A Novel Approach for Rain Pixel Detection and Recovery by using Motion Segmentation

Bhagyashri More
Scholar, ME(Computer)
Savitribai Phule Pune University
SITRC, Nashik

A. D. Potgantwar
Assistant Professor
Savitribai Phule Pune University
SITRC, Nashik

ABSTRACT
The visual effects of rain are complex. Rain removal is a very useful technique in applications such as security surveillance, video editing, vision based navigation, and video indexing/retrieval. Rain produces sharp intensity variation in images and videos, which put down the performance of outdoor vision systems. These intensity variations depend on different parameters, such as the camera properties, the parameter of rain, and the brightness and sharpness of the scene. Removal of rain stripes in video is a hard task due to the random spatial distribution and fast motion of rain. Photometric, chromatic, and probabilistic properties of the rain have been exploited to detect and remove the rainy effect. This paper is introducing the Rain Pixel Algorithm with better performance for rainy scenes with large motion than exiting algorithm.

Keywords
Motion segmentation, Motion buffering, Motion exclusion, Rain removal.

1. INTRODUCTION
In rainy videos pixels exhibit small but frequent intensity variations, and this variation could be caused by several other reasons besides rain fall, namely, global illumination change, camera move, and object motion etc. To remove the rainy effect, it is necessary to detect the variations that are caused by rain, and then replace them with their original value [1].

The dynamic bad weather model is evaluated, for the purpose of restoration. Rain is the major component of the dynamic bad weather. Due to the high velocity of the raindrops, their position projection forms the rain streaks. Removal of rain streaks in video is a challenging problem due to the random spatial distribution and fast motion of rain. In rainy videos pixels exhibit small but frequent intensity variations, and this variation could be caused by several other reasons besides rain fall, namely, global illumination change, camera move, and object motion etc. Weather conditions vary widely in their physical properties. And in the visual effects they produce in images. Based on their differences, weather conditions can be broadly classified as steady (fog, mist and haze) or dynamic (rain, snow and hail) [4]. Concentration on the problem of rain. Rain consists of a distribution of a large number of drops of various sizes, falling at high velocities. Each drop behaves like a transparent sphere, refracting and reflecting light from the environment towards the camera. An ensemble of such drops falling at high velocities results in time varying intensity fluctuations in images and videos. In addition, due to the finite vulnerability time of the camera, intensities due to rain are motion blurred and Therefore depend on the background. Thus, the visual manifestations of rain are a combined effect of the dynamics of rain and the photometry of the environment [2]. The Rain Pixel Removal algorithm is supported on motion segmentation of dynamic scene. Initial apply photometric and chromatic constraints for rain detection then rain removal filters are applied on pixels such that their dynamic property as well as motion occlusion clue is considered; both spatial and temporal information are then adaptively exploited during rain pixel recovery. Survey result show that this algorithm performs better output as compare to existing ones in highly dynamic scenarios [1].

2. RELATED WORK
Rain removal is a very useful and important technique in Applications such as security surveillance, video/movie editing, Vision based navigation, and video indexing/retrieval. Wide range of algorithms using various types of techniques is been Used by various authors.

2.1 Photometric and Dynamic Model
Kshitiz Garg and Shree K. Nayar in 2007 present the first complete analysis of the visual effects of rain on an imaging system and the different factors that affect it. To handle photometric rendering of rain in computer graphics and rain in computer vision they develop systematic algorithm. They first develop a photometric model that describes the intensities produced by individual rain streaks and then develop a dynamic model that captures the spatiotemporal properties of rain. Together, these models describe the complete visual appearance of rain. Using this model they develop a new algorithm for rain detection and removal. By modeling the scattering and chromatic effects, Narasimhan and Nayar successfully recovered “clear day” scenes from images taken in bad weather. But, their assumptions such as the uniform velocities and directions of the rain drops limited its performance [2].

2.2 Temporal and Chromatic Properties
By using both temporal and chromatic properties of rain Xiaopeng Zhang, Hao Li presents a K-mean clustering algorithm for rain detection and removal. 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. The algorithm can detect and remove rain streaks in both stationary and dynamic scenes, by using both temporal and chromatic properties which are taken by stationary cameras. But it gives wrong result for those scenes which are taken by moving cameras. To handle these situations 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. This method is only applicable with static background, and it gives out false result for particular foreground colors [3].

2.3 Probabilistic Model

K. Tripathi and S. Mukhopadhyay proposed a efficient, simple, and probabilistic model based rain removal algorithm. This algorithm is better to the rain intensity fluctuations. Probabilistic approach automatically adjusts the threshold and effectively differentiates the rain pixels and non-rain moving object pixels. Discrimination is done between the rain and non-rain moving objects by using the time evolution of pixels in consecutive frames. This algorithm does not assume the shape, size and velocity of the raindrops and intensity of rain, which makes it robust to different rain conditions. Advantage of this algorithm is that it automates the algorithm and reduces the user intervention. Here, it is assumed that the video capturing camera is static. There is a significant difference in time evolution between the rain and non-rain pixels in videos. This difference is analyzed with the help of the skewness and Pitman test for symmetry. This method is more robust dealing with dynamic scenes; however some statistical feature it proposes works poorly in many occasions, and it gives a lot of false detections [5].

3. METHODOLOGY

Rain Pixel Removal algorithm is based on motion segmentation of dynamic scene. The pixel intensity variance of a rainy scene is caused by rain and object motion. The variation caused by rain need to be removed, and the ones caused by object motion need to keep it as it is. Thus motion field segmentation naturally becomes a fundamental procedure of this algorithm. Proper threshold value is set to detect the intensity variation caused by rain. After applying photometric and chromatic constraints for rain detection, rain removal filters are applied on pixels such that their dynamic property as well as motion occlusion clue are considered; both spatial and temporal information are then adaptively use during rain pixel recovery. This algorithm gives better performance over others for rain removal in highly dynamic scenes with heavier rainfall. Figure 1 shows the block diagram of rain removal pixel using motion segmentation.

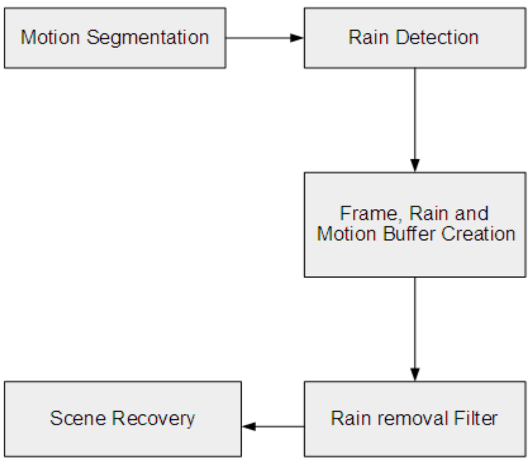

Figure 1: Block diagram for Rain Pixel removal algorithm

3.1 Motion Segmentation

The pixel intensity fluctuation of a rainy scene is caused by rain and object motion. The variations caused by rain need to be removed, and the ones caused by object motion need to be retained. Thus motion field segmentation becomes a fundamental procedure of this algorithm.

A. Evaluation of Motion Field

Motion field is a 2-D vector field projected from the 3-D velocity field of a dynamic scene. Independent moving objects could be detected using motion segmentation. Optical flow is used to evaluate the existence of motion. With the constraint of intensity conservation, apply the chain rule for differentiation [7],

$$(\partial I/\partial x)(dx/dt) + (\partial I/\partial y)(dy/dt) + dI/dt = 0 \qquad \ldots (1)$$

Where I (x, y, t) is the image brightness at pixel P(x, y) at time t. A binary motion pixel map $I_m(x, y)$ is retrieved based on the threshold optical flow field.

$$I_m(x, y) = \begin{cases} 0 & f(x,y) \geq \varrho m \\ 1 & f(x,y) < \varrho m \end{cases} \qquad \ldots (2)$$

Here m is the mean of the motion field ϱ is a parameter set according to how bad the rainy effect is. Gaussian Mixture Model is used (GMM) to simulate the distribution of $I_m(x, y)$. Considering the similarity between adjacent video frames, GMM parameters from the current frame can be used as initial parameters for EM iteration of the next frame, which will make the algorithm converge very fast and highly efficient [6].

B. Evaluation of Local Properties

In an effort to include the local properties (pixel location, and chromatic values) into the segmentation, a feature vector is formed consisting of each pixel's spatial and color information.

$$F_{ij} = (R_{ij}, G_{ij}, B_{ij}, \omega{*}i, \omega{*}j) \qquad \ldots(3)$$

In above Equation (3) *Rij* , *Gij* , *Bij* are the intensity values of the R, G, B channel at P(i, j), ω Is the weighting factor between the color and position space. For different frame size and scene complexity, ω should be adjusted as different values.

3.2 Rain Detection

First, grey scale intensity differences between two successive frames are calculated and threshold. The threshold values are set such that all the intensity fluctuations caused by rain can be detected.

$$I_{diff} = \begin{cases} 1 & I_N - I_{N-1} \geq Dth \\ 0 & I_N - I_{N-1} < Dth \end{cases} \qquad \ldots(4)$$

A binary difference map I_{diff} is calculated using above Equation (4).

After applying the photometric and chromatic constraints, the pixels in I_{diff} that fail the constraints are excluded from the final rain mask I_{rain} [9].

$$I_{rain} = I_{diff} - I_{fail} \qquad \ldots(5)$$

In motion exclusion rain pixels within the motion object and the background need to be treated separately, so here *IRain* is divided into two sets: rain candidate pixels in the motion target area will comprise the set *Sm*; rain candidate pixels in the background area will comprise the set *Sb*. Finally, the pixels that are not included in *Sm* or *Sb* form the set *Sp*, which are the pixels that are not covered by rain.

3.3 Frame, Rain and Motion Buffer

Three buffers are created for the rain removal: video frame buffer B_I (len, wid, stk), rain buffer B_R(len, wid, stk) and motion buffer B_M(len,wid, stk). Here len × wid is the video frame size, stk is the depth of the buffer, and it is set as stk = 9 in the experiment for a better recover performance.

Each layer of B_I comprises one video frame. The newest frame is pushed on top of the buffer (buffer layer 1), and the oldest frame is moved out from the bottom (buffer layer stk), and the rest of the layers update accordingly. The rain buffer B_R (len, ωid, stk) records the binary rain map I_{Rain} for each corresponding video frame in B_I, and motion buffer B_M (len,ωid, stk) records the corresponding binary motion decision map I_m. Both B_R, B_M update in sync with the video frame buffer BI [10].

3.4 Scene Recovery

A 88-spatial-temporal neighborhood V is designed for rain removal. Lie in the center of the neighborhood system is the pixel BI (x, y, n). V consists 8 pixels along the time axis in BI with location BI (x, y, (n−4) ~ (n+4)), and 80 pixels in the current frame with area BI ((x −4) ~ (x +4), (y−4) ~ (y +4), n). Similar neighborhood set-up could be found in [10], where they use a 26-connected 3×3 neighborhood. The temporal neighborhood is set to be 8 in our method for a better recovering visual result, and for a better resilience against motion occlusion. The spacial neighborhood is set as 80, for the fact that rain streak breadth is usually around 3 to 8 pixels [8].

For scene recovery first recover pixels that are covered by rain in static scene background then recover rain affected pixels in motion object. Finally, for the pixels that are not covered by rain (pixels in set *Sp*), this algorithm simply keep their values.

4. PERFORMANCE ANALYSIS

Performances were carried out on videos of highly dynamic rainy scenes. As per the Survey result, this algorithm are able to remove rain streaks in videos. The moving objects are not blurred by the rain removal algorithm in spite of its large motion, and no leaving trails (ghost effect) are observable. When we use rain removal algorithm then this algorithm is effective for scenes with complex motions and at the same time is insensitive to time-varying textures that have temporal frequencies similar to those due to rain [2].

Existing algorithms for rain removal performs poorly in highly dynamic scene. Based on the motion segmentation scheme which is defined in this paper it recovers the rain pixels such occlusion clue is considered; both spatial and temporal information are adaptively exploited during rain pixel recovery. Performance Analysis shows that this algorithm defines better results as compare to existing algorithms in highly dynamic scenario. Many research issues have been highlighted and directions for future work have been suggested [1].

5. CONCLUSION

This paper presents a Rain Pixel Removal algorithm to recover the rain affected pixels by using motion segmentation. Existing algorithms for rain removal gives poor performance for highly dynamic scene. Fundamental part of this given algorithm is motion segmentation. it recovers the rain pixels such that each pixel's dynamic property and motion occlusion clue is considered; spatial and temporal information are adaptively used during rain pixel recovery. Performance analysis shows that this algorithm gives more recovery of rain affected pixels as compared to existing ones. Many research issues have been highlighted and give direction for future work. Quality of recovered image will be improved as well as we can remove noise.

6. ACKNOWLEDGMENT

I would like to express my special appreciation and thanks to my adviser Prof. A. D. Potgantwar, you have been a tremendous mentor for me.

7. REFERENCES

[1] Jie Chen and Lap-Pui Chau, ‘‘ A Rain Pixel Recovery Algorithm for Videos with Highly Dynamic Scenes”, IEEE ImageProcessing, vol.23, no. 3, March 2014.

[2] K. Garg and S. K. Nayar, “Detection and removal of rain from videos, ” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 1,Jul. 2004, pp. 528–535.

[3] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, “Rain removal in video by combining temporal and chro matic properties, ” in Proc. IEEE Int. Conf. Multimedia Expo, Jul. 2006, pp. 461–464.

[4] K. Garg and S. K. Nayar, “Vision and rain, ” Int. J. Compute Vis., vol. 75, no.1, pp. 3–27, 2007.

[5] A. K. Tripathi and S. Mukhopad hyay, “A probabilis tic approach for Detection and removal of rain from videos,” IETE J. Res., vol. 57, no.1, p. 82–91, Mar. 2011

[6] A. Ogale, C. Fermuller, and Y. Aloimonos, “Motion segmentation using occlusions,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 988–992, Jun. 2005.

[7] J. P. Koh, “Automatic segmentation using multiple cues classification,” M.S. dissertation, School Electr. Electron. Eng., Nanyang Technol. University, Singapore, 2003.

[8] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer-Verlag, 2006, pp. 423–455, Ch. 9.

[9] R. J. Kubesh and K. V. Beard, “Laboratory measurements of spontaneous oscillations for moderate-size raindrops,” J. Atmos. Sci., vol. 50, no. 1, pp. 1089–1098, 1993.

[10] H. Stark and J. W. Woods, Probability and Random Process with Applications to Signal Processing, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002, pp. 589–601.

[11] B. Heisele, U. Kressel, and W. Ritter, “Tracking non-rigid, moving objects based on color cluster flow,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 1997, pp. 257–260.