A Rain Removal Method Using Chromatic Property for Image Sequence*

Liu Peng Xu Jing Liu Jiafeng Tang Xianglong Zhao Wei

College of Computer Science and Technology, Harbin Institute of Technology

Abstract

Raindrops degrade the performance of outdoor vision system, and bring difficulties for objects detection and analysis in image sequence. In this paper, we propose an algorithm for raindrop removal using chromatic based properties in order to improve the data quality and vision effect of image sequence. The raindrops detection method, considering the chromatic properties of image sequence, is induced, which is not affected by the velocity and time information of raindrops. Therefore, this function is suitable for all the blur effects caused by raindrops. Moreover the algorithm is effectual both for removing raindrops at background and foreground. The experiment results show that the proposal algorithm is able to remove the raindrops and improve the quality of image sequence remarkable.

Keywords: imaging model, complex scenes, video processing, outdoor vision and weather

*This work was supported by National Natural Science Foundation of China(60702032), Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2008.63), and China Academy of Space Technology Innovation Foundation (CAST200814)

1. Introduction

The image sequences of outdoor vision systems are affected by bad weather conditions and bring difficulties in computer vision. The degraded images also bring difficulties in areas such as object detection, tracking, segmentation, video surveillance, and so on. Some methods have been used to solve these difficulties. The weather conditions are classified into two types [1]. For steady weather such as fog and haze, the droplets are too small to be detected by a camera or by naked eyes. Many models are used to solve this problem [2, 3, 4]. For dynamic weather such as rain and snow, the different shapes and movements of the particles make the problem more complicated [1, 5, 6]. Due to the random distribution and complex performance of rain streaks, the classical image denoising algorithms are not suitable for restoration of rain-affected image. The process of finding an area to replace the rain-affected area is more like image inpainting [7, 8, 9]. Image inpainting should use two properties: (1) the occluders. (2) the information in the regions surrounding the areas to be inpainted. Since the rain-affected image is degraded seriously, and it is a difficult problem to detected rain-affected area, the image inpaintin algorithms are not directly used to remove raindrops.

Garg and Nayar [1] analyze the physical properties of raindrops. They supposed that rain-affected pixels appear similar intensity change in three consecutive frames. But when rain is heavier or raindrops are very close to the camera, the pixels in the same position in two consecutive frames are often both affected by rain, as shown in Fig. 1. So it will omit some rain-affected pixels when using a threshold to detect raindrops in three consecutive frames [1]. Zhang and Li [6] used the chromatic constraint to detect rain-affected area. They found when a pixel was covered by raindrops, the varieties of the intensities of R,G, and B , were approximately the same. But in some videos, the varieties of the intensities of R,G, and B , are distinctly different in rain-affected pixels. Otherwise, the varieties of three channel are almost the same in area belonging to a moving object. So it is difficult to find an appropriate threshold that is suitable for both stationary and dynamic objects.

This paper focuses on rain removal in video. We mainly focus on scenario comprising a stationary background and some moving objects captured by a stationary camera. Through a further study of the raindrop's model, a general detecting function is proposed using chromatic property. By using this method, we can distinguish rain-affected pixels from areas comprising moving objects. Therefore the rain-affected area is detectable. Then we give a discriminant function to reduce the improper detection so that image inpainting algorithms are useful. The result shows that our method is effective and has better performance.

2. Photometry Model of Rain

The falling raindrops generally are modeled as an oblate spheroid shape or a spherical shape. When A pixel is covered by a raindrop, its intensity is much brighter then its background [1].Due to the rapidly speed, the raindrop can hardly be captured clearly by a camera with normal exposure time. It appears as a greatly blurred rain streak. Garg and Nayar [1] show us the photometric model to illustrate the motion-blur effects.

$$I_r = \int_0^{\tau} I_E dt + \int_{\tau}^{T} E_b dt \qquad (1)$$

where T is the exposure time of an ordinary camera. τ is the time during which the pixel is covered by the raindrop. I_r is the intensity of this pixel affected by the raindrop. I_b is the background intensity of the pixel at the same position. I_E is the time-averaged irradiance caused by the raindrop during the time τ, E_b is the average irradiance of the background. If the background is stationary, or the motion of it is slow, we are able to use the average irradiance value E_b to calculate the background irradiance of the pixel over the time duration T. Let $E_b = \frac{I_b}{T}$, ΔI is the change of intensity at a pixel due to a raindrop. Substituting $\alpha = \frac{\tau}{T}$,we can obtain

$$I_r = \alpha I_E + (1 - \alpha) I_b \qquad (2)$$

$$\Delta I = \alpha I_E - \alpha I_b \qquad (3)$$

Garg and Nayar[1] use a threshold in image sequences to get candidate rain-affected pixels. Garg uses an assumption that all pixels along a rain streak appears as a linear relation with the background intensity I_b , using the equation (3), the slope of which is α ,

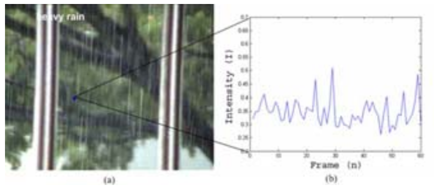

Fig. 1: Rain affected frame in video. (a) a rain-affected frame with stationary background. (b) the intensities change of the position in (a)

$\alpha \in [0 - 0.039]$. But sometimes this assumption is not valid. Fig. 1 shows the intensity at a pixel in image sequence. During heavy rain, pixels at the same position in consecutive frames are often affected by different raindrops. Therefore we cannot get the accurate background intensity I_b. In the next part, we further develop a detecting function using chromatic property, which is effective when pixels in consecutive frames are covered by raindrops.

3. Detecting Function Using Chromatic Property

3.1. Relation of Background pixel and rain-affected pixel

Consider the pixels at the same poison in two consecutive frames. One is background pixel. The other is rain-affected pixel. Generally when a raindrop is falling down, either close to the camera, or far from the camera, the pixels along a rain streak appear as complicate intensity changes along the falling direction. So the range of its slope is not always useful. But E_r and τ at each pixel are uniform respectively in R, G, and B channels. So we use $\Delta\vec{I}$, $\vec{I}_E$, and $\vec{I}_b$ to indicate ΔI,I_E, and I_b in three channels respectively, we get

$$\Delta\vec{I} = \alpha\vec{I}_E - \alpha\vec{I}_b \tag{4}$$

There are two variables in three equations. So for every pixel covered by raindrops, E and α calculated using two equations are suitable for the last equation. This can be used to detect rain-affected pixels on the background.

3.2. Relation of Two Rain Affected Pixels

If rain is heavy, or raindrops are very close to the camera, pixels at the same position can be probably covered by two different raindrops in consecutive frames. In this condition, $\vec{I}_r$ is the intensity of the brighter pixel. $\vec{I}'_r$ is the intensity of the other pixel. $\vec{I}_b$ is the intensity of background pixel.$\vec{I}_E$ and α are the two variables suitable for the brighter pixel. $\vec{I}'_E$ and α' are the two variables suitable for the other pixel. So according to equation (2), we can obtain

$$\vec{I}_r = \alpha\vec{I}_E - (1-\alpha)\vec{I}_b \tag{5}$$

and

$$\vec{I}'_r = \alpha'\vec{I}'_E - (1-\alpha')\vec{I}_b \tag{6}$$

From equation (5) and equation (6), we can obtain

$$\begin{aligned} \Delta\vec{I} \quad &= \quad \alpha\vec{I}_E - \frac{1-\alpha}{1-\alpha'}\alpha'\vec{I}'_E \\ &\quad - \frac{\alpha-\alpha'}{1-\alpha'}\vec{I}'_r \end{aligned} \tag{7}$$

So given two pixels, using two equations in Eq.(7), we will calculate ΔI_E and $\Delta\alpha$. If the left side of the third equation is equal to its right side using ΔI_E and $\Delta\alpha$ calculated before, those pixels are affected by rain. If the equation is not valid, those pixels belong to a moving object. Therefore the third equation is the detecting function. There is an exception: when $I_R = I_G$,and $I'_{bR} = I'_{bG}$, there are infinite solutions of ΔI_E and $\Delta\alpha$. In this

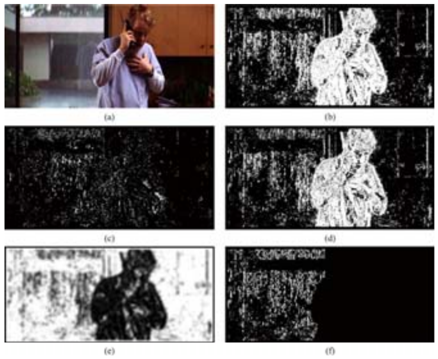

Fig. 2: Rain detection. (a) the original frame in video. (b) Different intensities at two consecutive frames. (c) Rain-affected pixels. (d) Moving object pixels. (e) Estimate values of E_u. (f) Candidate rain pixels

case, the pixels belonging to a moving object will be regarded as rain-affected pixels. We will solve this improperly detection in the next part.

4. Detection and Removal of Rain

4.1. Discriminant Function

Given a video captured in the rain (shown in Fig. 2), we apply the detecting function using chromatic property at two consecutive frames. Rain-affected pixels and moving-object pixels are recognized in Fig. 2 (c) and (d). Due to the blur caused by rain, some rain-affected pixels are regarded as moving-object pixels. And in some areas belonging to moving object with similar color, some pixels are regard as rain-affected pixels (the mistaken problem mentioned in section 3). Fig. 2 shows in rain-affected area, there are both rain-affected pixels and moving objects pixels. Moreover there are also lots of unaffected pixels because rain will never always cover one position throughout the whole video. In area of moving objects, most detected pixels are moving objects pixels. In area affected by movement of camera, there are little rain-affected pixels.

So by estimating the density of pixels in the selected area, rain-affected areas are distinguished from moving object pixels and camera affected pixels. This is effective to reduce the improperly detected. The detecting function is

$$\begin{cases} E_r = \dfrac{\int_\Omega r(x)dx}{\int_\Omega f(x)dx} > \delta_1 \\ E_m = \dfrac{\int_\Omega m(x)dx}{\int_\Omega f(x)dx} < \delta_2 \\ E_u = \dfrac{\int_\Omega (f(x)-r(x)-m(x))dx}{\int_\Omega f(x)dx} < \delta_3 \end{cases} \tag{8}$$

where Ω is the area to be recognized, $f(x)$ calculates the total number of pixels in Ω , if $x \in \Omega$, $f(x) = 1$. $r(x)$ calculates the total number of rain-affected pixels in Ω , if x meets the detecting function, $r(x) = 1$. $m(x)$ calculates the moving object pixels, if x doesn't meet the detecting function, $m(x) = 1$. In fact the equation E_u in (13) calculates the number of unaffected pixels in the area Ω. The estimate value of E_r shows the density of rain. $\delta_1, \delta_2, and \delta_3$ are estimated according to video information.

4.2. Removal of Rain

Using the detecting methods,rain-affected areas belonging to stationary background are distinguished from moving objects although the moving objects maybe in the rain or in room. So we only consider the rain-affected pixels on the stationary background. The rain-affected pixels are much brighter than background pixels. For every two related rain-affected pixels, we use the smaller value to replace the larger one. To estimate the background value I_b of rain-affected pixels, we use a neighborhood on some consecutive frames. The number of frames is adjustable according to image degradation. Only those pixels meeting the detecting function are considered.

$$I_b = \inf\{x \mid x \in (\Omega' \setminus M)\} \quad (9)$$

where Ω' is the searching space, M is the space of moving pixels in Ω'.When rain is heavy and more frames are affected by raindrops, this removal algorithm is also effective by increasing the number of searching frames.

5. Experimental Results

Our experiments use a threshold of 3 gray levels to detect the intensity change of pixels. And use 11×11 neighborhoods to calculate values of the discriminant function. The removal step uses 3×3 neighborhoods to search the more suitable background pixels in ten consecutive frames.

Fig. 3 (a) is an image of static scene from the video captured by Zhang and Li [6]. (b) is the removal result using Garg and Nayar's method[1]. The result shows that the method of Garg and Nayar is not effective when rain is heavy because their removal algorithm only calculates three consecutive frames. (c) is the result using Zhang and Li's method calculating thirty frames [6]. Both (c) and (e) have better quality in static scene. But (c) uses K-means clustering to calculate the background color which consumes more time. Our result, shown in (d) calculates ten consecutive frames, calculates thirty frames in (e). However the time consuming by our method is

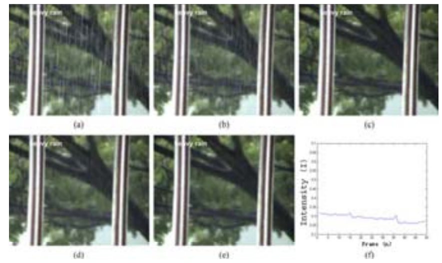

Fig. 3: One frame in static scene. (a) The original frame. (b) Result using Garg's method calculating three frames. (c) Result using Zhang's method calculating 200 frames. (d) Result using our method calculating 10 frames. (e) Result of our method using 50 frames. (f) The intensities change of the de-rained frames on the position in (a), calculating 10 frames

far less than that of zhang's. (f) is the intensity change at the position of the pixel in (a) in derained video with ten consecutive searching frames. Fig. 1 and Fig. 3 are from the same video. Fig. 3 (f) shows our algorithm is effective compared with Fig. 1 (b). Fig. 4 shows results in dynamic scenes. Fig. 4 (b) and (c) are result of Zhang's method. The improperly detected pixels obviously damage the visual quality of derained image. Our results show a better performance in this conditon.

6. Conclusions

By further studying the model of raindrops, we obtain a detecting function using chromatic property which is suitable for a general condition. It can distinguish rain-affected pixels between two arbitrary frames. Then we develop a discriminant function using the density of detected pixels to reduce the improperly recognized pixels. The removal method makes the pixels at the

Fig. 4: Results in dynamic scenes. The video is a chip from the movie "Magnolia". It has been used by Garg and Zhang. (a) Original frame. (b) Result of Zhang's method. (c) Local area of (b). (d) Our result calculating 10 frames. (e) ,(f) Local areas of (d)

same position of backgrounds in each frame similar and shows a better visual quality. Our method does not use any information about the shape, the velocity of raindrops, neither uses the value of camera's exposure time. Therefore it is effective in various rain conditions.

7. Acknowledgment

This work is financially supported by National Natural Science Foundation of China(60702032), Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2008.63), and China Academy of Space Technology Innovation Foundation (CAST200814).

References

[1] K. Garg and S. K. Nayar, Vision and Rain, *International Journal of Computer Vision*, 75:3-27, 2007.

[2] L. Mulvin and J. Lewis, Architectural detailing, weathering and stone decay, *Building and Environment*, 29:133-138, 1994.

[3] S. G. Narasimhan and S. K. Nayar, Contrast restoration of weather degraded images, *IEEE Trans. on PAMI*, 25(6):713- 724, 2003.

[4] Y. Yitzhaky, I. Dror, N. S. Kopeika, Restoration of atmospherically blurred images according to weather-predicted atmospheric modulation transfer functions, *Optical Engineering*, 36(17):3064-3072, 1997.

[5] R. J. Kubesh and K. V. Beard, Laboratory measurements of spontaneous oscillations of moderate size raindrops, *Journal of the Atmospheric Sciences*, 50:1089-1098, 1993.

[6] ,X. Zhang,H. Li, et al, Rain Removal in Video by Combining Temporal and Chromatic Properties, *Proc of 2006 IEEE International Conference on Multimedia and Expo*, pp.461-464, 2006.

[7] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, Navierstokes, fluid dynamics, and image and video inpainting, *Proc.of IEEE Computer Vision and Pattern Recognition*, pp.355-362, 2001.

[8] Y. Wexler, E. Shechtman, and M. Irani, Space-time video completion, *Proc.of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.120-127, 2004.

[9] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, Video inpainting of occluding and occluded objects, *IEEE International Conference on Image Processing*, 2:69-72, 2005.