Real-Time Detection of Abandoned and Removed Objects in Complex Environments

Ying-li Tian, Rogerio Feris, Arun Hampapur
IBM T.J. Watson Research Center
{yltian,rsferis,arunh}@us.ibm.com

Abstract

We present a new framework to robustly and efficiently detect abandoned and removed objects in complex environments for real-time video surveillance. In our system, the background is modeled by a mixture of Gaussians. Similar to Tian et al. [18], this mixture model is employed to detect the static foreground regions (i.e., static blobs potentially corresponding to abandoned or removed objects) without extra computation cost. Several improvements are implemented to the background subtraction method for shadow removal, quick lighting change adaptation, reduction of fragmented foreground regions, and stable background update rate for video streams with inconsistent frame rates. Then, the types of the static regions (either abandoned or removed) are determined by using a method that exploits context information about the static foreground masks, significantly outperforming previous edge-based techniques. Based on the type of the static regions and several user-defined parameters, a matching method is proposed to trigger alerts indicating abandoned and removed objects. Our method can handle occlusions in complex environments with crowds. The robustness and efficiency of the method was tested on our real time video surveillance system for public safety application in big cities and evaluated by several public databases such as i-Lids and PETS2006 datasets.

1. Introduction

Many methods have been recently proposed to automatically detect abandoned objects (parked vehicles and left-luggage) in video surveillance [1-9, 11-16, 18-20] for different applications such as traffic monitoring, public safety, retail, etc. Although some efforts have been made to establish some standards (e.g., Pets and iLids), the problem is very challenging because it is not well-defined. For example, Beynon et al. [3] defined an abandoned package as any stationary package away from anyone considered responsible for it. Bird et al. [4] 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. [7] defined an abandoned object as a static "non-human" object which splits from a "human". All above definitions cannot cover the complex situations in real life. For example, a car/truck is abandoned and then the driver leaves, or someone just throw a bag from long distance. Also, in very crowded environments, it is difficult to detect the relationship of the abandoned object and the owner.

In this paper, we simply 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 the abandoned and removed objects, we focus on how to detect static regions in the scene and how to determine whether they correspond to abandoned or removed objects.

1.1 Related Work

Most of the proposed techniques for abandoned object detection rely on tracking information [1, 3, 9, 11, 14, 16] to detect drop-off events, while fusing information from multiple cameras. As stated by Porikli [13], 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 [13] proposed a single camera, non-tracking-based system which makes use of two backgrounds for 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 stationary 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 [19] 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. [6].

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 complex environment applications.

1.2. System Overview

In this paper, we propose a novel solution to detect abandoned and removed objects. Fig. 1 shows our system diagram. The system includes three main components: (a) background subtraction and static region detection; (b) object type detection (abandoned or removed); (c) abandoned and removed object alert detection. A matching algorithm is employed to detect if the object is abandoned long enough to trigger the alert. 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 (for moving objects) and the static region mask (for stationary objects). 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 segmentation method is developed to detect the type of the static region (abandoned or removed), significantly outperforming previous edge-based techniques. Only those abandoned/removed objects that meet the user-defined alert requirements will trigger the alerts.

This paper is organized as follows: in section 2, we describe the first component of our system (background subtraction and static region detection). The second and third components (object type and alert detection) are then presented in section 3. Finally, section 4 covers our experimental results on standard datasets as well as other real-world surveillance scenarios.

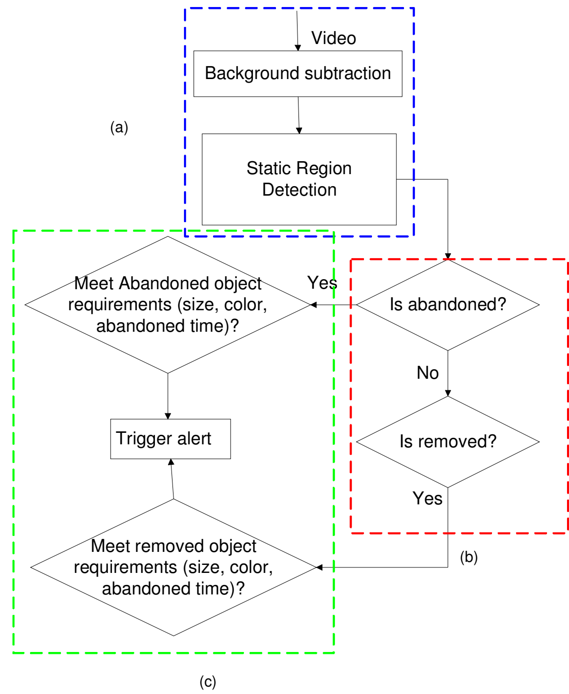

Figure 1: System diagram: (a) background subtraction and static region detection; (b) object type detection (abandoned or removed); (c) abandoned and removed object alert detection.

2. Static Region Detection

In this section, we describe how to detect static regions in the scene, which in general correspond to abandoned or removed objects. By static regions we mean those regions detected by background subtraction (i.e., foreground blobs) that remain stationary in the scene for a relatively long time.

2.1 Multi-Gaussian Adaptive Background Model and Improvements

Stauffer and Grimson [17] introduced a mixture of K

Gaussians (usually K is from 3 to 5) to build the background model and detect the moving objects. 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. But it cannot adapt to the 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 [18, 25- 27].

In order to make the mixture of Gaussians method work for quick lighting changes, we integrated texture information into the foreground mask computation (see [18] for details). The intuition is that false positive areas caused by lighting changes are usually similar to the texture in the background. We used a texture similarity measure based on gradient features to eliminate spurious foreground regions caused by quick lighting changes. 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.

2.2 Static Region Detection

Similar to Tian et al.[18], we model the background using a mixture of three Gaussians for each pixel in the image and detect the static regions without extra computation cost. Generally, the 1st Gaussian distribution shows the persistent pixels and represents the background image. The repetitive variations and the relative stationary regions are updated to the 2nd Gaussian distribution. The 3rd Gaussian represents the pixels with quick changes. In our system, if the weight of the 2nd Gaussian for a pixel is larger than the 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 1st Gaussian distribution)? (2) How to adjust the model update rate for video streams with inconsistent frame rate? (3) How to reduce static region fragmentation?

Static Region Healing: the static regions are healed (pushed into the background) when the area of the static region is biggest, i.e., before it starts shrinking. To push the static region to the background model, we reset the weight of the static region pixels as the maximum weight that was defined in the program. The mean and variance of the 2nd Gaussian distribution is exchanged with the 1st Gaussian distribution for each pixel in the static region mask.

Updating background models at a fixed rate for video streams with inconsistent frame rate: most existing adaptive background subtraction 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 speed for video streams with different frame rates, although the update rate parameter is the same. In real surveillance systems, 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 the background mixture models based on time instead of frame.

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 background subtraction 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, therefore causing fragmentation. We use a lower weight threshold for the static mask and a higher threshold for the foreground mask to avoid this problem. Dual thresholding has also been exploited by Boult et. al [23] 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” [24] for vision applications.

3. Abandoned / Removed Object Detection

After static regions are detected and healed (i.e., pushed into the background), we need to classify whether the static region corresponds to an abandoned or removed object event. In this section, we initially present a simple, yet quite robust algorithm that classifies the static regions into abandoned or removed objects. Then we describe our system interface and the process which keeps track of the abandoned/removed items under occlusions during a time period specified by the user.

3.1 Static Region 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 [6, 19]. 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 less 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.

Image inpainting could be used to "fill up" the static foreground blob so that the resulting image could be compared to the background image to determine the static region 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 [28], who 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 standard region growing algorithm. The arrows in the figure indicate the region growing direction. The result of this segmentation is shown in red color 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.

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.

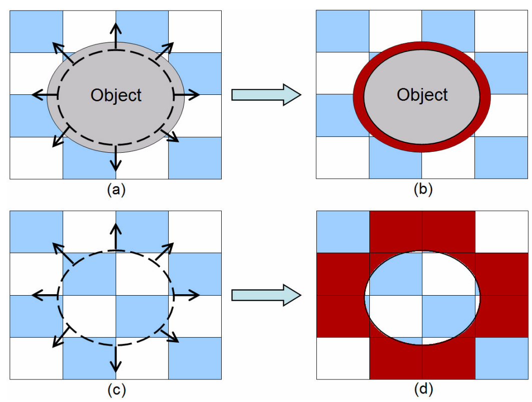

Figure 2: Static region type detection by region growing. (a) Object in a cluttered background. The dashed lines correspond to the eroded static region mask contour. (b) Segmented region after region growing. (c) The same region on the background image. (d) Segmented region after region growing, which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

The static region 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 static region 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 it 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 static region 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 static region type is correctly determined as abandoned object.

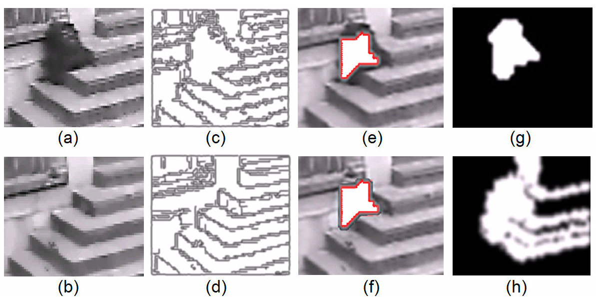

Figure 3: An example of an abandoned object is detected by our region growing method, while edge energy method failed. (a) Current frame with an abandoned black bag. (b) Background image. (c) Edge map for current frame. (d). Edge map for background image. (e) Eroded mask overlaid in current frame. (f)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

3.2 System Interface

After a 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. Since we use the 2nd 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 to the alert detection.

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

3.3 Matching under Occlusions

In order to verify the persistence of the abandoned / 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 object shape. As an example, healing may occur if different objects with different shape but same color frequently cross a specific image location. In this scenario, the region-based matching process is essential to eliminate false stationary regions.

4. Experimental Results

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

PETS 2006 Dataset

We have tested our approach in the Pets 2006 dataset [21], 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.

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 1 shows our obtained results for seven sequences. Figure 4 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.

# of sequences	abandoned objects	True Positives	Static Person	False Positives
7	7	7	1	0

Table 1 – Abandoned object detection for seven Pets2006 sequences.

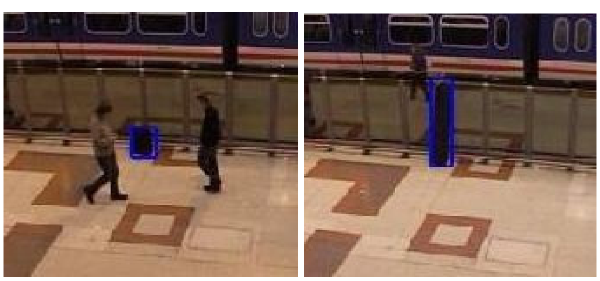

Figure 4 – Sample images of detected abandoned object detection events in PETS2006 dataset.

The i-LIDS Dataset

The i-LIDS video library provides a benchmark to facilitate the development and selection of video detection systems [22]. 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 5 and 6 show some examples of the detected abandoned baggage and parked vehicles. Tables 2 and 3 show the details of the detection results. Unlike in paper [19], 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 (baggages and parked vehicles) with a short number of 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 5.)

# of sequences	Abandoned objects	True Positives	Static Person	False Positives
5	8	8	9	4

Table 2 – Abandoned object detection for iLids dataset abandoned baggage scenario.

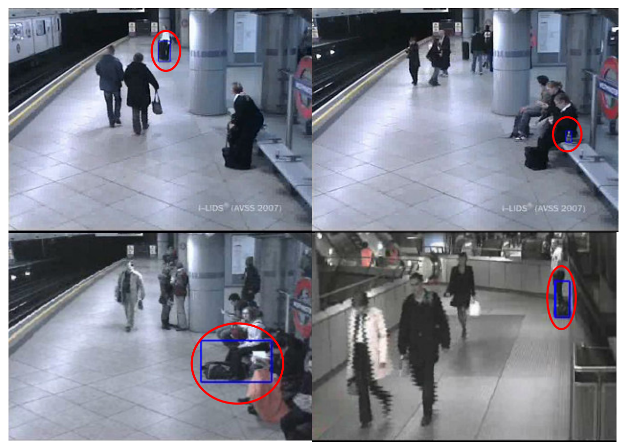

Figure 5 – Examples of detected abandoned objects in iLids dataset abandoned baggage scenario.

# of sequences	Parked Vehicle	True Positives	False Positives
5	6	6	1

Table 3 – Parked vehicle detection for iLids dataset parked vehicle scenario.

Results of Removed Objects

Since both Pets and iLids datasets are for abandoned object detection, we 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, book, laptop, car etc.) Table 4 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 7 shows examples of a parked car and a laptop when they are removed.

Figure 6 – Examples of detected parked vehicles in iLids dataset

Removed Objects	True Positives	False Positives
12	11	0

Table 4 – Detection results for removed object detection

Figure 7 – Examples of detected removed objects.

Big City On-site Test

Our system has been tested in a big city for public safety in a very complex environment (crowded, raining, night, lighting change). We cannot show sample image results due to the confidential agreement. For about 20 hours testing of 4 camera views which include scenarios of crowded, raining, daytime, and nighttime, there are in total 32 abandoned events. Our system detects 28 events, achieving 87.5% detection rate, with very few false positives. The sizes of the abandoned objects are from 75 – 700 pixels, and the abandoned time is longer than 2 minutes. We apply the alert detection for the whole region of the images.

Limitations

In the current version of our system, we do not have an object classifier to distinguish different types of objects in the scene. This means that a person who is stationary for a long time can be detected as a left behind item. In some circumstances, this can be an indication of a suspicious behavior, but in many cases these events are false alarms.

Lighting changes may also 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 can not 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.

Our static region type detection method (for classifying whether an object was removed or abandoned) achieves much better results than previous approaches based on edge energy analysis. But it may fail in situations where the color of the object is very similar to the background. Figure 8 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 static region type decision unclear. A possible solution to this problem is to use multispectral imaging to accentuate the contrast between object and background when they have the same color.

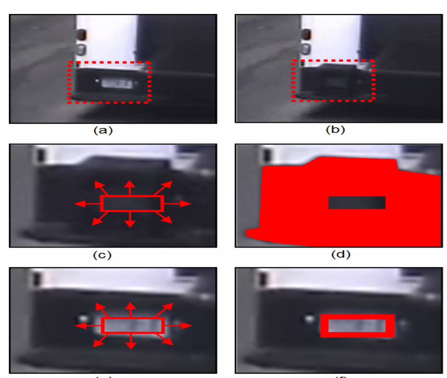

Figure 8: One example of wrong static region type detection. (a) Background image. (b) Current image with abandoned object covering license plate. (c) Erode mask overlaid in current image. (d) Resulting

segmentation for current image. (e) Eroded mask overlaid in background image. (f) Resulting segmentation for background image.

5. Discussion and Conclusion

We 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. Without using any tracking or motion information, static objects were detected by using the same Gaussian mixture model, and then 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. The testing results based on different scenarios proved that our approach is superior to previous methods and can be successfully applied in real-world surveillance scenarios.

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] 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

[3] 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

[4] 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.

[5] 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

[6] 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.

[7] 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

[8] S. Guler and M. K. Farrow, "Abandoned object detection in crowded places," in PETS, 2006, pp. 99–106.

[9] 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

[10] A. Hampapur, L. Brown, J. Connell, Max Lu, Hans Merkl, S. Pankanti, A.W. Senior, Chiao-fe Shu, and Y.-L. Tian, "Multi-scale Tracking for Smart Video Surveillance," IEEE Transactions on Signal Processing, 22: (2), March 2005.

[11] 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.

[12] 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

[13] 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

[14] J. M. del Rincn, 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.

[15] 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, Spet. 2000.

[16] K. Smith, P. Quelhas, and D. Gatica-Perez, "Detecting abandoned luggage items in a public space," in PETS, 2006, pp. 75–82.

[17] C. Stauffer and W.E.L. Grimson, "Adaptive Background mixture Models for Real-time Tracking", CVPR99, June, 1999.

[18] Ying-Li Tian, Max Lu, and Arun Hampapur, "Robust and Efficient Foreground Analysis for Real-time Video Surveillance," IEEE CVPR, San Diego. June, 2005.

[19] 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

[20] J. Wang and W. Ooi. Detecting static objects in busy scenes. Technical Report TR99-1730, Department of Computer Science, Cornell University, February 1999.

[21] PETS 2006 Benchmark Data, http://www.cvg.rdg.ac.uk/PETS2006/data.html

[22] i-LIDS Dataset for AVSS 2007, ftp://motinas.elec.qmul.ac.uk/pub/iLids.

[23] T. Boult, 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.

[24] T. Zhang. T. Boult, and R. Johnson, "Two Thresholds are Better than One", Visual Surveillance Workshop, Minneapolis, MN, 2007.

[25] 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.

[26] M. Harville, "A Framework for High-level Feedback to adaptive, per-pixel, Mixture-of-Gaussian Background Models", Proceedings on ECCV, 2002.

[27] 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.

[28] D. Ramanan, "Using Segmentation to Verify Object Hypotheses", Proceedings of CVPR, 2007.