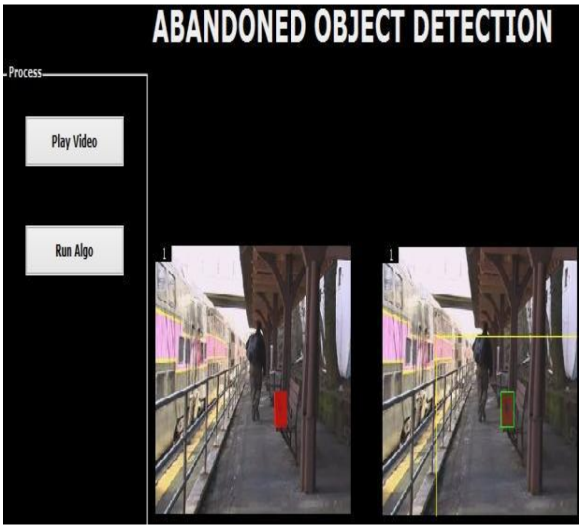


Fig. 4. Abandoned Object Detected

B. Hardware Implementation Details:

We have implemented hardware to run the Real Time Application of the proposed system. For the real time application we used the hardware such as, the Raspberry Pi processor and the Raspberry Pi camera. Also the data cables for Ethernet connection and power supply are used. The Raspberry Pi processor is shown in fig 5.

Here, we first introduce a code to run this real time application in MATLAB R2014b version. We connect two cables between the Raspberry Pi and laptop. A data cable for Ethernet connection and USB cable for power supply. We provide a 0.5 volt supply to the processor. The camera is connected to Camera Serial Interface.

Fig 5. The Raspberry Pi Processor

Initially the raspberry pi camera takes a preview then it sets the background as shown in fig 6. After setting a background image, the camera will trace the video scene. For tracing we give the timer and width. After pressing 'start tracing' the processor starts tracing the steady object as shown in fig 7. Here we take output for 60 frames and if the steady object gets detected and it appears for the time given, then it will considered as an' Abandoned object'. And there will be a message "steady object get detected" as shown in fig 8.

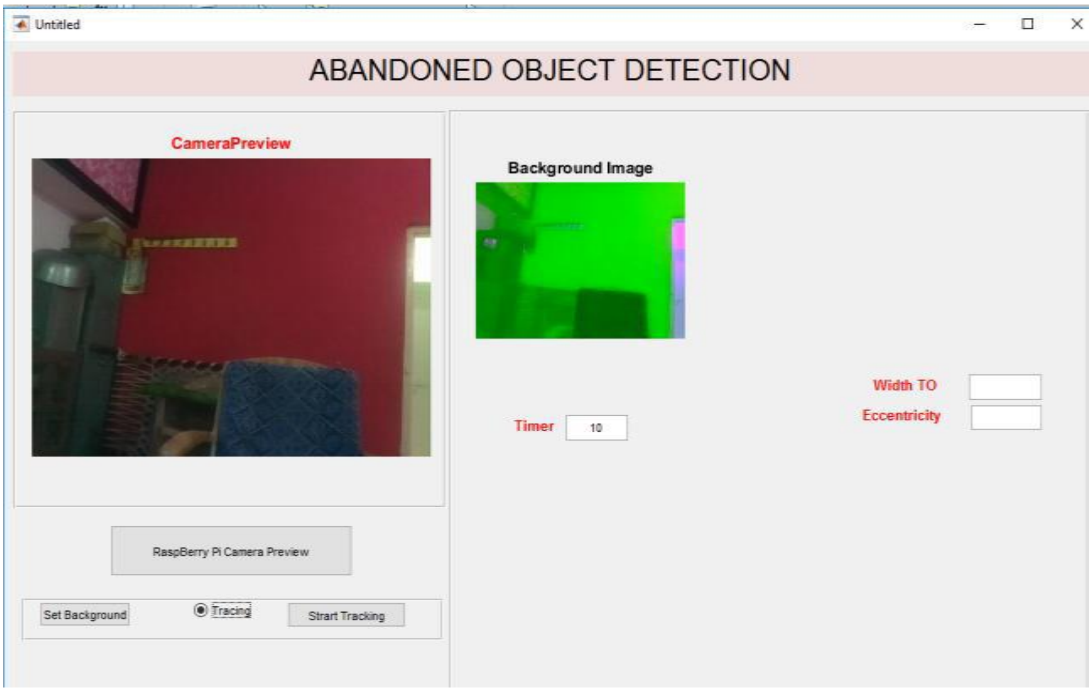


Fig 6. Camera Preview and Background Image

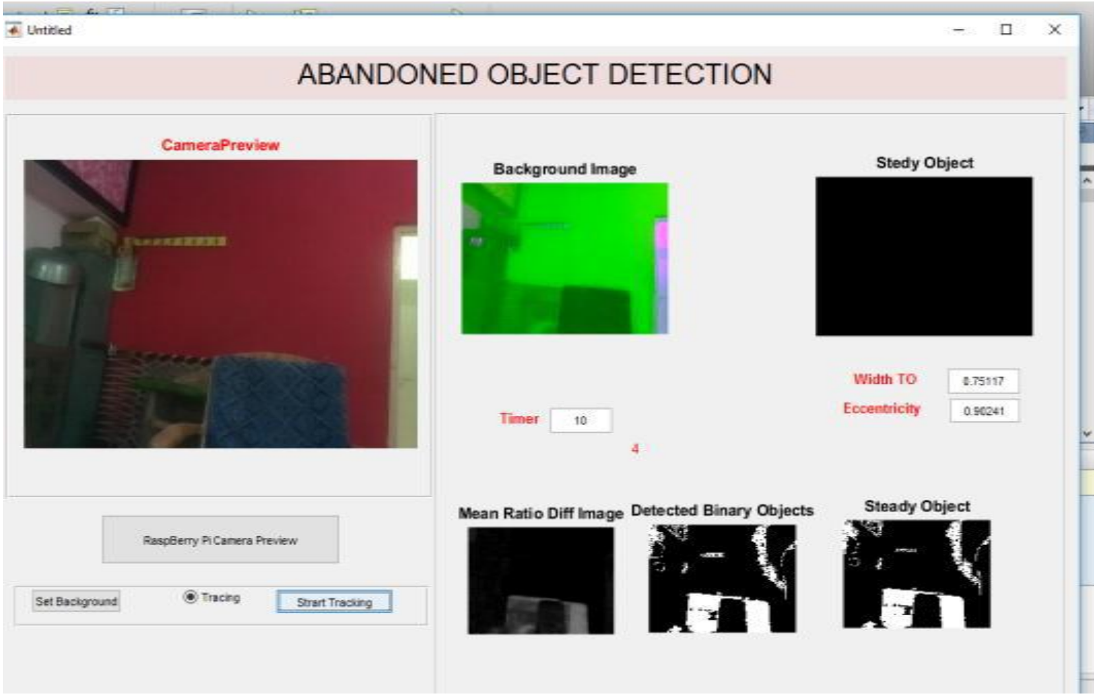


Fig 7. Tracing the Steady Object

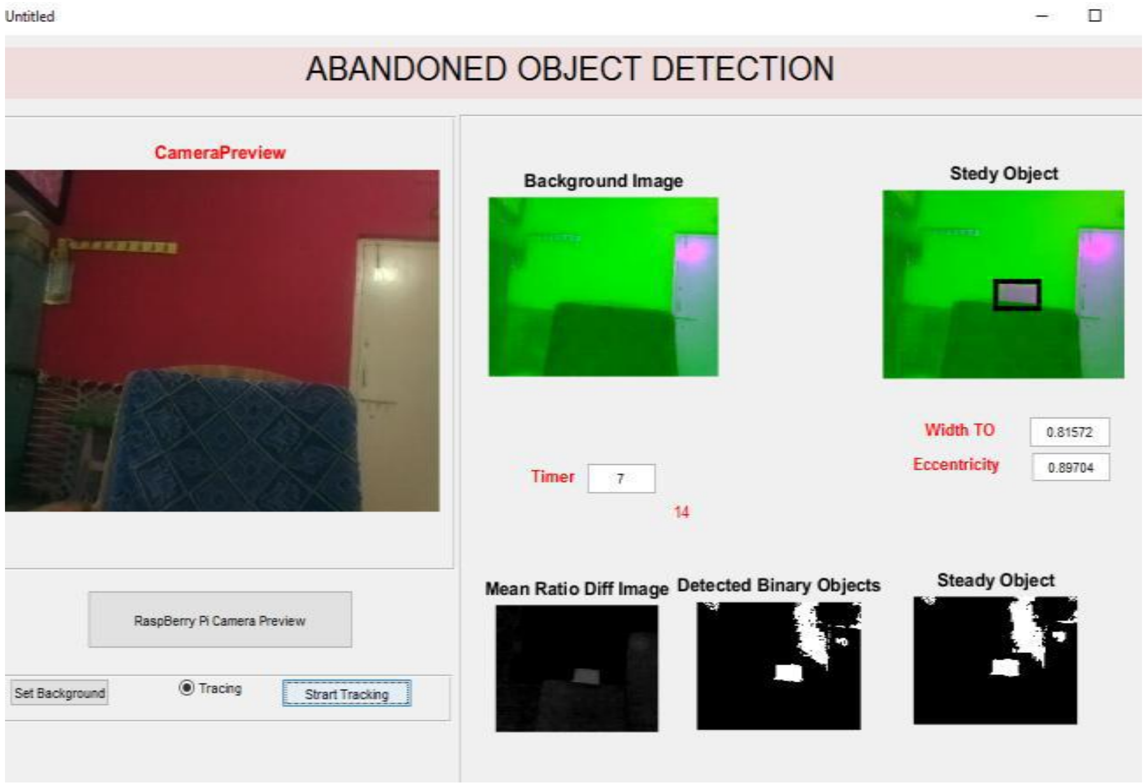


Fig 8. Steady Object Get Detected

5. CONCLUSION

A new framework is presented to robustly and efficiently detect abandoned and removed objects in complex environments for real-time video surveillance. This method can handle occlusions in complex environments with crowds. The testing results which are based on different scenarios have proved that approach used for the purpose can be successfully applied in real world surveillance applications. Here an abandoned object detection system is presented which is based on blob detection methods that are aimed at detecting regions.

6. 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. 8thA IEEE Int. Conf. AVSS*, Aug./Sep. 2011, pp. 36–41.

[6] Q. Fan, P. Gabbur, and S. Pankanti, "Relative attributes for largescale 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.

1370 IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 7, JULY 2015

[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, "Realtime 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.