International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Rain or Snow Detection in Image Sequences through use of a Histogram of Orientation of Streaks

Jérémie Bossu · Nicolas Hautière · Jean-Philippe Tarel

Received: date / Accepted: date

Abstract The detection of bad weather conditions is crucial for meteorological centers, specially with demand for air, sea and ground traffic management. In this article, a system based on computer vision is presented which detects the presence of rain or snow. To separate the foreground from the background in image sequences, a classical Gaussian Mixture Model is used. The foreground model serves to detect rain and snow, since these are dynamic weather phenomena. Selection rules based on photometry and size are proposed in order to select the potential rain streaks. Then a Histogram of Orientations of rain or snow Streaks (HOS), estimated with the method of geometric moments, is computed, which is assumed to follow a model of Gaussian-uniform mixture. The Gaussian distribution represents the orientation of the rain or the snow whereas the uniform distribution represents the orientation of the noise. An algorithm of expectation maximization is used to separate these two distributions. Following a goodness-of-fit test, the Gaussian distribution is temporally smoothed and its amplitude allows deciding the presence of rain or snow. When the presence of rain or of snow is detected, the HOS makes it possible to detect the pixels of rain or of snow in the foreground images, and to estimate the intensity of the precipitation of rain or of snow. The applications of the method are numerous and include the detection of critical weather conditions, the observation of weather, the reliability improvement of video-surveillance systems and rain rendering.

Keywords Rain or snow detection · Geometric moments · Expectation maximization · Histogram · Visual surveillance

1 Introduction

It is well known that for meteorological centers the detection of bad weather conditions is crucial, specially with demand for air, sea and ground traffic management. The detection and the characterization of weather conditions usually involves dedicated sensors, such as a visibilitymeter, a disdrometer or a radar, etc. However, the cost of these sensors sets a limit to their extensive deployment. As pointed out by Jacobs et al. [1], the massive deployment of video-surveillance cameras for security and safety reasons is an opportunity for using them to monitor weather conditions. The process of image formation in foggy weather [2] and the fog detection [3] have already been investigated. The constituent particles of rain are larger than those of fog, and the individual particles may be visible. A distribution of such drops falling at high velocities results in time varying intensity fluctuations in images and videos. In addition, due to the settings of the camera used to acquire the images, intensities due to rain are motion blurred and therefore depend on the background. Thus, the visual effects of rain are a combined result of the dynamics of rain and of the photometry of the environment. Stochastic models that capture the spatial and temporal effects of rain on vision are proposed by Garg and Nayar [4].

In this article, we address dynamic weather conditions such as rain, hail and snow, which are simply denoted as rain in the following of the article. In particular, a method is proposed to detect the presence of rain in images grabbed by fixed cameras. This method is a core component of a camera-based sensing system which aims at detecting the weather conditions such as fog and rain [5], while being able to run classical video surveillance applications, like pedestrian detection or road traffic monitoring. This application context gives us operating constraints for our computer vision system and leads us to use black and white images

---

J. Bossu · N. Hautière · J.-P. Tarel
Université Paris-Est, LEPSIS, INRETS-LCPC, 58 boulevard Lefebvre, 75015 Paris, France
Tel.: +33-1-40436519
Fax: +33-1-40435499
E-mail: nicolas.hautiere@lcpc.fr

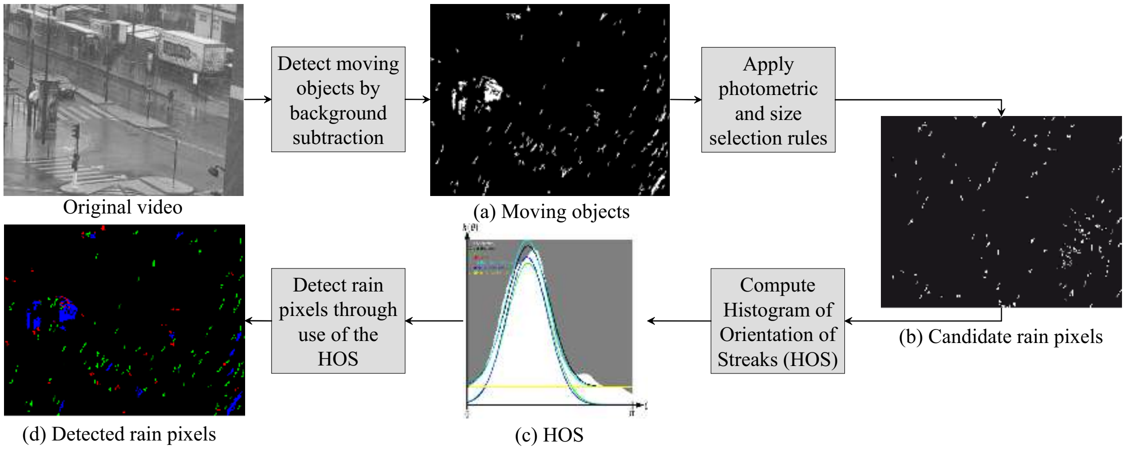


Fig. 1 The rain detection process applied to a video: (a) Detection of moving objects by background subtraction; (b) segmentation of candidate rain pixels by applying size and photometric selection rules; (c) computation of the Histogram of Orientation of Streaks (HOS) by accumulating the orientation obtained by the method of geometric moments of the different connected components; rain presence decision is taken based on the temporal stability as well as on the shape of the HOS; (d) detection of rain pixels using the HOS.

grabbed by fixed cameras in focus at the infinity with automatic exposure control.

Existing approaches dealing with rain mainly focus on the segmentation and the removal or addition of rain pixels in images for rendering applications [4, 6–9]. The first approach proposed by Hase et al. [6] uses a temporal median filter exploiting the fact that pixels are unaffected by rain most of the time. This idea was extended by Zhang et al. [7] who propose to detect rain with k-means clustering ($k = 2$), adding chromatic constraints so as to reduce the false detections. However, this method cannot be used online due to the k-means clustering algorithm which needs to be applied on the entire video sequence. Garg and Nayar [4] propose a method which uses the optical properties of a falling rain drop, so as to segment the video scenes in rain and non-rain regions. In addition, false detections are reduced by using a photometric constraint which models the appearance of rain streaks. Brewer and Liu [8] detect rain streaks by detecting their intensity spikes. They reduce false detections based on the aspect ratio and the orientation of the rain streaks. Barnum et al. [9] propose a global appearance model to identify the typical behavior of rain in the frequency-domain and present a filter in the frequency-domain to reduce or increase its visibility.

Contrary to previous approaches, which do not explicitly track the rain particles between the frames, Sakaino et al. [10] estimate the motion of falling snow based on a motion estimation method for semi-transparent objects with a long-range displacement between frames. This approach seems limited to the tracking of snowflakes. Finally, Halimeh and Roser [11] propose a photometric model for a raindrop on a car windshield and use it to detect rain presence for driver assistance purposes.

In the visual surveillance context, only the methods from Hase et al. [6], Garg and Nayar [4], Brewer and Liu [8] and Barnum et al. [9] are relevant. However, their methods do not directly address the detection of rain events. Consequently, their methods lead to the over-segmentation of rain pixels and cannot be used alone to detect rain presence. For this purpose, we develop a complementary method, which is based on a segmentation into blobs, whose core component is a histogram of orientation of streaks. For the segmentation, we use a classical background subtraction method which can be seen as a generalization of the temporal median filter [6], of the k-means temporal clustering [7] and of the photometric constraint proposed by Garg and Nayar [4]. It allows also an easier implementation of the method on existing visual surveillance systems. Finally even if it is not our primary objective, the method offers some advantages over existing methods of restoration of images altered by rain.

The outline of our method is illustrated in Fig. 1. First, we segment the potential rain pixels. Our approach relies on a classical background subtraction method [12] but the approaches issued from [4, 9] would also suit the purpose. We reduce false detections by applying selection rules based on photometry and size. Then, we build a so-called Histogram of Orientation of Streaks (HOS) by accumulating the orientations of the different connected components obtained by the method of geometric moments, which is also used in [8]. The data of this histogram are then modeled as a Gaussian-uniform mixture, whose parameters are estimated using an expectation maximization (EM) algorithm. A goodness-of-fit test allows deciding if the model fits the HOS well. If it is the case, a temporal smoothing filter is applied to the parameters of the Gaussian-uniform mixture. A decision criterion on the smoothed histogram then allows detecting the presence or absence of rain. When rain is detected, the rain pixels can be detected accurately and easily in the images and rain intensity can be estimated as well. The applications of the proposed method are numerous, for instance: detection of critical weather conditions for road safety, weather observation, improvement of the reliability of video-surveillance systems and rain rendering. Different experimental validations are presented to assess the performance of the proposed approach and to illustrate how complementary our method is compared with existing methods.

This article is organized as follows. In section 2, we present the process of segmentation of potential rain pixels. In section 3, we present the construction of the HOS. In section 4, we present the EM algorithm to estimate the parameters of the HOS. In section 5, we present the decision criterion, which is used to decide rain presence. Once rain presence is detected, it is easy to detect rain pixels as described in section 6. In section 7, we present a method to estimate the rain intensity using the HOS directly. In section 8, we present applications of the method. In Section 9, we evaluate how our method complements existing methods. Finally, we discuss the experimental results in section 10 and we conclude in section 11.

2 Segmentation of Potential Rain Streaks

The first stage of the method consists in segmenting rain streaks which are visible in the image. Different dedicated methods have been proposed to find potential rain streaks. Like in other methods, three different cues are used: motion, photometry and size of the rain streaks. Unlike other methods, motion segmentation, which is usually used in visual surveillance systems, is adapted to the problem of segmentation of rain streaks.

2.1 Visibility of Rain

According to Garg and Nayar [4], the visibility of rain in videos depends on the rain properties, on the scene properties and on the camera parameters:

$$v_r \propto \underbrace{\frac{a^2\sqrt{\rho}}{\sqrt{\nu}}}_{\text{Rain properties}} \overbrace{(L_r - L_b)}^{\text{Scene properties}} \underbrace{\frac{\sqrt{\mathcal{G}(f,N,z_0)}}{\sqrt{T_e}}}_{\text{Camera parameters}} \tag{1}$$

where $\mathcal{G}(f,N,z_0)$ is a function of focal length f, F-number N and the distance z_0 of the focus plane. T_e denotes the exposure time. (1) shows that the visibility v_r of rain increases as the square of the size a of the raindrop and as the square root of the rain density ρ. The visibility also decreases linearly with the difference between background brightness L_b and rain brightness L_r.

Given a scene in rain, camera parameters can be adjusted to see the rain droplets as proposed in [4]. Nevertheless in the visual surveillance context, the camera is in focus at the infinity, so that (1) can be slightly simplified:

$$v_r \propto \frac{a^2\sqrt{\rho}}{\sqrt{\nu}}(L_r - L_b)\frac{\sqrt{\mathcal{G}(f,N)}}{\sqrt{T_e}} \tag{2}$$

Consequently, the closest droplets are not in focus whereas the furthest ones are. The droplets which are in focus are motion blurred and create streaks in the images. Since the focalized droplets are the furthest ones, the streaks are generally small. To segment these streaks, we assume that a droplet is a moving elongated object which is small and brighter than its background. The segmentation process is deduced and is composed of three stages: motion segmentation, photometrical selection and size selection.

2.2 Segmentation of Moving Objects

Different methods have been proposed in order to detect moving objects. In the context of visual surveillance, approaches relying on background modeling [13] are commonly used since they allow taking into account gradual illumination changes in the scene, by constantly updating the background model. Basically, such methods allow computing a background model (BG) of the scene, which contains the static objects, as well as a foreground model (FG) of the scene, which contains the moving objects. Based on the review of literature proposed in [13], we chose the popular approach of the Mixture of Gaussians (MoG) [12]. In this approach, each pixel in the scene is modeled by a mixture of K Gaussian distributions. The probability that a certain pixel has a value X_t at time t can be written as:

$$P(X_t) = \sum_{k=1}^{K} \omega_k \mathcal{N}(X_t|\mu_{k,t}, \sigma_{k,t}) \tag{3}$$

where ω_k is the weight parameter of the k^{th} Gaussian component. $\mathcal{N}(X|\mu_k, \sigma_k)$ is the Gaussian distribution of the k^{th} component:

$$\mathcal{N}(X|\mu_k, \sigma_k) = \frac{1}{\sigma_k\sqrt{2\pi}} e^{-\frac{1}{2}\sigma_k^{-2}(X-\mu_k)^2} \tag{4}$$

where μ_k is the mean and σ_k is the standard deviation of the k^{th} component. The K distributions are ordered based on the fitness value $\frac{\omega_k}{\sigma_k}$ and the first B distributions are used as a model of the scene background where B is estimated as:

$$B = \underset{b}{\operatorname{argmin}} \left[\left(\sum_{j=1}^{b} \omega_j \right) > T \right] \tag{5}$$

T is the minimum prior probability of observing a background pixel. The Gaussian components that match the tested

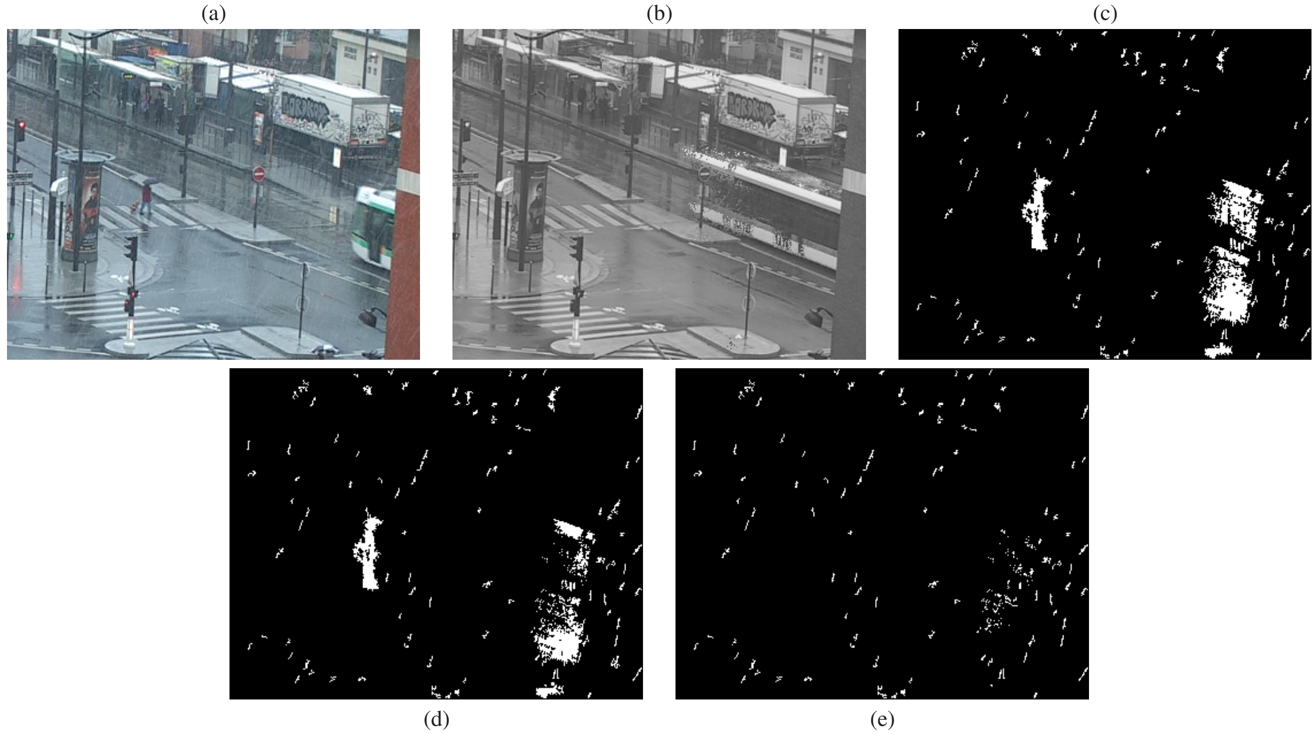


Fig. 2 Segmentation of potential rain streaks by background subtraction as well as selection rules on photometry and size: (a) original rainy image; (b) background model; (c) foreground model; (d) result after application of the photometric selection rule; (e) result after the application of the size selection rule.

value are updated by means of the following update scheme governed by the decay factor γ_1:

$$\begin{aligned} \omega_{k,t} &= (1-\gamma_1)\omega_{k,t-1} + \gamma_1 M_{k,t} \\ \mu_{k,t} &= (1-\rho)\mu_{k,t-1} + \rho X_t \\ \sigma^2_{k,t} &= (1-\rho)\sigma^2_{k,t-1} + \rho\left(X_t - \mu_{k,t}\right)^2 \\ \rho &= \gamma_1 \mathcal{N}(X_t|\mu_{k,t},\sigma_{k,t}) \\ M_{k,t} &= \begin{cases} 1; \text{ if } \omega_k \text{ is the first matched component} \\ 0; \text{ otherwise} \end{cases} \end{aligned} \tag{6}$$

$\frac{1}{\gamma_1}$ defines the time constant which determines change. Only two parameters, γ_1 and T need to be set. Then, background subtraction is performed by marking as foreground any pixel, whose intensity is more than D times the standard deviation away from any of the B distributions. Such a method has satisfactory performance for rain segmentation despite it does not use the relationships and the correlations amongst nearby pixels. Conversely, the more robust background modeling approaches, like [14–16], try to be robust to periodic motions, such as the moving trees. Consequently, if rain has a periodic motion, these methods are not likely to be relevant to detect the potential rain streaks. Finally, the MoG approach can be seen as a generalization of the median filter and the k-means algorithm, respectively used by Hase et al. [6] and Zhang et al. [7].

2.3 Applying a Photometrical Selection Rule

Garg and Nayar [4] propose a local method to detect rain streaks in images, which relies on two constraints. First, the change in intensity due to a rain drop in the n^{th} frame must satisfy the constraint:

$$\Delta I = I_n - I_{n-1} = I_n - I_{n+1} \geq c \tag{7}$$

where c is a threshold that represents the minimum transient change in intensity caused by a rain drop. The background subtraction approach proposed in the previous paragraph is similar to this process. Based on (7), a photometrical selection rule for the foreground pixels is built:

$$\Delta I = I_{FG} - I_{BG} \geq c \tag{8}$$

It is also assumed in [4] that the streak intensity is linearly related to the background intensity occluded by the rain. We do not use this strong assumption.

2.4 Applying a Size Selection Rule

The last step of the segmentation of rain streaks consists in filtering out objects in the FG which are too large and too small. This is done by finding the different connected components by a flood-fill algorithm, and then by suppressing the connected components whose size is not plausible. A similar approach is proposed by Brewer and Liu [8], who propose a method to build such a size template selection.

2.5 Results

Fig. 2 illustrates the segmentation of potential rain streaks. Fig. 2(a) and Fig. 2(b) show an image with rainy weather and its corresponding BG. Fig. 2(c) shows the FG, where rain streaks are visible. Then, the result of the photometric selection rule (8) is shown in Fig. 2(d). Finally, the size selection rule is applied (see Fig. 2(e)). Thereafter, only the rain streaks and some pixels of noise remain. The aim is now to separate the rain streaks from the pixels of noise.

3 Construction of the HOS

3.1 Methodology

Resulting from the segmentation process presented in the previous section, we have an image of small blobs, which are either rain streaks or pixels of noise. Rain streaks are assumed to be majority and to be almost vertically oriented. We rely on a soft-voting strategy to estimate the orientation of the rain, and to reject the pixels or blobs which do not correspond to this orientation. It is based on the computation of an Histogram of Orientation of Streaks (HOS) which takes into account the confidence in orientation measurement. In order to estimate the orientation of each streak, different methods may be used. Among them, the method of geometric moments applied to each segmented streak gives the best results.

3.2 Soft-Voting Algorithm

A soft-voting process is often used to obtain reliable data from multiple uncertain data sources. In the area of computer vision, such a process is often used to deduce from local information a global information, e.g. the Hough transform [17]. In a similar way, we propose to estimate the orientation of the rain by the accumulation of the local orientation of the rain streaks following a Parzen approach [18]. In this aim, a weighted cumulative histogram $h(\theta)$, made of 180 bins, is computed for $\theta \in [0, \pi]$ which takes into account a weight w_i and an uncertainty $d\theta_i$ on the estimation of the orientation θ_i of each blob $B_{i, 1 \leq i \leq P}$. For each blob, we cumulate a Gaussian distribution $\mathcal{N}(\theta | \theta_i, d\theta_i)$ with center θ_i and standard deviation $d\theta_i$ in the histogram. This histogram of orientation is thus expressed as:

$$h(\theta) = \sum_{i=1}^{P} w_i \mathcal{N}_i(\theta | \theta_i, d\theta_i) \tag{9}$$

where P denotes the total number of blobs.

3.3 Computation of the Orientation of the Streaks

Different methods may be used to estimate locally the orientation of a blob. We implemented and compared two different approaches. The first method is a pixel-based approach which relies on the computation of local gradient. It is similar to existing HOG approaches [19]. The second method is a patch-based approach which relies on the method of geometric moments, applied to each segmented blob.

3.3.1 Gradient Orientation (HOG)

Let us denote G_x and G_y the gradient of a pixel along x and y axis obtained with the Canny-Deriche filter [20]. The orientation of the gradient θ is obtained by:

$$\theta = \tan^{-1} \frac{G_y}{G_x} \tag{10}$$

The uncertainty on the estimation of the orientation $d\theta$ is obtained by computing the partial derivatives of (10) with respect to G_x and G_y:

$$d\theta = \frac{dG}{\sqrt{G_y^2 + G_x^2}} \tag{11}$$

assuming $dG_x = dG_y = dG$. The weight w of each sample in (9) is chosen as the magnitude of the gradient. Based on (9), the HOG is expressed as:

$$h(\theta) = \sum_{i=1}^{P} \sqrt{G_{x_i}^2 + G_{y_i}^2}\, \mathcal{N}\left(\theta \,\middle|\, \tan^{-1} \frac{G_{y_i}}{G_{x_i}}, \frac{dG}{\sqrt{G_{x_i}^2 + G_{y_i}^2}}\right) \tag{12}$$

The value of dG is chosen empirically by performing tests in simulation (see subsection 3.4).

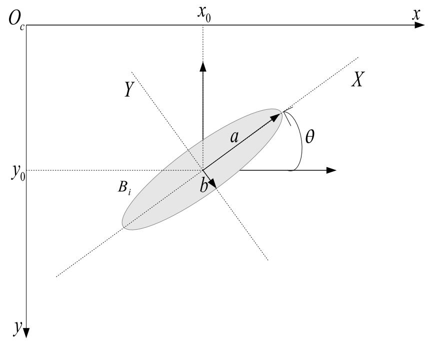


Fig. 3 An ellipse with its main parameters: Major semiaxis a, short semiaxis b, gravity center (x_0, y_0) and tilt angle θ.

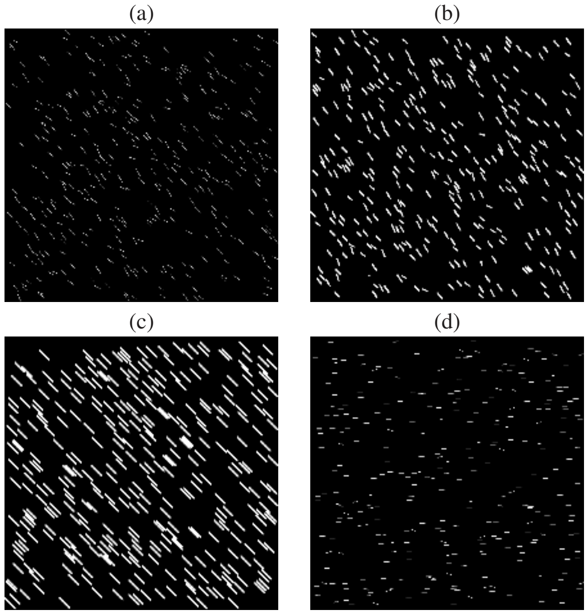

(a) (b) (c) (d)

Fig. 4 Synthesized images presenting streaks with different widths, lengths and orientations following a Gaussian distribution $\mathcal{N}(\mu,\sigma)$: (a) $l_s = 10$ pixel, $w_s = 1$ pixel, $\mu = 45^\circ$, $\sigma = 10^\circ$; (b) $l_s = 10$ pixel, $w_s = 2$ pixel, $\mu = 55^\circ$, $\sigma = 10^\circ$; (c) $l_s = 20$ pixel, $w_s = 2$ pixel, $\mu = 45^\circ$, $\sigma = 0^\circ$; (d) $l_s = 10$ pixel, $w_s = 1$ pixel, $\mu = 0^\circ$, $\sigma = 10^\circ$.

3.3.2 Geometric Moments (HOS)

Each segmented blob $B_{i,1\leq i\leq P}$ is assimilated with an elliptical disk. Fig. 3 shows an ellipse characterized by its 5 parameters: major semiaxis a, short semiaxis b, gravity center (x_0, y_0) and tilt angle θ. The method of geometric moments is used to compute the geometric parameters of the associated ellipse [21]. The central second-order moments are given by:

$$\begin{aligned} m_i^{20} &= \frac{1}{m_i^{00}} \sum_{(x,y)\in B_i} (x-x_0)^2 \\ m_i^{11} &= \frac{1}{m_i^{00}} \sum_{(x,y)\in B_i} (x-x_0)(y-y_0) \\ m_i^{02} &= \frac{1}{m_i^{00}} \sum_{(x,y)\in B_i} (y-y_0)^2 \end{aligned} \tag{13}$$

Then, the major and short semiaxis of the equivalent ellipse are given by:

$$a_i = 2\sqrt{\lambda_i^1} \tag{14}$$

$$b_i = 2\sqrt{\lambda_i^2} \tag{15}$$

where λ_i^1 and λ_i^2 are the eigenvalues of the matrix:

$$\begin{pmatrix} m_i^{20} & m_i^{11} \\ m_i^{11} & m_i^{02} \end{pmatrix} \tag{16}$$

The orientation θ_i of the blob B_i is given by:

$$\theta_i = \frac{1}{2}\tan^{-1}\left(\frac{2m_i^{11}}{m_i^{02} - m_i^{20}}\right) \tag{17}$$

The weight w_i is chosen as the value of the major axis a_i of the equivalent ellipse so as to give a higher weight to longer streaks. The uncertainty on the estimation of the orientation $d\theta_i$ is computed from:

$$(d\theta)^2 = \left(\tfrac{\partial\theta}{\partial m^{11}} dm^{11}\right)^2 + \left(\tfrac{\partial\theta}{\partial m^{02}} dm^{02}\right)^2 + \left(\tfrac{\partial\theta}{\partial m^{20}} dm^{20}\right)^2 \tag{18}$$

Assuming $dm^{11} = dm^{02} = dm^{20} = dm$, we deduce:

$$d\theta_i = \frac{\sqrt{(m_i^{02} - m_i^{20})^2 + 2(m_i^{11})^2}}{(m_i^{02} - m_i^{20})^2 + 4(m_i^{11})^2} dm \tag{19}$$

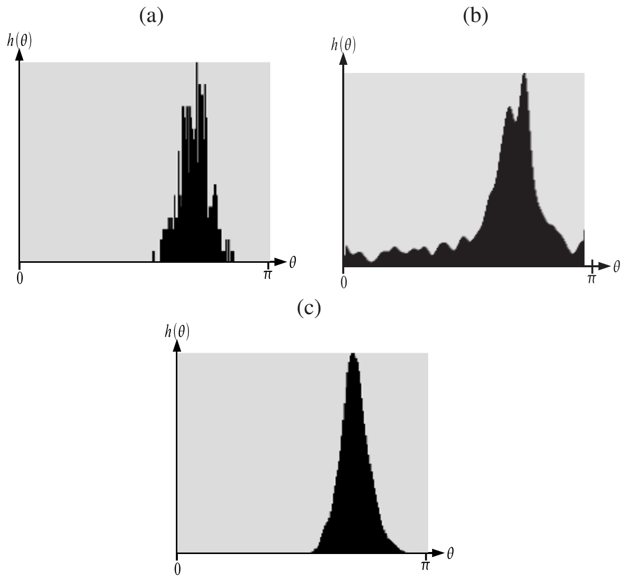
The value of dm is chosen empirically by performing tests in simulation (see subsection 3.4). In this way, the contribution of elongated ellipses to the HOS is a peaky Gaussian distribution. The contribution of ellipses with shapes close to disks is a flat Gaussian distribution. Finally, the HOS in (9) is expressed as:

$$h(\theta) = \sum_{i=1}^{P} \frac{a_i}{d\theta_i\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\theta - \theta_i}{d\theta_i}\right)^2} \tag{20}$$

where $d\theta_i$ is given by (19), and a_i by (14).

3.4 Evaluation of the HOG and HOS

In this paragraph, the ability of the HOG and of the HOS to estimate the orientation of streaks is assessed. In this aim, we use synthetic images with streaks having orientations following a Gaussian distribution. We vary the width w_s,



(a) (b) (c)

Fig. 5 Different histograms based on image 4 (b): (a) Histogram of samples orientation; (b) HOG; (c) HOS.

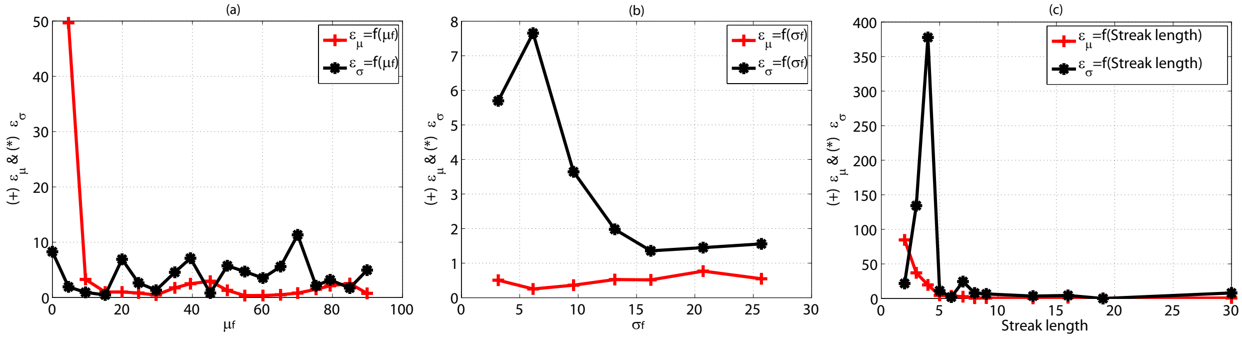


Fig. 6 Quantitative evaluation of the HOS representation on simulated data: Relative errors committed on the mean ε_μ and standard deviation ε_σ values of the histogram obtained with the method of geometric moments versus : (a) The specified mean μ_f; (b) The specified standard deviation σ_f; (c) The streak length.

the length l_s, the mean values μ and the standard deviation σ of the streaks. Fig. 4 shows samples of synthetic images. Fig. 5(a) shows the histogram of samples orientation from Fig. 4(b). Fig. 5(b) shows the HOG. Fig. 5(c) shows the HOS. From a qualitative point of view, one can see the greater ability of the HOS algorithm to approach the original distribution, allowing locating its main orientation by looking at the peak of the histogram. The noisy parts of the HOG are due to the fact that the gradient computed at the extremity of the rain streaks is not oriented like the longitudinal part of the rain streaks.

To assess the HOS quantitatively, we measure the mean value μ_m and the standard deviation σ_m of the peak of the HOS on different simulated images, where each parameter of the streak-noise distribution is varied independently. For each simulation, we compute the relative errors committed on the mean ε_μ and on the standard deviation ε_σ measured on the HOS:

$$\varepsilon_\mu = \frac{\mu_f - \mu_m}{\mu_f} \tag{21}$$

$$\varepsilon_\sigma = \frac{\sigma_f - \sigma_m}{\sigma_f} \tag{22}$$

In each simulation, 350 streaks are generated. The default parameters are: $w_s = 1$ pixel, $l_s = 11$ pixel, $\mu_f = 55^\circ$and $\sigma_f = 10^\circ$.

Fig. 6(a) shows the committed relative error versus the mean μ_f. Relative errors are smaller than 10% on the standard deviation ε_σ when μ_f is greater than 10 pixels. In the same way, relative errors are smaller than 5% on the mean ε_μ when μ_f is greater than 10 pixels. Fig. 6(b) shows the committed relative errors versus the standard deviation σ_f. The relative error is smaller than 10% and 1% respectively for the standard deviation ε_σ and the mean ε_μ. Relative errors on the standard deviation ε_σ are smaller than 2% when $\sigma_f > 13$. Fig. 6(c) shows the committed relative errors versus the length of the streak l_s. The method of geometric moments is efficient only if the streak length l_s is greater than 5 pixels with errors smaller than 5% on the mean ε_μ and on the standard deviation ε_σ. These results prove that the HOS is relevant for estimating the main orientation of the streaks distribution if $\mu > 10$ pixels and $l_s \geq 5$ pixels. We also compared quantitatively the HOS with the HOG. The HOS which relies on blobs is more robust to noise than to the HOG which relies on pixels. However, results of the HOG are not included in Fig. 6 for the sake of readability.

Finally, we simulated more realistic images of rain streaks using the database of rain streaks provided by Garg and Nayar [22], in which we added a stochastic orientation of the streaks. Two sample images are shown in Fig. 7. The corresponding HOS are shown, whose shape resembles the HOS in Fig. 5. The satisfactory results obtained with these realistic physical simulations allows establishing the relevance of the HOS representation in order to capture the orientation of actual rain and not only of artificial streaks.

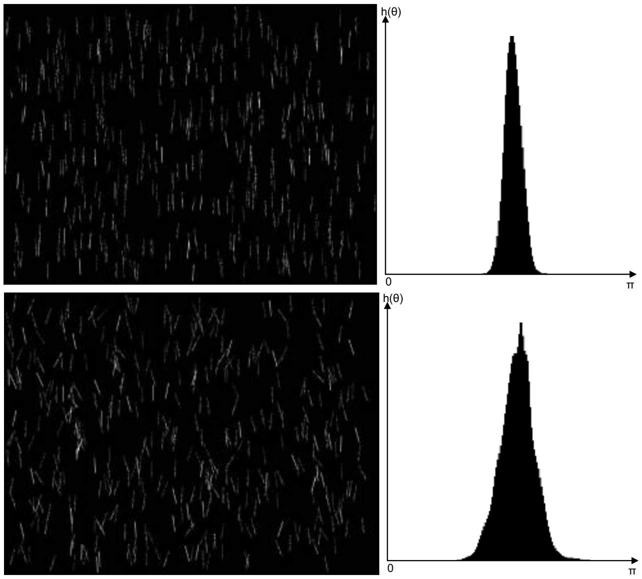


Fig. 7 Realistic rainy images simulated using the rain streak database provided by Garg and Nayar with different orientations [22] as well as the corresponding HOS.

4 Modeling of the HOS

In the previous section, the construction of the HOS has been presented and its ability to represent the majority orientation of a set of oriented blobs has been evaluated. In this section, we propose to model the data of the HOS and to design an algorithm to automatically estimate its parameters in the presence of noise.

4.1 Model

The FG is composed of rain streaks combined with pixels of noise. We assume that the rain streaks have an orientation which follows a Gaussian distribution. The remaining question is the modeling of the noise. It mainly comes from other elements of the scene which remain in the FG after segmentation, as well as from image compression artifacts. Generally, this type of noise does not have any preferred orientation. However, in structured environments such as streets, the main orientation of the noise may be directed along the vanishing lines of the scene. In practice, its standard deviation is so big, that it can be generally considered as a uniform distribution. From these considerations, we model the data in the HOS as a Gaussian-uniform mixture distribution:

$$y(\theta) \sim \Pi \mathscr{N}(\theta|\mu,\sigma) + (1-\Pi)\mathscr{U}_{[0,\pi]}(\theta) \tag{23}$$

where θ is an angle between $[0,\pi]$ and $y(\theta)d\theta$ is the probability of observing $[\theta, \theta + d\theta]$. Π denotes the surface of the Gaussian distribution $\mathscr{N}(\theta|\mu,\sigma)$ with mean μ and standard deviation σ. $\mathscr{U}_{[0,\pi]}(\theta)$ denotes the uniform distribution on the interval $[0,\pi]$.

4.2 Estimation of Model Parameters

The parameters are estimated using an expectation-maximization (EM) algorithm, see for instance [23]. The usual algorithm assumes a random sampling. Here, the angle θ is quantified as $(\theta_i)_{i=1,\ldots,N}$ and for each θ_i, we have y_i samples. Therefore, we adapted the algorithm to take into account the fact that x is quantified. The algorithm is still iterative in two steps. The k^{th} expectation step is given by:

$$\hat{z}_i^k = \frac{(1-\hat{\Pi}^{k-1})\mathscr{U}_{[0,\pi]}(\theta_i)}{\hat{\Pi}^{k-1}\mathscr{N}(\theta_i|\hat{\mu}^{k-1},\hat{\sigma}^{k-1}) + (1-\hat{\Pi}^{k-1})\mathscr{U}_{[0,\pi]}(\theta_i)} \tag{24}$$

where $i = 1,\ldots,N$. The k^{th} maximization step is given by:

$$\begin{aligned} \hat{\mu}^k &= \frac{\sum_{i=1}^N (1-\hat{z}_i^k)\theta_i y_i}{\sum_{i=1}^N (1-\hat{z}_i^k) y_i} \\ (\hat{\sigma}^k)^2 &= \frac{\sum_{i=1}^N (1-\hat{z}_i^k)(\theta_i - \hat{\mu}^k)^2 y_i}{\sum_{i=1}^N (1-\hat{z}_i^k) y_i} \\ \hat{\Pi}^k &= \frac{\sum_{i=1}^N (1-\hat{z}_i^k) y_i}{\sum_{i=1}^N y_i} \end{aligned} \tag{25}$$

To initialize the algorithm, $\hat{\mu}^0$, $\hat{\sigma}^0$ and $\hat{\Pi}^0$ are approximated using the following rules. We compute the median value λ of the y_i in the histogram. This value λ is subtracted from each y_i, which allows keeping only the upper part of the Gaussian distribution by removing the negative values. Then, the mean value $\hat{\mu}^0$, the standard deviation $\hat{\sigma}^0$ and the surface $\hat{\Pi}^0$ are computed from the upper part of the histogram. For complete genericity, the distance between angles must take into account the angular periodicity on the previous computations.

Finally, when the EM algorithm has converged, we apply a Goodness-of-Fit (GoF) test. This GoF test is used to know if the fitted distribution is close enough to the observed one using a decision threshold. The GoF test compares the cumulated distribution function of the observed histogram $F_o(x)$ with the cumulated distribution $F_e(x)$ obtained by the EM algorithm. The maximum distance D between the fitted and the observed functions is computed as:

$$D = \sup_{x\in[0,\pi]} |F_e(x) - F_o(x)| \tag{26}$$

If D exceeds a threshold D_c, the fitting is considered inaccurate and the frame is rejected from further processing.

4.3 Accuracy of the EM Algorithm

To test the accuracy of the EM algorithm, we have simulated images with two different distributions of streaks and we have evaluated the errors on the estimated parameters. Let us show an example of estimation result. The first distribution is a Gaussian distribution of rain streaks with $\mu = 65^\circ$and $\sigma = 10^\circ$. The second distribution is a uniform distribution with a surface of 40% of the total surface of the distribution and represents the noise. A sample synthetic image and the corresponding histogram of orientation samples are shown respectively in Fig. 8(a) and Fig. 8(b). The parameters estimated by the EM algorithm are $\mu = 64.3^\circ$and $\sigma = 11.2^\circ$and the surface of the uniform distribution is estimated to 37%. Fig. 8(c) shows the HOS as well as the Gaussian uniform mixture model superimposed. Based on the different tests carried out, the accuracy of the proposed EM algorithm is satisfactory.

5 Decision on Rain Presence

In the previous section, an EM algorithm has been designed to dynamically estimate the parameters of the HOS. In this section, a process is designed so as to decide if rain is present in the tested image sequence.

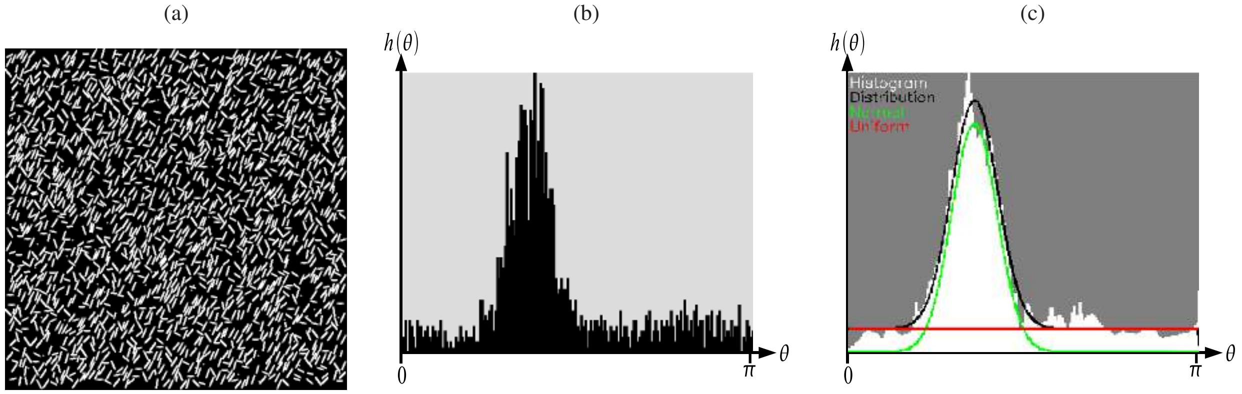


Fig. 8 Verification of the EM algorithm for Gaussian-uniform modeling: (a) image presenting different streaks with orientations following a Gaussian-uniform mixture distribution; (b) histogram of orientation samples; (c) HOS and fitted mixture model obtained by the EM algorithm. The black curve represents the fitted distribution, which is decomposed in the Gaussian distribution in green and the uniform distribution in red.

5.1 Principle

The decision on the presence or absence of rain is taken on the past images where the model is valid, i.e. the images where the GoF test is positive. If rain is present in the scene, the surface of the Gaussian distribution estimated by the EM algorithm should be quite high most of the time. If rain is not present in the scene, the surface of the Gaussian distribution should remain quite low most of the time.

A Kalman filter [24] is used to compute a temporally smoothed model of the HOS (three parameters Π, μ, σ) assuming a static evolution model. If the GoF test is negative, the filter is not updated. Then, a threshold on the surface of the Gaussian distribution of the filtered distribution is used to take the decision on the presence or absence of rain. If the surface is above a threshold denoted Π_{rain}, rain is detected. If it is not the case, rain is not detected in the scene. In the next paragraph, the effect of each step of the decision process is shown through a complete experimental validation.

5.2 Experimental Validation

5.2.1 Methodology

To assess the ability of the algorithm to detect rain presence correctly, two videos of the same place were used, one with rain and the other without rain. The scene shows a classical urban street with a lot of moving objects like cars, tramways, pedestrians, etc. Each video sequence is made of 6,000 video frames. A sample image of the scene in the presence of rain is shown in Fig. 12(a). First, a qualitative evaluation of the method is proposed. Second, a quantitative evaluation is performed. A confusion matrix [25] as well as a ROC curve (Receiver Operating Characteristics) [26] are used to assess the method. In our case, for a two-class problem (rain, no-rain), the confusion matrix is expressed with four terms, see Tab. 1. TP is the number of True Positives. FP is the number of False Positives. TN is the number of True Negatives. FN is the number of False Negatives.

		Detected	
		Rain	Noise
Truth	Rain	TP	FN
	No rain	FP	TN

Table 1 Confusion matrix for presence / absence of rain.

From this matrix, two terms are usually defined: the True Positives Rate $TPR = \frac{TP}{TP+FN}$ and the False Positives Rate $FPR = \frac{FP}{FP+TN}$. ROC curves are obtained by plotting the TPR with respect to the FPR for different values of the parameters of the considered detector. If the numbers of positive and negative samples are within the same order of magnitude, the accuracy AC of the detector is defined as:

$$AC = \frac{TP+TN}{TP+FP+FP+TN}. \tag{27}$$

In our experiments, we try to maximize AC with respect to the tested parameters values.

5.2.2 Qualitative Results

The parameters values of the decision process have been chosen using a trial and error process. Indeed, it is easy to detect rain presence in image sequences with rain. It is more difficult to not detect rain presence in image sequences in clear weather. The GoF test is useful to detect images which follow or not the Gaussian-uniform distribution. However in clear weather, blobs with Gaussian distributed orientation may be detected in a single image. Such detections are punctual events and they generally have different orientations in the following images. In rainy weather, the orientation of the rain streaks is more consistent. Consequently,

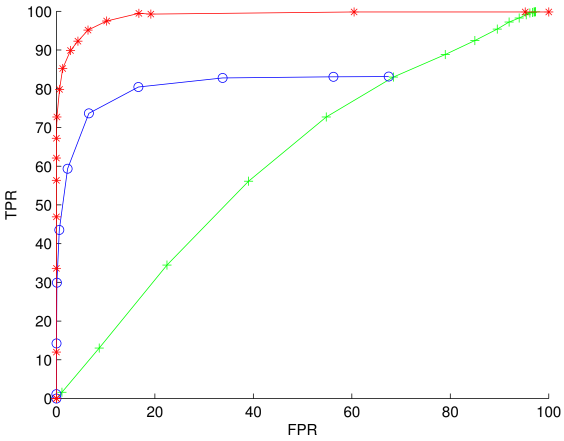


Fig. 9 ROC curve $TPR = f(FPR)$ for different processes of decision on rain presence. The green curve (+) shows the effect of the sole GoF test. The blue curve (o) shows the effect of the GoF test coupled with the threshold Π_{rain}. The red curve (*) shows the effect of the complete decision process for different values of the threshold Π_{rain}.

the proposed temporal filter allows distinguishing the two events. Finally, a uniform distribution is likely to be assimilated with a flat Gaussian distribution. To prevent this kind of error, the threshold Π_{rain} on the surface of the Gaussian distribution is very useful. The combination of these three thresholds gives the best results.

5.2.3 Quantitative Results

To see the effect of each parameter of the decision process, we have plotted the ROC curve for different alternative algorithms from the simplest one to the most complex one. First, we only use the GoF test, so as to set the value of the critical threshold D_c. The green curve (+) in Fig. 9 gives the ROC curve of the rain detector for different values of D_c. The best result (see previous paragraph) is obtained for $D_c = 0.06$, where $TPR = 73\%$ and $FPR = 55\%$ and $AC = 59\%$. The confusion matrix is given in Tab. 2.

		Detected	
		Rain	Noise
Truth	Rain	4361	1638
	No rain	3289	2709

Table 2 Detection results using only the GoF test.

Second, we use the GoF test coupled with a threshold on the surface of the Gaussian distribution. The blue curve (o) in Fig. 9 gives the ROC curve of the rain detector for different values of the threshold Π_{rain}. The best result is obtained for $D_c = 0.06$ and $\Pi_{rain} = 0.4$, where $TPR = 74\%$ and $FPR = 7\%$ and $AC = 84\%$. The confusion matrix is given in Tab. 3.

		Detected	
		Rain	Noise
Truth	Rain	4419	1581
	No rain	396	5603

Table 3 Detection results based on the use of the GoF test coupled with the use the threshold Π_{rain}.

Third, we use the complete decision process described in paragraph 5.1, where we add a Kalman filter between the GoF test and the test using Π_{rain}. The red curve (*) in Fig. 9 gives the ROC curve of the rain detector for different values of the threshold Π_{rain}. The best result is obtained for $D_c = 0.06$ and $\Pi_{rain} = 0.35$, where $TPR = 97\%$ and $FPR = 7\%$ and $AC = 95\%$. The confusion matrix is given in Tab. 4. In these tests, the parameters of the Kalman filter are set as follows: The variance of the process noise is equal to 0.01 and the variance of the measurement noise is equal to 0.1. The setting of these parameters was done empirically, again by maximizing the accuracy of the rain detector. We found that the sensitivity to these parameters was not important. The final results are satisfactory.

		Detected	
		Rain	Noise
Truth	Rain	5790	210
	No rain	396	5603

Table 4 Detection results based on the complete decision process.

Despite an accuracy of 95%, some errors still remain and can be explained. In several frames, many moving objects (pedestrians, cars, tramway) are simultaneously present in the scene, which leads to pixels of noise in the result of the segmentation process. In the same time, the useful surface for rain detection is reduced due to large moving objects, which makes rain more difficult to detect and explains the number of false negatives. In other frames, when there is no rain, the noise sometimes happens to be structured with the same orientation during few frames, which explains the number of false positives.

Another video was recorded on the parking lot of our laboratory in presence of rain. One image of the scene is shown in Fig. 12(b). The scene contains trees, cars, pedestrians and rain. The duration of the sequence is about one minute with a frame rate equal to 25 images per second (1500 images). The TPR is 98.5% and the FPR is equal to 1.5%. Yet another video was recorded on the same parking lot, but this time in presence of snow. One image of the scene is shown in Fig. 12(c). The duration of the sequence is about six minutes with a frame rate equal to 15 images per second (5438 images). The TPR is 98.9%. The FPR is equal to 1.1%. The errors can be explained as previously.

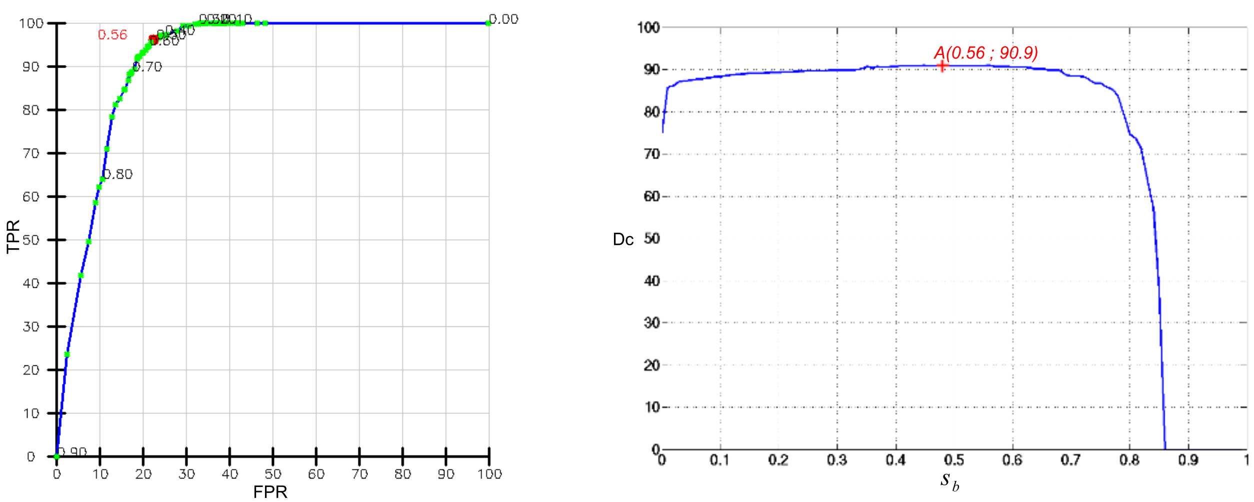

(a) (b)

Fig. 10 Quantitative evaluation of the rain pixels detection algorithm on synthetic Images: (a) ROC curve; (b) AC versus the threshold S_b.

In order to study the latency of our algorithm during a changeover, we used a sensor simulator, named Pro-SiVIC[1] [27], in order to create a scenario with alternate presence and absence of rain. From presence to absence of rain, the latency of the algorithm is about 25 images. From absence to presence of rain, the latency is about 50 images.

6 Rain Pixels Detection

In the previous section, a decision process on rain presence has been proposed and assessed. In this section, we propose to detect rain pixels through use of the HOS.

6.1 Approach

Once rain has been detected in the scene, the computation and modeling of the HOS offers a convenient and efficient way to segment the rain pixels in the images. From the estimated Gaussian-uniform mixture, we can compute the following probabilities:

$$P(\theta_i|rain) = \frac{\mathcal{N}(\theta_i|\mu,\sigma)}{\Pi\mathcal{N}(\theta_i|\mu,\sigma)+(1-\Pi)\mathcal{U}_{[0,\pi]}(\theta_i)} \tag{28}$$

$$P(\theta_i|noise) = \frac{\mathcal{U}_{[0,\pi]}}{\Pi\mathcal{N}(\theta_i|\mu,\sigma)+(1-\Pi)\mathcal{U}_{[0,\pi]}(\theta_i)} \tag{29}$$

Then, when the probability ratio $\frac{P(rain|\theta_i)}{P(noise|\theta_i)} > S_b$ the blob B_i is classified as rain streak. This ratio is written as a function of (28) and (29) using Bayes rules.

6.2 Experimental Validation

6.2.1 *Synthetic Images*

First, we created synthetic images with streaks having orientations following a Gaussian-uniform distribution with different percentage of pixels of noise ranging from 10 to 90%. In order to assess the influence of the parameter S_b (with $s_b \in [0,1]$) on the classification result, we computed a ground-truth classification in rain/non-rain pixels for different values of the threshold S_b. We then compared pixel classification with the ground truth. Since the numbers of rain pixels and of non-rain pixels are not within the same order of magnitude, Dice's coefficient is used to compute the accuracy:

$$D_c = \frac{2TP}{2TP+FP+FN}. \tag{30}$$

Fig. 10(b) shows Dice's coefficient D_c versus the threshold S_b and Fig. 10(a) the ROC curve. From these figures, we conclude that the optimal threshold equals $S_b = 0.56$ with $D_c = 90.9\%$ (point A in Fig. 10(b)), $TPR = 96.2\%$ and $FPR = 22.6\%$.

6.2.2 *Actual Images*

For experiments on actual images, we used the video introduced in section 5.2.1. We manually segmented the rain streaks on 33 images. This segmentation was performed by ten different persons, which allows building a ground-truth. To compute the accuracy of the rain pixels detection, the same method as before was then applied. Fig. 11(a) shows the obtained ROC curve. Fig. 11(b) shows Dice's curve with respect to the decision threshold S_b. The optimal threshold equals $S_b = 0.61$ with $D_c = 80.5\%$ (point A in Fig. 11(b)),

---

[1] http://www.civitec.net/

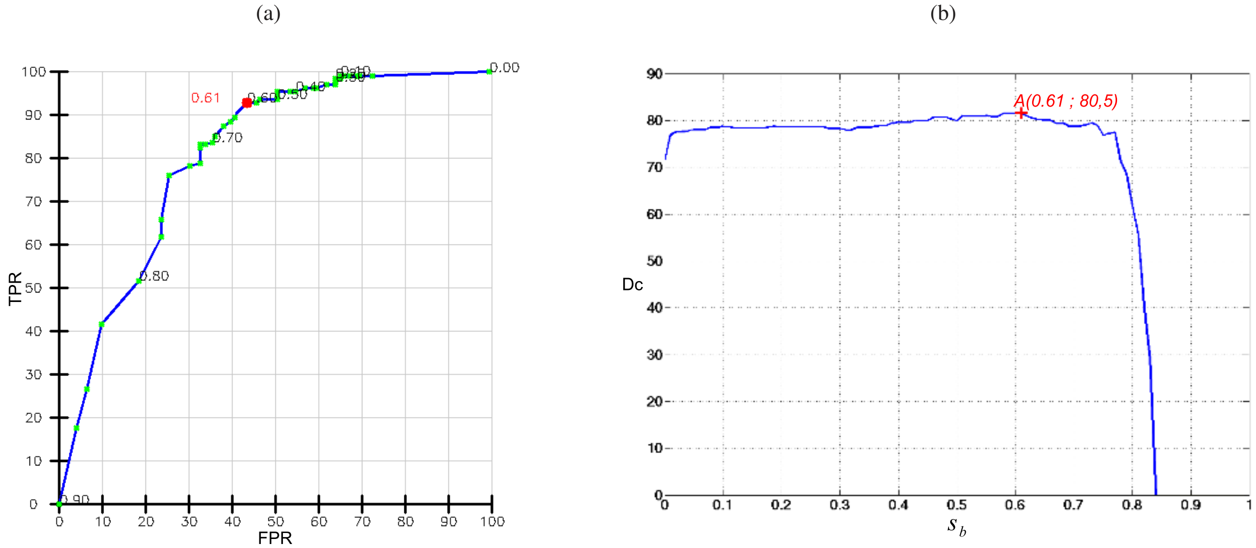


Fig. 11 Quantitative evaluation of the rain pixels detection algorithm on actual images: (a) ROC curve, $TPR = f(FPR)$. (b) Accuracy versus the threshold S_b.

$TPR = 93.0\%$ and $FPR = 41.1\%$. Both Dice's curves obtained with synthetic or actual images have approximately the same flat shape for S_b ranging from 0.1 to 0.7. The influence of S_b is thus limited within this interval. Thereafter, we consider a threshold of $S_b = 0.6$.

The entire process, from image acquisition up to the detection of rain pixels, is illustrated in different weather conditions in Fig. 12. Fig. 12(a) shows hail presence detection and the segmentation of hail pixels. Fig. 12(b) shows rain presence detection and the detection of rain pixels. Fig. 12(c) shows snow presence detection and the detection of snow pixels. In each case, the original image, the BG, the FG, the HOS and the detection of rain pixels are shown. The black curve represents the estimated Gaussian-uniform mixture. The green curve is the Gaussian distribution. The red curve is the uniform distribution. The cyan curve is the distribution smoothed by the Kalman filter decomposed into the Gaussian distribution in blue and the uniform distribution in yellow.

7 Estimation of Rain Intensity

In the previous sections, we showed how the HOS can be used to accurately estimate the rain orientation as well as how to classify the rain pixels. In this section, we show that the HOS provides a good proxy for the estimation of the rain density.

7.1 Related Works

The rain intensity is usually estimated using rain buckets [28] or optical devices [29]. Based on the visual effects of rain, Garg and Nayar [4] developed a camera-based rain gauge. In this aim, they observe the size and the numbers of drops in a small volume of rain over time. To prevent the overlapping of rain streaks, the camera is set to a low depth of field. Then in each image, rain streaks are segmented and a rain intensity is deduced by means of the estimation of the local rain density.

7.2 Contribution

In our case, the camera is in focus at the infinite and has automatic settings. Contrary to Garg and Nayar's approach [4], it is thus not possible to obtain an accurate estimate of the rain intensity. Nevertheless, we are able to derive a relative estimate of rain density by counting the number of rain pix-

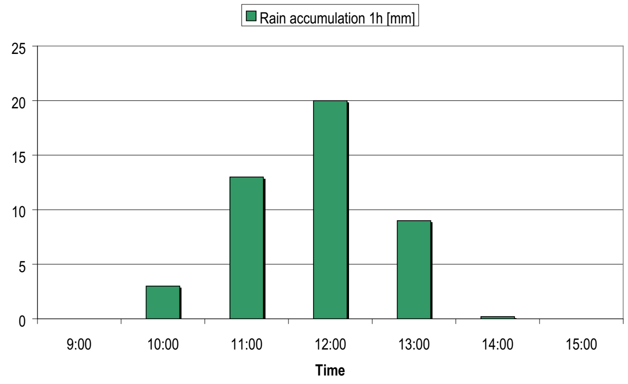


Fig. 13 Rain accumulation observed by the Montsouris weather station (Paris downtown) of Météo France (French national weather institute) during the storm of the 14^{th} July 2010. The rain accumulation observed every hour is plotted in green bars.

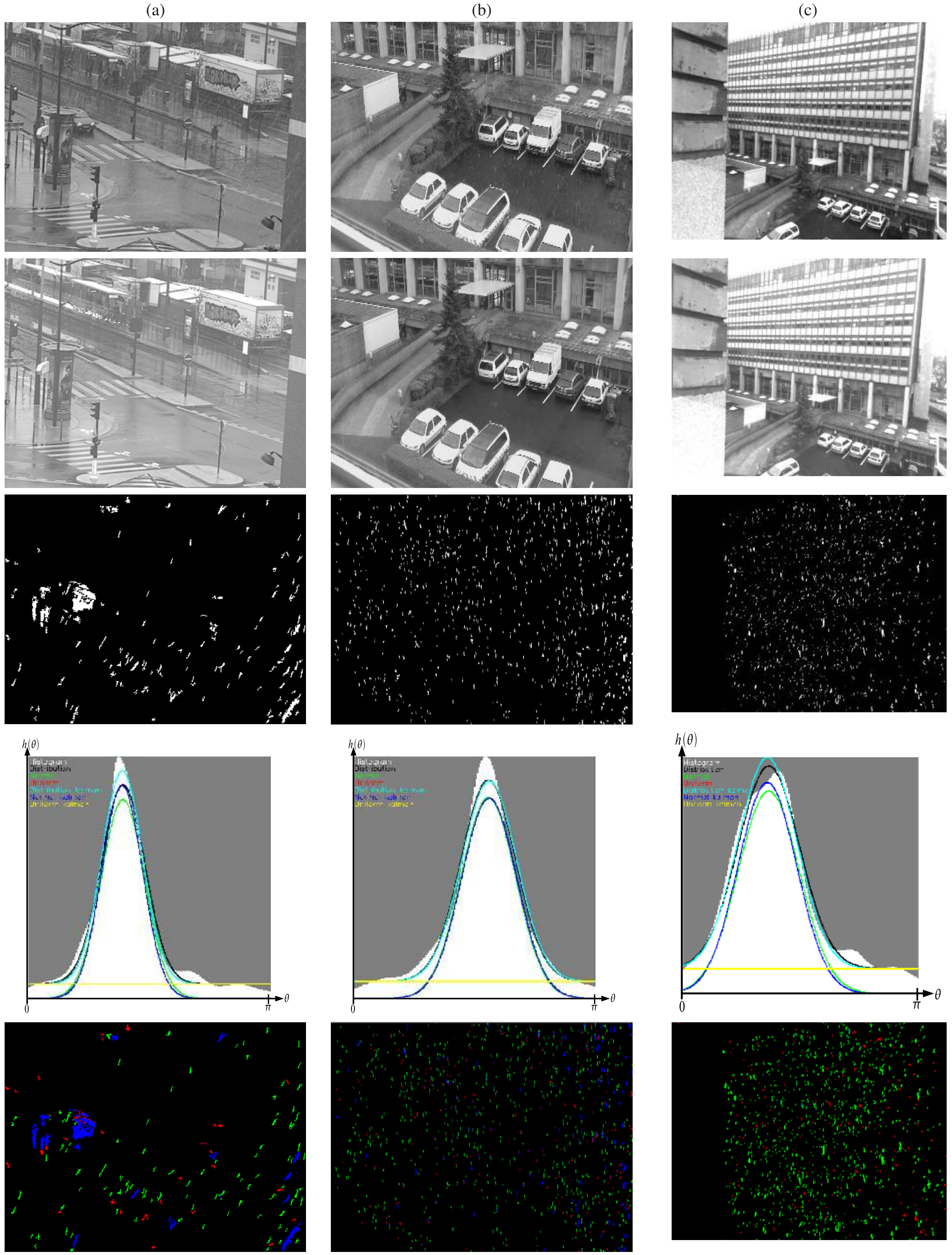


Fig. 12 First row: Original image. Second row: BG. Third row: FG. Fourth row: HOS with the different fitted curves. Fifth row: Rain pixels detection using $S_b = 0.6$ (the rain pixels are in green, the noise pixels in red and the objects filtered by size selection rule in blue). (a) is from the hail sequence; (b) is from the rain sequence; (c) is from of the snow sequence.

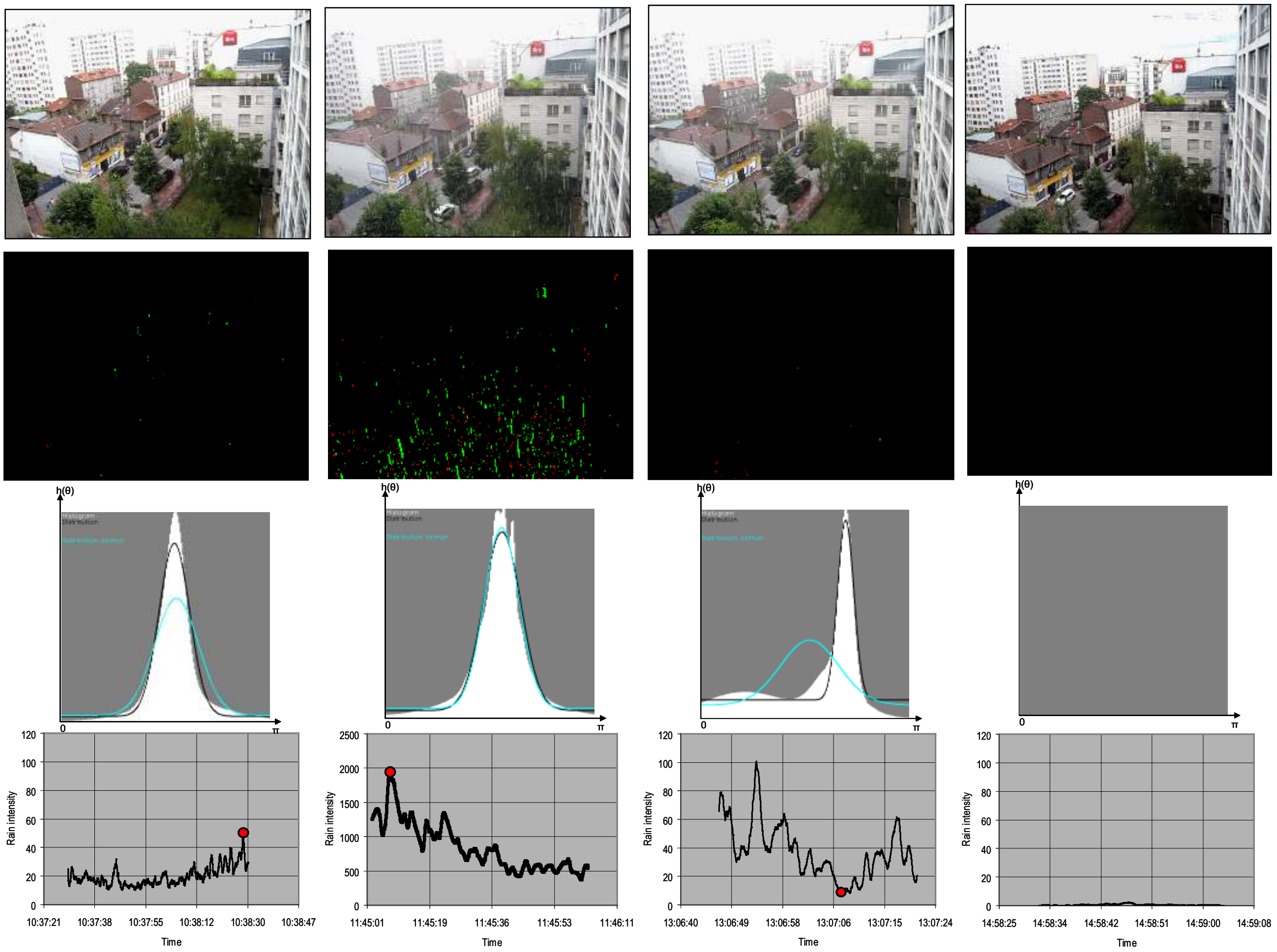


Fig. 14 Illustration of the ability of the method to estimate the rain intensity on videos grabbed in Paris on the 14^{th} July 2010. First row: Video frame corresponding to the red point of the graph below. Second row: Detected rain pixels. Third row: Corresponding HOS. Fourth row: Estimated rain density versus time.

els which have been detected by the approach detailed in section 6. This approach is more or less the method proposed by Garg and Nayar [4]. However, we think that it is not necessary to detect the rain pixels to have a rough idea of the rain intensity. Indeed, the HOS provides a good proxy of the rain intensity R as following:

$$R \approx \mathscr{S}\Pi \tag{31}$$

where $\mathscr{S}$ denotes the surface of the HOS before normalization and Π the surface of the Gaussian distribution in the HOS.

7.3 Experimental Evaluation

We have carried out experiments to check the quality of our estimator of rain intensity. In this aim, different videos have been grabbed during a severe summer storm which lasted a half day nearby Paris on the 14^{th} July 2010. In the same time, we have collected the data from the nearest weather station which is approximately 4 km away from the place where the videos were grabbed. These data are shown in Fig. 13. In Fig. 15, we plot our estimation of the rain intensity R versus the number of rain pixels, by using data collected during the peak of the storm. Both descriptors are strongly correlated.

We have also estimated R on different videos grabbed at different times of the day (10:37 - light rain, 11:45 - heavy rain, 13:06 medium rain, 14:58 - no rain). R is plotted versus time on the fourth row of Fig. 14. In the first row, we show the image corresponding to the red point of the graph. In the second row, we show the rain pixels which have been detected. In the third row, we show the corresponding HOS. R is coherent with the meteorological data which are shown in Fig. 13, as well as with the video frames, except in the third column. Indeed, the estimated rain intensity is very low. This is due to a strong wind which changes the rain orientation during the storm. Consequently, the Kalman filter does not converge correctly and the Gaussian-uniform

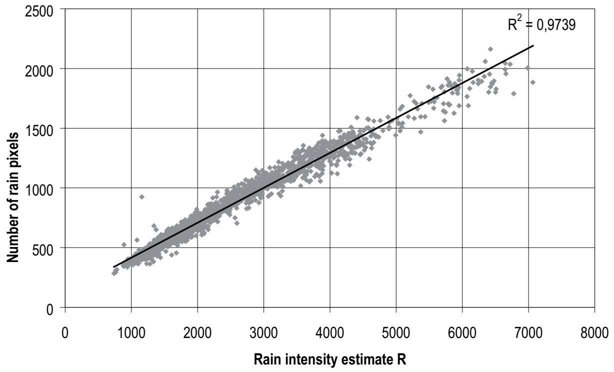


Fig. 15 Rain intensity estimation *R* versus the number of rain pixels. The rain intensity estimate is directly obtained by the HOS. The temporal integration of rain pixels was originally proposed by Garg and Nayar [4]. Both estimators are strongly correlated.

mixture is badly estimated. The rain intensity is thus also badly estimated. The Kalman filter can be adapted to cope with situations like rain storms. However, this will increase the false positives detections during rain episodes.

8 Potential Applications

In the previous sections, a method has been proposed and experimentally assessed which allows detecting rain episodes, detecting rain pixels in rainy images and estimating rain intensity as well as its orientation. These methods are mainly based on the use of the HOS. In this section, we present some potential applications which are related to road safety, weather observation, improvement of the reliability of video surveillance and rain rendering.

8.1 Detection of Safety Critical Weather Conditions

To monitor their networks, road operators use different sensors, among which optical sensors, especially the cameras which are the most convenient. Indeed, they are contact-less and they can serve multi-purpose applications: Incident detection, traffic monitoring, etc. The presence of bad weather conditions alter the operation of the transport system in different ways. It is obviously a source of accidents caused by reduced friction or reduced visibility [30]. It is also a source of reduction of highways capacity, which may increase traffic congestion. Hence, a traffic monitoring system must take into account adverse meteorological conditions, so as to help the road operator to take decisions as fast as possible. In other words, it must detect and quantify the meteorological conditions in real-time, in order to minimize their possible consequences, for instance by proposing adaptive speed limits [31, 32]. For instance, wireless communications were used in the European SAFESPOT project to combine data issued from vehicles sensors and roadside sensors [33] in order to prevent accidents. Different applications related to weather conditions were developed. To reach this goal, a camera-based sensing system as well as a data fusion methodology was designed to detect the weather conditions [5], in which the proposed algorithm is used.

8.2 Weather Observation

In the region of Paris (France), the storm of the 14^{th} July 2010 has caused numerous perturbations on the regional railway network and on the traffic of the international airport. These facts illustrate the needs for accurate weather predictions so as to prevent the paralysis of transports. The weather prediction and the observation are closely linked together. To refine their numerical schemes and calibrate them accurately, meteorological institutes need weather observation data at same spatial scales. As illustrated in section 7, we were able to monitor the storm locally. Such low cost data collection is thus of great interest for meteorological institutes.

8.3 Improvement of Video Surveillance Reliability

The operation range of video-surveillance systems is altered by the presence of degraded weather conditions. By detecting the weather conditions, it is possible to detect that the system is inoperative or to improve its operation. This methodology has been followed so far for the problem of degraded illumination [34] and the assessment of the operation range in foggy weather of optical sensors [35]. Like contrast restoration methods which improve the visual surveillance in foggy weather [2, 36–38], the proposed algorithm improves the reliability of video surveillance in rainy weather. Indeed, by detecting and filtering rain pixels in the FG, the detection of objects like cars or pedestrians, is simplified. This improves the security of the system.

8.4 Rain Rendering

Garg and Nayar [22] have shown that a realistic rendering of rain can be obtained based on the observation of natural image sequences in rain, by taking into account how light is refracted by rain. As pointed out by Barnum et al. [9], the combination of detection and of synthesis of rain offers great advantages since the illumination is already correct. By using the HOS, the distribution of the orientation of the rain can be captured and used to render the orientation of rain streaks more realistically. Samples of synthesized rain is shown in Fig. 7, based on the database of Garg and Nayar. This process is used in section 3 to evaluate the HOS.

9 Link with Rain Detection and Removal Methods

Various methods have been proposed to remove rain from images [4, 6–9]. Even if these methods and the method here proposed do not have the same objective, they share common ideas and principles. In this section, we compare the different methods qualitatively and quantitatively. In particular, we show how they complement each other.

9.1 Qualitative Comparison

The approach proposed by Hase et al. [6] uses a temporal median filter exploiting the fact that pixels are more often clear than altered by rain. This idea is extended by Zhang et al. [7] who proposed to detect rain using a k-means clustering algorithm ($k = 2$) and additional chromatic constraints to reduce false detections. However, this method cannot be used online because the k-means algorithm needs to be applied on the entire video sequence. Garg and Nayar [4] propose a method that uses the optical properties of a falling rain drop in order to segment the video scenes. The false detections are reduced via a photometric constraint which models the visual appearance of rain streaks. Brewer and Liu [8] detect rain streaks using their intensity spikes. They reduce false detections using their aspect ratio and their orientation. Barnum et al. [9] propose a global visual appearance model to identify the frequency-domain behavior of the rain. They present a filter in the frequency-domain to reduce its visibility. Finally, we propose to detect rain presence by using an histogram of the orientations of streaks (HOS). This HOS can then be used to detect rain pixels and to estimate the rain intensity.

Like Garg and Nayar [4] and Barnum et al. [9], we use a global constraint to detect the rain pixels. Garg and Nayar [4] detect the temporal correlation of rain using a binary field. Barnum et al. [9] directly use an appearance model of the rain in the frequency domain. Unlike Garg and Nayar [4] and Barnum et al. [9], we use of a soft-voting algorithm to accurately compute the distribution of the rain orientation by using the method of geometric moments applied to image blobs. This approach was also proposed independently by Brewer and Liu [8], but was not experimentally assessed. Apart from the method from Barnum et al. [9], the different methods use more or less the same photometric constraint. In this article, we show that this constraint can be adapted to the use of a classical background subtraction technique (MoG). Like Brewer and Liu [8], we compute the orientation of rain streaks locally. Whereas Brewer and Liu [8] directly reject the rain streaks which do not respect predefined orientations, we prefer to first compute the global orientation of the rain. Then, we reject the rain streaks which do not have the correct orientation. Finally, unlike other existing approaches, we apply our method on videos grabbed in clear weather and we propose a method to distinguish rainy weather from clear weather using the HOS.

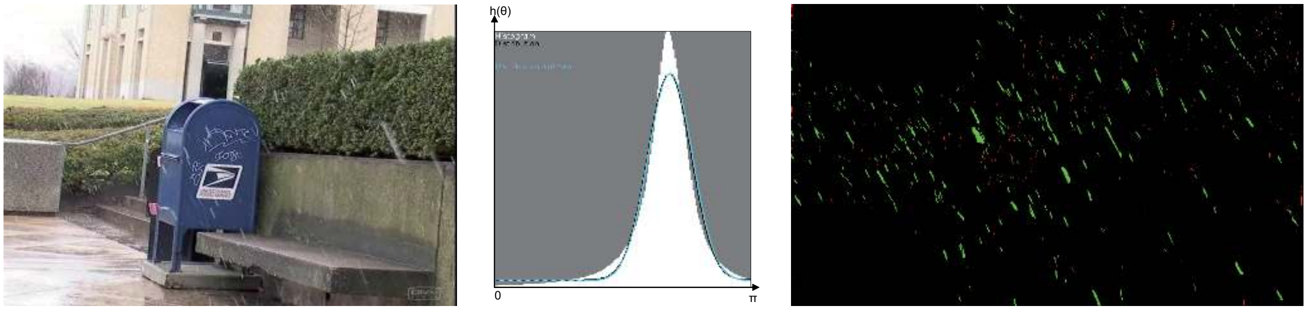
9.2 Quantitative Comparison

Our method does not aim primarily at segmenting the rain pixels in videos with rainy weather. Indeed, it aims at detecting rain presence in image sequences for visual surveillance purposes. To our best knowledge, no similar other method exists in the literature with which we could be compared. Nevertheless, the methodology we use to detect rain presence can also be applied to detect rain pixels, like previous methods [4,6–9]. In this paragraph, we propose to prove how complementary our method is with other rain segmentation methods.

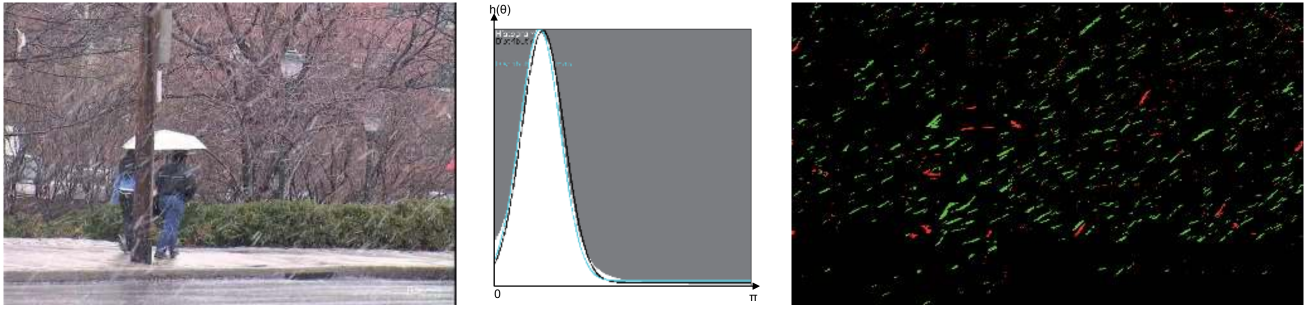
In this aim, we implemented existing methods or used existing results which were available online. Tests were carried out based on the video sequences shown in section 7. These videos are challenging ones since they contain different rain intensities. The camera moves slightly due to the storm. The background environment is complicated due to the moving trees. The method from [6] was easy to implement. Unfortunately due to the moving camera, the temporal median value had to be computed on only three images ($n = 3$). The results were not entirely satisfactory due to the moving and textured background. We then implemented the method of Garg and Nayar [4]. Due to the small size of rain particles, we were not able to use their binary field correctly. We were able to use their photometric constraint, see (7). We then removed the local computation of rain orientation from the method from Brewer and Liu [8]. Only the photometric constraint of (7) remained. However due to the complex background, this constraint was too strong and the resulting rain streaks were very fragmented. Finally, we used the following constraint:

$$I_n - I_{n-1} \geq c \text{ and } I_n - I_{n+1} \geq c \tag{32}$$

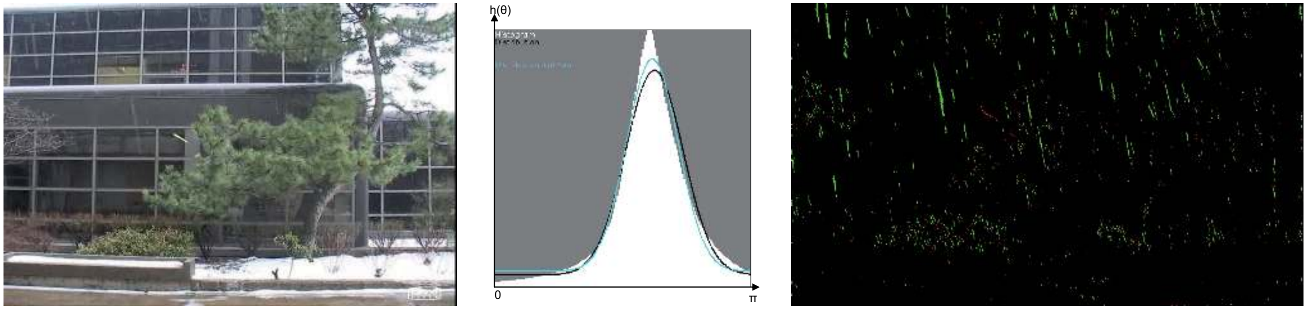
This constraint gave better results that the median filter ($n = 3$) and enabled us to use our method correctly. The results shown in Fig. 14 were obtained with this method. These results illustrate the ability of our method to use pixel based segmentation methods, e.g. [4, 6]. We also tried to adapt the method of Zhang et al. [7] to an online use but without success. Finally, we compared our method with the one from Barnum el al. [9]. Due to the relative complexity of the method, we used the videos of results which have been put online by the authors. As expected, we were able to compute the HOS using their segmentation of rain streaks. Furthermore, we were able to improve their segmentations. Some results are shown in Fig. 16. The rain pixels are put in green whereas noise pixels are put in red. In the "mailbox" sequence, the edges of the mailbox are detected as noise as



(a) The mailbox sequence: The edges of the mailbox are detected as noise, as well as part of the bushes.



(b) Walkers in the snow: The umbrella, the back and the feet of the walkers are detected as noise.



(c) Windowed building: The surveillance camera is detected as noise.

Fig. 16 Illustration of the complementarities between the spatio-frequency method of Barnum et al. [9] and our method. Left: Original sample image of rain videos. Middle: HOS computed using the segmentation results of Barnum et al. Right: Classification of the results of Barnum et al. in the rain pixels in green and in the noise pixels in red.

well as part of the bushes. In the "walkers" sequence, the umbrella, the backs and the feet of the walkers are detected as noise. In the "building" sequence, the surveillance camera is detected as noise. These results are not surprising since Barnum et al. fit their model to the videos assuming that the orientation of rain follows a uniform distribution ±0.2 rad. These results illustrate the advantage of our method based on a global segmentation.

In this section, we have shown that our semi-global method is generic enough to be used in conjunction with other rain segmentation methods, whether they use local approaches or global approaches. Furthermore, we were able to improve their results. Our rain detection method is thus complementary with existing methods.

10 Discussion and Perspectives

The algorithm is moderately demanding in terms of computing power. Our algorithm was developed with Microsoft Visual C++ .Net 2003 and RTMaps 3.2.2 using OpenCV 1.0. The computer used is composed of an Intel Core 2 Cpu 6600 (2.40 Ghz) processor and 3GB of ram and the operating system is Windows XP sp3. Using this configuration, the hail sequence (360x280 resolution) was processed at 15 Hz. For the rain sequence (resolution of 960x540), the frame rate is about 5 images per second. For the snow sequence (resolution 1360x1024), the frame rate is about 2 images per second. The frame rate not only depends on the resolution of the video, but on the number of streaks present in the image as well.

The proposed algorithm is able to detect rain presence with a good accuracy. Because we rely on a classical background subtraction method to segment potentiel rain pixels, our algorithm can be quite easily integrated in an existing video-surveillance platform. As such, it is thus sensitive to camera motion. In this case, image stabilization techniques could be useful. However, the proposed algorithm could also use other segmentation methods of rain streaks, such as those proposed in [4, 7–9], which are less sensitive to camera motion or already make use of image stabilization techniques.

The dataset used to test the method is made of different video sequences grabbed by ourselves. Consequently, this dataset may appear to be limited, but there is no other available on line. To capture a greater variety of scenes, we are equipping different test sites with video cameras as well as with rain sensors and visibilitymeters.

The type of falling particles is also an interesting topic. Indeed, making the difference between snowfalls and rain showers has important applications, e.g. for winter maintenance of roads. Following the idea proposed in [10], we have estimated the α of the snow and of the rain based on the following expression:

$$P = \alpha F + (1 - \alpha) B \tag{33}$$

where α is the transparency factor of the rain or of the snow, P is the intensity of the current pixel, F is the intensity of the FG pixel and B is the intensity of the BG pixel. Since snow is opaque and rain is transparent, we were expecting different values of α in rain and in snow. Unfortunately, we were not able to obtain satisfactory results, because the visual appearance of rain or snow depends primarily on the camera parameters and the environment (sun position, background), see (2).

Finally, the main problem of the proposed system is the ability of the camera to see rain. The rain with a small intensity is difficult to be seen even for human eyes, and thus to be detected with the proposed method. Our method, based on the MoG, is able to detect some snow episodes as well as strong rain. In the presence of a light rain, the MoG is no longer relevant. The photometric selection rule proposed in (32) is able to detect rain pixels in case of rain with a small intensity as well as strong rain. However in the absence of rain, this simple method may also detect rain presence. We thus believe that rain presence detection and the rain intensity estimation are two different applications which are difficult to run simultaneously. Nevertheless, guidelines to reduce or increase the visibility of rain in images by adjusting the camera parameters are proposed in [4]. They could be followed to improve the overall system. With the same idea, techniques which extend the depth of field [39] could help to increase the visibility of rain streaks close to the camera. In this way, we may be able to build a camera-based gauge in focus at the infinite.

11 Conclusion

In this article, a method is proposed which consists in detecting rain by using video cameras. To ease the integration of the algorithm in video-surveillance systems, we use a classical MoG model to separate the foreground from the background in image sequences. Because rain is a dynamic weather phenomenon, the foreground is used to detect it. From the foreground model, selection rules based on photometry and size are used in order to select the potential blobs of rain. Then we compute a Histogram of Orientations of Streaks (HOS) with the method of geometric moments. The HOS is assumed to follow a Gaussian-uniform mixture model, where the Gaussian distribution represents the orientation of the rain and the uniform distribution represents the orientations of the noise. We use an EM algorithm to separate these two distributions as well as a GoF test coupled with a Kalman filter and a test on the surface of the Gaussian distribution to deduce the information on rain presence. When rain presence is detected, we proceed with detecting the pixels of rain in the foreground model or with estimating the rain intensity. The applications of the method are numerous, among them the video-surveillance of road networks, the improvement of the reliability of video-surveillance systems are shown and rain rendering. Various methods have been proposed to remove rain in images. Even if these methods and the method proposed in this article do not have the same objective, they share common ideas and principles. We have compared the different methods qualitatively and quantitatively. In particular, we have shown how they complement each other. Numerous experiments have been carried out to rate the proposed method, which rely on the use of synthetic images as well as actual video sequences grabbed by ourselves in different weather conditions such as clear weather, hail, rain, snow and storm. Finally, we have proposed different perspectives for future work.

Acknowledgements

This work is partly supported by the SAFESPOT project initiated by the European Commission in the FP6 under the IST theme (IST-4-026963-IP). The authors would like to thank the reviewers for very constructive and detailed comments, as well as Eric Dumont for having filmed some of the video sequences.

References

1. N. Jacobs, B. W., N. Fridrich, A. Abrams, K. Miskell, B. Brswell, A. Richardson, R. Pless, The global network of outdoor webcams: Properties and apllications, in: ACM International Conference on Advances in Geographic Information Systems, 2009.

2. S. G. Narasimhan, S. K. Nayar, Vision and the atmosphere, International Journal of Computer Vision 48 (3) (2002) 233–254.
3. N. Hautière, J.-P. Tarel, J. Lavenant, D. Aubert, Automatic fog detection and estimation of the visibility distance through use of an onboard camera, Machine Vision Applications, 17 (1) (2006) 8–20.
4. K. Garg, S. Nayar, Vision and rain, International Journal of Computer Vision 75 (1) (2007) 3–27.
5. N. Hautière, J. Bossu, E. Bigorgne, N. Hiblot, A. Boubezoul, B. Lusetti, D. Aubert, Sensing the visibility range at low cost in the SAFESPOT road-side unit, in: ITS World Congress, 2009.
6. H. Hase, K. Miyake, M. Yoneda, Real-time snowfall noise elimination, in: IEEE International Conference on Image Processing, Vol. 2, 1999, pp. 406–409.
7. X. Zhang, H. Li, Y. Qi, W. Kheng, T. Khim, Rain removal in video by combining temporal and chromatic properties, in: IEEE International Conference on Multimedia and Expo, 2006, pp. 461–464.
8. N. Brewer, N. Liu, Using the shape characteristics of rain to identify and remove rain from video, in: Joint IAPR International Workshop SSPR & SPR, Orlando, USA, Vol. 5342 of Lecture Notes in Compter Science, Springer, 2008, pp. 451–458.
9. P. Barnum, S. Narasimhan, T. Kanade, Analysis of rain and snow in frequency space, International Journal of Computer Vision 86 (2-3) (2010) 256–274.
10. H. Sakaino, Y. Shen, Y. Pang, L. Ma, Falling snow motion estimation based on a semi-transparent and particle trajectory model, in: IEEE International Conference on Image Processing, 2009, pp. 1609–1612.
11. J. Halimeh, M. Roser, Raindrop detection on car windshield using geometric -photometric environment construction and intensity-based correlation, in: IEEE Intelligent Vehicles Symposium, 2009, pp. 610–615.
12. C. Stauffer, W. Grimson, Learning patterns of activity using real-time tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 747–757.
13. S. C. Cheung, C. Kamath, Robust techniques for background substraction in urban traffic video, in: Video Communications and Image Processing, SPIE Electroning Imaging, 2004, pp. 881–892.
14. A. Elgammal, R. Duraiswami, H. D., L. S. Davis, Background and foreground modeling using non-parametric kernel density estimation for visual surveillance, Proceedings of the IEEE 90 (7) (2002) 1151–1163.
15. A. Mittal, A. Monnet, N. Paragios, Scene modeling and change detection in dynamic scenes : A subspace approach, Computer vision and image understanding 113 (1) (2009) 63–79.
16. A. Tavakkoli, M. Nicolescu, G. Bebis, M. Nicolescu, Non-parametric statistical background modeling for efficient foreground region detection, Machine Vision and Applications 20 (6) (2009) 395–409.
17. R. O. Duda, P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM 15 (1972) 11–15.
18. E. Parzen, On estimation of a probability density function and mode, The Annals of Mathematical Statistics 33 (1962) 1065–1076.
19. N. Dallal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 886–893.
20. R. Deriche, Using canny's criteria to derive an optimal edge detector recursively implemented, International Journal of Computer Vision 2 (1) (1987) 167–187.
21. R. Safee-Rad, K. C. Smith, B. Benhabib, I. Tchoukanov, Application of moment and Fourier descriptors to the accurate estimation of elliptical-shape parameters, Pattern Recognition Letters 13 (1992) 497–508.
22. K. Garg, S. Nayar, Photorealistic rendering of rain streaks, ACM Transations on Graphics 25 (3) (2006) 996–1002.
23. N. Dean, A. Raftery, Normal uniform mixture differential gene expression detection for cDNA microarrays, BMC Bioinformatics 6 (173) (2005) 1–14.
24. R. E. Kalman, R. S. Bucy, New results in linear filtering and prediction theory, Transactions of the ASME - Journal of Basic Engineering 83 (1961) 95–107.
25. R. Kohavi, F. Provost, Glossary of terms, Machine Learning 30 (1998) 271–274.
26. J. Swets, Measuring the accuracy of diagnostic system, Science 240 (4857) (1988) 1285–1293.
27. D. Gruyer, C. Royère, J.-M. Blosseville, G. Michel, N. Du Lac, SiVIC and RTMaps, interconnected platforms for the conception and the evaluation of driving assistance systems, in: ITS World Congress, 2006.
28. W. E. K. Middleton, Invention of the Meteorological Instruments, John Hopkins Press, 1969.
29. D. Hauser, P. Amayenc, B. Nutten, P. Waldteufel, A new optical instrument for simultaneous measurement of raindrop diameter and fall speed distributions, Journal of Atmospheric and Oceanic Technology 1 (1984) 256–269.
30. J. B. Edwards, The relationship between road accident severity and recorded weather, Journal of Safety Research 29 (4) (1998) 249–262.
31. O. M. J. Carsten, F. N. Tate, Intelligent speed adaptation: accident savings and cost-benefit analysis, Accident Analysis and Prevention 37 (2005) 407–416.
32. R. Gallen, N. Hautière, S. Glaser, Advisory speed for intelligent speed adaptation in adverse conditions, in: IEEE Intelligent Vehicles Symposium, 2010, pp. 107–114.
33. R. Brignolo, L. Andreone, G. Burzio, The SAFESPOT Integrated Project: Co-operative systems for road safety, in: Transport Research Arena, 2006.
34. L. Wixson, K. Hanna, D. Mishra, Improved illumination assessment for vision-based traffic monitoring, in: IEEE International Workshop on Visual Surveillance, Vol. 2, 1998, pp. 34–41.
35. R. Kurata, H. Watanabe, M. Tohno, T. Ishii, H. Oouchi, Evaluation of the detection characteristics of road sensors under poor-visibility conditions, in: IEEE Intelligent Vehicles Symposium, 2004, pp. 538–543.
36. R. T. Tan, Visibility in bad weather from a single image, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.
37. K. He, J. Sun, X. Tang, Single image haze removal using dark channel prior, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1956–1963.
38. J.-P. Tarel, N. Hautière, Fast visibility restoration from a single color or gray level image, in: IEEE International Conference on Computer Vision, 2009, pp. 2201–2208.
39. C. Tisse, H. Nguyen, R. Tessières, M. Pyanet, F. Guichard, Extended depth-of-field (edof) using sharpness transport across colour channels, in: SPIE Conference, Vol. 7061, 2008.