Bayesian Background Modeling for Foreground Detection

Fatih Porikli, Oncel Tuzel

TR2005-101 April 2006

Abstract

We propose a Bayesian learning method to capture the background statistics of a dynamic scene. We model each pixel as a set of layered normal distributions that compete with each other. Using a recursive Bayesian learning mechanism, we estimate not only the mean and variance but also the probability distribution of the mean and covariance of each model. This learning algorithm preserves the multimodality of the background process and is capable of estimating the number of required layers to represent each pixel.

ACM Visual Surveillance & Sensor Networks - VSSN 2005

Bayesian Background Modeling for Foreground Detection

Fatih Porikli
Mitsubishi Electric Research Labs

Oncel Tuzel
Rutgers University

Abstract

We propose a Bayesian learning method to capture the background statistics of a dynamic scene. We model each pixel as a set of layered normal distributions that compete with each other. Using a recursive Bayesian learning mechanism, we estimate not only the mean and variance but also the probability distribution of the mean and covariance of each model. This learning algorithm preserves the multimodality of the background process and is capable of estimating the number of required layers to represent each pixel.

1 Introduction

Segmentation of the moving regions, so called as foreground, from the static part of a scene, commonly named as background, is one of the most fundamental tasks in computer vision with a wide spectrum of applications from compression to scene understanding.

A simple approach for detecting foreground regions in stationary camera setups is to select a reference frame in which no target objects are visible, and subtract the observed frames from this reference image. 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 illumination changes, camera noise, shadows, etc.

Earlier methods applied simple prediction filters to adapt the background pixel intensities. In [7] Kalman filtering is used to model background dynamics. Similarly Wiener filter is used in [11] to make a linear prediction of the pixel intensity values, given the pixel histories. An alternative approach models the probability distribution of pixel intensities. This approach ignores the order in which observations are made and focuses on the distribution of the pixel intensities. In [12], a single Gaussian model is used per pixel and the parameters are updated by alpha blending. Unfortunately, these approaches fail in case the distribution of the background color values do not fit into a single model.

Mixture models were proposed to handle the backgrounds that exhibit multimodal characteristics. A mixture of three Gaussians corresponding to road, vehicle and shadow pixels are defined in [2] for a traffic surveillance application. Likewise, Stauffer and Grimson [10] proposed to update the model parameters of a mixture of k Gaussian distributions using an online Expectation Maximization (EM) algorithm. In [5] and [6] integration of gradient information is suggested as another feature of the multiple models. Although mixture of Gaussian models can converge to any arbitrary distribution provided enough number of components, this is computationally not feasible for real-time applications. Another approach that approximates the probability distribution of a multimodal background is the nonparametric kernel density estimation [1]. This method 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. In [9], motion information is used to model dynamic scenes. One major disadvantage of the nonparametric approaches is that they require large amount of memory to keep the previous measurements. Besides, they have very high computation complexities, which is proportional to the size of the temporal windows, making them infeasible.

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.

2 Bayesian Background

Our background model is most similar to adaptive mixture models [10] 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 in the RGB 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 [9].

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 unknown and modeled as random variables [3, p.87-88]. Using Bayes theorem joint posterior density can be written as:

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

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:

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

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

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

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

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

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

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:

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

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

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

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:

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

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

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

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

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

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

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

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

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

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

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

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:

$$\mathbf{\Sigma}_t = \left(\begin{array}{ccc} \sigma^2_{t,r} & 0 & 0 \\ 0 & \sigma^2_{t,g} & 0 \\ 0 & 0 & \sigma^2_{t,b} \end{array} \right). \tag{14}$$

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.

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 model. If current sample is not inside the confidence interval we update the number of prior measurements parameter:

$$\kappa_t = \kappa_{t-1} - n \tag{15}$$

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:
        $d_i \leftarrow (\mathbf{x} - \boldsymbol{\mu}_{t-1,i})^T \mathbf{\Sigma}^{-1}_{t-1,i} (\mathbf{x} - \boldsymbol{\mu}_{t-1,i})$.
        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.

2.3 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. Using the confidence scores we determine how many layers are significant for each pixel. We order 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. Finally, connected component analysis is performed on foreground pixels.

3 Shadow Classifier

Shadow classifier evaluates each foreground pixel and decides whether it is a shadow pixel or belongs to an object. To find foreground pixels, we measure the Mahalanobis distance between the the pixel color and the mean values of confident background layers.

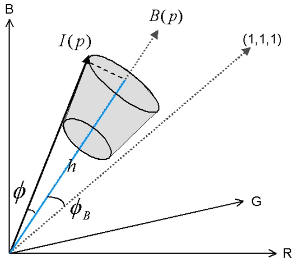

Figure 1. Weak shadow is defined as a conic volume around the corresponding background color of pixel.

Pixels that are outside of 99% confidence interval of all confident layers of the background are considered as foreground pixels.

First, we determine whether a pixel is a possible shadow pixel by evaluating the color variation as in [4]. We assume that shadow decreases the luminance and changes the saturation, yet it does not affect the hue. The projection of the color vector to the background color vector gives us the luminance change h

$$h = |I(p)| \cos \phi \tag{16}$$

where ϕ is the angle between the background $B_t^*(p)$ and $I_t(p)$. We define a luminance ratio as $r = |B_t^*(p)|/h$. We compute a second angle ϕ_B between the $B_t^*(p)$ and the white color $(1, 1, 1)$. For each possible foreground pixel obtained, we apply the following test and classify the pixel as a shadow pixel if it satisfies both of the conditions

$$\phi < \min(\phi_B, \phi_0) \;, \qquad r_1 < r < r_2 \tag{17}$$

where ϕ_0 is the maximum angle separation, $r_1 < r_2$ determines maximum allowed darkness and brightness respectively. Thus, we define shadow as a conic around the background color vector in the color space (Fig. 1). Those pixels that satisfy the above conditions are marked as possible shadow pixels, the rest remains as possible foreground.

At the second stage, we refine the shadow pixels by evaluating their local neighborhood. If the illumination ratio of two shadow pixels are not similar than they assigned as unclassified. Then, inside a window the number of foreground C, shadow S, and unclassified pixels U are counted for the center pixel, and following rules are applied iteratively: $(C > U) \wedge (C > S) \rightarrow C$, $(S > U) \wedge (S > C) \rightarrow S$, and else U. The shadow removal mechanism is proved to be effective and adjustable to the different lighting conditions.

		M-1	M-2	M-3	M-4	M-5
Real	Num.	10000	8000	3000	2000	
	Mean	0.200	0.600	0.300	0.800	
	Std.	0.015	0.030	0.050	0.050	
EM	Mean	0.203	0.203	0.599	0.599	0.938
	Std.	0.008	0.008	0.011	0.011	0.063
	Conf.	0.377	0.377	0.122	0.122	0.011
Bayes	Mean	0.200	0.599	0.302	0.800	0.938
	Std.	0.014	0.027	0.045	0.062	0.063
	Conf.	0.399	0.382	0.108	0.108	0.001

Table 1. Mixture of four Gaussians.

4 Comparison with Online EM

Although our model looks similar to Stauffer's GMM's [10], there are major differences. In GMM's, each pixel is represented as a mixture of Gaussian distribution and parameters of Gaussians and 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 threat 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 Fig. 2. 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.

Normalized confidence scores are shown in the bottom rows of each method in Table 1. 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 effec-

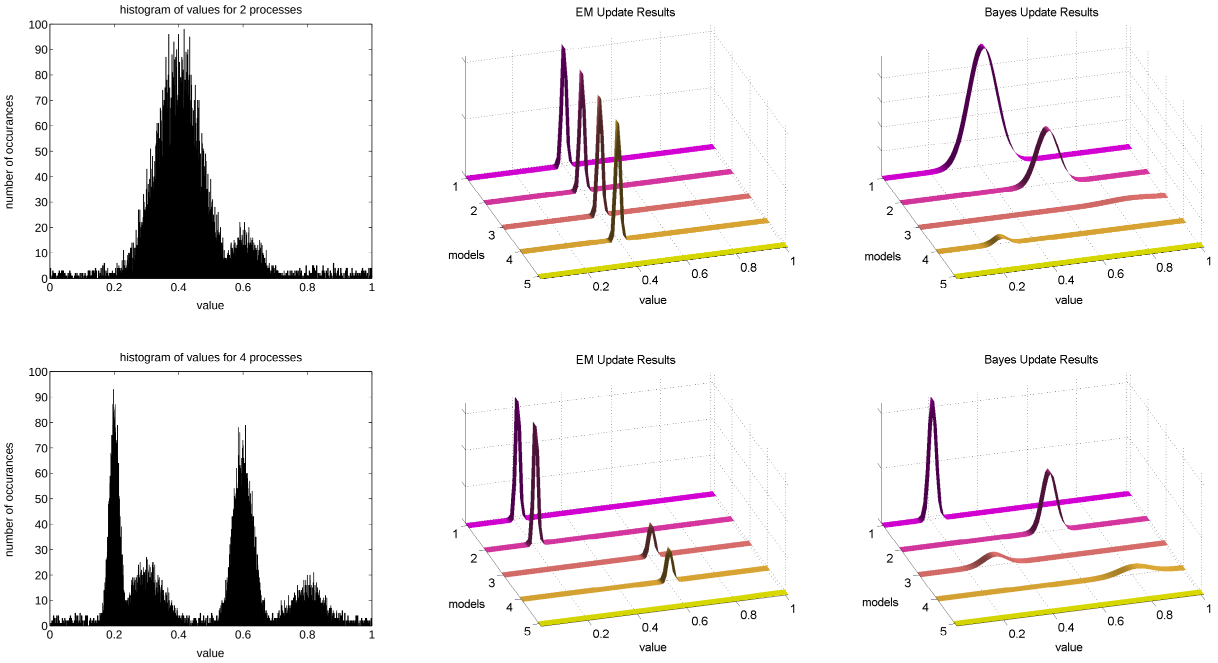

Figure 2. Left: Histograms of Gaussian data corrupted with uniform noise, **Middle:** Estimation results using conventional EM algorithm, **Right:** Using Bayesian update. As visible, EM fails to detect correct modes. (Upper row: 2-modes, lower row: 4-modes simulations)

tive. There is a big gap between the significant and redundant layers.

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

5 Performance Evaluation using VSSN 2005 Datasets

We made an initial evaluation of the proposed foreground-background detection method as given in Table 2 using the RGB color space (We observed that the accuracy does not change for the XYZ color space and it drops in case we use the HSV or Lab color spaces).

We computed four performance metrics, namely averages and maximums of false alarms and false misses, that are provided at the VSSN-Challenge web site. We also tested the implementation of the Li's method [8] as given at the same site.

		Bayesian	Li's
Video1	Ave False Alarm	4	3
	Ave False Miss.	244	858
	Max False Alarm	167	119
	Max False Miss.	856	3452
Video2	Ave False Alarm	0	3
	Ave False Miss.	186	361
	Max False Alarm	6	61
	Max False Miss.	786	1641
Video3	Ave False Alarm	261	282
	Ave False Miss.	381	385
	Max False Alarm	2135	4302
	Max False Miss.	1902	2049
Video4	Ave False Alarm	284	190
	Ave False Miss.	616	1007
	Max False Alarm	2192	3470
	Max False Miss.	1875	3632

Table 2. Detection Results using VSSN Datasets.

Our results show that the proposed method achieves much lower false alarms and false misses at the same time. Our maximum false alarms and false misses are also much lower than the Li's method.

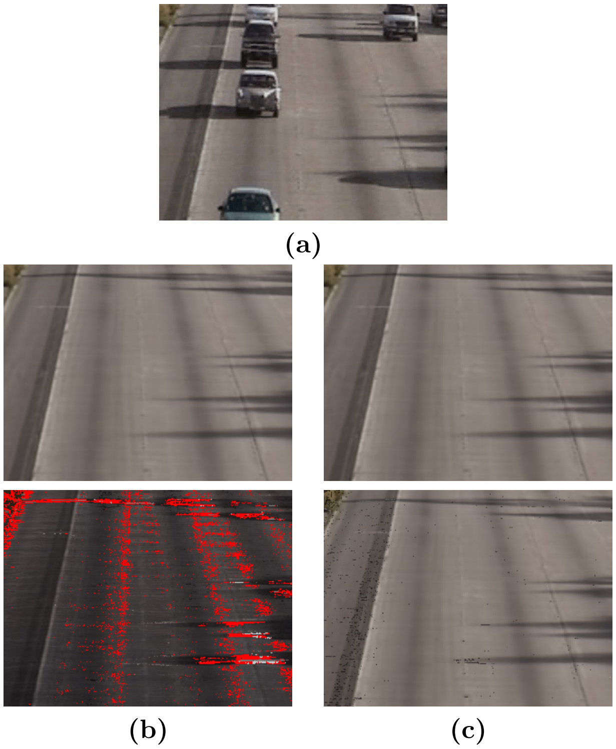

Figure 3. Traffic video with heavy shadows. (a) Original sequence. (b)Most confident two layers with recursive Bayesian learning. (c) Most confident two layers with online EM. With recursive Bayesian learning, we are able to model the shadows as the second layer of the scene whereas in EM first and second layers converge to most significant mode.

References

[1] 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.

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

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

[4] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *ICCV Frame-rate Workshop*, 1999.

[5] 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.

[6] 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.

[7] 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.

[8] L. Li, W. Huang, I. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *ACM Multimedia*, 2003.

[9] 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.

[10] 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.

[11] 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.

[12] 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.