A SPATIAL SAMPLING MECHANISM FOR EFFECTIVE BACKGROUND SUBTRACTION

Marco Cristani, Vittorio Murino
Computer Science Dep., Università degli Studi of Verona, Strada Le Grazie 15, Verona, Italy
cristanm@sci.univr.it, vittorio.murino@univr.it

Keywords: Background subtraction, mixture of Gaussian, video surveillance

Abstract: In the video surveillance literature, background (BG) subtraction is an important and fundamental issue. In this context, a consistent group of methods operates at region level, evaluating in fixed *zones of interest* pixel values' statistics, so that a per-pixel foreground (FG) labeling can be performed. In this paper, we propose a novel hybrid, pixel/region, approach for background subtraction. The method, named Spatial-Time Adaptive Per Pixel Mixture Of Gaussian (S-TAPPMOG), evaluates pixel statistics considering zones of interest that change continuously over time, adopting a sampling mechanism. In this way, numerous classical BG issues can be efficiently faced: actually, it is possible to model the background information more accurately in the chromatic uniform regions exhibiting stable behavior, thus minimizing foreground camouflages. At the same time, it is possible to model successfully regions of similar color but corrupted by heavy noise, in order to minimize false FG detections. Such approach, outperforming state of the art methods, is able to run in quasi-real time and it can be used at a basis for more structured background subtraction algorithms.

1 Introduction

Background subtraction is a fundamental step in automated surveillance. It represents a pixel classification task, where the classes are the *background* (BG), i.e., the expected part of the monitored scene, and the *foreground* (FG), i.e., the interesting visual information (e.g., moving objects). As witnessed by the related literature (see Sect.2), choosing the right class cannot be adequately performed by per pixel methods, i.e., considering every temporal pixel evolution as an independent process. Instead, region based methods better behave, deciding the class of a pixel value by inspecting the related neighborhood.

In this paper, we propose a novel approach for background subtraction which constitutes a per region extension of a widely used and effective per pixel BG model, namely the Time Adaptive Per Pixel Mixture Of Gaussian (TAPPMOG) model. The proposed approach, called Spatial-TAPPMOG (S-TAPPMOG), is based on a sampling mechanism, inspired by the particle filtering paradigm. The goal of the approach is to provide a per pixel characterization of the BG which takes into account *selectively* for contributions coming from the neighboring pixel locations. The result is constituted by a set of per pixel models which are built per region: this characterization turns out to be very robust to false FG alarms, especially when the scene is heavily cluttered, and in general highly robust to the FG misses (i.e., not detected FG pixel values). In particular, several problems that classically affect BG subtraction schemes are successfully faced by the proposed method. Theoretical considerations and extensive comparative experimental tests prove the effectiveness of the proposed approach.
The rest of the paper is organized as follows. Section 2 reviews briefly the huge BG subtraction literature. In Section 3, the needed mathematical fundamentals, i.e., the TAPPMOG model and the particle filtering paradigm, are reported. The whole strategy is detailed in Section 4, and, finally, in Section 5, experiments on real data validate our method and conclude the paper.

2 State of the art

The actual BG subtraction literature is large and multifaceted; here we propose a taxonomy in which the BG methods are organized in i) per pixel, ii) per region, iii) per frame and iv) hybrid methods. Note that our approach is located in the hybrid method class.

The class of per pixel approaches is formed by methods that perform BG/FG discrimination by considering each pixel signal as an independent process. One of the first BG modeling was proposed in the surveillance system Pfinder (Wren et al., 1997), where each pixel signal was modeled as a uni-modal Gaussian distribution. In (Stauffer and Grimson, 1999), the pixel evolution is modeled as a multimodal signal, described with a time-adaptive mixture of Gaussian components (TAPPMOG). Another per-pixel approach is proposed in (Mittal and Paragios, 2004): this model uses a non-parametric prediction algorithm to estimate the probability density function of each pixel, which is continuously updated to capture fast gray level variations. In (Nakai, 1995), pixel value probability densities, represented as normalized histograms, are accumulated over time, and BG label are assigned by MAP criterion.

Region based algorithms usually divide the frames into blocks and calculate block-specific features; change detection is then achieved via block matching, considering for example fusion of edge and intensity information (Noriega and Bernier, 2006). In (Heikkila and M.Pietikainen, 2006) a region model describing local texture characteristic is presented; the method is prone to errors when shadows and sudden global changes of illumination occur.

Frame level class is formed by methods that look for global changes in the scene. Usually, they are used jointly with other pixel or region BG approaches. In (Stenger et al., 2001), a graphical model was used to adequately model illumination changes of the scene. In (Ohta, 2001), a BG model was chosen from a set of pre-computed ones, in order to minimize massive false alarm.

Hybrid models describe the BG evolution using jointly pixel and region models, and adding in general post-processing steps. In Wallflower (Toyama et al., 1999), a 3-stage algorithm is presented, which operates respectively at pixel, region and frame level. Wallflower test sequences are widely used as comparative benchmark for BG subtraction algorithms. In (Wang and Suter, 2006), a non parametric, per pixel FG estimation is followed by a set of morphological operations in order to solve a set of BG subtraction common issues. In (Kottow et al., 2004) a region level step, in which the scene is modeled by a set of local spatial-range codebook vectors, is followed by an algorithm that decides at the frame-level whether an object has been detected, and several mechanisms that update the background and foreground set of codebook vectors.

3 Fundamentals

3.1 The TAPPMOG background modeling

In this paradigm, each pixel process is modeled using a set of R Gaussian distributions. The probability of observing the value $z^{(t)}$ at time t is:

$$P(z^{(t)}) = \sum_{r=1}^{R} w_r^{(t)} \mathcal{N}(z^{(t)} | \mu_r^{(t)}, \sigma_r^{(t)}) \qquad (1)$$

where $w_r^{(t)}$, $\mu_r^{(t)}$ and $\sigma_r^{(t)}$ are the mixing coefficients, the mean, and the standard deviation, respectively, of the r-th Gaussian $\mathcal{N}(\cdot)$ of the mixture associated with the signal at time t. The Gaussian components are ranked in descending order using the w/σ value: the most ranked components represent the "expected" signal, or the background.

At each time instant, the Gaussian components are evaluated in descending order to find the first matching with the observation acquired (a *match* occurs if the value falls within 2.5σ of the mean of the component). If no match occurs, the least ranked component is discarded and replaced with a new Gaussian with the mean equal to the current value, a high variance σ_{init}, and a low mixing coefficient w_{init}. If r_{hit} is the matched Gaussian component, the value $z^{(t)}$ is labeled FG if

$$\sum_{r=1}^{r_{\text{hit}}} w_r^{(t)} \; > \; T \qquad (2)$$

where T is a standard threshold. We call this assignment as the *FG test*.

The equation that drives the evolution of the mixture's weight parameters is the following:

$$w_r^{(t)} = (1-\alpha) w_r^{(t-1)} + \alpha M^{(t)}, 1 \leq r \leq R, \quad (3)$$

where $M^{(t)}$ is 1 for the matched Gaussian (indexed by r_{hit}) and 0 for the others, and α is the learning rate. The other parameters are updated as follows :

$$\mu_{r_{\text{hit}}}^{(t)} = (1-\rho)\mu_{r_{\text{hit}}}^{(t-1)} + \rho z^{(t)} \quad (4)$$

$$\sigma_{r_{\text{hit}}}^{2\,(t)} = (1-\rho)\sigma_{r_{\text{hit}}}^{2\,(t-1)} + \rho(z^{(t)}-\mu_{r_{\text{hit}}}^{(t)})^T(z^{(t)}-\mu_{r_{\text{hit}}}^{(t)}) \quad (5)$$

where $\rho = \alpha\mathcal{N}(z^{(t)}|\mu_{r_{\text{hit}}}^{(t)}, \sigma_{r_{\text{hit}}}^{(t)})$.

3.2 Particle filtering paradigm

The particle filtering (PF) paradigm (Isard and Blake, 1998) is a Bayesian approach that assumes that all information obtainable about the model $X^{(t)}$ is encoded in the set of observations $Z^{(t)}$. Such information can be extracted evaluating the posterior distribution $P(X^{(t)}|Z^{(t)})$. This probability is approximated using a set of samples $\{x^{(t)}\}$, where each sample represents an instance of the model $X^{(t)}$. The algorithm that performs particle filtering, in its general formulation, follows at each time instant t a set of rules for propagating the set of samples:
1) *sampling from prior (the posterior of step $t-1$)*: M samples are chosen from $\{x^{(t-1)}\}$ with probability $\{w^{(t-1)}\}$, obtaining $\{x^{(t)}\}$. In this way, samples with high probability at time t-1 have higher probability to "survive";
2) *prediction*: samples $\{x^{(t)}\}$ are propagated using a model dynamics; typically, this dynamics also contains a stochastic component;
3) *weighting*: samples obtained by previous step are evaluated considering the observations obtained at time t, i.e., $Z^{(t)}$, calculating the likelihood $P(Z^{(t)}|X^{(t)})$; at each sample $x^{(t)}$ is then assigned the weight $w^{(t)}$, proportional to the likelihood value.

4 The proposed method: S-TAPPMOG

Our approach models the visual evolution of the observed scene using a set of communicating per-pixel processes. Roughly speaking, the basis of the approach is a TAPPMOG scheme, where each pixel is modeled by a mixture of Gaussian components. The novelty of our method is that the per-pixel parameters are updated considering not only per-pixel observations, but also observations coming from the neighborhood zone throughout a sampling process. In details, we have four steps, whose last three are inspired by the PF paradigm[1]:

1) *per-pixel* step (see Fig.1a): at each location i, the classical FG test is performed; this step gives an initial estimation of the class {BG,FG} of the gray value $z_i^{(t)}$, and individuates a Gaussian component that models such value, indexed with $r_{i,hit}$ and with mean parameter $\mu_{r_{i,\text{hit}}}^{(t)}$, standard deviation $\sigma_{r_{i,\text{hit}}}^{(t)}$ and weighting coefficient $w_{r_{i,\text{hit}}}^{(t)}$;
2) *sampling from prior* step (see Fig.1b): if the value $z_i^{(t)}$ is labeled as FG (the FG test is applied, see Sect.3.1), no further analysis is applied; viceversa, if the value $z_i^{(t)}$ is estimated as BG, it is duplicated in a set of copies $\{x_i^{(t)}\}$. The number of sample produced M_{Sent} is proportional to the weight of the $r_{i,\text{hit}}$-th Gaussian component (which explains the certainty degree that a component models a BG signal, see Sect.3.1), i.e.,

$$M_{\text{Sent}} = \lceil \gamma_{\max} w_{r_{i,\text{hit}}}^{(t)} \rceil \quad (6)$$

where $\gamma_{\max}$ is the maximum number of samples that can be generated from a pixel location;
3)*prediction* step (see Fig.1b): the sampled values are spatially propagated at positions that follow a 2D Gaussian distribution (opportunely rounded to the nearest integer in order to be conform to the pixel locations lattice), with mean located at the pixel location i and spherical covariance matrix $\bar{\sigma}_i\mathbf{I}$ with

$$\bar{\sigma}_i = \rho_{\max} w_{r_{i,\text{hit}}}^{(t)} \quad (7)$$

where $\rho_{\max}$ is the maximum spatial standard deviation allowed.
4) *weighting* step (see Fig.1c,d); let $\{j\}$ be a set of pixel locations, whose pixel values $\{z_j^{(t)}\}$ are all classified as BG *after* the step 1, and let $\{x_j^{(t)}\}$ the values propagated from $\{j\}$ after the step 3. Now, considering the location i, let $\{\tilde{x}_j^{(t)}\}$ be the set of samples arrived at location i that are matched by the Gaussian component $r_{i,\text{hit}}$ (see Sect.3.1 for a formal definition of matching), and $\{\tilde{j}\}$ the locations that produced $\{\tilde{x}_j^{(t)}\}$. At this point, the following comments can be noticed: a) $\{\tilde{x}_j^{(t)}\}$ together with $z_i^{(t)}$ represent values which model a neighborhood zone $\{\tilde{j}\} \cup i$ characterized by a similar chromatic aspect; b) the visual aspect of such zone can be modeled by the mean value $\tilde{\mu}_i^{(t)}$ calculated from $\{\tilde{x}_j^{(t)}\} \cup z_i^{(t)}$; c) the degree

[1]Formal similarities of our algorithm with the PF paradigm hold mostly on steps 2 (∼step 1 of the PF) and 3 (∼step 2 of the PF); the step 4 (∼step 3 of the PF), as we can see in this section, approximates the non-parametric density modeled by $\{x_i^{(t)}\}$ with a Gaussian distribution. A slight different and more elegant theoretical explanation of our sampling method is currently under work.

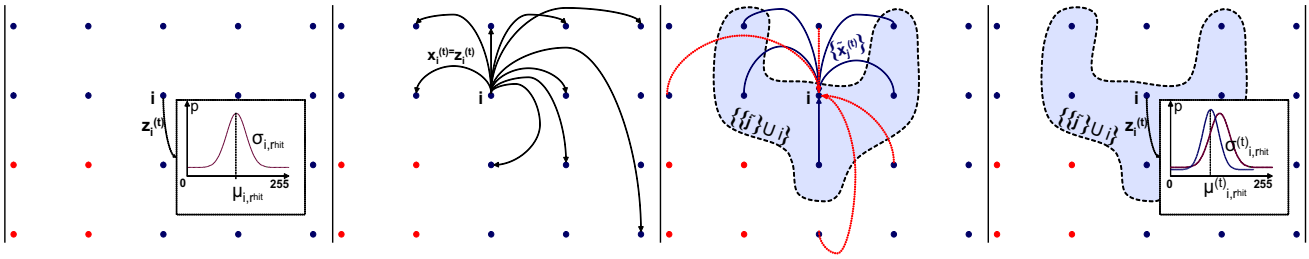


Figure 1: Overview of the proposed method: in all the figures, a set of pixel locations is depicted as a regular grid of points. a) step 1: in red the pixels discovered as FG values, in blue the BG values. In the box, the Gaussian component $r_{i,hit}$ matched at time t, representing the signal $z_i^{(t)}$; b) steps 2 and 3: a set of samples $\{x_i^{(t)}\}$ is generated from location i and propagated in a Gaussian neighborhood; c)step 4: a subset of the samples $\{x_j^{(t)}\}$ arrived at location i from locations $\{j\}$, that match with the Gaussian component $r_{i,hit}$ (note the blue-solid arrows), create the region formed by locations $\{\tilde{j}\} \cup i$. The region is highlighted in blue. The matching samples are called $\{\tilde{x}_j^{(t)}\}$; d) step 4: the samples $\{x_j^{(t)}\} \cup z_i^{(t)}$ concur to create the new Gaussian parameters for the location i.

of intra-similarity of $\{\tilde{x}_j^{(t)}\} \cup z_i^{(t)}$ can be modeled by evaluating the standard deviation of this set, let's say $\tilde{\sigma}_i^{(t)}$. If such value is very low, it means that the locations $\{\tilde{j}\} \cup i$ model a spatial portion of the scene which can be considered with high certainty as a single entity, with a well defined chromatic aspect. Therefore, we want to include this information in the final per-pixel modeling.
If $\tilde{\sigma}_i^{(t)}$ is very high, it means that the locations $\{\tilde{j}\} \cup i$ represent a zone which can be considered as a whole (actually, the locations are modeled by a single Gaussian component), but with a high variability, due most probably to heavy (Gaussian) noise. Therefore, the per-pixel models have to take into account for this spatial uncertainty. As additional example, an intermediate $\tilde{\sigma}_i^{(t)}$ can be due to a light chromatic gradient in a local region of the scene.
In other words, all the values assumed by $\tilde{\sigma}_i^{(t)}$ model smoothly a degree of uncertainty in considering the $\{\tilde{j}\} \cup i$ as a single entity.
All these considerations can be embedded in the weighting step by updating the per-pixel Gaussian parameters as follows:

$$w_{r_{\text{hit}}}^{(t)} = (1-\zeta) w_{r_{\text{hit}}}^{(t-1)} + \zeta \quad (8)$$

$$\mu_{r_{\text{hit}}}^{(t)} = (1-\zeta) \mu_{r_{\text{hit}}}^{(t-1)} + \zeta \tilde{\mu}_i^{(t)} \quad (9)$$

$$\sigma_{r_{\text{hit}}}^{(t)} = (1-\zeta) \sigma_{r_{\text{hit}}}^{(t-1)} + \zeta\ \tilde{\sigma}_i^{(t)} \quad (10)$$

where

$$\zeta = \alpha M_{\text{Rec}} \quad (11)$$

with $M_{\text{Rec}} = \| \{\tilde{x}_j^{(t)}\} \cup z_i^{(t)} \|$, and α is the learning rate of the process.
In this way, a pixel value that belongs to the background with more certainty sends more messages in a wider zone, influencing consequently the class labeling of the neighborhood. In the next section, further considerations about the method will be provided.

5 Results

Our algorithm has been applied to two different datasets; the first one is the "Wallflower" benchmark dataset; [2] the second one is composed by sequences depicting heavily cluttered outdoor scenarios.[3] As qualitative and quantitative comparisons, we present some results provided by recent and effective BG subtraction algorithms.
As general remarks of this section, please note that i)our method is completely free from high-level post-processing operations (e.g., blob analysis with morphological operators); ii) our method requires a computational effort similar to TAPPMOG ($O(NR)$, where N is the number of pixels and R is the number of Gaussian components, while S-TAPPMOG has complexity $O(N(R + M_{\text{Sent}}))$): this implies that our method can be intended as basic operation for structured applications of BG subtraction, so as TAPPMOG.

5.1 Wallflower dataset

The dataset contains 7 real video sequences, each one of them presenting a typical BG sub-

[2] Downloadable at `http://research.microsoft.com/users/jckrumm/WallFlower/TestImages.htm`.

[3] Downloadable at `http://i21www.ira.uka.de/image_sequences/`.

traction issue. The sequences are provided wi a frame manually segmented, representing t ground truth. Here, we processed four of t most difficult sequences, i.e., sequences for whi the results presented in literature are far from t ground truth.

The sequences are: 1) *Waving Tree* (WT): a tr is swaying and a person walks in front of the tre 2) *Camouflage* (C): a person walks in front of monitor, which has rolling interference bars the screen. The bars include color similar to t persons clothing; 3) *Bootstrapping* (B): the ima sequence shows a busy cafeteria and each frar contains people; 4) *Foreground Aperture* (FA): person with uniformly colored shirt wakes up a begins to move slowly.

All the RGB sequences are captured at resol tion of 160×120 pixels. After an easy initial step of parameters tuning, we fix a parameters set for the whole experimental evaluation. In details, we choose $\alpha = 0.005$, $\mu_{\text{init}} = 0.01$, $\sigma_{\text{init}} = 7.5$, and $\gamma_{\max} = 20$, $\rho_{\max} = 7$ (see Eq.6 and Eq.7 respectively).

In order to give a practical explanation of our method, we focus first on the WT sequence. In this sequence, 286 frames long, an outdoor situation is captured, in which a tree is manually kept oscillating, with strong oscillations that span a big portion of the scene. Here, the difficulty lies in the fact that, fixed a pixel in the center of the scene, the evolution profile of the related RGB signal is highly irregular and thus labeled as FG, due to the frequent occlusions of the tree. It turns out that the tree, which is intuitively a BG object, tends to remain labeled as FG. In Fig.2, an explicative comparison between TAPPMOG and our method is proposed, where the parameters of TAPPMOG are the same of the ones used in our method, except $\rho_{\max}$ and $\sigma_{\max}$, absent in TAPPMOG. Here, the standard deviations of the Gaussian components that model the pixel signals related to locations A and B are presented. For ease the visualization, only the R channel is considered, and only the frame interval $[100, 150]$ is analyzed.

At frame 108, locations A and B are focused both on the sky, but B depicts a zone more affected by color variations, due to the tree presence. In the two plots below the images, it can be noted that, at frame 108, the standard deviation value assumed by S-TAPPMOG is lower than the correspondent TAPPMOG value, highlighting the better precision with which S-TAPPMOG models a wide and uniform aspect of the scene, i.e., the

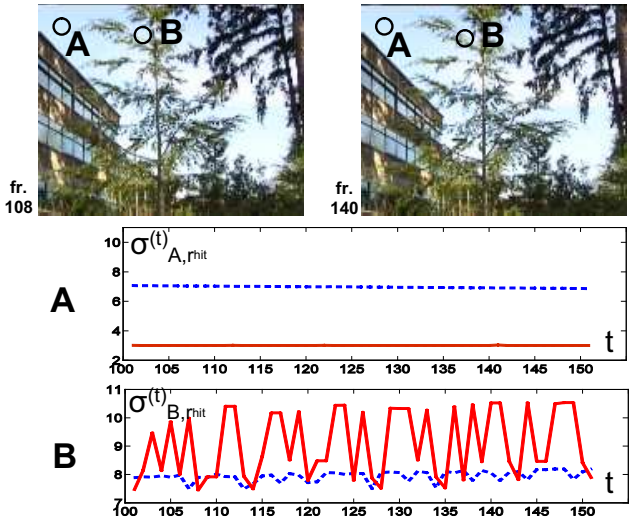


Figure 2: Evaluation of the standard deviation of the Gaussian components of the TAPPMOG (blue-dashed line) and S-TAPPMOG (red-solid line) models. Top: two frames (108 and 140) of the WT sequence. Bottom: the frame evolution of the standard deviations $\sigma^{(t)}_{r_{\text{hit}}}$ which characterize the Gaussian components modeling the pixel signal related to location A (top plot) and location B (bottom plot).

sky. At frame 140, location A presents again the sky, while in location B the tree is passing over. As a consequence, in the two plots below, we can see that our method models the signal representing the tree with higher standard deviation as compared to the correspondent TAPPMOG value. This indicates that S-TAPPMOG permits the tree of assuming a larger spectra of signal values, thus diminishing the presence of false FG positives.

Qualitative results obtained by our method with the WT sequence, together with the other dataset sequences and compared with the TAPPMOG method are present in Fig.3. Note that the parametrization chosen permits to TAPPMOG to obtain a better error rate than the one reported in (Toyama et al., 1999) for the same sequences. Quantitative results, in terms of false positives (per-pixel false FG detections) and false negatives (missed FG detections) with respect to other state of the art methods are visible in Fig.4. In particular, Wallflower, SACON, Tracey Lab LP, Bayesian Decision, and TAPPMOG refer to (Wang and Suter, 2006; Kottow et al., 2004; Nakai, 1995; Stauffer and Grimson, 1999), respectively, which have been previously discussed in Sect.2. As visible in Fig.4, our method outperforms globally Wallflower, Bayesian decision

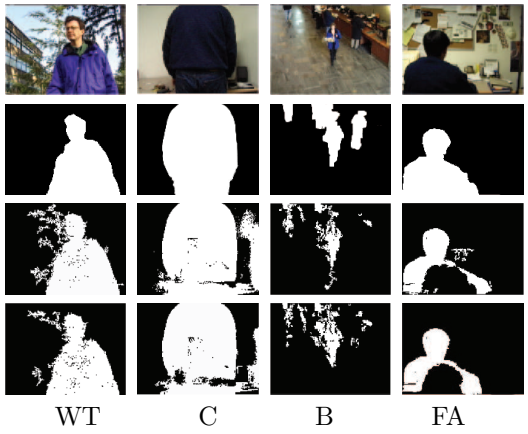


Figure 3: Wallflower qualitative results: on the first row, the frames of the different sequences for which a ground truth is provided; on the second row the ground truth; the third row presents the TAPPMOG results and, finally, results obtained with our method S-TAPPMOG are reported on the last row.

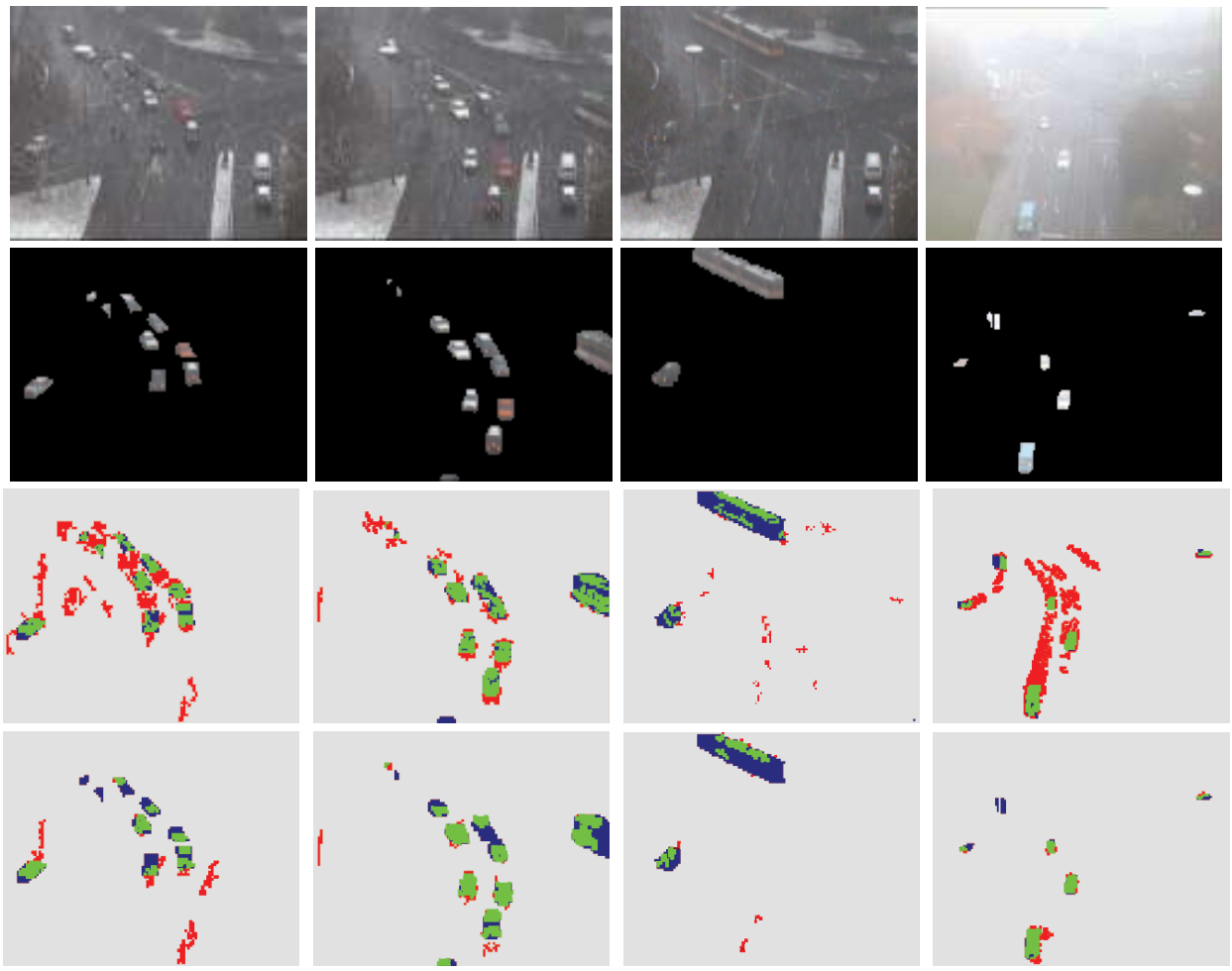
Methods	*Err.*	*WT*	*C*	*B*	*FA*	*T.Err.*
Wallflower	f.neg.	877	229	2025	320	9170
	f.pos.	1999	2706	365	649	
	t.e.	2876	2935	2390	969	
SACON	f.neg.	41	47	1150	1508	4084
	f.pos.	230	462	125	521	
	t.e.	271	509	1275	2029	
Tracey LAB LP	f.neg.	191	1998	1974	2403	7219
	f.pos.	136	69	92	356	
	t.e.	327	2067	2066	2759	
Bayesian decision	f.neg.	629	1538	2143	2501	14043
	f.pos.	334	2130	2764	1974	
	t.e.	963	3688	4907	4485	
TAPPMOG	f.neg.	56	220	1732	2217	10059
	f.pos.	1533	2398	1033	870	
	t.e.	1589	2618	2765	3087	
STAPPMOG	f.neg.	153	643	1414	1912	7844
	f.pos.	1152	1382	811	377	
	t.e.	1305	2025	2225	2289	

Figure 4: Quantitative results obtained by the proposed S-TAPPMOG method: *f.neg*,*f.pos.*,*t.e.*,and *T.Err* mean false negative, false positive per-pixel FG detections, total errors on the specific sequence and total errors summed on all the sequences analyzed, respectively. Our method outperformed the most effective general purposes BG subtraction scheme (Wallflower, Bayesian decision, TAPPMOG), and is comparable with methods which are more time demanding and strongly constrained by data-driven initial hypotheses (SACON and Tracey Lab LP).

and TAPPMOG methods, providing also good results with respect to Sacon and Tracey Lab LP methods, which are however more structured and time demanding techniques, tightly constrained to initial hypotheses. Please note that we did not report the good results reached in (H. Wang, 2005), because we are not convinced deeply about the RGB normalized signal modeling proposed in that paper. There, the RGB-normalized signal covariance matrix was modeled as a diagonal matrix, while this fact is not correct, as mentioned in (Mittal and Paragios, 2004).

5.2 "Traffic" dataset

This dataset is formed by outdoor traffic sequences. We focus on two of them, the "Snow" and the "Fog" sequences, which are characterized by very hard weather conditions, see Fig.5, first row.
As comparison against our method, we apply the TAPPMOG algorithm, choosing the following parameters set: $\alpha = 0.005$, $\mathbf{w}_{\text{init}} = 0.01$, $\sigma_{\text{init}} = 7.5$. With the same parameters setting, we apply the S-TAPPMOG algorithm with $\gamma_{\max} = 20$ and $\rho_{\max} = 7$. In order to speed up the processing, we down-sample both the sequences reducing them to 160×120 pixel frames, obtaining performances of 8 frames per sec. with the TAPPMOG method and 6 frames per sec. with the S-TAPPMOG algorithm, with MATLAB not-optimized code.
Some qualitative results are shown in Fig.5. In general, TAPPMOG method produces a large amount of false FG detections. The following considerations explain this phenomenon. In the "Snow" sequence (please refer to Fig.5, first three columns), the scene can be modeled by a bimodal BG, i.e., one mode modeling the outdoor environment, and the other modeling the snow. The snow generates a high-variance color intensity pattern, which can be intended as a spatial texture (i.e., a pattern which globally cover the scene). Modeling this texture by taking into account for signals coming from different close positions is equivalent to better capture the intrinsic high variance of the appearance of the snow. As an example, see the red false FG detections in the related figures, which are globally fewer than in the TAPPMOG approach. In particular, in Fig.5a, the snow causes more false FG detections in the center of the scene with the TAPPMOG model.
At the same time, the other component modeling the clean environment (not corrupted by the snow), can be learnt more precisely (with a smaller standard deviation), refining the per-pixel signal estimation with the neighboring similar pixels signals. Looking at Fig.5b), one can note that the car on the bottom is not discovered by TAPPMOG approach, whereas it is partially detected by S-TAPPMOG. A similar observation can be assessed by observing the car on Fig.5c, which is better modeled by S-TAPPMOG.
In any case, the per-region analysis of the S-TAPPMOG brings a side effect: when a white object passes over the scene, this can be absorbed by the white large variance BG Gaussian component which characterizes the snow, causing a FG miss. This is visible in Fig.5a, where the first car



a) fr.34 b) fr.118 c) fr.284 d) fr.284

Figure 5: "Traffic" dataset results: three frames of the "Snow" sequence and one frame of the "Fog" sequence (first row); hand-segmented ground truth (second row); TAPPMOG method (third row); our method (fourth row). In the two last rows (the figure will be printed in color), green pixels mean correct FG detections, red pixels mean false FG detections (false positives), and blue pixels mean undetected FG pixels (false negatives).

from the top is partially covered by the lamp on the upper left part of the image and some gray part of the tram on Fig.5b and Fig.5c.
As visual explanation of how differently the two methods model the scene, please refer to Fig.6. From the images depicting the σ values, it is visible that our method permits to better extract FG objects where the scene is more uniform, e.g., the street, whereas in the zones in which the scene can be confused with the snow, standard deviation values are higher. As a comparison, in the corresponding images of the TAPPMOG method, no spatial distinction is made in the FG discrimination, and, in general, the value of the standard deviation is higher. From the μ images, in S-TAPPMOG, we can see that the FG objects better protrude with respect to the rest of the BG scene. This means that the mean values that characterize FG and BG objects are better differentiated by S-TAPPMOG with respect to the TAPPMOG method. Similar considerations can be stated for the "Fog" sequence (refer to Fig.5, third column). Here, the scene can be characterized by a bimodal BG, where one component models the scene heavily occluded by the fog, and the other explains the scene when the fog drastically diminishes, due to the characteristic dynamics of the fog banks. In this case, the low-variance, per-pixel Gaussian components are not able to model sudden local changes of fog intensity, while the S-TAPPMOG model works better. Nevertheless, in some cases white FG objects are

Seq.	TAPPMOG err.	S-TAPPMOG err.
"Snow"	2253	1807
"Fog"	1501	845

Table 1: Accuracy test for the "Snow" and "Fog" sequences, in terms of total errors.

more difficult to discover for S-TAPPMOG than for the TAPPMOG method. In order to test

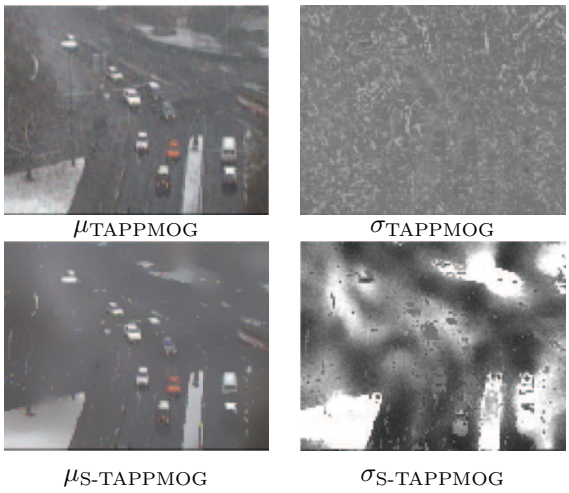


Figure 6: Different modeling for the frame 118 of the "Snow" sequence, performed by TAPPMOG and S-TAPPMOG. In the μ images the mean value of the Gaussian component modeling the signal is depicted for each pixel. The same holds for the σ images, where brighter pixels correspond to higher standard deviation values.

quantitatively the two algorithms, we perform a manual counting operation for each original frame of the two sequences, extracting the number of separated objects moving on the scene.
For each frame we manually label with a mark the center of each distinct moving object. Then, using a connected components operator, we extract the FG blobs from each output frame found by the two algorithms. After that, we control if each blob intersects one FG mark manually annotated. If a FG blob does not intersect any mark, we annotate a false FG detection and if a mark remains uncovered, we annotate a FG miss. The summation of all false negatives and false positives gives the total error rate, shown in Tab.1. This test can give an idea on how our method performs when embedded in a multi-object tracking framework, where the separation of different objects plays an important role in the data association. As visible by the results, in both the cases the errors are less for S-TAPPMOG. This, together with the analysis done with the Wallflower dataset, demonstrates the qualities of the proposed approach.

References

H. Wang, D. S. (2005). A re-evaluation of mixture of gaussian background modeling. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2005 (ICASSP '05)*, volume 2, pages ii/1017– ii/1020.

Heikkila, M. and M.Pietikainen (2006). A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):657–662.

Isard, M. and Blake, A. (1998). CONDENSATION: Conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 29(1):5–28.

Kottow, D., Köppen, M., and del Solar, J. (2004). A background maintenance model in the spatial-range domain. In *ECCV Workshop SMVP*, pages 141–152.

Mittal, A. and Paragios, N. (2004). Motion-based background subtraction using adaptive kernel density estimation. In *CVPR '04: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 302–309. IEEE Computer Society.

Nakai, H. (1995). Non-parameterized bayes decision method for moving object detection. In *Proc. Second Asian Conf. Computer Vision*, pages 447–451.

Noriega, P. and Bernier, O. (2006). Real time illumination invariant background subtraction using local kernel histograms. In *Proc. of the British Machine Vision Conference*.

Ohta, N. (2001). A statistical approach to background subtraction for surveillance systems. In *Int. Conf. Computer Vision*, volume 2, pages 481–486.

Stauffer, C. and Grimson, W. (1999). Adaptive background mixture models for real-time tracking. In *Int. Conf. Computer Vision and Pattern Recognition (CVPR '99)*, volume 2, pages 246–252.

Stenger, B., nad N. Paragios, V. R., F.Coetzee, and Buhmann, J. M. (2001). Topology free hidden Markov models: Application to background modeling. In *Int. Conf. Computer Vision*, volume 1, pages 294–301.

Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. In *Int. Conf. Computer Vision*, pages 255–261.

Wang, H. and Suter, D. (2006). Background subtraction based on a robust consensus method. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, pages 223–226, Washington, DC, USA. IEEE Computer Society.

Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.