A Rain Detection and Removal Algorithm Based on Rainy Intensity for Videos in Heavy Rainy Scene

Chao-qi Ma and Zheng-fa Liang

Abstract Detection and removal of rain for videos is a challenging problem because of the difficulty to build an appropriate imaging model for rainy weather. By far, almost all the algorithms proposed for rain removal considered a certain pixel in a video frame as rain affected pixel or non-rain affected pixel determinedly, which is not reasonable for heavy rainy scene. Because in the heavy rainy scene, almost all the pixels are covered by rain more or less. Instead of strictly considering a pixel as either rain covered or non-rain covered, the proposed algorithm calculates the rainy intensity for each pixel. The rainy intensity is soft information ranging from zero to one which contains more information about the rain. Furthermore, the rainy intensity can help estimate the original value of pixels that are covered by rain. Experiment results show that the rain removal videos processed by the proposed algorithm has much better vision effect than by other existing algorithms in heavy rainy scene.

Keywords Rain detection · Rain removal · Rainy intensity · Heavy rainy scene

1 Introduction

Weather conditions have great influence on the imaging quality of vision systems, thus making many applications of video analysis and computer vision such as motion detection, object tracking, video surveillance, and robot navigation and so on, lapse. It is a meaningful work to improve the image quality of videos degraded by bad weather conditions.

In general, weather conditions are broadly classified as steady (fog, mist and haze) or dynamic (rain and snow) according to the sizes of individual droplets [1, 2]. For the steady weather condition, taking haze removal for example, researchers have developed lots of algorithms based on the atmosphere scatter model, and have gotten very good results [3]. But there is no any imaging model for the rain removal

C. Ma (✉) · Z. Liang
National University of Defense Technology, Changsha 410073, China
e-mail: machaoqi1990@hotmail.com

B.-Y. Cao et al. (eds.), *International Conference on Oriental Thinking and Fuzzy Logic*, Advances in Intelligent Systems and Computing 443,
DOI 10.1007/978-3-319-30874-6_61

problem. The common assumption is that the image degraded by rainy weather is composed of the original image and the rain drop. So the task is to detect all the regions covered by rain for every frame in a video, and then use the spatial or temporal information to recover the original value of the rain affected pixels. Based on this assumption, many good algorithms have been proposed.

Garg and Nayar [4] are the first researchers dealing with the rain removal problem. They analyzed the physical and photometric properties of the rain, finding that there is a positive intensity change when the pixel is covered by rain, and the rain drops in a video have the uniform velocities and directions.

Zhang et al. [5] proposed an algorithm that incorporates both temporal and chromatic properties of rain in video. The temporal property states that an image pixel is never always covered by rain throughout the entire video. The chromatic property states that the changes of R, G, and B values of rain affected pixels are approximately the same. For rain removal, they used the entire video to compute its intensity histogram of the pixel, and then used K-means clustering with $K = 2$ to divide the pixel to background and rain. The new color of a pixel affected by rain is replaced by the -blending of its rain-affected color and background color.

The models proposed in early years cannot work well in many occasions, such as highly dynamic scene, heavy rainy scene, and give limited performance. In recent years, some robust algorithms have been proposed.

Tripathi [6] proposed a probabilistic spatiotemporal approach for detection and removal of rain. They found that the time evolution of intensity values of a pixel at particular position present in rain region for consecutive frames is quite different from the evolution of pixel present in moving object region. And they used two statistical features, i.e. intensity fluctuation range, and spread asymmetry collected from spatial-temporal neighborhood to detect the rain affected pixels. But this method still gives lots of false detections.

Zhao [7] proposed a pixel-wise framework combining a detection method with a removal approach. In their work, dynamic weather conditions are detected by a strategy-driven state transition, which integrates static initialization using K-means clustering with dynamic maintenance of Gaussian mixture model. Moreover, a variable time window is presented for removal of rain and snow. However, this method is sensitive to fast intensity change and is of high computing complexity.

Chen [8] proposed rain pixels recovery algorithm based on motion segmentation scheme, that each pixels dynamic property as well as motion occlusion clue is considered; both spatial and temporal information are adaptively exploited during rain pixel recovery. This method can deal with rainy scenes with large motion very well. But the rain detection in this method is too simple, and it still cannot give perfect result for heavy rainy scene.

Zhou et al. [9] proposed a rain removal algorithm based on optical flow and hybrid properties constraint. This approach firstly identified the candidate rain pixels by adopt the optical flow, and then used hybrid properties constraint of raindrop to refine the rain streaks. Once the rain streaks were detected, the scene can be restored by a weighted composing method.

The common problem shared by the above algorithms is that they considered a pixel as rain affected or not determinedly. It is not reasonable especially for heavy rainy scene. In a heavy rainy scene, almost all the regions are covered by rain more or less, that is the rain is lighter in some regions, and heavier in other regions. We should establish a model to estimate the rain heavy level of each pixel in a frame. The advantage of this model is that all the pixels including rain pixels and non-rain pixels, can be deal with in a unique framework. As a special case, we can simply consider the rain heavy level of background pixels as zero. And in the rain removal stage, we can estimate the original value of the rain affected pixels with the help of this model naturally.

In this paper, we proposed a model to estimate the rain heavy level with two features extracted from the time evolution information of a pixel. The rain heavy level is called rainy intensity. We calculate each pixels rainy intensity for each frame in a video, instead of detecting its pixels as rain affected or not. The rainy intensity is later used to help estimate the original value of a pixel. Experiment results show that the proposed algorithm performs better than existed algorithms in heavy rainy scene.

The rest of the paper is organized as follows. Section 2 describes the models establishing process based on analyzing the rains characters. Section 3 will show the rain removal framework based on the proposed model. In Sect. 4, we compare our algorithm with some existed excellent algorithms.

2 Rain Analysis and Rainy Intensity

In our work, we suppose that the camera is static and the video is stable. If the video is twittering, it always can be stabilized using algorithms like video stabilization [10]. And we suppose the rain drops are randomly distributed in the space just as many other researchers do.

In a rainy video, a main phenomenon of a pixel affected by rain is temporal fluctuations [4], that is, rain gives positive fluctuations in the intensity values without affecting the chrominance values. Here, we first study the time evolution for pixels of rain affected, background and motion objects. Then for the rain affected pixels, we consider the difference between heavy rain and light rain.

Figure 1 shows the time evolution of pixels that belong to rain, background, and motion object in the 6th frame (Fig. 1a) of 11 continuous frames respectively in a highly dynamic scene. For each situation, we take three pixels into consideration. The precise values of these pixels are listed in Table 1.

It is obviously that if a pixel is affected by rain, there is a positive intensity fluctuation along the pixels time evolution and its former and latter intensity values are almost symmetric. If the pixel belongs to background, almost no changes in intensity can be detected. And for motion objects, there is no regular pattern can be found.

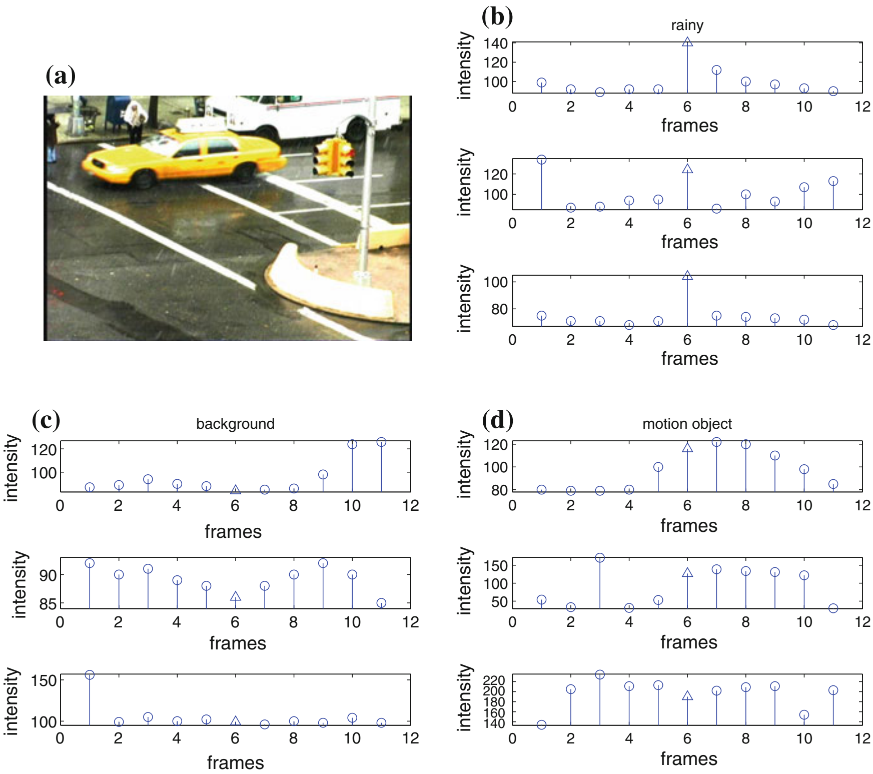

Fig. 1 The time evolution of pixels intensity. **a** The current frame. **b** Pixels affected by rain. **c** Pixels belong to background. **d** Pixels covered by motion object

Table 1 The specific values of the pixels in Fig. 1

	$N-2$	$N-1$	N	$N+1$	$N+2$
Rainy	92	92	140	112	100
	94	95	124	86	100
	68	71	104	75	74
Background	90	88	84	85	86
	89	88	86	88	90
	100	102	99	96	100
Motion object	80	100	116	122	120
	217	190	235	187	173
	211	213	190	202	209

Similar results can be found in A.K. Tripathi's work [6]. But they thought that the symmetric feature can be still retain when consider the pixels spatial neighbors thus using a spatiotemporal window to measure the pixels symmetric property. In fact, in the spatial domain, motion objects can present better symmetric property because

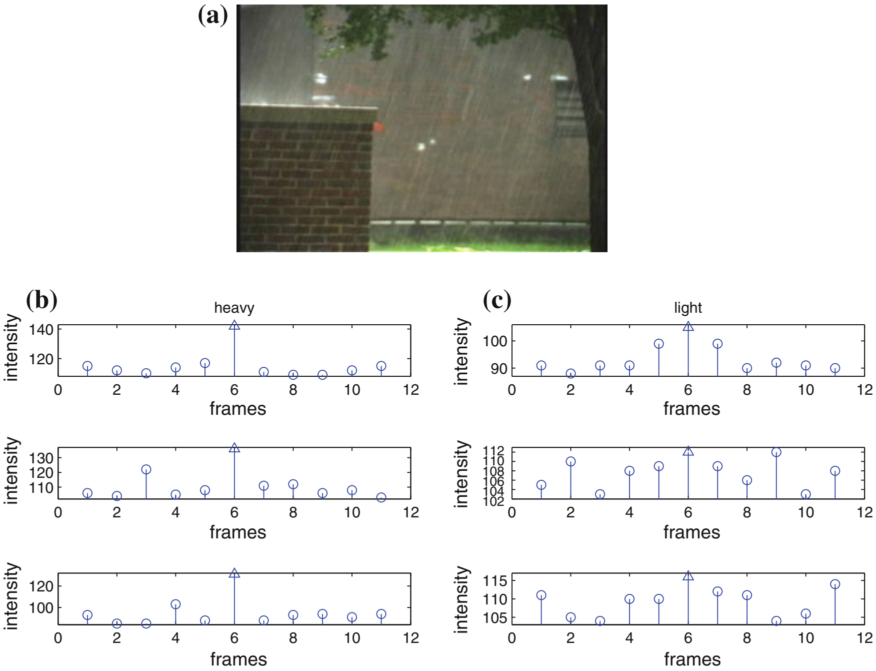

Fig. 2 The time evolution of pixels intensity. **a** The current frame. **b** Pixels affected by light rain. **c** Pixels affected by heavy rain

usually objects have the same color in large continuous regions. So in our work, we will not include spatial information to measure one pixels symmetric property, but only consider its temporal information.

Next, we study the difference between heavy rain and light rain in a heavy rainy scene. In actual world, when its rainy outside, rain drops exist in different visual depth. So a more reasonable assumption is that all the pixels in a frame are covered by rain, and for individual pixels, their rain intensity is different. Figure 2 shows the differences in pixels intensity which are covered by light rain and heavy rain respectively.

Table 2 gives the specific values of the pixels in Fig. 2. We found that the positive fluctuation of pixels affected by heavy rain is much bigger than that of pixels affected by light rain. So we propose two features extracted from continuous five frames to distinguish rain, background, and motion object, furthermore to distinguish heavy rain and light rain. The two features are named average positive intensity fluctuation $A(x)$, and symmetric level $B(x)$.

Table 2 The specific values of the pixels in Fig. 1

	$N-2$	$N-1$	N	$N+1$	$N+2$
Light	91	99	105	99	90
	108	109	112	109	106
	110	110	116	112	111
Heavy	114	117	142	111	109
	105	108	136	111	112
	103	88	131	88	93

$$A(x) = \begin{cases} \dfrac{mean\{d_n\}}{A_0}, & mean\{d_n\} \geq 0, n = -2, -1, 1, 2 \\ 0, & other \end{cases} \tag{1}$$

$$B(x) = \begin{cases} \dfrac{2d_{-1}d_1}{d_{-1}^2 + d_1^2 + d_0} \cdot \dfrac{2d_{-2}d_2}{d_{-2}^2 + d_2^2 + d_0}, & d_n > 0, n = -2, -1, 1, 2 \\ 0, & others \end{cases} \tag{2}$$

$$d_{-2} = I_N - I_{N-2}, d_{-1} = I_N - I_{N-1}, d_1 = I_N - I_{N+1}, d_2 = I_N - I_{N+2}$$

Here, A_0 is a constant chosen to guarantee $A(x) \in [0, 1]$. From the definition of $B(x)$ we can easily find that $B(x) \in [0, 1]$. $A(x)$ and $B(x)$ measure the positive fluctuation level and symmetric property of the current pixel over its temporal neighbors, respectively. The higher value means high positive fluctuation or high symmetric, which means the current pixel is more likely to be covered by rain and the rain heavy level is higher. We call the rain heavy level rainy intensity. One possible way to calculate the rainy intensity of a pixel can be:

$$p(x) = \frac{1 - e^{-f(A)g(B)}}{1 - e^{-1}} \tag{3}$$

Here, $f(x)$ and $g(x)$ are increase functions of variable x, and satisfy:

$$f(x) \in [0, 1], g(x) \in [0, 1], for\ x \in [0, 1]$$

For simplicity, we take

$$\begin{cases} f(x) = x \\ g(x) = x \\ p(x) = \dfrac{1 - e^{-AB}}{1 - e^{-1}} \end{cases} \tag{4}$$

This model is simple, but it works.

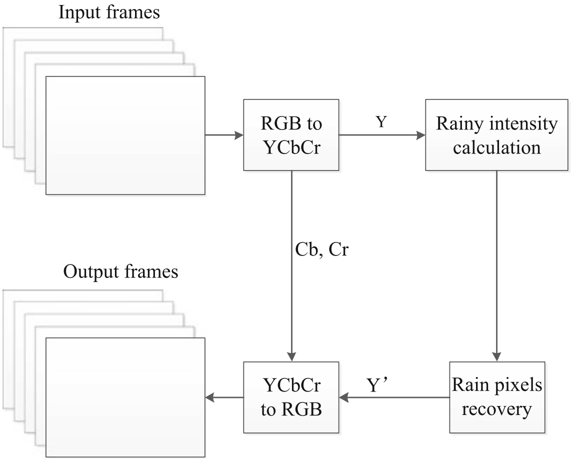

Fig. 3 Main framework of the proposed algorithm for rain removal

3 Rain Removal Based on Rainy Intensity

In this section, we will detailedly describe the rain removal method using the proposed rainy intensity.

The framework of the proposed algorithm is shown in Fig. 3. For the frames in the video taken in rainy weather, RGB to YCbCr conversion is first done so that the rain removal work can be exploited only on the intensity plane. After that, rainy intensity calculation is done to prepare the necessary information for rain pixels recovery. The recovered pixels then form a new intensity plane for current frame. At last, the new intensity plane together with the Cb and Cr component make up the final rain removal frame.

3.1 Rain Intensity Calculation

The rainy intensity of a frame is calculation based on the Eqs. (2)–(4) in Sect. 2. In our method, only five continuous frames and their corresponding rainy intensity are stored in a FIFO to reduce system cost. The unique frame work to calculate the rainy intensity, as well as the recovery method described next, make it easier to apply this algorithm to VLSI implementation.

A problem necessary to discuss is that the method proposed sometime give high rainy intensity value to large motion objects. In addition, motion objects may also be covered by rain drops, and the model is more complicate. In fact, all the algorithms working on pixel level cannot deal with this problem very well. We need some method working on semantic level to distinguish motion objects from the current frame. In [8], motion segmentation is applied and it seems that the results are very well. But the motion segmentation method proposed in [8] is very complicated which involves optical flow, Gaussian Mixture Model (GMM), and image segmentation using k-means clustering. So the algorithm in [8] cannot achieve real time processing. Developing a new simple method to deal with this problem then becomes very meaningful and it is our future work as well.

3.2 Rain Pixels Recovery

After calculating the rainy intensity for every pixel in the current frame, the rain pixels recovery job can be done.

According to the researches done before, the original value of the rain-affected pixel could be estimated by a weighted sum of the pixels from both temporal and spatial neighborhood. But experiments show that using spatial neighborhood to recover the original value of a pixel can blur the frame. So in our work we only use their temporal information to estimate the original value.

Different from other pioneer works, we assume that all the pixels in rainy scene are covered by rain more or less. We cant just simply use the temporal mean, or weighted temporal mean with fixed coefficient. An appropriate way is to adjust the coefficients according to their heavy level, which can be measured by the rainy intensity proposed ahead. So one possible way to estimate the rain affected pixels original value can be:

$$\hat{I}(x) = \frac{\sum_{n=-2}^{2} \left[1 - p_n(x)\right] \cdot I_n(x)}{\sum_{n=-2}^{2} \left[1 - p_n(x)\right]} \tag{5}$$

4 Experiment Results

Experiments are carried out on two videos of heavy rain scene and light rainy scene in *Matlab 2012a* environment. The results are shown in Figs. 4, 5 and 6.

The rainy intensity calculated for these two videos are shown in Fig. 4. From the results we can see that the proposed rainy intensity can measure the heavy level of rain very well.

Figures 5 and 6 show the rain removal results of two state of art algorithms, and compare them with our algorithm. The two algorithms were proposed by Tripathi [6] in 2012, and Chen [8] in 2014 respectively. From the results we can see that the rain streaks are well removed in these two videos. And the visual effect of the frame processed by our algorithm is better than that by other algorithms.

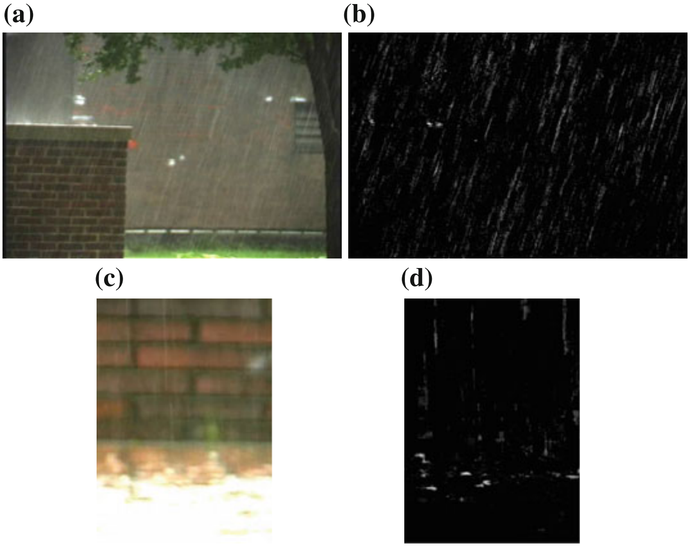

Fig. 4 Simulation results of two video showing the original frames and their corresponding rainy intensity

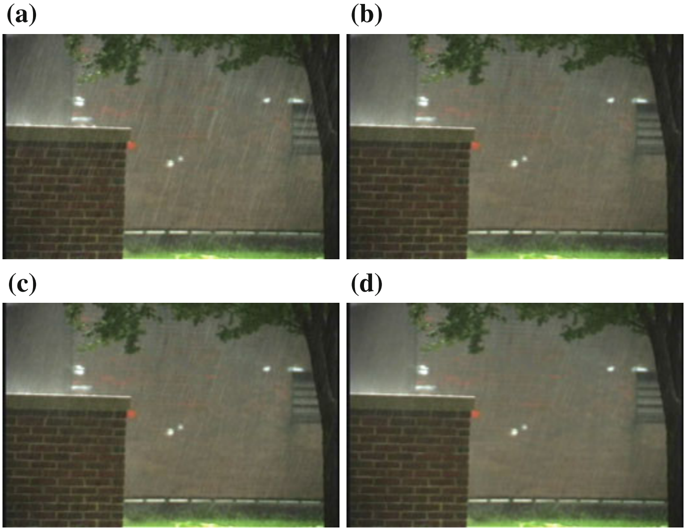

Fig. 5 Comparison on the first video using different rain removal algorithms. **a** Original video frame. **b** A. Tripathi's algorithm using spatial-temporal model. **c** Jie Chens algorithm using motion segmentation and adaptive filters. **d** Result from our proposed algorithm

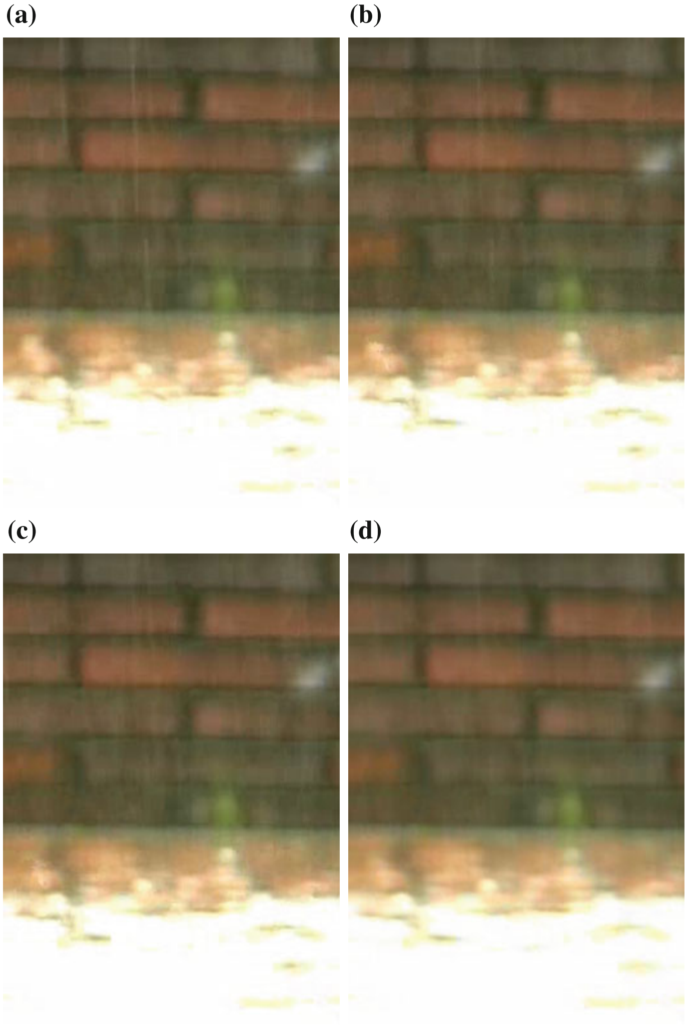

Fig. 6 Comparison on the second video using different rain removal algorithms. **a** Original video frame. **b** A. Tripathi's algorithm using spatial-temporal model. **c** Jie Chens algorithm using motion segmentation and adaptive filters. **d** Result from our proposed algorithm

5 Summary

In this paper, we present a novel method to remove rain streaks in videos based on rainy intensity proposed. The rainy intensity is defined according to the positive intensity fluctuations and intensity symmetric along the pixels time evolution. The rainy intensity measures the rains heavy level very well, and helps to estimate the pixels original value in highly rainy scene. Experiments show that our algorithm can deal with both light and heavy rainy scenes with lower computation. And most importantly the visual effect of video after rain removal by our algorithm is much better than existed algorithms.

References

1. Narasimhan, S.G., Nayar, S.K.: Vision and the atmosphere. Int. J. Comput. Vis. **48**(3), 233–254 (2002)
2. Garg, K., Nayar, S.K.: Vision and rain. Int. J. Comput. Vis. **75**(1), 3–27 (2007)
3. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2341–2353 (2011)
4. Garg, K., Nayar, S.K.: Detection and removal of rain from videos. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2004, vol. 1, pp. I–528. IEEE (2004)
5. Zhang, X., Li, H., Qi, Y., Leow, W.K., Ng, T.K.: Rain removal in video by combining temporal and chromatic properties. In: 2006 IEEE International Conference on Multimedia and Expo, pp. 461–464. IEEE (2006)
6. Tripathi, A.K., Mukhopadhyay, S.: Video post processing: low-latency spatiotemporal approach for detection and removal of rain. IET Image Process. **6**(2), 181–196 (2012)
7. Zhao, X., Liu, P., Liu, J., Tang, X.: Removal of dynamic weather conditions based on variable time window. Comput. Vis., IET **7**(4), 219–226 (2013)
8. Chen, J., Chau, L.-P.: A rain pixel recovery algorithm for videos with highly dynamic scenes. IEEE Trans. Image Proc. **23**(3), 1097–1104 (2014)
9. Zhou, Y., Han, Y., Zhou, P.: Rain removal in videos based on optical flow and hybrid properties constraint. In: 2015 Seventh International Conference on Advanced Computational Intelligence (ICACI), pp. 143–147. IEEE (2015)
10. Matsushita, Y., Ofek, E., Tang, X., Shum, H.-Y.: Full-frame video stabilization. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2005, vol. 1, pp. 50–57. IEEE (2005)