

# A Robust Algorithm for Rain Pixel Recovery in Highly Dynamic Videos

Sheeba Rani. A. S<sup>1</sup>, Preetha V.H<sup>2</sup>P.G Student, Department of ECE, SCT College of Engineering, Trivandrum, Kerala, India<sup>1</sup>Assistant Professor, Department of ECE, SCT College of Engineering, Trivandrum, Kerala, India<sup>2</sup>

**ABSTRACT:** In this study, an efficient and simple algorithm for detection and removal of rain from video is proposed. Rain removal is an important technique which is commonly used in application such as traffic surveillance, movie and video editing and vision based navigation. Previous methods generally work well with slightly dynamic videos and light rain. The proposed method gives better results for rainy videos with highly dynamic background. This method is based on motion segmentation of dynamic scene. Then photometric and chromatic constraints are applied for rain detection. Rain removal filters are then designed separately for rainy pixels in the foreground motion areas and background stationary areas.

**KEYWORDS:** Motion segmentation, motion occlusion, dynamic scene, rain removal.

## I. INTRODUCTION

Rain produces intensity fluctuations in images and videos. These fluctuations are caused by factors other than rain such as movement of camera, scene brightness, object motion etc. To remove rainy effect, it is necessary to detect the rainy pixels and then replace them with their original value. There are many algorithms proposed for rain pixel recovery[1]. Rain can be considered as a collection of spherical droplets which are moving at high velocities. Each drop refracts and reflects the environment which produces sharp intensity fluctuations in images. These high velocity raindrops create complex time varying signals in images and videos. The proposed method has three main steps, motion segmentation, rain detection, and scene recovery

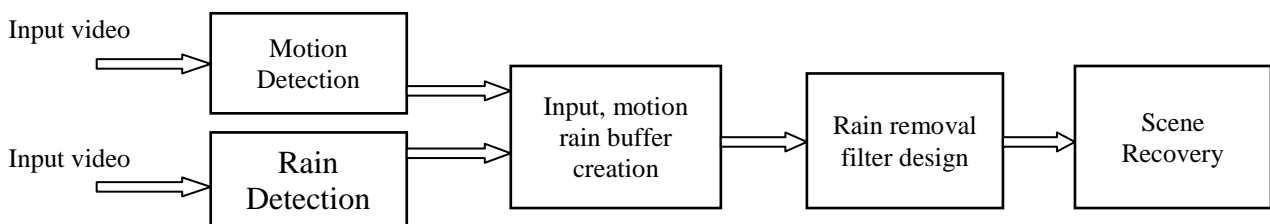


Fig.1. Block diagram of the proposed method.

## II. RELATED WORK

There are many algorithms exist for rain pixel removal and recovery. KshitizGarg and Shree K. Nayar in their work presented a detailed analysis of visual effects of rain in an imaging system [2], [9]. They combined the dynamic as well as physical properties of rain to develop efficient algorithms for detection and removal of rainy pixels from videos. Each rain drop refracts and reflects light from the environment towards the camera. Such high velocity drops results in intensity fluctuations in videos. Also due to the finite exposure time of camera, intensity due to rain are motion blurred and hence depend on the background. Thus the effect of rainy pixels on video depends on the dynamics of rain and the photometry of the environment.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

Xiaopeng Zhang proposed a method for rain removal and recovery by combining both temporal and chromatic properties of rain [9]. The temporal property says that no pixel is always covered by rain throughout the video. The chromatic property indicates that the changes of R, G and B values of rain affected pixels are approximately the same. The algorithm works based on these two assumptions.

Tripathi proposed a spatial temporal model to classify the rain affected pixels. This method is robust in dealing with dynamic videos.

### III. MOTION SEGMENTATION

Rain as well as moving object can cause pixel intensity fluctuations in rainy videos. We have to remove the intensity fluctuation due to rain and retain that caused by moving objects. Hence motion segmentation is the fundamental procedure of this method.

#### A. Estimation of Motion Field

Motion Field is the 2D vector field which is the projection of 3D velocity field of a moving scene on the image plane [3]. Optical flow is used to determine the existence of motion. Motion field can be derived from first order deviation of image brightness pattern.

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0, \quad (1)$$

where  $I(x, y, t)$  is the image brightness at pixel  $p(x, y)$  at time  $t$ . Let  $u = \frac{dx}{dt}$ ,  $v = \frac{dy}{dt}$ ,  $E_x = \frac{\partial I}{\partial x}$ ,  $E_y = \frac{\partial I}{\partial y}$ ,  $E_t = \frac{\partial I}{\partial t}$ , this equation can be rewritten as

$$E_x u + E_y v + E_t = 0, \quad (2)$$

Here  $u$  and  $v$  are optical flow velocities. Horn's method [4] is used for the estimation of these optical flow velocities. Optical flow is used to determine the relative displacement between two adjacent frames for many moving objects. But high velocity rain pixels are not considered in motion field by setting a pre-set threshold value  $\rho$  of  $f(x, y)$ . Thus a binary motion pixel map is created based on thresholded optical flow field. It is given by

$$I_{m(x, y)} = \begin{cases} 1, & f(x, y) < \rho m \\ 0, & f(x, y) \geq \rho m \end{cases}, \quad (3)$$

$$f(x, y) = \sqrt{u(x, y)^2 + v(x, y)^2}, \quad (4)$$

$$m = \frac{1}{MN} \sum_x \sum_y f(x, y), \quad (5)$$

Here  $f(x, y)$  is the pixel velocity,  $m$  is the mean of motion field, and  $\rho$  is a parameter set according to how badly the rainy effect is. Typically the value of  $\rho$  is 0.1.

A drawback of optical flow is that it cannot detect small changes in intensity. Therefore a Gaussian Mixture Model is used to simulate the distribution of binary motion pixel map, so that we can also detect areas that belong to target without obvious intensity changes. We assume that there exist  $K$  GMM components. Then we calculate optimal mean and variance for each component using Expectation Maximization algorithm. The probability of motion targets then can be expressed as [5]

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

$$p(F1(i, j)/C1) = \sum_{k=1}^K \pi_k N((i, j)/\mu_k, \Sigma_k), \quad (6)$$

where  $\pi_k$  is the mixing coefficient for each Gaussian component. It can be determined in the EM iteration with the constraint of  $\sum_{k=1}^K \pi_k = 1$ .  $F_1$  is the motion cue and  $C_1$  is the pixel class for foreground objects.

The proposed algorithm work well if the number of GMM components  $K$  is larger than the number of moving objects in the frame. In motion field of whole frame, the motion cue of each pixel is determined by the combination of all Gaussian components. Therefore, each motion object can be represented by one or more Gaussian components. Hence  $K$  should be chosen large enough.

## B. Including Local Properties

In order to include local properties such as pixel location and chromatic values, into motion segmentation, a feature vector  $F_{ij}$  is formed. This feature vector consist of spatial and color information of each pixel [6], [7];

$$F_{ij} = (R_{ij}, G_{ij}, B_{ij}, w \cdot i, w \cdot j) \quad (7)$$

Here,  $R_{ij}$ ,  $G_{ij}$ ,  $B_{ij}$  are intensity values of  $R$ ,  $G$ ,  $B$  channel at the pixel  $P(i, j)$ ,  $w$  is the weighting factor between the color and position space.

On this vector space k-means clustering is applied. Frame size and scene complexity is taken into consideration for the predetermination of the number of clusters. Then we find the number of motion pixels in each cluster. Then the cluster's likelihood of motion can be determined by

$$p(F2(i, j)/c1) = \frac{\text{no.of motion pixels within the cluster}}{\text{total no.of pixels of the cluster}} \quad (8)$$

Here  $F_2$  represents the locality cue.

## C. Combination of Motion and Locality Cues.

Assuming that the motion cue and locality cue are independent, the combined conditional probability is given by [8], [9];

$$p(A/C_i) = \prod_{k=1}^2 p(F_k/C_i), \quad (9)$$

Here  $p(A/C_i)$  is the combined conditional probability for the pixel class  $C_i$ .

## IV. RAIN DETECTION

Second step of our method is rain detection.

### A. Differencing and Thresholding

The grey scale intensity difference between two successive frames are calculated and then thresholded. The threshold value is set such that the intensity fluctuations caused by rain can be detected. The typical value is  $D_{th} = 3$ .

$$I_{diff} = \begin{cases} 0, & I_N - I_{N-1} < D_{th} \\ 1, & I_N - I_{N-1} \geq D_{th} \end{cases} \quad (10)$$

Thus we can calculate a binary difference map  $I_{diff}$ .

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

## B. Applying Photometric and Chromatic Constraints

Photometric and chromatic constraints [2], [8], are applied on binary difference map  $I_{diff}$  to detect rainy pixels. First, the constraint of intensity fluctuation range is applied as  $\epsilon = 3 \sim 20/255$ . Then by considering the speed of rain streak, the connected area of rain streak is limited as  $\theta = 30 \sim 50$  pixels. The chromatic condition [9] we applied here is the absolute sum of color channel differences caused by rain should be within the limit  $\phi = 15/255$ . Now the pixels in binary difference map that fail these conditions are eliminated to form the rain mask  $I_{Rain}$ .

## C. Motion Exclusion

The results of motion segmentation and rain detection are then compared and  $I_{Rain}$  is divided into three groups. Rain pixels in motion area will form the group  $S_m$ ; rain pixels in stationary area will form the group  $S_b$ ;  $S_p$  is the third group of pixels that are not at all covered by rain.

## V. SCENE RECOVERY

We have to create three buffers for rain removal. Input video buffer  $B_I(len, wid, stk)$ , rain buffer  $B_R(len, wid, stk)$ , and motion buffer  $B_M(len, wid, stk)$ . Here  $len \times wid$  is the frame size and  $stk$  is the depth of the buffer. Here depth is taken as 9. In input frame buffer, newest frame is pushed on the top and the oldest frame is pushed out from the bottom of the buffer. The rest of the frames update accordingly. For each input video frame, the rain buffer records the binary rain map  $I_{Rain}$  and motion buffer records the corresponding binary motion map  $I_m$ . Scene recovery algorithm work on the central frame  $B_I(x, y, n)$ , Here  $n$  is taken as  $n = \frac{(stk-1)}{2}$  so that we can get previous and future information in the video for better scene recovery.

### A. Calculating Weight

We have to design an 88-spatial-temporal neighborhood  $V$  for rain pixel recovery. The central pixel of the neighborhood is  $B_I(x, y, n)$  which has 8 temporal neighbors from  $B_I(x, y, n-4)$  to  $B_I(x, y, n+4)$  and 80 spatial neighbors in the area  $B_I((x-4) \text{ to } (x+4), (y-4) \text{ to } (y+4), n)$ . Similar neighborhood set up is given in [10]. Using the value of the pixels in the neighborhood we predict the center rainy pixel by a weighted sum. The definition of the pixel is given by

$$w_{x,y,t}(i,j,t) = \tilde{B}_R(x,y,n) \times \exp\left(-\alpha \|v(x,y,n) - v(x,y,t)\|^2 - \beta \|v(x,y,n) - v(i,j,n)\|^2\right) \quad (11)$$

here  $\tilde{B}_R$  is the binary compliment of  $B_R$ . It is used to avoid the pixels that are also covered by rain. The term  $(-\alpha \|v(x,y,n) - v(x,y,t)\|^2)$  regulates the filter weight along the time axis and  $(-\beta \|v(x,y,n) - v(i,j,n)\|^2)$  regulates the filter weight in the spatial neighborhood.

### B. Replacing Pixels in Static Background

These are the pixels in the set  $S_b$ . For this set, filter coefficients are set as  $\alpha = \frac{stk}{2}, \beta = 0$ . Here we can see that no spatial neighbor values are used here. It gives a Gaussian filter with variance  $\alpha$  along the time axis. The rain removal filter kernel for these pixels could be defined as

$$B_I(x,y,n) = \frac{\sum_{v(i,j,t) \in V} w_{x,y,n}(i,j,t) \tilde{B}_M(i,j,t) B_I(i,j,t)}{\sum_{v(i,j,t) \in V} w_{x,y,n}(i,j,t) \tilde{B}_M(i,j,t)} \quad (12)$$

where  $\tilde{B}_M$  is the binary compliment of motion buffer. This will avoid the pixels in motion objects.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

### C.Replacing Pixels in Motion Objects

For the rain covered pixels in motion objects  $\alpha = 1, \beta = \frac{1}{4\sqrt{2}}$ . Since it gives consideration to both spatial and temporal neighborhood it gives a two dimensional Gaussian filter. The filter kernel is given as

$$B_l(x, y, n) = \frac{\sum_{v(i,j,t) \in V} w_{x,y,n}(i, j, t) B_M(i, j, t) B_l(i, j, t)}{\sum_{v(i,j,t) \in V} w_{x,y,n}(i, j, t) B_M(i, j, t)} \quad (13)$$

### D.Retaining Pixels Uncovered by Rain

Finally the pixels that are not covered by rain are retained.

## VI. EXPERIMENTAL RESULTS

Figures shows the results of experiments carried out on video with highly dynamic scenes. Fig.1.shows the result of motion segmentation using Gaussian Mixture Model and k means clustering. The number of clusters is taken as 20. Fig.2shows rain detection.(b) is the binary difference map and(c) is the rain detection output. Rain removed output is shown in Fig.3.

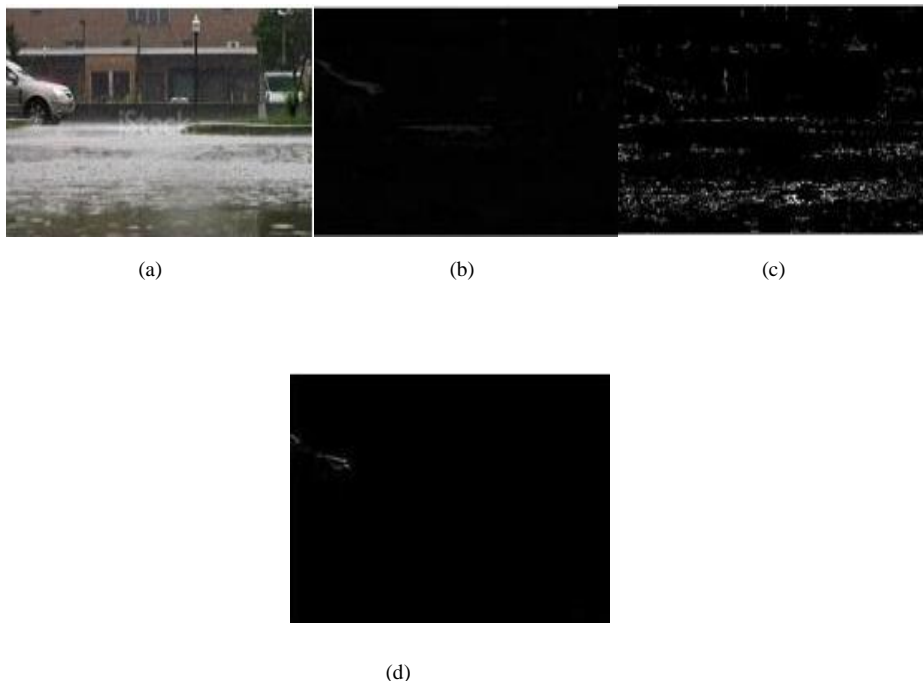


Fig.2. Motion segmentation (a) Input frame (b) Result of motion field estimation (c) is the result obtained after clustering and (d) shows motion segmentation result.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

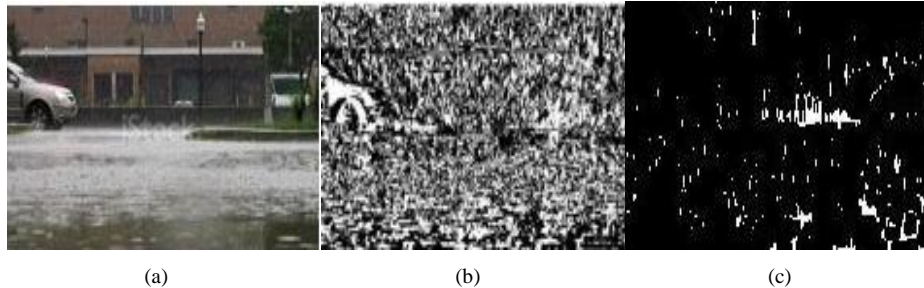


Fig.3. Rain detection (a) Input frame (b) result of differencing and (c) result of rain detection



(a)(b)

Fig.4. Rain removal (a) Input frame (b) result of rain removal

To recover rainy pixels an 88 neighborhood system is designed for each pixel in the input video frame shown in (c) and the recovered scene is shown in (d).

## VII. CONCLUSION

We have implemented a method for removing rainy pixels from highly dynamic videos. Since the method is based on motion segmentation, rainy pixels in moving target area and stationary background are treated separately. Chromatic and photometric constraints of rain are applied for rain detection. Both spatial and temporal information of each pixel is exploited during scene recovery for better result. Experiments are conducted on highly dynamic video and results are analysed.

## REFERENCES

- [1] J. Chen and L. P. Chau, "A rain pixel recovery algorithm for videos with highly dynamic scenes," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1097-1104, Mar. 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] Verry and T. Poggio, "Motion field and optical flow: Quantitative properties," *IEEE Trans. Pattern Anal. March. Intell.*, vol. 11, no. 5, pp. 490-498, May 1989.
- [4] K. Horn and B. G. Shunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185-203, Jan. 1981.
- [5] M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006, pp. 423-455, Ch. 9.
- [6] 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.
- [7] J. Fan, D. Yau, A. Elmagarmid, and W. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1454-1466, Oct. 2001.
- [8] K. Garg and S. K. Nayar, "When does camera see rain?" in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. . Oct. 2005, pp. 1067-1074.
- [9] K. Garg and S. K. Nayar, "Vision and rain," *int. J. Comput. Vis.*, vol. 75, no. 1, pp. 3-27, 2007.