RAIN REMOVAL IN VIDEO BY COMBINING TEMPORAL AND CHROMATIC PROPERTIES

Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, Teck Khim Ng

Dept. of Computer Science, National University of Singapore, Singapore 117543
{zhangxi7, lihao, qiyingyi, leowwk, ngtk}@comp.nus.edu.sg

ABSTRACT

Removal of rain streaks in video is a challenging problem due to the random spatial distribution and fast motion of rain. This paper presents a new rain removal 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. By using both properties, the algorithm can detect and remove rain streaks in both stationary and dynamic scenes taken by stationary cameras. To handle videos taken by moving cameras, the video can be stabilized for rain removal, and destabilized to restore camera motion after rain removal. It can handle both light rain and heavy rain conditions. Experimental results show that the algorithm performs better than existing algorithms.

1. INTRODUCTION

Computer vision of indoor situations has been intensively studied, whereas some outdoor conditions such as rain, snow, and fog remain as challenging problems for vision systems. The problem of rain removal in video sequences is similar to video inpainting [1, 2], which tries to recover occluded objects in video. But the major obstacles are: (1) the occluders, the rain drops in this case, are unknown, and (2) the video quality is seriously degraded by rain. So, it is very difficult to use information of neighboring pixels of occluded areas, as carried out by inpainting algorithms [1, 2].

Garg and Nayar classified the weather into two types [3]: *steady weather* such as fog and haze, and *dynamic weather* such as rain and snow, based on the size of the weather particles. In steady weather, water droplets or smoke particles are very small and steadily float in the air. By modeling the scattering and chromatic effects, Narasimhan and Nayar successfully recovered "clear day" scenes from images taken in bad weather [4].

In dynamic weather, rain drops or snow flakes are randomly distributed in the scene and move all the time. This makes them hard to detect and causes failures in vision applications such as tracking and surveillance. Garg and Nayar analyzed the physical and photometric properties of rain

drops and proposed a method to detect and remove rain in video based on these properties [3]. But their photometric model assumes that all rain drops have the same size and fall at almost the same velocity relative to the camera. If rain becomes heavier or lighter in the video or is distributed over a wide range of depth, their algorithm might fail to distinguish rain from other moving objects. Recently, they studied the image formation of rain drops distributed over different depths, and proposed that by properly setting a camcorder's parameters such as exposure time, aperture, etc., the rain could be removed when recording the video [5]. However, this method cannot handle the scenario of heavy rain and the parameters of consumer camcorders may not be adjustable.

This paper also focuses on the problem of rain removal in video. A new rain removal algorithm that incorporates both temporal and chromatic properties of rain in video is proposed. The temporal property states that an image pixel is not 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. By using both properties, the algorithm can detect and remove rain streaks in both stationary and dynamic scenes taken by stationary cameras. To handle videos taken by moving cameras, the video can be stabilized for rain removal, and destabilized to restore camera motion after rain removal. It can handle both light rain and heavy rain conditions. Test results show that the algorithm performs better than existing methods.

2. PROPERTIES OF RAIN

2.1. Temporal Property

In a natural scene, rain can be regarded as a collection of spherical droplets randomly distributed and moving at a high speed when they are near the ground. When rain drops are very far from the camera, their visual effect is very weak and they appear as fog [5]. So we focus only on rain drops that are close to the camera.

With a pinhole camera model, the length of rain streaks is inversely proportional to the depth of rain drops. Normally, rain streaks span several to tens of pixels in a frame. Due to the speed of rain drops, the same rain streak does not appear in two consecutive frames. Moreover, due to the random dis-
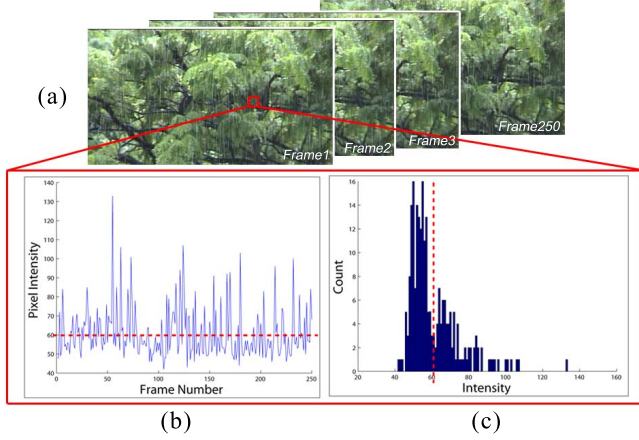
Fig. 1. Temporal property of rain. (a) Image frames of a video (b) Intensity of a pixel that is sometimes covered by rain. (c) Intensity histogram of the pixel exhibits two peaks, one for the background intensity distribution and the other for the rain intensity distribution.

tribution of rain drops, a pixel is not always covered by rain throughout the entire video (Fig. 1). Therefore, in a video of stationary scene taken by a stationary camera, the intensity histogram of a pixel that is sometimes covered by rain exhibits two peaks, one for the background intensity distribution and the other for the rain intensity distribution (Fig. 1(c)). On the other hand, the intensity histogram of a pixel that is never covered by rain throughout the entire video exhibits only one peak.

2.2. Chromatic Property

Garg and Nayar [3] showed that a spherical rain drop refracts a wide range of light. So the projection of rain drop in the image is much brighter than its background. Our further investigation shows that the increase in the intensities of R, G, and B channels is dependent on the background scene. Because of the difference in wavelength, blue light has a larger index of refraction and a wider field of view (FOV) than red light (Fig. 2). Therefore, a rain drop should refract more blue light coming from the background. Moreover, the amounts of change of R, G, and B channels, i.e., ΔR, ΔG, and ΔB, are related to the actual intensities of R, G, and B channels.
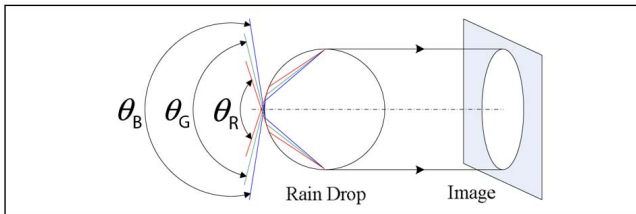
Fig. 2. Fields of Views of R, G, and B lights are different due to the difference in their refractive indices.
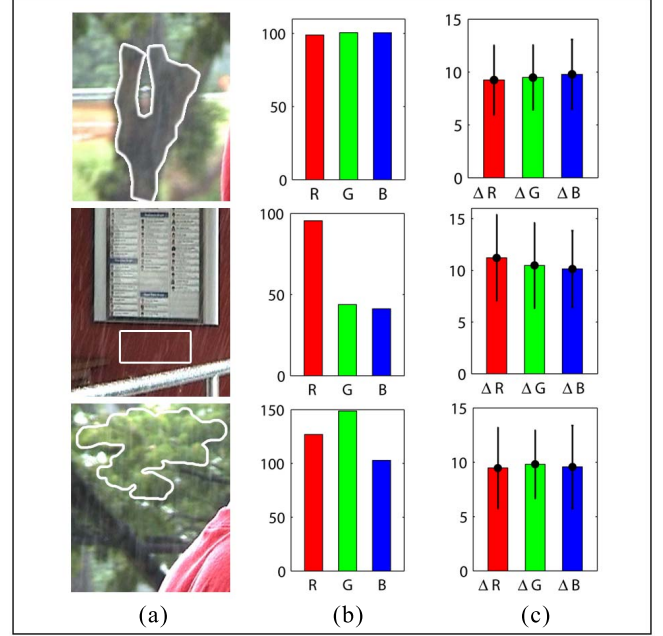
Fig. 3. (a) The regions selected for investigation. (b) The mean R, G, and B values of the pixels in the selected regions. (c) The corresponding means and standard deviations of ΔR, ΔG, and ΔB.

Fig. 3 illustrates three empirical examples. Fig. 3 (a) highlights regions in the video in which pixel colors are examined. Fig. 3 (b) shows the mean R, G, and B values of the pixels in the selected regions, and Fig. 3 (c) shows the corresponding means and standard deviations of ΔR, ΔG, and ΔB. These examples show that the mean ΔR, ΔG, and ΔB are indeed different and related to the mean R, G, and B intensities. However, the differences between the mean ΔR, ΔG, and ΔB are very small because the differences in refraction indices between the channels are very small. We found empirically that the FOVs of red, green, and blue lights are all around 165°, with very small differences between them. So, for ease of computation, we assume that ΔR, ΔG, and ΔB are roughly the same for pixels covered by rain drops.

3. RAIN DETECTION AND REMOVAL

3.1. K-means Clustering

From the temporal property discussed in Section 2.1, the intensity histogram of a pixel in a video taken by a stationary camera exhibits two peaks. K-means clustering algorithm can be used to identify the two peaks.

For each pixel in the image, its intensity over the entire video is collected to compute its intensity histogram. Then, K-means clustering with $K = 2$ is applied. The two initial cluster centers w_b for background and w_r for rain are initialized to be the smallest and the largest intensities of the his-

togram. The distance d between the intensity I of pixel p and cluster center w is computed as:

$$d(I_p, w) = |I_p - w|. \tag{1}$$

During K-means clustering, pixel p is distributed to the background cluster if $d(I_p, w_b) < d(I_p, w_r)$; otherwise, it is distributed to the rain cluster. After distributing the intensities of the pixels, the center of cluster C is updated as follows:

$$w(t+1) = \frac{1}{|C(t)|} \sum_{I_p \in C(t)} I_p. \tag{2}$$

K-means clustering is performed until converging to identify the clusters of background and rain intensities.

The above method is appropriate for videos of stationary scenes taken by a stationary camera. For videos of stationary scenes taken by a moving camera, the method can still be applied after performing video stabilization [6, 7] to remove camera motion. Video stabilization is performed by warping every video frame to align with the first frame. After removing rain, the stabilized video is destabilized by performing inverse warping to restore the original camera motion.

3.2. Chromatic Constraint

The K-means clustering method described in Section 3.1 is appropriate for videos of static scenes taken by a stationary or moving camera. When the scene contains moving subjects, the problem becomes more complex and K-means clustering is not sophisticated enough to detect rain correctly.

The discussion in Section 2.2 shows that the amounts of change ΔR, ΔG, and ΔB for pixels covered by rain are approximately the same. When ΔR, ΔG, and ΔB of a pixel between two successive image frames are significantly different, then the change is more likely due to the motion of object. Fig. 4 illustrates this observation. In the image frame in Fig. 4, region 1 contains a stationary background with rain and region 2 contains a part of a moving object. For region 1, the ΔR, ΔG, and ΔB of a pixel are approximately the same between two successive frames. On the other hand, for region 2, the ΔR, ΔG, and ΔB of a pixel are very different. So, false detection of rain can be reduced by using this chromatic constraint. That is, a candidate rain pixel detected by K-means clustering whose ΔR, ΔG, and ΔB are approximately the same (i.e., within a predefined threshold) is identified as an actual rain pixel. Otherwise, it is not an actual rain pixel.

An advantage of this method is that the chromatic constraint applies not only to rain in focus but also rain that is out of focus. The reason is that defocus is a weighted average of light around a pixel, which does not affect the amounts of change ΔR, ΔG, and ΔB. The limitation of the chromatic constraint is that it does not apply very well to gray regions. Gray regions have roughly the same R, G, and B values, and slight motion of gray regions results in very small ΔR, ΔG,

(a)   (b)   (c)

(d)   (e)
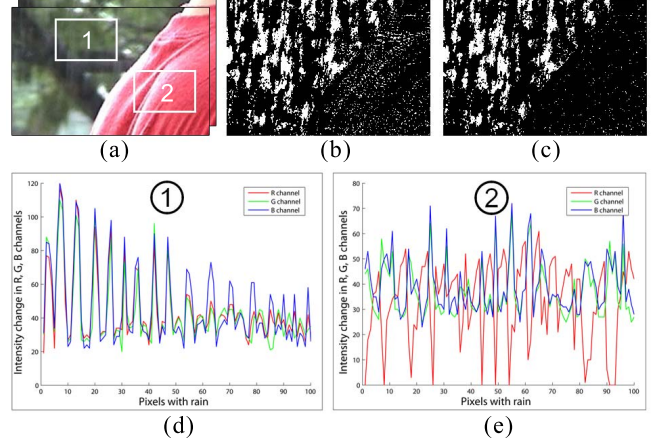
Fig. 4. Use of chromatic constraint. (a) Region 1 contains stationary scene with rain. Region 2 contains a moving object. (b) False detection of moving object as rain. (c) Reduced false detection of rain pixels. (d) The mean ΔR, ΔG, and ΔB of 100 randomly selected pixels in region 1 are approximately the same between two successive frames. (e) The mean ΔR, ΔG, and ΔB of 100 randomly selected pixels in region 2 are very different between two successive frames.

and ΔB that are approximately the same. So the chromatic constraint cannot distinguish between rain over gray regions and slight motion of gray regions.

3.3. Removal of Detected Rain Pixels

Removal of detected rain pixels can be achieved by replacing the colors of rain pixels with the corresponding background colors found by K-means clustering. Usually, most rain drops fall very fast and are out of focus. So rain streaks in the images are blurred by both motion and defocus. To improve the rain removal result, we applied dilation and Gaussian blurring on the detected rain pixels and use them as the alpha channel α to remove rain streaks by α-blending. That is, the new color C of a pixel is replaced by the α-blending of its rain-affected color C_r and background color C_b:

$$C = \alpha C_b + (1 - \alpha)C_r. \tag{3}$$

4. EXPERIMENTAL RESULTS

In our experiments, we used Sony DCR-TRV15E camera to take some videos with various raining scenes, including light, moderate and heavy rain in static and dynamic situations. Moreover, for comparison with the method in [3], we also tested our algorithm on the movie clips presented in [3]. Fig. 5 shows rain removal in static scene. Almost all the rain streaks are removed perfectly. The enlarged views show that both rain streaks in-focus and out-of-focus are removed completely.

Fig. 6 illustrates the detection and removal results of two frames in dynamic scenes. The results show that our algo-

Fig. 5. Rain removal in static scene. (a) A frame in the original video. (b) The derained result.
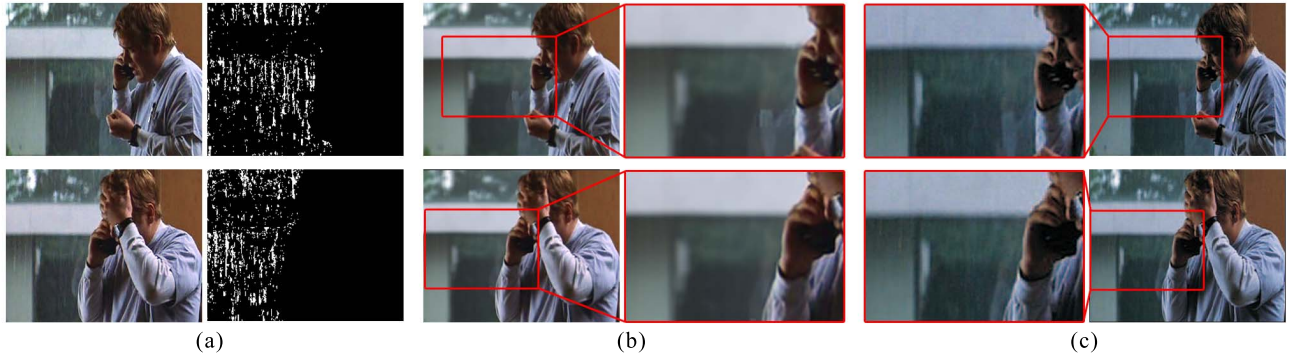
Fig. 6. Detection and removal of rain in dynamic scenes. The video is from the movie "Magnolia" and used by Garg and Nayar in [3]. (a) Original frames and rain detection results. (b) Our rain removal results. (c) Garg and Nayar's results.

rithm can effectively distinguish rain drops and moving human body. Furthermore, comparison with the results of Garg and Nayar [3] shows that our method detects the background colors more accurately and gives a rain removed video with better visual quality. A video demonstration of our experimental results can be downloaded from
http://www.comp.nus.edu.sg/ photo/projects/rain.html.

5. CONCLUSIONS

By careful studying of rain in video, we identified two important properties that characterize rain. 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. By using both properties, a new rain removal algorithm is developed which can detect and remove rain streaks in both stationary and dynamic scenes taken by stationary cameras. To handle videos taken by moving cameras, the videos can be stabilized for rain removal, and destabilized to restore camera motion after rain removal. It can handle both light rain and heavy rain conditions, as well as rain in focus and rain that is out of focus. Experimental results show that the algorithm performs better than existing methods.

6. REFERENCES

[1] K.A. Patwardhan and G. Sapiro, "Projection based image and video inpainting using wavelets," in *Proc. ICIP 2003.*, vol. 1, pp. 857–860.

[2] K.A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," in *Proc. ICIP 2005.*, vol. 2, pp. 69–72.

[3] K. Garg and S.K. Nayar, "Detection and removal of rain from videos," in *Proc. CVPR 2004*, vol. 1, pp. 528–535.

[4] S.G. Narasimhan and S.K. Nayar, "Vision and the atmosphere," *IJCV 2002*, vol. 48, no. 3.

[5] K. Garg and S.K. Nayar, "When does a camera see rain?," in *Proc. ICCV 2005.*, vol. 2, pp. 1067–1074.

[6] Y. Matsushita, E. Ofek, X. Tang, and H.Y. Shum, "Full-frame video stabilization," in *Proc. CVPR 2005.*, vol. 1, pp. 50–57.

[7] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proc. ECCV 1992*, pp. 237–252.