Phase Congruency Based Technique for the Removal of Rain from Video

Varun Santhaseelan and Vijayan K. Asari

University of Dayton, 300 College Park, Dayton, OH, USA
{santhaseelanv1,vijayan.asari}@notes.udayton.edu

Abstract. Rain is a complex dynamic noise that hampers feature detection and extraction from videos. The presence of rain streaks in a particular frame of video is completely random and cannot be predicted accurately. In this paper, a method based on phase congruency is proposed to remove rain from videos. This method makes use of the spatial, temporal and chromatic properties of the rain streaks in order to detect and remove them. The basic idea is that any pixel will not be covered by rain at all instances. Also, the presence of rain causes sharp changes in intensity at a particular pixel. The directional property of rain streaks also helps in the proper detection of rain affected pixels. The method provides good results in comparison with the existing methods for rain removal.

Keywords: Phase congruency, rain removal, alpha blending.

1 Introduction

Nowadays, video surveillance is an integral part of security applications. Outdoor video surveillance has helped in tackling serious law and order situations. It is only natural that with the increasing popularity of video surveillance equipment, the need for algorithms that improve video quality has also increased. One of the major challenges in video quality improvement when we consider outdoor vision systems is the effect of bad weather conditions on video.

Conditions that impede video quality include presence of haze, snow, fog, smoke, rain, hail, etc. Haze, smoke and fog can be considered as steady weather conditions and they fall in a different category of video enhancement. Rain and snow can be considered as dynamic weather conditions that change with every frame in the video. While rain is highly directional snow particles fall in completely random directions. This paper deals with the removal of rain from video.

The classification of weather into steady (haze, mist and fog) and dynamic (rain, snow and hail) weather was done by Garg and Nayar [1]. They developed models based on the physical and photometric properties of rain drops. They used these models to detect rain and to remove them from videos. The main assumption in that case was the uniform size of rain drops and the equal velocity of rain drops. The variation in depth was not taken into consideration. This became a problem while trying to remove rain from videos that contained heavy rain. Brewer and Liu also used the physical properties of rain drops to detect and remove rain from videos [2].

M. Kamel and A. Campilho (Eds.): ICIAR 2011, Part I, LNCS 6753, pp. 30–39, 2011.

Garg and Nayar [3] also introduced an idea of changing camera parameters in order to reduce the effect of rain on the video. This method involved changing the camera parameters like F-number and exposure time individually or in tandem to reduce the effect of rain. The parameters were changed according to the nature of the scene. This method cannot be used in outdoor surveillance systems since manual adjustment of the camera parameters is not possible according to the weather conditions.

Park and Lee [5] came up with the idea of using a Kalman filter for the detection and removal of rain from videos. This method requires a periodic reset and cannot be adopted for videos taken from a moving camera. Barnum et al. [5] did a frequency space analysis of rain and snow affected videos. They modeled rain and snow in the frequency space based on the statistical properties of rain and snow streaks. Each rain streak was assumed to be a blurred Gaussian. The number of desired cycles to remove rain increases the number of frames to be used in the process.

Zhang et al. [6] used the spatio-temporal and chromatic properties of rain to remove rain from videos. Their idea was based on the fact that a pixel will not be covered by rain in every frame. They used an intensity histogram for each pixel constructed from all the frames in the video and used K-means clustering to differentiate between background pixels and rain affected pixels. This method works well except for the fact that all the frames in the video are used to construct the histogram.

The method proposed in this paper is along the lines of the idea used by Zhang et al. The proposed method uses phase congruency to detect candidate rain pixels. Since phase congruency is used, it is easier to incorporate the directionality property into the algorithm. The main advantage of this method in comparison to the method proposed by Zhang et al. is the fact that the number of frames used for detection and removal of rain affected pixels is minimal. Only the frames in the neighborhood are considered in this process.

The second section of this paper deals with the properties of rain streaks that appear on a video. This study has helped in the formulation of the algorithm. The third section explains in detail the steps involved in the algorithm and the feature extraction methods that have aided in rain detection and removal. Results and related discussion are included in the fourth section. A comparison with the existing methods is also provided in this section. The fifth section summarizes the findings in this paper and also discusses about the future work possible in this area.

2 Properties of Rain Streaks in Video

Most of the spatial, temporal and chromatic properties have been studied in detail by Zhang et al. These properties are utilized in this paper as well and are described briefly in this section.

2.1 Temporal Property

The human eye is able to see through rain mainly because all parts of the scene are not occluded by rain at all instances. As the depth of view increases, it becomes harder to distinguish between drops and the layer of rain appears as haze or mist [1].

This property holds true for occlusions due to rain in videos too. In this paper, we consider the removal of rain drops that can be distinguished separately in each frame. As the depth of view increases, the rain drops are not visible separately and the image enhancement problem becomes equivalent to haze removal. A close study of the intensity variation will show that the pixel intensity varies sharply when rain occludes a scene. This is illustrated in Fig. 1.

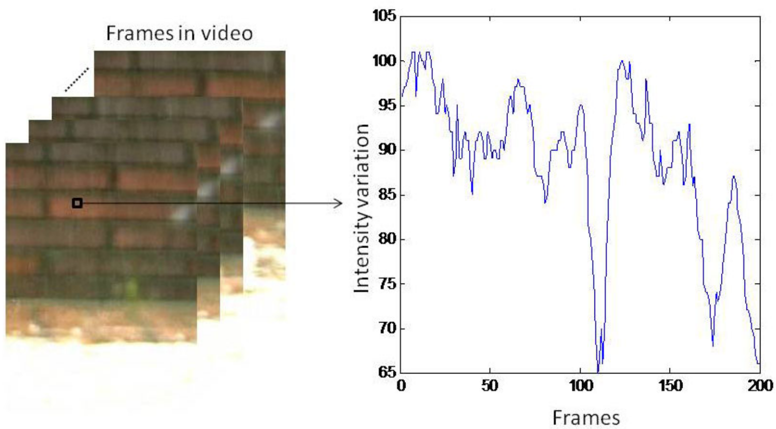

Fig. 1. Intensity variation for a pixel throughout a segment of the video containing heavy rain

The intensity variations plotted in Fig. 1 is for a video that contains heavy rain. It can be seen that the intensity tends to remain high if the density of rain is higher and therefore more frames will be required to compensate for the rain affected pixels. This is the case where considering one frame before and after the current frame becomes insufficient for rain removal.

2.2 Chromatic Property

While Garg and Nayar [1] showed that a rain drop refracts a wide range of light causing an increase in intensity at a particular pixel, Zhang et al. went ahead and showed that the change in levels for the individual color components of the pixel due to rain is proportional to its original intensity level. They showed that the standard deviation in each color component due to the presence of rain is almost the same.

2.3 Directional Property

Another observation that has been utilized by Garg and Nayar [1] is the directional property of rain in videos. If rain is present in a frame, all the rain streaks will be oriented in a single direction. They computed the correlation between neighboring pixels to detect rain affected pixels. This property is used in our proposed method while calculating phase congruency in a particular orientation.

3 Algorithm for Rain Detection and Removal

The proposed algorithm can be condensed into four steps as shown in Fig. 2.

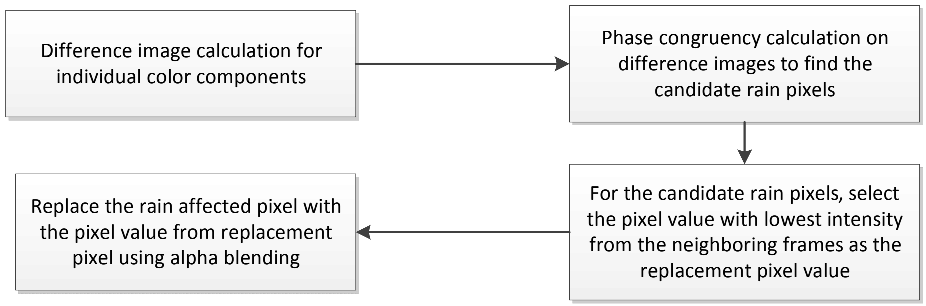

Fig. 2. Algorithm for rain detection and removal

3.1 Difference Image Calculation

The temporal property of rain described in the previous section indicates that there will be a positive change in intensity of a rain affected pixel. The chromatic property suggests that the standard deviation in all the three components will be the same when there is rain occluding a pixel. In this step, we compute the difference image of the current frame with respect to its neighbors. The difference image is computed for all the three color components separately. The neighboring frame is subtracted from the current frame. If the resultant value at a pixel is negative, it is clamped to zero. The presence of rain causes an increase in intensity. Therefore, only positive differences will be considered. For any pixel, if the standard deviation of the individual color components is different from each other, the pixel cannot be considered as rain-affected.

When differences of images are computed, the main criterion is the number of neighboring frames to be considered. As mentioned in section 2.1, if heavy rain is present, the number of frames to be considered will be more. In our case, we have used eight neighboring frames for the computation. This has resulted in good results with most of the rain removed from the video.

3.2 Applying Phase Congruency on the Difference Images

Phase information in the difference images are used to identify rain streaks in a particular frame. Phase congruency feature mapping gives an accurate measure of the variation in edges of rain streaks and is used in this paper.

3.2.1 Phase Congruency Features

The importance of phase information of an image is illustrated in Gonzalez and Woods [7]. The phase information in an image contains the essential details. When an image containing rain is considered, the rain streaks can be assumed to be the finer details in the image. These fine details will be reflected in the phase changes of the image. This basic idea is the reason behind the inclusion of phase congruency feature detection as part of rain streak detection algorithm.

The principal reason that humans are able to visually recognize individual rain streaks in a particular frame is because there is a step change in intensity along the edge of the rain streak. Phase congruency (PC) is a feature detection mechanism that recognizes those edges and is invariant to illumination and contrast. The key observation that led to the development of phase congruency algorithm is that the Fourier components of an image are maximal in phase where there are edges or lines. Features are identified according the extent to which the Fourier components are in phase.

The PC computation method adopted in this paper was proposed by Peter Kovesi [8]. His method was based on the local energy model developed by Morrone and Owens [9]. They observed that the point of strong phase congruency corresponds to a point of maximum energy. Let $I(x)$ be an input periodic signal defined in $[-\pi, \pi]$. $F(x)$ is the signal ($I(x)$) with no DC component and $H(x)$ is the Hilbert Transform of $F(x)$ which is a 90° phase shifted version of $F(x)$. The local energy, $E(x)$ can then be computed from $F(x)$ and its Hilbert Transform as in (1).

$$E(x) = \sqrt{F^2(x) + H^2(x)} \tag{1}$$

It has been shown in earlier research [10] that the energy is equal to the product of phase congruency and the sum of Fourier amplitudes as in (2).

$$E(x) = PC(x) \sum_n A_n \tag{2}$$

Therefore the peaks in phase congruency correspond to the peaks in the energy function. Equation (2) also shows that the phase congruency measure is independent of the overall magnitude of the signal, thus making the feature invariant to changes in illumination and contrast. The components, $F(x)$ and $H(x)$ are computed by the convolution of the signal with a quadrature pair of filters. Logarithmic Gabor filters are used in this case. Consider $I(x)$ as an input signal and M_n^e and M_n^o are the even symmetric and odd symmetric components of the log Gabor function at a particular scale, n. Then the amplitude and phase for the input signal in the transformed domain is obtained as in (3) and (4) where $o_n(x)$ and $e_n(x)$ are the responses for each quadrature pair of filters as given in (5).

$$A_n = \sqrt{e_n^2(x) + o_n^2(x)} \tag{3}$$

$$\phi_n(x) = tan^{-1}(o_n(x)/e_n(x)) \tag{4}$$

$$[e_n(x), o_n(x)] = [I(x) * M_n^e, I(x) * M_n^o] \tag{5}$$

The values for $F(x)$ and $H(x)$ can be computed as shown in (6) and (7).

$$F(x) = \sum_n e_n(x) \tag{6}$$

$$H(x) = \sum_n o_n(x) \tag{7}$$

When the Fourier components are very small, the problem of computing phase congruency becomes ill-conditioned. This problem is solved by adding a small constant to the sum of Fourier components as shown in (8).

$$PC(x) = \frac{E(x)}{\varepsilon + \sum_n A_n} \tag{8}$$

Equation (8) is the final equation for solving phase congruency. This equation can be applied to a two dimensional signal like an image for various orientations. In this paper, the analysis is to be done on an image.

For an image, the first step is to convolve the image with a bank of two dimensional log Gabor filters. The filter has a transfer function as shown in (9).

$$G(w) = e^{(-\log(w/w_0)^2)/(2 log(k/w_0)^2)} \tag{9}$$

where w_0 is the filter's center frequency and k/w_0 is kept constant for various w_0. The cross-section of the transfer function of the filter can be represented as in (10).

$$G(\theta) = e^{-(\theta-\theta_0)^2/(2\sigma_\theta^2)} \tag{10}$$

where θ_0 represents the orientation of the filter and σ_θ is the standard deviation of the Gaussian spreading function in the angular direction. As in equation (5) the even symmetric and odd symmetric components at a particular scale and orientation can be computed as shown in (11).

$$[e_{no}(x,y), o_{no}(x,y)] = [I(x,y) * M_{no}^e, I(x,y) * M_{no}^o] \tag{11}$$

The amplitude of the response at a particular scale and orientation can be computed as in (12), and the calculation of phase congruency for an image is as shown in (13).

$$A_{n0} = \sqrt{e_{no}^2(x,y) + o_{no}^2(x,y)} \tag{12}$$

$$PC(x,y) = \frac{\sum_o \sqrt{(\sum_n e_{no}(x,y))^2 + (\sum_n o_{no}(x,y))^2}}{\varepsilon + \sum_o \sum_n A_{no}(x,y)} \tag{13}$$

In this paper, all the orientations are not considered when phase congruency features are computed. This is because of the directional property of rain streaks. The rain drops always fall towards the ground and the variation in orientation is minimal. This fact helps in discarding most of the orientations. The calculation of difference images and phase congruency features are illustrated in Fig. 3.

3.3 Background Pixel Search

After applying phase congruency, only the candidate pixels (rain affected pixels) with intensity variations in neighboring frames remain in the processed image. The next step is to eliminate the false positives which may have occurred due to the presence of external noises. If a pixel is detected as a candidate rain pixel in all the phase congruency images of the difference images, it is very likely that it happened due to noise. These pixels are eliminated from the group of candidate rain pixels.

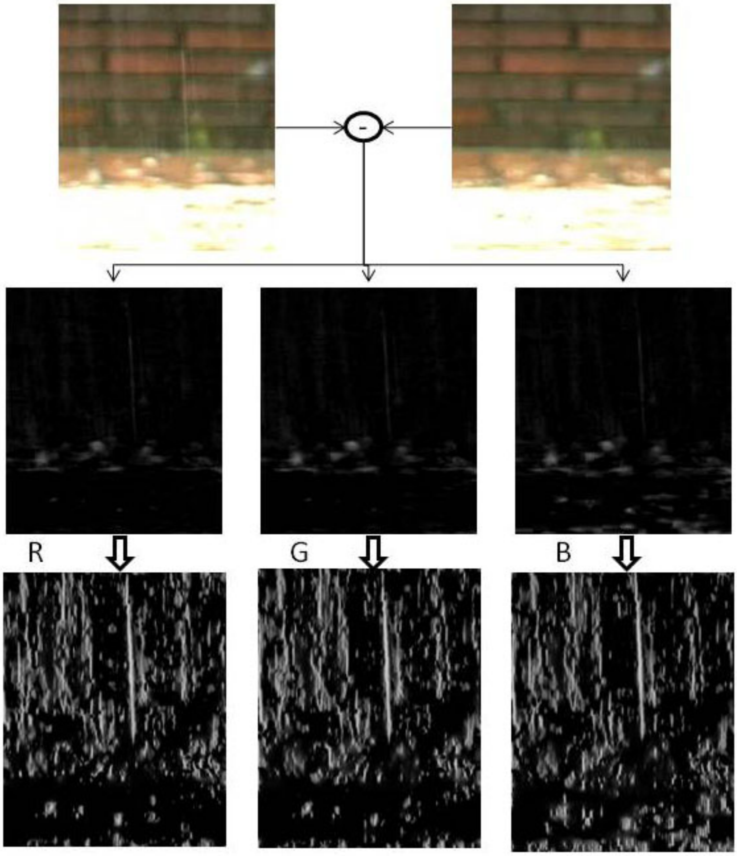

Fig. 3. The computation of difference image and the image with phase congruency features for R, G and B components

The next step is to find out the background intensity levels of the rain affected pixels. A search is performed on the neighboring frames. The pixel value that has the lowest intensity levels within the neighbors is selected as the background intensity of the rain affected pixel.

3.4 Compensate for Rain Affected Pixels

Garg and Nayar [1] used the average of the pixel intensities in neighboring two frames to compute the intensity value for the pixel to be replaced. This method fails when the pixel is affected by rain continuously. The method by Zhang et al. gave better results. They used alpha-blending to calculate the intensity value for the rain affected pixel as shown in (14).

$$C = \alpha C_b + (1-\alpha) C_r \tag{14}$$

The new color is denoted as C, the background color is denoted as C_b and the color of the rain-affected pixel is denoted as C_r.

4 Results and Discussion

The results shown in Fig. 4 show that phase congruency features can be used in differentiating rain streaks from the original scene with the help of the spatio-temporal and chromatic properties of rain.

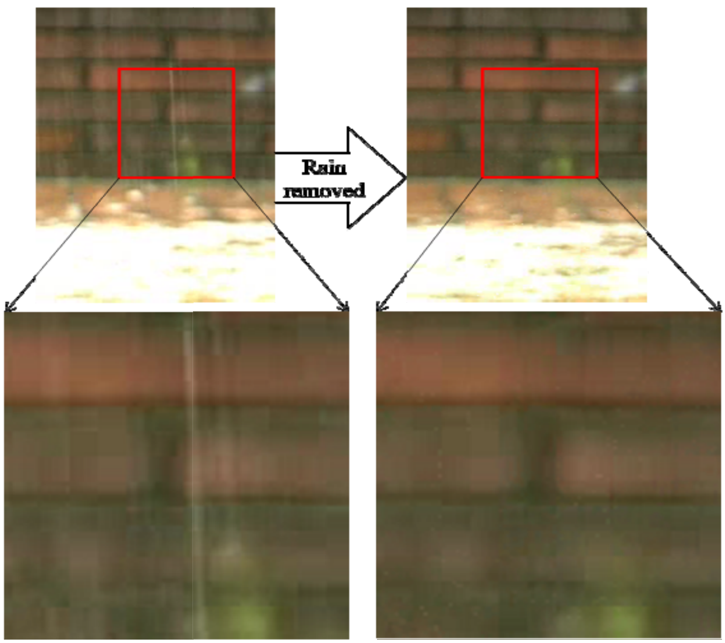

Fig. 4. The rain streaks in the frame shown on the left side have been removed and the resultant image is shown on the right side. Please refer the following link for the complete video: http://visionlab.udayton.edu/research/rain.php.

A performance comparison of the proposed algorithm is illustrated in Fig. 5. In comparison with the algorithm presented in [6], it is observed that the dynamic nature of the scene is preserved more in our method. For example, the intensity variations caused on the water puddles due to water drops are preserved more in Fig. 5(c). This is because our algorithm used comparatively much lesser number of frames (eight frames in the present experiment) for the removal of rain.

The important factor in our algorithm that affects the quality of the output video is the number of neighboring frames that are considered. It has been observed that increasing the number of frames will increase the quality of video, especially when the aim is to remove heavy rain. This increase in quality comes at the expense of loss of preservation of motion of objects in the frame. The effect of increasing the number of frames is illustrated in Fig. 6. While one trial used six neighboring frames, twelve

(a) (b) (c)

Fig. 5. This figure compares the result between our method and the method by Zhang et al. [6]. (a) The original frame; (b) Rain removed by the method in [6]. (c) Rain removed by our method.

Fig. 6. The image on left is a video frame from which rain was removed using six neighboring frames and the image on the right utilized twelve neighboring frames. The presence of extra streaks in image shown on the left are highlighted.

frames were used for the second trial. It was observed that the addition of more frames for compensation reduced the number of blurred streaks in every frame.

5 Conclusion

A new method based on phase congruency features was used to detect and remove rain from videos. The method was formulated based on the temporal, spatial and chromatic properties of rain streaks in video. In comparison with the method of Zhang et al., it has been found that our method provides results of the same quality with lesser number of frames. It was also observed that the slight movements of objects in the video are captured better in our method.

This paper dealt with removal of rain from videos that did not have any camera movement. One way to deal with such a scenario is to stabilize the video [11] before applying the algorithm for rain removal as done by Zhang et al. Another area for future improvement is to tackle the problem of moving objects in the foreground of the rain as well as in the rain. In such cases, the aim will be to estimate the rain component in video from lesser number of frames.

References

1. Garg, K., Nayar, S.: Vision and rain. International Journal of Computer Vision 75, 3–27 (2007)
2. Brewer, N., Liu, N.: Using the shape characteristics of rain to identify and remove rain from video. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 451–458. Springer, Heidelberg (2008)
3. Garg, K., Nayar, S.K.: When does a camera see rain? In: International Conference on Computer Vision 2005, pp. 1067–1074 (October 2005)
4. Park, W.J., Lee, K.H.: Rain removal using Kalman filter in video. In: International Conference on Smart Manufacturing Application, pp. 494–497 (April 2008)
5. Barnum, P., Kanade, T., Narasimhan, S.: Spatio-temporal frequency analysis for removing rain and snow from videos. In: Workshop on Photometric Analysis For Computer Vision (2007)
6. Zhang, X., Li, H., Qi, Y., Leow, W.K., Ng, T.K.: Rain removal in video by combining temporal and chromatic properties. In: IEEE International Conference on Multimedia and Expo 2006, pp. 461–464 (July 2006)
7. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc., Boston (1992)
8. Kovesi, P.: Image features from Phase Congruency. Videre: Journal of Computer Vision Research 1(3) (Summer 1999)
9. Morrone, M.C., Owens, R.A.: Feature detection from local energy. Pattern Recognition Letters 6, 303–313 (1987)
10. Venkatesh, S., Owens, R.A.: An energy feature detection scheme. In: The International Conference on Image Processing, pp. 553–557 (1989)
11. Matsushita, Y., Ofek, E., Tang, X., Shum, H.Y.: Full-frame video stabilization with motion inpainting. In: Proceedings of CVPR 2005, vol. 1, pp. 50–57 (2005)