A Bayesian Approach to Background Modeling

Oncel Tuzel[†§] Fatih Porikli[§] Peter Meer[†‡]

[†]CS Department & [‡]ECE Department
Rutgers University
Piscataway, NJ 08854

[§]Mitsubishi Electric Research Laboratories
Cambridge, MA 02139

Abstract

Learning background statistics is an essential task for several visual surveillance applications such as incident detection and traffic management. In this paper, we propose a new method for modeling background statistics of a dynamic scene. Each pixel is represented with layers of Gaussian distributions. Using recursive Bayesian learning, we estimate the probability distribution of mean and covariance of each Gaussian. The proposed algorithm preserves the multimodality of the background and estimates the number of necessary layers for representing each pixel. We compare our results with the Gaussian mixture background model. Experiments conducted on synthetic and video data demonstrate the superior performance of the proposed approach.

1. Introduction

Segmentation of foreground and background regions in image sequences is one of the most fundamental tasks in computer vision. The provided information is usually crucial for higher level operations such as visual surveillance.

The obvious way to detect moving regions in image sequences is to select a reference frame while scene is stationary, and to subtract the observed frame from this image. The resulting difference image is thresholded to extract the moving regions. Although this task looks like fairly simple, in real world applications this approach rarely works. Usually background is never static and varies by time due to several reasons. The most important factors are lighting changes, moving regions and camera noise. Moreover in many of the applications, it is desirable to model the different possible appearances of the background such as shadows.

To overcome these problems, adaptive background models became more popular. Earlier adaptive methods use simple adaptive filters to make a prediction of background pixel intensities. In [8, 9] Kalman filtering is used to model background dynamics. Similarly Wiener filter is used in [14] to make a linear prediction of the pixel intensity values, given the pixel histories.

An alternative approach is to model the probability distribution of the pixel intensity. This approach ignores the order in which observations are made and focuses on the distribution of the pixel intensities. Usually each pixel is modeled with a normal distribution $N(\mu, \sigma^2)$, varying over time. Noise is assumed to be coming from a zero mean normal distribution $N(0, \sigma^2)$. In [15], a single Gaussian model is used per pixel and the parameters are updated by a simple adaptive filter.

The mentioned models perform fine if the scene background is unimodal but usually this is not the case. Multimodalities in the background is due to dynamic nature of the scenes. Fast lighting changes, moving regions and shadowed regions are some of the sources of multimodalities. To handle multimodalities the idea of using Gaussian distribution per pixel is extended by using mixture of Gaussian distributions. Mixture of three Gaussians corresponding to road, vehicle and shadow are defined in [3] for a traffic surveillance application. Likewise, Stauffer and Grimson [12] uses mixture of k normal distributions. The model parameters are updated using an online Expectation Maximization (EM) algorithm. In these models feature vectors consists of color information of the pixel. In [5], Harville et. al. extends the feature vector by depth information coming from stereo cameras. In [6] and [7] gradient information is used to achieve a more accurate background subtraction.

Although mixture of Gaussian models can converge to any arbitrary distribution provided enough number of components, this is not computationally possible for real time applications. Generally three-five components are used per pixel. Another way is to approach probability distribution of background model by nonparametric kernel density estimation [2]. The model keeps samples of intensity values per pixel and uses these samples to estimate the density function. Background subtraction is performed by thresholding the probability of observed samples. As in the parametric methods, several variations of this method is presented. In [10], motion information is used to model dynamic scenes. Although nonparametric models seems like a reasonable choice for background modeling, it is usually to costly to perform in real time. Memory and computation requirements are linear in the size of temporal window.

Other recent approaches include representing scene in discrete states corresponding to environmental conditions and switching among these states with the observations.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

Hidden Markov Models (HMMs) are very suitable for this purpose. In [11], a three state HMM is used whereas in [13] topology is learned from the observations.

In this paper, we describe a Bayesian approach to per pixel background modeling. We model each pixel as layered normal distributions. Recursive Bayesian estimation is performed to update the background parameters. Proposed update algorithm preserves multimodality of the background model and the embedded confidence score determines the number of necessary layers for each pixel.

The paper is organized as follows. Background model and update mechanism is explained in Section 2. In Section 3, we compare our method with online EM algorithm [12]. Foreground segmentation is explained in Section 4.

2 Background Model

Our background model is most similar to adaptive mixture models [12] but instead of mixture of Gaussian distributions, we define each pixel as layers of 3D multivariate Gaussians. Each layer corresponds to a different appearance of the pixel. We perform our operations on (r,g,b) color space.

Using Bayesian approach, we are not estimating the mean and variance of the layer, but the probability distributions of mean and variance. We can extract statistical information regarding to these parameters from the distribution functions. For now, we are using expectations of mean and variance for change detection, and variance of the mean for confidence.

Prior knowledge can be integrated to the system easily with prior parameters. Due to computation of full covariance matrix, feature space can be modified to include other information sources, such as motion information, as discussed in [10].

Our update algorithm maintains the multimodailty of the background model. At each update, at most one layer is updated with the current observation. This assures the minimum overlap over layers. We also determine how many layers are necessary for each pixel and use only those layers during foreground segmentation phase. This is performed with an embedded confidence score. Details are explained in the following sections.

2.1 Layer Model

Data is assumed to be normally distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Mean and variance are assumed to be unknown and modeled as random variables [4, p.87-88]. Using Bayes theorem joint posterior density can be written as:

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X}) \propto p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma})p(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (1)$$

To perform recursive Bayesian estimation with the new observations, joint prior density $p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ should have the same form with the joint posterior density $p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X})$. Conditioning on the variance, joint prior density is written as:

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\boldsymbol{\mu}|\boldsymbol{\Sigma})p(\boldsymbol{\Sigma}). \quad (2)$$

Above condition is realized if we assume inverse Wishart distribution for the covariance and, conditioned on the covariance, multivariate normal distribution for the mean. Inverse Wishart distribution is a multivariate generalization of scaled inverse-χ^2 distribution. The parametrization is

$$\boldsymbol{\Sigma} \sim \text{Inv-Wishart}_{\upsilon_{t-1}}(\boldsymbol{\Lambda}_{t-1}^{-1}) \quad (3)$$

$$\boldsymbol{\mu}|\boldsymbol{\Sigma} \sim \text{N}(\boldsymbol{\theta}_{t-1}, \boldsymbol{\Sigma}/\kappa_{t-1}). \quad (4)$$

where υ_{t-1} and $\boldsymbol{\Lambda}_{t-1}$ are the degrees of freedom and scale matrix for inverse Wishart distribution, $\boldsymbol{\theta}_{t-1}$ is the prior mean and κ_{t-1} is the number of prior measurements. With these assumptions joint prior density becomes

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{-((\upsilon_{t-1}+3)/2+1)} \times e^{\left(-\frac{1}{2}tr(\boldsymbol{\Lambda}_{t-1}\boldsymbol{\Sigma}^{-1}) - \frac{\kappa_{t-1}}{2}(\boldsymbol{\mu}-\boldsymbol{\theta}_{t-1})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\boldsymbol{\theta}_{t-1})\right)} \quad (5)$$

for three dimensional feature space. Let this density be labeled as normal-inverse-Wishart$(\boldsymbol{\theta}_{t-1}, \boldsymbol{\Lambda}_{t-1}/\kappa_{t-1}; \upsilon_{t-1}, \boldsymbol{\Lambda}_{t-1})$. Multiplying prior density with the normal likelihood and arranging the terms, joint posterior density becomes normal-inverse-Wishart$(\boldsymbol{\theta}_t, \boldsymbol{\Lambda}_t/\kappa_t; \upsilon_t, \boldsymbol{\Lambda}_t)$ with the parameters updated:

$$\upsilon_t = \upsilon_{t-1} + n \quad \kappa_n = \kappa_{t-1} + n \quad (6)$$

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}\frac{\kappa_{t-1}}{\kappa_{t-1}+n} + \overline{\mathbf{x}}\frac{n}{\kappa_{t-1}+n} \quad (7)$$

$$\boldsymbol{\Lambda}_t = \boldsymbol{\Lambda}_{t-1} + \sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T + n\frac{\kappa_{t-1}}{\kappa_t}(\overline{\mathbf{x}} - \boldsymbol{\theta}_{t-1})(\overline{\mathbf{x}} - \boldsymbol{\theta}_{t-1})^T \quad (8)$$

where $\overline{\mathbf{x}}$ is the mean of new samples and n is the number of samples used to update the model. If update is performed at each time frame, n becomes one. To speed up the system, update can be performed at regular time intervals by storing the observed samples. During our tests, we update one quarter of the background at each time frame, therefore n becomes four. The new parameters combine the prior information with the observed samples. Posterior mean $\boldsymbol{\theta}_t$ is a weighted average of the prior mean and the sample mean. The posterior degrees of freedom is equal to prior degrees of freedom plus the sample size. System is started with the following initial parameters:

$$\kappa_0 = 10, \quad \upsilon_0 = 10, \quad \boldsymbol{\theta}_0 = \mathbf{x}_0, \quad \boldsymbol{\Lambda}_0 = (\upsilon_0 - 4)16^2\mathbf{I} \quad (9)$$

where $\mathbf{I}$ is the three dimensional identity matrix.

Integrating joint posterior density with respect to $\mathbf{\Sigma}$ we get the marginal posterior density for the mean:

$$p(\boldsymbol{\mu}|\mathbf{X}) \propto t_{\upsilon_t-2}(\boldsymbol{\mu}|\boldsymbol{\theta}_t, \mathbf{\Lambda}_t/(\kappa_t(\upsilon_t-2))) \quad (10)$$

where t_{υ_t-2} is a multivariate t-distribution with $\upsilon_t - 2$ degrees of freedom.

We use the expectation of marginal posterior distribution for mean and covariance as our model parameters at time t. Expectation for marginal posterior mean (expectation of multivariate t-distribution) becomes:

$$\boldsymbol{\mu}_t = E(\boldsymbol{\mu}|\mathbf{X}) = \boldsymbol{\theta}_t \quad (11)$$

whereas expectation of marginal posterior covariance (expectation of inverse Wishart distribution) becomes:

$$\mathbf{\Sigma}_t = E(\mathbf{\Sigma}|\mathbf{X}) = (\upsilon_t - 4)^{-1}\mathbf{\Lambda}_t. \quad (12)$$

Our confidence measure for the layer is equal to one over determinant of covariance of $\boldsymbol{\mu}|\mathbf{X}$:

$$C = \frac{1}{|\mathbf{\Sigma}_{\boldsymbol{\mu}|\mathbf{X}}|} = \frac{\kappa_t^3(\upsilon_t-2)^4}{(\upsilon_t-4)|\mathbf{\Lambda}_t|}. \quad (13)$$

If our marginal posterior mean has larger variance, our model becomes less confident. Note that variance of multivariate t-distribution with scale matrix $\mathbf{\Sigma}$ and degrees of freedom υ is equal to $\frac{\upsilon}{\upsilon-2}\mathbf{\Sigma}$ for $\upsilon > 2$.

System can be further speed up by making independence assumption on color channels. Update of full covariance matrix requires computation of nine parameters. Moreover, during distance computation we need to invert the full covariance matrix. To speed up the system, we separate (r, g, b) color channels. Instead of multivariate Gaussian for a single layer, we use three univariate Gaussians corresponding to each color channel. After updating each color channel independently we join the variances and create a diagonal covariance matrix:

$$\mathbf{\Sigma}_t = \begin{pmatrix} \sigma_{t,r}^2 & 0 & 0 \\ 0 & \sigma_{t,g}^2 & 0 \\ 0 & 0 & \sigma_{t,b}^2 \end{pmatrix}. \quad (14)$$

In this case, for each univariate Gaussian we assume scaled inverse-χ^2 distribution for the variance and conditioned on the variance univariate normal distribution for the mean. The Bayesian update equations for the parameters can be found in [4, p.78-80].

2.2 Background Update

We initialize our system with k layers for each pixel. Usually we select three-five layers. In more dynamic scenes more layers are required. As we observe new samples for each pixel we update the parameters for our background model. We start our update mechanism from the most confident layer in our model. If the observed sample is inside the 99% confidence interval of the current model, parameters of the model are updated as explained in equations (6), (7) and (8). Lower confidence models are not updated.

For background modeling, it is useful to have a forgetting mechanism so that the earlier observations have less effect on the model. Forgetting is performed by reducing the number of prior observations parameter of unmatched models. If current sample is not inside the confidence interval we update the number of prior measurements parameter:

$$\kappa_t = \kappa_{t-1} - n \quad (15)$$

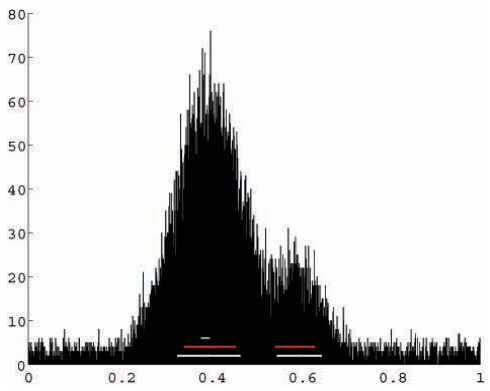
and proceed with the update of next confident layer. We do not let κ_t become less than initial value 10. If none of the models are updated, we delete the least confident layer and initialize a new model having current sample as the mean and an initial variance (9). The update algorithm for a single pixel can be summarized as follows.

Given: New sample $\mathbf{x}$, background layers $\{(\boldsymbol{\theta}_{t-1,i}, \mathbf{\Lambda}_{t-1,i}, \kappa_{t-1,i}, \upsilon_{t-1,i})\}_{i=1..k}$
Sort layers according to confidence measure defined in (13). $i \leftarrow 1$.
while $i < k$
  Measure Mahalanobis distance [1, p.36]:
  $d_i \leftarrow (\mathbf{x} - \boldsymbol{\mu}_{t-1,i})^T \mathbf{\Sigma}_{t-1,i}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{t-1,i})$.
  if sample $\mathbf{x}$ is in 99% confidence interval
    then update model parameters according to equations (6), (7), (8) and **stop**.
    else update model parameters according to equation (15).
  $i \leftarrow i + 1$
Delete layer k, initialize a new layer having parameters defined in equation (9).

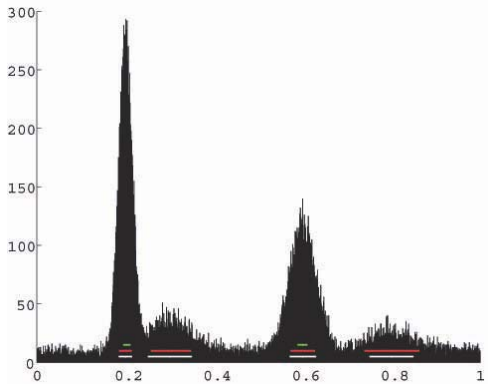
With this mechanism, we do not deform our models with noise or foreground pixels, but easily adapt to smooth intensity changes like lighting effects. Embedded confidence score determines the number of layers to be used and prevents unnecessary layers. During our tests usually secondary layers corresponds to shadowed form of the background pixel or different colors of the moving regions of the scene. If the scene is unimodal, confidence scores of layers other than first layer becomes very low.

3 Comparison with online EM

Although our model looks similar to [12], there are major differences. In [12], each pixel is represented as a mixture of Gaussian distribution and parameters of Gaussians and



(a)



(b)

Figure 1: Mixture of 1D Gaussian data corrupted with uniform noise. Lines show one standard deviation interval around the mean. Parameters are estimated with recursive Bayesian learning and online EM [12] with five Gaussians. Bottom line is the real parameters. Middle line shows estimation with recursive Bayesian learning. Topmost line shows estimation with online EM. (a) Mixture of two Gaussians. Most confident two layers estimated by two methods are shown. (b) Mixture of four Gaussians. Most confident four layers estimated by two methods are shown. There are multiple Gaussians at the same place in online EM and some modes are not detected.

mixing coefficients are updated with an online K-means approximation of EM. The approach is very sensitive to initial observations. If the Gaussian components are improperly initialized, every component eventually converges to the most significant mode of the distribution. Smaller modes nearby larger modes are never detected. We model each pixel with multiple layers and perform recursive Bayesian learning to estimate the probability distribution of model parameters. We interpret each layer as independent of other layers, giving us more flexibility.

To demonstrate the performance of the algorithm, mixture of 1D Gaussian data with uniform noise is generated. First data set consists of 12000 points corrupted with 3000 uniform noise samples and second data set consists of 23000 points corrupted with 10000 uniform noise samples. We assume that we observe the data in random order. We treat the samples as observations coming from a single pixel and estimate the model parameters with our approach and online EM algorithm. One standard deviation interval around the mean for actual and estimated parameters are plot on the histogram, in Figure 1. Results show that, in online EM, usually multimodality is lost and models converge to the most significant modes. With our method, multimodality of the distribution is maintained. Another important observation is, estimated variance with online EM algorithm is always much smaller than the actual variance. This is not surprising because the update is proportional to the likelihood of the sample, so samples closer to the mean become more important.

		Mode1	Mode2	Mode3	Mode4	Mode5
	Num.	10000	2000			
	Mean	0.4000	0.6000			
Real	Std.	0.0700	0.0500			
	Mean	0.3923	0.3919	0.3919	0.3919	0.4545
	Std.	0.0093	0.0093	0.0093	0.0093	0.0631
EM	Conf.	0.2538	0.2482	0.2481	0.2481	0.0016
	Mean	0.4021	0.5906	0.8488	0.2561	0.1133
	Std.	0.0572	0.0440	0.0820	0.0268	0.0670
Bayes	Conf.	0.7047	0.2519	0.0214	0.0208	0.0009

Table 1: Mixture of two Gaussians.

		Mode1	Mode2	Mode3	Mode4	Mode5
	Num.	10000	8000	3000	2000	
	Mean	0.2000	0.6000	0.3000	0.8000	
Real	Std.	0.0150	0.0300	0.0500	0.0500	
	Mean	0.2033	0.2033	0.5993	0.5993	0.9382
	Std.	0.0085	0.0085	0.0113	0.0113	0.0633
EM	Conf.	0.3772	0.3772	0.1221	0.1221	0.0111
	Mean	0.2002	0.5998	0.3026	0.8004	0.9387
	Std.	0.0146	0.0277	0.0451	0.0620	0.0632
Bayes	Conf.	0.3996	0.3820	0.1088	0.1087	0.0007

Table 2: Mixture of four Gaussians.

Normalized confidence scores are shown in the bottom rows of each method in Table 1 and 2. Our confidence score is very effective in determining the number of necessary layers for each pixel. Although we estimate the model parameters with five layers, it is clear from our confidence scores that how many layers are effective. There is a big gap between significant and insignificant layers.

Real data results are presented in Figure 2 and 3 where the first sequence is a traffic sequence with heavy shadows and the second sequence is a dynamic outdoor scene. In the first sequence, first and second layers of our background corresponds to the original and shadowed version of the background. The locations where most of the cars move have higher variances, so usually they are less confident.

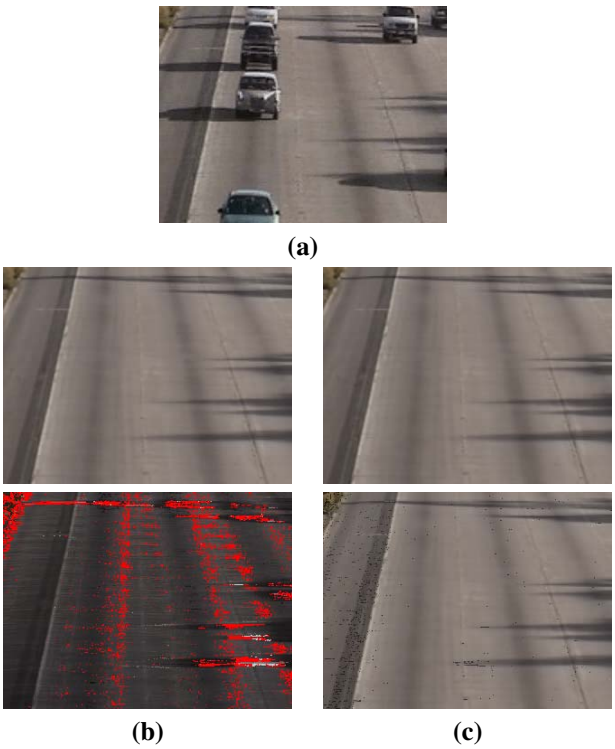


Figure 2: Traffic video with heavy shadows. (a) Original sequence. (b) Most confident two layers with recursive Bayesian learning. (c) Most confident two layers with online EM. With recursive Bayesian learning, we are able to model the shadows as the second layer of the scene whereas in EM first and second layers converge to most significant mode.

Those pixels are shown in red. First and second layers converged to the most significant mode in online EM algorithm.

In the second sequence, most significant three layers estimated by two algorithms are shown. As seen in original images, the sky and the trees are changing appearance by time. Our background model successfully modeled the different appearances of these regions. The appearance of grass does not change much with time. As expected, confidence score of second and third layers of our background are very low for this region.

4 Foreground Segmentation

Learned background statistics is used to detect the changed regions of the scene. Number of layers required to represent a pixel is not known beforehand so background is initialized with more layers than needed. As seen in Table 1, we learn background with five layers, whereas there are actually two modes. Using the confidence scores we determine how many layers are significant for each pixel. We order

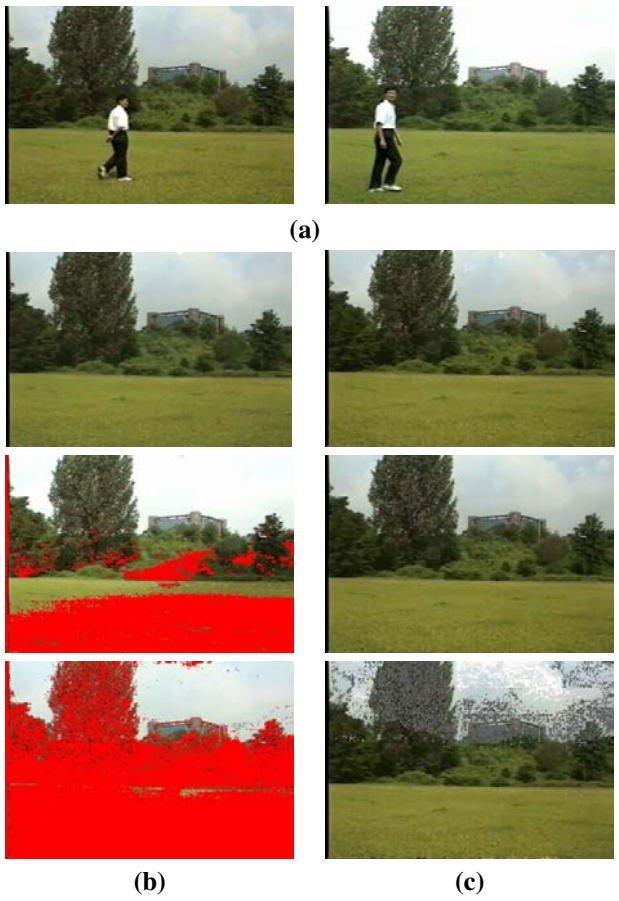
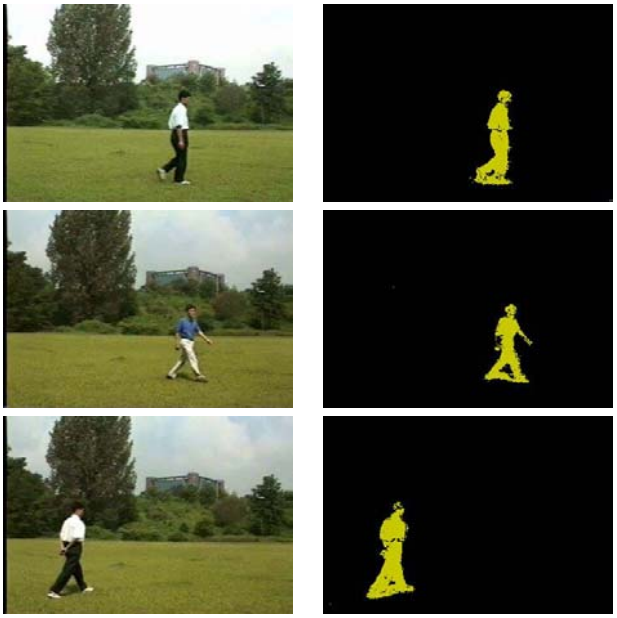


Figure 3: Outdoor video. (a) Samples from original sequence. (b) First three layers of recursive Bayesian learning. Different appearances of the background is captured with first three layers. Red pixels are unconfident layers. (c) First three layers of online EM. Second and third layers are almost same with first layer.

the layers according to confidence score (13) and select the layers having confidence value greater than the layer threshold T_c. We refer to these layers as confident layers. Note that, T_c is dependent on the covariance of mean of the pixel so it is dependent on color range of the pixel. We perform our operations in 0-255 color range and select T_c=1.0. For different color ranges T_c should be modified.

We measure the Mahalanobis distance of observed color from the confident layers. Pixels that are outside of 99% confidence interval of all confident layers of the background are considered as foreground pixels.

In Figure 4, we present foreground segmentation results of a dynamic scene. As seen in Figure 4a, appearance of background is changing with time. After some time period, our background algorithm learns the difference appearances of the background. Although the method is very sensitive



(a) **(b)**

Figure 4: Foreground segmentation of a dynamic scene. (a) Original sequence. (b) Detected foreground pixels.

to foreground objects, it does not give false alarms in background changes (Figure 4b).

5. Conclusions

Adaptive mixture models and nonparametric models are two popular methods for background modeling. Both methods have serious shortcomings. Adaptive mixture models have problems in representing multimodal scenes, whereas nonparametric methods are both memory and computationally inefficient. We have introduced a computationally efficient method for modeling background using recursive Bayesian learning approach. Our background model is very effective in estimating model parameters and representing multimodal scenes. The processing time of one frame is 0.02 seconds on a 320x240 colored video on a Pentium IV 2.4Ghz processor with five layers per background pixel. Although, this study is part of a tracking application, due to page limitation the second part is removed.

References

[1] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley, second edition, 2001.

[2] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. European Conf. on Computer Vision,* Dublin, Ireland, volume II, 2000, pp. 751–767.

[3] N. Friedman and S. Russell, "Image segmentation in video sequences," in *Thirteenth Conf. on Uncertainty in Artificial Intelligence(UAI)*, 1997, pp. 175–181.

[4] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*. Chapman and Hall, second edition, 2003.

[5] M. Harville, G. Gordon, and J. Woodfill, "Foreground segmentation using adaptive mixture models in color and depth," in *IEEE Workshop on Detection and Recognition of Events in Video*, 2001, pp. 3–11.

[6] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, "Location of people in video images using adaptive fusion of color and edge information," in *Proc. 15th Int'l Conf. on Pattern Recognition,* Barcelona, Spain, volume 4, 2000, pp. 627–630.

[7] K. Javed, O. Shafique and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *IEEE Workshop on Motion and Video Computing*, 2002.

[8] K.-P. Karman and A. von Brandt, "Moving object recognition using an adaptive background memory," in Capellini, editor, *Time-varying Image Processing and Moving Object Recognition*, volume II, (Amsterdam, The Netherlands), Elsevier, 1990, pp. 297–307.

[9] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," in *Proc. European Conf. on Computer Vision,* Stockholm, Sweden, 1994, pp. 189–196.

[10] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Washington, DC, volume II, 2004, pp. 302–309.

[11] J. Rittscher, J. Kato, S. Joga, and A. Blake, "A probabilistic background model for tracking," in *Proc. European Conf. on Computer Vision,* Dublin, Ireland, volume II, 2000, pp. 336–350.

[12] C. Stauffer and E. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Fort Collins, CO, volume II, 1999, pp. 246–252.

[13] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Buhmann, "Topology free hidden Markov models: Application to background modeling," in *Proc. 8th Intl. Conf. on Computer Vision,* Vancouver, Canada, 2001, pp. 294–301.

[14] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th Intl. Conf. on Computer Vision,* Kerkyra, Greece, 1999, pp. 255–261.

[15] C. Wren, A. Azarbayejani, T. Darell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 780–785, 1997.