Background Subtraction Framework Based on Local Spatial Distributions

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

†D.I.R.O.
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. Most statistical background subtraction techniques are based on the analysis of temporal color/intensity distributions. However, learning statistics on a series of time frames can be problematic, especially when no frames absent of moving objects are available or when the available memory isn't sufficient to store the series of frames needed for learning. In this paper, we propose a framework that allows common statistical motion detection methods to use spatial statistics gathered on one frame instead of a series of frames as is usually the case. This simple and flexible framework is suitable for various applications including the ones with a mobile background such as when a tree is shaken by wind or when the camera jitters. Three statistical background subtraction methods have been adapted to the proposed framework and tested on different synthetic and real image sequences.

Key words: Background subtraction, spatial distributions.

1 Introduction

Motion detection methods are generally used to evaluate and locate the presence (or absence) of motion in a given animated scene. To this end, one class of solutions that enjoys tremendous popularity is the family of background subtraction methods [1]. These methods are based on the assumption that the scene is made of a static background in front of which animated objects with different visual characteristics are observed. Typical applications for background subtraction methods include camera surveillance [2], traffic monitoring [3, 4] and various commercial applications [5].

As the name suggests, the most intuitive background subtraction method involves one background image and an animated sequence containing moving objects. These moving objects are segmented by simply thresholding the difference between the background and each frame. The threshold can be *a priori* known or estimated on the fly [6]. Unfortunately, such a simple solution is sensitive to background variations and can only meet the requirements of simple applications.

Background variations can be caused by all kinds of phenomena. For instance, noise induced by a cheap low-quality camera or by motion jitter caused by an unstable camera are typical situations that can't be handled properly by simplistic background subtraction methods. Also, there are many applications for which some background objects aren't perfectly static and induce local false positives. It's the case, for instance, when a tree is shaken by wind or when the

background includes animated texture such as wavy water. Another common source of variation is when the global illumination isn't constant in time and alters the appearance of the background. Such variation can be gradual such as when a cloud occludes the sun, or sudden such as when a light is turned on or off.

For all these situations, a more elaborate background subtraction strategy is required. In this perspective, many methods proposed in the literature, model each pixel of the background with a statistical model learned over a series of training frames. For these methods, the detection becomes a simple probability density function (PDF) thresholding procedure. For instance, a single-Gaussian distribution per pixel [7, 8] can be used to compensate for uncorrelated noise. However, this single-Gaussian approach is limited by the assumption that the color distribution of each pixel is unimodal in time. Although this assumption is true for some indoor environments, it isn't true for outdoor scenes made up of moving background objects or for sequences shot with an unstable camera. Therefore, because the color distribution of moving background pixels can be multimodal, many authors use a mixture of Gaussians (MoG) [9, 3, 10] to model the color distribution of each pixel. The number of Gaussians can be automatically adjusted [9] or predefined based on the nature of the application [3]. Non-parametric modeling [2, 11] based on a kernel density estimation has also been studied. The main advantage of this approach is the absence of parameters to learn and its ability to adapt to distributions with arbitrary shape. Let us also mention that block-based methods [12], Markovian methods [13] and predictive methods [14, 15], to name a few, have been also proposed.

The main limitation with statistical solutions is their need for a series of training frames absent of moving objects. Without these training frames, a non-trivial outlier-detection method has to be implemented [9]. Another limitation with these methods is the amount of memory some require. For example, in [3], every training frame needs to be stored in memory to estimate the MoG parameters. Also, for kernel-based methods [2, 11], a number of N frames need to be kept in memory during the entire detection process which, indeed, is costly memory-wise when N is large. In this paper, we propose a novel framework that allows training on only one frame and requires a small amount of memory during runtime. Our framework considers two kinds of illumination variations : a unimodal variation (caused by noise) and a multimodal variation (caused by local movement). The methods we have adapted to our framework are thus robust to noise and background motion.

The rest of the paper is organized as follows. In Section 2, we present our framework before Section 3 explains how statistical background subtraction methods can be adapted to it. Several results are then presented in Section 4 to illustrate how robust our framework is. Section 5 draws conclusions.

2 Proposed Framework

As mentioned earlier, the choice for a pixel-based background model is closely related to the nature of background variations. In this perspective, let us consider two kinds of background variations. The first one concerns variations due to noise. In this case, the background B is considered as being stationary and the color of each pixel s at time t is defined as $\boldsymbol{B}_t(s) = \boldsymbol{B}(s) + n$ where n is an uncorrelated noise factor and $\boldsymbol{B}(s)$ is the *ideal* noise-free background color (or intensity) of site s. In this case, the distribution of $\boldsymbol{B}_t(s)$ in time is considered to be unimodal and centered on $\boldsymbol{B}(s)$. The second kind of variations we consider is the one due to background movement caused by, say, an animated texture or by camera jitter. Considering that variations are due to local movements, it can be assumed that the distribution of $\boldsymbol{B}_t(s)$ in time is similar to the spatial distribution of $\boldsymbol{B}(s)$, *i.e.*, the spatial distribution $\{\boldsymbol{B}(r), \forall r \in \eta_s\}$ where η_s is a $M \times M$ neighborhood centered on s. As a matter of fact, as shown in Fig.1 (a)-(b), when a site s is locally animated, the color observed in time over s often corresponds to the color observed locally around s. Therefore, since the spatial distribution is often multimodal, the distribution of $\boldsymbol{B}_t(s)$ often turns out to be multimodal too.

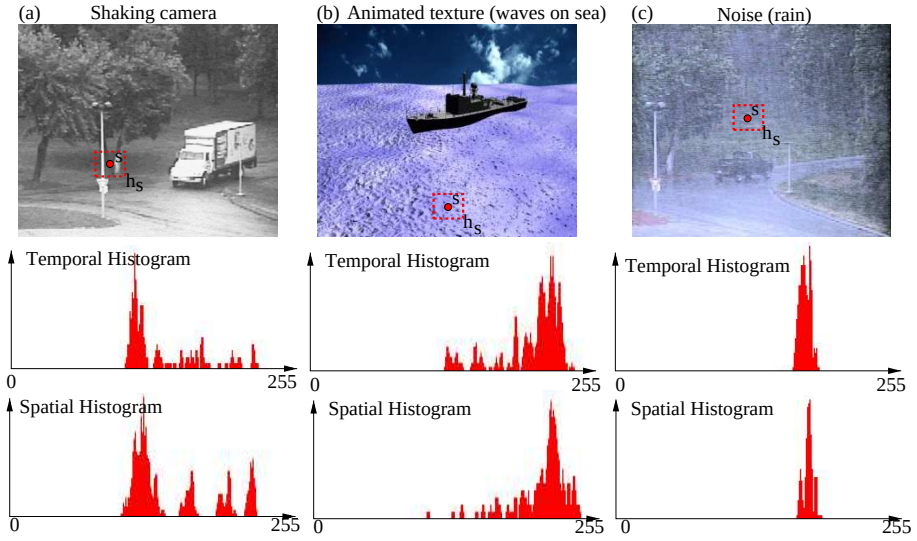

Fig. 1. *Three image sequences exhibiting local illumination variations. From left to right: a sequence shot with an unstable camera, a sequence including animated texture and a noisy sequence. The histograms show that the spatial intensity distributions often resemble the temporal distribution.*

In this paper, we propose a novel framework which uses only one frame for training. Unfortunately, with only one training frame, it isn't possible to deter-

mine whether the distribution of a site s in time is unimodal or multimodal. One can even think of applications where unimodal regions become multimodal after a while. A good example is when the wind starts to blow and suddenly animates a tree. In this way, since the modality of each pixel distribution isn't known *a priori* and can't be estimated with the analysis of only one background frame, we decided to use a *decision fusion* strategy. To this end, each pixel is modeled with two PDFs: one unimodal PDF (that we call P^u) and one multimodal PDF (called P^m). Both these PDFs are trained on one single background frame "B" (see Section 3 for more details on training). The goal of these two PDFs is to estimate a motion label field L_t which contains the motion status of each site s at time t (typically, $L_T(s) = 0$ when s is motionless and $L_T(s) = 1$ otherwise). The detection criterion can be formulated as follows

$$L_t(s) = \begin{cases} 0 & \text{if } P_s^u(I_t) > \tau \text{ OR } P_s^m(I_t) > \tau \\ 1 & \text{otherwise.} \end{cases} \quad (1)$$

where I_t is a frame observed at time t . Estimating the motion label field L_t with this equation turns out to be the same as blending two label fields L_t^u and L_t^m that would have been obtained after thresholding $P_s^u(I_t)$ and $P_s^m(I_t)$ separately. Other configurations for blending P^u and P^m as well as other thresholding procedures have been investigated during this research. It turned out that a decision criterion such as the one of Eq.1 is a good compromise between simplicity and efficiency.

3 Spatial Training

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

Single Gaussian

As mentioned earlier, P^u models variations due to uncorrelated noise and assumes that the background is perfectly motionless. To this end, P^u is modeled with a single Gaussian distribution

$$P_s^u(I_t) = \frac{1}{(2\pi)^{d/2}|\Sigma_s|^{-1/2}} \exp(-\frac{1}{2}(\boldsymbol{I}_t(s) - \boldsymbol{\mu}_s)^{\mathrm{T}} \Sigma_s^{-1}(\boldsymbol{I}_t(s) - \boldsymbol{\mu}_s) \quad (2)$$

where $d = 1$ for grayscale sequences and $d = 3$ for color sequences. Notice that for color sequences, Σ_s is a 3×3 variance-covariance matrix that, as suggested by Stauffer and Grimson [9], is assumed to be diagonal for efficiency purposes. Since only one training frame is available in this framework, $\boldsymbol{\mu}_s$ and Σ_s have to be estimated with data gathered around s. Of course, by the very nature of P^u, the spatial data should ideally have an unimodal distribution that resembles the one observed temporally over site s. Although some spatial neighborhoods of a scene offer that kind of unimodal distribution (see neighborhood A in Fig.

2), others are obviously multimodal (see neighborhood B in Fig. 2) and can't be used as is for training. In fact, using every pixels within a neighborhood η_s would often lead to corrupted (and useless) parameters. Thus, to prevent $\boldsymbol{\mu}_s$ and Σ_s from being corrupted by outliers (the gray pixels of the street near B in Fig. 2 for example), a robust function ρ is used to weight the importance of each sample $\boldsymbol{B}(r)$ [16]. More specifically, the parameter estimation can be expressed as

$$\begin{aligned}\boldsymbol{\mu}_s &= \frac{1}{\sum_{r\in\eta_s}\rho_{s,r}}\sum_{r\in\eta_s}\rho_{s,r}\boldsymbol{B}(r)\\ \Sigma_s(j) &= \frac{1}{\sum_{r\in\eta_s}\rho_{s,r}}\sum_{r\in\eta_s}\rho_{s,r}(B(r,j)-\mu_s(j))^2 \qquad \forall j\in[1,d]\end{aligned} \tag{3}$$

where $\Sigma_s(j)$ is the variance of the j^{th} color space dimension and η_s is a $M\times M$ neighborhood centered on s. As suggested by Huber [16], we defined $\rho(s,r)$ as

$$\rho(s,r) = \begin{cases} 1 & \text{if} ||\boldsymbol{B}(s)-\boldsymbol{B}(r)||_2 \le c \\ \frac{c}{||\boldsymbol{B}(s)-\boldsymbol{B}(r)||_2} & \text{otherwise}\end{cases} \tag{4}$$

where c is a constant that we set to 5. This robust estimator leads to interesting results as shown by the black dotted Gaussian of Fig. 2.

Notice that global lighting variations can be compensated by updating $\boldsymbol{\mu}_s$ and Σ_s at every frame [7, 8] as follows

$$\boldsymbol{\mu}_s \leftarrow \alpha\boldsymbol{I}_t(s)+(1-\alpha)\boldsymbol{\mu}_s, \qquad \forall\, L_t(s)=0 \tag{5}$$

$$\Sigma_s(j) \leftarrow \alpha(I_t(s,j)-\mu_s(j))^2+(1-\alpha)\Sigma_s(j) \qquad \forall\, L_t(s)=0, \forall j\in[1,d]. \tag{6}$$

where $\alpha\in[0,1[$ is the so-called *learning rate* [3].

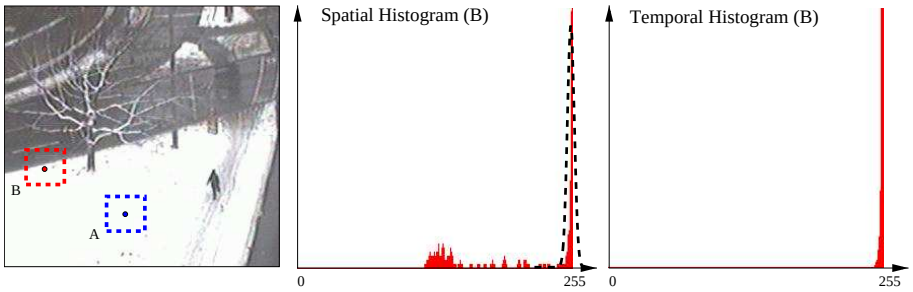

Fig. 2. *A sequence shot with a perfectly static camera. While the temporal intensity distribution of pixel B is unimodal and centered on intensity* 254*, the spatial intensity distribution around B is bimodal. However, estimating the Gaussian parameters with Eq. (3) leads to a distribution (in black) centered on the main mode, uncorrupted by the graylevels of the street.*

Mixture of Gaussians

Multimodal histograms such as the ones in Fig. 1 (a)-(b) can't be appropriately modeled with one single Gaussian. However, a mixture of K Gaussians can be a good choice to model such distributions:

$$P_s^m(I_t) = \sum_{i=1}^{K} w_{s,i}\, \mathcal{N}(\boldsymbol{I}_t(s), \boldsymbol{\mu}_{s,i}, \Sigma_{s,i}) \tag{7}$$

where $\mathcal{N}(.)$ is a Gaussian similar to the one of Eq.(2), $w_{s,i}$ is the weight of the i^{th} Gaussian and K is the total number of Gaussians (between 2 and 5 typically). In this context, a number of $3 \times K$ parameters per pixel need to be estimated during the training phase. To do so, the well known K-means algorithm has been implemented. The objective of K-means is to iteratively estimate the mean of K clusters by minimizing the total intra-cluster variance. In our framework, K-means takes as input for each pixel s, the $M \times M$ background pixels $\{\boldsymbol{B}(r), r \in \eta_s\}$ contained withing the square neighborhood η_s. When the algorithm converges and the mean $\boldsymbol{\mu}_{s,i}$ of every cluster has been estimated, the variance $\Sigma_{s,i}$ and the weight $w_{s,i}$ are then estimated. In this paper, the number of K-means iterations was set to 6. For further explanations on K-means, please refer to [17].

As we mentioned for P_s^u, the MoG parameters can be updated at every frame to account for illumination variations. As suggested by Stauffer and Grimson [9], at every frame I_t, the parameters of the Gaussian that matches the observation $\boldsymbol{I}_t(s)$ can be updated as follows

$$\boldsymbol{\mu}_{s,i} \leftarrow \alpha \boldsymbol{I}_t(s) + (1-\alpha)\boldsymbol{\mu}_{s,i}, \qquad \forall L_t(s) = 0 \tag{8}$$

$$\Sigma_{s,i}(j) \leftarrow \alpha(I_t(s,j) - \mu_{s,i})^2 + (1-\alpha)\Sigma_{s,i}(j) \qquad \forall L_t(s) = 0 \tag{9}$$

$$w_{s,i} \leftarrow (1-\alpha) w_{s,i} + \alpha M_{s,i} \tag{10}$$

where $M_{s,i}$ is 1 for the Gaussian that matches and 0 for the other models. The weights $w_{s,i}$ are then normalized.

Nonparametric Density Estimation

Since the color distribution of each pixel can't always be assumed to follow a parametric form, a multimodal density can be estimated with an unstructured approach. One such nonparametric approach is the kernel-based density estimation (also called *Parzen density estimate* [18]) which locally estimates density from a small number of neighboring samples. The kernel method we have implemented has been inspired by the work of Elgammal *et al.* [2].

Considering η_s as being a $M \times M$ neighborhood centered on site s, the density at s is estimated with

$$P_s^m(I_t) = \frac{1}{M \times M} \sum_{r \in \eta_s} K_\sigma(I_t(s) - B(r)) \tag{11}$$

for grayscale sequences and, for color sequences,

$$P_s^m(I_t) = \frac{1}{M \times M} \sum_{r \in \eta_s} \prod_{j=1}^{3} K_{\sigma_j}(I_t(s,j) - B(r,j)). \tag{12}$$

where j is the color-space index (red, green or blue) and B is a background frame. Here, K is a *kernel function* –*i.e.* some PDF– which, in practice, turns out to be normal or uniform [18]. As suggested by Elgammal *et al.* [2], we implemented K_σ as being a zero-mean Gaussian of the form

$$K_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-x^2}{2\sigma^2}. \tag{13}$$

Although σ can be estimated on the fly for each pixel [2], we used a constant value. In this way, a single global 256-float-long lookup table can be precalculated to allow significant speedup during runtime. The table values are accessed with the intensity value difference $I_t(s,j) - B(r,j)$ as index.

Let us mention that the background frame B used to model P^m can be updated at every frame to account for lighting variations. This can be easily done with the following operation:

$$\boldsymbol{B}(s) \leftarrow \alpha \boldsymbol{B}(s) + (1-\alpha)\boldsymbol{I}_t(s). \tag{14}$$

4 Experimental Results

To validate our framework, we have segmented sequences representing different challenges. These tests aim at validating how stable and robust our framework is with respect to traditional methods using a temporal-training approach. For each example presented in this section, a neighborhood η_s of size between 11×11 and 15×15 has been used.

The first sequence we segmented is the one presented in Fig.3 (a) which contains strong noise caused by rain. The segmentation has been performed by thresholding independently a single-Gaussian PDF (see Eq. (2)) trained over each pixel. At first, the Gaussian parameters have been computed with a 20-frame temporal training. Then, the parameters have been trained spatially on one frame using Eq. (3). As shown in Fig. 3, the label fields obtained with both approaches are, to all practical ends, very much similar. They both exhibit some few isolated false positives (due to rain) and some false negatives over the truck window.

The second sequence we have segmented (see Fig. 3 (b)) is one with no training frame absent of moving objects. The background frame B needed to learn the Gaussian parameters has been computed with a simple five-frame median filter : $\boldsymbol{B}(s) = \text{Med}[I_0(s), I_{40}(s), I_{80}(s), I_{120}(s), I_{160}(s)]$. This median filter led to the middle frame of Fig. 3 (b) which, after a spatial training, gave the result of Fig. 3 (c).

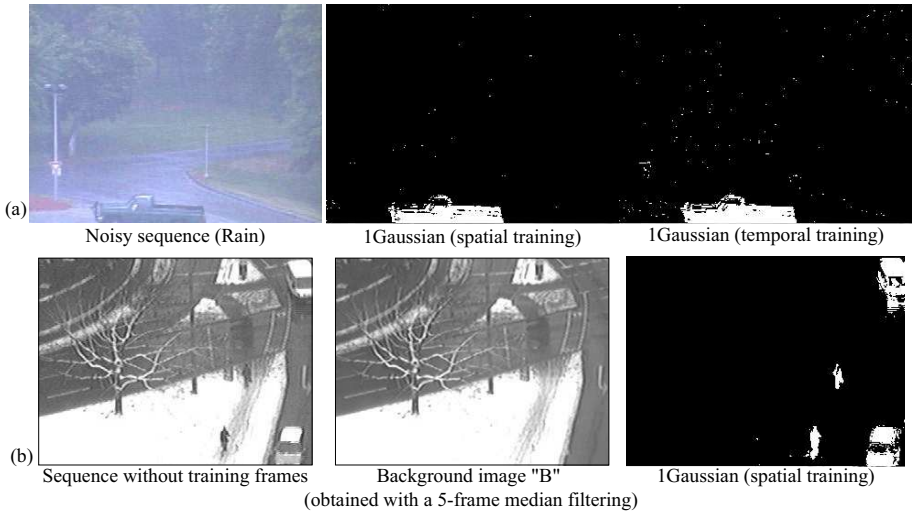

Fig. 3. *(a) A noisy sequence segmented with one Gaussian per pixel whose parameters have been spatially and temporally trained. (b) From a sequence without training frames absent of moving objects, a five-frame median filter has been used to produce the background frame B.*

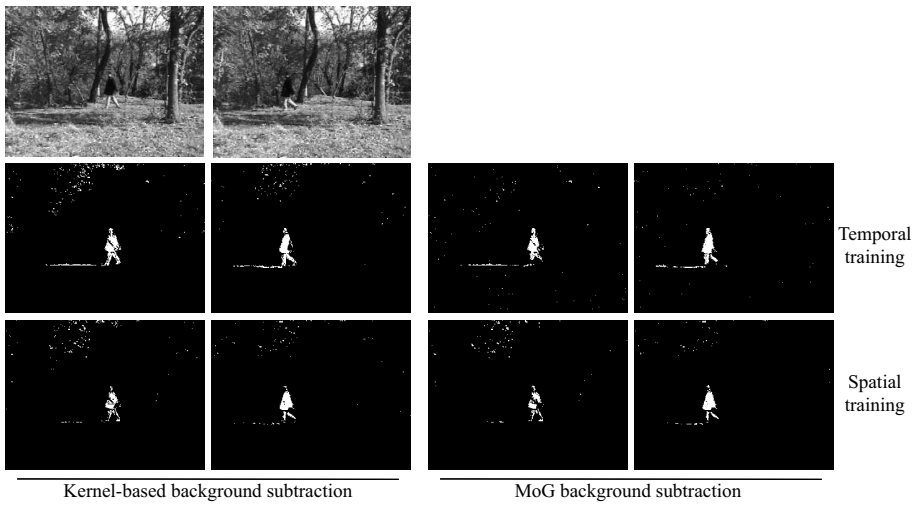

Fig. 4. *Two frames of a sequence exhibiting a multimodal background made of trees shaken by wind. The MoG and the kernel-based method have been trained either temporally on a series of frames or spatially, on single frame.*

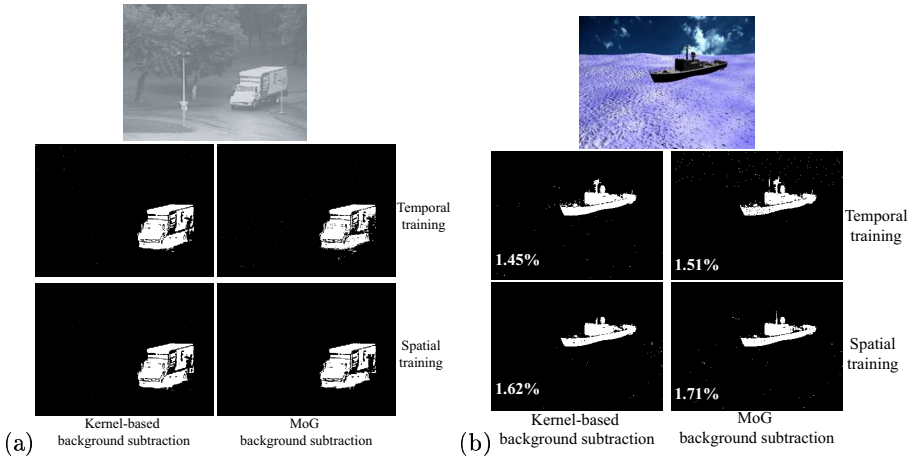

Fig. 5. *(a) Sequence shot with an unstable camera and (b) a synthetic sequence made of a boat sailing on a wavy sea. The numbers in the lower left corner of the boat label fields, indicate the average percent of mis-matched pixels. In each case, the MoG and the kernel-based method have been used. Both were trained either temporally on a series of frames or spatially on one frame.*

The third sequence we have segmented is the one shown in Fig. 4. This sequence has been shot with a perfectly static camera and contains a highly animated background made of trees shaken by wind. The sequence has been segmented with the MoG and the kernel-based background subtraction method presented in Section 3. Both have been trained temporally on a series of 150 frames and spatially on one single frame. Since the sequence shows a great deal of movement in the background, the parameters of each method have been updated at every frame following Section 3's explanations. In Fig.4, two frames are presented. As can be seen, the spatial results are very much similar to the temporal ones although the latter shows a bit more precision. In both cases, a few isolated false positives due to wind are present at the top, and false negatives due to camouflage can be seen on the walker. Notice that these isolated pixels can be efficiently eliminated with a basic 3×3 or 5×5 median filter.

The third sequence we have segmented is the one presented in Fig. 5 (a) which exhibits a moving truck. The sequence was shot with an unstable camera making the background highly multimodal (notice that for the purposes of this paper, additionnal carema jitter has been added to the original scene). Again, the MoG and the Kernel-based approach have been used to segment the sequence. Here, 20 frames were used for temporal training. Again, the results obtained after a temporal training are similar to the ones obtained after a spatial training. However, the latter seems a bit more precise, mostly because only 20 frames were available for temporal training. A larger number of training frames would have had certainly a positive influence on the results' sharpness (at the expense of processing time of course).

The fourth sequence shows a boat sailing on a wavy sea. The sequence consists of 200 frames from which the first 80 have been used for temporal training. The sequence has been computer generated and has a ground-truth label field for each frame. With these ground-truths, the average percentage of mis-matched pixels has been computed to empirically compare the four methods. The average percentage presented in Fig.5 (b) shows, again, how small is the difference between the spatially-trained and the temporally-trained methods.

5 Discussion

In this paper, a novel spatial framework for the background subtraction problem has been presented.

Our framework is based on the idea that, for some applications, the temporal distribution observed over a pixel corresponds to the statistical distribution observed spatially around that same pixel. We adapted three well known statistical methods to our framework and showed how these methods can be trained spatially instead of temporally.

Our framework offers three main advantages. First, the statistical parameters can be learned over one single frame instead of a series of frames as is usually the case for single-Gaussian or the MoG model. This has the advantage of requiering much less memory and being more flexible in presence of sequences having no training frames absent of moving objects. Second, as opposed to the kernel-based method, only one frame (instead of N) is kept in memory during runtime. This, again, is a major advantage memory-wise. Also, it makes our kernel-based methods much easier to be implemented on programmable graphics hardware [19] having a limited amount of memory. Last, but not least, our framework maintains the conceptual simplicity and strength of the background subtraction methods adapted to it. The segmentation 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 under a temporal framework.

Finally, we have shown that the results obtained with our framework on sequences with high noise, camera jitter and animated background are, to all practical ends, identical to the ones obtained with methods trained temporally.

6 Acknowledgment

The authors are very grateful to the group of Prof. Dr. H.-H. Nagel for the winter sequence of Fig. 3 (b) and to Dr. Ahmed Elgammal for the truck sequences of Fig. 3 (a) and Fig. 5 (a) and the walker sequence of Fig. 4.

References

1. McIvor A. Background subtraction techniques. In *Proc. of Image and Video Computing*, 2000.
2. 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*, volume 90, pages 1151–1163, 2002.

3. Friedman N. and Russell S.J. Image segmentation in video sequences: A probabilistic approach. In *UAI*, pages 175–181, 1997.
4. Cheung S.-c. and Kamath C. Robust techniques for background subtraction in urban traffic video. In *proc of the SPIE, Volume 5308*, pages 881–892, 2004.
5. Zhou D. and Zhang H. 2d shape measurement of multiple moving objects by gmm background modeling and optical flow. In *proc of ICIAR*, pages 789–795, 2005.
6. Rosin P. and Ioannidis E. Evaluation of global image thresholding for change detection. *Pattern Recognition Letters*, 24:2345–2356, 2003.
7. Wren C.R., Azarbayejani A., Darrell T., and Pentland A.P. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):780–785, 1997.
8. Jabri S., Duric Z., Wechsler H., and Rosenfeld A. Detection and location of people in video images using adaptive fusion of color and edge information. In *ICPR*, pages 4627–4631, 2000.
9. Stauffer C. and Grimson E.L. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):747–757, 2000.
10. Y. Dedeoglu. Moving object detection, tracking and classification for smart video surveillance. Master's thesis, Bilkent University, 2004.
11. Mittal A. and Paragios N. Motion-based background subtraction using adaptive kernel density estimation. In *Proc. of CVPR*, pages 302–309, 2004.
12. Ohya T. Matsuyama T. and Habe H. Background subtraction for non-stationary scenes. In *Proc. of 4th Asian Conference on Computer Vision*, pages 662–667, 2000.
13. Sheikh Y. and Shah. Bayesian object detection in dynamic scenes. In *Proc of CVPR*, pages 74–79, 2005.
14. Toyama K., Krumm J., Brumitt B., and Meyers B. Wallflower: Principles and practice of background maintenance. In *ICCV*, pages 255–261, 1999.
15. Zhong J. and Sclaroff S. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Proc of ICCV*, pages 44–50, 2003.
16. Huber P.J. *Robust Statistical Procedures, 2nd Ed.* SIAM, Philadelphia, 1996.
17. Bishop C. *Neural Networks for Pattern Recognition.* Oxford University Press, 1996.
18. Fukunaga K. *Introduction to statistical pattern recognition (2nd ed.).* Academic Press Professional, Inc., San Diego, CA, USA, 1990.
19. http://www.gpgpu.org/.