# Abandoned Object Detection with Region of Interest

**Divya C. Patil, Pravin S. Patil**

***Abstract: Abandoned object detection is an essential requirement in many video surveillance contexts. In this paper, we propose a method to detect abandoned object from surveillance video. Different from conventional approaches that mostly rely on pixel-level processing, we perform region-level analysis in both background maintenance and static foreground object detection. In background maintenance, region-level information is fed back to adaptively control the learning rate. In static foreground object detection, region-level analysis double-checks the validity of candidate abandoned blobs. Different from conventional approaches that mostly rely on pixel-level processing, we perform region-level analysis. In this paper, we present an abandoned object detection system based on blob detection methods are aimed at detecting regions. In a digital image that differs in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant. All the points in a blob can be considered in some sense to be similar to each other. In this paper we are performing a real time application using the Raspberry Pi processor and a Raspberry Pi camera.***

***Keywords: Abandoned object, Video surveillance, Framing, Image, Pixels.***

## I. INTRODUCTION

In recent years, have seen a stark rise in terrorist attacks on crowded public places such as airports, train stations and subways, nightclubs, shopping malls, markets, etc. Many surveillance tools have been employed in the fight against terror. Although video surveillance systems have been in operation for the past two decades, the analysis of the CCTV footage has seldom ventured out of the hands of human operators. Recent studies [1-3] have brought into for the limits to human effectiveness in analyzing and processing crowded scenes, particularly in video surveillance systems consisting of multiple cameras. The advent of smart cameras with higher processing capabilities has now made it possible to design systems which can possibly detect suspicious behaviours (in general) and abandoned objects (in particular). A number of algorithms [5, 7, 8] have been suggested in the recent past to deal with the problem of abandoned-object-detection. Due to their dependence on complex probabilistic mathematics, most of these algorithms have failed to perform satisfactorily in real time scenarios. In addition, the other difficulty of detecting an abandoned object under occlusion adds to the overall complexity. Some proposed algorithms [4-5] have dealt with partial occlusion (by moving people) but complete or prolonged occlusion (by another object) has not yet been tackled.

**Revised Version Manuscript Received on January 04, 2017.**

**Ms. Divya C. Patil,** Scholar, Department of Electronics Engineering, S.S.V.P.S's B.S.D. College of Engineering, Dhule (Maharashtra)-424005, India.

**Dr. Pravin S. Patil,** Professor, Department of Electronics Engineering, S.S.V.P.S's B.S.D. College of Engineering, Dhule (Maharashtra)-424005, India.

In this paper, we present an abandoned object detection system based on blob detection methods are aimed at detecting regions. In a digital image that differs in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant. All the points in a blob can be considered in some sense to be similar to each other.

## II. RELATED WORK

Most of the proposed techniques for abandoned object detection rely on tracking information [1, 4] to detect drop-off events, while fusing information from multiple cameras. As stated by Porikli, these methods are not well suited to complex environments like scenes involving crowds and large amounts of occlusion. In addition, they require solving a difficult problem of object tracking and detection as an intermediate step. Aiming to address these limitations, Porikli proposed a single camera, non-tracking-based system which makes use of two backgrounds for the detection of stationary objects. The two backgrounds are constructed by sampling the input video at different frame rates (one for short-term and another for long-term events). This technique, however, is difficult to set appropriate parameters to sample the input video for different applications, and has no mechanism to decide whether a persistent foreground blob corresponds to an abandoned object event or a removed object event. In many surveillance scenarios, the initial background contains objects that are later removed from the scene (e.g., parked cars or static people that move away). Correctly classifying whether a foreground blob corresponds to abandoned or removed objects is an essential problem in background modelling, but most existing systems neglect it.

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 colour representation (PCR) into a template-matching scheme, and also by estimating the status (e.g., occluded or removed) of a stationary object. 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.

## III. ABANDONED / REMOVED OBJECT DETECTION

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance. Losed-circuit television (CCTV), also known as video surveillance, is the use of video cameras to transmit a signal to a specific place, on a limited set of monitors. It differs from broadcast television in that the signal is not openly transmitted, though it may employ point to point (P2P), point to multipoint (P2MP), or mesh wireless links. Though almost all video cameras fit this definition, the term is most often applied to those used for surveillance in areas that may need monitoring such as banks, casinos, airports, military installations, and convenience stores. Video telephony is seldom called "CCTV" but the use of video in distance education, where it is an important tool, is often so called.

In computer vision, blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other.

**Fig. 1. Abandoned Object Detection**

Given some property of interest expressed as a function of position on the image, there are two main classes of blob detectors: (i) *differential methods*, which are based on derivatives of the function with respect to position, and (ii) *methods based on local extrema*, which are based on finding the local maxima and minima of the function. In mathematical analysis, the maxima and minima (the respective plurals of maximum and minimum) of a function, known collectively as extrema (the plural of extremum), are the largest and smallest value of the function, either within a given range (the local or relative extrema) or on the entire domain of a function (the global or absolute extrema). Pierre de Fermat was one of the first mathematicians to propose a general technique, adequality, for finding the maxima and minima of functions. With the more recent terminology used in the field, these detectors can also be referred to as *interest point operators*, or alternatively interest region operators (see also interest point detection and corner detection).

There are several motivations for studying and developing blob detectors. One main reason is to provide complementary information about regions, which is not obtained from edge detectors or corner detectors. In early work in the area, blob detection was used to obtain regions of interest for further processing. These regions could signal the presence of objects or parts of objects in the image domain with application to object recognition and/or object tracking. In other domains, such as histogram analysis, blob descriptors can also be used for peak detection with application to segmentation. Another common use of blob descriptors is as main primitives for texture analysis and texture recognition. In more recent work, blob descriptors have found increasingly popular use as interest points for wide baseline stereo matching and to signal the presence of informative image features for appearance-based object recognition based on local image statistics. There is also the related notion of ridge detection to signal the presence of elongated objects.

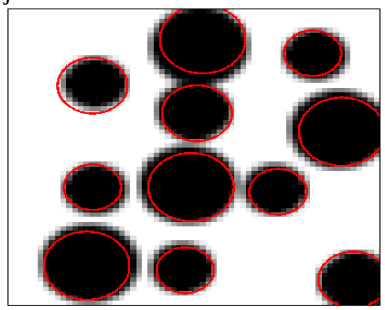


**Fig. 2.Blob Detection**

The blob descriptors obtained from these blob detectors with automatic scale selection are invariant to translations, rotations and uniform rescaling in the spatial domain. The images that constitute the input to a computer vision system are, however, also subject to perspective distortions. To obtain blob descriptors that are more robust to perspective transformations, a natural approach is to devise a blob detector that is *invariant to affine transformations*. In practice, affine invariant interest points can be obtained by applying affine shape adaptation to a blob descriptor, where the shape of the smoothing kernel is iteratively warped to match the local image structure around the blob, or equivalently a local image patch is iteratively warped while the shape of the smoothing kernel remains rotationally symmetric (Lindeberg and Garding 1997; Baumberg 2000; Mikolajczyk and Schmid 2004, Lindeberg 2008). In this way, we can define affine-adapted versions of the Laplacian/Difference of Gaussian operator, the determinant of the Hessian and the Hessian-Laplace operator (see also Harris-Affine and Hessian-Affine).

## IV. INTEREST POINT DETECTION

**Interest point detection** is a recent terminology in computer vision that refers to the detection of interest points for subsequent processing. An interest point is a point in the image which in general can be characterized as follows:

- it has a clear, preferably mathematically well-founded, definition,
- it has a well-defined *position* in image space,
- the local image structure around the interest point is rich in terms of local *information contents* (e.g.: significant 2D texture), such that the use of interest points simplify further processing in the vision system,
- It is *stable* under local and global perturbations in the image domain as illumination/brightness variations, such that the interest points can be reliably computed with high degree of *repeatability*.
- Optionally, the notion of interest point should include an attribute of *scale*, to make it possible to compute interest points from real-life images as well as under scale changes.

Historically, the notion of interest points goes back to the earlier notion of corner detection, where corner features were in early work detected with the primary goal of obtaining robust, stable and well-defined image features for object tracking and recognition of three-dimensional CAD-like objects from two-dimensional images. In practice, however, most corner detectors are sensitive not specifically to corners, but to local image regions which have a high degree of variation in all directions. The use of interest points also goes back to the notion of regions of interest, which have been used to signal the presence of objects, often formulated in terms of the output of a blob detection step. While blob detectors have not always been included within the class of interest point operators, there is no rigorous reason for excluding blob descriptors from this class. For the most common types of blob detectors (see the article on blob detection), each blob descriptor has a well-defined point, which may correspond to a local maximum, a local maximum in the operator response or a centre of gravity of a non-infinitesimal region. In all other respects, the blob descriptors also satisfy the criteria of an interest point defined above. It is true that a number of blob descriptors contain complementary information. But these additional attribute should not disqualify blob descriptors from being included within the class of interest points.

In terms of applications, the use of corner detection and blob detection are also overlapping. Today, a main application of interest points is to signal points/regions in the image domain that are likely candidates to be useful for *image matching* and *view-based object recognition*. For this purpose, several types of corner detectors and blob detectors have been demonstrated to be highly useful in practical applications (see respective articles for references). Blob detectors and corner detectors have also been used as primitives for texture recognition, texture analysis and for constructing 3D models from multiple views of textured objects.

If one aims at drawing a distinction between corner detectors and blob detectors, this can often be done in terms of their localization properties at corner structures. For a junction structure in the image domain that corresponds to an intersection of physical edges in the three-dimensional world, the localization properties of a corner detector will in most cases is much better than the localization properties that would be obtained from a blob detector. Hence, for the purpose of computing structure and motion from multiple views, corner detectors will in many cases have advantages compared to blob detectors in terms of smaller localization error. Notwithstanding this, blob descriptors have also been demonstrated to be useful when relating object models to temporal imagery.

## V. CORNER DETECTION

Corner detection is an approach used within computer vision systems to extract certain kinds of features and infer the contents of an image. Corner detection is frequently used in motion detection, image registration, video tracking, image misaiming, panorama stitching, 3D modeling and object recognition. Corner detection overlaps with the topic of interest point detection.

A corner can be defined as the intersection of two edges. A corner can also be defined as a point for which there are two dominant and different edge directions in a local neighbourhood of the point. An interest point is a point in an image which has a well-defined position and can be robustly detected. This means that an interest point can be a corner but it can also be, for example, an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal.

"Corner", "interest point" and "feature" are used interchangeably in literature, confusing the issue. Specifically, there are several blob detectors that can be referred to as "interest point operators", but which are sometimes erroneously referred to as "corner detectors". Moreover, there exists a notion of ridge detection to capture the presence of elongated objects. Corner detectors are not usually very robust and often require large redundancies introduced to prevent the effect

of individual errors from dominating the recognition task. One determination of the quality of a corner detector is its ability to detect the same corner in multiple similar images, under conditions of different lighting, translation, rotation and other transforms.

A simple approach to corner detection in images is using correlation, but this gets very computationally expensive and suboptimal. An alternative approach used frequently is based on a method proposed by Harris and Stephens (below), which in turn is an improvement of a method by Moravec.

## VI. SYSTEM INTERFACE

Static region is healed and classified as abandoned or removed object; some conditions need to be verified before triggering an alert. These conditions are specified by the user using our system interface, which include: 1) Sizes: minimum and maximum object size; 2) Regions of Interest: polygonal regions manually drawn by the user in the image (events are detected only on those regions) and 3) Time: indicates how long a foreground region corresponding to an abandoned/removed object should stay stationary in the scene in order to trigger an alert. The conditions based on size and regions of interest are trivial to implement. For the time condition, we need to keep track of the healed static region and check whether it is persistent during the time period specified by the user.

This shows how to track objects at a train station and to determine which ones remain stationary. Abandoned objects in public areas concern authorities since they might pose a security risk. Algorithms, such as the one used in this example, can be used to assist security officers monitoring live surveillance video by directing their attention to a potential area of interest.

A region of interest (often abbreviated ROI), is a selected subset of samples within a dataset identified for a particular purpose. The concept of a ROI is commonly used in many application areas. In computer vision and optical character recognition, the ROI defines the borders of an object under consideration. In many applications, symbolic (textual) labels are added to a ROI, to describe its content in a compact manner.

This example illustrates how to use the Blob Analysis and MATLAB® Function blocks to design a custom tracking algorithm. The example implements this algorithm using the following steps: 1) Eliminate video areas that are unlikely to contain abandoned objects by extracting a region of interest (ROI). 2) Fill in small gaps in the detected objects. 3) Calculate object statistics using the Blob Analysis block. 4) Call the helper function that tracks the identified objects and returns the bounding boxes and the number of the abandoned objects. 5) Display the abandoned object detection results step (h Abandoned Objects, IMR). 6) Display the entire detected objects step (h All Objects, IMR). 7) Display the segmented video.

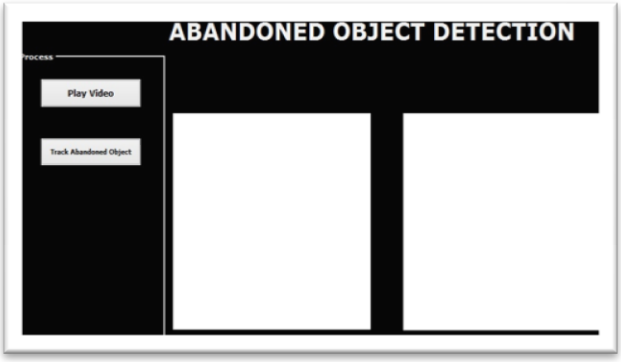
## VII. EXPERIMENTAL RESULTS

We have done our project by implementing a software and hardware simulation.

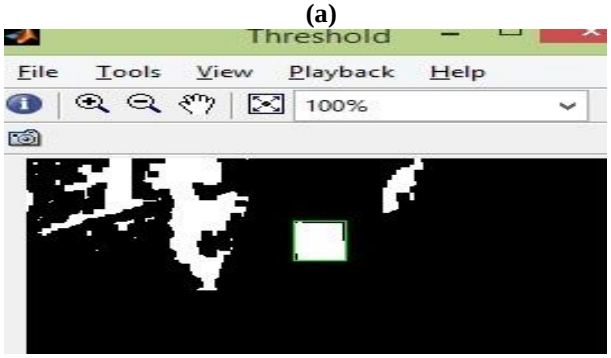
### *A. Implementation Details for Software*

The proposed system was developed using the programming language MATLAB R2010b version. The overall computation speed is 30 fps when testing the video of 360×240 pixel video by using a general purpose laptop with a 2.0 GHz Intel Core-i3 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 Open CV library, is used in this work because of its high performance.

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 fi




**(a)**




**(b)**

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


**Fig. 4. Abandoned Object Detected**

### *B. Implementation Details for Hardware*

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.

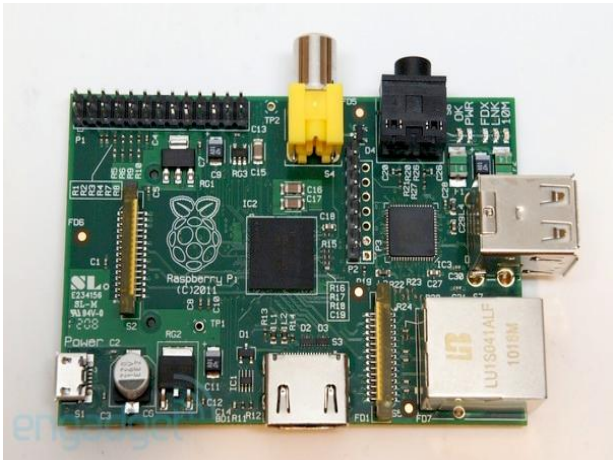


**Fig 5. The Raspberry Pi Processor**

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.

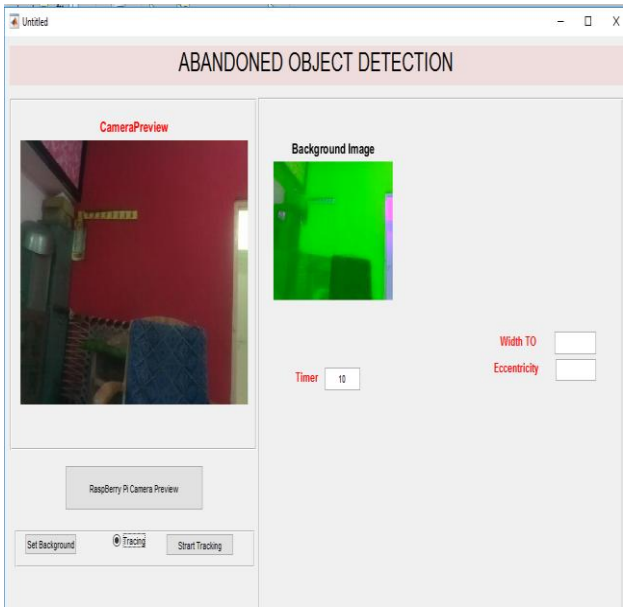



**Fig 6. Camera Preview and Background Image**

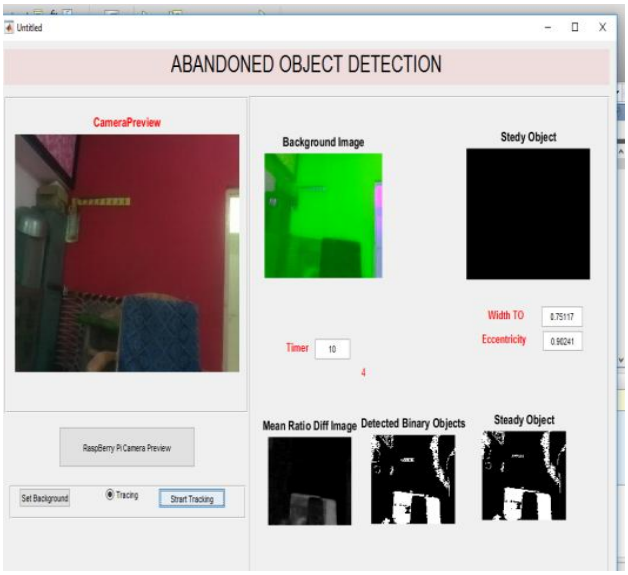



**Fig 7. Tracing the Steady Object**

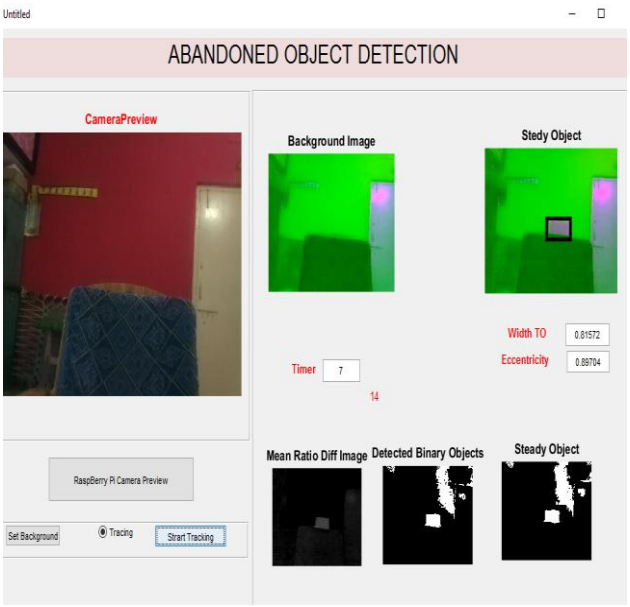



**Fig 8. Steady Object Get Detected**

## VIII. APPLICATIONS

The robustness and efficiency of the method was tested on our system for public safety application in big cities. Many methods have been recently proposed to automatically detect abandoned objects (parked vehicles and left-luggage) in video surveillance for different applications such as traffic monitoring, public safety, retail, etc.

## IX. CONCLUSION

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

## 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 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.
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. PETS 2006 Dataset. [Online]. Available: http://www.cvg.reading.ac.uk/ PETS2006/data.html, accessed Mar. 17, 2015.