Detection of Abandoned Objects in Real Time

Jagdish Lal Raheja, Chaitanya Malireddy,Aniket Singh
Jagdish@ceeri.ernet.in, chaitanyaaa@gmail.com, sinkin.heat@gmail.com

***Abstract*—In this paper we describe an intuitive model for accurate and efficient detection of abandoned objects. The system is built on the backbone of the Gaussian Mixture Model for background subtraction. We apply a simple and robust method for shadow detection . Next, detection of stable blobs is carried out using Mathew et al's method, and an important modification is suggested that is resistant to temporary occlusions, and removes unnecessary parameters from the model. Changes to the background itself are identified via a 'ghost' removal procedure that can distinguish between true and removed objects. Results of testing the model are presented with conclusions.**

Keywords- Background Subtraction, Gaussian Mixture Model, Shadow Removal

I. Introduction

Detecting abandoned objects is an area of active research in video surveillance. Abandoned objects pose a security risk in crowded public places. Smart surveillance systems such as the one presented in this paper can be used to alert security officers monitoring live video to potentially dangerous situations.

Most existing approaches focus on tracking multiple objects in a scene. In [1-4], a low-level stage of foreground segmentation is followed by a stage of object tracking. Such an approach is not only computationally costly but also very challenging in a complex environment with a large number of occlusions.

Moreover, in [3], Ferrando et al define an abandoned object as a static 'non-human' blob following a split from a composite 'human + object' blob. In [4] Bhargava et al determine the owner of the object by backtracking to the drop-off point and try to locate him in the scene. In this paper, however, we simply define an abandoned object to be any stationary object that has introduced in the scene after the system is started.

A comprehensive review of many background subtraction based foreground static object detection methods is presented in [5].The Mixture of Gaussians method proposed by Stauffer and Grimson in [6] has proven to be highly successful in modeling complex backgrounds. We use the information inherent in this model to detect stable objects. Our work in this respect has largely been inspired by the work of Mathew et al in [7].

II. System overview

Fig 1. shows the system flowchart. The system has four main modules: (a) background subtraction; (b) shadow removal; (c) static region detection; (d) static object type detection (abandoned/ removed).

The system is based on a Mixture of Gaussians background subtraction scheme. First background subtraction is performed to detect any new object that may have entered the scene. After that we determine which objects remain stationary for a certain number of frames. The final module differentiates between removed and abandoned objects. The system notifies the user of an abandoned object by raising an alarm.

The following sections will elaborate on each of these steps.

III. BACKGROUND SUBTRACTION

The background subtraction method in [6] uses Gaussian Mixture Models (GMM), which are commonly used in machine learning and classification. A GMM is a mixture probability density function (pdf) where the GMM itself is a linear combination of K Gaussian pdfs. Details of the model are presented in the following sections.

A. Basic Equations

A mixture of K Gaussians is used to model the time series of values observed at a particular pixel. The probability of occurrence of the current pixel value is given by

$$P(Z) = \sum_{i=1}^{k} w_i \, N(\mu_t, \Sigma_t, Z) \qquad (1)$$

N is the Gaussian probability density function whose mean vector is μ and covariance isΣ.

w_i is the weight of the i^{th} Gaussian such that $\Sigma w_i = 1$. The covariance matrix is assumed to be of the form $\Sigma = \sigma^2 I$ for computational reasons.

B. Parameter Updates

The new pixel value $\boldsymbol{Z_t}$ is checked against each Gaussian. A Gaussian is labeled as matched if

$$\|Z - \mu_h\| < d\sigma_h \qquad (2)$$

Then its parameters may be updated as follows:

$$w_{i,t} = (1 - \alpha) * w_{i,t-1} + \alpha * M_{i,t} \qquad (3)$$

$$\mu_t = (1 - \rho) * \mu_{t-1} + \rho * Z_t \qquad (4)$$

$$\sigma_t^2 = (1 - \rho) * \sigma_{t-1}^2 + \qquad (5)$$
$$\rho * (Z_t - \mu_t)^T * (Z_t - \mu_t)$$

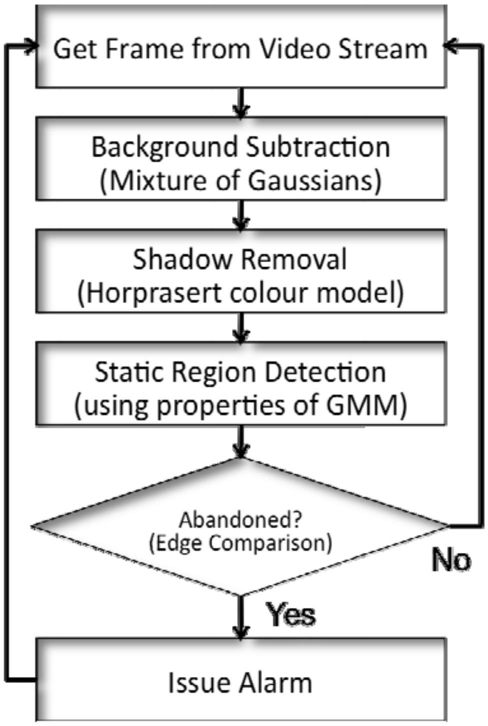
$$\rho = \alpha * N(\mu_{t-1}, \Sigma_{t-1}, Z_t) \qquad (6)$$

Where α is the learning rate for the weights.

If a Gaussian is labeled as unmatched only its weight is decreased as

$$w_{i,t} = (1 - \alpha) * w_{i,t-1} \qquad (7)$$

If none of the Gaussians match, the one with the lowest weight is replaced with as mean and a high initial standard deviation.



Get Frame from Video Stream
Background Subtraction (Mixture of Gaussians)
Shadow Removal (Horprasert colour model)
Static Region Detection (using properties of GMM)
Abandoned? (Edge Comparison)
No
Yes
Issue Alarm

Figure 1. System Flowchart.

C. *Background Heuristic*

The rank of a Gaussian is defined as w/σ. This value gets higher if the distribution has low standard deviation and it has matched many times. When the Gaussians are sorted in a list by decreasing value of rank, the first is more likely to be background. The first B Gaussians that satisfy (8) are thought to represent the background.

(8)

The Gaussian mixture model (GMM) is adaptive; it can incorporate slow illumination changes and the removal and addition of objects into the background. Further it can handle repetitive background changes like swaying branches, a flickering computer monitor etc. The higher the value of T in (8), the higher is the probability of a multi-modal background.

IV. SHADOW REMOVAL

Shadows detected as foreground can cause several problems when extracting and labeling objects, two examples are object shape distortion and several objects merging together. It is especially crucial that these problems be avoided given the method we use to decide whether an object is abandoned or not (see Section VI).

In most traditional methods [9] the image is first transformed to a color space that segregates the chromaticity information from the intensity. In the HSV color space, the hypothesis is that a shadowed pixel value's value and saturation will decrease while the hue remains relatively constant. In CIELAB, the luminance component should decrease and the chromaticity coordinates should remain relatively constant.

However, the color model of Horprasert et al [8] gave the best results in the test conducted. It doesn't require any complicated conversion formulae like in the case of HSV and CIELab. Also the simple choice of parameters is a distinct advantage.

In this model each pixel value in the RGB color space is assumed to lie on a chromaticity line, which connects the pixel value and the origin. The authors specify a way to calculate the deviation of a foreground pixel value from the background value. The foreground value is compared with the means of each of the B background Gaussians in (8).

The brightness distortion (BD) and chromaticity distortion (CD) are defined as (see Fig. 2):

— (9)

Shadow points (sp) can now be ascertained as:

(10)

In Fig. 2, OB is the background RGB vector and OF is the foreground RGB vector. FP is the perpendicular dropped from F onto OB. The shaded cylinder is the locus of all shadow color values.

V. STATIC REGION DETECTION

A. *Review of Mathew, Yu and Zhang's scheme*

In [7] Mathew et al exploit the information in the state transition history of the Gaussians of the GMM to detect stable objects (see Fig. 3).

When a new foreground object is encountered, a new Gaussian centered on the observed pixel is created. If the object is transient, the Gaussian will persist in the FG state. Otherwise it will progress to the BG state and then on to the BDG state.

The authors make note of the time a new Gaussian is created, T_{CR} and the time the Gaussian in the BDG state changes, T_{BDG}. They stipulate the following conditions before a pixel can be considered stable.

(11)

(12)

(13)

FG Gaussians that have not been used for a long time are 'deleted' by setting their mean to a large negative value (see Fig 3). This ensures that a more recent creation time is recorded in T_{CR} for Gaussians representing a re-entering object.

We take this idea as our starting point and build upon it. We introduce a more intuitive implementation of this scheme.

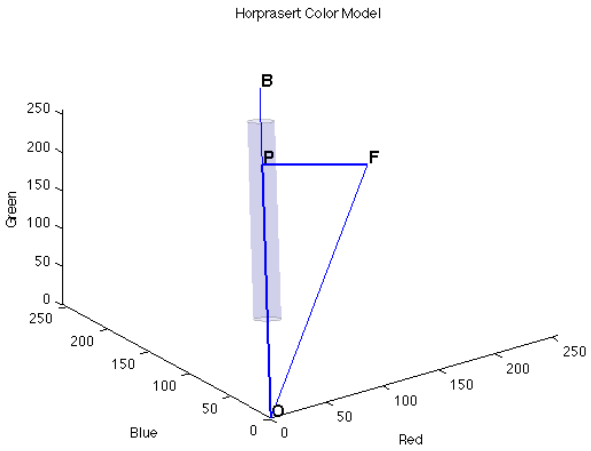


Figure 2. Brightness and chromaticity distortion in the RGB color space.

B. Modifications to the algorithm

We don't physically rearrange the Gaussians at the end of background subtraction (see section III.C).Therefore we can identify each Gaussian at any time by its index (1, 2, ..., K). Rather than record the time of occurrence of various events, we directly track the progression of a Gaussian through the various states.

The index of the Gaussian of interest is stored in *trkIdx*. A Gaussian is in the BDG state if it occupies the first position when the Gaussians are ordered according to their ranks. Similarly it is in the FG state if it is at the last position. It is therefore easy to detect when the Gaussian has reached the BDG state. Now the variable *ctr* counts the number of frames it is in BDG state. *ctr* is reset to zero when the Gaussian is displaced from the BDG state. If *ctr* exceeds and the weight of this Gaussian is greater than 0.5, the pixel is considered stable.

C. Temporary occlusions

Transient objects occluding a stable object will cause new Gaussians to be created, but we must not start tracking them right away. Whenever a non-background Gaussian matches, its index is stored in *crIdx*. Initially *trkIdx* is set equal to *crIdx*. Thereafter the value in *crIdx* is copied into *trkIdx* only if the weight of the Gaussian represented by the former has grown larger than that of the latter.

In comparison to Mathew et al, there is no need to introduce an artificial 'deletion' state based on timing constraints. In this manner temporary occlusions are handled with ease and efficiency. A similar issue is the dependence of the model on (see (11)). There are no concrete guidelines for estimating this parameter, requiring a 'hit and trial' approach. In eliminating we provide a stronger foundation to the model.

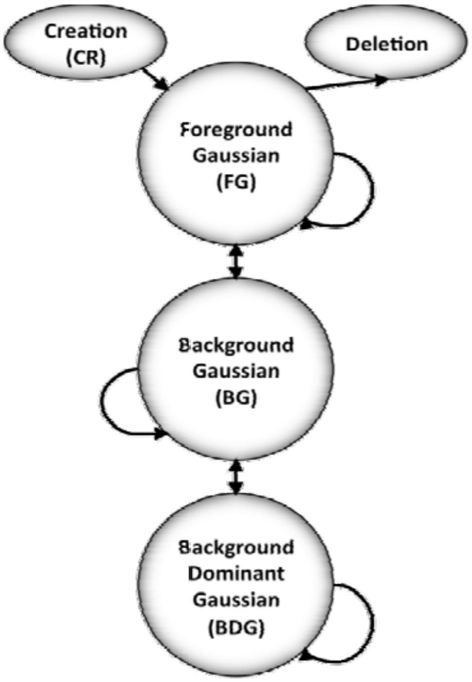


Figure 3. GMM state level transitions.

VI. STATIC OBJECT TYPE DETECTION: ABANDONED OR REMOVED

A common problem in any background subtraction based system is that of removed objects. When an object that was originally part of the background is removed, the region uncovered behind it is detected as foreground. The same region can later come up as a stable object, a misclassification referred to as a 'ghost'.

At the pixel level there is no way to distinguish between new objects and revealed background. The edge-matching solution we propose resembles that of Martiriggiano et al in [10].

Let the number of pixels in the edges of the segmented stable blob be *nBrdEg*. The edges in the current frame are detected using a canny edge detector. The edges of the blob are dilated to get a 'fleshy border' since they generally do not correspond very well with any real edges (see Fig 4). We then take the intersection of this 'border' image and the frame edge image. Let the number of 'on' pixels in the result be *nObjEg*. The ratio *isObj* is defined as . If *isObj* is less than a certain threshold the blob is discarded as a ghost.

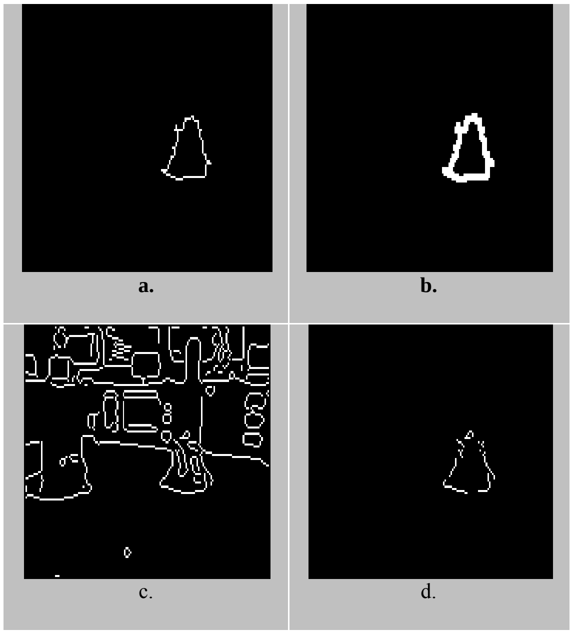


Figure 4. Distinguishing between abandoned and removed objects.(a) Blob edges. (b) Dilated blob edges. (c) Edges detected in the frame. (d) Binary and of (b) and (c).

VII. RESULTS

This section highlights the tests performed on the algorithm at the various steps of the development process. The algorithm, runs at about 18 fps on 3 gHz 3gb Pentium IV computer with Windows Vista and MATLAB 2008a.

Fig. 5 shows the results of shadow detection. The figure shows the actual scene accompanied by the result of the segmentation. The foreground pixels are red, whereas the shadow pixels are green. The results obtained using Horprasert model are satisfactory. However, as can be seen from Fig. 5 2^{nd} row, there is some tendency for shadow regions to encroach upon the object.

Next, the detection of abandoned objects is shown in Fig. 6. First we have the undisturbed background showing which objects were present in the scene at the start. Then a bag is introduced into the scene. After the requisite number of frames has passed, it is detected as a stable object, as seen in the last row.

Finally, we have a combination of a true and a removed object in Fig.7. First the chair which was part of the background is removed. Then a second bag is left on the floor. We see that only the bag is registered as an abandoned object while the chair is not detected. Since there are no edges in the frame that align enough with those of the blob corresponding to the chair, it was dismissed as a removed object.

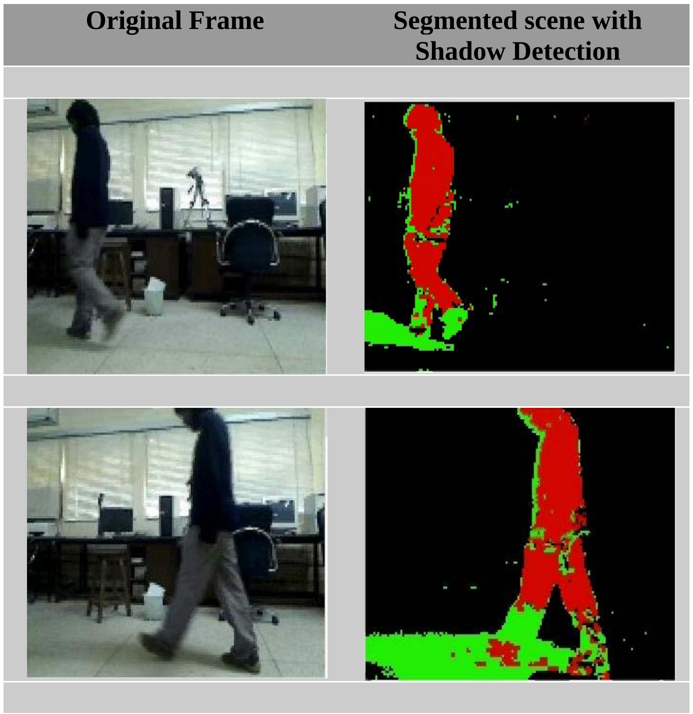


Figure 5. Results of shadow detection.

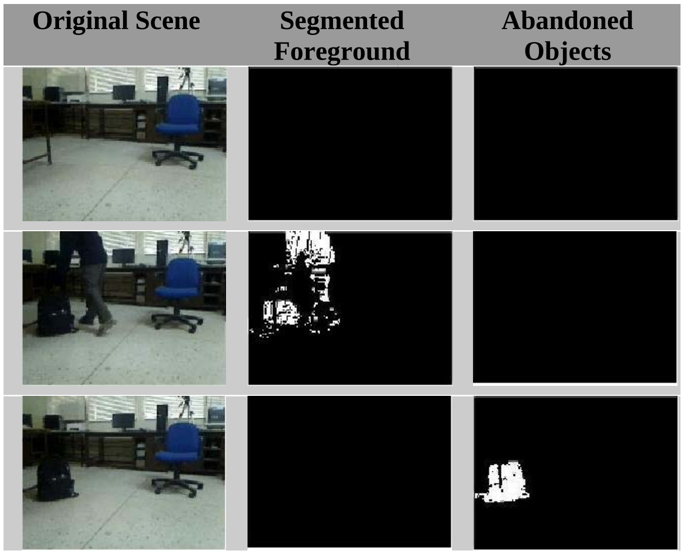


Figure 6. Abandoned object detection.

VIII. CONCLUSIONS

We have described a robust system for analyzing and extracting information about abandoned objects from foregrounds. First the system is modeled as a Gaussian mixture model. To get more accurate segmentation shadow detection and removal is performed on the foreground obtained. Then stable pixels are detected and using correspondence between scene and blob edges we assert whether an object has been abandoned or removed. The test results show the reliability of the system in real world applications.

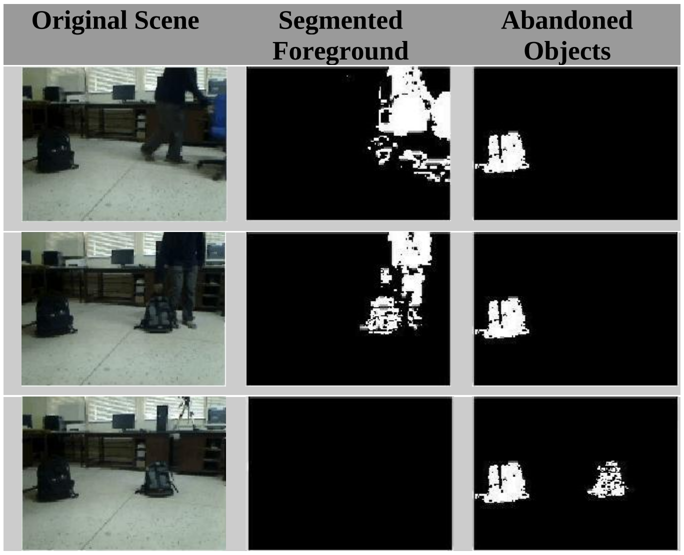


Figure 7. Processing of abandoned and removed objects.

References

[1] N.Bird, S. Atev,N. Caramelli, R. Martin, O. Masoud, and N. Papanikolopoulos. “Real time, online detection of abandoned objects in public areas”, Proc. Of AVSS, 2008, pp 3775-3780.

[2] J.C.SanMigueland J.M.Martinez, “Robust unattended and stolen object detection by fusing simple algorithms”, *Proc.Of AVSS, 2008*, pp 18-25.

[3] S.Ferrando, G. Gera and C.Regazzoni.“Classification of unattended and stolen objects in video-surveillance systems”, *Proc.Of AVSS, 2006*, pp 21-21.

[4] M.Bhargava, C.C. Chen, M.S.Ryoo and J.K. Aggarwal.“Detection of abandoned objects in crowded environments”, *Proc.Of AVSS, 2007,* pp 271-276.

[5] A.Bayona, J.C. SanMigueland J.K.Martinez. “Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques”, Proc. Of AVSS, 2009, pp 25-30.

[6] C.Stauffer and W.E.L. Grimson. “Adaptive background mixture models for real-time tracking”, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, Vol. 2 (06 August 1999), pp. 246-252

[7] R.Mathew, Z. Yu, and J. Zhang, “Detecting new stable objects in surveillance video”, IEEE 7th workshop on Multimedia Signal Processing, 2005, pp 1-4.

[8] T. Horprasert, D. Harwood, and L. Davis, “A statistical approach for real-time robust background subtraction and shadow detection”, Proc. IEEE Int’l Conference on Computer Vision ’99, FRAME-RATE Workshop 1999

[9] J. Wood, “Statistical background models with shadow removal for video based tracking”, Thesis, Helsinki University of Technology in Linkopings,2007.

[10] T.Martiriggiano, A. Caroppo, M.Leo, P.Spagnolo and T. D’Orazio, “An innovative approach for abandoned or removed objects detection”, Proc.Of ISCCSP, 2006.