Background Subtraction Techniques

Alan M. McIvor

Reveal Ltd
PO Box 128-221, Remuera, Auckland, New Zealand
alan.mcivor@reveal.co.nz

Abstract

Background subtraction is a commonly used class of techniques for segmenting out objects of interest in a scene for applications such as surveillance. This paper surveys a representative sample of the published techiques for background subtraction, and analyses them with respect to three important attributes: foreground detection; background maintenance; and postprocessing.

Keywords: *Background subtraction, surveillance, segmentation*

1 Introduction

Background subtraction is a commonly used class of techniques for segmenting out objects of interest in a scene for applications such as surveillance. It involves comparing an observed image with an estimate of the image if it contained no objects of interest. The areas of the image plane where there is a significant difference between the observed and estimated images indicate the location of the objects of interest. The name "background subtraction" comes from the simple technique of subtracting the observed image from the estimated image and thresholding the result to generate the objects of interest.

This paper surveys several techniques which are representative of this class, and compares three important attributes of them: how the object areas are distinguished from the background; how the background is maintained over time; and, how the segmented object areas are postprocessed to reject false positives, etc.

Several algorithms were implemented to evaluate their relative performance under a variety of different operating conditions. From this, some conclusions are drawn about what features are important in an algorithm of this class.

2 Heikkila and Olli

In [8], a pixel is marked as foreground if

$$|\mathbf{I}_t - \mathbf{B}_t| > \tau \tag{1}$$

where τ is a "predefined" threshold. The thresholding is followed by closing with a 3×3 kernel and the discarding of small regions.

The background update is

$$\mathbf{B}_{t+1} = \alpha \mathbf{I}_t + (1-\alpha)\mathbf{B}_t \tag{2}$$

where α is kept small to prevent artifical "tails" forming behind moving objects.

Two background corrections are applied:

1. If a pixel is marked as foreground for more than m of the last M frames, then the background is updated as $\mathbf{B}_{t+1} = \mathbf{I}_t$. This correction is designed to compensate for sudden illumination changes and the appearance of static new objects.
2. If a pixel changes state from foreground to background frequently, it is masked out from inclusion in the foreground. This is designed to compensate for flucuating illumination, such as swinging branches.

3 Adaptive Mixture of Gaussians

Each pixel is modeled separately [3, 9, 10] by a mixture of K Gaussians

$$P(\mathbf{I}_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(\mathbf{I}_t; \mu_{i,t}, \Sigma_{i,t}) \tag{3}$$

where $K = 4$ in [9] and $K = 3 \ldots 5$ in [10]. In [3, 10], it is assumed that $\Sigma_{i,t} = \sigma_{i,t}^2 \mathbf{I}$.

The background is updated, before the foreground is detected, as follows:

1. If $\mathbf{I}_t$ matches component i, i.e., $\mathbf{I}_t$ is within λ standard deviations of $\mu_{i,t}$ (where λ is 2 in [3] and 2.5 in in [9, 10]), then the ith component is updated as follows:

$$\begin{aligned} \omega_{i,t} &= \omega_{i,t-1} && (4)\\ \mu_{i,t} &= (1-\rho)\mu_{i,t-1} + \rho \mathbf{I}_t && (5)\\ \sigma_{i,t}^2 &= (1-\rho)\sigma_{i,t-1}^2 + \rho(\mathbf{I}_t - \mu_{i,t})^{\mathsf{T}}(\mathbf{I}_t - \mu_{i,t}) && (6) \end{aligned}$$

   where $\rho = \alpha \Pr(\mathbf{I}_t | \mu_{i,t-1}, \Sigma_{i,t-1})$.

2. Components which $\mathbf{I}_t$ don't match are updated by

$$\begin{aligned} \omega_{i,t} &= (1-\alpha)\omega_{i,t-1} && (7)\\ \mu_{i,t} &= \mu_{i,t-1} && (8)\\ \sigma_{i,t}^2 &= \sigma_{i,t-1}^2 && (9)\\ & && (10) \end{aligned}$$

3. If $\mathbf{I}_t$ does not match any component, then the least likely component is replaced with a new one which has $\mu_{i,t} = \mathbf{I}_t$, $\Sigma_{i,t}$ large, and $\omega_{i,t}$ low.

After the updates, the weights $\omega_{i,t}$ are renormalised.

The foreground is detected as follows. All components in the mixture are sorted into the order of decreasing $\omega_{i,t}/\|\Sigma_{i,t}\|$. So higher importance gets placed on components with the most evidence and lowest variance, which are assumed to be the background. Let

$$B = \text{argmin}_b \left(\frac{\sum_{i=1}^{b} \omega_{i,t}}{\sum_{i=1}^{K} \omega_{i,t}} > T \right) \tag{11}$$

for some threshold T. Then components $1 \ldots B$ are assumed to be background. So if $\mathbf{I}_t$ does not match one of these components, the pixel is marked as foreground. Foreground pixels are then segmented into regions using connected component labelling. Detected regions are represented by their centroid [11].

4 Pfinder

Pfinder [13] uses a simple scheme, where background pixels are modeled by a single value, updated by

$$\mathbf{B}_t = (1 - \alpha)\mathbf{B}_{t-1} + \alpha\mathbf{I}_t \tag{12}$$

and foreground pixels are explicitly modeled by a mean and covariance, which are updated recursively. It requires an empty scene at start-up.

5 $\mathbf{W}^4$

In [5, 6, 7], a pixel is marked as foreground if

$$|\mathbf{M} - \mathbf{I}_t| > \mathbf{D} \quad \text{or} \quad |\mathbf{N} - \mathbf{I}_t| > \mathbf{D} \tag{13}$$

where the (per pixel) parameters $\mathbf{M}$, $\mathbf{N}$, and $\mathbf{D}$ represent the minimum, maximum, and largest interframe absolute difference observable in the background scene. These parameters are initially estimated from the first few seconds of video and are periodically updated for those parts of the scene not containing foreground objects.

The resulting foreground "image" is eroded to eliminate 1-pixel thick noise, then connected component labelled and small regions rejected. Finally, the remaining regions are dilated and then eroded.

6 LOTS

In [1], three background models are simultaneously kept, a primary, a secondary, and an old background. They are updated as follows:

1. The primary background is updated as

$$\mathbf{B}_{t+1} = \alpha\mathbf{I}_t + (1 - \alpha)\mathbf{B}_t \tag{14}$$

if the pixel is not marked as foreground, and is updated as

$$\mathbf{B}_{t+1} = \beta\mathbf{I}_t + (1 - \beta)\mathbf{B}_t \tag{15}$$

if the pixel is marked as foreground. In the above, α was selected from within the range $[0.0000610351 \ldots 0.25]$, with the default value $\alpha = 0.0078125$, and $\beta = 0.25\alpha$.

2. The secondary background is updated as

$$\mathbf{B}_{t+1} = \alpha \mathbf{I}_t + (1-\alpha)\mathbf{B}_t \tag{16}$$

at pixels where the incoming image is not significantly different from the current value of the secondary background, where α is as for the primary background. At pixels where there is a significant difference, the secondary background is updated by

$$\mathbf{B}_{t+1} = \mathbf{I}_t \tag{17}$$

3. The old background is a copy of the incoming image from 9000 to 18000 frames ago.

Foreground detection is based on adaptive thresholding with hystersis, with spatially varying thresholds. Several corrections are applied:

1. Small foreground regions are rejected.
2. The number of pixels above threshold in the current frame is compared to the number in the previous frame. A significant change is interpreted as a rapid lighting change. In response the global threshold is temporarily increased.
3. The pixel values in each foreground region are compared to those in the corresponding parts of the primary and secondary backgrounds, after scaling to match the mean intensity. These eliminate artifacts due to local lighting changes and stationary foreground objects, respectively.

7 Halevy

In [4], the background is updated by

$$\mathbf{B}_{t+1} = \alpha S(\mathbf{I}_t) + (1-\alpha)\mathbf{B}_t \tag{18}$$

at all pixels, where $S(\mathbf{I}_t)$ is a smoothed version of $\mathbf{I}_t$. Foreground pixels are identified by tracking the maxima of $S(\mathbf{I}_t - \mathbf{B}_t)$, as opposed to thresholding. They use $\alpha = [0.3 \ldots 0.5]$ and rely on the streaking effect to help in determining correspondence between frames.

They also note that $(1-\alpha)^t < 0.1$ gives an indication of the number of frames t needed for the background to settle down after initialisation.

8 Cutler

In [2], colour images are used because it is claimed to give better segmentation than monochrome, especially in low contrast areas, such as objects in dark shadows.

The background estimate is defined to be the temporal median of the last N frames, with typical values of N ranging from 50 to 200.

Pixels are marked as foreground if

$$\sum_{C \in R,G,B} |I_t(C) - B_t(C)| > K\sigma \tag{19}$$

where σ is an offline generated estimate of the noise standard deviation, and K is an apriori selected constant (typically 10).

This method also uses template matching to help in selecting candidate matches.

9 Wallflower

In [12], two auto-regressive background models are used:

$$\mathbf{B}_t = -\sum_{k=1}^{p} a_k \mathbf{B}_{t-k} \tag{20}$$

$$\hat{\mathbf{I}}_t = -\sum_{k=1}^{p} a_k \mathbf{I}_{t-k} \tag{21}$$

along with a background threshold

$$\mathcal{E}(e_t^2) = \mathcal{E}(\mathbf{B}_t^2) + \sum_{k=1}^{p} a_k \mathcal{E}(\mathbf{B}_t \mathbf{B}_{t-k}) \tag{22}$$

$$\tau = 4\sqrt{\mathcal{E}(e_t^2)} \tag{23}$$

Pixels are marked as background if

$$|\mathbf{I}_t - \mathbf{B}_t| < \tau \quad \text{and} \quad |\mathbf{I}_t - \hat{\mathbf{I}}_t| < \tau \tag{24}$$

The coefficients a_k are updated each frame time from the sample covariances of the observed background values. In the implementation, the last 50 values are used to estimate 30 parameters.

If more than 70% of the image is classified as foreground, the model is abandoned and replaced with a "back-up" model.

10 Discussion

Many other algorithms, which have not been discussed here, assume that the background does not vary and hence can be captured apriori. This limits their usefulness in most practical applications. Very few of the papers describe their algorithms in sufficient detail to be able to easily reimplement them.

A significant number of the described algorithms use a simple IIR filter applied to each pixel independently to update the background, e.g., (2), and use thresholding to classify pixels into foreground/background. This is followed by some postprocessing to correct classification failures.

In [4], it was noted that the performance of the method in Section 7 was found to degrade if more than one secondary background was used. It was postulated that this is because it introduces a greater range of values that a pixel can take on without being marked as foreground. However, the adaptive mixture of Gaussians approach operates effectively with even more component models. From this it can be seen that using more models is beneficial only if by adding them the range (e.g., variance) of the individual components gets reduced such that the nett range of background values actually decreases.

The Wallflower algorithm requires the storage of over 130 images, many of which are float valued. It requires significant statistical analysis per pixel per frame to adapt the coefficients.

References

[1] T. E. Boult, R. Micheals, X. Gao, P. Lewis, C. Power, W. Yin and A. Erkan: Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets in: *Second IEEE Workshop on Visual Surveillance* Fort Collins, Colorado (Jun. 1999) pp. 48–55.

[2] R. Cutler and L. Davis: View-based detection and analysis of periodic motion in: *International Conference on Pattern Recognition* Brisbane, Australia (Aug. 1998) pp. 495–500.

[3] W. E. L. Grimson, C. Stauffer, R. Romano and L. Lee: Using adaptive tracking to classify and monitor activities in a site in: *Computer Vision and Pattern Recognition* Santa Barbara, California (Jun. 1998) pp. 1–8.

[4] G. Halevy and D. Weinshall: Motion of disturbances: detection and tracking of multi-body non-rigid motion *Machine Vision and Applications* **11** (1999) 122–137.

[5] I. Haritaoglu, R. Cutler, D. Harwood and L. S. Davis: Backpack: Detection of people carrying objects using silhouettes in: *International Conference on Computer Vision* (1999) pp. 102–107.

[6] I. Haritaoglu, D. Harwood and L. S. Davis: W^4: Who? when? where? what? a real time system for detecting and tracking people in: *Third Face and Gesture Recognition Conference* (Apr. 1998) pp. 222–227.

[7] I. Haritaoglu, D. Harwood and L. S. Davis: Hydra: Multiple people detection and tracking using silhouettes in: *Second IEEE Workshop on Visual Surveillance* Fort Collins, Colorado (Jun. 1999) pp. 6–13.

[8] J. Heikkila and O. Silven: A real-time system for monitoring of cyclists and pedestrians in: *Second IEEE Workshop on Visual Surveillance* Fort Collins, Colorado (Jun. 1999) pp. 74–81.

[9] Y. Ivanov, C. Stauffer, A. Bobick and W. E. L. Grimson: Video surveillance of interactions in: *Second IEEE Workshop on Visual Surveillance* Fort Collins, Colorado (Jun. 1999) pp. 82–90.

[10] C. Stauffer and W. E. L. Grimson: Adaptive background mixture models for real-time tracking in: *Computer Vision and Pattern Recognition* Fort Collins, Colorado (Jun. 1999) pp. 246–252.

[11] G. P. Stein: Tracking from multiple view points: Self-calibration of space and time in: *Computer Vision and Pattern Recognition* Fort Collins, Colorado (Jun. 1999) pp. 521–527.

[12] K. Toyama, J. Krumm, B. Brumitt and B. Meyers: Wallflower: Principles and practice of background maintenance in: *International Conference on Computer Vision* (1999) pp. 255–261.

[13] C. Wren, A. Azabayejani, T. Darrell and A. Pentland: Pfinder: Real-time tracking of the human body *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19** (1997) 780–785.