# Robust Recursive Learning for Foreground Region Detection in Videos with Quasi-Stationary Backgrounds

Alireza Tavakkoli, Mircea Nicolescu and George Bebis
Computer Vision Laboratory
University of Nevada, Reno, NV 89557, USA
{tavakkol,mircea,bebis}@cse.unr.edu

## Abstract

*Detecting regions of interest in video sequences is the most important task in many high level video processing applications. In this paper a robust technique based on recursive learning of video background and foreground models is presented. Our contributions can be described along four directions. First, a recursive learning scheme is developed to build pixel models based on their colors. Second, we generate background and foreground models to enforce the temporal consistency of detected foregrounds. Third, we exploit dependencies between pixel colors to insure that the model is not restricted to using only independent features. Finally, an adaptive pixel-wise criterion is proposed that incorporates different spatial situations in the scene*[1].

```
1. Initialization; $\Delta, \alpha_0, \beta, \kappa$ and $th$
2. For each pixel in new frames
   2.1. Update $\alpha_t = \frac{1-\alpha_0}{h(t)} + \alpha_0$ and $\Delta$
   2.2. Update $\theta_t^B = (1-\beta_t)\theta_{t-1}^B + \alpha_t \cdot H_\Delta$
   2.3. If $\theta_t^B \geq th_{ij}$
            Update $\theta_t^F$
   2.4. If $ln\left(\text{med}(\theta_t^F)/\text{med}(\theta_t^B)\right) \geq \kappa_{ij}$
            Label pixel as foreground.
   2.5. Update $\kappa_{ij}$ and $th_{ij}$
3. Label and store foreground masks.
```

**Figure 1. Our proposed recursive learning algorithm.**

## 1. Introduction

In visual surveillance systems, stationary cameras are typically used. However, due to camera shake, or inherent changes in the background itself, the background of the video may not be completely stationary. In these types of backgrounds, a single background frame is not sufficient to detect moving regions. Pless *et al.* [6] evaluated different models for dynamic backgrounds. In [8], a single 3-dimensional Gaussian model for each pixel in the scene is built. A Mixture of Gaussians technique was proposed in [7] to address the multi-modality of the underlying background. The convergence speed of mixture models can be improved by sacrificing memory as proposed in [4]. However the problem of specifying the number of Gaussians and the adaptation in later stages still exists.

In [1], El Gammal *et al.* proposed a non-parametric kernel density estimation for pixel-wise background modeling without making any assumption on its probability distribution. Therefore, this method can easily deal with multi-modality in background pixel distributions without determining the number of modes in the background. In order to adapt the model a sliding window is used in [5]. However the model convergence is critical in situations where the illumination suddenly changes. In order to update the background for scene changes Kim *et al.* in [2] proposed a layered modeling technique where an additional model, called *cache*, is introduced.

In this paper we propose an adaptive learning technique in a recursive formulation. There are four major contributions presented in our proposed method. (i) The recursive formulation of the model accumulates sufficient evidence for background/foreground models through time. (ii) Dependencies between the pixel features are exploited in our implementation, resulting in more accurate models. (iii) We build up a model for both background and foreground pixels. In the classification, these models are compared and the pixels are classified as foreground or background based on the winner model to achieve temporal coherency of the modeling. (iv) Instead of a global threshold for all the pixels

[1] This work was supported in part by a grant from the University of Nevada Junior Faculty Research Grant Fund and by NASA under grant # NCC5-583. This support does not necessarily imply endorsement by the University of research conclusions.

Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

in the scene an independent threshold is trained over time to effectively perform the classification.

The rest of this paper is organized as follows: in Section 2 we present the building block of the proposed background modeling technique and we explain how the model incorporates the dependencies between features. In Section 3, classification by using a threshold map as well as enforcing the spatial consistency of the neighboring models are discussed. In Section 4 the experimental results of the proposed method are presented and the performance of this method is compared with existing techniques. Finally the conclusion of this paper is drawn in Section 5.

## 2. Adaptive Background Learning

The formulation discussed here is in one dimension as the extension to higher dimensions is straightforward. The proposed method, in pseudo-code, is shown in Figure 1.

### 2.1. Recursive Model

Let $x(t)$ be the the intensity value of a pixel at time $t$. The non-parametric estimation of the background model that accurately follows its multi-modal distribution can be reformulated in terms of recursive filtering:

$$\theta_t(\cdot) = [1 - \beta_t] \cdot \theta_{t-1}(\cdot) + \alpha_t \cdot H_\Delta \left[x_t; \theta_{t-1}(\cdot)\right] \qquad (1)$$

where $\theta_t(\cdot)$ is the model at time $t$ and is updated by the local kernel $H\left[x_t; \theta_{t-1}(\cdot)\right]$ with bandwidth $\Delta$, and $\alpha_t$ and $\beta_t$ are the learning rate and forgetting rate schedules, respectively.

In order to speed up the modeling convergence, in the proposed method we build a schedule for learning the background model at each pixel based on its history. At early stages the learning occurs faster ($\alpha(t) = 1$) and by time it decreases and converges to the target rate ($\alpha(t) \to \alpha_0$). The forgetting rate schedule is used to account for removing those values that have occurred long time ago and no longer exist in the background. These schedules will make the adaptive learning process converge faster, without compromising the stability and memory requirements of the system. Also training these rates independently for each pixel based on spatial changes in the scene makes the convergence more effective for different situations. This learning schedule is shown in equation (2).

$$\alpha(t) = \left( \frac{1 - \alpha_0}{h(t)} + \alpha_0 \right) \qquad (2)$$

Function $h(t)$ is a monotonically increasing function, used instead of $t$, to make the updating process adaptive to different situations. Once the system detects a sudden change, the function $h(t)$ resets to 1 and the learning rate jumps to its original large value. In the current implementation we assume that the forgetting rate is a portion of the learning rate; $\beta(t) = k \cdot \alpha(t)$, where $k \leq 1$.

For each pixel, all the intensities have the same probability of being foreground. However, as time passes, the background model is updated, resulting in larger model values ($\theta^B$) at some intensities in which the likelihood of having a foreground decreases. Also because the foreground models tend to be consistent over time, once a pixel is detected as foreground, the likelihood of having the same intensity value for that pixel in the next frame becomes higher. So the foreground models are updated with larger rate at those intensity values:

$$\theta_t^F = (1 - \beta_t^F) \cdot \theta_{t-1}^F + \alpha_t^F \cdot H_\Delta \left[x_t^F; \theta_{t-1}^F\right] \qquad (3)$$

### 2.2. Capturing Feature Dependencies

To extend the modeling in higher dimensions and using color and spatial information, we can consider each pixel as a 5 dimensional feature vector in $\mathbf{R}^5$, as $f\left(R, G, B, x, y\right)$. The kernel $H$ in this space is a multivariate kernel $H_\Delta$. In this case, instead of using a diagonal matrix $H_\Delta$, we use a full multivariate kernel. The kernel bandwidth matrix $\Delta$ is a symmetric positive definite $d \times d$ matrix. Once each pixel is labeled as background, having accumulated enough evidence, its features are used to update the bandwidth matrix. Let's assume that we have $N$ pixels, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, labeled as background. We build a $3 \times N - 1$ matrix $\mathbf{X} = \left\{\mathbf{x_i} - \mathbf{x_{i-1}} | i = 2, \cdots, N; \mathbf{x_i} = [r_i, g_i b_i]^T\right\}$ of successive deviations. The bandwidth matrix is a updated by:

$$\Delta_{d \times d} = \begin{bmatrix} \Sigma & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \quad ; \quad \Sigma = X \cdot X^T \qquad (4)$$

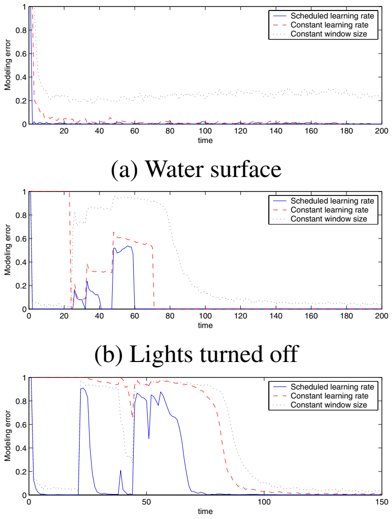
## 3. Foreground/Background Classification

For each pixel at time $t$, we have a function $\theta_B(t)$ for the background model and $\theta_F(t)$ for the foreground. For simplicity, assume the one dimensional case again, where $\theta_X(t)$ is the background/foreground model whose domain is $[0, 255]$. From equation (3), each model ranges between 0 to 1 and its value shows the amount of evidence accumulated in the updating process. The classification step uses a MAP criterion, $ln \left( \frac{\theta_B(t)}{\theta_F(t)} \geq \kappa \right)$, to label the pixel as foreground if this criterion is satisfied.

Because not all the pixels in the scene follow the same changes, the decision threshold $\kappa$ should be adaptive and independent for each pixel and has to be driven from the history of that pixel. Figure 2 explains this issue, where Figure

(a) Arbitrary frame (b) Threshold map

**Figure 2. Adaptive classification criteria**



(a) Water surface

(b) Lights turned off

(c) Lights turned on

**Figure 3. Convergence and recovery speed**

2(a) shows an arbitrary frame of a video sequence containing water surface. We expect that for pixels with more fluctuations, such as water surface, the value $\kappa$ needs to be small in short term. This can be observed in Figure 2(b), where darker parts refer to smaller values for $\kappa$ and brighter ones show larger values. As mentioned in Section 2 and Figure 1, we have two set of thresholds, *th* and $\kappa$. Thresholds $th_{ij}$, for each pixel $(i, j)$, should adapt to a value T, where $\int_{\theta^B \geq T} \theta_t^B(x)dx \geq 0.95$. For the competitive thresholds, $\kappa_{ij}$, we use a measurement from changes in the intensity at its pixel position, $(i, j)$:

$$\kappa_{ij} = \frac{1}{255} \sum_{0}^{255} (\theta_{ij}^B - \text{mean}(\theta_{ij}^B))^2 \qquad (5)$$

## 4. Experimental Results and Comparison

In this section, we present the results of the proposed method on difficult situations and compare its performance with existing techniques quantitative and qualitatively.

**Convergence speed.** Our first experiment compares the convergence and recovery speed of our proposed scheduled learning rates with the fixed learning rate and constant window size used in non-parametric density estimation. One sample frame of *water surface* video is shown in Figure 2(a). Figure 3(a) shows the convergence speed of the proposed method where the modeling error is plotted against time. The solid curve shows the model error, using the proposed scheduled learning. The dashed curve shows the effect of a constant, large learning rate, which converges slower than our proposed method and finally the dotted curve shows the effect of a non-parametric density estimation, with a constant small window size.

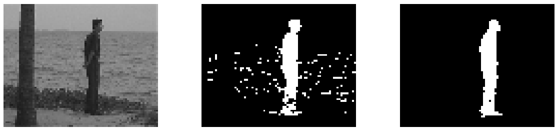


**Figure 4. Water surface: (a) Arbitrary frame, Foreground masks detected: by KDE (b), and our method (c).**

**Figure 5. Shopping mall: (a) First frame. (b) Background model after 50 frames.**

**Recovery speed.** Figures 3(b) and 3(c) show the comparison for the recovery speed of the model from an expired background to the new one. This happens in the situation where in an indoor scene, lights go off (Figure 3(b)) or they go on (Figure 3(c)). As it can be seen in Figure 3(b), our method recovers the background model fast. The constant, large learning rate recovers much slower, shown by the dashed curve, and the non-parametric density estimation technique, the dotted curve, is not able to recover even in a long term. A similar situation, when lights are turned on, is shown in Figure 3(c).

**Irregular motion.** By using the *water surface* video sequence, we compare the results of foreground region detection using our method and the method in [1]. For this comparison the sliding window of size L=150 is used in the KDE method. The results of the KDE method are shown in Figure 4(b) and the foreground masks detected by our proposed technique are shown in Figure 4(c).

**Initially non-empty scene.** Figure 5, the *Shopping mall* sequence, shows the performance of the proposed method in situations where the first frames do not contain only the

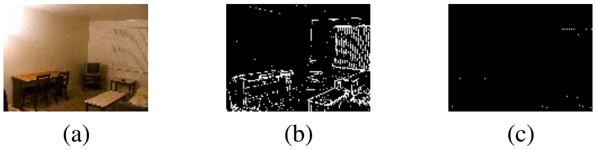

(a) (b) (c)

**Figure 6. Hand-held camera.**

**Table 1. Quantitative comparison.**

| Videos | MR | LB | CAM | SW | WS | FT | Avg |
|---|---|---|---|---|---|---|---|
| Proposed | 0.92 | 0.87 | 0.75 | 0.72 | 0.89 | 0.87 | 0.84 |
| [3] | 0.91 | 0.71 | 0.69 | 0.57 | 0.85 | 0.67 | 0.74 |
| [7] | 0.44 | 0.42 | 0.48 | 0.36 | 0.54 | 0.66 | 0.49 |

background, but some foreground objects as well. In this situations both traditional parametric and non-parametric background modeling techniques fail. As it can be observed in Figure 5(a), the video does not have a clear set of background frames to be modeled. Our technique starts with the first frame and incorporates the information from new frames to build its background and foreground models. The resulting background model is visualized in Figure 5(b) after 50 frames, where the objects existing in the first frame faded to achieve a clear background model.

**Hand-held camera.** Figure 6, the *Room* video sequence, shows an experiment on a video taken with a hand-held camera. The camera movement is quite noticeable, yet it is not large enough to classify this video under categories containing global motion. Because the movement of the camera does not follow a specific pattern and is slow, it is very difficult to use a global motion filter to detect its background and foreground regions. One arbitrary frame of such a video is shown in Figure 6(a). Figures 6(b) and (c) show the result of proposed background modeling on frames 2 and 247, respectively. It can be seen that the amount of false positives decreases by time.

**Quantitative evaluation.**

The performance of our proposed method is evaluated quantitatively on randomly selected samples from different video sequences, taken from [3]. The similarity measure between two regions $\mathcal{A}$ and $\mathcal{B}$ is defined by, $\mathcal{S}(\mathcal{A},\mathcal{B}) = \frac{\mathcal{A}\cap\mathcal{B}}{\mathcal{A}\cup\mathcal{B}}$. This measure is monotonically increasing with the similarity of the detected masks and the ground truth, with values between 0 and 1. We calculated the average of similarity measure of the foreground masks detected by our proposed method, the Mixtures of Gaussians in [7] and [3]. By comparing the average of the similarity measure over different video sequences in Table 1, we can see that the proposed method outperforms techniques proposed in [7] and [3], while there are no parameters to be heuristically selected in our proposed method.

## 5. Conclusion and Future Work

As the main contribution of this paper, a recursive learning scheme for background and foreground modeling is presented. The adaptive learning and forgetting rates proposed here make the generated models adapt to gradual and sudden changes. Our second contribution is that the decision criterion for each pixel is trained independently, based on the pixel model. Because these criteria are data driven, they are automatically updated and add to the accuracy of the overall performance. Third, two models are built separately for foreground and background and the detection is performed by competitively comparing these models to achieve temporal coherence. Finally, dependencies between pixel features are captured using multivariate models. Spatial consistency of models and the extracted foreground regions are enforced using the spatial coherency of pixel values. The experimental results show that the system converges reasonably fast to the underlying models and is able to recover fast from each expired model.

One direction of future investigation is to use this work in non-parametric tracking approaches. Also by optimizing the learning rate schedules we can improve the result of foreground object detection.

## References

[1] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90:1151–1163., 2002.

[2] K. Kim, D. Harwood, and L. S. Davis. Background updating for visual surveillance. *In Proceedings of the International Symposium on Visual Computing*, 1:337–346, December 2005.

[3] L. Li, W. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. on Image Processing.*, 13(11):1459–1472, November 2004.

[4] S. McKenna, Y. Raja, and S. Gong. Object tracking using adaptive color mixture models. *In Proc. Asian Conferenc on Computer Vision*, 1:615–622, January 1998.

[5] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. *In Proceedings of CVPR*, 2:302–309, July 2004.

[6] R. Pless, J. Larson, S. Siebers, and B. Westover. Evaluation of local models of synamic backgrounds. *In proceedings of the CVPR*, 2:73–78, June 2003.

[7] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on PAMI*, 22(8):747–757, August 2000.

[8] C. Wern, A. Azarbayejani, T. Darrel, and A. Petland. Pfinder: real-time tracking of human body. *IEEE Transactions on PAMI*, 19(7):780–785, July 1997.

0-7695-2521-0/06 $20.00 © 2006 IEEE