FPGA Accelerated Abandoned Object Detection

$Rajesh\,Rohilla^{*}$, Aman Raj, Saransh Kejriwal, Rajiv Kapoor
Department of Electronics & Communication Engineering
Delhi Technological University, Delhi, India

***Abstract*—In this paper, a method to address the problem of detecting abandoned object(s) in a live video feed has been presented. The proposed technique utilizes a static background modeling algorithm and identifies any object lying abandoned for a given period of time. Our approach uses a FIFO queue for each pixel in a video frame as an essential element and applies a processing logic. Since the serial processing of such pixel queues on a conventional computing platform is relatively a slow process, the proposed algorithm is synthesized on an FPGA with the aim of making a custom chip that can be embedded in a development board for abandon object detection applications. Such hardware implementation also speeds up the execution of the algorithm by exploiting the parallel nature of algorithm and makes the computation real-time. The proposed technique successfully detects abandoned objects of different sizes present in the video.**

Keywords—Abandoned object detection, Video surveillance, FPGA

I. Introduction

Abandoned objects are a common sight in any modern day crowded area such as a marketplace, railway station, school, and public transport. However, these abandoned objects can contain dangerous explosives planted by people with malevolent intentions to cause heavy loss of life and property. Such explosives are often disguised as commonplace objects, which are easily ignored by a passer-by and left unattended at a location, whilst keeping the bomber at a safe distance to trigger the explosive.

In an urban area, we are constantly monitored by Closed-Circuit Television (CCTV) cameras, to locate and identify any such object and trace its origins. However, the current surveillance system is largely man-powered, where a small group of security personnel is required to constantly gaze at scores of cameras, spread throughout the area to find anything suspicious. It is easy to imagine that this is not an easy task to continue for hours, and it is likely that the aforementioned personnel may miss an object here or there. Also, due to low manpower, an area cannot be monitored by an individual constantly, leaving abundant room for error.

Several tactics are deployed to address this problem. High-resolution overhead cameras are used for surveillance in significant locations such as national monuments or private establishments, to detect such unattended objects of even small size. However, these cameras are expensive, and even with their resolution, there is no guarantee that the security personnel will spot the object. Also, in some cases security officers often patrol the area under surveillance, to maintain vigilance over any such object. This is commonly seen in railway stations and areas of high-security zones. However, this tactic requires excessive manpower, especially if the area under surveillance is huge. Also, if there happens to be a bomb concealed in an abandoned object, this raises the risk of grievous physical harm to the officers who would examine this object.

Also, the public is often encouraged to raise an alarm in case they find any such object, once they have established that it does not belong to anyone nearby. Television advertisements and loudspeaker announcements are used to motivate people to survey the area by themselves to confirm its safety. However, most people do not follow such instructions out of ignorance, or the inhibition of raising a false alarm. Even in a genuine case, a common person can find it difficult to send the message to the appropriate authority. None of the above methods uses any sort of autonomous solution. This creates a need to develop an automated system which intelligently detects such situations and inform the concerned people.

Our approach takes input from the camera a video stream and models a reference static frame and other objects which are ordinarily present in it. It then updates the background to incorporate any static object that has been abandoned or has been removed from the original background. Finally, it compares the new background with the original background and generates their difference frame which only contains the abandoned objects enclosed inside ovals, informing the security personnel who is monitoring the area in the control room. By implementing this approach in the form of a custom hardware design on FPGA, we hope to make a chip in future that can be easily embedded on a development board for such applications.

II. Related Work

Many methods have been previously proposed to automate the detection of abandoned objects. Jing Chang [2] utilizes selective tracking to determine whether the owner of the so-detected abandoned luggage is in proximity by detecting “skin color information” and body contours. However, it may be difficult to determine in crowded situations where people are rapidly moving towards and away from said luggage. Also, detecting such attributes of a person is subject to lighting changes and occlusions due to an ambient crowd. Most of the proposed methods, such as the one proposed by M. Bhargava [1] search for an owner of the luggage in the midst of the crowd, which requires the system to have a lot of memory at its disposal. Background Subtraction [7] is also commonly utilized, but it is a practical only if the background so acquired updates itself with lighting changes over time, to prevent noise contours. Gaussian Mixture Model (GMM) is commonly utilized to achieve background updating [12] [3] [10]. Although GMM is an efficient method for background modeling but it is memory intensive. We present a memory

*Corresponding author, E-mail: *rajesh@dce.ac.in*

efficient algorithm to achieve a result that is comparable to that of GMM, also, we can explicitly control the amount of memory utilized for storage.

Fig. 1: A static frame- **I** without object and **II** with object

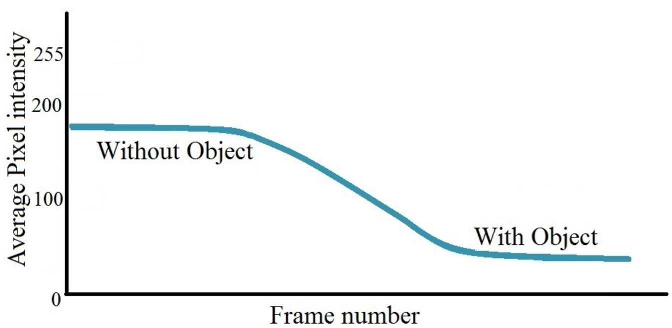


Fig. 2: Variation of pixel intensity at encircled position in Fig.1(I)-(II) with incoming frames. Averaging is performed over all the coming inputs video frames, highlighting effect of introduction of an object that is black in this case.

III. Algorithm

A. Obtaining Reference Static Frame

The proposed algorithm considers an assumption during the initialization period. It assumes that there are no abandoned objects in the first **'n'** frames of the video. We model an initial static background from the video feed which contains only those objects, which are at rest. Such objects form a general composition of the background, which can be trees, poles, signboards, buildings, roads and more. This is computed by forming an image using the average value of each pixel in the first **'n'** frames. This is done as per the designed algorithm flow which takes only a few seconds. This computed initial static background is stored in memory for further use.

B. Updating Current Static Frame

After initializing the static background containing solely the stationary objects that are expected to be present in the video feed, we constantly update by doing computations as proposed in *III-A* and models an updated static background. This is done after the desired interval to incorporate incoming of new static objects into the video. Thus, if a new object comes into the cameras field of view and it has not been moved for a certain period of time, its presence will be visible in the updated static background. This happens even if the object is blurred due to the low resolution of surveillance video. Since the effect of rapidly moving objects while updating the static background will be very less, hence, this updating is unaffected by object occlusion due to crowds. This helps the proposed algorithm to detect such objects in the presence of spatial and temporal occlusion successfully, experimental results of the same is shown in the result section.

C. Current Frame Comparison

In case an object is left unattended in the video feed for an extended period of time, its presence will be seen in the updated static frame. This updated static frame is compared with the previous reference static frame without the object, such comparison will reveal the object as a highlighted blob. This comparison is performed by calculating a difference between both frames.

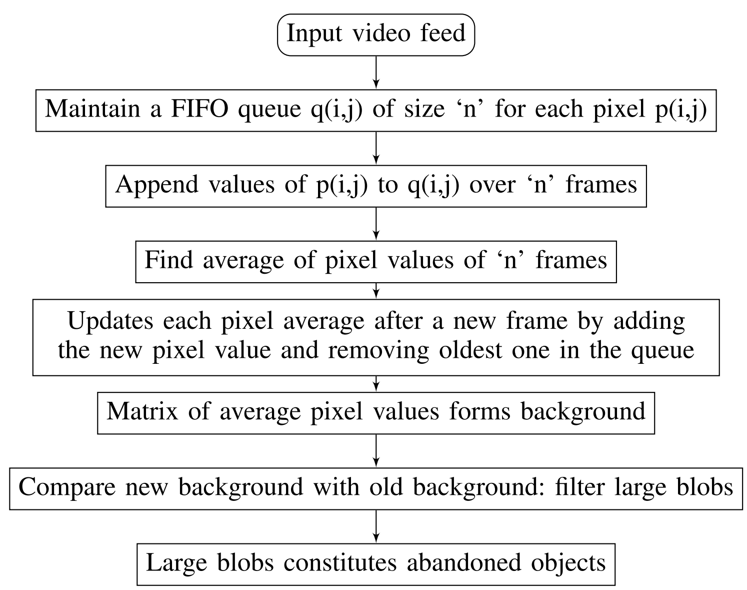
D. Blob Detection

Blob detection (also known as connected component labeling) is applied on the grayscale image produced by comparison in the previous section and attempts to find the bright areas. In order to remove the effect of intermittent movement in a video feed that results in the presence of many smaller blobs during frame comparison, we merge such blobs that are close together into one larger blob.

E. Decision making

We use a very efficient approach of comparing sizes of various blobs in the frame. If any blob is too small to be a significant object, we discard it as noise. We display the larger blobs in a separate window, making it easy for the security personnel to locate such objects and take necessary action.

IV. Flowchart



978-1-5090-0082-1/16/$31.00 ©2016 IEEE

V. Mathematical Formulation

Consider the n^{th} frame of the video as I_n, such that $I_n(i,j)$ represents the value of pixel present at i^{th} row and j^{th} column of n^{th} frame. For each such pixel value a queue *Q(i,j)* of size *N*, a sum of pixel values *S(i,j)* and average of pixel values *A(i,j)* is maintained over the incoming frames. The total number of frames *N* to be considered for modeling a static background can be selected by user as per the requirement. It is directly related to the measure of time after which an object will be declared abandoned. Starting from the first frame, *n = 1 :*

- If $n < N$

Q(i,j) = $I_1(i,j)$, $I_2(i,j)$, ... ,$I_{n-1}(i,j)$, $I_n(i,j)$

Hence sum of elements of this queue is given by:

$$S_n(i,j) = \sum_{k=1}^{n} I_k(i,j) \tag{1}$$

Average of elements of this queue is given by:

$$A_n(i,j) = 0 \tag{2}$$

As we are calculate the average value of each pixel in pixel queue only when first *N* frames has been observed by system and hence the background frame is blank here.

- If $n \geq N$

Q(i,j) = $I_{n-N+1}(i,j)$, $I_{n-N+2}(i,j)$, ... ,$I_{n-1}(i,j)$, $I_n(i,j)$

Now sum of elements of this queue is given by:

$$S_n(i,j) = S_{n-1}(i,j) + I_n(i,j) - Q(i,j).front \tag{3}$$

where $I_n(i,j)$ is pixel value in the latest frame and $Q(i,j).front$ is the oldest pixel value in the queue $Q(i,j)$

Average of elements of this queue is given by:

$$A_n(i,j) = S_n(i,j)/N \tag{4}$$

We model a background now such that:

$$\Big[B_n(i,j)\Big]_{a\times b} = \Big[A_n(i,j)\Big]_{a\times b} \tag{5}$$

where $a \times b$ is the dimension of the frame and $B_n(i,j)$ is the pixel value in the static background frame. This ensures that we are always computing the average of the latest *N* number of frames coming in the video feed. Hence for an appropriate value of *N*, the proposed algorithm automatically updates background. We simultaneously maintain a separate video frame $W_n(i,j)$ that contains the abandoned objects detected by the algorithm. If there are no such objects, this frame remains empty.

VI. FPGA Implementation

The proposed algorithm is implemented on Xilinx Zynq-7020 all programmable system on chip (SoC) FPGA board. It has Processing System (PS) that contains Dual ARM Cortex-A9 MPCore processor with CoreSight that provides high-speed sequential logic implementations and a powerful Programmable Logic (PL) that contains Artix-7 FPGA with 85,000 logic cells and other resources that give flexibility to designers to implement complex parallel operations on it with sequential processes. The graphical view of this SoC is shown in *Fig-3*. We use an RGB camera to capture live video feed which is pre-processed by the PS part before giving input to PL part where the FPGA design of our proposed algorithm for abandoned object detection lies.

The PL section is programmed using Vivado High-Level Synthesis (HLS) library provided by Xilinx. The data transfer is done using the AXI-Stream bus which is highly efficient and fast for real-time high-bandwidth data transfer. Every new frame sent to PL part is converted into a FIFO matrix having a FIFO queue for each pixel as outline in the previous section. This FIFO queue updated with each new incoming video frame. The sum and averages of pixel values in FIFO queue is calculated after a *N* number of latest frames is observed that can be set while running the algorithm. The design gives output the frame W_n in the form of FIFO buffer that contains the abandoned objects blobs detected by the algorithm. This is sent to PS part where it is converted into video format frame and is projected on a monitor.

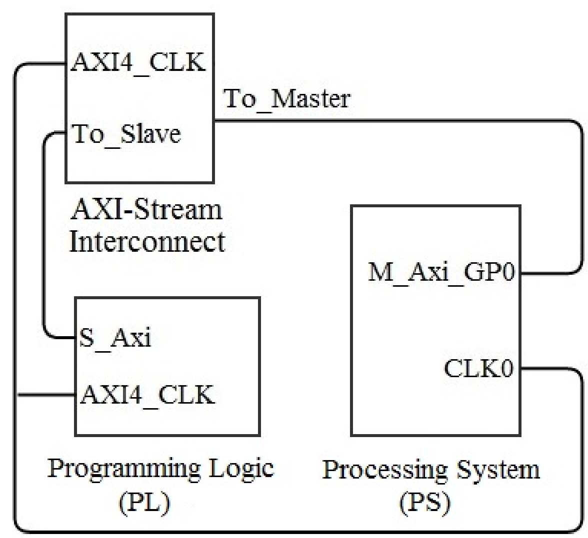


Fig. 3: Block diagram of our FPGA system

Results

We tested our proposed approach on the AVSS2007 dataset. This dataset contains videos from different scenarios, such as abandoned objects and parked vehicles. Since the abandoned object scene fit our problem, so we tested our algorithm on the sequences - AB-Easy, AB-Medium, and AB-Hard that contains abandoned luggage placed on a platform. We also compared our method with current state of the art studies of [9], [4], [6], [11], [5], [8]. Precision, Recall and F-measure values for this comparison are shown in TABLE-I. The luggage left is easily detected in the case of AB-Easy as shown in Fig. 4 due to no crowd and large size of luggage, on the other hand for AB-Medium and AB-Hard this is challenging due to the small size of the abandoned object and crowd on the platform.

 978-1-5090-0082-1/16/$31.00 ©2016 IEEE

Noteworthily, our method localizes the abandoned objects in all three sequences.

Fig. 4: Detection results of the sequence AB-Easy of AVSS2007

Fig. 5: Detection results of the sequence AB-Medium of AVSS2007

	[9]	[4]	[6]	[11]	[5]	[8]	Ours
Precision	0.05	0.21	0.40	0.35	0.97	1.0	1.0
Recall	1.0	1.0	0.67	1.0	1.0	1.0	1.0
F-measure	0.09	0.35	0.50	0.52	0.98	1.0	1.0

TABLE I: Comparison of different methods on AVSS2007 video dataset

We also tested our algorithm on our own dataset for two types of camera positions having different environment situations:
(i) A horizontally placed camera on the table top in minimally crowded place, for example, our lab.
(ii) An overhead surveillance camera in very crowded place.

Since occlusion of the object of interest in object detection algorithms is a major concern, so we created various temporal and spatial occlusion situations while testing the performance of our algorithm. Also, the situations when there are objects of various dimensions and it's variable distance from the camera were created. The results as shown in TABLE-II reflects the state of the art performance on such tasks.

References

[1] M. Bhargava, C.-C. Chen, M. S. Ryoo, and J. K. Aggarwal. Detection of object abandonment using temporal logic. *Machine Vision and Applications*, 20(5):271–281, 2009.

[2] J.-Y. Chang, H.-H. Liao, and L.-G. Chen. Localized detection of abandoned luggage. *EURASIP Journal on Advances in Signal Processing*, 2010(1):1–9, 2010.

[3] Y. Dedeoğlu. Human action recognition using gaussian mixture model based background segmentation.

[4] R. H. Evangelio, T. Senst, and T. Sikora. Detection of static objects for the task of video surveillance. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 534–540. IEEE, 2011.

[5] Q. Fan, P. Gabbur, and S. Pankanti. Relative attributes for large-scale abandoned object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2743, 2013.

[6] Q. Fan and S. Pankanti. Modeling of temporarily static objects for robust abandoned object detection in urban surveillance. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 36–41. IEEE, 2011.

[7] H. Lee, S. Hong, and E. Kim. Probabilistic background subtraction in a video-based recognition system. *KSII Transactions on Internet & Information Systems*, 5(4), 2011.

[8] H.-H. Liao, J.-Y. Chang, and L.-G. Chen. A localized approach to abandoned luggage detection with foreground-mask sampling. In *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*, pages 132–139. IEEE, 2008.

[9] F. Porikli, Y. Ivanov, and T. Haga. Robust abandoned object detection using dual foregrounds. *EURASIP Journal on Advances in Signal Processing*, 2008(1):1–11, 2007.

[10] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[11] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M.-T. Sun. Robust detection of abandoned and removed objects in complex surveillance videos. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 41(5):565–576, 2011.

[12] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.

978-1-5090-0082-1/16/$31.00 ©2016 IEEE

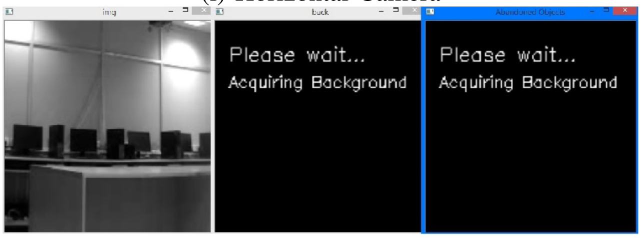
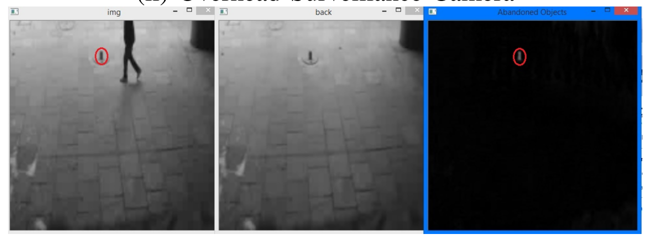
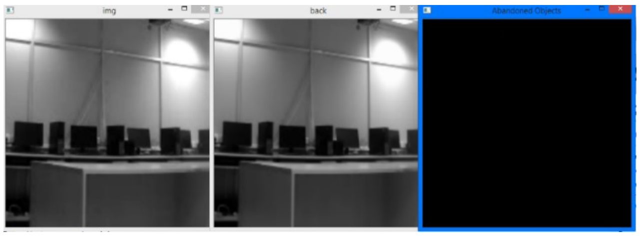
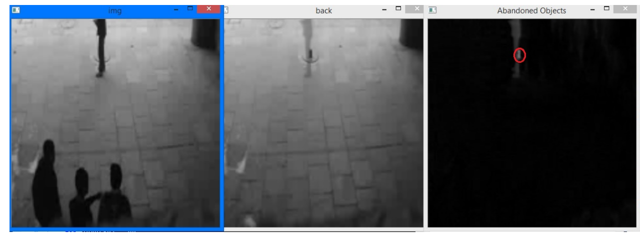
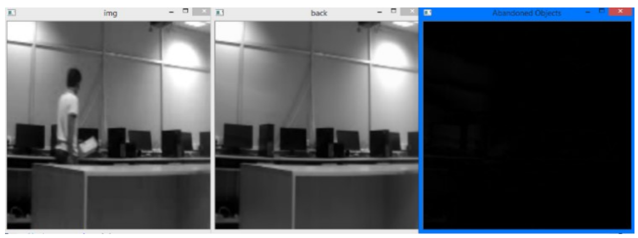
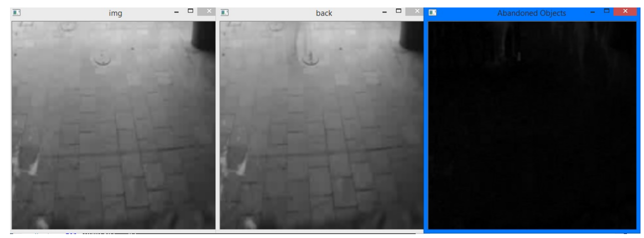
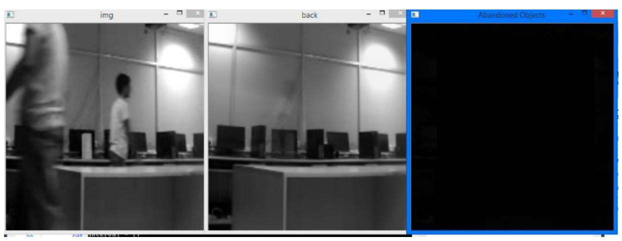
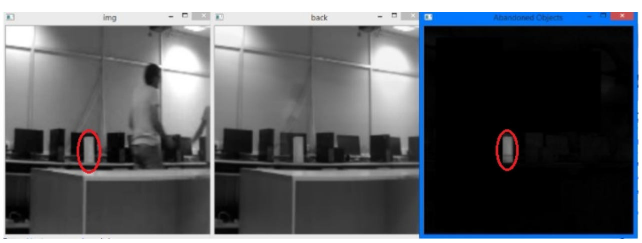
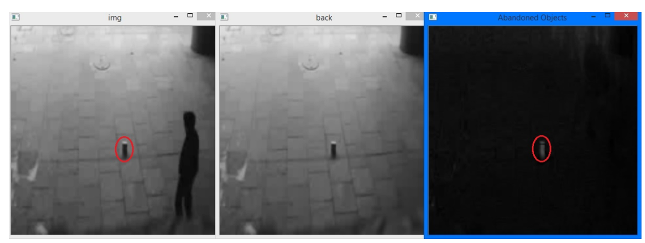
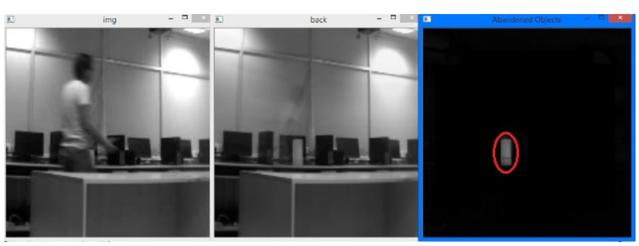
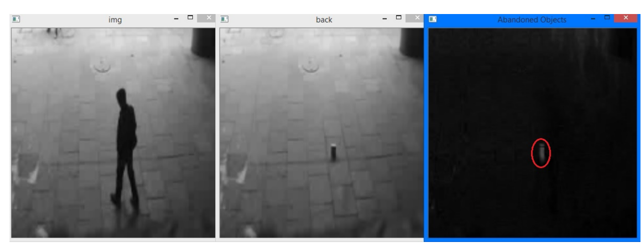
	(i) Horizontal Camera	(ii) Overhead Surveillance Camera
1	 Static frame acquisition stage	 Far-away object detected
2	 Static frame acquisition complete	 Temporal occlusion test on far-away object
3	 Spatial Occlusion test	 Image disappears once object is removed
4	 Object Placed	 Spatial occlusion test on far-away object
5	 Abandoned Object detected	 Nearby object detected
6	 Temporal Occlusion Test	 Temporal occlusion test on nearby object

TABLE II: Results of proposed algorithm on video feed in different conditions

978-1-5090-0082-1/16/$31.00 ©2016 IEEE