Shadow Flow: A Recursive Method to Learn Moving Cast Shadows

Fatih Porikli
Mitsubishi Electric Research Lab
Cambridge, MA 02139, USA

Jay Thornton
Mitsubishi Electric Research Lab
Cambridge, MA 02139, USA

Abstract

We present a novel algorithm to detect and remove cast shadows in a video sequence by taking advantage of the statistical prevalence of the shadowed regions over the object regions. We model shadows using multivariate Gaussians. We apply a weak classifier as a pre-filter. We project shadow models into a quantized color space to update a shadow flow function. We use shadow flow, background models, and current frame to determine the shadow and object regions. This method has several advantages: It does not require a color space transformation. We pose the problem in the RGB color space, and we can carry out the same analysis in other Cartesian spaces as well. It is data-driven and adapts to the changing shadow conditions. In other words, accuracy of our method is not limited by the preset values. Furthermore, it does not assume any 3D models for the target objects or tracking of the cast shadows between frames. Our results show that the detection performance is superior than the benchmark method.

1. Introduction

Cast shadows poses one of the most challenging problems in many vision tasks, especially in object tracking, by distorting the true shape and color properties of the target objects. They correspond to the areas in the background scene that are blocked from the light source. It is essential to eliminate only cast shadows since removal of self shadows, which are the parts of the object that are not illuminated, will result in incomplete object silhouettes.

After all, what is a shadow? There are a number of cues that indicate the presence of a shadow. For instance, pixel luminance within the shadow regions decrease, when compared to the reference background. Shadows retain texture of the underlying surface under general viewing conditions, thus, the intensity reduction rate changes smoothly between neighboring pixels. Furthermore, it is also true that most shadow regions do not have strong edges [1]. Spatially, moving cast shadow regions should adjoin to the objects.

Most of the current shadow removal approaches are based on an assumption that the shadow pixels have the same chrominance as the background but are of lower luminance. For instance, Horprasert *et al.* [2], classify a pixel into one of the four categories depending on the distortion of the luminance and the amount of the chrominance of the difference. Stauder [3] provided a similar approach by verifying the above criteria by integrating a color model similar to Phong. Mikic *et al.* [4] classified pixels on the basis of a statistical method. Some approaches ([5]) prefer remapping of the color space since a shadow cast on a background does not change significantly its hue. Finlayson [6] has, in fact, pioneered recovery of an invariant image from a 3-band color image. The method devised finds an intrinsic reflectivity image based on assumptions of Lambertian reflectance, approximately Planckian lighting, and fairly narrowband camera sensors. Jiang *et al.* [7] also made use of an illumination invariant image. If lighting is approximately Planckian, then as the illumination color changes, a log-log plot of (R/G) and B/G) values for any single surface forms a straight line. Thus lighting change reduces to a linear transformation along an almost straight line. One of the approximately illumination invariant spaces devised by Gevers *et al.* [8] is first used to transform the color space. This color space is approximately invariant to shading and intensity changes, albeit only for matte surfaces under equi-energy white illumination.

Other approaches perform image segmentation. Javed *et al.* [9] divided the potential shadow region into sub regions. Each shadow candidate segment and its respective background the gradient's are correlated. If the correlation results in more than a threshold then the candidate segment is considered a cast shadow, and is removed from the foreground region. One obvious drawback is that not all images contain statistically significant amount of object surfaces corresponding to

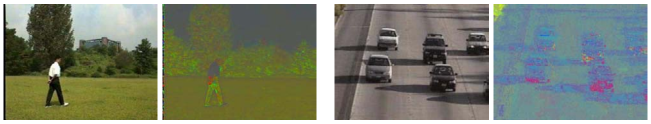

Figure 1. Normalization of color space cannot always remove shadows. Images are transferred from RGB to rgb. Shadow is removed and pedestrian can be detected (left), however shadows are amplified and vehicles have now similar colors as shadows (right).

both directly lit and shadowed pixels. Besides, the lighting color of the umbra region is not always the same as that of the sunshine. Sato [10] proposed a method to remove shadows using a measure of brightness. The image is segmented into several regions that have the same density and shadow regions are determined based on the brightness and the color. Baba [11] extended this idea by applying a maximum and minimum value filters, followed by a smoothing operator the image to get a global brightness of the image. From the global brightness, he calculate the shadow density. Salvador [12] presented a shadow segmentation algorithm which includes two stages. The first stage extracts moving cast shadows in each frame. The second stage tracks the extracted shadows in the subsequent frames. Obviously, the segmentation based approaches inherently degraded by the segmentation inaccuracies.

On the other hand, Zhao [13] proposed a geometrical approach by assuming the shadow of an ellipsoid on the ground. Any foreground pixel which lies in the shadow ellipse and whose intensity is lower than that of the corresponding pixel in the background by a threshold is classified as a shadow pixel.

Unfortunately, the assumptions of these approaches are difficult to justify in general. Detection based on the luminance based criteria will fail when pixels of foreground objects are darker than the background and have a uniform gain with respect to the reference surface they cover. Color space transformations are deficient if background color is gray as in Fig. 1. Geometrical shadow models depend heavily on the view-point and object shape. It is not possible to achieve robust shadow elimination for a wide spectrum of conditions with several predefined parameters. Another main limitation of these methods is that they do not adapt to different types of shadow, e.g. light (due to ambient light source), heavy (due to strong spot lights).

One key observation is that cast shadows constitute a "prevalent" change in such scenarios. In other words, color change at a pixel due to objects has higher variance (since objects may be in different colors) in comparison to the color change due to cast shadows. For a pixel, cast shadows cause identical background color change. However, color changes caused by object motion will not be same in case object colors are different, which is the usual case.

To address the disadvantages of the above techniques, we propose a recursive learning based method that models the color change induced by shadows in terms of multivariate Gaussians for surveillance settings. We take advantage of the above observation, and train models for shadow changes at each pixel in addition to the set of background models. We use multiple models, and these models are updated with each frame. Thus, if the lighting condition (as a result, the shadow properties) changes, these models will dynamically adapt themselves to the new condition.

We update the shadow models of a pixel if only that pixel is labeled as a shadow by a weak shadow classifier, which serves as a pre-filter for the global evaluation. Our data-driven method recursively adapts the shadow models. In other words, the accuracy of our method is not limited by the preset values, which was a major drawback of the existing approaches. The accuracy of our method improves as it processes frames that contain shadows. Furthermore, our method does not require tracking of shadow regions. Our method requires no special color space; we pose the problem in the RGB color space, and we can carry out the same analysis in other Cartesian spaces as well. Besides, no 3D models of the target objects is necessary.

In the following section, we explain the weak classifier and model update mechanism. In section 3, we present experimental results that show the effectiveness of the proposed method.

2. Learning Cast Shadows

A flow diagram of the proposed method is shown in Fig. 2. Let the color of the current image be $I_t(p)$, where p is a pixel. We train two sets of models: background $B_t^n(p)$ and shadow $S_t^m(x)$, where n and m are number of models for the background and shadow, respectively. We perform our operations in the RGB color space.

First, we estimate a reference model (background) for the stationary part of the scene and compare the current frame with the background to determine the changed regions (foreground) in the image, which may contain both objects and their moving cast shadows. We use background subtraction that is the most common approach for discriminating a moving object in a *relatively* static scene to find foreground pixels

As we explain in the next section, we define each

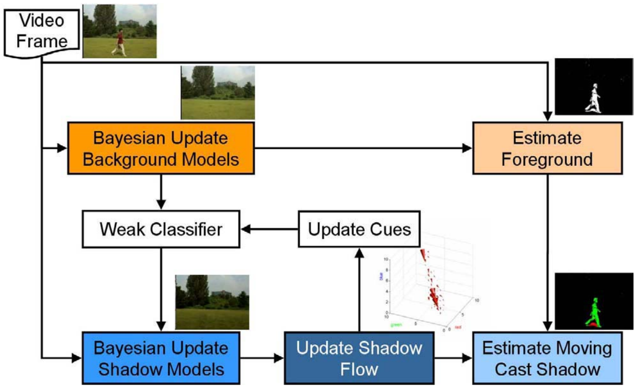
Video Frame
Bayesian Update Background Models
Estimate Foreground
Weak Classifier
Update Cues
Bayesian Update Shadow Models
Update Shadow Flow
Estimate Moving Cast Shadow

Figure 2. Each frame updates the background models. Using the pixels detected as shadow by the weak classifier, we refine the shadow models and compute shadow flow, which will then steer the weak classifier.

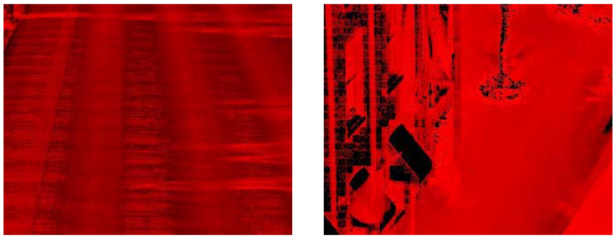

Figure 3. Shadow model confidence.

pixel as layers of multivariate Gaussians. We refer "layer" as the set of ordered models of all pixels with respect to model condifences. Each layer corresponds to a different appearance of the pixels. The most consistent of these layers constitute the background. To find foreground pixels, we compare the current image to the background layers.

We apply a weak shadow classifier that evaluates the color and spatial changes the foreground pixels undergo. This classifier basically defines a range of possible colors with respect to the background color, and iteratively updates pixel labels using a local consistency of given labels. If a pixel p is detected as foreground and its color is in the weak classifier's range then we update the shadow models $S_t^m(p)$ of the pixel p as illustrated in Fig. 5.

Bayesian update computes a confidence score for each shadow model. We determine the most confident shadow model $S_t^*(p)$ for each pixel. We compare the most confident shadow model $S_t^*(p)$ with the most confident background model $B_t^*(p)$ and compute a disparity vector $S_t^*(x, \boldsymbol{\mu}) - B_t^*(p, \boldsymbol{\mu})$, where $\boldsymbol{\mu}$ is the mean of the model.

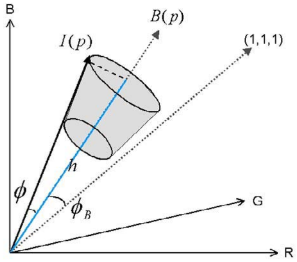

Figure 4. Weak shadow is defined as a conic volume around the corresponding background color of pixel.

We project the disparity vectors into the quantized color space, in which each bin represents a color value visible in the background. Note that, more than one disparity vectors may be assigned to each bin since there may be same color pixels with different disparities in the background. We aggregate disparity vectors weighted by the model confidences (Fig. 3) and compute their mean $F_t(c, \boldsymbol{\mu})$ and variance $F_t(c, \sigma)$ at each bin c to obtain a shadow flow $F_t(c, \boldsymbol{\mu})$ as shown in Fig. 7. Note that, shadow flow vectors for different shadow types and different backgrounds are different. This is another reason why the shadow elimination should be data-drive. The shadow projection projection also enables us to remove the inconsistent or erroneous shadow detection results.

To find the shadow pixels in the foreground, we back-project the shadow flow $F_t(c)$ to a shadow image $SI_t(p) = F_t(B_t^*(p), \boldsymbol{\mu})$ using the background image (Fig. 6). Finally, we compare the current foreground pixels and the shadow image to determine the shadow pixels using the shadow color variance $F_t(c, \sigma)$. Figure 8 shows a single shadow bin, the corresponding shadow flow vector (blue) and color changes for the current image. It is possible to use the shadow flow to determine the quality of foreground estimation.

This refinement process continues for the next image in the video sequence.

2.1 Weak Shadow Classifier

Weak shadow classifier evaluates each foreground pixel and decides whether it is a shadow pixel or belongs to an object. Here, we do not make a final decision; but we select pixels that will be used to update the multivariate shadow models in the next step.

To find foreground pixels, we measure the Mahalanobis distance between the the pixel color and the mean values of confident background layers. Pixels

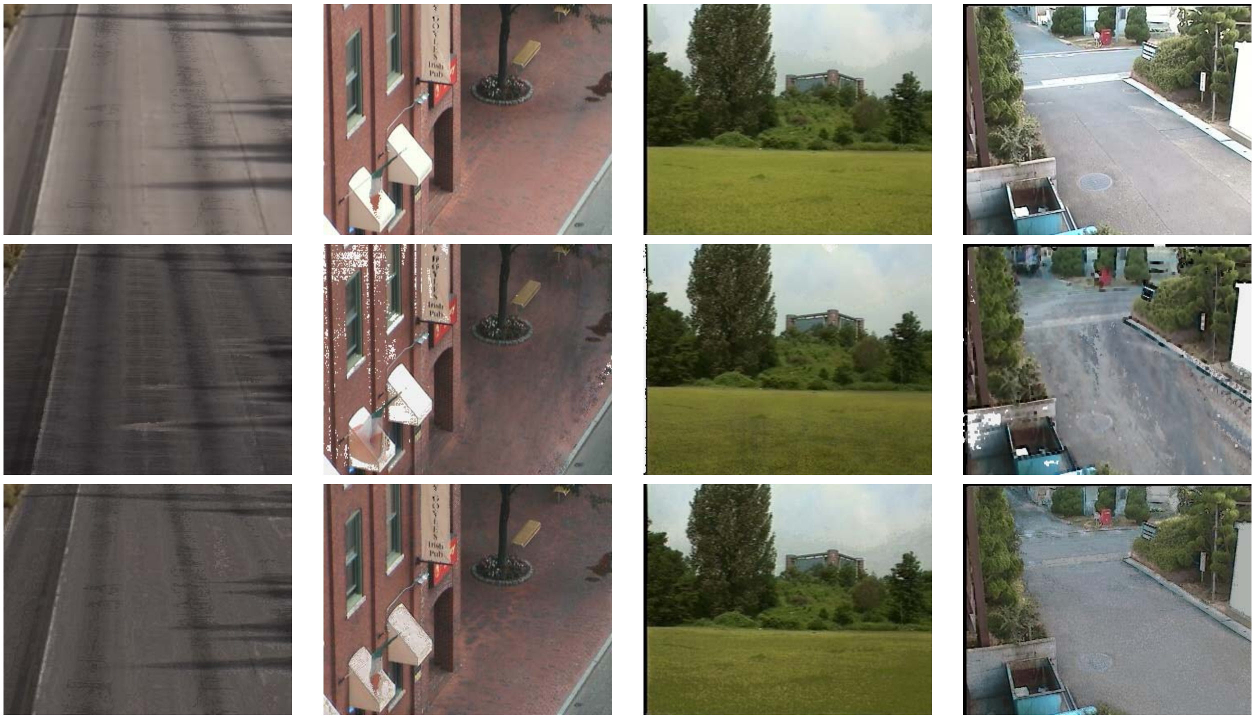

Figure 6. **First row:** Most confident background layer B_t^*, **second row:** most confident shadow layer S_t^*, **third row:** shadow image SI_t, .

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 [2]. 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

$$h = |I(p)| \cos \phi \qquad (1)$$

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

$$\phi < \min(\phi_B, \phi_0) \ , \qquad r_1 < r < r_2 \qquad (2)$$

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. 4). Those pixels that satisfy the above conditions are marked as possible shadow pixels, the rest remains as possible foreground.

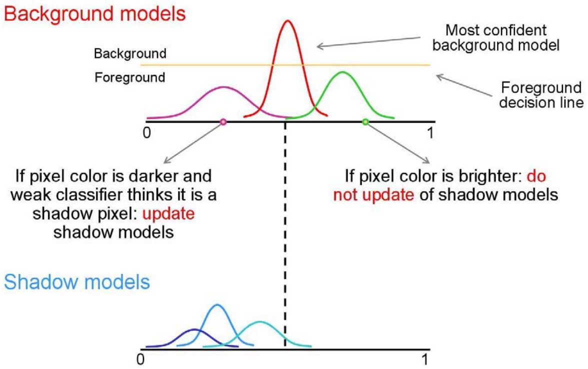
Background models
Most confident background model
Background
Foreground
Foreground decision line
0
1
If pixel color is darker and weak classifier thinks it is a shadow pixel: update shadow models
If pixel color is brighter: do not update of shadow models
Shadow models
0
1

Figure 5. A simple illustration of background and shadow model update in 1D. We update the shadow models if only the pixel is classified as shadow by the weak classifier.

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

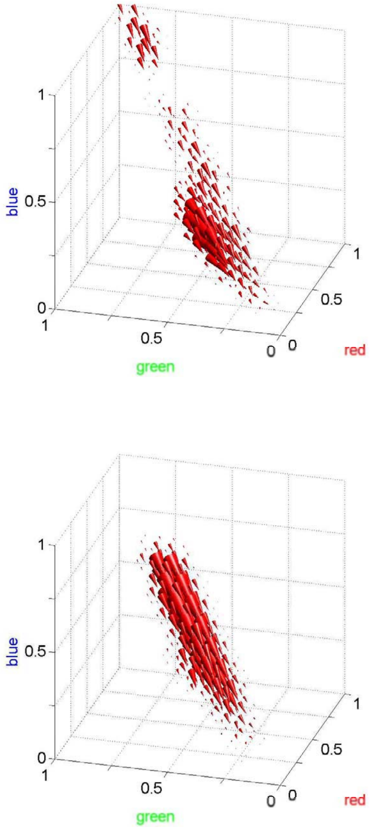

Figure 7. Shadow flows of two different sequences.

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.

Using the shadow flow, we adapt the above parameters of the weak classifier using simple α-blending after we aggregate the flow vectors for all color bins.

After we select the shadow pixels, we refine our multivariate shadow models using a Bayesian update technique that is explained in the next section. Note that, we use the same update mechanism for the background models as well. With this mechanism, we do not deform our models with noise or foreground pixels, but easily adapt to smooth intensity changes. Embedded confidence score determines the number of layers to be used and prevents unnecessary layers. 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.

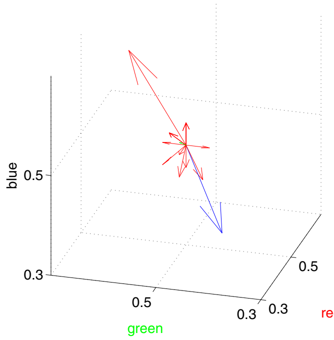

Figure 8. Shadow flow (blue) and current image color change vectors (red) corresponding to a single color bin of the background image.

2.2 Bayesian Update

Instead of a “mixture” of Gaussian distributions, we define each pixel as “competitive” layers of multivariate Gaussians. Each layer corresponds to a different appearance of the pixel. 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. Mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ of the pixel color history are assumed unknown and modeled as normally distributed random variables. 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:

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\boldsymbol{\mu}|\boldsymbol{\Sigma})p(\boldsymbol{\Sigma}). \tag{3}$$

Above condition is realized if we assume inverse Wishart distribution [15] 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

$$\boldsymbol{\Sigma} \sim \text{Inv-Wishart}_{\upsilon_{t-1}}(\boldsymbol{\Lambda}_{t-1}^{-1}) \tag{4}$$

$$\boldsymbol{\mu}|\boldsymbol{\Sigma} \sim \mathrm{N}(\boldsymbol{\theta}_{t-1}, \boldsymbol{\Sigma}/\kappa_{t-1}). \tag{5}$$

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 normal inverse-Wishart $(\boldsymbol{\theta}_t, \boldsymbol{\Lambda}_t/\kappa_t; \upsilon_t, \boldsymbol{\Lambda}_t)$ with the parameters updated:

$$\begin{aligned} \upsilon_t &= \upsilon_{t-1} + n \quad \kappa_n = \kappa_{t-1} + n & (6) \\ \boldsymbol{\theta}_t &= \boldsymbol{\theta}_{t-1}\frac{\kappa_{t-1}}{\kappa_{t-1}+n} + \overline{\mathbf{x}}\frac{n}{\kappa_{t-1}+n} & (7) \\ \boldsymbol{\Lambda}_t &= \boldsymbol{\Lambda}_{t-1} + \sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T + \\ & \quad n\frac{\kappa_{t-1}}{\kappa_t}(\overline{\mathbf{x}} - \boldsymbol{\theta}_{t-1})(\overline{\mathbf{x}} - \boldsymbol{\theta}_{t-1})^T & (8) \end{aligned}$$

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.

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:

$$\boldsymbol{\mu}_t = E(\boldsymbol{\mu}|\mathbf{X}) = \boldsymbol{\theta}_t \quad (9)$$

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

$$\boldsymbol{\Sigma}_t = E(\boldsymbol{\Sigma}|\mathbf{X}) = (\upsilon_t - 4)^{-1}\boldsymbol{\Lambda}_t. \quad (10)$$

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

$$C = \frac{1}{|\boldsymbol{\Sigma}_{\boldsymbol{\mu}|\mathbf{X}}|} = \frac{\kappa_t^3(\upsilon_t - 2)^4}{(\upsilon_t - 4)|\boldsymbol{\Lambda}_t|}. \quad (11)$$

If our marginal posterior mean has larger variance, our model becomes less confident. Note that variance of multivariate t-distribution with scale matrix $\boldsymbol{\Sigma}$ and degrees of freedom υ is equal to $\frac{\upsilon}{\upsilon-2}\boldsymbol{\Sigma}$ for $\upsilon > 2$. Instead of multivariate Gaussian for a single layer, it is possible to use three univariate Gaussians corresponding to each color channel. 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.

We initialize our system with k layers for each pixel. Usually we select three-five layers (for both background and shadow). 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. It is also useful to have a depletion mechanism so that the earlier observations have less effect on the model. Depletion 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 $\kappa_t = \kappa_{t-1} - n$ and proceed with the update of next confident layer. We do not let κ_t become very small. If none of the models are updated, we delete the least confident layer and initialize a new model with the current sample.

2.3 Comparison with Online EM

Although our model looks similar to Stauffer's GMM's [14], 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. 9. 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.

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.

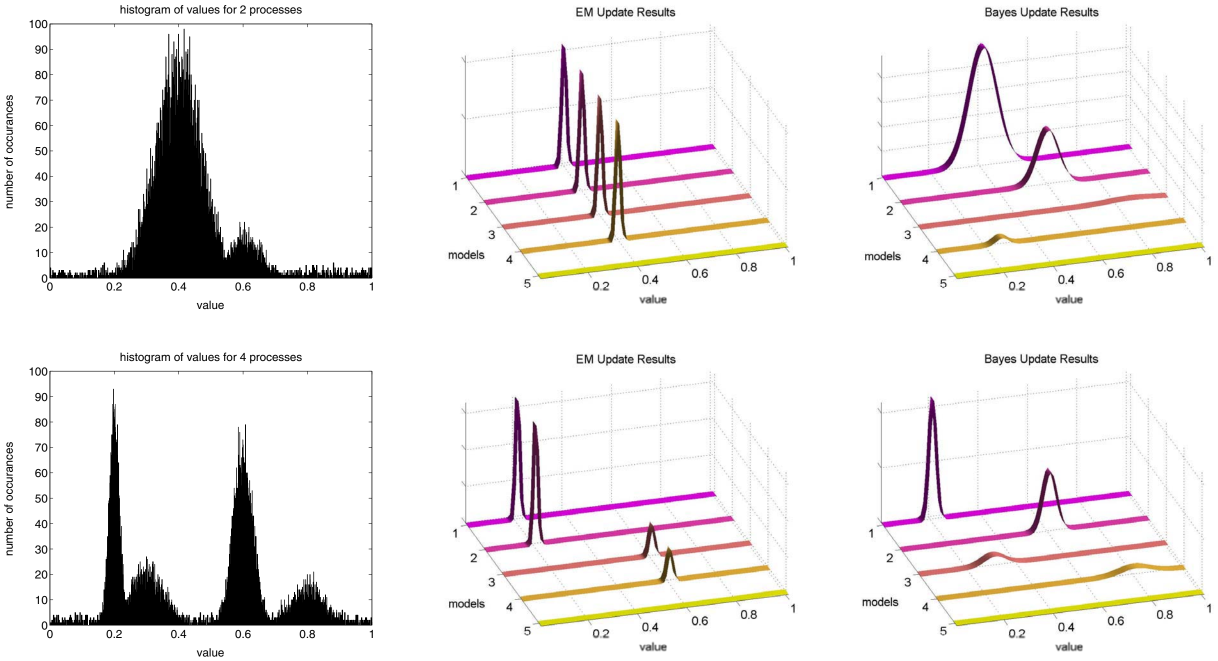

Figure 9. **Left:** Histograms of Gaussian data corrupted with uniform noise, **Middle:** Estimation results using conventional EM algorithm, **Right:** Using Bayesian update. As visible, EM fails to detect correct modes. (Upper row: 2-modes, lower row: 4-modes simulations)

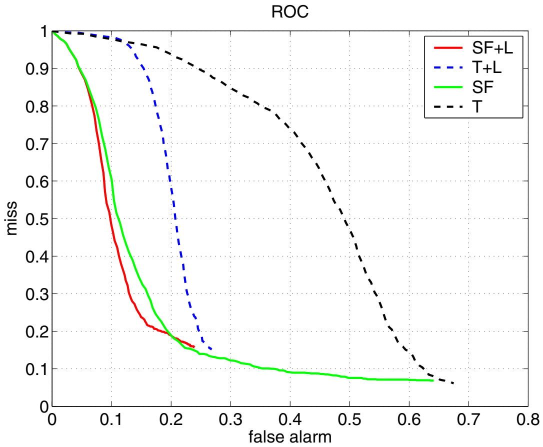

Figure 10. Ratio of foreground pixels detected as shadow (false-alarm) to ratio of misclassified shadow pixels (miss). Lower left of the chart is the desired region (low false-alarms, low misses). SF: proposed shadow flow, SF+L, with luminance clipping, T is a benchmark method [2], and T+L its clipped version.

3. Experimental Results

We tested the proposes method with several real data sequences that contains heavy to light shadow conditions (to name a few; *traffic* - heavy - 440 frames, *corner* - light- 7200 frames, *green* - varying - 3100 frames, *street* - light- 8000 frames).

We used 5 Gaussian models for the background, however, we assumed the color channels (RGB) are independent and covariance matrix is diagonal. For the shadow, we assigned 3 models. We manually marked the shadow regions in these sequences to generate ground truth data, which was a tedious task. We quantize the background color space using 64-bins for each color channel (a total of 2^{18} bins). The presented performance scores is similar down to 16-bins, however, it drops for more severe quantizations.

Using the ground truth we evaluated the performance of our algorithm and another algorithm presented in [2] since this algorithm also uses background images. Note that, our algorithm does not require training with ground truth. In Fig. 10 we give the detection performance, which is obtained by changing various parameters. We also integrated an additional heuristic (luminance clipping) for the bench-

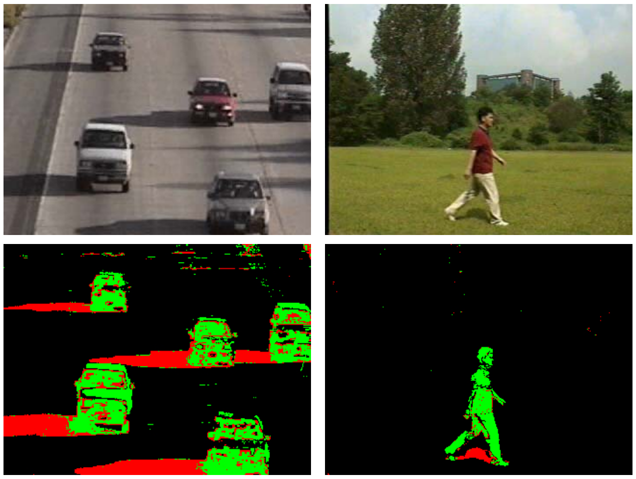

Figure 11. Sample detection results without any filtering (red: shadow, green: foreground).

mark result since its performance for the above dataset was comparatively poor. The graph SF (green) and SF+L shows the ROC curves for the proposed shadow flow method, and T, T+L are for the the benchmark method. As visible, the proposed algorithm doubled the detection accuracy from 20% false-alarms on average to 10% on average for most miss ratios. We also want to point out that the benchmark method has a hard limit, i.e. it can not decrease the false alarms less than 23% at the equal error rate. On the other hand, shadow flow method achieves 15% for most of the parameter assignments, and it always out performs the benchmark method. In Fig. 11 we show sample shadow detection results without clipping without any filtering or morphological operations.

4. Conclusion

In this paper, we propose a shadow removal algorithm for surveillance scenarios. The main contribution of this work is an adaptive shadow flow method that learns the properties of cast shadows automatically by using multivariate Gaussians. We also present an accurate model update mechanism.

This method has several advantages: 1) It does not require a color space transformation. We pose the problem in the *RGB* color space, and we can carry out the same analysis in other Cartesian spaces as well. 2) Our data-driven method dynamically adapts the shadow models to the changing shadow conditions. In other words, accuracy of our method is not limited by the preset threshold values, which is a major drawback of the existing approaches. The accuracy of this method improves as it process more video frames. 3) Furthermore, it does not assume any 3D models for the target objects or tracking of the cast shadows between frames.

Our results prove that the shadow flow doubles the detection accuracy by consistently decreasing the percentage of false-alarms.

References

[1] H.T. Chen, H.H. Lin, T.L. Liu, “Multi-object tracking using dynamical graph matching”, *CVPR*, 2001

[2] T. Horprasert, D. Harwood, and L. Davis, “A statistical approach for real-time robust background subt. and shadow detection”, *ICCV Frame-rate Workshop*, 1999

[3] J. Stauder, R. Mech, and J. Ostermann, “Detection of moving cast shadows for object segmentation”, *IEEE Transactions on Multimedia*, 1999

[4] I. Mikic, P. Cosman, G. Kogut, and M. Trivedi, “Moving shadow and object detection in traffic scenes”, *ICPR*, 2000

[5] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting objects, shadows and ghosts in video streams by exploiting color and motion information”, *CIAP*, 2001

[6] G. D. Finlayson, M.Drew, and C. Lu, “Intrinsic images by entropy minimization”, *ECCV*, 2004

[7] H. Jiang, M. Drew, “Shadow resistant tracking in video”, *ICME*, 2003

[8] T. Gevers and A.W. Smeulders, “Color-based object recognition”, *Patt. Rec.*, vol. 32, pp. 453464, 1999

[9] O. Javed and M. Shah, “Tracking and object classification for automated surveillance”, *ECCV*, 2002

[10] I. Sato and K. Ikeuchi, “Illumination distribution from brightness in shadows”, *ICCV* (2), 875-882, 1999

[11] M. Baba and N. Asada, “Shadow removal from a real picture”, *SIGGRAPH Conference on Sketches and Applications*, 2003

[12] E. Salvador, A. Cavallaro, and T. Ebrahimi, “Shadow identification and classification using invariant color models”, *ICASSP*, 2001

[13] T. Zhao, R. Nevatia, “Tracking multiple humans in complex situations”, *PAMI*, vol.26, no.9, 2004

[14] C. Stauffer and W.Grimson, “Adaptive background mixture models for real-time tracking”, *CVPR*, 1999

[15] A.Gelman, J.Carlin, H.Stern, D.Rubin, “Bayesian Data Analysis”, *Chapman and Hall Publications*, 2003