An efficient foreground extraction algorithm with robust shadow removal for video surveillance

Bangjun Lei
Broadband Applications Research Centre, BT Research and Venturing

ABSTRACT

In this paper, for online video surveillance based on a stationary camera, which is the basic building block of more advanced surveillance systems, we propose an efficient and practical background learning and subtraction algorithm to extract the foreground. We further introduce a novel two-branch shadow removal method to remove almost all shadows present while preserving the integrity of each foreground object. We will show that this foreground extraction algorithm with shadow removal runs very fast on a normal PC. By embedding it into an integrated video surveillance system, we can further prove that it provides sufficient support to the robust object tracking.

1. INTRODUCTION

Intelligent video surveillance investigates the use of video analysis and object characterisation techniques for extracting meaningful object information (spatial, temporal and appearance features) from the scene to serve various surveillance purposes (e.g. traffic control, crime monitoring, forensic evidence searching etc.).

Background subtraction has been a popular technique that uses a reference background model to detect changes in the monitored scene. Those changes are most likely due to the interesting moving foreground objects. The features used for building this model can be pixel intensity, colour, edge intensity, depth map or certain mixture of them. Several frequently adopted background models are: (1) A single reference image plus certain global or dynamic thresholds; (2) A Mixture of Gaussian distribution (MoG) for modelling each pixel of the scene [1]; (3) Non-parametric approaches such as the kernel density estimation technique (KDE) [2].

Following this background-subtraction approach, the operations of a video surveillance system can be divided into three inter-related phases including (1) background learning, (2) foreground object extraction and tracking, and (3) high-level applications such as behaviour analysis. In addition, to eliminate the camera shaking effect, a pre-processing step such as video stabilisation may be necessary. The object extraction and tracking phase further involves two processing steps, namely *foreground extraction* and *object tracking*. Optionally, an object-classification step can be performed to decide on the type (human, vehicle etc.) of each object (area) before or after the object tracking step.

The main focus of this paper is to develop a real-time foreground extraction method for robustly segmenting objects out of practical video sequences. For this purpose, efficient background subtraction and robust *shadow detection* play vital roles. Although more advanced techniques such as KDE may deliver better quality of object segmentation, they are in general computationally expensive to be applied for real-time systems. Instead of single reference image, multi-layered background modelling (e.g. MoG) is necessary for accommodating the possible scene changes (due to either structural rearrangement or lighting condition changes). Subsequently a proper choice of the number of layers and an appropriate way to learn and to update the model are important. On the other hand, a *shadow* is a region that becomes darker compared with its true intensity due to the blocking of light by other objects (*cast shadow*) or by the object itself where this region resides (*self-shadow*). While cast-shadowed background may be wrongly detected as foreground by a background subtraction technique (*false positive*), self-shadowed object regions may be removed by the shadow-removal process (*false negative*). Therefore, a too-weak shadow-removal scheme (assuming shadows are highly distinguishable and of small amount) will remove only a small portion of shadows, thus producing too distorted foreground shape. And, a too-strong shadow-removal scheme (assuming shadows are everywhere) will likely to treat certain foreground regions (e.g. self-shadows) also as shadows, hence fragmenting the shape of a true object. A proper balance is therefore needed.

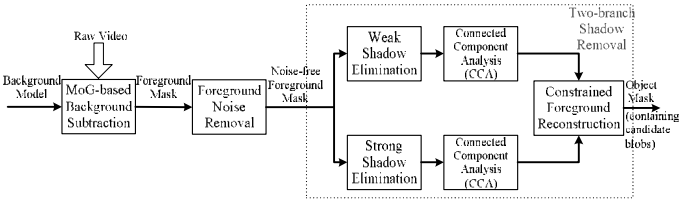


Figure 1: The work flow of foreground extraction task.

Figure 1 shows the work flow of our proposed foreground extraction system, which contains mainly three modules: background subtraction, foreground noise removal, and shadow removal. The purpose is to obtain a true object mask free from shadow, noise and spurious foreground.

According to this work flow, the remainder of this paper is organised as follows. The implementation of our background subtraction method and the three noise removal steps is detailed in Section 2. The two-branch shadow removal idea is introduced subsequently in Section 3. Experimental studies are carried out in Section 4. Because we are mainly interested in object-level applications, we report the performance of our algorithm also at this level. And finally, conclusions are drawn in Section 5.

2. BACKGROUND SUBTRACTION WITH NOISE REMOVAL

In current system, the MoG model is chosen, albeit in a simplified form of the original version discussed in [1].

In MoG, each pixel is modelled independently by a fixed number of Gaussians (G_1, G_2, ..., G_K). At any time t, the feature value of an arbitrary pixel p is noted as $\mathbf{I}_t \in \Re^c$ (c=1 for grey-level images and c=3 for colour images). Each Gaussian G_i (i = 1...K) is modelled by a mean $\boldsymbol{\mu}_i \in \Re^c$ and a variance σ_i^2. G_i is also associated with a weighting factor w_i. All the Gaussians are always sorted in non-increasing w/σ^2 order.

To determine if the pixel p belongs to foreground, $\mathbf{I}_t$ is compared to each G_i as follows:

$$(\mathbf{I}_t - \boldsymbol{\mu}_{i(t-1)})^T (\mathbf{I}_t - \boldsymbol{\mu}_{i(t-1)}) \begin{cases} < 6.25 * \sigma_{i(t-1)}^2 & Matched \\ otherwise & Unmatched \end{cases}$$

Following this, three cases are identified:

1. If no match is found, then the Gaussian with least w/σ^2 (i.e. G_K) is updated with a new $\boldsymbol{\mu}$ set to $\mathbf{I}_t$, σ^2 being a big value σ_{big}^2 and its weight w being 0. Pixel p is declared to be a foreground pixel.

2. If multiple matches are obtained, then only the Gaussian with biggest w/σ^2 is set to "matched" and all others are set to "unmatched". This becomes case 3 below.

3. If only one match is found, the following check on this matched G_j is performed:

$$\begin{cases} p \in background & if\ j \le B\ (where\ B = \arg\min_b (\sum_{i=1}^{b} w_i \ge T \cdot \sum_{i=1}^{K} w_i)) \\ p \in foreground & otherwise \end{cases}$$

where $0<T<1$ and it is a portion factor by which the background should at least appear in the video sequence.

All unmatched distributions maintain their statistical properties ($\boldsymbol{\mu}$ and σ^2) while their weights are reduced as $w_t = (1-\alpha) w_{t-1}$. The matched Gaussian G_j is updated by IIR (Infinite Impulse Response) filters as:

$$\boldsymbol{\mu}_{jt} = (1-\alpha) \cdot \boldsymbol{\mu}_{j(t-1)} + \alpha \cdot \mathbf{I}_t \quad (1)$$

$$\sigma_{jt}^2 = (1-\alpha) \cdot \sigma_{j(t-1)}^2 + \alpha \cdot (1-\alpha) \cdot (\mathbf{I}_t - \boldsymbol{\mu}_{j(t-1)})^T (\mathbf{I}_t - \boldsymbol{\mu}_{j(t-1)}) \quad (2)$$

$$w_{jt} = (1-\alpha) \cdot w_{j(t-1)} + \alpha \cdot \mathbf{1.0} \quad (3)$$

where $0<\alpha<1$ is the learning factor on the incoming data (or $\mathbf{I}_t$).

Eq. 2 can be deduced as follows: Suppose measurements taken on all video frames are independent. For the pixel p this means $\mathbf{I}_1$, $\mathbf{I}_2$, ... are all independent of each other. Let η be an estimate of the average of squared intensity of pixel p, we would have:

$$\eta_{jt}^2 = (1-\alpha) \cdot \eta_{j(t-1)}^2 + \alpha \cdot \mathbf{I}_t^T \mathbf{I}_t$$

then,

$$\begin{aligned} \sigma_{jt}^2 &= \eta_{jt} - \boldsymbol{\mu}_{jt}^T \boldsymbol{\mu}_{jt} \\ &= [(1-\alpha) \cdot \eta_{j(t-1)} + \alpha \cdot \mathbf{I}_t^T \mathbf{I}_t] - [(1-\alpha) \cdot \boldsymbol{\mu}_{j(t-1)} + \alpha \cdot \mathbf{I}_t]^T [(1-\alpha) \cdot \boldsymbol{\mu}_{j(t-1)} + \alpha \cdot \mathbf{I}_t] \\ &= (1-\alpha) \cdot (\eta_{j(t-1)} - \boldsymbol{\mu}_{j(t-1)}^T \boldsymbol{\mu}_{j(t-1)}) + \alpha \cdot (1-\alpha) \cdot (\mathbf{I}_t - \boldsymbol{\mu}_{j(t-1)})^T (\mathbf{I}_t - \boldsymbol{\mu}_{j(t-1)}) \\ &= (1-\alpha) \cdot \sigma_{j(t-1)}^2 + \alpha \cdot (1-\alpha) \cdot (\mathbf{I}_t - \boldsymbol{\mu}_{j(t-1)})^T (\mathbf{I}_t - \boldsymbol{\mu}_{j(t-1)}) \end{aligned}$$

Considering that the initial video segment for background model learning may also contain moving objects, we use the foreground extraction procedure described above also for the initialisation purpose. At the beginning of the initialisation, for each pixel, all Gaussians are initialised as $\sigma = \sigma_{big}$, w=0 and $\boldsymbol{\mu}$ being a small negative value so that $\boldsymbol{\mu}^T \boldsymbol{\mu} > \mathbf{6.25 * \sigma^2}$ (e.g., $\boldsymbol{\mu} = [\mathbf{-2*\sigma \ \ -2*\sigma \ \ -2*\sigma}]^T$). It turns out that in this way the background can normally be initialised within 100 frames.

Note that here we have simplified the background model to requiring only two parameters α and T. α controls the speed of updating process. A bigger value of α makes the system prone to noise. Therefore a small value is preferable for α. In our proposed system, through extensive experiments, we found 0.005 to be the most appropriate. T represents the frequency that the background should at least present. Without any a priori knowledge about the scene, T can be safely set as 0.6.

The choice of K, as discussed in above, is important. For adaptability of the MoG model, the minimal value of K is 3. The reason behind this is: G_1 models the current background; G_2 accumulates a new background layer (the key to background updating); G_3 accommodates any new incoming non-background pixel which has potential to become background. When repetitive moving patterns such as waving leaves need to be modelled, K should be larger than 3. However, when K becomes bigger, the computation becomes far more expensive. To our opinion, in a highly correlated system, problems should be addressed in different steps where appropriate and in a balanced way. To guarantee a real-time system, we leave the problem of getting rid of repetitive moving patterns the object tracking step, where we can distinguish those patterns out by studying their special motion characteristics. Therefore, here we permanently set K = 3.

Further, optionally, at every time instance, after the foreground extraction, we judge if the background is still valid by assuming the maximum number of objects and

the maximum size of each individual object. If it is not valid any more, the background model is then re-learned. By doing this we can prevent sudden overall lighting condition change breaking down the whole system.

After the background subtraction, three steps are taken to remove three different kinds of foreground noises:

1. To smooth the shape of extracted objects, for each foreground pixel *p*, its 8-connected neighbours are examined: First, a vector **a** is constructed consisting of the 8 neighbouring pixels in clockwise order, i.e. $\mathbf{a} = [a_1\, a_2\, a_3\, a_4\, a_5\, a_6\, a_7\, a_8]^T$ with a_1 being the top-left neighbour. $a_i = 1$ if it is a background pixel, otherwise, 0; a constant vector $\mathbf{b} = [1\,2\,1\,2\,1\,2\,1\,2]^T$ is also defined. If $\mathbf{a}^T\mathbf{b}$ >6, then *p* is re-classified as a background pixel, otherwise, it remains as a foreground pixel.

2. Isolated foreground pixels like pepper-and-salt noises are eliminated by a speckle noise removal filter.

3. To avoid spurious objects produced by sudden lighting condition change, we use the normalised zero-mean cross correlation (NZCC) technique, which can verify if an extracted foreground object is actually a lightened or darkened background region or not.

3. TWO-BRANCH SHADOW REMOVAL

For each foreground pixel *p*, its Y (illumination) value and that of its counterpart on the background are calculated as: $Y = 0.114*R + 0.587*G + 0.299*B$. In the following, we use the subscript f and b to denote, respectively, the foreground and corresponding background pixel.

First, the brightness distortion value is computed as defined in [1], or simply

$$BD = \frac{R_f R_b + G_f G_b + B_f B_b}{{R_b}^2 + {G_b}^2 + {B_b}^2}$$

If $Y_f < Y_b$, or the pixel is darker than the background, a check for possible shadow is performed as:

$$\begin{cases} BD > T_{Shadow} & p \in shadow \\ otherwise & p \in foreground \end{cases}$$

where $0 < T_{Shadow} < 1$.

On the other hand, if $Y_f > Y_b$, or the pixel is brighter than the background, another check for possible highlight is performed as:

$$\begin{cases} BD < T_{Highlight} & p \in highlight \\ otherwise & p \in foreground \end{cases}$$

where $T_{Highlight}$ is a value slightly larger than 1.

To obtain a clean and true foreground mask, an appropriate setting of T_{Shadow} and $T_{Highlight}$ is very important, though hard to define. To solve this problem, in [3] Horprasert et al. proposed an automatic threshold determination scheme. However, to suit for real-time application, a better solution would be to fix the value of T_{Shadow} . In this case, to ensure the connectivity of each object's multiple body parts, either certain heuristic rules are needed or a morphological closing should be applied to the shadow-removed foreground mask with a large kernel. The former approach is highly application dependent. And the latter one will inevitably retain substantial shadows. A similar analysis applies to $T_{Highlight}$.

In view of the problems discussed above, we propose a two-branch shadow/highlight removal method. The idea is:

1. **Weak shadow elimination**: Removing fewer shadows and highlights using a bigger value of T_{Shadow} and a smaller value of $T_{Highlight}$. As a result, the smallest bounding box that covers each foreground connected region is extracted as the *object mask*.

2. **Strong shadow elimination**: Removing most shadows and highlights using a smaller value of T_{Shadow} and a bigger value of $T_{Highlight}$. The result is over-segmented foreground regions, which are named as *part regions*.

3. **Constrained foreground reconstruction**: For each object mask, all part regions lying inside it are declared to belong to one foreground object. Therefore all pixels in those part regions form one *blob mask* and are used for computing the features of this object.

It should be noted that this two-branch idea is different from the hysteresis method mentioned in [4]. The hysteresis method only used the salient features to validate detected regions. Here we give more trust to salient pixels and use vague while bigger masks to connect related salient features together. We instead solved the problem addressed by the hysteresis problem by adopting the NZCC algorithm as mentioned in section 2.

4. EXPERIMENTAL STUDIES

The proposed foreground object extraction system has been implemented using MS VC++. Currently, it runs around 33fps for frame size 320x240 on a 2.8GHz modern PC (including video decoding and encoding) without due optimisation. For all experiments, we use the same set of parameters as given in Table 1.

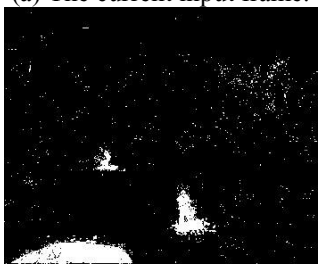
Table 1: Parameter settings used for the experiments

Background learning: K=3, α=0.005, T=0.6
Weak shadow elimination: $T_{Shadow} = 0.9$, $T_{Highlight} = 1.05$
Strong shadow elimination: $T_{Shadow} = 0.7$, $T_{Highlight} = 1.25$

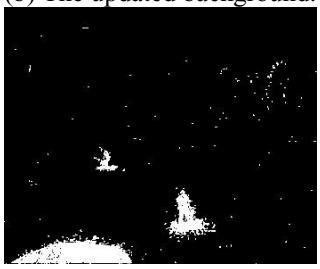
An example set of processing results is shown in Figure 2. Note that although this sequence has been subjected to twice compressions (JPEG and XviD), the system can still extract the foreground objects and track them throughout the sequence.

(a) The current input frame.

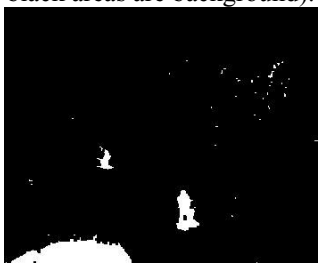
(b) The updated background.



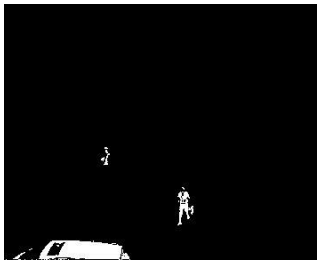
(c) The foreground mask (white pixels are foreground, black areas are background).



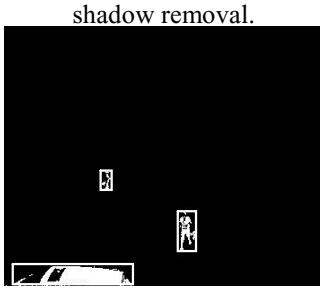
(d) The foreground mask after noise removal.



(e) Mask out from weak shadow removal.



(f) Mask out from strong shadow removal.



(g) Blob masks.

(h) Final bounding objects.

Figure 2: Foreground extraction results from an example frame.

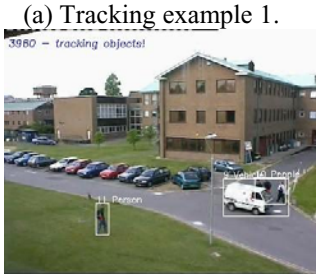
It has been further evaluated on various highly compressed video sequences (especially compressed PETS2001 benchmark database) and consistent good performances, in terms of high extraction accuracy and robustness to noise influences, have been observed. For PETS2001 sequences based on single stationary camera (both training set and corresponding test set were used), under smooth lighting change (thus excluding the background learning and re-initialisation parts), the object detection rate of false positive[1] is about 5% and that of the false negative[2] is only around 2%.

This module is further embedded into a video surveillance system, and its good performance contributes to the high quality of the whole system. Thanks to the clear skeleton provided by the strong shadow removal, a simple object classification module has also been realised.

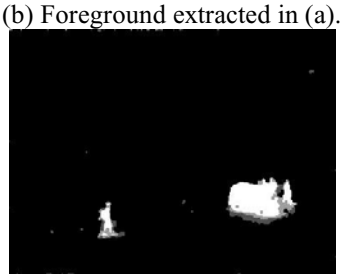
Figure 3 shows two sampled results of a different video sequence.

(a) Tracking example 1.

(b) Foreground extracted in (a).



(c) Tracking example 2.



(d) Foreground extracted in (c).

Figure 3: Two examples of foreground extraction and tracking. Note that in the segmented masks, the grey level pixels are shadows detected by strong shadow elimination module but not by weak shadow elimination module.

5. CONCLUSIONS

We have presented in this paper an effective foreground extraction procedure with shadow removal that is very practical and easily realisable. It can robustly tolerate noises introduced especially by video compression, which is unavoidable for network transmission. It works successfully in tandem with a blob-based tracking algorithm in a video surveillance system. To extend this work, we will add video stabilisation pre-processing to accommodate moving cameras and consider more carefully the handling of crowded situations.

6. REFERENCES

[1] C. Stauffer and W.E.L. Grimson, "Learning patterns of activity using real time tracking," *IEEE Trans on PAMI*, **22**(8):747-757, 2000.

[2] A. Elgammal, R. Duraiswami and L. S. Davis, "Efficient Kernel Density Estimation using the fast Gauss transform for computer vision," *IEEE Trans. on PAMI*, **25**(11):1499-1504. Dec 2003.

[3] T. Horprasert, D. Harwood, and L.S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," *Proc. of ICCV'99 Frame-Rate Workshop*, 1999.

[4] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, "Detecting Moving Objects, Ghosts and Shadows in Video Streams". *IEEE Transactions on PAMI*, vol. 25, n. 10, pp. 1337-1342, 2003.

[1] Ratio of spurious objects to total number of objects detected.

[2] Ratio of non-detected objects to total number of true objects.