International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 5, May - 2013

Abandoned or Removed Objects Detection from Surveillance Video using Codebook

Sajith K
M.Tech Student (CSE)
Viswajyothi College of Engineering and Technology
Vazhakulam, Ernakulam (dist), Kerala, India

Dr. K. N. Ramachandran Nair
Head of the Department (CSE)
Viswajyothi College of Engineering and Technology
Vazhakulam, Ernakulam (dist), Kerala, India

***Abstract*—Detection of abandoned or removed objects from complex surveillance videos isdifficult due to many factors like occlusion, quick lighting changes etc. The proposed system efficiently detects abandoned and removed objects in surveillance video from a stationary camera based on foreground analysis. As the most important phase, the background subtraction is implemented using CODEBOOK method with several improvements like dynamic updation of the background model, reduced computation time and efficient memory utilization. The same codebook is used for static object detection by generating two background models at different frame rates. Moreover the static object detection is combined with tracking to reduce negative results.Classification of abandoned and removed objects are achieved by a matching method based on edge detection and user defined parameters such as size and position. The system is further improved with a human detection method based on histogram of oriented gradients.**

Keywords—abandoned object; removed object; codebook; background subtraction (BGS); video surveillance; antiterrorism

I. INTRODUCTION

Video Surveillance allows us to remotely monitor a live or recorded video feed which often includes people. As a tool for crime control and investigations video surveillance now receives increased attention and hence there has been a significant increase in the use of video surveillance cameras in public locations such as stores, ATMs, schools, buses, subway stations, and airports in order to combat crime. Now automatic systems to detect abnormal activities from surveillance videos are gaining importance as a means to help the security personals. One among them is the abandoned objects detection system.

Detection of abandoned objects is very important to prevent attacks on landmarks, public transportation, and critical assets. It is very challenging for security officers as well as video surveillance solutions to quickly detect objects that have been left behind at places with high traffic flows like train/subway stations, airports, big cities, and other public places. The existing tracking based approaches for abandoned object detection are unreliable in complex surveillance videos due to problems like occlusions, lighting changes, and other factors. An abandoned object can be defined 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. Hence the objective is to detect static regions that have recently changed in the scene and determine whether they correspond to abandoned or removed objects using background subtraction and foreground analysis based on user defined parameters.

II. BACKGROUND GENERATION

A. *Codebook For Background Model Generation*

The codebook method proposed by Kyungnam Kim, Thanarat.H.Chalidabhongse, David Harwood and Larry Davis [1] presents a background subtraction algorithm where backgrounds are modeled using multiple codewords. The key features of the algorithm

- A compact background model to capture structural background motion over a long period of time under limited memory.
- Can model moving or repeatedly changing backgrounds.
- Can cope with local and global illumination changes.
- Unconstrained training that allows moving foreground objects in the scene during the initial training period.
- Layered modeling generates multiple layers of background.

B. *Construction of Codebook*

In order to generate a background model that captures moving or repeatedly changing environment, each pixel in the video frame is modelled using a codebook. A codebook is a collection of codewords**{ c_1, c_2, …..c_n}**corresponding to each pixel and the number of codewords for each pixel may be different. Each codeword**c_i**is a set of 9 values **{ R_i, G_i, B_i, Imin, Imax, f, MNRL, p, q }**.

R_i, G_i, B_i: the RGB values corresponding to the pixel.

Imin, Imax: the min and max brightness, for the pixels assigned to this codeword

f: the frequency with which the codeword has occurred.

MNRL : the maximum negative run-length (MNRL) defined as the longest interval during which thecodeword has not recurred.

p, q : the first and last access times, respectively, that thecodeword has occurred.

During background generation value of the pixels in the current frame is compared to the codebook to determine which codeword c_m it matches (if any). In order to find a matching codeword a color distortion measure and brightness bounds is used.

The color distortion for a pixel xwith RGB values {R, G, B} w.r.t a codeword c_i with RGB values {R_i, G_i, B_i} can be calculated as below :

$$colordist(x, c_i) = \sqrt{(R^2 + G^2 + B^2) - \left(\frac{(RR_i + GG_i + BB_i)^2}{{R_i}^2 + {G_i}^2 + {B_i}^2}\right)}$$

The brightness of a pixel x with RGB values {R, G, B} can be calculated as

$I = \sqrt{R^2 + G^2 + B^2}$achcodeword stores Imin and Imax, the minimum and maximum brightness of all pixels assigned to that codeword respectively. The brightness is allowed to vary a certain range [Ilow, Ihigh] defined by two thresholds α and β as given below:

$$I_{low} = \alpha I_{max}$$

$$I_{high} = min\left\{\beta I_{max}, \frac{I_{min}}{\alpha}\right\}$$

where α<1 and β>1. Typically, α lies between 0.4 (for large bounds) and 0.7 (for tight bounds), and β limits Ihigh, inorder to handle shadow regions, its value lies between 1.1 and 1.5. The brightness function is given by:

$$brightness(I, \{I_{min}, I_{max}\}) = \begin{cases} \boldsymbol{true} & if\ I_{low} \le I \le I_{high} \\ \boldsymbol{false} & otherwise \end{cases}$$

C. Construction of Background Model

The background model is computed through a per-pixel on-line statistical analysis of the video frames in order.Let F(t) be the W x H input frame at any time t, C be the W x H codebook where C(x, y) is the codebook corresponding to pixel at position (x, y) and L(x, y) gives the number of codewords in each codebook C(x, y). FDM and FDC are the W x H foreground detection mask and the foreground detection count (number of codewords to be considered for FDM generation) respectively.

(i) Initialize the code book.

L=0,

C=Φ (null set),

FDM=0 (black frame),

FDC=1(first codeword)

(ii) For each frame F(t) do

(iii) For each pixel F(x, y) do

(a) Assign $X = (R, G, B)$ and $I = \sqrt{R^2 + G^2 + B^2}$

(b) Find a matching codeword c_i from C(x, y) based on two conditions (1) and (2):

1. $colordist(X, c_i) \le \varepsilon$ (ε is a threshold value)
2. $brightness(I, \{I_{min}, I_{max}\}) = true$

(c) If C=Φ or no matching codeword is found

1. Mark the pixel as forground, by setting it to white, ie FDM(x, y) = 1
2. Increment L(x, y) by 1.
3. Create a new codeword
$$c_L = \{\mathrm{R}, \mathrm{G}, \mathrm{B}, \mathrm{I}, \mathrm{I}, 1, t-1, t, t\}$$
4. Add it to codebook C(x, y).

(d) Else update the matching codeword

$c_i = \{R_i, G_i, B_i, I_{min}, I_{max}, f, MNRL, p, q\}$as

1. $c_i = \left\{\frac{fR_i+R}{f+1}, \frac{fG_i+G}{f+1}, \frac{fB_i+B}{f+1}, \min(I, I_{min}), \max(I, I_{max}), f+1, \max(MNRL, t-q), p, t\right\}$
2. If $i > FDC(x, y)$
   - Filter the codebook C(x,y).
   - Set FDC(x, y) = number of codewords in thefiltered codebook

(iv) End For

(v) End For

D. Filtering the Codebook

Filtering is the process of removing the unwanted codewords from each codebook, thereby reducing the size of the codebook. The codebook needs to store only those codewords that represent the static background pixels or repeatedly occurring background pixels, removing the moving foreground pixels. So the codewords that doesn't occur continuously for a specific number of frames (μ) are removed from the codebook based on the MNRL value.

The codebook is filtered at regular intervals ie after every N^{th} frames. The filtered codebook contains those

codewordswhose MNRL value is less than a threshold μ. That is the filtered codebook $M = \{c_i | c_i \in C \wedge MNRL \leq \square\}$ .The codewords with large MNRL values ie greater than μ will be eliminated. Usually μ = N/2, where N is the number of frames after which the codebook is filtered and the codewords should recur at least every N/2 frames.

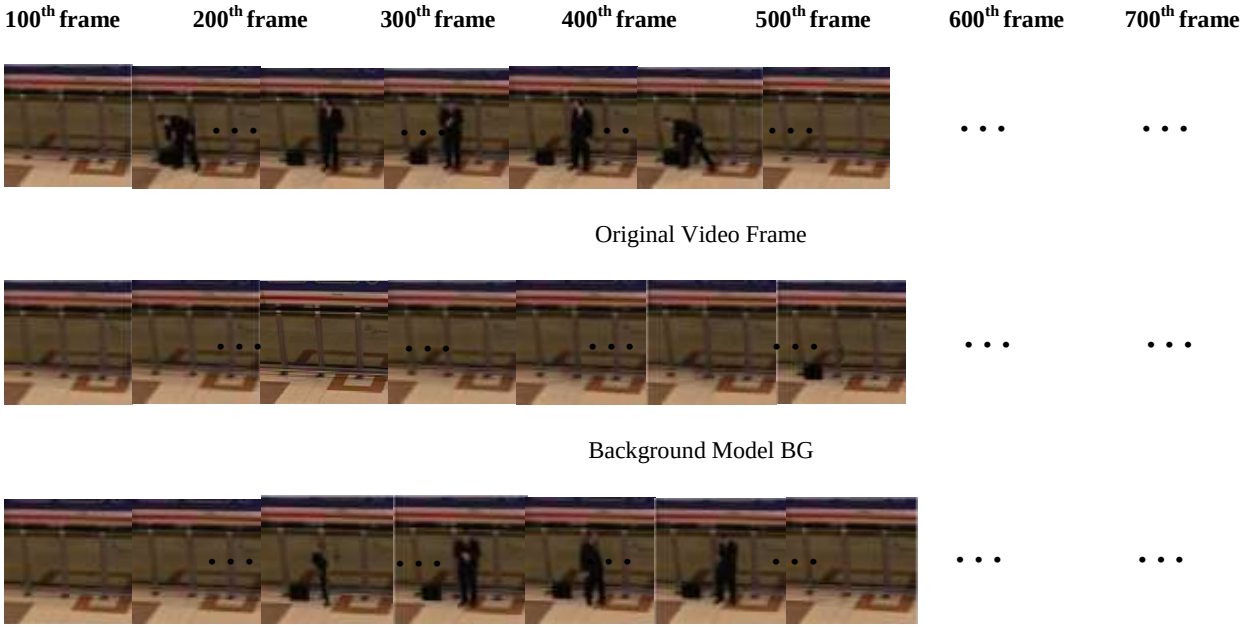

Fig1 Generated Background Models

E. Detecting the Static Foreground

The static foreground objects may possibly represent an abandoned or removed object. Hence two background models together with foreground region tracking is employed to detect static foreground objects. Once a codebook is constructed, there will be at least one codeword corresponding to each pixel. After the initial filtering of the codebook, the first codeword represents the static background pixels. Hence the background model is constructed using the RGB values of the first codeword of each pixel, ie the background $B(x,y) = \{(R,G,B)|RGB \in c_1, c_1 \in C(x,y)\}$.

For detecting the static foreground pixels two codebooks C_1 and C_2 are employed. The difference between the two codebooks is only in the filtering rate and value of filtering threshold (μ). The first codebook C1 is used for the generation of the FDM (foreground detection mask), hence the filtering rate N must be sufficiently high and the filtering threshold $\square = \frac{N \times n}{2}$ for n^{th} filtering. The second codebook C2 is used solely for detecting static foreground pixels, hence the filtering rate N is equal to the minimum number of frames for which an object must remain static before it can be classified as a static foreground object and the filtering threshold is $\square = \frac{N}{2}$ for every filtering. That is the filtering threshold μ for C1 will be changing with time while it remains constant for C2.

Let BG be the background model generated from codebook C1 and SBG be the background model generated from codebook C2 as given below:

$$BG(x,y) = \{(R,G,B)|RGB \in c_1, c_1 \in C1(x,y)\}$$

$$SBG(x,y) = \{(R,G,B)|RGB \in c_1, c_1 \in C2(x,y)\}$$

The figure 1 represents the two background models:

1. BG generated from C1 with the filtering rate 100 and $\square = \frac{100 \times n}{2}$ for n^{th} filtering.
2. SBG generated from C2 with the filtering rate 40 and $\square = \frac{40}{2}$ for every filtering.

III. STATIC OBJECT DETECTION

The static object detection consist of the following process:

1. Generate the Static Object Detection Mask (SDM) and Foreground Detection Mask (FDM).

2. Tracking the blobs in the FDM.

3. Combining the SDM with tracking results.

A. Generating the FDM and SDM

The Foreground Detection Mask (FDM) is generated by the background model generation method described in section above using codebook C1, while the Static Object Detection Mask (SDM) generation consists of two process:

1. Generating the Background detection mask. (BDM)

2. Multiplying BDM with FDM

The BDM is generated from the background models SBG and BG, using the color distortion value as given below:

$$BDM(x,y) = \begin{cases} 1 & if\ colordist\big(BG(x,y), SBG(x,y)\big) \le \varepsilon \\ 0 & otherwise \end{cases}$$

After which each pixel in BDM is multiplied with corresponding pixels in FDM to generate SDM as given below:

$$SDM(x,y) = \begin{cases} 1 & if\ BDM(x,y) \wedge FDM(x,y) = 1 \\ 0 & otherwise \end{cases}$$

The figure 2 represents the FDM, BDM and SDM generated for every 100th frame.

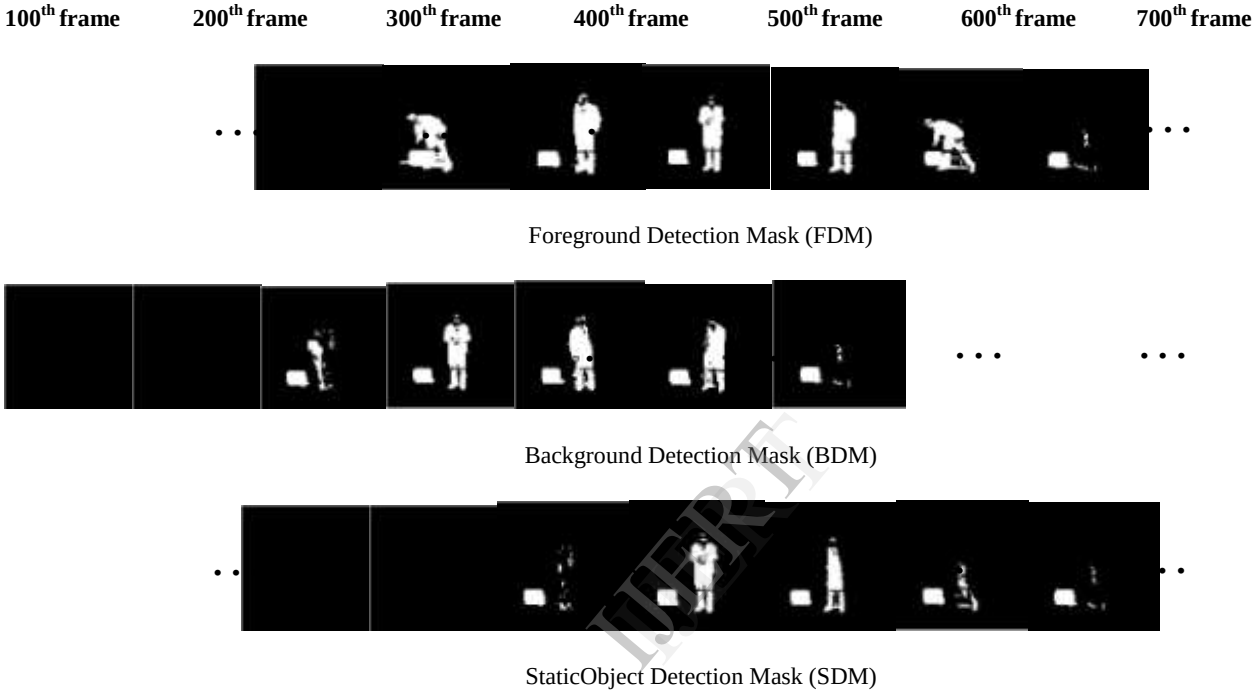

Fig 2. Generated Foreground Masks

B. Tracking the Blobs

The Foreground Detection mask consists of black and white pixels, where the white ones denote the foreground pixels as well as noise. The collection of neighboring white pixels or the blobs in the FDM represents foreground objects which must be tracked for detecting the static objects. Whenever an object (blob) is detected in a frame a new track is generated for it and each track consist of two values hit count and miss count. In the next frame if the blob is in the same position and the size (area) of the blob is same, the existing track for that object will be updated by incrementing the hit count by 1 and setting miss count to 0, else the miss count will be incremented by 1.

The existing tracks will be deleted when the miss count of the corresponding track is greater than a threshold λ_1. λ_1 can be set as the number of adjacent frames for which the object can remain occluded before dropping the track. So for a moving object a new track will be generated for the object in each frame and will be dropped after λ_1 frames while for a static object there will be only a single track with hit count equal to the number of frames for which the object remained static. Therefore a static object can be classified as an abandoned or removed object if the hit count is greater than a threshold λ_2.λ_2 can be set as the number frames for which an object must remain static before classifying it as an abandoned or removed object.

C. Combining SDM with Tracking Results

The blobs in SDM represent the static objects in the video frame. As the static objects are slowly updated to the background model the detected blobs may be only a part of the static object at the beginning and will increment with time. Hence the static objects detected in SDM is classified as abandoned or removed object based on conditions (1) and (2).

(1) whetherarea of the detected object in SDM is greater than half the size of the original object:

$a \geq A/2$, where a is the area of object in SDM and A is thearea of the original object.

(2) whether hit count of the track corresponding to the detectedobject in SDM is greater than a threshold $hit\ count \geq \lambda_2$, where λ_2 is the number frames for which an object must remain static before classifying it as an abandoned or removed object.

If the above two conditions are satisfied the static object is classified as abandoned or removed object. From figure 3 we can see that the proposed method can easily detect removed and abandoned objects from the scene.

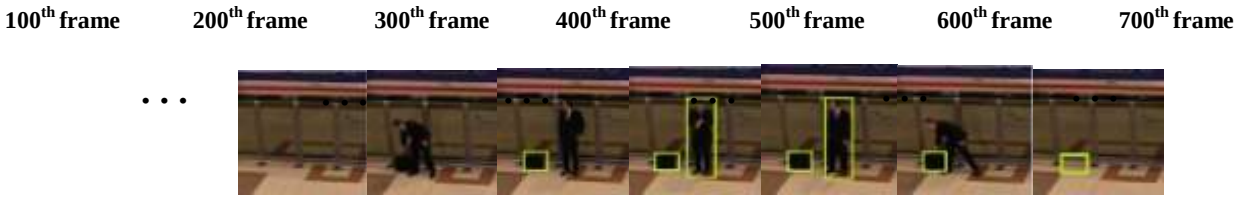

Fig 3. Static Object Detection

IV. STATIC OBJECT CLASSIFICATION

A. Static region type detection

For static type detection we first detect the edges in the original video frame using one of the standard edge detection algorithms. Then we start a fill in function from the interior of the detected object to the boundaries. The fill in function stops at the boundaries of the object, leading to a segmented region. The same process is then applied in the background image BG, leading to the generation of another segmented region. This method is inspired by the work done by YingLiTian and others for heal type detection [2].

We can classify a detected static object into an abandoned object or removed object by comparing the area of segmented region. Let 'A' be the area of segmented region generated for the background BG and 'a' be the area of segmented region generated for the current frame. Then the static object:

- Is classified as an abandoned objectif A>a
- Is classified as a removed object if A<a
- Is ignored if A = a as it can be caused by lighting changes not objects

From figure 4 we can see that the segmented region is larger for the generated background than the frame with the actual abandoned object.

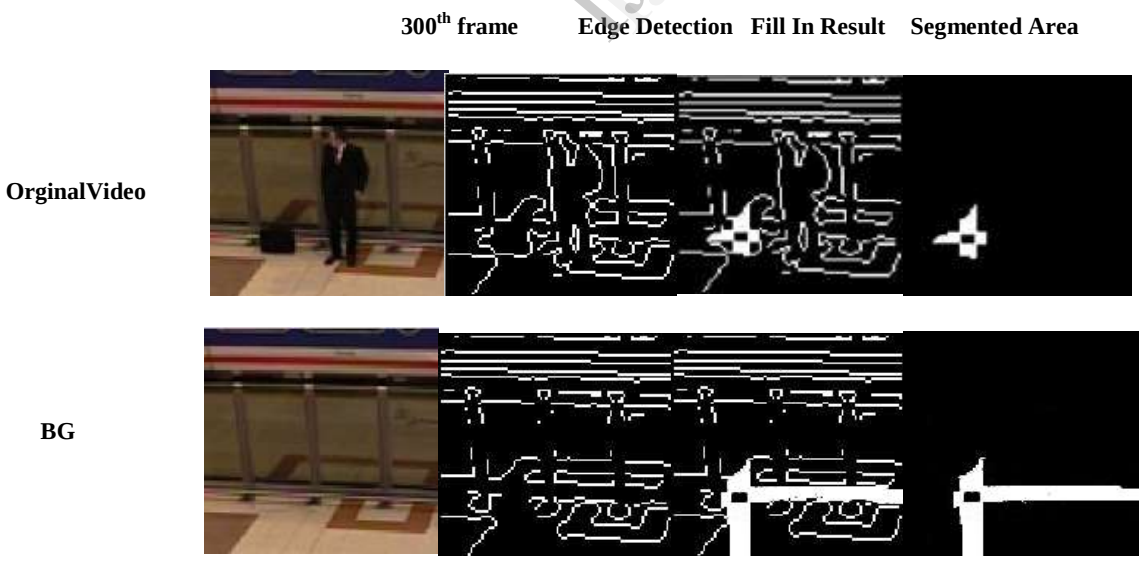

Fig 4. Static Type Detection

B. Human Detection

The main problem we confront in the above detection method is that the system detects every static object whether human or nonhuman. Hence a Human detection method based on (Histogram of Oriented Gradients) HOG descriptor is employed for Feature extraction [3]. HOG parameters are computed for a training data set consisting of human and nonhuman images and are used to train a Neural Network classifier for categorizing objects and humans.

C. User Interface

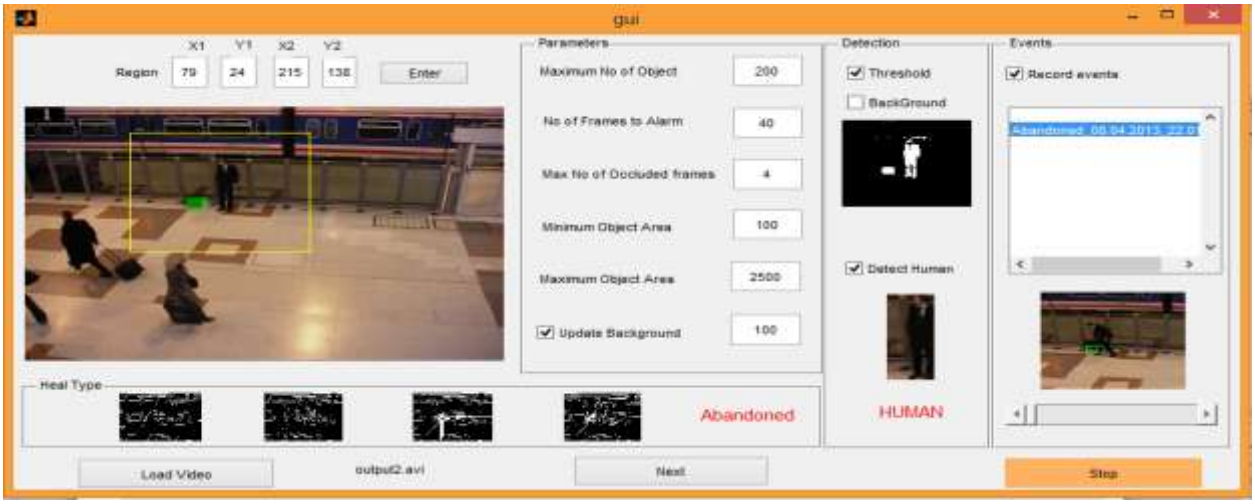

Fig 5.User Interface

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 Fig., which include the following inputs:

1) Sizes: Minimum and maximum object size.

2) Regions of interest: Rectagular regions manually drawn by the user in the image (objects are detected only on those regions).

3) Abandoned/removed time: It indicates how long a foreground region corresponding to an abandoned/removed object should stay stationary in the scene in order to trigger an alert.

4) Maximum Occlusion time: It indicates how long a foreground region corresponding to an abandoned/removed object can remain completely occluded before removing the alert.

5) Human detection: If both human and nonhuman object classes are selected for abandoned and removed object detection, the human detection process will be skipped.

6) Event Recording: Record the events of dropping or removing an object from the scene.

V. TEST RESULTS

The proposed system is tested using pets 2006 and 2007 dataset [9] [10] to identify the effectiveness of the system for abandoned/removed object detection in a variety of environments. The system provided good results with the training set.

Table 1: PETS 2006 Test Results

Dataset	Sequence	Camera view	Static objects	Static persons	False Result
Pets 2006	S3	View 1	1	1	1
		View 2	1	1	1
		View 3	1	1	0
		View 4	1	1	0

A. PETS 2006 and 2007 Dataset

PETS 2006 and 2007 datasets were designed to test abandoned object detection algorithms in a public space. The PETS dataset consists of sequences containing left-luggage scenarios with increasing scene complexity. There

are different scenarios captured by four cameras from different viewpoints. The scenarios are relatively simple, without many occlusions and crowds. The proposed algorithm detected all abandoned items, with zero false alarms. A static person is detected as an abandoned item in sequence S3 of PETS 2006 dataset.

Table 2: PETS 2007 Test Results

Dataset	Sequence	Camera view	Static objects	Static persons	False Result
Pets 2007	S8	View 1	1	0	0
		View 2	1	0	0
		View 3	1	0	0
		View 4	1	0	0

B. Limitations

The accuracy of detection system is influenced by many factors.

1. The size of the abandoned object is too small or the abandoned object is occluded.

2. Low-light conditions reduce the ability to discern one object from another, causing higher error rates.

3. Static-object detection in crowded scenes is difficult, leading to higher error rates.

4. Quick-lighting changes cause problems to detect abandoned or removed objects.

5. Low contrast situationslead to missed detections.

VI. Conclusion

The proposed system robustly and efficiently detects abandoned and removed objects in real-time video surveillance. Two background models are employed to detect both background and static foregrounds by using the same Codebook method. Then, the static foregrounds were classified into abandoned or removed objects by segmenting and comparing the background model with the foreground image. This method can handle occlusions in complex environments with crowds. The testing results shows the system can be successfully applied in real-world surveillance applications.

References

[1] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, Larry Davis,"Real-time foreground–background segmentation using codebook model", Real-Time Imaging, ELSEIVER, 2005

[2] YingLi Tian, Rogerio Schmidt Feris, Haowei Liu, Arun Hampapur, and Ming-Ting Sun, "Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos", IEEE Transactions on systems, man, and cybernetics part c: applications and reviews, vol. 41, no. 5, september 2011.

[3] Oswaldo Ludwig Junior, David Delgado, Valter Gonc¸alves, Urbano Nunes, "Trainable Classifier-Fusion Schemes: an Applicationto Pedestrian Detection,",Internatoinal IEEE Conference on Intelligent transportation systems, October 2009.

[4] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," inProc. CVPR99, Jun, pp. II-2246–2252.

[5] Domenico Bloisi and Luca Iocchi," Independent Multimodal Background Subtraction", Department of Computer, Control, and Management Engineering - Sapienza University of Rome, Italy.

[6] Mrs.Megha V.Gupta, Dr.S.D.Sawarkar "Change Detection based Real Time Video Object Segmentation", International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 7, September – 2012

[7] J. Wang and W. Ooi, "Detecting static objects in busy scenes," Dept. Computer. Science, Cornell University, Tech. Rep. TR99–1730, Feb. 1999

[8] F. Porikli, "Detection of temporarily static regions by processing video at different frame rates," inProc. IEEE Int. Conf. Adv. Video Signal-BasedSurveillance, London, U.K., Sep. 2007.

[9] www.cvg.rdg.ac.uk/PETS2006/data.html

[10] www.cvg.rdg.ac.uk/PETS2007/data.html