A FAST ALGORITHM FOR RAIN DETECTION AND REMOVAL FROM VIDEOS

Minmin Shen, Ping Xue

School of Electrical& Electronic Engineering,
Nanyang Technological University, Nanyang Avenue, Singapore 639798.
Emails: mmshen@ntu.edu.sg EPXue@ntu.edu.sg

ABSTRACT

Detection and removal of rain is important in outdoor surveillance vision systems, since the appearance of rain strikes degrades the performance of various vision-based applications. The existing algorithms address the issue of detecting rain only in the irradiance light field, thus require dozens of successive frames to compute the temporal correlation of rain. Combining the properties of rain in irradiance light field and motion field, this paper presents a new approach for rain detection and removal using only three successive frames. In this approach, motion data are used to differentiate rain from other moving objects. A smoothing method based on anisotropic diffusion is proposed for rain removal. Experimental results verify the efficacy of our algorithm.

***Index Terms*—** Rain detection and removal, video processing, optical flow

1. INTRODUCTION

Bad weather conditions, such as rain, snow, fog and mist, should always be considered in outdoor surveillance vision systems. The performance of tracking, recognition, navigation, etc. could be seriously affected by weather conditions. It is essential for an all-weather outdoor vision system to preprocess the acquired video by removing the visual effects of various weather conditions. This paper focuses on the problem of rain.

Real-time processing is required in some vision-based applications, e.g. driver assistant systems. In such cases, the preprocessing for removing the visual effect of rain should be fast and within a small size of group of picture (GOP). The existing algorithms for rain detection and removal seek the solution only in the irradiance light field and require a large number of successive frames to compute the temporal correlation. The detection of rain is a difficult task because both rain and moving objects produce sharp intensity changes. Garg and Nayar [1] studied the dynamics of rain in irradiance light field and detect rain-affected pixels as the ones with high directional temporal correlation. However, a large number of successive frames are needed for computing the temporal correlation (e.g. 30 frames used in [1]). In [2], Zhang et al. exploited the chromatic property of rain, which states that the changes of R, G and B values of rain-affected pixels are approximately the same. For rain removal, they collect the intensity of each pixel over the entire video (e.g. 100 frames used in [2]) to compute its histogram. The set of intensity values of each pixel is divided into background and rain using K-means clustering with K=2. Apparently, this algorithm is not suitable for real-time implementation. In addition, the chromatic property is not available for gray-level videos. Another study of Garg and Nayar investigated the removal of rain by properly setting a camcorder's parameters such as exposure time and aperture [3], which may not be available in some consumer-level camcorders, and the performance could deteriorate at heavy rains. It is proposed in [4] to use Kalman filter for rain removal, but the detection is not addressed. In this paper, we address the problem of rain detection in both irradiance light field and motion field by investigating the property of rain in motion field, and use it to detect rain within two frames.

Once the region of rain is detected, the rain-affected pixels are filtered to remove the fluctuation of intensity. As an efficient smoothing method with meaningful edges preserved, anisotropic diffusion is widely explored in image de-noising. A piecewise smoothing method is proposed by Perona and Malik aiming to localize edges indicated by the gradient of the image [5]. An anisotropic diffusion equation is constructed encouraging smoothing within a region in preference to smoothing across the boundaries. Its extension to the time domain for the removal of noised in video has also been proved to be effective [6]. The temporal derivative is introduced to involve the adjacent frames when smoothing the current frame, thus the three-dimensional diffusion is useful when applied to moving pictures. In this work, the rain removal is applied on the pixels in the detected rain region. For each pixel in the rain region, its value is updated by the weighted sum of the pixels of its temporal-spatial neighborhood. The weights are computed as the diffusion coefficients based on anisotropic diffusion. Three successive frames are required for processing each frame.

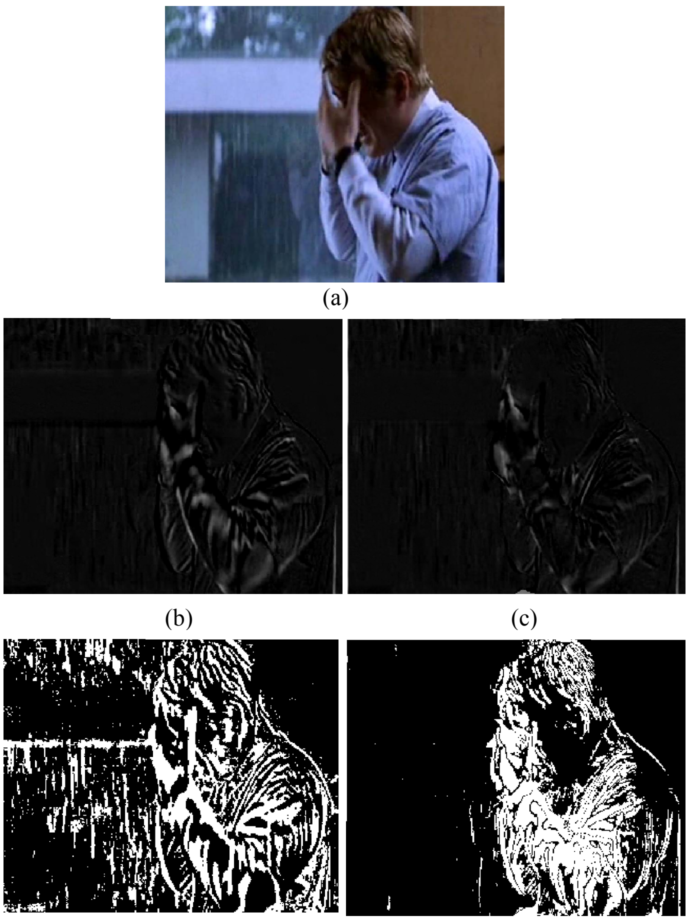
978-1-61284-350-6/11/$26.00 ©2011 IEEE

The paper is organized as follows: Section II briefly reviews the existing work on rain detection considering the properties of rain in irradiance light field, the properties of rain in motion field is then analyzed, and incorporated for detection of rain region. A new rain removal method based on anisotropic diffusion is proposed in Section III. Experimental results are provided in Section IV to verify the efficacy of the proposed algorithm. Section V gives the conclusion.

2. RAIN DETECTION

2.1. Rain Characters Used for Rain Detection

2.1.1. Irradiance Light Field



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

Fig.1. (a) Current frame I_n . (b) Difference between the previous frame I_{n-1} and the current frame I_n . (c) Difference between the registered previous frame $\tilde{I}_n$ and the current frame I_n . (d) Pixels satisfying $\left|I_n - I_{n-1}\right| > 3$ (e) Pixels satisfying $\left|I_n - \tilde{I}_n\right| > 10$ distinguish from rain affected pixels.

The physical and photometric properties of rain is analyzed in [1] and used for rain detection and removal in videos based on the assumptions that all rain drops have the same size and fall at almost the same velocity relative to camera. However, as pointed out in [2], the algorithms in [1] could fail if these assumptions do not hold. Zhang et al. [2] studied the chromatic property of rain and showed that the change of each chromatic component ($\Delta R, \Delta G, \Delta B$) are approximately the same for rain-affected pixels. This interesting property, however, cannot be used for rain detection in grey-level videos.

As exploited in both [1] and [2], an apparent property of rain-affected pixels is that a rain drop produces a positive intensity change. And this property is also used in this paper.

2.1.2 Motion Field

Motion field is a 2-D vector field, which is the perspective projection on the image plane of the 3-D velocity field of a moving scene [7]. Independent moving objects could be detected using motion segmentation [8]. As rain consists of a large number of drops, each of which behaves like a transparent sphere of very small size, it results in intensity fluctuations in videos. This property of rain facilitates our investigation in distinguishing rain and other moving objects, and is used for detect rain in a simple way.

Optical flow, defined as an estimation of motion field, is derived from the first order variation of image brightness pattern. Let the image brightness at the point (x, y) in the image plane at time t be denoted by $I(x, y, t)$. Usually, the brightness of a particular point in the pattern is constant, thus we have

$$I_x u + I_y v + I_t = 0 \tag{1}$$

where I_x , I_y and I_t denote the partial derivatives of image brightness with respect to x , y and t respectively, and u, v are pixel displacements in the x and y directions (i.e. $u = \partial x / \partial t, v = \partial y / \partial t$) .

Optical flow is accurate for estimating motion field when the sharp changes in intensity over time are due to physical events on the moving surface, thus is capable of estimating motion trajectories of most moving objects in a sub-pixel precision [9]. Therefore, the registration (motion compensation) could be applied using the optical flow. The flow velocity $(u_{n-1,n}, v_{n-1,n})$ from the previous frame I_{n-1} to the current frame I_n could be computed using some optical flow techniques as [8]. The previous frame I_{n-1} is then registered for I_n by

$$\tilde{I}_n(x,y) = I_{n-1}(x+u_{n-1,n}, y+v_{n-1,n}) \quad (2)$$

For most moving objects, optical flow is accurate for estimating the relative displacement between two adjacent frames, thus is used for sub-pixel registration. But for rain, which is an ensemble of rain drops falling at high velocities, its motion trajectory is unlikely to be estimated accurately by optical flow. Fig.1 is shown to demonstrate this property of rain: (b) is the difference between I_n and I_{n-1}, and (c) shows the difference between I_n and $\tilde{I}_n(x,y)$. It can be seen that the intensity change due to the moving person is significantly reduced after registration, while the one caused by rain remains. For clarity, (d) shows the Pixels satisfying $|I_n - I_{n-1}| > T_1$, and (e) shows the pixels whose value changes before and after registration is greater than 10 ($|I_n - \tilde{I}_n| > T_2$). The threshold T_1 is set to be 3 following [1], and T_2 is set to be 10, which gives the best results in our experiments. The thresholds are used in all of our experiments. Compared to (d), where only the property of rain in irradiance light field is considered, (e) distinguishes the intensity change caused by rain from the other moving objects with the help of motion data.

2.2. Rain Detection

As discussed in Section 2.1, the properties of rain in both irradiance light field and motion field are explored for rain detection. In (e), most of the detected pixels are on or around the moving person, and the rain-affected pixels are almost excluded. After applying dilation to both (d) and (e), we can obtain the region R_1 including both moving person and rain-affected pixels in (d) and the one R_2 with only moving person in (e). The rain region could then be determined as $R_1 - R_2$. The proposed algorithm only requires storing one previous frame to detect the rain region in the current frame. For computational complexity, optical flow computation is the main concern, and some work has been reported in computing the optical flow in video sequences using advanced VLSI system architecture [10]. Therefore, the proposed algorithm is feasible for real-time rain detection.

3. RAIN REMOVAL

3.1. Rain Characters Used for Rain Removal

Given the rain region detected, rain removal is applied by replacing the rain-affected pixels with new pixel values. A simple scheme is employed in [1] using the average of the temporal neighboring pixels ($(I_{n-1} + I_{n+1})/2$) or the spatial neighboring pixels (on either side) as the substituted pixel value. The work in [2] takes only the temporal correlation into account and uses the K-means clustering to classify the pixels in the same location over the entire video into background and rain. The result for rain removal is improved over [1] at the cost of an increased computational complexity as well as an increased storage (a large set of video frames are stored for removing rain from each frame). In this paper, we exploit both temporal and spatial correlation and propose an algorithm to remove the visual rain effects using three frames.

As pointed out in [1] that two properties of rain could be used for rain removal: 1) the same pixel may not be affected by raindrops in more than three consecutive frames, and 2) a drop produces a positive intensity change of unit frame width at a pixel. Thus the pixel value of the rain-affected pixel could be estimated by a weighted sum of the pixels from both temporal and spatial neighborhood. The algorithm in [1] simply estimates the new pixel value to be half of the sum of two pixels of either temporal or spatial neighborhood, as it is believed that these neighboring pixels are of background. Given the background color and rain color after classifying, a similar scheme is also proposed in [2] by replacing the rain-affected pixel with the α -blending of its rain-affected color and background color, which is also the weighted sum of the temporal neighboring pixels (pixels at the same location through the whole video). It should be noted that the property 2 of rain is not always held if it appears to be out-of-focus streak, and we will show in Section 4 that the proposed algorithm can also handle this scenario.

3.2. Removal of Rain

Let us consider the three-dimensional 26-connected neighborhood. As shown in Fig.2, for each pixel $v_n(x,y)$ located at the coordinate (x,y) from the n -th frame, its 26-connected neighborhood consists of the spatial 3×3 neighborhood and the pixels at the same locations from the previous and next frames. The rain removal is applied by updating each pixel in rain region with the weighted sum of its spatial-temporal neighborhood. We propose an edge-preserving algorithm based on anisotropic diffusion to successively adjust each pixel. At each iteration, the new pixel value is determined by a weighted average of its neighboring pixels with proper weights if it is smaller than the old one, i.e.

$$I_n^{t+1}(x,y)=\begin{cases}\dfrac{\sum_{v_m(i,j)\in V} w_{x,y}(i,j)I_m^t(i,j)}{\sum_{v_m(i,j)\in V} w_{x,y}(i,j)}, & if \quad I_n^{t+1}(x,y)<I_n^t(x,y)\\ I_n^t(x,y), & otherwise\end{cases} \tag{3}$$

where $w_{x,y}(i,j)$ are diffusion coefficients, V denotes the set of neighborhood, $v_m(i,j)$ is a vector representing the location of each pixel, and t is the elapsed time of diffusion. The old pixel value is updated only when the new one is smaller than it, since the pixel value of background is smaller than the rain-effected pixel according to the second property stated in Section 3.1. All the pixels in the rain region are updated in a raster scanning order, thus the smoothing takes effect even the rain streaks are at the width larger one pixel. The diffusion coefficient is determined by magnitude of gradients to emphasize the structural content, i.e.

$$w_{x,y}(i,j)=\frac{1}{A}\exp\left(-\frac{d(v_n(x,y),v_m(i,j))}{4t}\right), \forall v_m(i,j)\in V \tag{4}$$

where A is a constant for normalization, and

$$d(v_n(x,y),v_m(i,j)) = (v_n(x,y)-v_m(i,j))^T\Psi^{-1}(v_n(x,y)-v_m(i,j)) \tag{5}$$

The diffusion tensor Ψ is used to perform the smoothing of the manifold constrain implicitly [11]. Such a diffusion

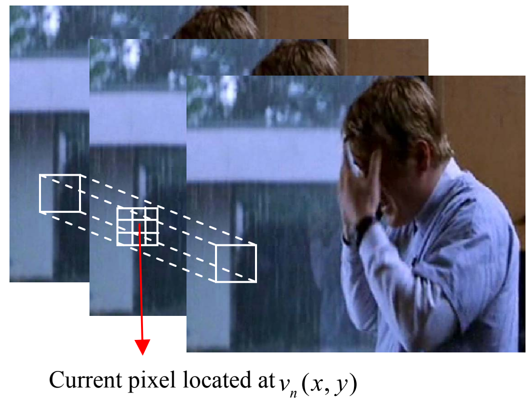


Fig.2. Current pixel at the location (x,y) of the n-th frame and its 26-connected neighborhood

tensor is used in this paper based on a temporal-spatial

(a) (b) (c)

Fig.3. Sample frames from the movie "Magnolia" of (a) original video, (b) result of rain removal and (c) detection results.

(a) (b) (c)

Fig.4. Comparison with the existing algorithm: sample frames from the movie "Magnolia" of (a) original video, (b) result of the proposed rain removal algorithm and (c) the result of [1].

feature combining both temporal and spatial derivatives. As it involves adjacent frames in the diffusion process, the fluctuation due to rain is smoothed, leading to the suppression of the visual rain effect.

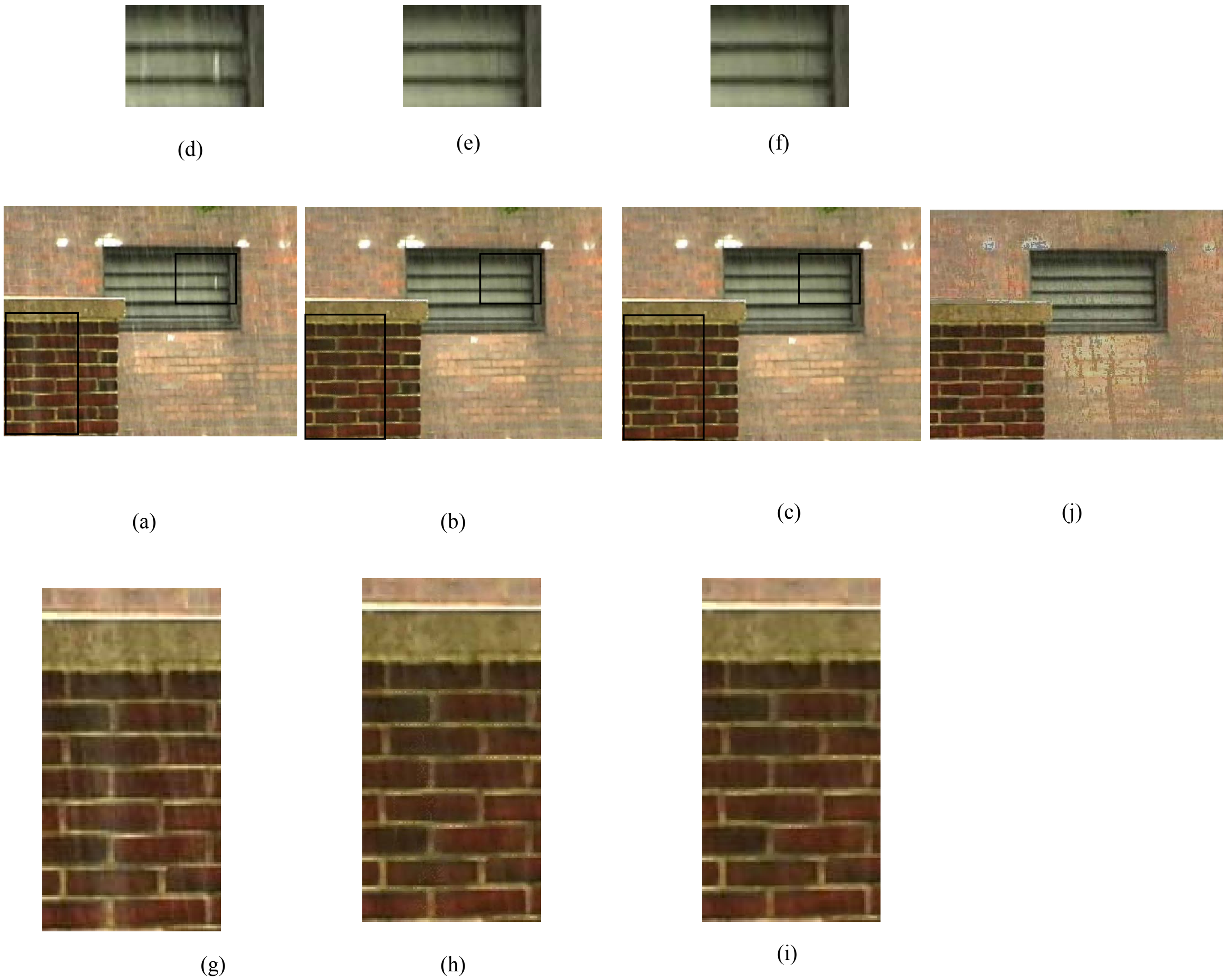


Fig.5. (a) Original frame (b) Rain removal output after 1 iteration (c) Rain removal result after 3 iterations, (d) (e) (f) are the enlarged view of in-focus streak (the region within the black square of the window), and (g) (h) (i) are the enlarged view of out-of-focus one (the region within the black square of the wall). For comparison, we also show the result of [1] in (j), which is obtained from [4].

4. EXPERIMENTAL RESULTS

We use two sequences as the testing videos for examining our algorithm of rain detection and removal. For rain removal, we set the number of iteration in (3) as 3 for every frame in each sequence. In Fig.3, a sample frame of the movie "Magnolia" are shown to demonstrate that the proposed algorithm detects the rain region accurately, and the result of rain removal is satisfying. The edges remain sharp in the rain region, which verifies the edge preserving of the proposed smoothing method. Our result of rain removal is compared with the result of [1], which is available in [12], and we found that it is slightly inferior as some vertical marks are still visible in Fig.4. The edge-preserving nature of the proposed rain removal algorithm suppresses the "abrupt" positive intensity change, while preserving the streak-shape marks. We also found that our algorithm performs very well when rain is densely distributed, as shown in Fig.5. The result looks even better than the result of [1], which is presented in [4], since the photometric model of [1] is no longer held in this scenario [2].

The second experiment is constructed to verify the efficacy of the proposed algorithm for removing rain which is distributed over a wide range of depth. The scene shown in Fig.5 includes both streaks in-focus (enlarged view in (d)) and out-of-focus (enlarged view in (g)). It is noted that the rain streak out-of-focus in (d) is very near to the camera, and appears uneven brightness within a band wider than one pixel. Although the output of (3) after one iteration contains some brighter pixels (as shown in (h)), the rain removal result of (3) after three iterations gives a satisfactory view with the uneven brightness removed. On the other hand, the white rain streak in (g) is also removed successfully by the proposed algorithm. The rain in this sequence is densely distributed, and our rain removal is better than [1] (as shown in (j)), which has uneven brightness in the rain region.

5. CONCLUSION

We proposed in this paper an efficient algorithm for rain detection and removal from videos, as rain could seriously affect the performance of outdoor surveillance vision systems. The new algorithm can process each frame with the help of only three successive frames, comparing with some existing algorithms where dozens of frames are needed. Combined with some recent VLSI development, a real-time rain detection and removal can be achieved. For rain detection, we explored the properties of rain in motion field, based on which a simple but efficient algorithm is developed to differentiate rain from other moving objects with which the rain detection is fulfilled in two frames only. As for removal of the pixels within the detected rain region, an anisotropic diffusion smoothing method is proposed as a temporal-spatial filter. In this filtering procedure, each pixel in the current frame takes into account spatial neighboring pixels and some pixels from the temporal neighboring frames, thus three frames are used. Experimental results have verified that the proposed algorithm detects the rain region accurately and produces the satisfactory rain removal results.

6. REFERENCES

[1] K. Garg and S.K. Nayar, "Detection and removal of rain from videos," in *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 528-535, 2004.

[2] X. Zhang, H. Li, Y. Qi, W.K. Leow and T.K. Ng, "Rain Removal in Video by Combining Temporal and Chromatic Properties," in *IEEE Int. Conf. Multimedia and Expo,* pp. 461-464, 2006.

[3] K. Garg and S.K. Nayar, “When does a camera see rain?”in *IEEE Int. Conf. Computer Vision*, vol. 2, pp. 1067–1074, 2005.

[4] W.J. Park and K.H. Lee, “Rain removal using Kalman filter in video,” in *Int. Conf. Smart Manufacturing Application,* pp. 494-497, 2008.

[5] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion”, *IEEE trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 629–639, 1990.

[6] S.H. Lee and M.G. Kang, “Spatio-temporal Video Filtering Algorithm based on 3-D Anisotropic Diffusion Equation,” in *IEEE Int. Conf. Image Processing*, vol.2, pp. 447 – 450, 1998.

[7] A. Verri and T. Poggio, “Motion Field and Optical Flow: Qualitative Properties”, *IEEE trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 5, May 1989.

[8] A.S. Ogale, C. Fermuller, and Y. Aloimonos, “Motion Segmentation Using Occlusions”, *IEEE trans. Pattern Anal. Mach. Intell.*, vol. 27, no.6, Jun. 2005.

[9] C. Wang, P. Xue, and W. S. Lin, "Improved Super-Resolution Reconstruction from Video," *IEEE trans. Circuits and Systems for Video Technology,* vol. 16, no. 11, pp. 1411- 1422, Nov. 2006.

[10] K. Bhattacharya, M. Venkataraman, and N. Ranganathan, "A VLSI system architecture for optical flow computation," in *IEEE Int. Symp. Circuits and Systems*, pp.357-360, 2009.

[11] A. F. E. Ouafdi, D. Ziou, and H. Krim. “A smart stochastic approach for manifolds smoothing,” in *Comput. Graph. Forum*, vol.27, pp.1357–1364, 2008.

[12] http://www1.cs.columbia.edu/CAVE/projects/rain_detection/rain_detection.php