An Adaptive Recursive Learning Technique for Robust Foreground Object Detection

Alireza Tavakkoli[1], Mircea Nicolescu[1], and George Bebis[1]

Computer Vision Laboratory, University of Nevada,
Reno, Nevada, 89557 USA
{tavakkol, mircea, bebis}@cse.unr.edu
http://www.cse.unr.edu/CVL

Abstract. Detection of foreground objects in video requires a robust technique to model its background. Current background modeling techniques use heuristics to build a representation of the background, while it would be desirable to obtain this model automatically. Also permanent changes to the background model, such as removed/added objects to the scene are not addressed explicitly in currently existing methods. In this paper a robust technique based on recursive learning of video background model is presented. The proposed modeling technique achieves a fast convergence speed and an adaptive, accurate background model. Our contributions can be described along four directions. First, a recursive learning scheme is developed to build the models based on pixel features; i.e. color. Second, for each pixel in the scene, a distinct classification criterion is derived from its background model and is used to label it as background/foreground. 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 schedule is proposed used to adapt the model convergence. The proposed method has this ability to re-insert the uncovered parts of the background into its model while handling difficult dynamic backgrounds.

1 Introduction

In visual surveillance systems, stationary cameras are typically used. However, due to camera shake, or inherent changes in the background itself, such as fluctuations in monitors and fluorescent lights, waving flags and trees, water surfaces, etc. the background of the video may not be completely stationary. In these types of backgrounds, referred to as quasi-stationary backgrounds, a single background frame is not useful to detect moving regions. Pless *et al.* [1] evaluated different models for dynamic backgrounds. Typically background models are defined independently on each pixel, and depending on the complexity of the problem, use the expected pixel features (i.e. colors) [2] or consistent motion [3]. Also they may use pixel-wise information [4] or regional models of the features [5].

In [4], a single 3-dimensional Gaussian model for each pixel in the scene is built, where the mean and covariance of the model were learned in each frame. Kalman Filtering [6] is also used to update the model. These background models were unable to follow and represent multi-modal situations. A Mixture of Gaussians modeling technique was proposed in [7] and [8] to address the multi-modality of the underlying background. There are several shortcomings for the mixture learning methods. First of all, the number of Gaussians needs to be specified. Second, this method does not specifically deal with spatial dependencies. Also, even with the use of incremental-EM, the parameter estimation and its convergence is noticeably slow where the Gaussians adapt to a new cluster. The convergence speed can be improved by sacrificing memory as proposed in [9], limiting its applications where mixture modeling is pixel-based and over long temporal windows. A recursive filter formulation is proposed by Lee in [10]. However the problem of specifying the number of Gaussians as well as the adaptation in later stages still exists. Also this model does not account for the situations where the number of Gaussians change due to occlusion or uncovered parts of the background.

In [2], 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. However there are several issues to be addressed using non-parametric kernel density estimation. First, these methods are memory and time consuming. For each pixel in each frame the system has to compute the average of all the kernels centered at each training feature vector. Second, the size of the temporal window used as the background buffer needs to be specified. Too small a window increases the estimation speed, while it does not incorporate enough history for the pixel, resulting in a less accurate model. Also the adaptation will be critical by using small window sizes. Increasing the window size improves the accuracy of the model but with the cost of memory and slower convergence. Finally, the non-parametric KDE methods are pixel-wise techniques and do not use the spatial correlation of the pixel features. In order to adapt the model a sliding window is used in [11]. However the model convergence is critical in situations where the illumination suddenly changes. In order to update the background for

scene changes such as moved objects, parked vehicles, or opened/closed doors, Kim *et al.* in [12] proposed a layered modeling technique. This technique needs an additional model called *cache* and assumes that the background modeling is performed over a long period. It should be used as a post-processing stage after the background is modeled.

In this paper we propose an adaptive learning technique in a recursive formulation to generate and maintain the background model. There are four major contributions presented in our proposed method. (i) The recursive formulation accumulates sufficient evidence for background model through time, and unlike the mixture learning techniques does not assume any parameters and probability densities for the background pixels. Unlike non-parametric density estimation techniques, our method does not use any fixed size temporal window, as the learning rates are independent and adaptive at each pixel. This builds the background model, using information in all frames from the beginning, without compromising the stability of the model and memory needs. By incorporating a learning rate, the model converges to the actual one at each pixel and a forgetting rate is proposed to account for those background samples that are not valid anymore due to occluded or uncovered parts of the background. This implies an independent, variable and adaptive window size for the background pixels that can deal with difficult situations such as suddenly changing background. (ii) Dependencies between the pixel features are exploited in our implementation, resulting in more accurate models. (iii) We build up a model for each background pixel and these models are used to derive an adaptive decision criterion for each pixel. In the classification, these models are compared and the pixels are classified as foreground or background based on the adaptive classification criteria. (iv) In the proposed method instead of a global learning rate for all the pixels in the scene an independent, adaptive learning schedule is used over time to enhance the model convergence. Finally, we use the spatial correlation of the models for neighboring pixels to achieve the spatial consistency of the background and foreground models.

The rest of this paper is organized as follows: in Section 2 the proposed algorithm is presented 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

In this section we describe the proposed recursive learning scheme. The formulation is discussed in one dimension as the extension to higher dimensions is straightforward. Then we discuss how dependencies of pixel features in higher dimensions can be captured. The proposed method, in pseudo-code, is shown in Figure 1.

1. Initialization; Δ, α_0, β and κ
2. For each pixel in new frames $\mathbf{x}(i,j)$:
   2.1. Update $\alpha_t = \frac{1-\alpha_0}{h(t)} + \alpha_0$ and Δ
   2.2. Update $\theta_t = (1-\beta_t)\theta_{t-1} + \alpha_t \cdot H_\Delta$
   2.3. If $\text{med}(\theta_t) \leq \kappa_{ij}$
        Label pixel as foreground.
   2.4. Update κ_{ij}
3. Label and store foreground masks.
4. Proceed to the next frame.

Fig. 1. Our proposed recursive learning algorithm.

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 as:

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

where $\theta_t(\cdot)$ is the probability density function of each pixel at time t and is updated by the local kernel $H\left[x_t; \theta_{t-1}(\cdot)\right]$ with bandwidth Δ, and α_t and β_t are the learning rate and forgetting rate schedules, respectively. In currently existing methods, both parametric and non-parametric, the learning rates are selected to be constant and have small values. This makes the convergence of the pixel model to be slow and keeps its history in the recent temporal window of size $L = 1/\alpha$. This window size in non-parametric models is critical as we need to cover all the possible fluctuation of the background model. In other words, the changes of a pixel intensity value may not be periodic and regular, thus do not fit in a temporal window. In such cases larger windows are needed, resulting in more memory requirements and computational power to achieve real-time modeling. Another issue in existing non-parametric techniques is that window size is fixed and the same for all pixels in the scene. Notice that some pixels may have less fluctuations and therefore need smaller windows to be accurately modeled, while others may need a much longer history to cover all of their possible changes.

In order to speed up the convergence of the modeling, 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).

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

Function $h(t)$ is a monotonically increasing function, used instead of t, to make the updating process adaptive to different situations, such as sudden changes in the illumination and removal/adding new objects to the background. Once the system detects a sudden change, the function $h(t)$ resets to 1 and the learning rate jumps to its original large value, improving the model recovery speed. In the current implementation we assume that the forgetting rate is a portion of the learning rate; $\beta(t) = k \cdot \alpha(t)$, where $k \leq 1$ and is chosen based on the amount of inherent changes in the background. The less forgetting rate is chosen, the more history of covered background is preserved.

2.1 Capturing Feature Dependencies

In the above section we described the recursive learning scheme in 1-D where background and foreground models are updated using intensity values of the pixel at each time. 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 $f(R, G, B, x, y)$. The kernel H in this space is a multivariate kernel H_Δ. In this case, instead of using a diagonal matrix H_Δ, we use a full multivariate kernel. The kernel bandwidth matrix Δ 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:

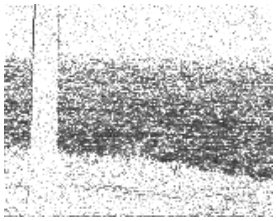
$$\Delta_{d \times d} = \begin{bmatrix} \Sigma & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} ; \Sigma = X \cdot X^T \tag{3}$$

3 Foreground/Background Classification

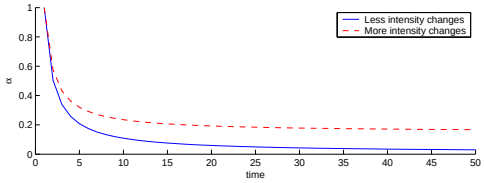
For each pixel, considering that current time is t, we have a function $\theta_t(\cdot)$ for the background model. The domain of this function is $\mathbf{R}^N$, where N is the dimensionality of the pixel feature vector. For simplicity, assume the one dimensional case again, where $\theta_t(\cdot)$ is the background model whose domain is $[0, 255]$, because intensity values are gray scale and take values between 0 to 255. From equation (2), each model ranges between 0 to 1 and its value shows the amount of evidence accumulated in the updating process; i.e. the estimated probability. For each new intensity value, I, we have the evidence of each pixel model as $\theta_t(I)$. The classification uses a simple decision rule, $\theta_t(\cdot) \leq \kappa$ to label the pixel as foreground if this criterion is satisfied.

In many applications with dynamic or quasi-stationary backgrounds, we need an adaptive classification criteria. Because not all the pixels in the scene follow the same changes the decision threshold κ should be adaptive and independent for each pixel and has to be driven from the history of that pixel. Figure

(a) Arbitrary frame



(b) Threshold map



(c) Learning rate schedule

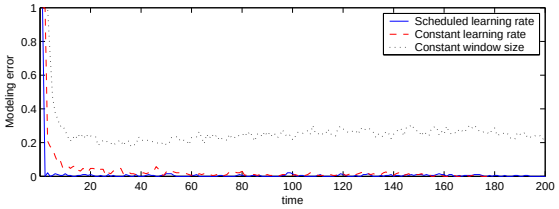
Fig. 2. Adaptive classification criteria

2(a) shows an arbitrary frame of a video sequence containing water surface. As expected, pixels in the sky do not have much inherent fluctuation, but those belonging to the water surface change their values through time. It is expected that when the pixel values do not change much, fewer samples give enough evidence for the background (or foreground) model, but those with more fluctuation need more samples to gather the same amount of evidence. We expect that for pixels with more inherent changes, the value κ needs to be small in short term, while for those pixels with less changes, larger values for κ work well to label them correctly as background or foreground. This can be observed in Figure 2(b), where darker parts refer to smaller values for κ and brighter ones show larger values. Thresholds κ_{ij}, for each pixel (i, j), should adapt to a value T such that the classifier gives the 5% false negative rate:

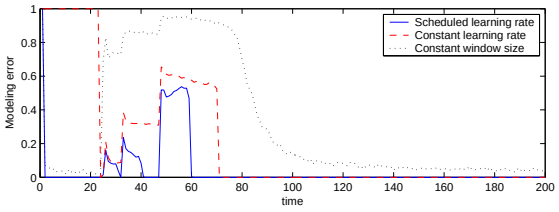
$$\sum_{x=0,\theta\geq T}^{255} \theta_t(x) \geq 0.95 \tag{4}$$

The same argument is valid for the learning schedules. For those pixels with more changes, the learning schedule has to decrease slower to compensate for the small rate of evidence accumulation; shown in Figure 2(c). Thus the derivative of function $h(t)$, in equation (2) is inversely proportional to the model variance; $h(t) = \frac{t}{\lambda \mathrm{var}(\theta(\cdot))}$ where λ is chosen so that equation (2) is normalized.

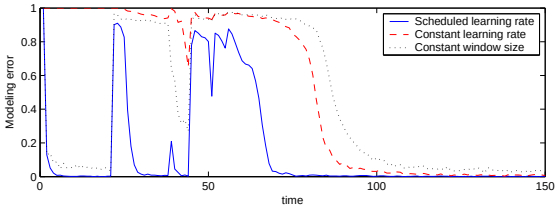
The temporal consistency is addressed in the recursive background model learning, but we have not explicitly incorporated spatial consistency of the models. In other methods such as [7] and [13] the spatial consistency is addressed using connected components and morphological post processing. The correlation between neighboring pixels in video frames can be modeled using the Markov Random Field property [14].



(a) Water surface



(b) Lights turned off



(c) Lights turned on

Fig. 3. Convergence and recovery speed

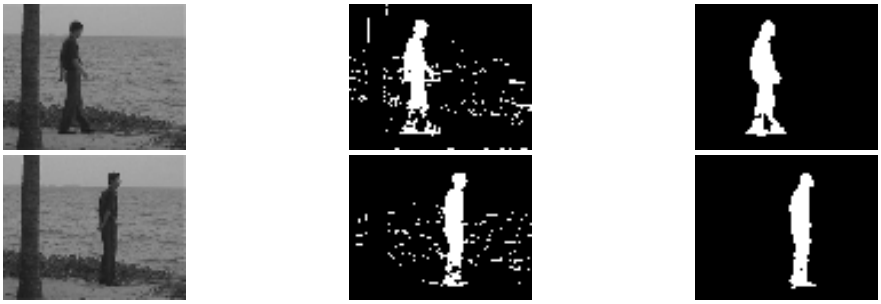
The spatial consistency in our proposed method is enforced on foreground and background regions as an intermediate process. The main idea is apply the classification criterion on the median on the models in a spacial neighborhood instead of the model itself. Given a pixel values $I(x, y)$ at time t its model is $\theta_t^{xy}(I(x, y))$. The median of models in an 8-connected neighborhood $\mathbf{C}$, can be computed by $\hat{\theta}^{\mathbf{C}} = \text{med}(\theta_t^{\mathbf{C}}(I_{\mathbf{C}}))$ where $I_{\mathbf{C}}$ are pixel values in the neighborhood.

This explicitly addresses the coherence between neighboring pixels. However, because the proposed adaptive learning technique uses long-term information, the effect of noise becomes less dominant, as the model learns the complex modes of the underlying data.

4 Experimental Results and Comparison

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

Convergence speed. Our first experiment compares the convergence and recovery speed of our proposed scheduled learning rates with the fixed learning



(a) Original frame (b) KDE (c) Our method

Fig. 4. Water surface: Comparison of methods.

rate and constant window size used in non-parametric density estimation. In this experiment we took a video containing water surface as a part of its background. One sample frame of this video is shown in Figure 2(a). Figure 3(a) shows the convergence speed of the proposed method. In this figure we have taken the first 300 frames of the *water surface* video sequence. The modeling error is plotted against time. As there were no objects in the scene, the modeling error is the difference of the background model from the actual background, which can also be considered as normalized number of false positives. The solid curve shows the error of the model using the proposed scheduled learning. The model converges to the actual background in less than 10 frames. The dashed curve shows the effect of a constant, large learning rate, which converges slower than our method and finally the dotted curve shows the effect of a non-parametric density estimation, with a constant small window size. Because the size of the window is small, the model converges in about 20 frames, but it can not learn all the possible changes in the background model, so it converges to a higher error. By increasing the window size, the error decreases with the price of convergence speed.

Recovery speed. Figures 3(b) and 3(c) show the comparison of the recovery speed of the model from an expired background model 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)) or when a new object is permanently added to or removed from the background. In Figure 3(b) there are three global illumination changes at frames 23, 31 and 47, consequently and it stabilizes after frame 47. As it can be seen in Figure 3(b), our proposed method recovers the background model after these changes in less than 4 frames. 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 150 frames. A similar situation, when lights are turned on, is shown in Figure 3(c). It needs to be mentioned that the mixture learning algorithms are even slower in convergence and recovery. A typical mixture learning technique proposed in [7], converges in more than 1000 frames.

Irregular motion. By using the *water surface* video sequence, we compare the results of foreground region detection using our proposed method with a

(a) Office (b) Fountain (c) Campus

Fig. 5. Result of the proposed foreground region detection.

(a) (b) (c)

Fig. 6. Shopping mall: (a) First frame of the scene. (b) The background model after 50 frames and (c) after 95 frames.

typical non-parametric kernel density estimation [2]. For this comparison the sliding window of size L=150 is used in KDE method. The results of KDE method are shown in Figure 4(b) and the foreground masks detected by our proposed technique are shown in Figure 4(c). Because in the water surface the changes occur slowly and do not have any regular patterns, the model (even with a large window size), is not able to learn all the changes, resulting in detection of some waves on the water surface. Our technique learns all the possible changes and the temporal consistency of the foreground masks is maintained by using both foreground and background models.

Challenging environments. Shown in Figure 5 are the results of the proposed foreground detection method on several challenging video sequences. In Figure 5(a), an indoor situation with *flickering monitors* is shown. In Figure 5(b), there is a fountain in the background of the scene having dynamic texture and in Figure 5(c) there are *waving trees* as a typical outdoor scenario causing an irregular dynamic pattern in the background. In all of the above cases, our proposed method is able to detect the foreground regions accurately and ignores the background movements.

Initially non-empty scene. Figure 6, *shopping mall* sequence, shows the performance of the proposed method in situations where the first frames do not contain only the background, but some foreground objects as well. In this situations both traditional parametric and non-parametric background modeling

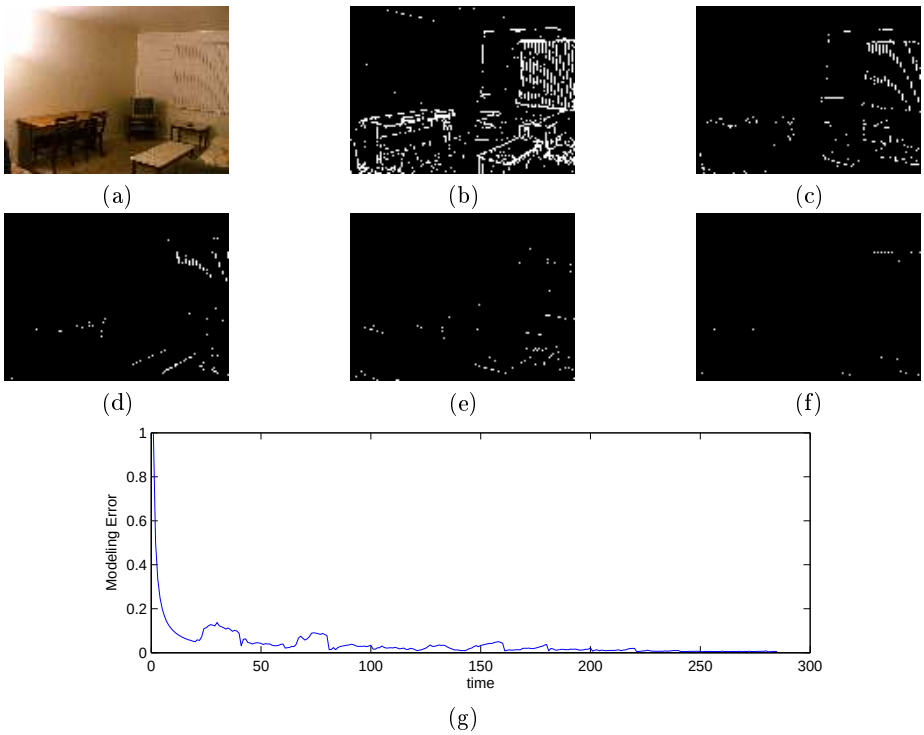


Fig. 7. Room sequence: Result of the proposed method on modeling the background of a video taken by a hand-held camera.

techniques fail to incorporate the uncovered background fast enough. As it can be observed in Figure 6(a), the video does not have a clear set of background frames to be modeled by a parametric or non-parametric technique using a constant sized temporal window. Our proposed learning 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 6(b) and 6(c) after 50 and 95 frames (about 1.5 and 3 seconds, respectively.) Our proposed method fades the objects that existed in the first frame to achieve a clear background model.

Hand-held camera. Figure 7, *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 7(a). Figures 7(b)-(f) show the result of proposed background modeling on frames 2, 32, 61, 120 and 247, respectively. These frames are approximately 0.067, 1, 2, 4 and 8 seconds after the camera starts taking the video. White

pixels show those parts of the background erroneously labeled as foreground. It can be seen that the amount of misclassified background pixels decreases by time, showing that those pixels have gathered enough evidence and have seen all the possible movement of the camera. This is also quantitatively illustrated in Figure 7(g).

Table 1. Quantitative evaluation and comparison. The sequences are Meeting Room, Lobby, Campus, Side Walk, Water Surface and Fountain, from left to right from [13].

Videos	MR	LB	CAM	SW	WS	FT	Avg
Proposed	0.92	0.87	0.75	0.72	0.89	0.87	0.84
[13]	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

Quantitative evaluation. The performance of our proposed method is evaluated quantitatively on randomly selected samples from different video sequences, taken from [13]. Detection results of some of these sequences are shown in Figures 4 and 5. Figure 4 shows *water surface* video sequence and Figure 5 (b) and 5 (b) show *fountain* and *campus* vide sequences, respectively. 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 [13]. 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 [13], while there are no parameters to be heuristically selected in our proposed method. This can also be observed by the fact that the masks detected by the proposed method are more consistent on different video sequences.

5 Conclusion and Future Work

As the main contribution of this paper, an adaptive learning scheme for background and foreground modeling is presented in a recursive formulation. The adaptive learning and forgetting rates proposed here make the generated models adapt to gradual and sudden changes. As our second contribution, 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, by introducing adaptive learning rate schedules, modeling is temporally coherent and accounts for increased recovery rate in situations where new objects are introduced to or removed from the background. Finally, dependencies between pixel features can be captured using multivariate models. The experimental results show that the system converges reasonably fast to the underlying models and is able to recover fast from

each expired model. This ensures that our method re-inserts the uncovered parts of the background into the background model while handling difficult dynamic backgrounds such as water surface, waving trees, rain/snow, etc.

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 and recovery speed.

6 Acknowledgements

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.

References

1. Pless, R., Larson, J., Siebers, S., Westover, B.: Evaluation of local models of synamic backgrounds. In proceedings of the CVPR **2** (2003) 73–78
2. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of the IEEE **90** (2002) 1151–1163.
3. Pless, R., Brodsky, T., Aloimonos, Y.: Detecting independent motion: The statistics of temporal continuity. IEEE Transactions on PAMI **22** (2000) 68–73
4. Wern, C., Azarbayejani, A., Darrel, T., Petland, A.: Pfinder: real-time tracking of human body. IEEE Transactions on PAMI **19** (1997) 780–785
5. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principels and practive of background maintenance. In proceedings of ICCV **1** (1999) 255–261
6. Koller, D., adn T. Huang, J.W., Malik, J., Ogasawara, G., Rao, B., Russel, S.: Towards robust automatic traffic scene analysis in real-time. ICPR **1** (1994) 126–131
7. Stauffer, C., Grimson, W.: Learning patterns of activity using real-time tracking. IEEE Transactions on PAMI **22** (2000) 747–757
8. Friedman, N., Russell, S.: Image segmentation in video sequences: A probabilistic approach. Annual Conference on Uncertainty in Artificial Intelligence (1997) 175–181
9. McKenna, S., Raja, Y., Gong, S.: Object tracking using adaptive color mixture models. In Proc. Asian Conferenc on Computer Vision **1** (1998) 615–622
10. Lee, D.S.: Effective gaussian mixture learning for video background subtraction. IEEE Transactions on PAMI **27** (2005) 827–832
11. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In Proceedings of CVPR **2** (2004) 302–309
12. Kim, K., Harwood, D., Davis, L.S.: Background updating for visual surveillance. In Proceedings of the International Symposium on Visual Computing **1** (2005) 337–346
13. Li, L., Huang, W., Gu, I., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. IEEE Trans. on Image Processing. **13** (2004) 1459–1472
14. Sheikh, Y., Shah, M.: Bayesian object detection in dynamic scenes. In Proceedings of the CVPR **1** (2005) 74–79