Abandonded Object Detection

Group 7

Pradeep Turan
201601064
CSE, UG2
Indian Institute of Information Technology, Sricity
pradeep.t16@iiits.in

***Abstract*—Abandoned object detection is one of the most practically useful areas of computer vision due to its application in automated video surveillance systems for the detection of suspicious activities that might endanger public safety, especially in crowded places like airports, railway stations, shopping malls, movie theatres and the like. An abandoned object is defined as one that has been lying stationary at a certain place with no apparent human attendance for an extended period of time. Such objects are usually inconspicuous commonplace objects that people often carry around including backpacks, suitcases and boxes. Detection of abandoned objects is of prime importance in uncovering and forestalling terrorist activities since it is a reasonable supposition that an abandoned object, if left behind on purpose, may be hiding dangerous items like explosives.**

I. Introduction

An automatic abandoned object detection system typically uses a combination of background subtraction and object tracking to look for certain pre-defined patterns of activity that occur when an object is left behind by its owner.[11] 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.

II. Objectives

A. *It should be able to identify abandoned objects in real time and therefore must employ efficient and computationally inexpensive algorithms.*

B. *It should be robust against illumination changes, cluttered backgrounds, occlusions, ghost effects and rapidly varying scenes.*

C. *It should try to maximize the detection rate while at the same time minimizing false positives.*

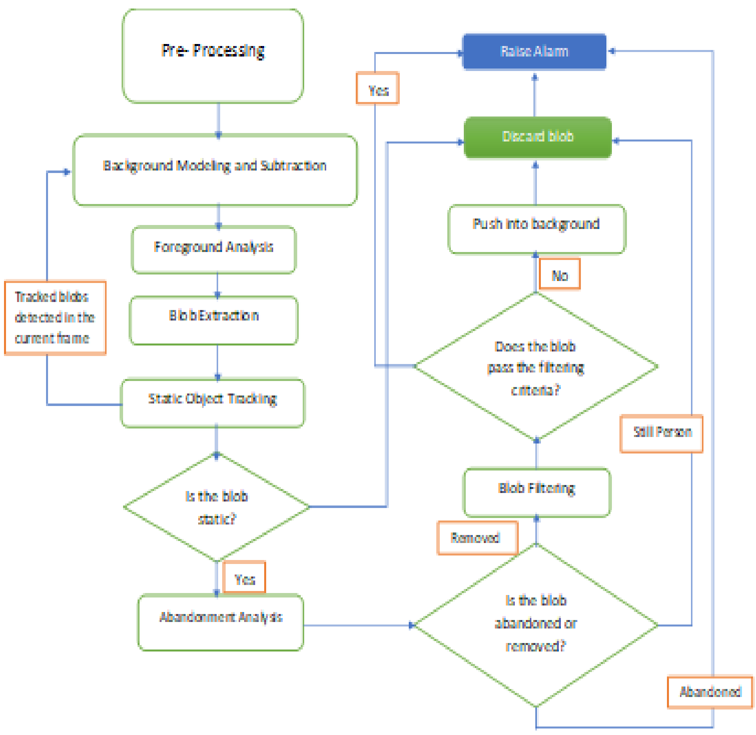
designations.

III. Items to be Considered

Includes all types of language that can be carried by hand. But We would consider mainly the following dataset items.

- Briefcase
- Suitcase
- 25 liters rucksack
- 70 liters backpack
- Ski gear carrier

IV. Flowchart of System Methodology



V. Analysis and Development

- Background modeling and subtraction (BGS)

- Pre-processing
- Foreground analysis
- Blob extraction
- Blob tracking
- Abandonment analysis

A. Background modeling and Subtraction (BGS):

- The first frame of the video is taken as the background.
- We will perform YCbCr for the background and the current frame.

BGS can be done by various algorithms, but here GMM is followed.

Gaussian mixture model: It assumes that the overall intensity at any pixel at each instant is produced by a combination of background and foreground processes, and each such process can be modeled by a single Gaussian probability distribution function. [1][9][10][12] For each pixel in the current frame, the probability of observing the current intensity is given by:

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

Here, K is the no. of distributions (K=3 here); $\omega_{i,t}$ is the weight associated with the i^{th} distribution at time t while $\mu_{i,t}$ is the mean and $\Sigma_{i,t}$ is the co-variance matrix of this distribution, η is the exponential Gaussian probability density function given by:

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

Here, n is the dimensionality of each pixel's intensity value (e.g. n=1 for grayscale image and n=3 for RGB image). In order to avoid a costly matrix inversion and decrease computation cost, it is also assumed that the red, green and blue channels in the input images are not only independent but also have the same variance $\varsigma^2_{,k,t}$, so the variance matrix becomes:

$$\Sigma_{k,t} = \sigma_{k,t}^2 \mathbf{I}$$

The K Gaussian distributions are always ordered in the decreasing order of their contribution to the current background model. This contribution is measured by the ratio ω/ς under the assumption that higher is the weight and lower is the variance of a distribution, more is the likelihood that it represents the background process.
For each pixel in an incoming frame, its intensity value is compared with the means of the existing distributions starting from the first one and a match is said to be obtained if its Euclidean distance from the mean is less than m standard deviations (m=3 is used here), i.e. it satisfies the following condition:

$$|I_t - \mu_{k,t}| < m * \sigma_{k,t}$$

Here, It is the pixel intensity while $\mu_{k,t}$ and $\varsigma_{k,t}$ are the mean and standard deviation of the k_{th} distribution at time t.
Since the background model is dynamic, it needs to be updated with each frame. While the weights are updated for all distributions, the mean and variance are updated only for the matched distributions. Following are the standard update equations used for this purpose:

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

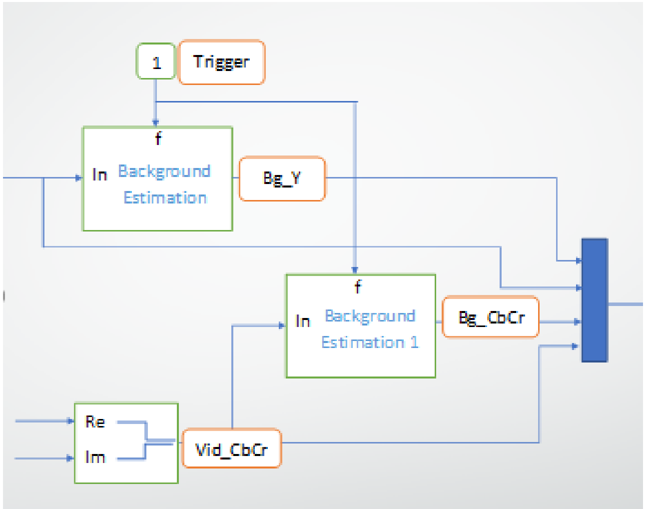
$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t-\mu_t)^T(X_t-\mu_t)$$

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

Here X_t is the pixel intensity while $M_{k,t}$=1 for matched distributions and 0 for unmatched ones while ρ and α are learning rates. In the current work ρ and α are related as:

$$\rho_{k,t} = \frac{\alpha}{\omega_{k,t}}$$

Thus, while α is fixed for all distributions (α = 0.001 used here), ρ is smaller for higher weighted distributions. If none of the existing distributions match the current intensity, the least probable distribution (i.e. with the smallest value of ω/ς) is replaced by a new distribution with a high initial variance, low prior weight and the new intensity value as its mean.
This is the simplest way for background subtraction.



B. Pre-Processing:

Pre-Processing involves 2 functions:

- Contrast Enhancement
- Noise Reduction

Contrast Enhancement: It improves the quality of low light video by normalizing the difference between the maximum & minimum intensities, using **YCbCr approach**. Convert RGB to YCbCr using "vision.ColorSpaceConverter" function. This let us convey the color information in a smaller amount of data than in the case of RGB. We perform YCbCr on the background and current frame. [8]

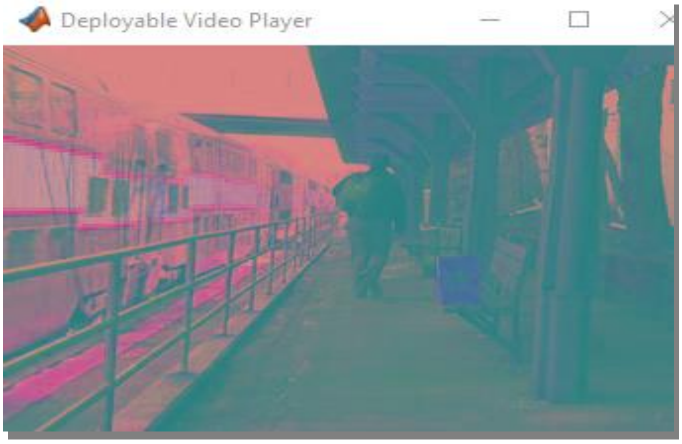


Fig.1 YCbCr Image

Noise Reduction: The resultant binary matrix is all set for Noise Reduction. It reduces the white noise present in an input frame by smoothing the frame. It is needed to control the amount of noise that becomes visible in low light videos after applying contrast enhancement.

Fig.2 Binary frame with Noise

Fig.3 Binary frame after Noise Reduction

C. Foreground analysis

The noisy portions of the BGS output may contain false foregrounds produced for example during sudden lighting

changes (a light is switched on or off) as well as actual foregrounds that are of no interest to further processing but can complicate it significantly. [5][10] Foreground regions detected due to moving shadows are important examples of this latter category. Thus, this system has a separate foreground analysis stage that takes as its input the noisy foreground mask produced by the BGS module and removes all the false and uninteresting foreground pixels from it.

This can be done by NCC (Normalized Cross-Correlation) method.[2]

$$NCC = \frac{\sum_{u \in W} I_f . I_b - \frac{1}{MN} \sum_{u \in W} I_f \sum_{u \in W} I_b}{\sqrt{(\sum_{u \in W} I_f^2 - \frac{1}{MN} (\sum_{u \in W} I_f)^2)(\sum_{u \in W} I_b^2 - \frac{1}{MN} (\sum_{u \in W} I_b)^2)}}$$

Here, W denotes the M x N neighborhood centered at that pixel while If and Ib respectively denote the current frame and background intensity values at a particular pixel. A pixel is classified as shadow if $NCC \geq Tssadow$ and $If \geq Tintensity$. The second condition has been added to avoid misclassification of very dark areas as shadows. $Tshadow$ = 0.60 and $Tintensity$ = 5 have been used in this work.

D. Blob extraction:

- The refined mask produced by the last step is subjected to a connected component detection function.
- This function i.e. **[Area, Centroid, BBox] = step(hBlob, Segmented);** gives area , centroid and Boundary box for a particular blob, discarding blobs that are smaller than a specified threshold. [3][14]

E. Blob Tracking:

- First step involves comparison of incoming frame with the existing blobs.
- If the centroid position and size(criteria) of the blobs are in conformity, then the Hit Rate of that blob is increased by 1.
- If at any instant the criteria of the blobs are not in conformity, then its Hit Rate is set to 0.
- Moreover, if any object is occluded, we will increase its Miss Rate by 1.
- A blob with Hit Rate above certain threshold is labeled as a static object; while any static object with Miss Rate more than certain user specified value will be discarded. [4][13]

F. Abandonment Analysis:

- Once the Hit Rate of any static Object exceeds Abandoned Threshold, it is labeled as an abandoned object.
- An object detected as abandoned causes an alarm to be raised immediately.

VI. ALGORITHM

- Maximum number of objects we will be tracking are 200.
- System object for reading the video with each frame of the type 'Single'. For this purpose, we have used vision.VideoFileReader function.
- Offsets for drawing bounding boxes in original input video. Here, we have bounded all moving objects in boxes, for this we used int32(repmat([roi(1), roi(2), 0, 0],[maxNumObj 1]));
- In the next step, RGB image is converted into YCbCr using vision.ColorSpaceConverter('Conversion', 'RGB to YCbCr');
- Next step is background subtraction.
- Remove noise and really small blobs using vision.MorphologicalClose('Neighborhood', strel('square',10));
- Find the blobs in the segmented images, properties of blobs also specified.
- Create system objects for players with their locations.
- Will run a loop till all the frames are completed, storing all the frames and background.
- In this step, background is subtracted on the luminosity, Chroma part of YCbCr.
- Fill in small gaps in the detected objects and club the separated image.
- Blob analysis, storing area, centroid, bounding box in a list.
- Traversing through the list of centroids of all the blobs.
- Will check if the position of a particular blob has changed or not, on the basis of the area of blob, and distance between the centroid of that blob in the background frame and the current frame, if yes, increase its count by 1, if not, increase its miss count by 1.
- If the stationary object remains there for 75 frames, it will be declared as an abandoned object, while if the stationary object, moves or occluded by some interference for more than 7 frames, it will be no longer stationary object.
- As soon as the object becomes an abandoned object, its bounding box becomes red, and an alarm is raised.

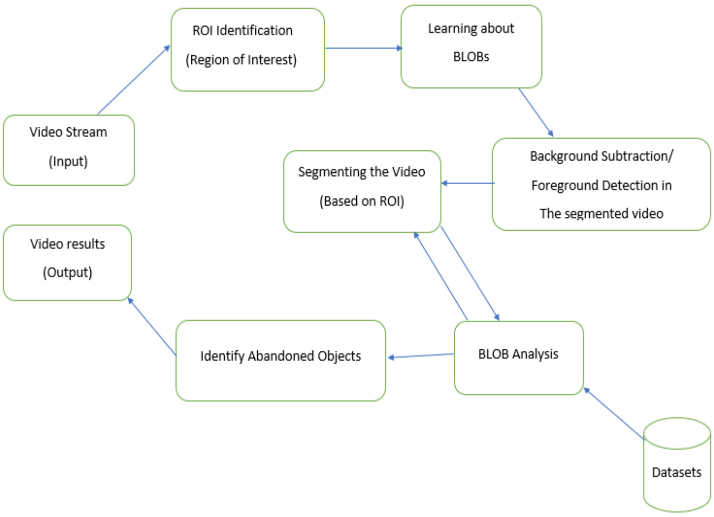


Fig.4 Algorithm basic schema

VII. DATASETS

Different datasets have been used for testing and analysis. Single abandoned object in one of the dataset video, one has no abandoned object, one has a huge crowd on a railway station with incoming and outgoing trains in the background, one dataset was in a class, one was of abandoned object in the parking area, one particular video has, switching-on and off, of the light in the room.[6][7] The last one was the toughest to deal with due to change in the intensity of light in the background.

VIII. SOFTWARES AND HARDWARES

MATLAB R2016a was used for video processing and analysis. No specific extra hardware wasn't used. DELL Inspiron 15, with Intel 6th generation i5 core processor was enough to handle the heavy MATLAB application.

IX. CONCLUSION

This paper proposed a system for automatic Abandoned object detection without any human interference. Such system proves to be efficient in public place for providing security. The application was tested on a large number of publicly available as well as custom made videos and was found to give accuracy comparable to most contemporary systems. One particularly noteworthy aspect of its performance was that it was able to detect the object in real time aspect with video speed almost equal to the real-time aspect.

X. LIMITATIONS

- Mistook Stationary person as an abandoned object. If any person doesn't show significant movement, the system may mistake it as an abandoned object.

- As mentioned earlier, the project is inefficient for immediate lightning change. [2] It may mistake some shadows as an abandoned object.
- If some passenger keeps his language near him and doesn't move it for a while, the system may assume it as an abandoned object
- It won't work efficiently for extremely poorly lit videos.
- No learning rate.

XI. Future work

In future we can go for object classification to avoid false detection.
Machine Learning feature can be added so as to prevent the detection of a stationary person as an abandoned object.
Machine learning can also help in dealing with the objects kept by their owners near them,[15] untouched and unmoved.[16]
Multiple numbers of cameras can be used for more efficient analysis of the particular area.

Acknowledgment

I would like to thank Prof. Dr. Anish Chand Turlapaty and T.A. Viswanatha Reddy who gave me such a great opportunity to work on this project. I am also thankful to my colleagues, Paras Dahiya, Sagar, Sarin Nikose, who supported me on each and every moment to out pass all the hurdles in the implementation and execution of this project.

References

[1] C. Stauffer and W. E. L. Grimson, ―Adaptive background mixture models for real-time tracking,‖ in Proc. IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 246–252, Feb. 1999. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Y. Tian, M. Lu and A. Hapapur, ―Robust and efficient foreground analysis for real-time video surveillance,‖ in Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 1182-1187, June 2005.

[3] F. Chang, C. Chen and C. Lu, ―A linear-time component labeling algorithm using contour tracing technique,‖ Computer Vision and Image Understanding, Vol. 93, Issue 2, pp. 206-220, Feb. 2004.

[4] A.Singh, S.Sawan, M.Hanmandlu, V.K.Madasu, B.C.Lovell ―An Abandoned Object Detection System Based On Dual Background Segmentation,‖ in Proc. Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 352-357, 2009.

[5] R. Singh, S. Vishwakarma, A. Agrawal and M. D. Tiwari, ―Unusual activity detection for video surveillance,‖ in Proc. 1st International Conference on Intelligent Interactive Technologies and Multimedia, pp. 297-305, Dec. 2010.

[6] PETS2006BenchmarkData,‖http://www.cvg.rdg.ac.uk/PETS2006/data.html, June 2013.

[7] PETS2007BenchmarkData, http://www.cvg.rdg.ac.uk/PETS2007/data.html, June 2013.

[8] J.C.S. Jacques, C.R. Jung and S.R. Musse, ―Background Subtraction and Shadow Detection in Grayscale Video Sequences,‖ in Proc 18th Brazilian Symposium on Computer Graphics and Image Processing, pp. 189-196,Oct.2005.

[9] Z. Zivkovic, ―Improved adaptive Gausian mixture model for background subtraction,‖ in Proc. International Conference on Pattern Recognition, Vol. 2, pp. 28–31, Aug. 2004.

10] A.Elgammal, D. Harwood, R. Duraiswami and L.S. Davis, ―Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance,‖ in Proc. The IEEE, Vol. 90, Issue7,pp.1151-1163,Jul.2002.

[11] Bhargava, M.; Chia-Chih Chen; Ryoo, M.S.; Aggarwal, J.K. ―Detection of Abandoned Objects in Crowded Environments.‖ In Proc. IEEE Conference on Advanced Video and Signal Based Surveillance, pp.271-276,Sept.2007.

[12] Chih-Yang Lin and Wen-Hao Wang ―An Abandoned Objects Management System Based on the Gaussian Mixture Model.‖ International Conference on Convergence and Hybrid Information Technology,2008.

[13] Ying-li Tian , Rogerio Feris , Arun Hampapur ―Modeling of temporarily static objects for robust abandoned object detection in urban surveillance.‖ in Proc. Eight IEEE International Conference on Advanced Video and Signal-Based Surveillance, pp. 36-41, Sept. 2011.

[14] OpenCV based blob extraction library ―cvBlobsLib‖ available at http://opencv.willowgarage.com/wiki/cvBlobsLib.

[15] Silvia Ferrando, Gianluca Gera, Massimo Massa, Carlo Regazzoni ―A New Method for Real Time Abandoned Object Detection and Owner Tracking.‖, IEEE International Conference on Image Processing, pp. 3329-3332,Oct.2006.

[16] J. Canny, ―A Computational Approach to Edge Detection,‖ IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, Issue 6,pp.679-698,Nov.1986.