

# Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos

YingLi Tian, *Senior Member, IEEE*, Rogério Feris, *Senior Member, IEEE*, Haowei Liu, *Student Member, IEEE*, Arun Hampapur, *Senior Member, IEEE*, and Ming-Ting Sun, *Fellow, IEEE*

**Abstract**— Tracking-based approaches for abandoned object detection often become unreliable in complex surveillance videos due to occlusions, lighting changes, and other factors. We present a new framework to robustly and efficiently detect abandoned and removed objects based on background subtraction and foreground analysis with complement of tracking to reduce false positives. In our system, the background is modeled by three Gaussian mixtures. In order to handle complex situations, several improvements are implemented for shadow removal, quick lighting change adaptation, fragment reduction, and keeping a stable update rate for video streams with different frame rates. Then, the same Gaussian mixture models used for background subtraction are employed to detect static foreground regions without extra computation cost. Furthermore, the types of the static regions (abandoned or removed) are determined by using a method that exploits context information about the foreground masks, which significantly outperforms previous edge-based techniques. Based on the type of the static regions and user-defined parameters (e.g., object size and abandoned time), a matching method is proposed to detect abandoned and removed objects. A person-detection process is also integrated to distinguish static objects from stationary people. The robustness and efficiency of the proposed method is tested on IBM Smart Surveillance Solutions for public safety applications in big cities and evaluated by several public databases such as iLids and PETS2006 datasets. The test and evaluation demonstrate our method is efficient to run in real-time while being robust to quick lighting changes and occlusions in complex environments.

**Index Terms**— Anti-terrorism, background subtraction, foreground analysis, abandoned object, removed object, video surveillance.

## I. INTRODUCTION

The failed car bombing happened recently in Times Square at New York City demonstrated that effective and efficient detection of abandoned objects is very important to prevent attacks on landmarks, public transportation and

critical assets. Many methods have been recently proposed to automatically detect abandoned objects (parked vehicles and left-luggage) in video surveillance [1, 3-5, 7, 9, 13-15, 19-20, 24-25, 28-32, 36-38, 44] for different applications such as traffic monitoring, public safety, retail, etc. At train/subway stations, airports, big cities, and other public spaces with high traffic flows, it becomes very challenging for security officers as well as video surveillance solutions to quickly detect objects that have been left behind. Although efforts have been made to establish some standards (e.g., PETS and iLids), the problem is not well-defined and still an open problem in video surveillance. For example, Beynon *et al.* [4] defined an abandoned package as any stationary package away from anyone considered responsible for it. Bird *et al.* [5] defined an abandoned object to be a stationary object that has not been touching a person (someone had to leave it) for some time threshold. Ferrando *et al.* [13] defined an abandoned object as a static “non-human” object which splits from a “human”. Spengler and Schiele [32] defined an abandoned object as a “non-human” foreground which keeps still over a certain period of time and without humans being close by. All above definitions cannot cover the complex situations in real life. For example, a car/truck is parked and then the driver leaves, or someone just throws a bag to an area from long distance. Also, in very crowded environments, it is difficult to detect the relationship of an abandoned object and its owner such as someone leaves a bag to his/her friend.

We define an **abandoned object** to be a stationary object that has **not** been in the scene **before**, and a **removed object** to be a stationary object that has been in the scene **before** but is **not** there anymore. To detect abandoned and removed objects, we focus on how to detect static regions that have recently changed in the scene and how to determine whether they correspond to abandoned or removed objects.

## II. RELATED WORK

Most of the proposed techniques for abandoned object detection rely on tracking information [1, 4, 15, 19, 27, 30-32] to detect drop-off events, while fusing information from multiple cameras. As stated by Porikli [24, 25], 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 [24, 25]

Manuscript received January 21, 2010, revised May 29, 2010.

YingLi Tian is with the City College, City University of New York, New York, NY 10031 USA (phone: 212-650-7046; fax: 212-650-8249; e-mail: ytian@ccny.cuny.edu). Prior to joining the City College in September 2008, she was with IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA.

Rogério Feris is with IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: rsferis@us.ibm.com).

Haowei Liu is with the University of Washington, Seattle, WA 98195 USA (email: hwliu@u.washington.edu).

Arun Hampapur is with IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: arunh@us.ibm.com).

Ming-Ting Sun is with the University of Washington, Seattle, WA 98195 USA (email: mts@u.washington.edu).

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 modeling, but most existing systems neglect it.

The ObjectVideo surveillance system [37] keeps track of background regions which are stored right before they are covered by an abandoned object. In case the same object is removed (i.e., the background is uncovered), the stored region can be matched with the current frame to determine that the object was removed. Clearly, this approach fails when the static object stays long enough in the scene, which makes the matching of the current frame with the stored background region more difficult due to differences in lighting. Another problem occurs when an object is already part of the initial background. For these cases, the ObjectVideo system relies on analyzing the edge energy associated with the boundaries of the foreground region for both the current frame and the background model. The assumption is that the edge energy of the current frame is higher for abandoned objects and lower for removed objects. This method was originally proposed by Connell *et al.* [9].

Relying on edge energy to distinguish abandoned and removed objects works well for simple, homogeneous backgrounds. However, the edge energy assumption is clearly violated in complex scenes with cluttered backgrounds. Another big limitation of the edge energy based method is that only parts of the static objects are often detected due to the imperfect background subtraction in real surveillance systems for complex environment applications.

### III. OVERVIEW OF OUR APPROACH

In this paper, we propose a novel solution to detect abandoned and removed objects. An earlier version of this paper can be found in [36]. Compared to our previous work, there are four major extensions that merit being highlighted: **1)** the previous work did not keep a history of the background objects in the scene. The extended method provides more accurate results to classify static regions into abandoned and removed objects (Section V-B); **2)** the previous work was not able to distinguish stationary human from non-human objects. In this paper, we integrate human detection in near-field, mid-field, and far-field scenarios into the framework (Section VI-A); **3)** the previous work didn't use any tracking information. In order to reduce the false positives in complex videos, we employ the tracking trajectories as complementary

information (Section VI-D); **4)** we add more quantitative experimental results in complex city scenarios (Sections VII-D and VII-E) and demonstrate the efficiency and robustness of the proposed work. In addition, the mathematical framework behind the technique is described in this paper.

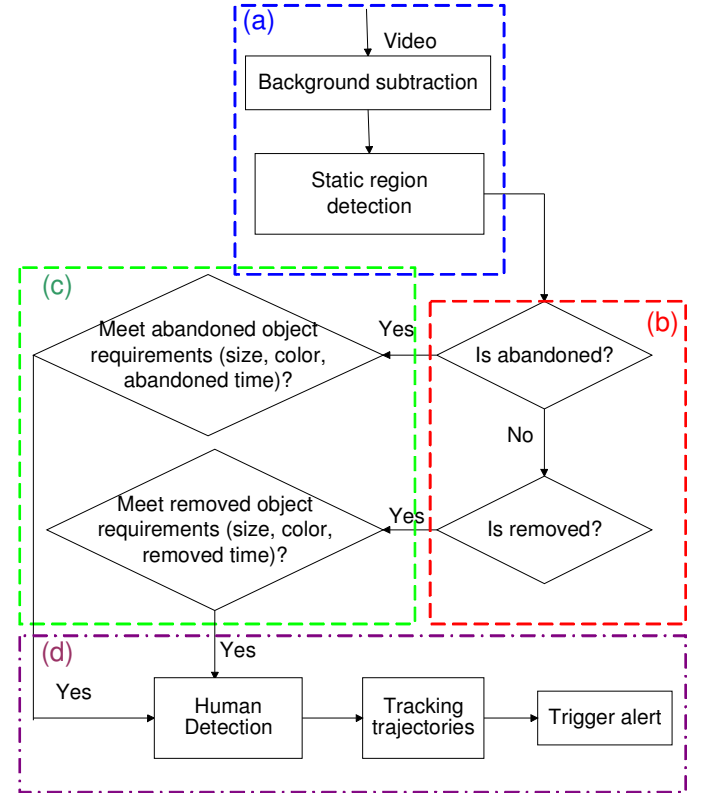


Figure 1: System diagram: (a) background subtraction and static region detection; (b) object type detection (abandoned or removed); (c) abandoned/removed object alert detection; and (d) integration with human detection and tracking information for abandoned and removed object alert detection.

Fig. 1 shows our system diagram. The system includes four main components: (a) background subtraction and static region detection; (b) object type detection (abandoned or removed); (c) abandoned/removed object alert detection; and (d) integration with human detection and tracking information for abandoned and removed object alert detection. Overall, the work introduced in this paper offers the following main contributions to robust detection of abandoned and removed objects in complex surveillance videos:

- We employ a mixture of Gaussians method to analyze the foreground as *moving objects*, *abandoned objects*, or *removed objects (ghosts)* while detecting the background.
- Different thresholds are used to obtain the foreground mask and the static region mask.
- For the foreground mask, the intensity and texture information are integrated to remove shadows and to make the algorithm working for quick lighting changes.
- For the static region mask, a new segmentation method

is developed to detect the type of the static region (abandoned or removed), significantly outperforming previous edge-based techniques.

- A matching algorithm is employed to detect if the object is abandoned/removed long enough even under partial occlusion. Here, partial occlusion means either only a part of the object is occluded or the whole object is only occluded for a short period time (less than the user defined abandoned/removed time.) The history of background objects in the scene is kept to make the matching algorithm robust to lighting changes in complex videos.
- Human detection which can detect people in near-field, mid-field, and far-field scenarios is integrated into the framework to distinguish people who stay stationary from static objects.
- The tracking trajectories are employed as complement information to reduce the false positives in complex videos. Only for those abandoned/removed objects that meet the alert requirements that are defined by users will trigger the alerts.

This paper is organized as follows. In Section IV, we describe the method of background subtraction and static region detection. The object type detection (abandoned or removed) is presented in Section V. We describe the human detection, abandoned/removed object alert detection, system interface, and complement with tracking information in Section VI. Section VII covers our experimental results on standard datasets as well as quantitative evaluation in real-world complex surveillance scenarios.

#### IV. STATIC OBJECT DETECTION

In this section, we describe how to detect the static objects from the scene. Here the static objects are the changes of the scene that stay in the same position for relatively long time. These static objects can be classified as abandoned objects and removed objects. We employ the mixture of Gaussian method to detect scene changes due to its robustness and efficiency. We further extend the method to detect static objects by using different mixture models.

##### A. Multi-Gaussian Adaptive Background Models and Improvements

Stauffer and Grimson [33] introduced a mixture of  $K$  Gaussians ( $K$  is from 3 to 5) to build the background model and detect the moving objects. For a pixel  $X$  at time  $t$ , the probability of the pixel can be written as [33]:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \quad (1)$$

Where  $K$  is the number of Gaussian mixtures,  $\omega_{i,t}$  is an estimation of the update weight of the  $i$ th Gaussian

distribution,  $\mu_{i,t}$  is the mean of the  $i$ th Gaussian distribution,  $\Sigma_{i,t}$  is the covariance matrix of the  $i$ th Gaussian distribution, and  $\eta$  is a Gaussian probability density function:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)}, \quad (2)$$

where  $n$  is the dimension of the intensity at the pixel  $X$  and

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}). \quad (3)$$

Where  $\alpha$  is the learning rate and  $M_{k,t}$  is 1 for the model which matched and 0 for the remaining models. By assuming the red, green, and blue pixel values are independent and have the same variances,  $\Sigma_{k,t} = \sigma_k^2 I$ . After the Gaussians are ordered by the value of  $\omega/\alpha$ , the first  $B$  distributions are chosen as the background model, where

$$B = \arg \min_b \left( \sum_{k=1}^b \omega_k > T \right), \quad (4)$$

where  $T$  is the minimum portion of the background model. The  $\mu$  and  $\sigma$  parameters for unmatched distributions remain the same. The parameters of the distribution which matches the new observation are updated as follows:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t, \quad (5)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t). \quad (6)$$

where

$$\rho = \alpha \eta(X_t | \mu_k, \sigma_k) \quad (7)$$

In the implementation, two significant parameters --  $\alpha$  and  $T$  need to be set. See more details in Stauffer and Grimson [33]. In our system, we set  $K = 3$  (three Gaussians),  $\alpha = 0.005$ , and  $T = 0.4$ . We implement the method on both grayscale and RGB video inputs.

The mixture of Gaussians method is robust to slow lighting changes, periodical motions from clutter background, slow moving objects, long term scene changes, and camera noises. However, it cannot adapt to quick lighting changes and cannot handle shadows well. A number of techniques have been developed to improve the performance of the mixture of Gaussians method [10, 16-18, 34].

In order to make the mixture of Gaussians method work for quick lighting changes, we integrated the texture information to the foreground mask to remove the false positive areas by using the gradient features since the texture in the false positive foreground areas which are caused by lighting changes should be similar to the texture in the background, and the gradient value is less sensitive to lighting changes and is able to derive an accurate local texture difference measure. To remove the false foreground masks that are caused by shadows, the normalized cross-correlation of the intensities is calculated at each pixel of the foreground region between the current frame and the background image [34].

## B. Static Object Detection

Similar to Tian *et al.* [34], we model the background using three Gaussian mixtures and detect the static region without extra computational cost. Generally, the 1<sup>st</sup> Gaussian distribution shows the persistent pixels and represents the background image. The repetitive variations and the relative static regions are updated to the 2<sup>nd</sup> Gaussian distribution. The 3<sup>rd</sup> Gaussian represents the pixels with quick changes. As shown in equation (4), the  $(B+1)$ th mixture of Gaussians of the background model is used to detect if a foreground pixel belongs to the static region:

$$pixel \in \text{static region}, \text{ if } \omega_{B+1} > T. \quad (5)$$

In our system, we use three Gaussian mixtures. If the weight of the 2<sup>nd</sup> Gaussian for a pixel is larger than a threshold, the pixel belongs to the static region. The connected component process is performed for both foreground mask and the static region mask.

For the system implementation, we need to solve the following problems: (1) When to heal the static region which means when to push the static region to the background model (the 1<sup>st</sup> Gaussian distribution)? (2) How to adjust the model update rate for video streams with different frame rates? (3) How to reduce static region fragmentation?

**Static Region Healing:** Foreground fragments are usual for many background subtraction methods. In the mixture of Gaussians background subtraction (BGS) method, the different parts of a static region are often updated to the background model at different speeds based on the similarity of the pixel values between the static region and the background model. By pushing back the static region to the background model when the static region is biggest (i.e., before it starts shrinking), we can avoid the fragment of the foreground. To push the static region back to the background model, we reset the weight of the static region as the maximum weight which was defined in the program. The mean and variance of the 2<sup>nd</sup> Gaussian distribution is exchanged with the 1<sup>st</sup> Gaussian distribution for each pixel in the static region mask.

**Updating BGS models at a fixed rate for video streams with different frame rate:** most existing adaptive BGS methods update the background models based on input frames and a predefined update rate parameter. In this case, the background models are updated at different speeds for video streams with different frame rates although the parameter of the update rate is the same. In real surveillance systems which use live videos as inputs, the video frame rate often changes dramatically even for the same camera view due to multiple engines running on one machine and the complexity of the scenario. To detect abandoned objects and removed objects by the mixture of Gaussians method, the abandoned/removed time is directly related to the model update rate. To ensure stability from the time the object is abandoned or removed till the system detects the static region, we update BGS models based on time stamp instead of frame number.

**Setting two thresholds for foreground mask and static region mask:** In order to avoid static region fragments, we employ two different weight thresholds for foreground mask and static mask. In the mixture of Gaussians BGS method, the different parts of a static region are often updated to the background model at different speeds based on the similarity of the pixel values between the static region and the background model. Some pixels in the static region are often updated to the background model before the static region is healed. We use a lower weight threshold for the static mask and a higher threshold for the foreground mask. Dual thresholding has also been exploited by Boulton *et al.* [6] in the context of background modeling. More recently, Zhang *et al.* used this idea in a more general framework, arguing that “two thresholds are better than one” [43] for vision applications.

## V. ABANDONED AND REMOVED OBJECT DETECTION

After static regions are detected and healed (i.e., pushed into the background), we need to classify whether the healing corresponds to an abandoned or removed object event. In this section, we initially present a robust algorithm that classifies the static regions into abandoned or removed objects. Then we describe a method to reduce false static region classification based on the history of background regions.

### A. Heal Type Detection

Very few methods have been proposed in the literature to classify static regions into abandoned or removed objects. Existing techniques rely on the analysis of the intensity edges along the static region in the background image and the current frame [8, 37]. The intuition is that, in many cases, covering the background with an object will introduce more edges in the image due to the object boundaries (occluding contours). Based on this assumption, the static foreground region may be classified as abandoned object if the background image contains fewer edges than the current frame (along the static foreground blob) and conversely for removed items.

Although these methods work well for simple scenarios with a smooth background, *they are not suitable for complex environments involving crowds and occlusions.* Below we depict two key limitations that arise under these conditions:

- The edge energy assumption is clearly violated when the *background is cluttered* with many intensity edges.
- For scenes where the object is constantly occluded, it is possible that only *part of the object is healed*. In this case, the static region will not contain the occluding contours, potentially having fewer intensity edges.

The key insight of our method to solve these problems is to exploit the *surroundings* (i.e., context information) of the static blob to classify it into abandoned or removed object. In fact, the surrounding image information has rich features to infer what is inside the blob, as it has been demonstrated by the impressive results obtained by image inpainting techniques

[2].

Image inpainting can be used to “fill up” the static foreground blob so that the resulting image could be compared to the background image to determine the heal type (abandoned or removed). However, this operation is computationally expensive and may fail for large regions with complex texture patterns.

Rather than going from the surroundings to the interior of the blob as in inpainting, our strategy takes the opposite way. We start at the boundaries of the static blob and use a segmentation process to grow into the exterior, in order to verify how the static region is compatible with its surroundings. Our method is inspired in some sense by the work of Ramanan [26], which uses segmentation to verify object hypotheses in pattern classification.

Figure 2 illustrates the basic idea of our technique. Assume that an object was abandoned in a cluttered background. We first erode the static foreground region to make sure its boundaries fall completely inside the object. The boundaries of the eroded region are shown in dashed line in Figure 2(a). Then, we use these boundary points as seeds in a segmentation process. The arrows in the figure indicate the region growing direction. The result of this segmentation is shown in Figure 2(b). Note that the region growing stops at the boundaries of the object, leading to a smaller segmented region which is not compatible with its surroundings.

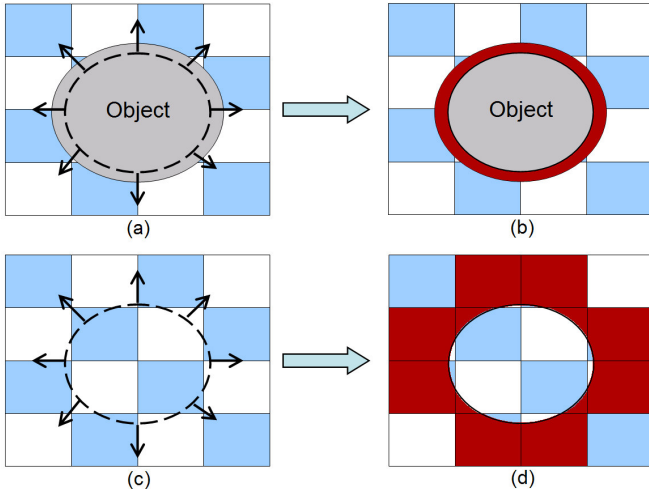


Figure 2: Heal type detection by region growing. (a) Object in a cluttered background. The dashed lines correspond to the eroded static region mask contour. (b) Segmented region after region growing. (c) The same region on the background image. (d) Segmented region after region growing, which is larger than the region in (b).

The same segmentation process is then applied in the background image, as shown in Figure 2(c). In this case, we can see that the resulting segmented region in Figure 2(d) is much larger, indicating compatibility with its surroundings.

The heal type is finally determined by just comparing the size of the two segmented regions. If the background segmentation is larger than the current frame segmentation, then the foreground region is classified as abandoned object. Otherwise, it is classified as a removed item. If the segmented regions have similar sizes, the heal type is set to “unclear”, which may occur when the static foreground blob corresponds

to lighting changes or other artifacts.

Our approach is simple to implement, runs in real-time, and is very reliable for real-world surveillance scenarios. It offers substantial improvement over previous edge-based methods in complex environments. Figure 3 shows a typical scene, where an object is left in a cluttered background. Note that the change in terms of edge energy (Figures 3c and 3d) is not a good feature to determine the heal type due to the background clutter. Figures 3e and 3f show the eroded mask overlaid in the current frame and the background, respectively. Finally, figures 3g and 3h show the segmented regions after the region growing process. Clearly, the segmented region in the background is larger than that of the current frame. As a result, the heal type is correctly determined as abandoned object.

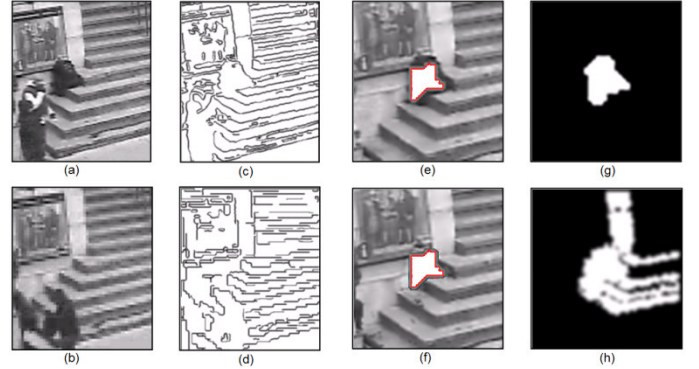


Figure 3: An example of an abandoned object is detected by our region growing method, while the edge energy method failed. (a) Current frame with an abandoned black bag. (b) Background image. (c) Edge map for the current frame. (d) Edge map for the background image. (e) Eroded mask overlaid in the current frame. (f) Eroded mask overlaid in the background image. (g) Resulting segmentation for the current frame. (h) Resulting segmentation for the background image. Heal type (abandoned) is determined by comparing the sizes of the two segmented regions in (g) and (h).

### B. History of Background Regions

In order to further improve the accuracy of the static region classification method described above, we keep a history of the background objects in the scene and use this information as an additional cue to classify static regions into abandoned and removed objects.

Consider an object entering the scene and becoming stationary. Before healing the object into the background, we store the corresponding background region for that object (i.e., the region behind the object) as history information (see Figure 4).

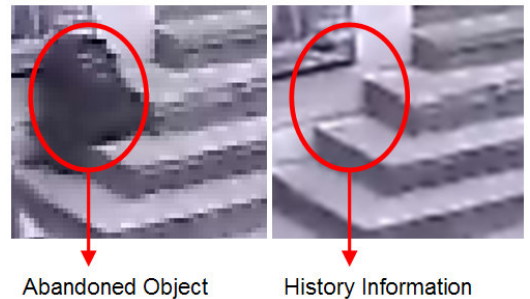


Figure 4: When an object is abandoned, the corresponding background region is stored prior to pushing the object into the background. This information can be used to improve heal type detection when the object is removed.



Now assume the object after being healed stays stationary for a while and then moves again, leaving the scene. At this moment, after the object is removed, we have a piece of background which will be healed. Note that the static region corresponding to this new healing can be directly classified as “removed object” by just comparing it with the history information and verifying that they are similar. In case they are not similar (or there is no corresponding stored history information), we invoke the heal type detection method described in the previous section. This way we avoid any problems with lighting changes when matching the history information due to objects that stay stationary for a long time.

## VI. ABANDONED/REMOVED OBJECT ALERT DETECTION

In this section, we describe the process of abandoned/removed object alert detection which includes 3 parts: 1) Human detection method, 2) system interface, and 3) occlusion handling by keeping track the abandoned/removed items during a time period specified by the user.

### A. Human Detection

In order to distinguish stationary human or non-human objects in the static regions, we developed a learning framework for human detection based on adaptive local features. This framework can be applied to detect humans in near-field, mid-field, and far-field surveillance scenarios, which deal with images with different levels of detail. In order to account for these differences, for each scenario we designed a human detector in a scale specifically tailored to the available resolution which can be configured for different camera views by users. Figure 5 shows the examples of near-field (left), mid-field (middle), and far-field (right) surveillance scenarios.



Figure 5: Examples of near-field (left), mid-field (middle), and far-field (right) surveillance scenarios.

**Near-field Human Detection:** In near-field surveillance videos (Figure 5-left), the resolution is sufficient to make facial features of people clearly visible. We developed a face detector and a tracking system using the learning method to detect people in near-field scenes [12]. To design the face detector, we used a frontal face dataset containing 4000 face images for training purposes. Each training image was cropped and rescaled to a 24x24 patch size. A pool of adaptive features was generated by running a non-linear optimization process, with different wavelet settings (wavelet type, frequency, etc.) for each sample. As a result, a pool of 80000 adaptive features was generated, containing a large variety of wavelet filters. It takes less than a second to create hundreds

of adaptive features for a particular 24x24 sample in a conventional 3GHz desktop computer.

For the second step of the algorithm (learning general features), we used an additional database of about 1000 background (non-face) images from which 24x24 patches are sampled. A cascade classifier was trained by considering 4000 faces and 4000 non-faces at each level, where the non-face samples were obtained through bootstrap [28]. Each level in the cascade was trained to reject about half of the negative patterns, while correctly accepting 99.9% of the face patterns. A fully trained cascade consisted of 24 levels. A Haar filter corresponding to the first 18 levels of the cascade was used in our experiments, in order to achieve real-time performance.

**Mid-field Human Detection:** In mid-field scenes, facial features may not be visible due to the poor resolution (Figure 5-middle). However, the lines that delimit the head and shoulders of an individual are still informative cues to find people in images. For these scenes, we developed a system for tracking and detection which locates people by scanning a window through the image and applying a head and shoulders detector at every position and scale [11]. This detector is designed according to the same learning framework based on Adaboost learning and optimized wavelet features. Similarly to the face detector for near-field scenarios, a training set of 4000 images containing the head and shoulders region was used for training.

**Far-field Person Detection:** In far-field imagery, pedestrians may appear as small as 30-pixels tall (Figure 5-right). In this scenario, the camera is known to be in a fixed position, making it feasible to use background modeling techniques to segment moving objects. In [8], we described how our far-field surveillance system classifies blobs obtained from background subtraction into one of three classes: cars, people and groups of people.

### B. System Interface

After a static region is healed and classified as an 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 as shown in Figure 6, 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);
- 3) **Abandoned/Removed 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.

In addition to the above conditions, the object class (e.g. human or non-human) can also be configured by users. If both human and non-human object classes are selected for abandoned and removed object detection, the human detection

process is skipped. The conditions based on size, object class, 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. Since we use the 2<sup>nd</sup> Gaussian distribution to detect the static regions, the time from the object has been abandoned/removed till it has been healed to the background model is determined by the model update rate, weight threshold, and the similarity of the object and the background models. This time is also counted in the alert detection.

In crowded scenes, the abandoned object (or the ghost due to object removal) may be constantly occluded. In the next section, we describe our technique to verify the persistence of a static region under occlusions.

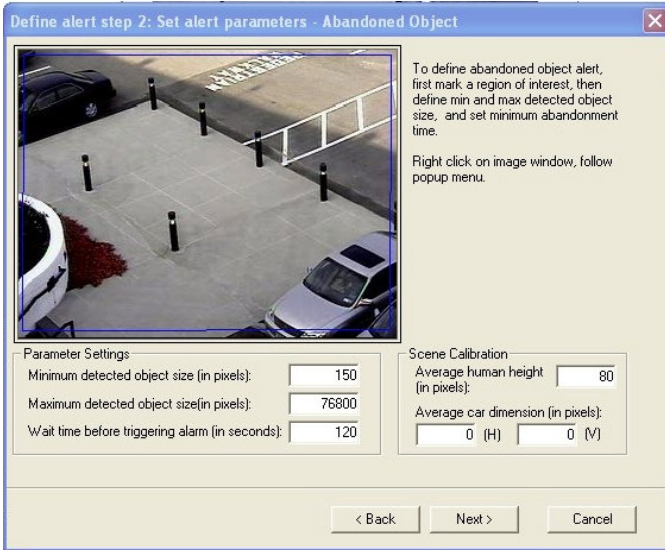


Figure 6: System interface and parameter configuration for abandoned and removed object detection.

### C. Matching under Occlusions

In order to verify the persistence of the abandoned and removed object in the scene during the time period specified by the user, we use the healed static region as a template and apply cross-correlation in each incoming frame to detect the object (or the ghost) at that specific image location. Occlusions are clearly a problem here, as they lead to low correlation scores.

Let  $StaticTimeThr$  be the time duration specified by the user and  $OccTimeThr$  be the maximum allowed continuous occlusion time.

After the static region is healed, in case the object is not detected (low correlation score) for a continuous time duration greater than  $OccTimeThr$ , we terminate the process and no alert is triggered.

In case the object is detected, we check whether the current time since the region became stationary is greater than  $StaticTimeThr$ , in which case we trigger the alert indicating an abandoned or removed item. This process handles occlusions

quite well in crowded environments, while meeting the user specified time conditions.

This matching process is also important to bring a spatial, region-based analysis into the pixelwise background adaptation model. Pixelwise adaptation is very useful for handling multimodal backgrounds (like waving trees, etc.), but may also lack higher-level information about the object shape. As an example, healing may occur if different objects with different shapes but same color frequently cross a specific image location. In this scenario, the region-based matching process is essential to eliminate false stationary regions.

### D. Complement with Tracking

In some complex scenarios of video surveillance, the false alarm rate could be high because of varying lighting, crowded traffic, cluttered background, weather condition, etc. As an example of the failed car bombing in Times Square at New York City, the police department would like to automatically detect illegal parking in a crowded urban environment all day long. As shown in Figure 11, our experimental results of the big city on-site test (4 cameras for 20 hours) demonstrate that the average false-alarm rate is about 2.5 false alarms per minute with a detection rate of 87.8%. The false-alarm rate can be adjusted by changing the user defined parameters, however, the detection rate decreases to 65.9% when the false-alarm rate is less than 0.5 false alarms per minute (see Figure 11.)

In order to keep the detection rate and decrease the false-positive rate, we employ an additional module by using tracking information. In the above application, we observe that genuine abandoned objects, i.e., the parked vehicles typically come into parked space with clear trajectories, while the false alarms due to random lighting variations do not associate with regular and smooth trajectories. Hence, instead of directly triggering an alarm using the heal type detection and matching process, we combine our algorithm with a tracking algorithm to further reduce the false-alarm rate.

After detecting a candidate abandoned object, we further verify the detection as follows using the tracker in [9], but in practice any tracking algorithms can be used. The tracker tracks all the moving objects and produces the corresponding trajectories. We query the tracker with the detected abandoned object in order to retrieve the trajectories associated with it. Let  $v$  be the speed of the tracked object computed from its trajectory over a period of  $W$  frames up to the current frame, and  $bCross$  a Boolean variable indicating whether the trajectory intersects the boundary of the region of interest. We trigger an alarm if  $bCross = true$  and  $v \geq SpeedThr$ , and reject the candidate otherwise.  $W$  and  $SpeedThr$  are user controllable parameters. In short, we check with the tracker to see if the candidate parked vehicle comes from outside the region of interest and travels at a reasonable speed.

Different from previous works [1, 4, 15, 19, 27, 31] that use tracking to solve the generic abandoned object detection problem, in this application, we only use tracking information to provide an additional cue to reduce the false-alarm rate. The

implication is that instead of using long-term trajectories, which are usually unreliable under complex environments, we consider them in a small temporal window and use them only as some of the criteria for the abandoned object detection, reducing the impact of spurious and noisy trajectories.

## VII. EXPERIMENTAL RESULTS

The proposed algorithm is being used in our real-time smart video surveillance system – IBM Smart Surveillance Solutions [35]. In this section, some examples and quantitative results demonstrate the effectiveness of our algorithm for abandoned/removed object detection in a variety of environments.

### A. PETS 2006 Dataset

We have tested our approach in the Pets 2006 dataset [41], which was designed to test abandoned object detection algorithms in a public space. The ground truth for the testing sequences include the number of persons and luggage involved in the event, and also spatial relationships between the luggage and person (to check whether the luggage is being attended or not). As we stated before, we just classify persistent foreground objects as abandoned items, without taking into consideration whether the object is being attended by a person or not.

TABLE I: ABANDONED OBJECT DETECTION FOR 7 PETS2006 SEQUENCES.

Sequence	abandoned objects	True Positives	Static Person	False Positives
S1	1	1	0	0
S2	1	1	0	0
S3	1	1	1	0
S4	1	1	0	0
S5	1	1	0	0
S6	1	1	0	0
S7	1	1	0	0

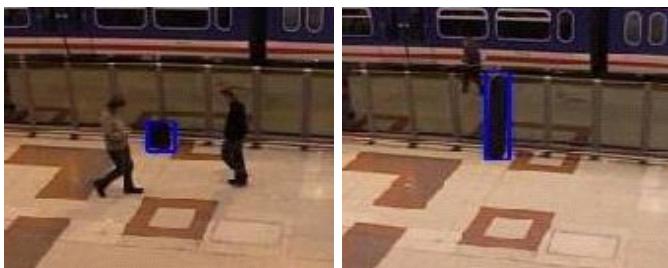


Figure 7: Sample images of detected abandoned object detection events in the PETS2006 dataset.

The Pets 2006 dataset consists of multi-sensor sequences containing left-luggage scenarios with increasing scene complexity. There are seven different scenarios captured by four cameras from different viewpoints. Since our algorithm is based on a single camera, we used just one of the camera views in each scenario, totalizing seven testing sequences. We chose the camera where the object appears bigger in the video. The whole image region is used to detect the abandoned objects. Table I shows our obtained results for the seven sequences. Figure 7 shows a sample image of a detected

abandoned object event. The scenarios are relatively simple, without many occlusions and crowds. Our algorithm detected all abandoned items, with zero false alarms. A static person is detected as an abandoned item in sequence S3. This could be removed by incorporating a person classifier.

TABLE II: ABANDONED OBJECT DETECTION FOR THE ILIDS DATASET ABANDONED BAGGAGE SCENARIO.

Sequence	Abandoned objects	True Positives	Static Person	True False Positives
AB Easy	1	1	0	0
AB Medium	1	1	3	1
AB Hard	4	4	3	1
ABTRA102a09	1	1	3	2
ABTRA204a06	1	1	0	0

TABLE III: PARKED VEHICLE DETECTION FOR THE ILIDS DATASET PARKED VEHICLE SCENARIO.

Sequence	Parked Vehicle	True Positives	False Positives
PV Easy	2	2	0
PV Medium	1	1	0
PV Hard	1	1	1
PV night	1	1	0
PVTRA101a022	1	1	0

### B. The i-LIDS Dataset

The i-LIDS video library provides a benchmark to facilitate the development and selection of video detection systems [42]. Our evaluation is based on two scenarios: abandoned baggage and parked vehicles. The abandoned baggage scenario contains alarm events of unattended bags on the platform of an underground station. The parked vehicles scenario contains alarm events of suspiciously parked vehicles in an urban setting. Figure 8 and 9 show some examples of the detected abandoned baggage and parked vehicles. Tables II and III show the details of the detection results. Unlike in paper [37], which only small regions are selected to detect the events, we use the whole camera view to detect the abandoned events. In both scenarios, we detected all the abandoned events (baggage and parked vehicles) with low false positives. Some static people are detected as abandoned items because we do not incorporate a person classifier. Note that a very small static water bottle is detected (the top-right image in Figure 8.)

### C. Results of Removed Objects

Both Pets and iLids datasets are for abandoned object detection. We also collected a dataset that includes removed object events in different situations (retail stores, parking lot, lab, conference room, etc.) with different sizes and types of the removed objects (a bottle of water, a book, a laptop, a car, etc.) Table IV shows the detection results. For a total of 12 removed objects, we detected 11 of them. One is missing because the pixels of the region are randomly updated to the background model, so the region after the object removed is



not detected as a static region. Figure 10 shows examples of a parked car and a laptop when they are removed.

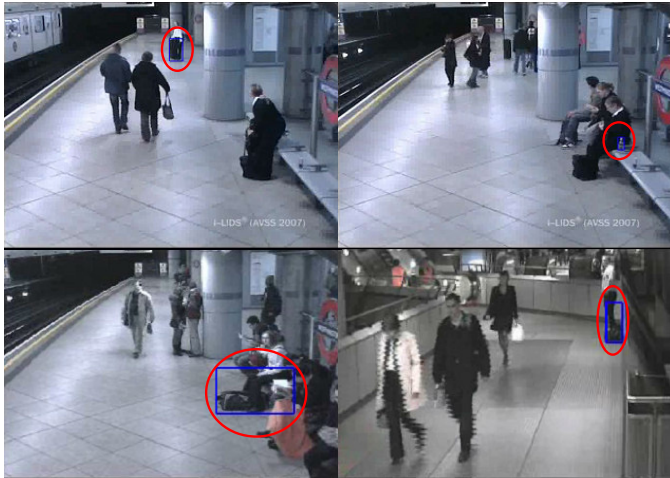


Figure 8: Examples of detected abandoned objects in the iLids dataset abandoned baggage scenario.



Figure 9: Examples of detected parked vehicles in the iLids dataset.

TABLE IV: DETECTION RESULTS FOR REMOVED OBJECT DETECTION

Removed Objects	True Positives	False Positives
12	11	0



Figure 10: Examples of detected removed objects in our dataset.

#### D. Big City On-site Test

Our system has been evaluated with repeated testing in a big city for public safety under very complex environments (crowded, raining, night, lighting change). For about 20 hours testing of four camera views which include scenarios of crowded, raining, daytime, and nighttime, there are in total 41 abandoned events. Our system detects 36 events, achieving an 87.8% detection rate. As shown in the system interface (Figure 6), the user can specify the alert requirements for different scenarios by changing parameters of “Minimum detected object size (in pixels)”, “Maximum detected object size (in pixels)”, and “Wait time before triggering alarm (in seconds).” Only for those abandoned/removed objects that meet the alert requirements that are defined by users will trigger the alerts.

In the test, the minimum size of the abandoned objects is changed from 75 – 700 pixels, the maximum size of the abandoned object is set as the pixels of whole image frame, and the abandoned time is set as 2 minutes. The interest region for alert detection is selected as the whole region of the images. Figure 11 displays the detection accuracy and false positive rate changes along the minimum object size by keeping other conditions same for our system on-site test in the big city. When the value of the minimum object size is bigger, both the detection accuracy and the false positives decrease. That means that abandoned objects with smaller sizes could be missed. However, a lower cost of assessment and response is needed from the security office.

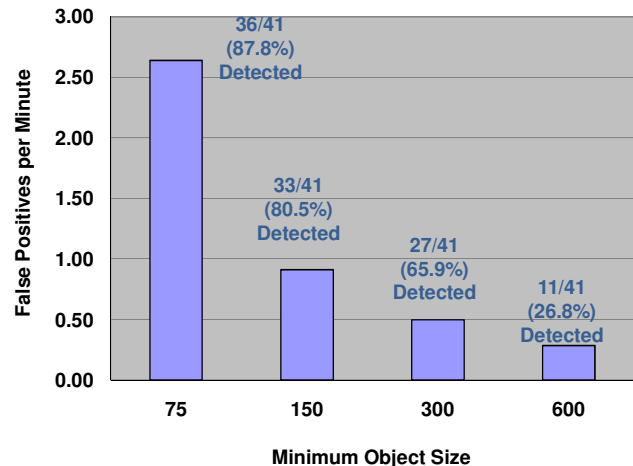


Figure 11: False positives and accuracy change based on minimum object size.

#### E. Complement with Tracking

To show the effectiveness of leveraging tracking information, we compare the performances of our system with/without using the tracking algorithm. The evaluations are done in a big city to detect illegally parked vehicles under complex environments where lighting varies from bright daytime to fluorescent nighttime, weather from sunny to raining, and camera angle from frontal to profile. The region of interest is set to the area where parking is not allowed. The sizes of abandoned objects are set to 500 pixels and the abandoned time is set to 5 minutes.

Table V shows the comparison of the parked vehicle detection results on video sequences of 53 hours long in total. The false-alarm rate is reduced from 44.5% to 20.7%, a  $\sim 25\%$  reduction by incorporating the trajectories. The improvement is even more significant during nighttime where the interplay between the headlights of vehicles and street lights causes a lot more false positives. Table VI shows the detection results on nighttime video sequences of 28 hours in total. Note that without further verification, the false positives outnumber the true positives, rendering the system uninformative. Figure 12 shows examples of detected parked vehicles under different lighting conditions.



Figure 12: Examples of detected parked vehicles. The dashed rectangle indicates user defined region of interest.

TABLE V: COMPARISON OF PARKED VEHICLE DETECTION WITH AND WITHOUT USING TRACKING INFORMATION FOR REAL WORLD DEPLOYMENT (OVERALL)

Approaches	Parked Vehicle	True Positives	False Positives
Detection w/o Trajectory Verification	81	66	53
Detection with Trajectory Verification	81	61	16

TABLE VI: COMPARISON OF PARKED VEHICLE DETECTION WITH AND WITHOUT USING TRACKING INFORMATION FOR REAL WORLD DEPLOYMENT (NIGHTTIME)

Approaches	Parked Vehicle	True Positives	False Positives
Detection w/o Trajectory Verification	4	4	23
Detection with Trajectory Verification	4	2	4



Figure 13: Examples of typical false positives and false negatives.

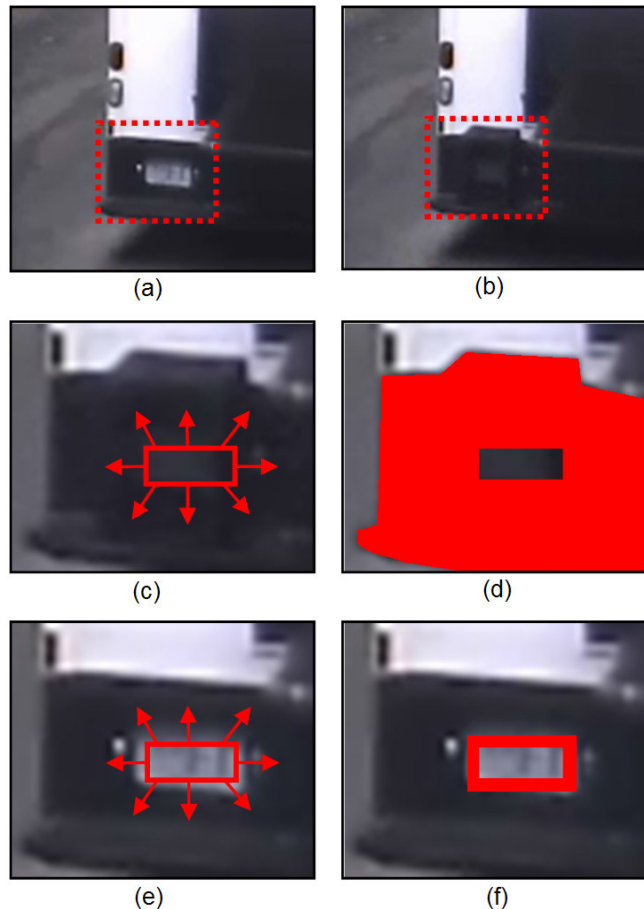


Figure 14: One example of wrong heat type detection. (a) Background image. (b) Current image with abandoned object covering license plate. (c) Erode mask overlaid in current image. (d) Resulting segmentation for current image. (e) Eroded mask overlaid in background image. (f) Resulting segmentation for background image.

## F. Limitations

The accuracy of abandoned object detection is influenced by several factors:

- Foreground fragments due to imperfect background subtraction (see Figure 13-left example).
- The size of the abandoned object is too small or the abandoned object is occluded (see Figure 13-right example).
- Adverse weather conditions like rain and snow cause higher false-alarm rates.
- Low light conditions reduce the ability to discern one object from another, causing higher error rates.
- Static object detection in extremely crowded scenes is much more difficult, leading to higher error rates.



- Quick lighting changes cause problems to detect abandoned or removed objects. Although our background model adapts to quick lighting changes, there are few cases where significant and quick illumination changes occur after an object has been abandoned, but before the alarm has been triggered. In this situation, the whole background model is updated with the abandoned item, which cannot be detected. If the lighting change is just temporary, then our system is able to recover using the previous background model. False negatives in this scenario occur only when the change is persistent.
- Low contrast situations, e.g., black bag on a black background, lead to missed detections (see example in Figure 14). Our heal type detection method (for classifying whether an object was removed or abandoned) achieves much better results than previous approaches based on edge energy analysis. However, it may fail in situations where the color of the object is very similar to the background. Figure 14 shows a black bag abandoned in a black background. In this case, the segmentation process applied in the image containing the object does not stop in the object boundaries, but leaks over all the background. The resulting segmented region may be similar to the segmentation applied in the background image, making the heal type decision unclear. A possible solution to this problem is to use multispectral imaging to accentuate the contrast between the object and the background when they have the same color.

#### VIII. DISCUSSION AND CONCLUSION

We have presented a new framework to robustly and efficiently detect abandoned and removed objects in complex environments for real-time video surveillance. The mixture of Gaussians background subtraction method is employed to detect both background and static foregrounds by using the same Gaussian mixture model. Then the static foregrounds were classified into abandoned or removed objects by segmenting and comparing the surrounding areas of the background model and the foreground image. Our method can handle occlusions in complex environments with crowds. Furthermore, in order to reduce false alarms, we have employed tracking information in a small temporal window to provide an additional cue to filter out the impact of spurious and noisy trajectories for abandoned object detection. The testing results which are based on different scenarios have proved that our approach can be successfully applied in real-world surveillance applications.

#### ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their constructive comments and suggestions that help to improve the quality of this manuscript.

#### REFERENCES

- [1] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, and J. Meunier, "Left-luggage detection using homographies and simple heuristics," in PETS, 2006, pp. 51–58.
- [2] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image inpainting", Proceedings of SIGGRAPH 2000, New Orleans, USA, July 2000.
- [3] A. Bevilacqua and S. Vaccari, "Real Time Detection of Stopped Vehicles in Traffic Scenes", IEEE International Conference on Advanced Video and Signal-Based Surveillance, London , UK , September 2007
- [4] M. Beynon, D. Hook, M. Seibert, A. Peacock, and D. Dudgeon, "Detecting Abandoned Packages in a Multi-camera Video Surveillance System", IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2003
- [5] N. Bird, S. Atev, N. Caramelli, R. Martin, O. Masoud, N. Papanikolopoulos, "Real Time, Online Detection of Abandoned Objects in Public Areas", IEEE International Conference on Robotics and Automation, May, 2006.
- [6] T. Boulton, R. Micheals, X. Gao, P. Lewis, C. Power, W. Yin, and A. Erkan, "Frame-rate Omnidirectional Surveillance and Tracking of Camouflaged and Occluded Targets", Workshop on Visual Surveillance, 1999.
- [7] S. Boragno, B. Boghossian, J. Black, D. Makris, and S. Velastin, "A DSP-based system for the detection of vehicles parked in prohibited areas", IEEE International Conference on Advanced Video and Signal-Based Surveillance, London , UK , September 2007
- [8] L. Chen, R. Feris, Y. Zhai, L. Brown, and A. Hampapur, An integrated system for moving object classification in surveillance videos. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, 2008.
- [9] J. Connell, A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti, "Detection and Tracking in the IBM PeopleVision System.", in IEEE ICME, June 2004.
- [10] H. Eng, J. Wang, A. Kam, and W. Yau, "Novel Region-based Modeling for Human Detection within High Dynamic Aquatic Environment," Proceedings of CVPR, 2004.
- [11] R. Feris, Y. Tian and A. Hampapur. "Capturing People in Surveillance Video," IEEE International Workshop on Visual Surveillance, 2007.
- [12] R. Feris, Y. Tian, Y. Zhai, and A. Hampapur, "Facial Image Analysis Using Local Feature Adaptation Prior to Learning", IEEE Int'l Conf. on Automatic Face and Gesture Recognition, Amsterdam, 2008.
- [13] S. Ferrando, G. Gera, C. Regazzoni, "Classification of Unattended and Stolen Objects in Video Surveillance System", IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), 2006
- [14] S. Guler and M. K. Farrow, "Abandoned object detection in crowded places," in PETS, 2006, pp. 99–106.
- [15] S. Guler, J. A. Silverstein, and I. H. Pushee, "Stationary Objects in Multiple Object Tracking", IEEE International Conference on Advanced Video and Signal-Based Surveillance, London , UK , September 2007
- [16] M. Harville, "A Framework for High-level Feedback to adaptive, per-pixel, Mixture-of-Gaussian Background Models", Proceedings on ECCV, 2002.
- [17] O. Javed, K. Shafique, and M. Shah, "A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information," IEEE Workshop on Motion and Video Computing, 2002.
- [18] P. KaewTraKulPong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," In Proc. 2<sup>nd</sup> European Workshop on Advanced Video Based Surveillance Systems, 2001.
- [19] P. T. N. Krahnstoever, T. Sebastian, A. Perera, and R. Collins, "Multiview detection and tracking of travelers and luggage in mass transit environments," in PETS, 2006, pp. 67–74.
- [20] Ser-Nam Lim, Larry Davis, "A One-Threshold Algorithm for Detecting Abandoned Packages Under Severe Occlusions Using a Single Camera", CS-TR-4784, CS dept., Feb 13 2006, University of Maryland, College Park
- [21] S. Lu, J. Zhang, D. Feng, Detecting unattended packages through human activity recognition and object association, PR(40), No. 8, August 2007, pp. 2173-2184.
- [22] S. Lu, J. Zhang, D. Feng, An efficient method for detecting ghost and left objects in surveillance video, AVSBS07 pp. 540-545.

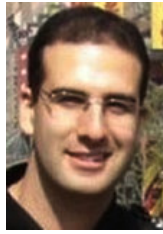
- [23] M. Magno, F. Tombari, D. Brunelli, L. Di Stefano, L. Benini, Multimodal Abandoned/Removed Object Detection for Low Power Video Surveillance Systems, AVSBS09 (188-193).
- [24] F. Porikli, "Detection of Temporarily Static Regions by Processing Video at Different Frame Rates", IEEE International Conference on Advanced Video and Signal-Based Surveillance, London, UK, September 2007
- [25] F. Porikli, Y. Ivanov, and T. Haga, Robust Abandoned Object Detection Using Dual Foregrounds, EURASIP Journal on Advances in Signal Processing, Volume 2008, Hindawi Publishing Corporation.
- [26] D. Ramanan, "Using Segmentation to Verify Object Hypotheses", Proceedings of CVPR, 2007.
- [27] J. M. del Rincón, J. E. Herrero-Jaraba, J. R. Gomez, and C. Orrite-Urunuela, "Automatic left luggage detection and tracking using multi cameras," in PETS, 2006, pp. 59–66.
- [28] H. Rowley, S. Baluja, T. & Kanade, (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(1), 23–38.
- [29] C. Sacchi and C. Regazzoni, "A Distributed Surveillance System for Detection of Abandoned Objects in Unmanned Railway Environments," IEEE Transactions on Vehicular Technology, Vol. 49, Nov. 5, Sept. 2000.
- [30] V. K. Singh, P. K. Atrey, and M. S. Kankanhalli. "Cooperative multi-camera surveillance using model predictive control". Springer Journal of Machine Vision and Applications, 19(5-6):375-393, October 2008.
- [31] K. Smith, P. Quelhas, and D. Gatica-Perez, "Detecting abandoned luggage items in a public space," in PETS, 2006, pp. 75–82.
- [32] M. Spengler and B. Schiele, "Automatic Detection and Tracking of Abandoned Objects", IEEE International Workshop on VSPETS 2003.
- [33] C. Stauffer and W.E.L. Grimson, "Adaptive Background mixture Models for Real-time Tracking", CVPR99, June, 1999.
- [34] Y. Tian, Max Lu, and Arun Hampapur, "Robust and Efficient Foreground Analysis for Real-time Video Surveillance," IEEE CVPR, San Diego, June, 2005.
- [35] Y. Tian, A.W. Senior, A. Hampapur, L. Brown, C. Shu, and M. Lu, "IBM Smart Surveillance System (S3): Event Based Video Surveillance System with an Open and Extensible Framework", Machine Vision and Applications, 19:315-327, 2008.
- [36] Y. Tian, R. S. Feris and A. Hampapur. "Real-Time Detection of Abandoned and Removed Objects in Complex Environments". IEEE International Workshop on Visual Surveillance (in conjunction with ECCV'08), Marseille, France, 2008
- [37] P. L. Venetianer, Z. Zhang, W. Yin, A. J. Liptop. "Stationary Target Detection Using the ObjectVideo Surveillance System", IEEE International Conference on Advanced Video and Signal-Based Surveillance, London, UK, September 2007
- [38] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Kauai, Hawaii, 2001.
- [39] P. Viola, M. Jones and D. Snow, Detecting Pedestrians Using Patterns of Motion and Appearance. In *IEEE International Conference on Computer Vision (ICCV'03)* (Vol. 2, pp. 734-741), 2003.
- [40] J. Wang and W. Ooi. Detecting static objects in busy scenes. Technical Report TR99-1730, Department of Computer Science, Cornell University, February 1999.
- [41] PETS 2006 Benchmark Data, <http://www.cvg.rdg.ac.uk/PETS2006/data.html>.
- [42] i-LIDS Dataset for AVSS 2007, <ftp://motinas.elec.qmul.ac.uk/pub/iLids>.
- [43] T. Zhang, T. Boulton, and R. Johnson, "Two Thresholds are Better than One", Visual Surveillance Workshop, Minneapolis, MN, 2007.
- [44] H. Zhao, H. Yang, S. Zheng, H. Liao, J. Chang, L. Chen, A Localized Approach to Abandoned Luggage Detection with Foreground-Mask Sampling, AVSBS08 (132-139).



**YingLi Tian** (M'99–SM'01) received her BS and MS from TianJin University, China in 1987 and 1990 and her PhD from the Chinese University of Hong Kong, Hong Kong, in 1996. After holding a faculty position at National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, she joined Carnegie Mellon University in 1998, where she was a postdoctoral fellow of the Robotics Institute. Then she worked as a research staff member in IBM T. J. Watson Research Center from

2001 to 2008. She is one of the inventors of the IBM Smart Surveillance Solutions.

She is currently an associate professor in Department of Electrical Engineering at the City College of New York. Her current research focuses on a wide range of computer vision problems from motion detection and analysis, to human identification, facial expression analysis, and video surveillance. She is a senior member of IEEE.



**Rogerio Schmidt Feris** received a PhD in computer science from the University of California, Santa Barbara, an MS in computer science from University of Sao Paulo, Brazil, and a BS in computer engineering from the Federal University of Rio Grande, Brazil, in 2006, 2001, and 1998, respectively.

He is currently a research scientist at IBM T. J. Watson Research Center, New York, and an Affiliate Assistant Professor at University of Washington. His research interests span a wide range of topics in computer vision, graphics, and machine learning, with emphasis on visual surveillance, intelligent user interfaces, and digital photography applications.

Dr. Feris received several awards and distinctions including IBM Emerging Leader in Multimedia in 2005 and Best Computer Science MSc Thesis in Brazil-2<sup>nd</sup> Prize in 2002. He is currently working on the IBM Smart Surveillance System, which received a Frost & Sullivan Award in 2008.



**Haowei Liu** received the BSc degree from National Taiwan University in 2001 and the MSc degree from the University of California, San Diego in 2005, both in computer science. He was a software design engineer in Microsoft from 2005 to 2006 and is currently pursuing his PhD degree in the University of Washington, Seattle. He has interned with and was supported by the IBM T.J. Watson Research Center and the Intel Seattle Lab.



**Arun Hampapur** is an IBM Distinguished Engineer and Director in the Business Analytics and Math Sciences Department of IBM Research. In his current role, he is leading solution development and commercialization of Business Analytics and Optimization technologies across a broad range of industries. His role includes developing novel analytics offerings which combine structured and unstructured information analytics technologies into vertically integrated solutions.

In 2008 he was on assignment to IBM Global Services as the CTO of Physical Security, where he led the technology commercialization of the IBM Smart Surveillance System. He led the original research team which invented IBM Smart Surveillance System (S3) at IBM T.J. Watson Research Center prior to its commercialization. He has led the S3 effort from its inception as an exploratory research effort, thru the building of the first lab prototype, first customer pilot engagement, to the commercialization as a services offering. He has developed several algorithms for video analytics and video indexing.

Dr. Hampapur has published more than 80 papers on various topics related to video analysis, pattern recognition, searchable video and video surveillance and holds 20 US patents and more than 70 patent applications. He is a member of the IBM Academy of Technology, an IBM Master Inventor, and an IEEE Senior Member. He obtained his PhD from the University of Michigan in 1995.

**Ming-Ting Sun** (S'79–M'81–SM'89–F'96) received the B.S. degree from National Taiwan University in 1976, the M.S. degree from University of Texas at Arlington in 1981, and the Ph.D. degree from University of California, Los Angeles in 1985, all in electrical engineering.



He joined the University of Washington in August 1996 where he is a Professor. Previously, he was the Director of the Video Signal Processing Research Group at Bellcore. He was a chaired professor at TsingHwa University at Beijing, and a visiting professor at Tokyo University and National Taiwan University. He holds 11 patents and has published over 200 technical papers, including 13 book chapters in the area of video and multimedia technologies. He co-edited a book "Compressed Video over



Networks” (Marcel Dekker, New York). He was the Editor-in-Chief of the *IEEE Transactions on Multimedia (TMM)* and a Distinguished Lecturer of the Circuits and Systems Society from 2000 to 2001. He received an IEEE CASS Golden Jubilee Medal in 2000, and was the general co-chair of the Visual Communications and Image Processing 2000 Conference. He was the Editor-in-Chief of the *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* from 1995 to 1997. He received the *TCSVT* Best Paper Award in 1993. From 1988 to 1991, he was the chairman of the IEEE CAS Standards Committee and established the IEEE Inverse Discrete Cosine Transform Standard.

Dr. Sun received an Award of Excellence from Bellcore for his work on the digital subscriber line in 1987. He is a Fellow of IEEE.