IMPROVEMENT IN ABANDONED OBJECT DETECTION USING BACKTRACKING ALGORITHM

R.Keerthana[1], A.Lakshmi priya[2], K.K.Senthil kumar[3]

[1]B.E, ECE, Prince Shri Venkateshwara Padmavathy Engineering College, TN, India
[2]B.E, ECE, Prince Shri Venkateshwara Padmavathy Engineering College, TN, India
[3]M.E, Ph.D., ECE, Prince Shri Venkateshwara Padmavathy Engineering College, TN, India

ABSTRACT

The ultimate aim of different smart visual surveillance applications is to extract semantics from video to be used in higher level activity analysis tasks. Categorizing the type of a detected video object is a crucial step in achieving this goal. With the help of object type information, more specific and accurate methods can be developed to recognize higher level actions of video objects. The information obtained during classification along with tracking results is then used for abnormal event detection like anomalous behavior in parking lots, object removal, object abandon, fire detection, etc. Here, the system present a video object classification method based on object shape similarity as part of visual surveillance system. And, instead of keeping the abnormal event detection as an abstract goal an abnormal event is predefined and detect that event in the video. In this case, the abnormal event is 'abandoning' of a luggage/bag by a person. In Mainly a video-surveillance system for the detection of abandoned objects and its owner is proposed. The approach is based on dual background for the detection of static foreground and motion analysis for its owner. The static foreground detection performs in two steps: constructing and updating a pixel-based dual background to obtain the dual foreground, carrying a register for the evidence frame to give an alarm at the same time message sending to the regarding person.

Keyword: *abandoned, static foreground, dual background*

1. INTRODUCTION

The detection of abandoned objects is more or less the detection of idle/inactive (stationary or non-moving) objects that remain stationary over a certain period of time. The period of time may be adjustable. In several types of images or frames idle objects should be detected. For example in complex near Elevator bag is left by some person. An unknown object is any object that is not a person or a vehicle. In general, unknown objects cannot move they are considered as stationary.

Security of public places is a continuously burning issue. Though the CCTV (Closed Circuit Television) camera/Webcam has installed at the places but the footage is only used after incident happen to that place. That CCTV cameras/Webcam can be used to prevent such incidents from happening. Surveillance cameras are cheap and ubiquitous. The advent of smart surveillance cameras with higher processing capabilities has now made it possible to design video surveillance systems which can contribute to the safety of people in the home and in public places such as shopping malls, airports, railways stations and etc.

The system will process the live footage of the CCTV camera with the image processing. If a person is left some bag or any such suspicious thing and leaving it, the camera captures this activity. And if such bag is untouched for some time period decided by analyzer then it gives notification to authorize person. An automated video surveillance system for robustly and effectively detecting of abandoned objects is increasing the worldwide attention in many contexts, especially, in the consumer world of applications. In these systems, it should be a sufficiently high accuracy enabling a real-time performance. Thus, a prime goal of automated visual surveillance is to obtain a live description of what is happening in a monitored area and take appropriate action.

Many surveillance tools have been employed in the fight against terror. Although video surveillance systems have been in operation for the past two decades, the analysis of the CCTV footage has seldom ventured out of the hands of human operators. Recent studies have brought into fore the limits to human effectiveness in analyzing and processing crowded scenes, particularly in video surveillance systems consisting of multiple cameras.

The advent of smart cameras with higher processing capabilities has now made it possible to design systems which can possibly detect suspicious behaviors (in general) and abandoned objects (in particular). A number of algorithms have been suggested in the recent past to deal with the problem of abandoned-object-detection. Due to their dependence on complex probabilistic mathematics, most of these algorithms have failed to perform satisfactorily in real time scenarios. In addition, the other difficulty of detecting an abandoned object under occlusion adds to the overall complexity. Some proposed algorithms have dealt with partial occlusion (by moving people) but complete or prolonged occlusion (by another object) has not yet been tackled. Furthermore, the background subtraction methods employed in the above methods are either computationally intensive or lack dynamically updating features.

An automatic abandoned object detection system typically uses a combination of background subtraction and object tracking to look for certain predefined patterns of activity that occur when an object is left behind by its owner. Though humans are much better at this task than even state of the art systems, it is often practically unfeasible to employ enough manpower to continuously monitor each one of the very large number of cameras that are required in a large scale surveillance scenario. An automatic detection system therefore helps to complement the available manpower by serving as both a standalone monitoring system for less critical areas and also integrated into manually monitored cameras so that it can detect any drops that the human may have missed.

2. RELATED WORK

The algorithms for identifying a static foreground or abandoned object can be classified into three categories. The first category involves constructing double-background models for detecting a static foreground [1], [2], [3]. The double background models are constructed using fast and slow learning rates. Subsequently, the static foreground is localized by differentiating between the two obtained foregrounds. A weakness of these methods is the high false alarm rate, which is typically caused by imperfect background subtraction resulting from a ghost effect, stationary people, and crowded scenes. In addition, these methods involve using only the foreground information per single image to locate regions of interest (ROIs) of abandoned-object candidates. Consequently, temporally-consistent information that may be useful for identifying sequential patterns of ROIs may be overlooked. The second category of methods for extracting static foreground regions involves using a specialized mixture of Gaussian (MOG) background model. In previous researches [4], [5], [6], three Gaussian mixtures were used to classify foreground objects as moving foreground, abandoned objects, and removed objects by performing background subtraction. In addition, the approach proposed in [6] uses visual attributes and a ranking function to characterize various types of alarm events. The third category involves accumulating a period of binary foreground images or tracking foreground regions to identify a static foreground. The methods proposed in [7], and [8] involved localizing the static foreground based on the pixels with the maximal accumulated values, which were subsequently considered the candidate regions of stationary objects. However, this category of methods fails in complex scenes. LV *et al.* [9] used a blob tracker to track foreground objects based on their size, aspect ratio, and location. Left luggage is identified when a moving foreground blob stops moving for a long period. Li *et al.* [10] tracked moving objects by incorporating principle color representation (PCR) into a template-matching scheme, and also by estimating the status (e.g., occluded or removed) of a stationary object.

To fulfill the semantic requirement of abandoned luggage events where a person drops their luggage and then leaves, some of the aforementioned methods combine a tracker to track the involved person(s) for further verification. Liao *et al.* [7] tracked luggage owners based on skin color information and by performing contour matching with a Hough transform. In [1], Kalman filter (KF) and unscented KF (UKF) were used to track foreground objects (including people and carried luggage) based on low-level features, such as color, contour, and trajectory. Tian *et al.* [4] integrated a human detector and blob tracker to track the owner of abandoned luggage, and the corresponding trajectory was recorded for further analysis. Fan *et al.* [6] used a blob tracker to track moving people close to the left-luggage. The obtained movement information was used as an input for their attribute-based alert ranking function

3. PROPOSED SYSTEM

The proposed system presents an effective approach for detecting abandoned luggage in surveillance videos. The system combine short- and long-term background models to extract foreground objects, where each pixel in input image is classified as a 2-bit code. Subsequently, it introduce a framework to identify static foreground regions based on the temporal transition of code patterns, and to determine whether the candidate regions contain abandoned objects by analyzing the back-tracked trajectories of luggage owners.

Also the system uses propose a temporal dual-rate foreground integration method for static foreground estimation for single camera video images. The approach involves constructing both short- and long-term background models learned from an input surveillance video on-line. It also introduce a simple pixel-based finite-state machine (PFSM) model that uses temporal transition information to identify the static foreground based on the sequence pattern of each object pixel. Because the proposed approach involves using temporal transition information, this can reduce the influence of imperfect foreground extractions in the double background models, thereby improving the accuracy of the constructed static foreground inference. Contributions of the proposed method over previous methods are summarized as follows.

- Here a dual-rate background modelling framework with temporal consistency is introduced. It performs considerably better than the single-image-based double background models.
- A simple spatial-temporal tracking method is developed for back-tracking verification. Compared to the frame by- frame tracking approaches such as the KF- or UKF, this approach is superior in handling temporary occlusions and is still highly efficient to implement.

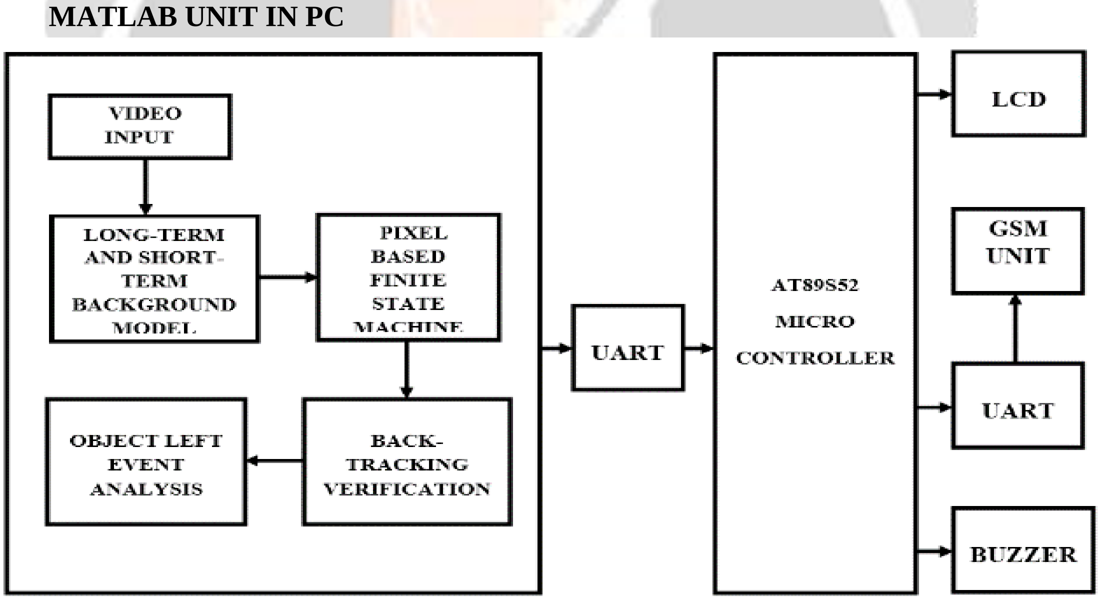
MATLAB UNIT IN PC
VIDEO INPUT
LONG-TERM AND SHORT-TERM BACKGROUND MODEL
PIXEL BASED FINITE STATE MACHINE
OBJECT LEFT EVENT ANALYSIS
BACK-TRACKING VERIFICATION
UART
AT89S52 MICRO CONTROLLER
LCD
GSM UNIT
UART
BUZZER

Figure 3.1 Block diagram for abandoned object detection

The abandoned object detection consist of two section namely software and hardware which is shown in the figure 3.1. The video input is given in the form of frames and a reference image i.e. background image is stored. The reference frame is compared with current frame using background subtraction. The object is checked in long term and short term background model to verify whether it is stable or not using pixel based finite state machine. If stable pixel (threshold value) occurs again and again then that object is identified as abandoned in object event analysis. Then that output is given to microcontroller to indicate abandoned object detection.

In hardware AT89S52 Microcontroller is interfaced to LCD (Liquid Crystal Display) and interfaced to GSM (Global System for Mobile communication) modem through UART (Universal Asynchronous Receiver Transmitter). The output is displayed in LCD and a sound is produced in buzzer if object is detected.

3.1 Algorithm Working Steps

The various steps involved in abandoned object detection are as follow:
1) Eliminate video areas that are unlikely to contain abandoned objects by extracting a region of interest (ROI).
2) Perform video segmentation using background subtraction.
3) Calculate object statistics using the Blob Analysis block.
4) Track objects based on their area and centroid statistics.
5) Visualize the results.

The abandoned object detection system explains step by step process used to find object which is left abandoned. A region of interest (ROI) is a portion of an image that one want to filter or perform some other operation on. The pixels that define the ROI are set to 1 and all other pixels set to 0. ROI defines the borders of objects under consideration. Store background subsystem uses the first frame of the video as the background. To improve accuracy, here system uses both intensity and color information for the background subtraction operation.

During this operation, Cb and Cr color channels are stored in a complex array. The Detect subsystem contains the main algorithm. Inside this subsystem, the Luminance Segmentation and Color Segmentation subsystems perform background subtraction using the intensity and color data. The subsystem combines these two segmentation results using a binary OR operator. Abandoned Object Tracker subsystem, uses the object statistics to determine which objects are stationary.

3.2 Background Subtraction

Background subtraction, also known as Foreground Detection, is a technique in the fields of image processing and computer vision wherein an image's foreground is extracted for further processing (object recognition etc.). Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground. After the stage of image pre-processing (which may include image denoising, post processing like morphology etc.) object localization is required which may make use of this technique.

Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called “background image”, or “background model”. Background subtraction is mostly done if the image in question is a part of a video stream. Background subtraction provides important cues for numerous applications in computer vision, for example surveillance tracking or human poses estimation. However, background subtraction is generally based on a static background hypothesis which is often not applicable in real environments. With indoor scenes, reflections or animated images on screens lead to background changes. In a same way, due to wind, rain or illumination changes brought by weather, static backgrounds methods have difficulties with outdoor scenes.

A robust background subtraction algorithm should be able to handle lighting changes, repetitive motions from clutter and long-term scene changes. The following analyses make use of the function of V(x,y,t) as a video sequence where t is the time dimension, x and y are the pixel location variables.

A motion detection algorithm begins with the segmentation part where foreground or moving objects are segmented from the background. The simplest way to implement this is to take an image as background and take the frames obtained at the time t, denoted by I(t) to compare with the background image denoted by B. Here using simple arithmetic calculations, it can segment out the objects simply by using image subtraction technique of computer vision meaning for each pixels in I(t), take the pixel value denoted by P[I(t)] and subtract it with the corresponding pixels at the same position on the background image denoted as P[B].
In mathematical equation, it is written as in equation 4.1

$$P[F(t)] = P[I(t)] – P[B] \qquad (4.1)$$

The background is assumed to be the frame at time t. This difference image would only show some intensity for the pixel locations which have changed in the two frames.

Though we have seemingly removed the background, this approach will only work for cases where all foreground pixels are moving and all background pixels are static. A threshold "Threshold" is put on this difference image to improve the subtraction which is shown in equation 4.2

$$\left| P[F(t)] - P[F(t+1)] \right| > \text{Threshold} \qquad (4.2)$$

This means that the difference image's pixels' intensities are 'thresholded' or filtered on the basis of value of Threshold. The accuracy of this approach is dependent on speed of movement in the scene. Faster movements may require higher thresholds.

a. Mean filter:

For calculating the image containing only the background, a series of preceding images are averaged. For calculating the background image at the instant t, use the equation 4.3

$$B(x,y) = \frac{1}{N}\sum_{i=1}^{N} V(x,y,t-i) \qquad (4.3)$$

Where N is the number of preceding images taken for averaging. This averaging refers to averaging corresponding pixels in the given images. N would depend on the video speed (number of images per second in the video) and the amount of movement in the video.

After calculating the background B(x,y), then subtract it from the image V(x,y,t) at time t = t and threshold it. Thus the foreground is given in equation 4.4

$$\left| V(x,y) - B(x,y) \right| > Th \qquad (4.4)$$

Where Th is threshold. Similarly it can also use median instead of mean in the above calculation of B(x,y).

b .Running Gaussian average:

For this method, Wren et al. propose fitting a Gaussian probabilistic density function (pdf) on the most recent n frames. In order to avoid fitting the pdf from scratch at each new frame time t, a running (or on-line cumulative) average is computed. The pdf of every pixel is characterized by mean μ_t and variance σ_t^2. The possible initial condition (assuming that initially every pixel is background): is shown in equation 4.5 and equation 4.6

$$\mu_0 = I_0 \qquad (4.5)$$

$$\sigma_0^2 = \text{<Some default value>} \qquad (4.6)$$

Where I_t is the value of the pixel's intensity at time t. In order to initialize variance, for example, use the variance in x and y from a small window around each pixel.

Note that background may change over time (e.g. due to illumination changes or non-static background objects). To accommodate for that change, at every frame t, every pixel's mean in equation 4.7 variance in equation 4.8 and distance in equation 4.9 must be updated, as follows:

$$\mu_t = \rho I_t + (1 - \rho)\, \mu_{t-1} \qquad (4.7)$$

$$\sigma_t^2 = d^2\rho + (1 - \rho)\, \sigma_{t-1}^2 \qquad (4.8)$$

$$d = \left| (I_t - \mu_t) \right| \qquad (4.9)$$

Where ρ determines the size of the temporal window that is used to fit the pdf (usually ρ = 0.01) and d is the Euclidean distance between the mean and the value of the pixel. Now classify a pixel as background if its current intensity lies within some confidence interval of its distribution's mean is given in equation 4.10 and equation 4.11

$$\frac{\left| (I_t - \mu_t) \right|}{\sigma_t} > K \rightarrow \text{Foreground} \qquad (4.10)$$

$$\frac{\left| (I_t - \mu_t) \right|}{\sigma_t} \leq K \rightarrow \text{Background} \qquad (4.11)$$

Where the parameter is a free threshold (usually K = 2.5). A larger value for k allows for more dynamic background, while a smaller increases the probability of a transition from background to foreground due to more subtle changes.

In a variant of the method, a pixel's distribution is only updated if it is classified as background. This is to prevent newly introduced foreground objects from fading into the background. The update formula for the mean is changed accordingly in equation 4.12

$$\mu_t = M\mu_{t-1} + (1 - M)\,(I_t\rho + (1 - \rho)\, \mu_{t-1}) \qquad (4.12)$$

Where M = 1 when I_t is considered foreground and M = 0 otherwise. So when M = 1, that is, when the pixel is detected as foreground, the mean will stay the same. As a result, a pixel, once it has become foreground, can only become background again when the intensity value gets close to what it was before turning foreground. This method, however, has several issues: It only works if all pixels are initially background pixels (or foreground pixels are annotated as such). Once the foreground object is gone, the new background intensity might not be recognized as such anymore.

3.3 Backtracking Algorithm

Backtracking is a general algorithm for finding all (or some) solutions to some computational problems, notably constraint satisfaction problems, that incrementally builds candidates to the solutions, and abandons each partial candidate c ("backtracks") as soon as it determines that c cannot possibly be completed to a valid solution. Backtracking can be applied only for problems which admit the concept of a "partial candidate solution" and a relatively quick test of whether it can possibly be completed to a valid solution. It is useless, for example, for locating a given value in an unordered table. When it is applicable, however, backtracking is often much faster than brute force enumeration of all complete candidates, since it can eliminate a large number of candidates with a single test.

a. Description of the method:

The backtracking algorithm enumerates a set of partial candidates that, in principle, could be completed in various ways to give all the possible solutions to the given problem. The completion is done incrementally, by a sequence of candidate extension steps. Conceptually, the partial candidates are represented as the nodes of a tree structure, the potential search tree. Each partial candidate is the parent of the candidates that differ from it by a single extension step; the leaves of the tree are the partial candidates that cannot be extended any further.

The backtracking algorithm traverses this search tree recursively, from the root down, in depth-first order. At each node c, the algorithm checks whether c can be completed to a valid solution. If it cannot, the whole sub-tree rooted at c is skipped (pruned). Otherwise, the algorithm (1) checks whether c itself is a valid solution, and if so reports it to the user; and (2) recursively enumerates all sub trees of c. The two tests and the children of each node are defined by user-given procedures. Therefore, the actual search tree that is traversed by the algorithm is only a part of the potential tree. The total cost of the algorithm is the number of nodes of the actual tree times the cost of obtaining and processing each node. This fact should be considered when choosing the potential search tree and implementing the pruning test.

b. Pseudo code:

In order to apply backtracking to a specific class of problems, one must provide the data P for the particular instance of the problem that is to be solved, and six procedural parameters, root, reject, accept, first, next, and output. These procedures should take the instance data P as a parameter and should do the following:

- root(P): return the partial candidate at the root of the search tree.
- reject(P,c): return true only if the partial candidate c is not worth completing.
- accept(P,c): return true if c is a solution of P, and false otherwise.
- first(P,c): generate the first extension of candidate c.
- next(P,s): generate the next alternative extension of a candidate, after the extension s.
- output(P,c): use the solution c of P, as appropriate to the application.

4. RESULT AND DISCUSSION

The result is shown in both software and in hardware. In the software section the images of identified abandoned object is shown using MATLAB unit. In the hardware the abandoned object is indicated using AT89s52 microcontroller which is interfaced through UART to the PC.

4.1 Software Section

The abandoned object is first identified by running the MATLAB code. The target will be uniquely shown by using blob marking.

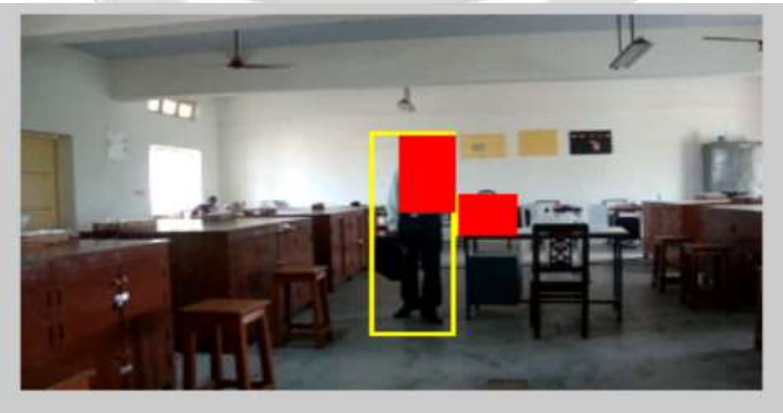

Figure 4.1 Identification of person and object

Figure 4.1 shows the person and objects that are to be tracked to detect the abandoned object. Initially all the parameters needed to detect the abandoned object are to be set. Then initialize video display and also a window for the video display. Apply detection and tracking algorithms to each frame of the video. The approach is to first find all objects in the video then to use tracking algorithm to identify objects that have been stationary for a specified length of time. Finally, overlay this information on the original video. Add rectangles for current objects. Overlay red on objects that have been abandoned.

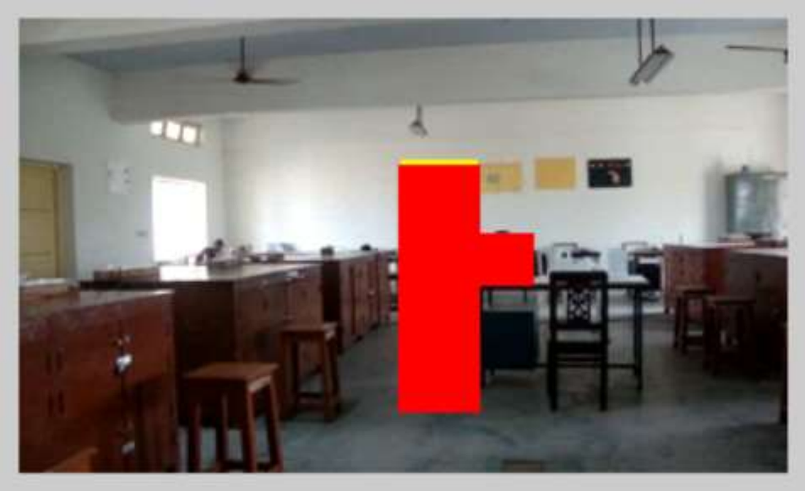

Figure 4.2 Masking of person and object

Figure 4.2 show an operation of masking two objects. Structuring element have been used to help find objects. Compute the absolute difference between frames. Create object masks for Y and a combined one for CbCr.

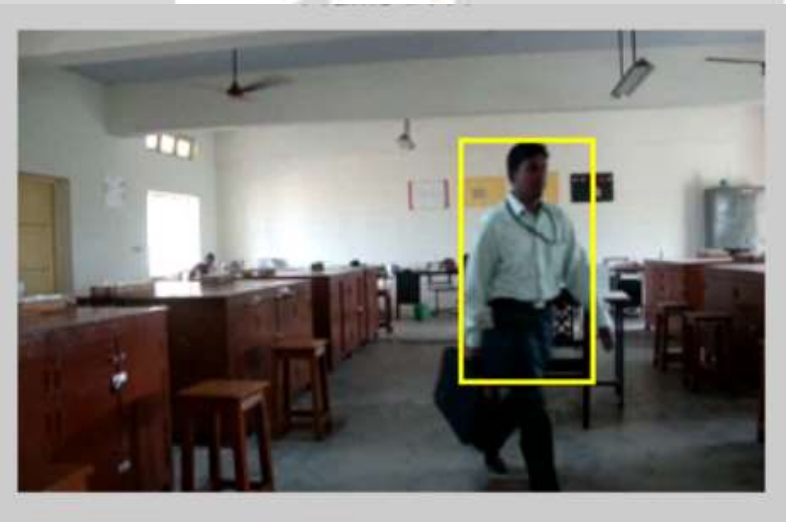

Figure 4.3 Tracing of person

Figure 4.3 shows the tracing of moving object. The object tracing is done by object tracker. It tracks the objects from frame to frame. For each object in the current frame, scan through existing list to find matches. If no match exists, then add to tracking list (Object List). Here in figure 6.3 the person in motion have been traced.

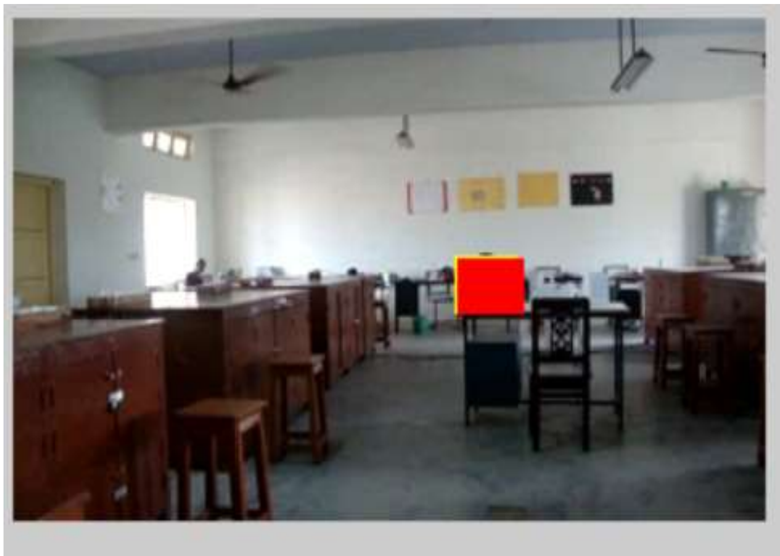

Figure 4.4 Abandoned object detection

Figure 4.4 show the detection of abandoned object which have been masked. Output the bounding box of objects that have been stationary for more frames than "alarm Count" and declare it as an abandoned object.

4.2 Hardware Section

If an abandoned object is detected then the hardware interfaced through serial communication port (UART) sends the alert message to the monitoring person through GSM unit, also gives an alarm signal and displays the information about abandoned object in the LCD as shown in figure 4.5.

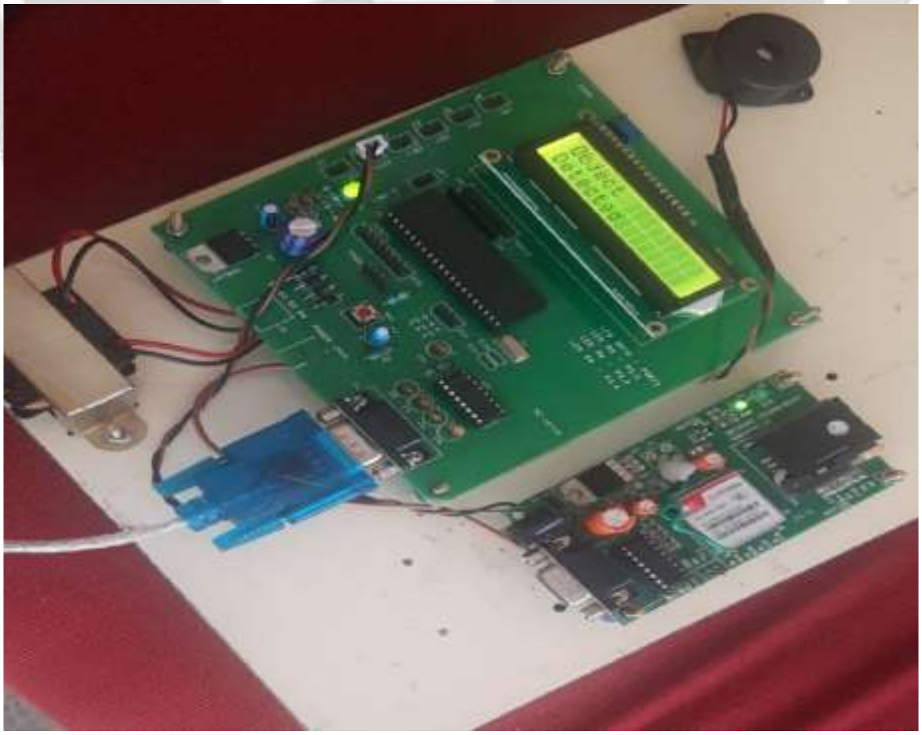

Figure 4.5 LCD displaying object detected message

5. CONCLUSION

The proposed system presents a temporal consistency model combining a back-tracking algorithm for abandoned object detection. Characteristics of the proposed approach are summarized as follows:

1) The temporal consistency model is described by a very simple FSM. It exploits the temporal transition pattern generated by short- and long-term background models, which can accurately identify static foreground objects.

2) The back-tracking algorithm iteratively tracks the luggage owner by using spatial-temporal windows to efficiently verify left-luggage events.

In the future, the proposed system plan to enhance a method for handling more challenging situations such as sudden changes in lighting and overly crowded scenes.

6. REFERENCES

[1] J. Mart´ınez-del Rinc´on, J. E. Herrero-Jaraba, J. R. G´omez, and C. Orrite- Urunuela, "Automatic left luggage detection and tracking using multicamera ukf," in Proc. IEEE Workshop on PETS, 2006, pp. 59–66.

[2] F. Porikli, Y. Ivanov, and T. Haga, "Robust abandoned object detection using dual foregrounds," EURASIP Journal on Advances in Signal Processing, p. 30, 2008.

[3] R. H. Evangelio, T. Senst, and T. Sikora, "Detection of static objects for the task of video surveillance," in Proc. WACV, 2011, pp. 534–540.

[4] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M.-T. Sun, "Robust detection of abandoned and removed objects in complex surveillance videos," IEEE TSMC Part C, vol. 41, no. 5, pp. 565–576, 2011.

[5] Q. Fan and S. Pankanti, "Modeling of temporarily static objects for robust abandoned object detection in urban surveillance," in Proc. AVSS, 2011, pp. 36–41.

[6] Q. Fan, P. Gabbur, and S. Pankanti, "Relative attributes for large-scale abandoned object detection," in Proc. ICCV, 2013, pp. 2736–2743.

[7] H.-H. Liao, J.-Y. Chang, and L.-G. Chen, "A localized approach to abandoned luggage detection with foreground-mask sampling," in Proc. AVSS, 2008, pp. 132–139.

[8] J. Pan, Q. Fan, and S. Pankanti, "Robust abandoned object detection using region-level analysis," in Proc. ICIP, 2011, pp. 3597–3600.

[9] F. Lv, X. Song, B. Wu, V. K. Singh, and R. Nevatia, "Left-luggage detection using bayesian inference," in Proc. IEEE Workshop on PETS, 2006, pp. 83–90.

[10] L. Li, R. Luo, R. Ma, W. Huang, and K. Leman, "Evaluation of an ivs system for abandoned object detection on pets 2006 datasets," in Proc. IEEE Workshop on PETS, 2006, pp. 91–98.