2014 IEEE 17th International Conference on
Intelligent Transportation Systems (ITSC)
October 8-11, 2014. Qingdao, China

Research on Abandoned and Removed Objects Detection Based on Embedded System

Bin Hu, Yiyue Li, Zizhang Chen, Gang Xiong, Fenghua Zhu

Abstract—**This paper presents a real-time detection algorithm which detects abandoned objects in embedded Intelligent Video Surveillance (IVS) System. This algorithm uses two different Gaussian mixture models (GMM) with different learning rates to extract the foreground in order to detect the abandoned objects and output alarms. Experimental results show the algorithm is robust and well-performed in different circumstances.**

I. INTRODUCTION

The demand of security protection is rising with the economic development. Public area such as airport, bank, underground and waiting room can be dangerous. Therefore, real-time video surveillance system is needed. Traditional video surveillance system can only provide evidences after events, but not detect the events in time. The real-time abandoned objects detection can inform relative staffs the potential dangerous of the suspicious objects.

The detection is achieved by three steps: capturing images from network cameras, analyzing the images in the servers and finally returning the output results. As the development of the software and hardware, embedded system shows its advantages: it has real-time detection and largely reduces the calculation needed in the servers.

The current technology can only detect a suspicious object, but not classify it is an abandoned object or an object being removed. In addition, the common algorithms have two disadvantages: the calculation is so heavy that the algorithm must rely on the hardware of servers. The other one is that the performance becomes poor if the environment changes such as lighting condition, weather, vehicles that stop in the detection area. These problems should be solved to reduce the false positive detection results.

II. GMM BACKGROUND

Each pixel is regarded as an independent variable that has distribution $P(X_t)$ on the time axis. Each distribution can be expressed as a mixture of K independent Gaussian distribution [1]-[2], where K is normal between 3 and 5, as in

$$P(X_t)=\sum_{i=1}^{K}\omega_{i,t}\times\eta(X_t\mid\mu_{i,t},\sigma_{i,t}^2) \quad (1)$$

Bin Hu is with the Qingdao Academy of Intelligent Industries, Institute of Automation, Chinese Academy of Sciences. (The corresponding author is Bin Hu: binhu@ia.ac.cn)

Yiyue Li, Zizhang Chen, Gang Xiong, Fenghua Zhu are with the Cloud Computing Center, Chinese Academy of Sciences Songshan Lake, Dongguan, 523808, China.

where $\boldsymbol{\omega_{i,t}}$ is the weight of i[th] distribution and satisfies $\sum_{i=1}^{K}\omega_{i,t}=1$. $\boldsymbol{\mu_{i,t}}$ and $\boldsymbol{\sigma^2{}_{i,t}}$ represent the mean and the variance of the i[th] distribution respectively. $\boldsymbol{\eta(X_t\mid\mu_{i,t},\sigma^2{}_{i,t})}$ is the Gaussian density function and can be expressed as:

$$\eta(X_t\mid\mu_{i,t},\sigma_{i,t}^2)=\frac{1}{(2\pi)^{n/2}\sigma_{i,t}}e^{-\frac{1}{2}(X_t-\mu_{i,t})^T\sigma_{i,t}^{-2}(X_t-\mu_{i,t})} \quad (2)$$

where X_t has n degrees of freedom. For gray-scale image, n =1 and for RGB-scale image, the Gaussian functions are independent. The mean and the variant of the Gaussian functions can be updated by:

$$\mu_t=(1\text{-}\rho)\mu_{t-1}+\rho X_t$$
$$\sigma_t^2=(1\text{-}\rho)\sigma_{t-1}^2+\rho(X_t\text{-}\mu_t)^T(X_t\text{-}\mu_t) \quad (3)$$

where $\rho=\alpha\eta(X_t\mid\mu_{i,t},\sigma^2{}_{i,t})$ and α is learning rate. If a pixel in the current frame does not match any of the Gaussian functions, the Gaussian function which cannot be the background and with the smallest weight will be re-calculated. Also, the weights of all K Guassian functions will be updated. The updated weights should be $\omega_{i,t}=(1\text{-}\alpha)\,\omega_{i,t}+\alpha M_{i,t}$. If the pixel value matches the Guassian functions at time t, $M_{i,t}$ equals 1. Otherwise $M_{i,t}$ equals 0.

A. Updating Bi-model background

The Bi-model background [3]-[4] is formed from GMMs in YUV color-space. YUV color-space is a common method of recording images from videos. The method contains 3 values: brightness value Y and color values U and V. Using YUV color-space can remove shadows to distinguish the foreground and the background more efficiently than that using RGB color-space. Moreover, cameras do not have to transform the image from RGB to YUV color-space, so that cameras work more efficiently.

First, build two GMM backgrounds with different learning rates. They are long-period GMM background and short-period GMM background respectively. The short-period GMM background M_S is built based on the YUV brightness value Y. It detects the binary foreground with small noise at the background. A long-period GMM background M_L is built based on the YUV color values U and V. It detects suspicious stationary objects. It is independent to the brightness value Y, so that the lighting condition and shadows have little effect on the detection.

B. Stationary foreground detection

Two binary foreground images F_L and F_S are obtained from the Bi-model background, by subtracting the GMM background images B_L and B_S and filtering by a threshold value in each frame. In detailed, it is shown as follow:

The long-period GMM background M_L is updated by the color values U and V. Also the binary foreground is filtered by a threshold and recorded as a matrix F_L, which has the same size as the original image.

$$F_L = \begin{cases} 1 & \text{if } |U_{n-1} - U_b| + |V_n - V_b| > T_{UV} \\ 0 & otherwise \end{cases} \tag{4}$$

For the short-period GMM background, a foreground can be calculated from the brightness value Y

$$F_s = \begin{cases} 1 & \text{if } |Y_n - Y_b| > T_Y \\ 0 & otherwise \end{cases} \tag{5}$$

Therefore, the foreground F_L indicates moving targets, abandoned objects, change in lighting effect and noise. The foreground F_S indicates moving targets and noise. The abandoned objects can be detected by comparing the two foregrounds:

$$F = \begin{cases} 1 & \text{if } \ F_L = 1 and F_s = 0 \\ 0 & otherwise \end{cases} \tag{6}$$

An example foreground F is shown in Fig. 1:

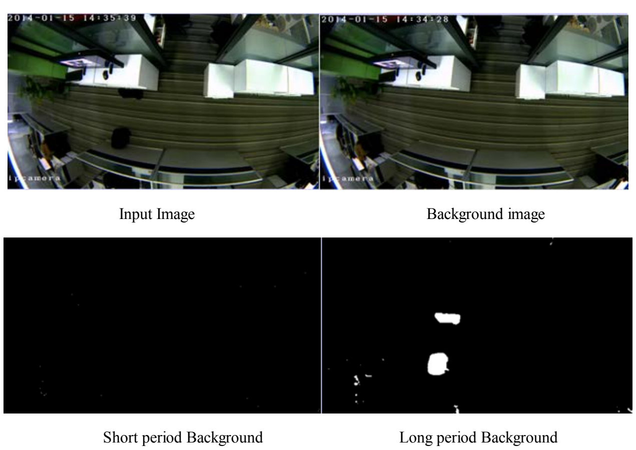

Figure 1: Example foreground

After that, the final foreground F is filtered by median and morphological filters to remove noise and interferences.

As the light may change in the detection area, the foregrounds may contain large area of false positive detection. The number of binary foreground pixels *SumF* is counted to avoid this problem. If $0<SumF<T_1$, the foreground is regarded as a true positive result but no suspicious object appears in the detection area. However, if $SumF>T_2$, the foreground is regarded as a false positive result due to the lighting effect. Then the GMM models are rebuilt using different thresholds. If $T_1<SumF<T_2$, there is a suspicious object.

Once a suspicious object appears in the detection area, a timer starts to count the time $T(x, y)$ that the object stays in the detection area. If the time reaches a threshold Th_{max},y an alarm will be triggered. The size, color and location of the object will be shown as an output result to relative staffs.

$$T(x,y) = \begin{cases} T(x,y)+1 & F(x,y)=1 \\ T(x,y)-k & F(x,y)=0 \\ Th_{max} & T(x,y) > Th_{max} \end{cases} \tag{7}$$

where $F(x,y)$ is the binary foreground, threshold Th_{max} depends on the frame rate and the alarm setting. The whole detection process is shown in Fig. 2.

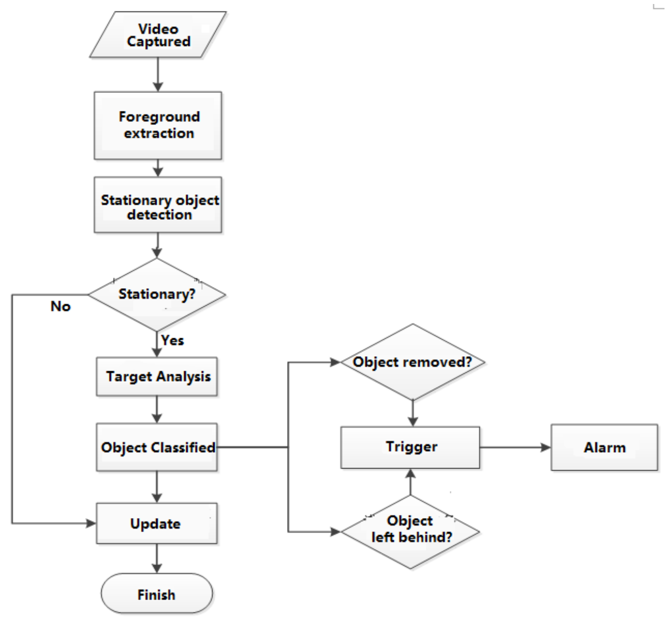

Figure 2: Detection process

III. CLASSIFYING DETECTED OBJECTS

Due to detecting a stationary object, the system needs to confirm whether it is a fake stationary object by detecting the contour of the object. The contours in different frames are accumulated and recorded in a binary matrix $M(i,j)$. The stability score of the contour is then calculated by a mask. The smaller the score is, the less stable the contour is. For example, if the stability score $V_{avg}=sum(M(i,j))\ /\ S<T$, the object is considered as a false positive detection. The system then stops tracking this object. Where S is the number of non-zero pixels in N frames, T is the stability threshold.

If the object is confirmed as a true positive detection, it should be classified into ‘object being removed’ class or ‘abandoned objects’ class [5].

The object is classified by the following steps:

1) The pixels in the original image with binary foreground value equals 1 are filled with color *SetValue*.

2) The filled area $Area_2$ is compared with the original detected area $Area_1$. If $(Area_2\text{-}Area_1)/Area_1$ is larger than a threshold, the object is an object being removed S_i. Otherwise, the object is an abandoned object A_i. Some detection results are shown in Figure 3.

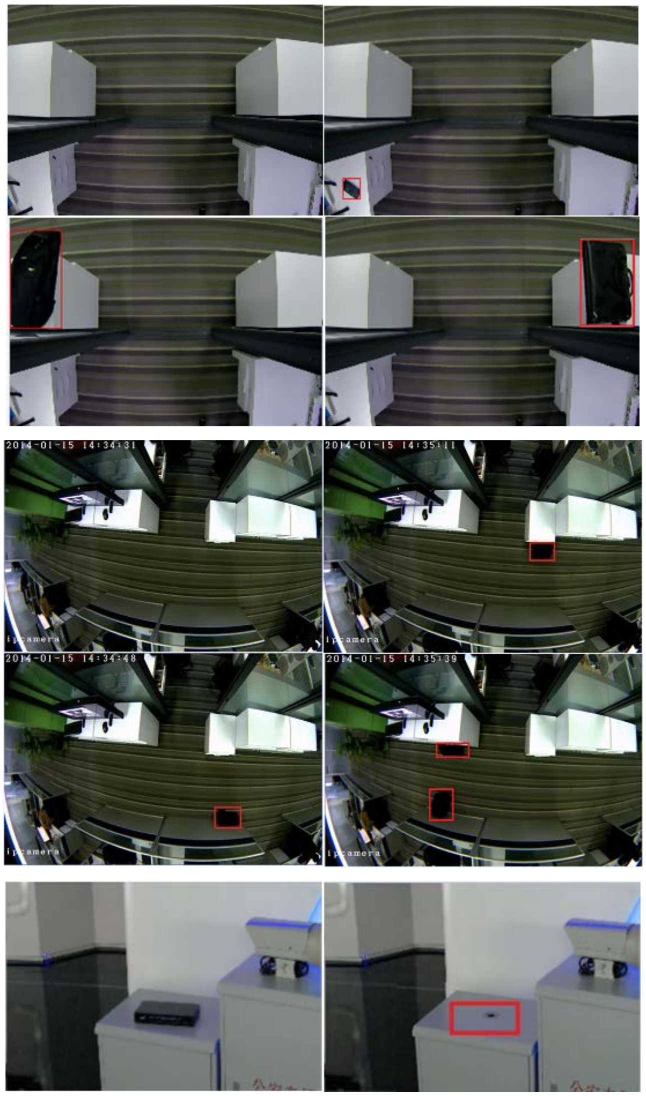

Figure 3: Detection Results

IV. Updating and Matching Target

When an object is detected, it can be an object being moved from one location to another in the detection area. In other words, there is a historical detection of a removed object and a new detection of an abandoned object. The two objects are obviously the same.

Therefore, the new detected object contour should be compared with the historical removed object contour. By matching the shape features and the color histograms [6] which are based on U and V values, Euler distance can be calculated by the formula (8).

$$d^2 = \sum_{i=0}^{n} \left(\frac{A(\mathrm{i})}{S_A} - \frac{S(\mathrm{i})}{S_s}\right)^2 \tag{8}$$

The smaller the Euler distance is, the more similar the two contours are. Then the normalized correlation of the invariant of the matching contours is:

$$D = \frac{\sum_{k=1}^{N} A_k S_k}{[\sum_{k=1}^{N} (A_k)^2]^{1/2} [\sum_{k=1}^{N} (S_k)^2]^{1/2}} \tag{9}$$

The larger the correlation is, the more similar the two contours are. Finally, three rules are applied to the detection:

1. Both areas have color Euler distances that satisfy $d^2 < T_{E1}$ and $D > T_{D1}$
2. Normalized correlation satisfies $d^2 < T_{E1}$ and $D > T_{D2}$
3. Thresholds satisfy $T_{D2} > T_{D1}$ and $T_{E1} < T_{E2}$

If the new detected object satisfies the above three rules, it is regarded as a false positive detection, and the GMM backgrounds are updated.

V. Embedded System Platform

In this paper, all the algorithms are achieved on network cameras. For the hardware, the system uses Haisi Hi3516SOC embedded platform, 3516 high resolution 1080P@30fps video input and coding system, ARM Cortex A9 Processor with dominant frequency 800MHz and hardware process accelerator modulus.

Images are captured on Linux platform and loaded through ISP optimization from DMA (DirectMemoryAccess). Then SOC hardware is used directly to change the RGB image into gray-scale image.

For the software, we use image preprocessing, background modeling and feature extraction algorithm based on OpenCV. OpenCV is a multi-platform visual software package developed by Intel. The detection of abandoned and removed objects algorithm is achieved in the embedded system in the cameras. Therefore, no calculation is needed to analysis the videos in the servers. Also, the cameras can be installed anywhere as many as possible.

VI. Conclusion

This paper proposes a robust and accurate algorithm to detect abandoned objects. Experiments show that the algorithm effectively detects abandoned objects and removed objects. The basic idea is to build GMM backgrounds with different learning rates and extract the foregrounds. Then filters are applied to reduce the false positive rate and the final results are obtained. The system has its unique advantages:

1. The algorithm can be run in embedded system. It provides real-time detection and reduces the calculation needed in the servers.

2. The algorithm only needs to detect stationary objects. So it can largely reduce the false positive percentage.
3. The algorithm classifies an object as 'abandoned objects' or 'removed object'.
4. The algorithm can adapt to the environment in different lighting effect.
5. The sensitivity and the time that the objects stay in the detection area can be adjusted manually.

VII. ACKNOWLEDGMENT

This work is supported in part by Chinese MoT's S&T project (2012-364-221-108; 2012-364-221-101). Chinese MIIT's IoT development special fund project: Real time image recognition technology and application system for social security. Guangdong's S&T project (2012B091100213; 2012B090400004). Dongguan's Innovation Talents Project (Gang Xiong).

VIII. REFERENCES

[1]. C Stauffer, W Grimson. Adaptive background mixture mdels for real-time tracking [C], Proc IEEE Conference on Compute Vision and Pattern Recognition, 1999.2: 246-252

[2]. Li XL, Zhang C, Zhang D. Abandoned Objects Detection Using Double Illumination Ivariant Foreground Masks:20th International Conference on Pattern Recognition. Istanbul,2010.ICPR,2010.436-439

[3]. F Porikli, Y Ivanov, T Haga, Robust Abandoned Object Detection Using Dual Foregrounds [J], Journal o Advances in Signal Processing,2008

[4]. Fan Jun-Jun, Zhan Yin-Wei, An Abandoned Object Detection Method Based on Dual Background Models[J], Computer System Application, 2012, 21(8):201-205

[5]. Wang Wei-jia, Liu Hui, Sha Li, Real time detection and classification algorithm for abandoned and stolen objects[J]. Computer Applicaions, 2007,27(10): 2591-2594

[6]. Li Kun-lun, Sun Da-yue, Zhao Dong-sheng, Color Histogram Matching Algorithm in video Detection[J], Computer and Information Technology, 2008, 16(5):6-10