LIGHT AND FAST STATISTICAL MOTION DETECTION METHOD BASED ON ERGODIC MODEL

Pierre-Marc Jodoin† *Max Mignotte*† *Janusz Konrad*††

†Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
P.O. Box 6128, Stn. Centre-Ville, Montréal, Qc, Canada

††Department of Electrical and Computer Engineering
Boston University
8 St-Mary's st., Boston, MA 02215

ABSTRACT

In this paper, we propose a light and fast pixel-based statistical motion detection method based on a background subtraction procedure. The statistical representation of the background relies on its spatial color distributions herein modeled by a mixture of Gaussians. The Gaussian parameters are obtained after segmenting one reference frame with an unsupervised Bayesian approach whose parameter estimation step is ensured by the K-Means and the Iterated Conditional Estimation (ICE) algorithms. Since the motion detection function only depends on a global mixture of M Gaussians, only a few bits per pixel need to be stored in memory. Our method achieves real-time performances, especially when look up tables are used to store pre-calculated data. Results have been obtained on synthetic and real video sequences and compared with other statistical methods.

Index Terms— Image motion analysis, Object detection

1. INTRODUCTION

Although the definition of motion detection can vary from one approach to another, we define it as being the task of labeling moving objects shot in front of a background that has little or no movement. This is typically the case for applications using a fixed camera such as video surveillance or traffic monitoring.

Most motion detection methods based on a background subtraction criterion can be classified into two large categories, namely: (1) pixel-based methods which segment independently each pixel and (2) spatial methods which use regional constraints. The fastest motion detection methods often fall into the first category of pixel-based methods[1].

Probably the most intuitive and fast pixel-based methods are the ones comparing intensity changes between frames. The intensity difference is usually computed between two successive frames or between a frame and a reference image containing no moving objects [1, 2, 3]. The intensity difference is then thresholded with predetermined global threshold. Although adaptive thresholds [4, 5] can be used, these methods are sensitive to phenomena that violate the basic assumptions of motion detection. Thus, to better account for noise, Wren *et al.* [6] used a statistical background model to detect and track people. Based on the assumption that the temporal noise is uncorrelated and can be modeled by a zero-mean Gaussian distribution, their model uses one Gaussian function per pixel to model the temporal local color level distribution. The Gaussian parameters are learned on a sequence of frames exhibiting a static background while motion detection is performed by finding the set of pixels (at each frame) whose probability value of its color level is below a predetermined threshold.

For many outdoor applications, natural events such as wind shaking trees, animated water, snow or other natural phenomena, the background cannot be assumed to be static. For those applications, the color distribution at each pixel is often multimodal and can hardly be modeled by one single Gaussian. In this way, a direct extension of the single Gaussian method, involving a mixture of Gaussians, has demonstrated a good deal of robustness [7, 8]. Friedman and Russel [7] argue that when monitoring highway traffic, a single pixel can cover more than one object over time. For this reason, they model the color distribution of each pixel by a mixture of three Gaussians. The Gaussian mixture parameters are learned on a sequence of frames with an Expectation Maximization (EM) algorithm. Similarly, Stauffer and Grimson [8] use a mixture of Gaussians to perform real-time tracking. For both methods, the Gaussians are weighted by the frequency with which they account for the background. A more general method is presented by Mittal and Huttenlocher [9] who use Gaussian mixtures to model the pixels of a panorama (obtained by stitching images obtained with a pivoting camera).

Among the non-parametric models used to represent the pixel color distribution, we can cite the ones using kernel density estimators (also called *Parzen Windows* in some communities [10]). Such method was used by Elgammal *et al.* [11] and more recently by Mittal and Paragios [12] who came up with a variable bandwidth formulation. In [13], Toyama *et al.* use a Wiener filter to predict the color value of each pixel based on the history of recent values. Another linear predictor based on Kalman filter has been proposed by Koller *et al* [14]. More recently, Zong and Sclaroff [15] proposed a Kalman-based generative model to represent dynamic backgrounds. All these linear predictors have the advantage to learn repetitive patterns and thus detect moving object sharing a similar grayscale distribution with the background.

In this paper, we propose a statistical background subtraction method based on the *spatial* distribution of its color-scale values. The main objective of our method is to be fast and light at the same time in such a way that it could eventually be implemented on an architecture having limited hardware (PDA, cell phone, etc.). Under the assumption that neighboring pixels often share a similar color and a similar temporal noise distribution, our method models the spatial background with a global mixture of M Gaussian distributions. Modeling the entire background with M Gaussians requires much less memory than, say, one (or many) Gaussian per pixel [6, 8]. Our method also needs only one training frame as opposed to a sequence of frames. Although our method uses a

[1] Since the body of literature addressing motion detection is signifi cant, we will only focus on the methods close to ours.

global statistical model, the results obtained are similar to the ones obtained by methods using local statistics.

The rest of the paper is organized as follows. In Section 2, the overall framework of our method is presented. Then, the Markovian framework and the motion detection criteria of our method are explicitly formulated. Section 3 shows how our method can be adapted to take care of problems such as illumination variation or the lack of training frames absent of foreground movement. Section 4 then shows some results before Section 5 concludes.

2. PROPOSED METHOD

Let $B_t = \{\vec{B}_t(s)|s \in S\}$ and $I_t = \{\vec{I}_t(s)|s \in S\}$ be random fields at time t defined on an $\mathcal{A} \times \mathcal{B}$ orthogonal lattice S. B_t represents the background at time t and I_t is a single frame containing B_t with moving objects in front of it. For implementation purposes, $\vec{B}_t(s)$ and $\vec{I}_t(s)$ take a value between 0 and 255 for grayscale sequences and between $(0,0,0)$ and $(255,255,255)$ for color sequences.

The method we propose is based on two assumptions. The first one stipulates that the background distribution is temporally stationary, *i.e.* $P(\vec{B}_k(s))$ is independent of k in such a way that $P(\vec{B}_0(s)) \approx P(\vec{B}_1(s)) \approx P(\vec{B}_2(s)) \ldots$ (illumination variation will be addressed in Section 3). The second assumption supposes that the background is piecewise temporal ergodic *i.e.*, composed of piecewise constant regions for which spatial average color equals its temporal and also its ensemble average [16]. With these two assumptions, the background spatial color-scale distribution is used to represent temporal color distribution. Modeling of the background is done with a Gaussian mixture of the form $P(\vec{B}_{\text{REF}}(s)) = \sum_{c=0}^{M-1} P(\vec{B}_{\text{REF}}(s)|\omega_c, \vec{\theta}_c)P(\omega_c)$ where ω_c is a class label, $\vec{\theta}_c = (\vec{\mu}_c, \Sigma_c)$ is the parameter vector (to be estimated) of the c^{th} Gaussian distribution, $P(\omega_i)$ is the prior probability of class ω_i, and B_{REF} is the reference background frame on which the mixture is estimated [10].

This being said, B_{REF} can be segmented into regions of uniform color (or grayscale) by estimating a label field X for which x is a realization and $x(s)$ takes a value in $\{\omega_0, ..., \omega_{\text{M-1}}\}$. In this way, a region assigned to class ω_i means that the pixels contained in that region have a color that follow the Gaussian distribution determined by the parameter vector $\vec{\theta}_{\omega_i}$. 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 too. Therefore, the probability of observing color $\vec{B}_f(s)$ at time f is well approximated by $P(\vec{B}_{\text{REF}}(s)|x(s), \vec{\theta}_i)$. In other words, our two assumptions allow us to say that the spatial statistics observed on frame B_{REF} can be used to represent the color distribution of each pixel in time. In this way, our method can model the background with M Gaussians as opposed to $\mathcal{A} \times \mathcal{B}$ Gaussians [6] or $\mathcal{A} \times \mathcal{B} \times M$ Gaussians [8].

2.1. Parameter Estimation

Since our method is unsupervised, the Gaussian mixture parameters $\theta = \{(\vec{\mu}_c, \Sigma_c)|\forall c \in [0, M[\}$ and the label field X are both initially unknown. In order to obtain a reliable estimate for θ and X simultaneously, we resort to Pieczynski's Iterative Conditional Estimation (ICE) algorithm [17] that we outline as follows:

1. [Initialization Step] The parameters $\hat{\theta}^{[0]}$ and the label field $\hat{x}^{[0]}$ are initialized with the K-Means [18] algorithm. $p = 0$.
2. [Stochastic Step] With a Gibbs sampler, a label field $\hat{x}^{[p]}$ is generated according to the posterior distribution $P(\hat{x}^{[p]}|B_{\text{REF}}, \hat{\theta}^{[p]})$.
3. [Estimation Step] With a maximum likelihood estimator, $\hat{\theta}^{[p+1]}$ is recomputed based on $\hat{x}$ and B_{REF}. In our implementation, $\vec{\mu}_c^{[p+1]}$ and $\Sigma_c^{[p+1]}$ 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 Stochastic Step.

In this procedure, the posterior distribution is modeled after Bayes theorem [10]: $P(\hat{x}^{[p]}|B_{\text{REF}}, \hat{\theta}^{[p]}) \propto P(B_{\text{REF}}|\hat{x}^{[p]}, \hat{\theta}^{[p]})P(\hat{x}^{[p]})$. Assuming independence of each random variables $\vec{B}_{\text{REF}}(s)$ given $\hat{x}^{[p]}(s)$ and $\hat{\theta}^{[p]}$, it can be stated that $P(\hat{x}^{[p]}|B_{\text{REF}}, \hat{\theta}^{[p]}) \propto \prod_s P(\vec{B}_{\text{REF}}(s)|\hat{x}^{[p]}(s), \hat{\theta}^{[p]})P(\hat{x}^{[p]}(s)|\eta_s)$ where, in our implementation, $P(\vec{B}_{\text{REF}}(s)|\hat{x}^{[p]}(s), \hat{\theta}^{[p]})$ is a Gaussian distribution with the shape $\mathcal{N}(\vec{B}_{\text{REF}}(s); \vec{\mu}^{[p]}_{\hat{x}(s)}, \Sigma^{[p]}_{\hat{x}(s)})$ with $\vec{\mu}_{x(s)}$ and $\Sigma_{x(s)}$ taken from $\hat{\theta}^{[p]}$. As for $P(\hat{x}^{[p]}(s)|\eta_s)$, we use the simple Ising model [19] based on a Gibbs distribution of the form $\exp[-\alpha U(\hat{x}^{[p]}(s), \eta_s)]$ where α is a constant and $U(\hat{x}^{[p]}(s), \eta_s)$ is a function that counts the number of sites in the neighborhood η_s with a label different than $\hat{x}^{[p]}(s)$. In our implementation, we use binary cliques linking site s to its eight spatial neighbors [20].

2.2. Detecting Motion

Once ICE has converged, we have in hand both the M-class Gaussian mixture parameters modeling the background and the label field "x" indicating to which class each pixel has been assigned. Like most previous 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 $\vec{\theta}_{x(s)}$, we can expect that $P(\vec{I}_t(s)|x(s), \theta) \cong P(\vec{B}_{\text{REF}}(s)|x(s), \theta)$ when pixel s in image I_t is part of the background and $P(\vec{I}_t(s)|x(s), \theta) \neq P(\vec{B}_{\text{REF}}(s)|x(s), \theta)$ when pixel s is covered by a moving object. In this way, considering $L_t = \{L_t(s)|s \in S\}$ as being the to-be-estimated binary motion label field, a detection criterion may be formulated as

$$L_t(s) = \begin{cases} 1 & \text{if } P(\vec{I}_t(s)|x(s), \theta)/P(\vec{B}_{\text{REF}}(s)|x(s), \theta) < Tr \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

or, for computational reasons, the Mahalanobis distance [6] can also be used as follows

$$L_t(s) = \begin{cases} 1 & \text{if } |D(\vec{I}_t(s), \theta_{x(s)}) - D(\vec{B}_{\text{REF}}(s), \theta_{x(s)})| > Tr' \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

where $D(\vec{I}_t(s), \theta_{x(s)}) = (\vec{I}_t(s) - \vec{\mu}_{x(s)})^T \Sigma^{-1}_{x(s)} (\vec{I}_t(s) - \vec{\mu}_{x(s)})$. To further reduce processing times, as suggested by [8], the variance-covariance matrix Σ_c is assumed to be diagonal. Also, $D(\vec{B}_{\text{REF}}(s), \theta_{x(s)})$ is precalculated and kept in memory in a look-up table. Since $D(\vec{B}_{\text{REF}}(s), \theta_{x(s)})$ rarely goes above 16, four bits per pixel are used to store it in memory.

3. MAKING OUR METHOD ROBUST

To account for illumination variation (such as when a cloud occludes the sun for instance) the background pixels may have their

statistics updated at each frame with the following strategy [6, 7]

$$\vec{\mu}_c = (1-\alpha)\vec{\mu}_c + \alpha\frac{1}{N_c}\sum_{\substack{s\in S \\ L_t(s)=0}} \delta(x(s),\omega_c)\vec{I}_t(s) \quad (3)$$

$$\Sigma_c = (1-\alpha)\Sigma_c + \alpha\frac{1}{N_c}\sum_{\substack{s\in S \\ L_t(s)=0}} \delta(x(s),\omega_c)(\vec{I}_t(s)-\vec{\mu}_c)^2 \quad (4)$$

where α is a *forgetting constant* [7] and N_c is the number of pixels assigned to class ω_c and detected as being part of the background at time t. Also, in case of local illumination variations, the label field x may be re-updated (once every few minutes or so), prior to compute equations (3) and (4).

Another issue often addressed in the literature is when no training period absent of foreground objects is available. This is a fundamental issue since the Gaussian mixture parameters θ and the label field x need a background image B_{REF} to be estimated. However, since our method requires only *one* training frame, for applications for which no background frame is available (such as traffic monitoring for instance), B_{REF} can be estimated by applying a median filter to a sequence of ζ frames, *i.e.*

$$B_{\text{REF}}(s) = \text{Median}(\{B_{k+1}(s), B_{k+2}(s), \ldots, B_{k+\zeta}(s)\}). \quad (5)$$

Such a median filter was used on the two sequences of Fig.3.

4. EXPERIMENTAL RESULTS

Our method is tested on synthetic and real image sequences. In both cases, our method is compared with three statistical methods which model each background pixel with either one Gaussian, a mixture of N Gaussians and a non-parametric kernel summation. Each of these methods is trained on 20 to 60 frames depending on the sequence. As for our method, the number of classes M is set between 7 and 10. Notice that every methods has been tuned to produce the best possible results and that a simple 3×3 median filter was used to smooth out L_t. The four methods are compared with respect to precision, speed and the amount of memory they use to model the background.

To evaluate precision, a color and a grayscale synthetic sequence with known ground truth have been used (see Fig. 1). These sequences –that we call *statue* and *car on park*– are made of real images whose motion has been artificially simulated. To see how robust our method is, some Gaussian noise has been added to these sequences. The precision was evaluated by computing the average percentage of false negatives and false positives over the entire sequence. As can be seen in Table 1, although our method uses global statistics, the error it produces is in the same order of magnitude as the one produced by the other three methods.

To evaluate processing times, we ran the four methods on color and grayscale sequences of different sizes. As can be seen in Table 2 (a), our method is significantly faster, especially for grayscale sequences for which look-up tables have been used to store $D(\vec{I}_t(s), \theta_{x(s)})$ in memory. Notice that these processing rates include the 3×3 median filtering. Also, as shown in Table 2 (b), the minimum amount of memory required to model the background is significantly smaller for our method than for the other ones. Every program has been executed on a 2.2 Ghz AMD Athlon processor. We also compared the four methods using a real sequence shown in Fig 2. Again, the visual quality of the results returned by our method is similar to the other ones.

Car on Park sequence Statue sequence

Fig. 1. Snapshots of the synthetic sequences used.

Methods	*Car on Park* sequence StdDev=10	StdDev=20
Our Method	11/0.1	16/0.08
One Gaussian	7/0.04	12/0.04
Mixture	9/0.03	16/0.03
Kernel	8/0.03	15/0.03

Methods	*Statue* sequence StdDev=10	StdDev=20
Our Method	6/0.2	18/0.4
One Gaussian	2/0.09	16/0.05
Mixture	2/0.08	21/0.04
Kernel	6/0.05	15/0.1

Table 1. Percentage of false negatives/positives for two synthetic sequences. Each sequence has been corrupted by random Gaussian noise (with standard deviation of 10 and 20).

Frame rate (fps)

		Grayscale sequence		Color sequence	
	Methods	256 × 256	640 × 480	256 × 256	640 × 480
	Our Method	240	44	136	26
	One Gaussian	128	25	108	20
	Mixture	23	5	12	3
(a)	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
	Our Method	82	348	72	338
	One Gaussian	128	600	384	1800
	Mixture	256	1200	768	3600
(b)	Kernel	1280	6000	3840	18000

Table 2. Tables showing respectively the frame rate and the minimum amount of memory needed to model the background for each statistical method.

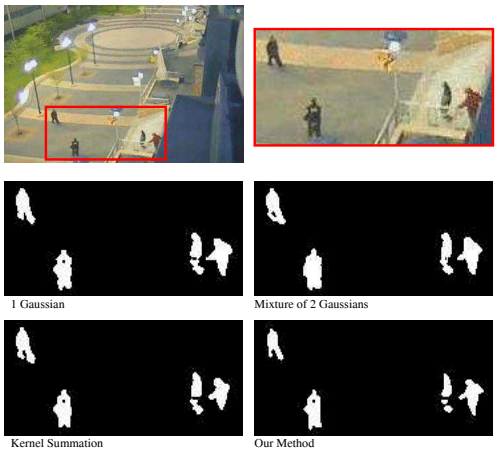

Fig. 2. Real sequence on which our method was compared to three other statistical methods.

We finally tested our method on real sequences having no frames absent of moving objects (these sequences have been taken from a database at the university of Karlshrue [21]). As mentioned in Section 3, we obtained B_{REF} by applying a median filter to a series of frames. We compared our results with the ones returned by a global thresholding procedure [7]. As shown in Figure 3, our method seems more robust to noise (left column) and to small camera jitter (right column).

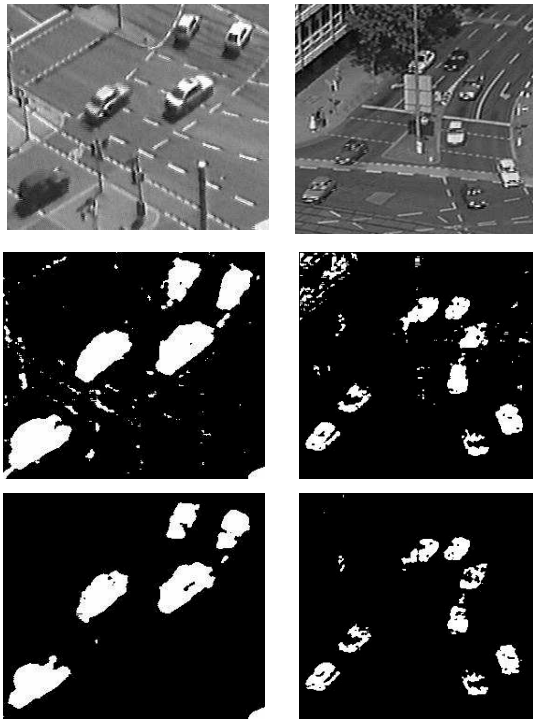

Fig. 3. From top to bottom: two traffic sequences having no training frames absent of moving object, results obtained with a global threshold and results obtained with our method.

5. CONCLUSION

In this paper, a light and fast motion detection method has been presented. As opposed to most previous background subtraction methods based on *temporal* statistics, our model uses *spatial* statistics (a Gaussian mixture) to detect motion. Among other things, this has the advantage of requiring only one training frame and little memory to model the background. Also, since the motion detection function is based on a simple Mahalanobis distance, the processing times are low, especially when look-up tables are being used. Also, our method can be easily parallelized and could fit architectures having limited hardware such as PDAs or cellular phones. The main limitation of our method is its sensitivity to background motion. This situation is in part due to the assumption that the temporal noise is unimodal (cf. Eq. (1)) which isn't true when the background moves. We are currently working on a version that compensates background motion and large camera jitter.

6. REFERENCES

[1] Zhang D. and Lu G., "Segmentation of moving objects in image sequence: A review," *Circuits, Systems and Signal Process.*, vol. 20, no. 2, pp. 143–183, 2001.

[2] Bovik A., Ed., *Handbook of Image and Video Processing*, pub-ACADEMIC, 2000.

[3] Rosin P. and Ioannidis E., "Evaluation of global image thresholding for change detection," *Pattern Recognition Letters*, vol. 24, pp. 2345–2356, 2003.

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

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

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

[7] Friedman N. and Russell S.J., "Image segmentation in video sequences: A probabilistic approach.," in *UAI*, 1997, pp. 175–181.

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

[9] Mittal A. and Huttenlocher D., "Scene modeling for wide area surveillance and image synthesis," in *Proc. of CVPR*, 2000, pp. 160–167.

[10] Duda R., Hart P., and Stork D., *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000.

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

[12] Mittal A. and Paragios N., "Motion-based background subtraction using adaptive kernel density estimation.," in *Proc. of CVPR*, 2004, pp. 302–309.

[13] Toyama K., Krumm J., Brumitt B., and Meyers B., "Wallflower: Principles and practice of background maintenance.," in *ICCV*, 1999, pp. 255–261.

[14] Koller D., Weber J., Huang T., Malik J., Ogasawara G., Rao B., and Russell S., "Towards robust automatic traffi c scene analysis in real-time," in *Proc of ICPR*, 1994, pp. 126–131.

[15] Zhong J. and Sclaroff S., "Segmenting foreground objects from a dynamic textured background via a robust kalman fi lter," in *Proc of ICCV*, 2003, pp. 44–50.

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

[17] Pieczynski W., "Statistical image segmentation," *Machine Graphics and Vision*, vol. 1, no. 1, pp. 261–268, 1992.

[18] Dubes R.C., *Cluster Analysis and Related Issues*, Handbook of Pattern Recognition and Computer Vision, C.H. Chen, L.F. Pau, and P.S.P. Wang (editors), 1992.

[19] Geman S. and Geman D., "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, 1984.

[20] Besag J., "On the statistical analysis of dirty pictures," *J. Roy. Stat. Soc.*, vol. 48, no. 3, pp. 259–302, 1986.

[21] "http://i21www.ira.uka.de/image_sequences/," .