Robust Left Object Detection and Verification in Video Surveillance

Yijun Xiao, Abdul Farooq, Melvyn Smith
University of the West of England
{Yijun.Xiao, Abdul2.Farooq,
Melvyn.Smith}@uwe.ac.uk

Daniel Wright, Glynn Wright
Aralia System Ltd.
13 North Parade, Horsham, United Kingdom
{Daniel.Wright, Glynn.Wright}aralia.co.uk

Abstract

Left objects pose a real threat to security in public areas such as railway stations and airports. Detection of these objects therefore forms an important part in any intelligent video surveillance system that is deployed at such locations. Successful left object detection algorithms must operate in real time and produce sufficient detection accuracy with low false positive rates. However in reality, the requirement of both speed and performance is not often achieved due to the huge variation in image appearance caused by illumination, scene, and foreground objects (both dynamic and static). This paper tackles the challenge using a background subtraction scheme coupled with three other techniques. Short-term frame averaging is used to reduce the effect of moving objects such as pedestrians and vehicles. Statistical image background modelling is applied to enhance the visual contrast between the object and the background. Pixel colour modelling is employed to verify the results of left object segmentation. All three techniques are computationally lightweight and thus enable the left object detection to operate in real time.

1. Introduction

The advances in digital imaging and computing technology have made intelligent video surveillance possible in the last two decades. What lies in the core of a surveillance system is a video analytic software component that can identify events and objects of interest in the video data. For example, a security system deployed in a railway station needs to detect various objects (bags, cases, etc.) being left on platforms since they may present a high security risk to the public. However, due to the large variation in object appearance, scene background, camera setup and lighting, etc., it is extremely challenging to come up with a robust solution for left object detection, especially when it has to be done in real time.

This paper discusses a real-time left object detection scheme based on the background subtraction principle [1-2]. Background subtraction methods are known to be fast but prone to imaging noise and illumination changes, and therefore are likely to generate unnecessary false positive detection results. To remedy this drawback, this paper proposes three techniques to improve detection performance. Short-term frame averaging is used to reduce the effect of moving objects in the scene such as pedestrians and vehicles. Statistical background modelling is applied to enhance the visual contrast between the left objects and the image background. Pixel colour modelling is employed to verify the results of left object segmentation after the background subtraction. All three techniques are computationally lightweight, and thus can be implemented to operate in real time. When combined with the other parts of the background subtraction scheme, a much improved performance of left object detection is achieved.

2. Left Object Detection

A typical security system deployed in a railway station (or airport) usually consists of numerous cameras monitoring different parts of the station. The cameras are connected to a control room where the data is stored and processed. The video analytic software operates in the control room to extract useful information in the video frames and detect events and objects of interest according to the users' criteria. For potential threats that can present high security risks such as left objects, the detection has to be fast and accurate to make it of practical use.

2.1 Background subtraction

Object detection has been studied extensively in computer vision and good progress has been made towards object detection in complex scenes [3-4]. However methods that demand excessive computational resources are impractical to apply in a real surveillance system since the amount of data acquired from the cameras to be analysed can be very large. We chose background subtraction methods since they are computationally simple and can deliver real-time performance. The background subtraction principle can be expressed as follows:

$$I_d = I_f - I_b \tag{1}$$

It takes the difference image between the foreground image I_f(the image that contains the objects of interest) and the background image I_b (the image with scene background only). In an ideal world (noise and shadow free, constant lighting, etc.), the difference image I_d will contain the foreground objects only. However in reality, factors such as image noise, lighting changes, shadows, etc., constantly render the difference image I_d far from ideal just containing the clean foreground objects. False positive detection occurs when pixels belonging to the background are classified as foreground objects. There are two ways of reducing false positive detection rates. One way is to post-process the difference image I_d to make it 'cleaner'. Popular post-processing techniques include morphological operations to eliminate, for example, a small bulk of isolated pixels. Another way is to improve the quality of the background image I_b in order to make it as close to the foreground image I_f as possible except for the foreground pixels in I_f. This technique is known as background modelling. We use both methodologies in this work to improve detection performance.

The left object detection process is illustrated in Fig. 1. Frames of images in a video stream are averaged within a short period of time. The background of the scene is reconstructed from the short-term averaged image through a background model. The short-term averaged image and the reconstructed background image are subtracted to obtain a foreground image. The objects are segmented from the foreground image. Post-processing is applied to clean up the segmentation result and then the result is verified by a pixel colour model.

2.2 Short term averaging

Short-term averaging is applied to eliminate moving objects in video frames such as pedestrians, vehicles, escalators etc., since the objective is to detect left objects. Left objects are not permanent fixtures in the scene. Instead they stay static from a few minutes up to a few hours depending on how quickly they are detected and removed. Since the left objects are not moving, their appearance will not change in the average image. In contrast, moving objects will largely disappear since the averaging operation smoothes over the video data in the temporal domain:

$$I_v(t) = \frac{1}{f\tau}\sum_{i=0}^{f\tau-1} I(t+i/f) \qquad (2)$$

f in Eq(2) denotes the frame rate of the video and τ represents the duration of the short term and controls the smoothing effect of the averaging process. The bigger τ is, the more smoothed the average image $I_v(t)$ is and the less visible the moving objects are in the averaged image. However, a bigger τ will increase the reaction time of the left object detection. We normally choose τ to be 60-120 seconds.

2.3 Background modelling and reconstruction

The scene background is recovered from the short-term average image. This is done through a statistical background model which is learnt from a pool of training images that contain the scene background only. The training images are selected from the video frames recorded at different times by the same camera. The training images are used to build a representative image space that captures the main characteristics of the scene background appearance. First we calculate the mean of the training images $\{I_1, I_2, \ldots, I_n\}$

$$\tilde{I} = \frac{1}{n}\sum_{k=1}^{n} I_k \qquad (3)$$

Then we compute the covariance matrix:

$$C = XX^T, X = [I_1 - \tilde{I}, I_2 - \tilde{I}, \ldots, I_n - \tilde{I}] \qquad (4)$$

Eigenvectors of C can be obtained by solving the following equation:

$$CV = \gamma V \qquad (5)$$

The eigenvectors are orthogonal and each eigenvector captures certain characteristics of the scene background. The mean image and the background eigenvectors form a background model that is used to recover the scene background from an image containing foreground objects. We project the difference vector between the foreground image I_f and the background mean image $\tilde{I}$ to the background eigenvectors $\{V_1, V_2, \ldots, V_m\}$, and then reconstruct the image of the background from the projected vector:

$$I_b = \tilde{I} + \sum_{j=1}^{m} < I_f - \tilde{I}, V_j > V_j \qquad (6)$$

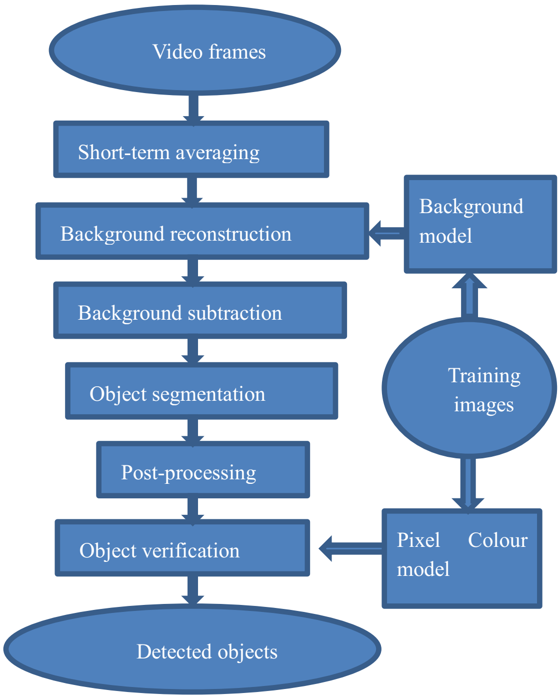

Figure 1. Left object detection diagram.

2.4 Object segmentation and verification

Image I_b from Eq(6) is reconstructed using the background characteristic eigenvectors and therefore contains no foreground objects. The difference image I_d between I_b and I_f (see Eq(1)) is then used to identify the pixels belonging to the foreground. For speed reasons, we simply classify a pixel to foreground if the difference value of the pixel is greater than a threshold. The threshold is calculated from the histogram of the difference image I_d using Ostu's method [5]. The dilation and erosion morphological operations are applied on the background/foreground label map to eliminate isolated pixels. Only connected regions in the resultant label map with more than 50 pixels are selected as candidates for foreground objects. This reduces the chance of false positive detection caused by image noise. A further verification is applied to identify the true foreground objects from the candidates.

The final verification is performed in the normalized RGB colour space. Most surveillance cameras nowadays are able to produce colour images. A full colour image carries both brightness and colour information. In the process for left object segmentation described above, only the brightness information is used (images I_b, I_f, I_v, I are all assumed to be gray level). The colour information is now used independently to check if an object segmented in the brightness channel is a real left object. The raw *RGB* values of a pixel are normalized as follows:

$$Y = (R + G + B)/3, r = R/3Y, g = G/3Y \qquad (7)$$

where Y represents the average brightness of the *RGB* channels and r, g carry colour information about red and green chromaticity of the pixel respectively. r, g are normalized so that they are not affected by the pixel brightness.

For each pixel of a surveillance camera, we establish a colour distribution model using the training images collected for background modelling (see Fig. 1). The colour distribution model is characterized by the mean and covariance of the rg values of the pixel in the training images, which are calculated as follows:

$$\tilde{r} = \frac{1}{n}\sum_{k=1}^{n} r_k \, , \tilde{g} = \frac{1}{n}\sum_{k=1}^{n} g_k \qquad (8)$$

$$C_{rg} = \left[X_r, X_g\right]^T \left[X_r, X_g\right]$$

$$X_r = [r_1 - \tilde{r}, r_2 - \tilde{r}, \dots, r_n - \tilde{r}]^T$$

$$X_g = [g_1 - \tilde{g}, g_2 - \tilde{g}, \dots, g_n - \tilde{g}]^T \qquad (9)$$

Once a left object has been identified in a foreground image, pixels assigned to this object are checked to ensure the object is valid. The object is defined as valid if more than 50% of its pixels have different colours from the scene background. A pixel is considered to be different from background in colour if the Mahalanobis distance between its rg values and the rg mean of the same pixel in the training images is greater than 1. Mahalanobis distance is a L2 metric normalized by covariance:

$$d(r,g) = \sqrt{[r - \tilde{r} \quad g - \tilde{g}] C_{rg}^{-1} [\begin{matrix} r - \tilde{r} \\ g - \tilde{g} \end{matrix}]} \qquad (10)$$

By covariance normalization, the data are transformed to a ‘whitened’ space and the other statistics measures such as significance can then be applied.

2.5 Complexity analysis

Left object detection must be done in real time to prompt an immediate alarm in the security system. The techniques and algorithms in the diagram of Fig. 1 must be computationally efficient to make the detection fast. Let us assume the image resolution is $w \times h$, the computationally complexity of each step for left object detection in the diagram of Fig. 1 is listed in Table 1.

Table 1.Computational Complexity Analysis.

method	Complexity
Averaging	O($f \times \tau \times w \times h$)
Reconstruction	O($m \times w \times h$)
Subtraction	O($w \times h$)
Segmentation	O($w \times h$)
Post-processing	O($w \times h$)
Verification	O($w \times h$)
Colour modelling	O($n \times w \times h$)
Background modelling	O($n \times w \times h + n^2$)

It can be seen that all the methods in Table 1 are computationally linear except the background modeling. Background modeling requires O(n^2) extra computation, where n is the number of training images. In practice, we choose around 100 training images for each camera, which we found allows fast background training and yet still achieving an acceptable quality of background reconstruction for the left object detection. In addition, the background and colour model training regimes are not required to be undertaken in real time. In fact, once the models are learnt, they are valid until circumstances change such as cameras being relocated and/or lightings re-adjusted. Therefore we conclude that the diagram in Fig. 1 can be implemented to meet the real-time speed requirement.

Among the linear techniques listed in Table 1, the most time consuming method is frame averaging. The time needed for frame averaging is linearly proportioned to the multiplication of frame rate, averaging duration and image resolution. For a further speed gain, frame rate and image resolution can be compromised. In practice, we only use key frames in mpeg video streams for left object detection, which effectively reduces the frame rate and makes the left object scheme practical to implement using only a few computers for handling hundreds of cameras.

3. Validation

We validated the techniques described in Section 2 using the following experiments. Fig. 2 illustrates the effect of averaging video frames. The image on the left shows a frame in which an inspector is dropping a bag on the platform. The video frames in the next 60 seconds are averaged to obtain the image shown on the right of Fig.2, where it can be seen clearly that the moving object (inspector) is gone and the static object (left bag) remains unchanged.

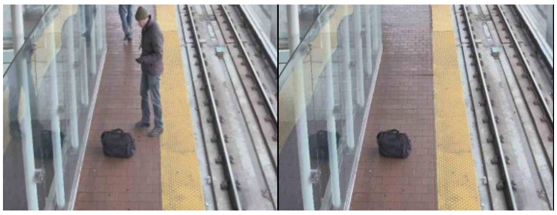

Figure 2. Averaging video frames to remove moving objects.

The left object segmentation algorithm described in Section 2.2 is applied on the intensity channel (Y from Eq(7)) of the averaged image. The reconstructed background image is displayed on the left of Fig.3 and the segmented result on the right. It can be seen that the scene background has been reconstructed perfectly and the segmentation result is tidy and clean.

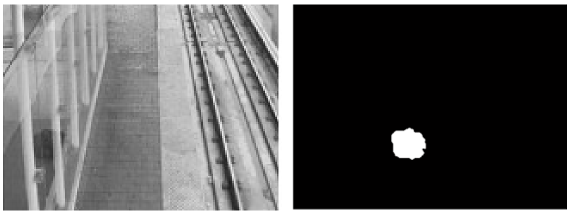

Figure 3. Background reconstruction and object segmentation.

Fig. 4 shows an example of a less optimal background reconstruction. The object in the image on the left casts strong shadows on the platform which is poorly illuminated (only intensity channel is displayed). The reconstructed background image on the right looks largely consistent with the scene background appearance on the left, though some small details are missing. The fidelity of the reconstructed background is also reflected in the subtracted image shown on the left of Fig. 5.

The Mahalanobis distance map between the rg chromaticity values of the image pixels in Fig. 4 and their colour distribution models are illustrated on the right of Fig.5. The object pixels have the largest distances and stand out from the scene background pixels. The same can be observed in the background subtracted image (left of Fig.5). Since the operations of the background subtraction and the colour verification are performed independently in the intensity channel and chromaticity channels, the object segmentation is verified. Note that the polygonal effect in Fig. 5 is due to the region of interest operation. Only pixels of the platform within the polygon are considered in the background reconstruction and colour verification.

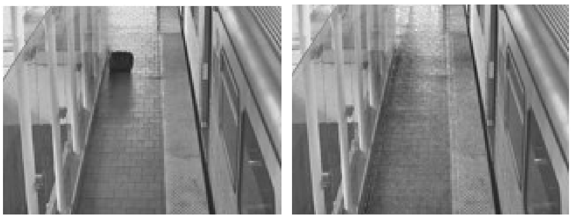

Figure 4. .Reconstruction of scene background in insufficient illumination condition.

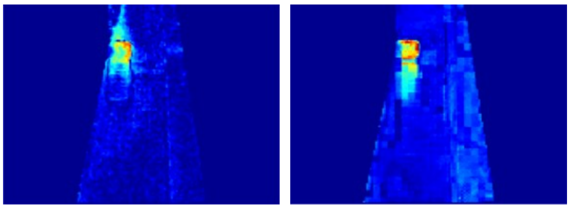

Figure 5. Background reconstruction image and Mahalanobis distance map.

4. Application

The left object detection scheme has been applied to a security system used in a light rail system containing 23 stations in US. Each station has around 30 surveillance cameras, which are networked to 2 or 3 processing computers to perform video analytics, data management and other video surveillance tasks. The processing computers are then connected to a central server which manages the data from all 23 stations. The configuration of the hardware is illustrated in Fig. 6.

Users can monitor the cameras in the control room of each station or in a central control room. Detected left objects will trigger alarms and users can look into the video recording and check to see what is going on. The detected objects will be measured by their photometric and geometric properties such as colour, size, etc., and then classified and indexed. Users can carry out a retrospective search to identify a particular object in the indexed database and confirm the triggered security alarms.

The cameras deployed in this project are SONY IP cameras of model DH-180, DH-70, RZ-25, ER-580. Around 50% of the cameras are deployed underground where the scenes are only illuminated by artificial lights. The other half are placed overground where the main illumination during the day is natural (direct or ambient) light. The large variation in illumination is addressed by the background modeling method. Technicians are required to collect representative training images for each camera separately. Training images can be re-adjusted to re-configure the system or to improve the video analytics performance for each camera. Protocols are established to perform the configuration. Feedbacks from the end users are very positive.

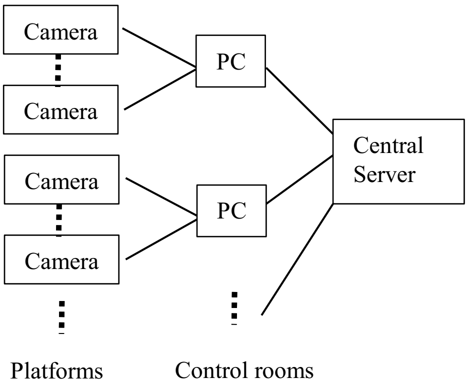

Figure 6. Configuration of a railway surveillance system.

5. Conclusion

We described a solution to left object detection for video surveillance. The real time speed requirement and constraints on data bandwidth and computational power in a real security system hold back the use of many advanced object detection algorithms that demand excessive computing resources. The scheme discussed in this paper delivers a real time performance and has a mechanism to handle large scene variations. It will appeal to commercial applications for a number of reasons:

(1) The methods are computationally simple and can be implemented efficiently.

(2) The solution is quite robust against large illumination changes.

(3) The performance can be controlled in each step of the process.

(4) The methods are scalable and can be integrated into a background subtraction framework for any video analytics task.

References

[1] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts and shadows in video streams”, IEEE Trans. on Patt. Anal. and Machine Intell., vol. 25, no. 10, pp. 1337-1342, 2003.

[2] B.P.L. Lo and S.A. Velastin, “Automatic congestion detection system for underground platforms,” Proc. of 2001 Int. Symp. on Intell. Multimedia, Video and Speech Processing, pp. 158-161, 2000.

[3] B.Leibe, E. Seemann, and B. Schiele. “Pedestrian detection in crowded scenes” IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pages 1:878-885, 2005

[4] W. Zhang, Q.M.J. Wu, G. Wang, and H. Yin. “An adaptive computational model for salient object detection”, IEEE Trans. on Multimedia, Vol. 12 No.4 , pp.300–316, 2010

[5] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66