Statistical Background Subtraction Using Spatial Cues

Pierre-Marc Jodoin, Max Mignotte, and Janusz Konrad, *Member, IEEE*

***Abstract*—Most statistical background subtraction techniques are based on the analysis of temporal color/intensity distribution. However, learning statistics on a series of time frames can be problematic, especially when no frame absent of moving objects is available or when the available memory is not sufficient to store the series of frames needed for learning. In this letter, we propose a *spatial* variation to the traditional *temporal* framework. The proposed framework allows statistical motion detection with methods trained on one background frame instead of a series of frames as is usually the case. Our framework includes two *spatial* background subtraction approaches suitable for different applications. The first approach is meant for scenes having a nonstatic background due to noise, camera jitter or animation in the scene (e.g.,waving trees, fluttering leaves). This approach models each pixel with two PDFs: one unimodal PDF and one multimodal PDF, both trained on one background frame. In this way, the method can handle backgrounds with static and nonstatic areas. The second *spatial* approach is designed to use as little processing time and memory as possible. Based on the assumption that neighboring pixels often share similar temporal distribution, this second approach models the background with one global mixture of $\mathcal{M}$ Gaussians.**

***Index Terms*—Background detection, motion detection.**

I. Introduction

MOTION detection methods are used to locate the presence (or absence) of motion in a given animated scene. A specific class of motion detection methods is the one meant for video surveillance [1], traffic monitoring [2], and various commercial applications [3] using a static camera. Because the background is constant in time, a class of solutions that enjoys tremendous popularity is the family of background subtraction methods [4]. These methods are based on the assumption that the animated objects to be segmented have different visual characteristics than the static background. As the name suggests, the most intuitive background subtraction method involves one background image and an animated sequence containing moving objects [4]. In this case, motion is detected by simply thresholding the intensity/color difference between a frame at time t and the background image [5], [6]. Although the threshold can be estimated on the fly [7], [8] and locally adapted [9], these methods are sensitive to phenomena that violate the basic assumptions of background subtraction. For instance, noise induced by a low-quality camera or jitter caused by an unstable camera are typical situations that can't be handled properly by simplistic background subtraction methods. There are also numerous situations in which some background objects are not perfectly static and induce local false positives. This is the case, for instance, when a tree is shaken by the wind or when the background includes animated texture such as wavy water. Another common source of background variations is when the global illumination is not constant in time and alters the appearance of the background. Such variation can be gradual, for example, when a cloud occludes the sun, or sudden, when a light is turned on or off.

For all these scenarios, a more elaborate background subtraction strategy is required. In this perspective, many authors propose modeling each background pixel with a probability density function (PDF) learned over a series of training frames. In this case, the motion detection problem often becomes a PDF-thresholding problem. For instance, to account for noise, some authors [10], [11] use a Gaussian distribution for each pixel. Also, to account for unstable backgrounds, multimodal PDF models have been proposed such as a mixture of Gaussians (MoG) [2], [12], [13], kernel-based methods [1], [14], and predictive methods [15], [16]. Let us also mention that several block-based methods [17], codebook methods [18], and statistical Markovian methods [19], to name a few, have been proposed.

The main limitation of most traditional statistical solutions is their need for a series of training frames absent of moving objects. Without these training frames, a nontrivial outlier detection method needs to be implemented [12]. Another limitation of these methods is the amount of memory some require. For example, in [2], every training frame needs to be stored in memory to estimate the MoG parameters and, for some nonparametric methods [1], [14] many frames need to be kept in memory during runtime which may be costly memory-wise.

In this letter, we investigate a framework that uses one background frame to train statistical models (be they parametric or not). Our aim for such a framework is fourfold:

1) reduce the number of frames (thus the amount of memory) needed during the training period;
2) reduce the amount of memory required during the detection phase;
3) better deal with sequences having no training frames absent of moving objects;
4) investigate how speed and memory can be improved in the context of statistical background subtraction.

II. Motivation

Two kinds of background variation may be identified. The first one is variations due to noise typically induced by a low-quality camera or by an environmental phenomenon, such as

Manuscript received August 2, 2006; revised December 13, 2006 and January 30, 2007. This paper was recommended by Associate Editor C. N. Taylor.

P.-M. Jodoin is with the Université de Sherbrooke, Département d'Informatique, Sherbrooke, QC J1K 2R1, Canada (e-mail: Pierre-Marc.Jodoin@USherbrooke.ca).

M. Mignotte is with the Université de Montréal, Département d'Informatique et de Recherche Opérationnelle, Montréal, QC H3C 3J7 Canada (e-mail: mignotte@iro.umontreal.ca).

J. Konrad is with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA (e-mail: jkonrad@bu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2007.906935

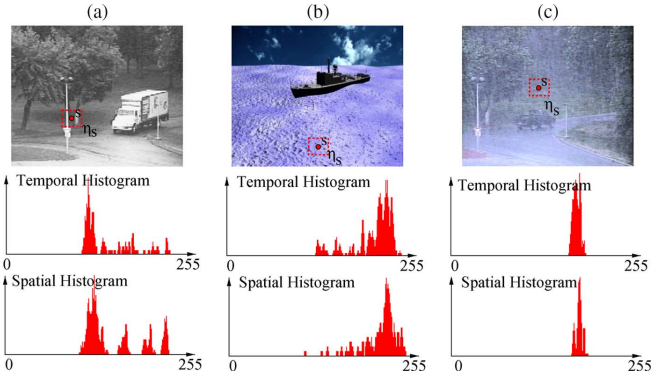

Fig. 1. Sample frame (top row), temporal histogram (middle row) and spatial histogram (bottom row) for three test sequences: "Shaky camera" captured with unstable camera (left column), "Boat" that includes animated texture modeling water (middle column) and "Rain" that is an example of severe noise (right column). The spatial histograms were calculated from 11 × 11 neighborhoods while the temporal histograms were calculated from 90, 200, and 100 frames, respectively. Note a good correspondence between spatial and temporal histograms.

rain [see Fig. 1(c)]. In this case, the background B is considered static and the intensity/color of a pixel at site s and discrete time t is defined as $\vec{B}_s^t = \vec{B}_s + n$ where n is an uncorrelated noise variable and $\vec{B}_s$ is the *ideal* noise-free background color (or intensity) at site s. In this case, the temporal distribution $P(\vec{B}_s^t)$ is considered unimodal with expectation $\vec{B}_s$. The second kind of variations is due to background movements caused, for example, by an animated texture or by camera jitter. In this case, the temporal distribution $P(\vec{B}_s^t)$ is often multimodal. But whether the temporal distribution is unimodal or multimodal, the goal is to estimate at every time instant t a *label field* L^t (sometimes called a *motion mask*) containing the motion status of each site s (typically, $L_s^t = 0$ when s is motionless and $L_s^t = 1$ otherwise). Since each background pixel is modeled by PDF $P(\vec{B}_s^t)$, the detection criterion can be formulated as:

$$L_s^t = \begin{cases} 0, & \text{if } P\left(\vec{I}_s^t\right) > \tau \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

where I^t the intensity of examined frame at time t and τ is a predetermined threshold.

In this letter, the goal is to use the same thresholding procedure and the same distribution functions proposed in the literature, but with different training data gathered inside a single frame instead of a series of frames as is usually the case. The use of spatial data is motivated by the fact that the color distribution of a given pixel observed over a certain period of time is also frequently observed spatially around that same pixel. For example, in the case of background movements, considering that variations are due to local movements, it can be assumed that the distribution of $\vec{B}_s^t$ in time is similar to a spatial distribution around s, i.e., $\{\vec{B}_r, \forall r \in \eta_s\}$ where η_s is a $M \times M$ neighborhood centered on s. Fig. 1(a)–(b) shows that when a site s is locally animated, the color/intensity distribution observed over a certain period of time often corresponds to the distribution observed locally around s. As shown in Fig. 1(c), the same observation may also hold for unimodal distribution. In the next sections, two *spatial* approaches will be presented. The first one is designed to be robust to background movements while the second one minimizes processing complexity and memory usage during runtime.

III. Robust Method

As we mentioned previously, background variation may be unimodal or multimodal. Unfortunately, with only one training frame, determining whether the distribution of a site s in time is unimodal or multimodal is an ill-posed problem. In order to overcome this, we use a *decision fusion* strategy. To this end, each pixel is modeled by two PDFs: one unimodal PDF (that we call P^u) and one multimodal PDF (called P^m), both being trained on one background frame B (see Section III-A for more details on training). Having each pixel modeled by two PDFs, the detection criterion may be formulated as follows:

$$L_s^t = \begin{cases} 0, & \text{if } P^u\left(\vec{I}_s^t\right) > \tau \text{ OR } P^m\left(\vec{I}_s^t\right) > \tau \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

Estimating L^t with this equation turns out to be the same as blending two label fields resulting from thresholding $P^u(\vec{I}_s^t)$ and $P^m(\vec{I}_s^t)$ separately. The reader should be aware that this fusion strategy is meant for sequences whose content is unknown *a priori*. However, if the scene is known to have either a perfectly static background or camera jitter, only P^u or P^m may be thresholded.

A. Spatial Training

In this section, we present how P^u and P^m can be trained on data gathered inside a single background training frame B.

1) Single Gaussian: As mentioned in Section I, $P^u(\vec{B}_s^t)$ can be modeled by a single Gaussian distribution

$$\frac{1}{(2\pi)^{d/2}|\Sigma_s|^{1/2}} \exp\left(-\frac{1}{2}\left(\vec{B}_s^t - \vec{\mu}_s\right)^{\mathrm{T}} \Sigma_s^{-1} \left(\vec{B}_s^t - \vec{\mu}_s\right)\right) \tag{3}$$

where $d = 1$ for grayscale sequences and $d = 3$ for color sequences. Note that for color sequences, Σ_s is a 3×3 variance-covariance matrix that, as suggested by Stauffer and Grimson [12], is assumed to be diagonal for efficiency. Since only one training frame is available in this framework, $\vec{\mu}_s$ and Σ_s are estimated with data gathered inside a spatial neighborhood η_s centered on site s. Of course, the spatial data should be drawn from a unimodal distribution that resembles the one observed temporally. Although some spatial neighborhoods of a scene have that kind of unimodal distribution (neighborhood A in Fig. 2), others are obviously multimodal (neighborhood C in Fig. 2) and cannot be used for training as is. Thus, to prevent $\vec{\mu}_s$ and Σ_s from being corrupted by outliers (such as gray pixels of the pavement near C in Fig. 2), a robust function ρ is used to weigh the influence of each sample $\vec{B}_r$ [20]. More specifically, the parameter estimation can be expressed as

$$\vec{\mu}_s(j) = \frac{1}{\sum_{r\in\eta_s} \rho_{s,r}} \sum_{r\in\eta_s} \rho_{s,r} \vec{B}_r(j) \tag{4}$$

$$\Sigma_s(j) = \frac{1}{\sum_{r\in\eta_s} \rho_{s,r}} \sum_{r\in\eta_s} \rho_{s,r} \left(\vec{B}_r(j) - \vec{\mu}_s(j)\right)^2 \tag{5}$$

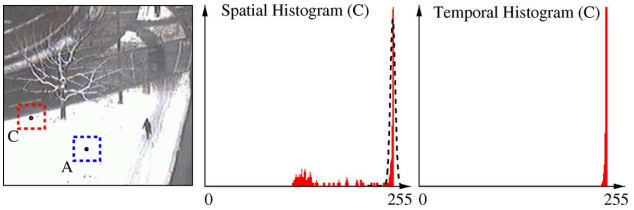

Fig. 2. Sample frame from a sequence captured by a perfectly static camera. While the temporal intensity distribution of pixel C is unimodal and centered at intensity 254, the spatial intensity distribution around C is bimodal. Estimating the Gaussian parameters with (4) and (5) leads to a distribution (in black) centered on the main mode, uncorrupted by the gray levels of the street.

where $j \in \{1,\ldots,d\}$, $\Sigma_s(j)$ is the j^{th} variance. As suggested by Huber [20], we use

$$\rho(s,r) = \begin{cases} 1, & \text{if } \|\vec{B}_s - \vec{B}_r\|_2 \leq c \\ \frac{c}{\|\vec{B}_s - \vec{B}_r\|_2}, & \text{otherwise} \end{cases} \quad (6)$$

where c is a constant and $\|.\|_2$ denotes the Euclidean norm. Clearly, with (6), as intensity/color of pixels in η_s deviate from those of the central pixel at s, their contribution to the mean and variance estimates diminishes. Note that global illumination variations can be compensated by updating $\vec{\mu}_s$ and Σ_s at every time t [10], [11] as follows:

$$\vec{\mu}_s(j) \leftarrow (1-\alpha)\vec{\mu}_s(j) + \alpha\vec{I}_s^t(j) \quad (7)$$

$$\Sigma_s(j) \leftarrow (1-\alpha)\Sigma_s(j) + \alpha\left(\vec{I}_s^t(j) - \vec{\mu}_s(j)\right)^2 \quad (8)$$

for all s such that $L_s^t = 0$, $j \in \{1,\ldots,d\}$, and $\alpha \in [0,1[$ is the so-called *learning rate* [2].

2) Mixture of Gaussians: A mixture of K Gaussians is used to model multimodal histograms such as those in Fig. 1(a)-(b)

$$P^m\left(\vec{I}_s^t\right) = \frac{1}{\sum_i^K w_{s,i}} \sum_{i=1}^{K} w_{s,i}\mathcal{N}\left(\vec{I}_s^t, \vec{\mu}_{s,i}, \Sigma_{s,i}\right) \quad (9)$$

where $\mathcal{N}(.)$ is a Gaussian similar to the one of (3), $w_{s,i}$ is the weight of the i^{th} Gaussian and K is the total number of Gaussians (between 2 and 5 typically). Thus, $P^m(\vec{I}_s^t)$ has $3 \times K$ parameters to be estimated: $w_{s,i}$, $\vec{\mu}_{s,i}$, and $\Sigma_{s,i}$. To do so, we use the well-known K-means algorithm. In our framework, K-means takes as input for each pixel s, the $M \times M$ background pixels $\{\vec{B}_r | r \in \eta_s\}$ contained within the square neighborhood η_s. When the algorithm converges, the mean $\vec{\mu}_{s,i}$ of every cluster is known and every pixel $r \in \eta_s$ has been assigned a class label $i \in [0,K[$. The covariance matrix $\Sigma_{s,i}$ and the weight $w_{s,i}$ of each class i can then be estimated from the pixels $r \in \eta_s$ having label i. Note that the EM [21] algorithm could eventually be used to refine these estimates.

As we mentioned for P^u, the MoG parameters can be updated at every frame to account for illumination variations. As suggested by Stauffer and Grimson [12], at each time frame I^t, parameters of the Gaussian that matches the observation $\vec{I}_s^t$ can be updated as follows:

$$\vec{\mu}_{s,i}(j) \leftarrow (1-\alpha)\vec{\mu}_{s,i}(j) + \alpha\vec{I}_s^t(j) \quad (10)$$

$$\Sigma_{s,i}(j) \leftarrow (1-\alpha)\Sigma_{s,i}(j) + \alpha\left(\vec{I}_s^t(j) - \vec{\mu}_{s,i}\right)^2 \quad (11)$$

$$w_{s,i} \leftarrow (1-\alpha)w_{s,i} + \alpha\xi_{s,i} \quad (12)$$

for all s such that $L_s^t = 0$, $j \in \{1,\ldots,d\}$, where $\xi_{s,i}$ is 1 for the Gaussian that matches and 0 for the other models.

3) Nonparametric Density Estimation: An unstructured approach can also be used to model multimodal distributions. We propose a kernel-based density estimation inspired by the work of Elgammal *et al.* [1]. The density at s is estimated using

$$P^m\left(\vec{I}_s^t\right) = \frac{1}{M \times M} \sum_{r \in \eta_s} \prod_{j=1}^{d} K_{\sigma_j}\left(\vec{I}_s^t(j) - \vec{B}_r(j)\right). \quad (13)$$

where K_{σ_j} is a kernel function, j is the color-space index and B is a background image. As suggested by Elgammal *et al.* [1], we implemented K_σ with a zero-mean, σ^2-variance Gaussian. In this way, a single global 1-D lookup table with 256 entries can be pre-calculated to allow speedup during runtime. The table values are then accessed with the intensity difference $\vec{I}_s^t(j) - \vec{B}_r(j)$ as index. Let us also mention that the background frame B used in P^m can be updated at every time t to account for illumination variations as follows: $\vec{B}_s \leftarrow (1-\alpha)\vec{B}_s + \alpha\vec{I}_s^t$.

IV. Light and Fast Method

Our *light and fast* method is based on two assumptions. The first one stipulates that the background distribution is temporally stationary, i.e., $P(\vec{B}_s^t)$ is independent of t in such a way that $P(\vec{B}_s^0) \approx P(\vec{B}_s^1) \approx P(\vec{B}_s^2) \approx \ldots$ (illumination variation will be addressed later). In other words, the background is assumed to be motionless. The second assumption stipulates that the background is piecewise temporal ergodic in time i.e., composed of piecewise constant regions for which spatial average color equals its temporal and also its ensemble average [22]. This means that we assume that the average temporal color/intensity of a pixel s is the same as the average color of a uniform region containing s.

Here, we represent the background as a whole with a Gaussian mixture of the form

$$P(\vec{B}_s) = \sum_{c=0}^{\mathcal{M}-1} P(\vec{B}_s|\omega_c, \vec{\theta}_c)P(\omega_c) \quad (14)$$

where ω_c is a class label, $\theta = \{\vec{\theta}_c = (\vec{\mu}_c, \Sigma_c) | c \in [0, \mathcal{M}[\}$ is the set of Gaussian parameters to be estimated, $P(\omega_i)$ is the prior probability of class ω_i, and B is the background image from which the mixture is estimated [21]. This equation stipulates that each pixel s is a member of a class ω_c with proportion $P(\omega_c)$ and likelihood $P(\vec{B}_s|\omega_c, \vec{\theta}_c)$.

This being said, B can be segmented into regions of uniform color (or intensity) by estimating a label field x for which x_s takes a value in $\{\omega_0,\ldots,\omega_{\mathcal{M}-1}\}$. In other words, it is possible to assign one class ω_i to each pixel s in such a way that every pixel having label ω_i has a color that follows the Gaussian distribution $\mathcal{N}(\vec{\mu}_i, \Sigma_i)$. Since the color of a background pixel s is assumed to be constant in time (more or less a noise factor), its class label x_s is also assumed to be constant in time. Therefore, the probability of observing color $\vec{I}_s^t$ at time t is well approximated by $P(\vec{B}_s|x_s, \vec{\theta}_{x_s})$ where $x_s \in \{\omega_0, \omega_1, \ldots, \omega_{\mathcal{M}-1}\}$. Thus, a consequence of our piecewise-ergodic assumption is that the spatial noise within a class is assumed to be the same as the temporal background variation (noise) of the pixels belonging to that class, i.e., $P(\vec{I}_s^t) \approx P(\vec{B}_s|x_s, \vec{\theta}_{x_s})$. In this way,

our method can model the background with a global mixture of $\mathcal{M}$ Gaussians as opposed to $\mathcal{A}\times\mathcal{B}$ Gaussians [10] or $\mathcal{A}\times\mathcal{B}\times\mathcal{M}$ Gaussians [12] where $\mathcal{A}\times\mathcal{B}$ is the total number of pixels.

A. Gaussian Parameter Estimation

In order to obtain a reliable estimate for θ (Gaussian mixture parameters) and x (label field) we use Pieczynski's Iterative Conditional Estimation (ICE) algorithm [23] since it allows simultaneous estimation of θ and x. The ICE algorithm can be outlined as follows.

1) [Initialization Step] The parameters $\hat{\theta}^{[0]}$ and the label field $\hat{x}^{[0]}$ are initialized with the K-Means [24] algorithm. $p = 1$.
2) [Stochastic Step] With a Gibbs sampler, a label field $\hat{x}^{[p]}$ is generated according to the posterior distribution $P(\hat{x}^{[p-1]}|B, \hat{\theta}^{[p-1]})$.
3) [Estimation Step] With a maximum likelihood estimator, $\hat{\theta}^{[p]}$ is recomputed based on $\hat{x}$ and B. In our implementation, $\vec{\mu}_c^{[p]}$ and $\Sigma_c^{[p]}$ are computed with a classical empirical mean and variance-covariance estimator for each class.
4) If $\|\vec{\mu}^{[p]} - \vec{\mu}^{[p-1]}\| < T$, where T is a fixed threshold, then Stop. Else, $p = p + 1$ and go back to the Stochastic Step.

Here, the posterior distribution is modeled after Bayes' theorem

$$P\left(\hat{x}^{[p]}|B, \hat{\theta}^{[p]}\right) \propto P\left(B|\hat{x}^{[p]}, \hat{\theta}^{[p]}\right) P\left(\hat{x}^{[p]}\right). \tag{15}$$

Assuming independence of $\vec{B}_s$ given $\hat{x}_s^{[p]}$ and $\hat{\theta}^{[p]}$

$$P\left(\hat{x}^{[p]}|B, \hat{\theta}^{[p]}\right) \propto \prod_s P\left(\vec{B}_s|\hat{x}_s^{[p]}, \hat{\theta}^{[p]}\right) P\left(\hat{x}_s^{[p]}\right) \tag{16}$$

where $P(\vec{B}_s|\hat{x}_s^{[p]}, \hat{\theta}^{[p]})$ is a Gaussian distribution $\mathcal{N}(\vec{B}_s; \vec{\mu}_{\hat{x}_s}^{[p]}, \Sigma_{\hat{x}_s}^{[p]})$. As for $P(\hat{x}_s^{[p]})$, we use the simple Potts model based on a Gibbs distribution of the form $P(\hat{x}_s^{[p]}) \propto \exp[-\gamma U_{\eta_s}(\hat{x}_s^{[p]})]$ where γ is a constant and $U_{\eta_s}(\hat{x}_s^{[p]})$ is an energy function that counts the number of sites in the neighborhood η_s with a label different than $\hat{x}_s^{[p]}$. In our implementation, we use binary cliques linking site s to its eight spatial neighbors (second-order neighborhood).

B. Detecting Motion

Once the ICE algorithm has converged, we have in hand both the $\mathcal{M}$-class Gaussian mixture parameters modeling the background and the label field x indicating to which class each pixel has been assigned. Like most already-published motion detection methods, we assume that the color distribution of the moving objects is different from the background. In this way, since each pixel s is modeled by a global Gaussian distribution with parameters $\vec{\theta}_{x_s} = (\vec{\mu}_{x_s}^{[p]}, \Sigma_{x_s}^{[p]})$, we can expect that $P(\vec{I}_s^t) \approx P(\vec{B}_s|x_s, \vec{\theta}_{x_s})$ when pixel s in image I^t is part of the background and $P(\vec{I}_s^t) \neq P(\vec{B}_s|x_s, \vec{\theta}_{x_s})$ when pixel s is covered by a moving object. In this way, the detection criterion may be formulated as

$$L_s^t = \begin{cases} 1, & \text{if } P\left(\vec{I}_s^t|x_s, \vec{\theta}_{x_s}\right)/P\left(\vec{B}_s|x_s, \vec{\theta}_{x_s}\right) < \tau \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

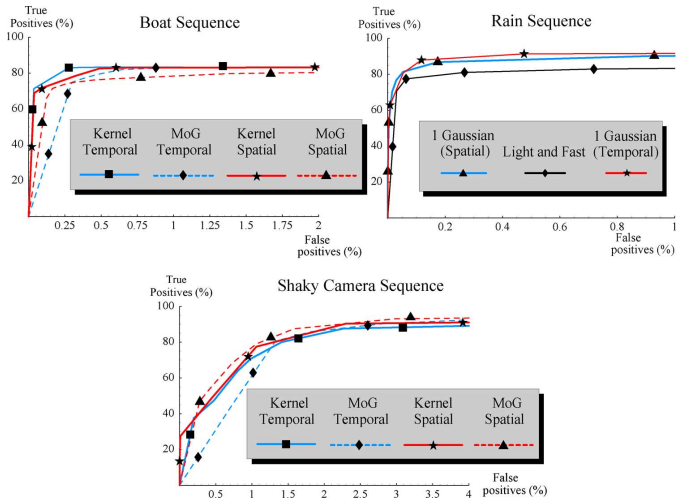

Fig. 3. ROC curves obtained for three sequences shown in Fig. 1.

Alternatively, for computational reasons, the Mahalanobis distance [10] can also be used as follows:

$$L_s^t = \begin{cases} 1, & \text{if } |D\left(\vec{I}_s^t, \vec{\theta}_{x_s}\right) - D\left(\vec{B}_s, \vec{\theta}_{x_s}\right)| > \tau' \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $D(\vec{a}_s, \vec{\theta}_{x_s}) = (\vec{a}_s - \vec{\mu}_{x_s})^T \Sigma_{x_s}^{-1} (\vec{a}_s - \vec{\mu}_{x_s})$. To further reduce processing times, for grayscale sequences, $D(\vec{I}_s^t, \vec{\theta}_{x_s})$ is pre-calculated and kept in memory in a global look-up table. Also, because empirically $D(\vec{B}_s, \vec{\theta}_{x_s})$ rarely goes above 32, we pre-calculate $D(\vec{B}_s, \vec{\theta}_{x_s})$ and store it in a 2-D array allowing five bits per pixel. To account for illumination variation, the background pixels may have their statistics updated at each time t [25].

V. Experimental Results

The tests aim to evaluate the performance of our two spatial methods compared to temporally trained methods. Three temporal methods have been implemented which, respectively, model every pixel with one Gaussian, a mixture of 3 Gaussians and a nonparametric kernel summation. Each of these temporal methods is trained on 15 to 80 frames depending on the sequence. A neighborhood η_s of size between 11×11 and 15×15 has been used for our *robust* method and a number of classes $\mathcal{M}$ between 7 and 10 was used for the *light and fast* method. Also, the threshold τ for every method has been varied to estimate the ROC curves of Fig. 3 and tuned to produce the best possible results for the sequences presented in Figs. 4 and 5. Also, $c = 5$, $\sigma = 2$, $\gamma = 2$, the learning rate α was set to 0 for every sequence (thus exploiting statistics of a single frame), and a simple 3×3 median filter was used to smooth out every motion mask L_t.

In order to gauge performance of our approach, sequences with known ground truth have been used (see Fig. 6). The first sequence is computer generated and exhibits a boat sailing on a wavy sea. Since the pixels representing water are multimodal, we used the kernel-based and MoG approaches to segment the sequence. The second sequence shows a moving truck filmed with an unstable camera making the background highly multimodal. The MoG and the Kernel-based approaches have

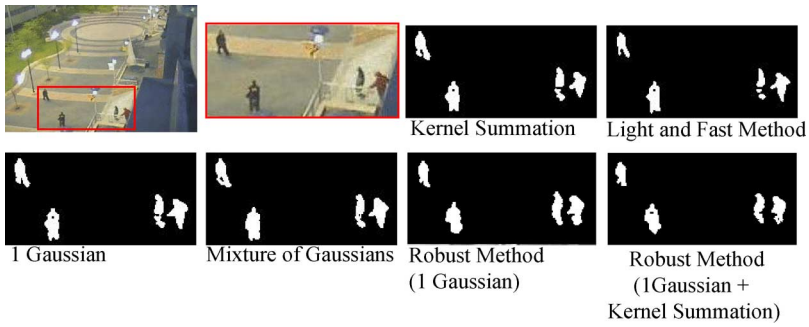

Fig. 4. Results for a 100-frame noisy sequence segmented with one Gaussian per pixel whose parameters have been spatially and temporally trained.

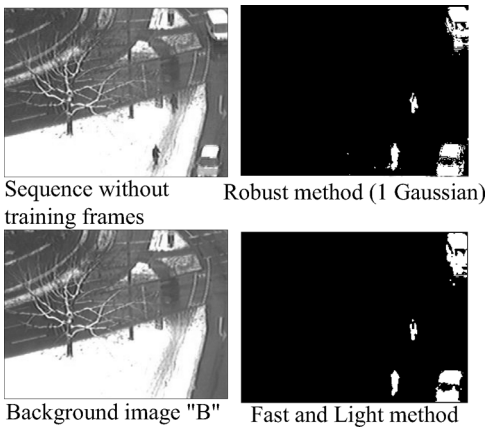

Fig. 5. Results for a 200-frame sequence with no frame absen of moving objects. A median filter was used to produce the background frame B.

TABLE I
TABLES SHOWING RESPECTIVELY THE FRAME RATE AND THE MINIMUM AMOUNT OF MEMORY NEEDED TO MODEL THE BACKGROUND

Frame rate (fps)				
	Grayscale sequence		Color sequence	
Methods	256×256	640×480	256×256	640×480
Light and Fast	240	44	136	26
One Gaussian	128	25	108	20
Mixture	23	5	12	3
Kernel	10	4	9	2
Minimum memory to model the background (KB)				
	Grayscale sequence		Color sequence	
Methods	256×256	640×480	256×256	640×480
Light and Fast	72	308	64	300
One Gaussian	512	2400	1536	7200
Mixture	1536	7200	3584	16800
Kernel	1921	9001	5761	27001

been used again. The third sequence [Fig. 6(c)] with known groundtruth, contains fixed background, strongly corrupted by rain. For this sequence, three methods have been used to detect motion: our *light and fast* method, our *robust* method with one Gaussian, and a single-Gaussian temporal method similar to Wren *et al.*'s [10]. Note that for this sequence, only the unimodal PDF P^u has been used for the *robust* methods. For these three sequences, ROC curves [21] have been generated and, as it can be seen, the difference between the spatially trained methods and the temporally trained methods is small. Note that for the Shaky camera and Rain sequences, the percentages of false positives and true positives have been obtained based on hand-segmented ground truth images.

We also segmented a sequence having no training frame without moving objects (see Fig. 5). The background frame B used to learn the Gaussian parameters was computed with a basic five-frame median filter: $\vec{B}(s) = \mathrm{Med}[I_s^0, I_s^{40}, I_s^{80}, I_s^{120}, I_s^{160}]$. The reader should note that, aside from the median filter, no outlier-detection method has been used.

To evaluate the computational complexity, we ran the methods on color and grayscale sequences of different sizes. Here, the MoG mixes two Gaussians, the temporal kernel-based method uses 30 training frames, and our *light and fast* method uses 8 classes. As it can be seen in Table I, our *light and fast* method is significantly faster, especially for grayscale sequences for which look-up tables have been used to store $D(\vec{I}_t(s), \vec{\theta}_{x_s})$ in memory (these processing rates include the 3×3 median filtering). Also, the minimum amount of memory required to model the background is significantly smaller for our *light and fast* method than for the other ones. Every program has been executed on a 2.2-GHz AMD Athlon processor.

VI. CONCLUSION

In this letter, a novel spatial framework for the background subtraction problem has been presented. Our framework is based on the idea that, for many applications, the temporal intensity/color distribution observed over a pixel corresponds to the statistical distribution observed spatially around that same pixel.

Our framework offers three main advantages. First, the statistical parameters can be learned with one background frame instead of a series of frames as is usually the case for single-Gaussian and the MoG model. This has the advantage of requiring less memory and being more flexible in the presence of sequences having no training frames without moving objects. Second, as opposed to the kernel-based method, only one frame (instead of N) is kept in memory at runtime. This again is a major advantage memory-wise. Last, but not least, our framework maintains the conceptual simplicity and strength of the background subtraction methods it is based on. The detection function, the adaptation to global illumination variations and the statistical learning phase are implemented in a way that is very similar to the one originally proposed in the temporal framework. In spite of these advantages, our framework does have one limitation. As we mentioned previously, our approach is based on the assumption that the temporal distribution of a pixel

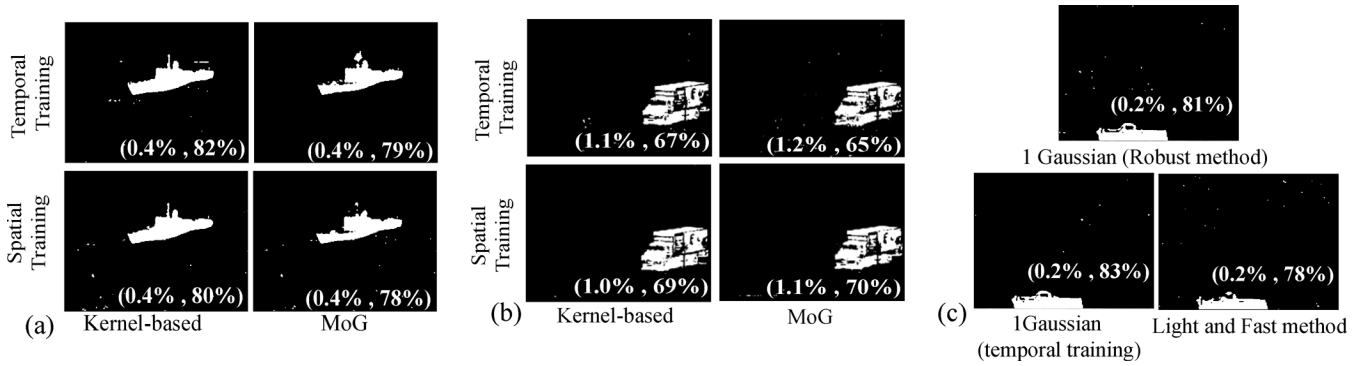

Fig. 6. (a) 20-frame synthetic sequence of a boat sailing on a wavy sea and (b) 90-frame sequence shot with an unstable camera. In each case, the MoG and the kernel-based method have been used. Both were trained either temporally or spatially. (c) Results for a 100-frame sequence corrupted by rain segmented with one Gaussian per pixel. The numbers in the lower right corner indicate the average percentage of false positives and true positives. These sequences are shown in Fig. 1.

s can be approximated by a spatial distribution. Although this assumption is true for many real-life scenarios, it nonetheless fails to recognize "temporal textures" arising from a temporally periodic event such as a blinking light. To compensate for this limitation, a specific updating scheme such as the one from (10)–(12), would need to be implemented.

In summary, we have shown that results obtained with our framework, on sequences with noise, camera jitter and animated background are, for all practical purposes, similar to the ones obtained with temporal methods.

ACKNOWLEDGMENT

The authors would like to thank the group of Prof. H.-H. Nagel for the winter sequence of Fig. 5 and Dr. A. Elgammal for the sequences of Fig. 1(a) and (c).

REFERENCES

[1] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density for visual surveillance," *Proc. IEEE*, vol. 90, pp. 1151–1163, 2002.

[2] N. Friedman and S. J. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proc. Uncertainty Artif. Intell. Conf.*, 1997, pp. 175–181.

[3] X.-S. Zhou, D. Comaniciu, and A. Gupta, "An information fusion framework for robust shape tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 115–129, Jan. 2005.

[4] A. McIvor, "Background subtraction techniques," in *Proc. Image Video Comput.*, 2000, pp. 147–153.

[5] R. Mech and M. Wollborn, "A noise robust method for 2d shape estimation of moving objects in video sequences considering a moving camera," *Signal Process.*, vol. 66, no. 2, pp. 203–217, 1998.

[6] C. Dumontier, F. Luthon, and J.-P. Charras, "Real-time dsp implementation for mfr-based video motion detection," *IEEE Trans. Image Processing*, vol. 8, no. 10, pp. 1341–1347, Oct. 1999.

[7] P. L. Rosin, "Unimodal thresholding," *Pattern Recognit.*, vol. 34, no. 11, pp. 2083–2096, 2001.

[8] P. L. Rosin and J. Herva, "Remote sensing image thresholding methods for determining landslide activity," *Int. J. Remote Sens.*, vol. 26, no. 6, pp. 1075–1092, 2005.

[9] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving object and background separation," *Signal Process.*, vol. 66, no. 2, pp. 219–232, Apr. 1998.

[10] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, 1997.

[11] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, "Detection and location of people in video images using adaptive fusion of color and edge information," in *Proc. Int. Conf. Pattern Recognit.*, 2000, pp. 4627–4631.

[12] C. Stauffer and E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[13] A. Mittal and D. Huttenlocher, "Scene modeling for wide area surveillance and image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2000, pp. 160–167.

[14] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 302–309.

[15] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust kalman filter," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 44–50.

[16] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, 2006.

[17] D. Russel and S. Gong, "A highly efficient block-based dynamic background model," in *Proc. IEEE Conf. Ser. Video Signal Based Surveill.*, 2005, pp. 417–422.

[18] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[19] Y. Sheikh and M. Shah, "Bayesian object detection in dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 74–79.

[20] P. J. Huber, *Robust Statistical Procedures*, 2nd ed. Philadelphia, PA: SIAM, 1996.

[21] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2000.

[22] V. Oppenheim and R. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[23] W. Pieczynski, "Statistical image segmentation," *Mach. Graph. Vis.*, vol. 1, no. 1, pp. 261–268, 1992.

[24] R. C. Dubes, "Cluster analysis and related issues," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. Singapore: World Scientific, 1992.

[25] P.-M. Jodoin, M. Mignotte, and J. Konrad, "Light and fast statistical motion detection method based on ergodic model," in *Proc. IEEE Int. Conf. Image Process.*, 2006, pp. 1053–1056.