

Utilizing Local Phase Information to Remove Rain from Video

Varun Santhaseelan · Vijayan K. Asari

Received: 5 December 2013 / Accepted: 27 August 2014 / Published online: 13 September 2014
© Springer Science+Business Media New York 2014

Abstract In the context of extracting information from video, bad weather conditions like rain can have a detrimental effect. In this paper, a novel framework to detect and remove rain streaks from video is proposed. The first part of the proposed framework for rain removal is a technique to detect rain streaks based on phase congruency features. The variation of features from frame to frame is used to estimate the candidate rain pixels in a frame. In order to reduce the number of false candidates due to global motion, frames are registered using phase correlation. The second part of the proposed framework is a novel reconstruction technique that utilizes information from three different sources, which are intensities of the rain affected pixel, spatial neighbors, and temporal neighbors. An optimal estimate for the actual intensity of the rain affected pixel is made based on the minimization of registration error between frames. An optical flow technique using local phase information is adopted for registration. This part of the proposed framework for removing rain is modeled such that the presence of local motion will not distort the features in the reconstructed video. The proposed framework is evaluated quantitatively and qualitatively on a variety of videos with varying complexities. The effectiveness of the algorithm is quantitatively verified by computing a no-reference image quality measure on individual frames of the reconstructed video. From a variety of experiments that are performed on output videos, it is shown that the proposed technique performs better than state-of-the-art techniques.

Keywords Rain removal · Phase congruency · Monogenic signal · Optical flow

1 Introduction

The advent of low cost technology in the field of video capture systems has made it easier for various organizations to adopt surveillance technology. However, the major challenge when it comes to videos from surveillance videos is the detrimental effect of bad weather conditions. Bad weather manifests itself in videos in the form of low lighting, blurred scene content, highly saturated regions of illumination, etc. Different kinds of weather conditions affect the captured video in different ways. Research by [Narasimhan and Nayar \(2002\)](#) makes a broad classification of weather conditions into two, static and dynamic. The classification is based on the size of particles that cause a particular type of weather. The larger sized particles are visibly affected by gravity and hence cause a large change in the video over a sequence of frames. Static weather conditions are caused by particles that are very small (less than $10\mu m$) and manifests itself as haze, fog or cloud. In the case of dynamic weather conditions, particles are larger than $100\mu m$. Rain and snow are examples of dynamic weather conditions and the changes in video over subsequent frames is very much pronounced.

In this research, the focus is on removing rain streaks from videos. The objective is to study the characteristics of rain streaks in video and thus devise a framework for accurate reconstruction of a scene by removing all the rain streaks. The process to remove rain involves two steps: (1) detect the presence of rain or locate where rain streaks appear in a frame of video, (2) estimate the actual intensity of a pixel that is affected by rain. In order to detect a rain streak, the primary challenge is to characterize rain streaks that appear

Communicated by Srinivasa Narasimhan.

V. Santhaseelan (✉) · V. K. Asari
University of Dayton, 300 College Park, Dayton, OH 45469, USA
e-mail: santhaseelanv1@udayton.edu

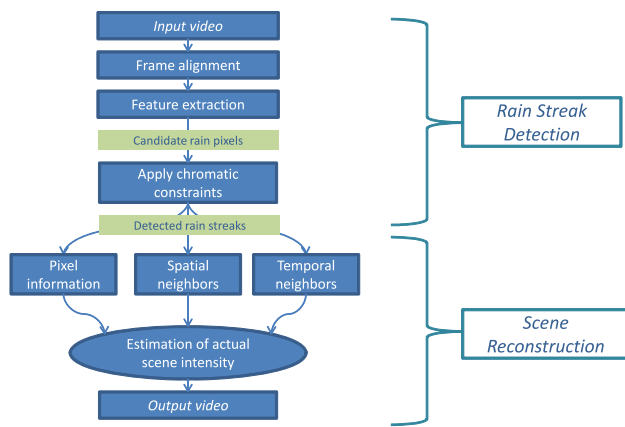


Fig. 1 Architecture of the proposed framework to remove rain from video

in video in terms of edge information, chromatic properties and their spatio-temporal behavior. The challenges associated with scene reconstruction are edge preservation for the actual scene, maintaining temporal motion smoothness and prevent loss of useful information.

In the first part of this research, a feature based framework for detecting rain streaks is developed. In the second part, an effective method for compensation of rain affected pixels is developed. Figure 1 gives a brief overview of the complete framework for rain streak detection and removal from video.

In the method for rain streak detection, frames in the video are aligned using global phase correlation to eliminate the effect of global motion in the scene caused by movement of the camera. Variations in the scene from one frame to the next are captured using phase congruency features providing a set of candidate rain pixels. The number of false detections are reduced by the application of a chromatic constraint. The final output of the rain streak detection algorithm is the set of rain streaks present in the scene along with certain false detections caused by the local motion components in the scene. The second part of the framework reconstructs the actual scene in a robust manner by eliminating the effect of false detections due to local motion and have the best possible estimate of the actual scene intensity. The algorithm utilizes information embedded in the rain affected pixel, information from its spatial neighbors and the information from temporal neighbors for the estimation process.

The proposed framework addresses the problem of removing rain by giving equal emphasis for the rain streak detection and scene reconstruction. The aim is for both parts of the framework to complement each other to have a better solution. Therefore, the constraints involved in the part for detection of rain streaks may be relaxed to make sure all the rain streaks are detected while increasing the number of false detections. The second part of the framework would

make sure that the effect of false detections have no effect on the quality of the resultant video.

1.1 Previous Work

Most of the methods to remove rain that exist in the field of computer vision were reviewed by [Tripathi and Mukhopadhyay \(2012a\)](#). The procedure for rain streak detection has been the focus of attention for most methods in current literature.

Since rain streaks do not occlude a scene at all times, the logical step is to filter each pixel in a temporal direction to have an estimate of the intensity of actual scene. The initial approach towards removing rain was the use of temporal median filter as done by [Starik and Werman \(2003\)](#) and [Hase et al. \(1999\)](#). The method was successful when there was no motion component associated with the video, in the global or local sense. [Zhang et al. \(2006\)](#) exploited the spatio-temporal and chromatic properties of rain streaks to detect rain streaks in video. A temporal histogram was constructed for every pixel from which a decision was made on whether a pixel is part of rain streak or background. The main disadvantage is that the technique requires a histogram to be constructed and therefore requires at least fifty frames. [Shen and Xue \(2011\)](#) proposed a fast method based on optical flow that is used to detect rain. A three-dimensional anisotropic diffusion method is used to estimate the background scene intensity from the spatio-temporal neighbors. The method for detection is incomplete because it is highly probable that the optical flow components could be incorrectly calculated for small regions. Another method utilizing spatio-temporal properties of rain was proposed by [Xue et al. \(2012\)](#). Spatial features, wavelet features and motion constraints were combined to increase the accuracy of rain streak detection. The major disadvantage of this technique is the use of bilateral filter in order to eliminate false detections. It is highly probable that large streaks are not eliminated by the filter. The image inpainting technique used for reconstructing the scene to remove rain would be incapable of compensating for rain affected pixels in complex scenes and when heavy rain is to be removed. [Park and Lee \(2008\)](#) modeled the variation of intensity at a pixel in a scene using a Kalman filter. However, this method fails when motion component is present in the scene. A method based predominantly on the shape properties of rain streaks was presented by [Brewer and Liu \(2008\)](#). The latest research with regard to removing rain streaks was presented by [Kim et al. \(2013\)](#). The paper presents a method to detect rain streaks on a single image by characterizing the shape of a streak. The shape of a single streak is assumed to be an elongated ellipse. The authors used kernel regression to detect rain streaks in an image. The detected rain affected pixels are reconstructed using a non-local means filter. The method works well for well defined rain streaks with defi-

nite shape. However, streaks could be really blurred affecting the performance of the algorithm. The reconstruction technique utilizes information from the same image. In the case of heavy rainfall, this technique would not find enough information for reconstruction.

Garg and Nayar (2007) developed models for rain streaks based on the physical and photometric properties of rain drops. They used these models to detect rain streaks 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. The process of estimating the background intensity was not sufficient in regions with rapid movement. Barnum et al. (2010) 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 model was unsuccessful in eliminating blurred streaks from video.

Bossu et al. (2011) segmented out candidate rain streaks from the foreground based on Gaussian Mixture Models and by applying constraints based on shape and size characteristics of rain streaks. The method utilized the property of uniform direction of rain streaks to create histogram of orientated streaks (HOS) for reducing the false detections. Tripathi and Mukhopadhyay (2011) developed a Bayesian framework to detect rain. The method is extended to remove rain from video by Tripathi and Mukhopadhyay (2012b).

A learning based method to remove rain from a single image was proposed by Kang et al. (2012a). Rain streaks were detected using a method based on Morphological Component Analysis (MCA) and dictionary learning. The method uses a bilateral filter to get the initial set of candidate rain pixels. The bilateral filter could miss large and blurred streaks. The method was extended to have a self-learning mechanism by Kang et al. (2012b). Context information in a scene was used for assisting rain streak detection as an improvement of this method by Huang et al. (2012).

In almost all the aforementioned methods, the main concentration has been on improving the process of detecting rain streaks. This would affect the ability of algorithms to remove all rain streaks present in the video, and blurred rain streaks are mostly missed by detection techniques. This motivated the development of a robust reconstruction technique proposed in this paper. By making use of a detection process that does not miss any rain streaks, more emphasis is laid on the design of the reconstruction process to compensate for rain affected pixels and preserve the details present in false detections. The following contributions are made in this paper—(1) development of a novel technique for rain streak detection based on phase congruency features, and (2)

development of a robust reconstruction technique to remove rain from video by compensating for rain affected pixels.

The paper is organized as follows. Section 2 provides an overview of the characteristics of rain streaks in video. The method for detecting rain streaks is presented in Sect. 3. The proposed technique for reconstruction to remove rain streaks is given in Sect. 4. In Sect. 5, the experimental procedures are explained. The paper concludes in Sect. 6.

2 Characteristics of Rain Streaks in Video

A set of generalized characteristics can be inferred from the various models of raindrop explained in the previous section. These characteristics form the foundation for the framework developed for detecting rain streaks. In addition to the elaborate research presented by Garg and Nayar (2003) regarding properties of rain, some significant characteristics of rain are presented by Zhang et al. (2006). Some of the essential properties of rain streaks in video are explained in this section.

2.1 Temporal Property

The human eye is able to see through rain due in large part to the fact that all parts of the scene are not occluded by rain at all instances. The removal of rain streaks, that can be considered as dynamic components that vary from frame to frame, is the focus of this research. As the depth of view increases, the rain streaks are not separately visible and the image enhancement problem becomes equivalent to haze removal. Previous research by Garg and Nayar (2007) has shown that the pixel intensity increases sharply when rain occludes a scene. This is due to the fact that the resultant intensity of any raindrop on an image is the result of the radiances due to refraction, specular reflection and internal reflection. In the case of heavy rain, the intensity tends to remain high in comparison to the actual scene intensity. Therefore, more neighboring frames may 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 to estimate the intensity of the background.

2.2 Chromatic Property

Garg and Nayar (2007) showed that a rain drop refracts a wide range of light causing an increase in intensity at a particular pixel. Zhang et al. (2006) showed that the change in levels for the individual color components of the pixel due to rain is the same. Assume that the change in color components red (R), green (G) and blue (B) is ΔR , ΔG , and ΔB respectively. It is observed that the mean of ΔR , ΔG , and ΔB for any spatial neighborhood is the same. For the same neighborhood, the standard deviation of ΔR , ΔG , and ΔB is also the same.

2.3 Directional Property

Another observation that has been utilized by [Garg and Nayar \(2004\)](#) is the directional property of rain in videos. If rain is present in a frame, all the rain streaks will be oriented in similar direction. They computed the correlation between neighboring pixels to detect rain affected pixels. This property is used in the method for detecting rain streaks. It is embedded in the calculation of phase congruency.

3 Detecting Rain Using Phase Congruency Features

From a human visual perspective, rain streaks are sensed because of the rapid changes from frame to frame. Therefore, by finding the difference between two successive frames, it is possible to identify possible rain-affected pixels. The number of non-rain pixels detected as part of rain increases with the increase in motion component associated with the frame, either local or global. It is necessary to detect those features that are significantly altered from frame to frame in terms of visual perception. By incorporating phase based edge feature detection into the algorithm, it is observed that the detection of rain streaks could be done effectively.

3.1 Significance of Phase Information

The importance of phase information of an image was illustrated by [Oppenheim and Lim \(1981\)](#). In the case of a scene affected by rain, the dominant structural information between frames remains mostly the same. This helps in having phase based correlation techniques used to register two frames that are affected by rain. Previous research by [Mechler et al. \(2002\)](#) indicates that the human feature detection mechanism tends to be more aligned towards regions of phase congruency. Phase based information is robust towards changes in illumination as well. The local intensities act as a confidence measure for the reliability of sensing. Phase based reconstruction has been found to be much better in terms of perception. The latest research by [Wadhwa et al. \(2013\)](#) for magnification of videos using phase processing has achieved really good results.

In the context of rain streak detection, the spatial variations are much more localized. These localized bright streaks need to be extracted from a frame irrespective of the illumination in the local neighborhood. Therefore, phase congruency based features are used to detect rain streaks. Any kind of noise that caused intensity variations in a scene could be completely captured using phase congruency features irrespective of the local illumination component. With the help of the oriented filter in the feature computation technique, edges in a particular orientation can be isolated to utilize the directional property of rain streaks.

3.2 Phase Congruency

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 to the extent to which the Fourier components are in phase.

3.2.1 Choice of Band-Pass Filter

In order to extract phase information from images, the first step is to convolve the two dimensional signal with a pair of quadrature filters. Gabor filters have been very popular as band pass filters, especially in the case of phase-based information extraction. They are efficient in providing high localization in terms of space and frequency. However, the Gabor filter is not very efficient when the information that needs to be extracted is spread over a broad spectrum and spatial localization is required. In the case of rain streaks, such is the case as proven by [Barnum et al. \(2010\)](#).

The log-Gabor filter by [Field \(1987\)](#) overcomes the aforementioned drawbacks of the Gabor filter and is adopted as the band-pass filter in this research. In frequency space, the filter can be represented as in (1).

$$G(\omega) = \exp\left(\frac{-(\log(\omega/\omega_0))^2}{2(\log(k/\omega_0))^2}\right) \quad (1)$$

where ω_0 is the filter's center frequency and k/ω_0 is kept constant for various ω_0 . The cross-section of the transfer function of the filter can be represented as in (2).

$$G(\theta) = \exp\left(\frac{-(\theta - \theta_0)^2}{2\sigma_\theta^2}\right) \quad (2)$$

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.

The log Gabor filter does not have any DC component associated with it as evident from the equation. The filter response also has an extended tail that covers a wide range of frequencies providing higher localization in space when compared to the Gabor filter.

3.2.2 Method to Compute Phase Congruency Features

The PC computation method adopted in this research was proposed by [Kovesi \(1999\)](#). His method was based on

the local energy model developed by [Morrone and Owens \(1987\)](#). 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 $f_H(x)$ 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 (3).

$$E(x) = \sqrt{f^2(x) + f_H^2(x)} \tag{3}$$

It has been shown in earlier research by [Venkatesh and Owens \(1989\)](#) that the energy is equal to the product of phase congruency PC and the sum of Fourier amplitudes A_n as in (4).

$$E(x) = PC(x) \sum_n A_n \tag{4}$$

Therefore the peaks in phase congruency correspond to the peaks in the energy function. Equation (4) 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 $f_H(x)$ 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 . M_n^e and M_n^o can be represented in frequency domain as \mathcal{M}_n^e and \mathcal{M}_n^o and is expressed as in (5).

$$\begin{aligned} \mathcal{M}_n^e &= G(\omega) \\ \mathcal{M}_n^o &= i \text{sign}(\omega)G(\omega) \end{aligned} \tag{5}$$

where $i = \sqrt{-1}$. Then the amplitude and phase for the input signal in the transformed domain is obtained as in (6) and (7) where $o_n(x)$ and $e_n(x)$ are the responses for each quadrature pair of filters as given in (8).

$$A_n = \sqrt{e_n^2(x) + o_n^2(x)} \tag{6}$$

$$\phi_n(x) = \tan^{-1} \left(\frac{o_n(x)}{e_n(x)} \right) \tag{7}$$

$$[e_n(x), o_n(x)] = [I(x) * M_n^e, I(x) * M_n^o] \tag{8}$$

where $'*$ ' represents the convolution operation.

The values for $f(x)$ and $f_H(x)$ can be computed using $e_n(x)$ and $o_n(x)$ as shown in (9) and (10).

$$f(x) = \sum_n e_n(x) \tag{9}$$

$$f_H(x) = \sum_n o_n(x) \tag{10}$$

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 ϵ to the sum of Fourier components as shown in (11).

$$PC(x) = \frac{E(x)}{\epsilon + \sum_n A_n} \tag{11}$$

Equation (11) is the final expression to calculate phase congruency for a one dimensional signal. The computation of phase congruency features for a two-dimensional signal is as follows.

As in (8), the even symmetric and odd symmetric components at a particular scale n and orientation o can be computed as shown in (12).

$$[e_{no}(x, y), o_{no}(x, y)] = [I(x, y) * M_{no}^e, I(x, y) * M_{no}^o] \tag{12}$$

The amplitude A_{no} of the response at a particular scale and orientation can be computed as in (13). For an image, the calculation of phase congruency PC is as shown in (14).

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

$$PC(x, y) = \frac{\sum_o \sqrt{(\sum_n e_{no}(x, y))^2 + (\sum_n o_{no}(x, y))^2}}{\epsilon + \sum_o \sum_n A_{no}(x, y)} \tag{14}$$

3.3 Framework for Detection of Rain Streaks in Video

The initial framework developed by [Santhaseelan and Asari \(2011\)](#) for rain streak detection and removal is shown in Fig. 2.

The algorithm starts with the difference image computation for the three color components. Phase congruency features are calculated for the difference image, after which chromatic constraints are applied to get the candidate rain pixels. The detected streaks are compensated for using temporal neighbors that are not rain streaks.

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. In the first step (as shown in (15)), the difference image of the current frame with respect to its

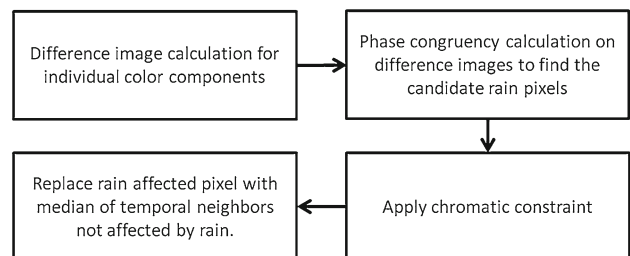


Fig. 2 Initial framework for rain streak detection and compensation

neighbors is computed. 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. Positive differences are preserved.

$$\Delta I(x, y, t) = \begin{cases} I(x, y, t) - I(x, y, t-1) & \text{if } I(x, y, t) > I(x, y, t-1) \\ 0 & \text{if } I(x, y, t) \leq I(x, y, t-1) \end{cases} \quad (15)$$

where $I(x, y, t)$ represents all the color components.

The next step is to find the phase congruency features of the difference image, ΔI for all the color components as in (16).

$$PC(x, y, t) = \mathcal{P}(\Delta I) \quad (16)$$

where the function $\mathcal{P}()$ represents the calculation of phase congruency features. Again, the features are calculated for each color component separately.

An illustration of the aforementioned steps is given in Fig. 3.

The final result has a large amount of directional structure due to the orientation selection in the log-Gabor filter. In this case, orientation of 90° was selected. For most cases of rain

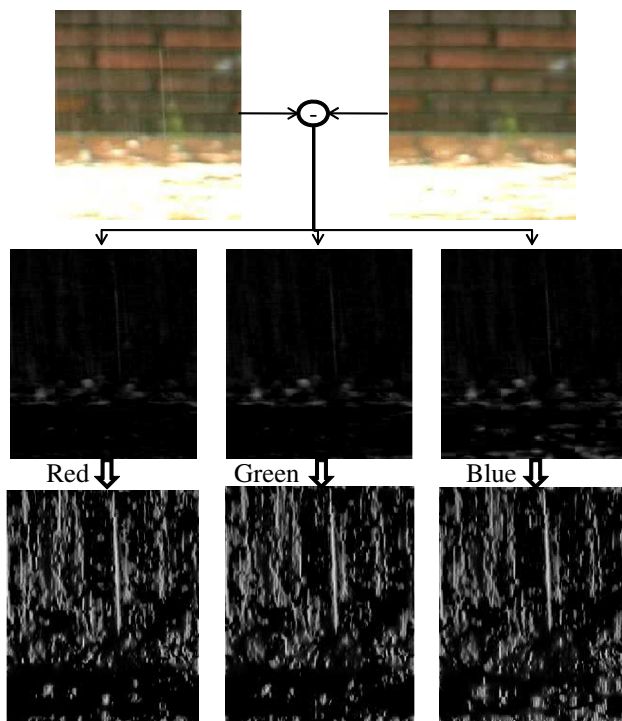


Fig. 3 Illustration of rain streak detection. *Top row* original frames with rain, *middle row* difference image for different color components, *bottom row* phase congruency features for the difference images

in videos, this orientation is sufficient to capture inter-frame variations. The variation in direction of rainfall is assumed to be minimal. However, there are scenarios where wind would cause the streaks to be oriented in directions other than downward. A variation of -45 to $+45$ degrees would be a good range. In order to compensate for such variations, the combination of phase congruency features from two more orientations can be considered during the detection of rain streaks.

After applying phase congruency, only the candidate pixels (rain affected pixels) with intensity variations in neighboring frames remain in the processed image. Chromatic property suggests that the change in all the three color components (*Red*(R), *Green*(G), *Blue*(B)) will be the same in terms of the strength of the phase congruency features when rain occludes a pixel. Since the change in all the three components will be the same when there is rain occluding a pixel according to the chromatic property, the following expression (17) provides another constraint to reduce false detections.

$$PC_{red}(x, y, t) \approx PC_{green}(x, y, t) \approx PC_{blue}(x, y, t) \quad (17)$$

where PC_{red} , PC_{green} and PC_{blue} represent the phase congruency features of the R , G , and B components respectively. The difference between the three PC components are assumed to be lesser than an empirical constant (0.02 for the experiments in this research). Pixels that do not satisfy the constraint in (17) are eliminated from the set of candidate rain pixels. The result of eliminating such false detections is illustrated in Fig. 4.

In the context of a more permissive rain streak detection model being used in conjunction with the reconstruction technique, this constraint is not very critical. However, when used with a naive reconstruction technique, the constraint seemed to provide better visual quality than without it as illustrated in [Santhaseelan and Asari \(2011\)](#).

In order to remove rain, the next challenge is to estimate the background intensity levels of the rain affected pixels. In

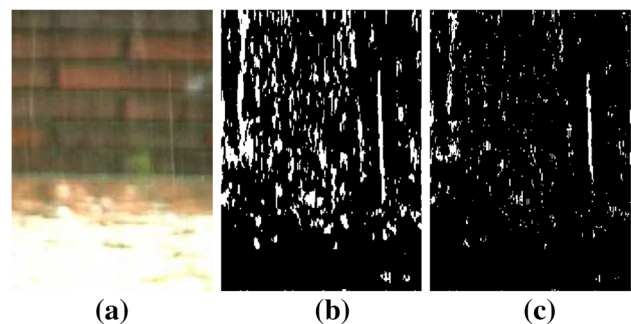


Fig. 4 Effect of applying constraint based on chromatic property. **a** original frame, **b** frame with candidate rain pixels, **c** frame with candidate rain pixels after removing false detections based on chromatic property

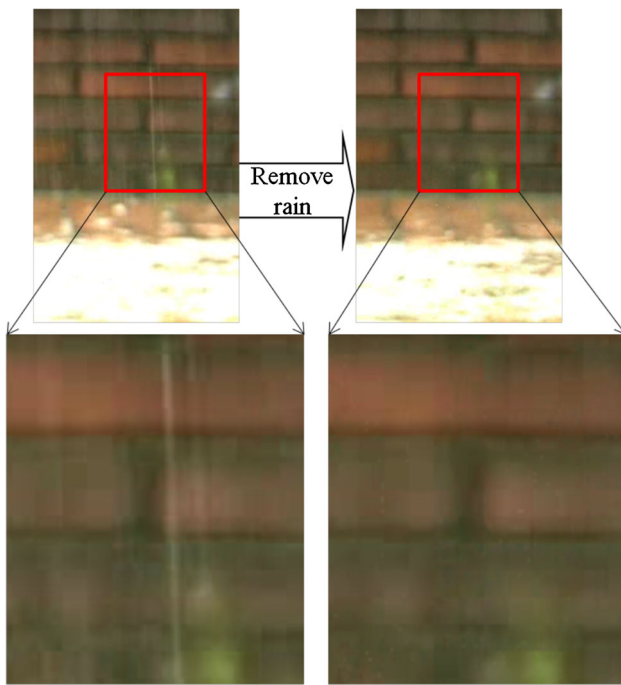


Fig. 5 Result of removing rain from a static video

order to verify the performance of the detection framework, a naive approach to scene reconstruction is adopted. A search is performed on the neighboring frames for a corresponding rain affected pixel. The value of the background intensity I_{bg} is estimated as the median of the temporal neighbors of the rain affected pixel.

Alpha-blending was used to calculate the intensity value for the rain affected pixel as shown in (18).

$$I_{no-rain} = \alpha I_{bg} + (1 - \alpha) I_{rain} \quad (18)$$

The new intensity is denoted as $I_{no-rain}$, the background intensity is denoted as I_{bg} and the intensity of the rain-affected pixel is denoted as I_{rain} . The global blending parameter is α , which is an empirical value that gave the best possible output in terms of visual quality. A sample frame where rain is removed is shown in Fig. 5.

It is observed in the resultant video that the proposed method is able to preserve dynamic components of the scene like variation in the pool of water. This can be attributed to the better selection of candidate rain pixels obtained by using phase congruency features.

3.4 Frame Alignment using Phase Correlation

One of the major challenges in order to detect the presence of rain is the movement of camera. During the computation of candidate rain pixels, global motion creates false detections. Therefore, it is essential that successive frames are

aligned before the computation of difference image. Phase correlation (Reddy and Chatterji (1996)) is the method used to align frames as it is resilient to noise to a very large extent. Phase correlation can be used to compute the translational shift in images from the phase information. Since most of the processing is done on videos with a frame rate of 30 fps, the global motion is approximated to be translational. Other kinds of movement is canceled out during the process of reconstruction explained in the next section.

When feature based techniques like Scale Invariant Feature Transform (SIFT) (Lowe (2004)) are used for stabilization, the feature points could be on the rain streaks. Such feature points cannot be matched reliably from frame to frame. This could cause stabilization procedures to fail, especially for videos containing heavy rain. Another option for stabilization could be using region based matching techniques for stabilization. However, it is likely that rain streaks closer to the camera could cause errors in matching between frames. This problem warrants the use of a stabilization technique that has high resilience to the presence of noise. In the case of phase correlation, the similarity is accounted for in terms of the global structure.

3.5 Modified Framework for Detection of Rain Streaks in Video

The initial algorithm (shown in Fig. 2) is modified to account for movement of camera. While the basic structure of the detection framework remains the same, a pre-processing step to stabilize video frames is added. Neighboring frames are always aligned with respect to the current frame being processed to detect rain streaks. The modified framework of the algorithm is shown in Fig. 6 (Santhaseelan and Asari (2012)).

The first step in the modified algorithm is to align neighboring frames with respect to the current frame in which rain streaks are to be detected. Then the difference image between various components is calculated. Phase congruency features are computed on the difference images. Chromatic constraints are applied to segment out the candidate rain pixels. These pixels are then compensated using information from the temporal neighbors that are aligned with the current frame as well.

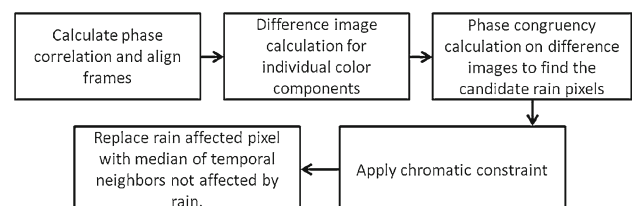


Fig. 6 Modified algorithm for detection of rain streaks with compensation for global motion

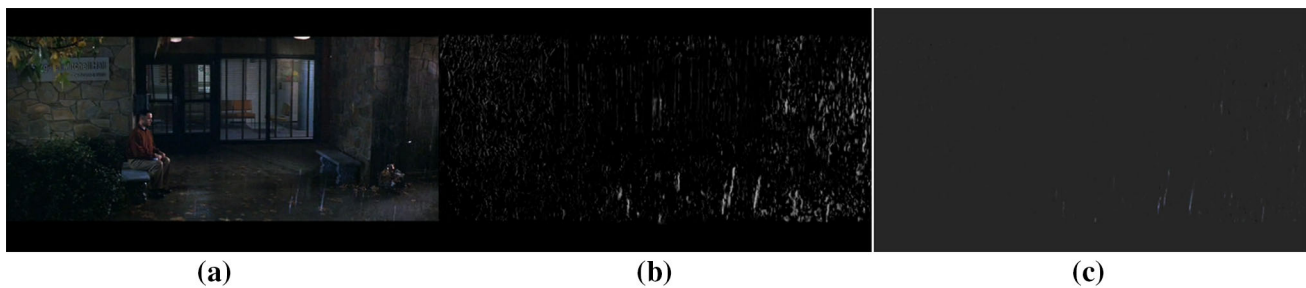


Fig. 7 Example frame from a video with camera movement and where rain streaks are detected. **a** original frame with rain, **b** frame with candidate rain streaks and **c** the final frame containing just the rain streaks

In the modified algorithm, the previous frame $I(x, y, t - 1)$ is aligned with respect to the current frame $I(x, y, t)$ and is modified into $I_a(x, y, t - 1)$ as in (19).

$$I_a(x, y, t - 1) = I(x + d_1, y + d_2, t - 1) \quad (19)$$

Thus the equation for finding difference between frames changes as shown in (20).

$$\Delta I(x, y, t) = \begin{cases} I(x, y, t) - I_a(x, y, t - 1) & \text{if } I(x, y, t) > I_a(x, y, t - 1) \\ 0 & \text{if } I(x, y, t) \leq I_a(x, y, t - 1) \end{cases} \quad (20)$$

The rest of the algorithm for detection remains the same as mentioned in the previous section. In order to compensate for the rain affected pixels, the neighboring frames are aligned with respect to the current frame before estimation of the intensity of replacement pixel. The computational expense for phase correlation is very less. The reason is that the technique consists of processing in the frequency domain and has two operations to compute Fourier Transform.

An example is shown in Fig. 7 to illustrate the effectiveness of the framework.

It can be observed that rain streaks are the only differences that exist after reconstruction. It can also be observed that there are no differences along vertical edges or along the outline of the person.

The algorithm for rain removal based on phase congruency has been found to be effective in situations where there are no moving objects in the scene. The presence of moving objects causes an increase in the number of false detections. Even though attempts have been made at eliminating the false detections based on local phase correlations, the quality of output video is poor as the result contains block effects. Even though the noise in individual frames appear to be diminished, temporal smoothness of the video is lost. This requires the design of a robust reconstruction algorithm that takes into account the effect of smaller streaks as well as maintaining the quality of video in terms of temporal smoothness.

4 Scene Reconstruction Based on Optical Flow of Local Phase

The observation that any part of the scene is not occluded by rain at all instances forms the basis of all reconstruction algorithms. In the case of spatial reconstruction, the assumption is that rain streaks are high frequency components in the image or frame. In some other cases, it is observed that by blending the intensity of the rain affected pixel along with the estimated actual scene intensity provides a good reconstruction of the original scene in terms of visual quality. In this research it has been observed that all the aforementioned statements hold true subject to certain constraints. This calls for the development of a technique for scene reconstruction that removes rain with minimal loss of information and maximum increase in video quality. The advantage of having a strong algorithm for reconstruction can allow for relaxation of constraints during rain streak detection.

There are three main sources of information to accurately estimate the intensity of the background (a) intensity of the rain affected pixel, (b) information from spatial neighbors, and (c) information from temporal neighbors. Within the constraints of the given sources of information, an optimized solution needs to be designed whereby the salient edges in the scene can be preserved while the temporal smoothness of video is not compromised.

4.1 Utilizing Pixel Information

As mentioned in the properties of rain, the presence of rain causes an increase in the intensity at a pixel location. Therefore, it is imperative that the estimated actual intensity of the scene does not exceed the intensity of pixel in the presence of rain which leads to the following guiding principle for the reconstruction of a scene to remove rain.

Optimization criteria 1: *The intensity of the actual background cannot be greater than the intensity of the pixel with rain occluding it.*

In mathematical terms the following expression (21) should hold good.

$$I_{rr}(x, y) \leq I(x, y) \tag{21}$$

where $I_{rr}(x, y)$ is the intensity in the reconstructed scene (scene with rain removed) and $I(x, y)$ is the original intensity of the pixel which is affected by rain.

This criterion was introduced to reduce chances of a false estimation of background intensity. Since the background intensity cannot be greater than the intensity of the pixel with rain, this would be a constraint that is easy to apply without too much computational effort. There could be instances where rain has to be removed in scenes containing a lot of textural background. The presence of textures can cause the number of false detections to increase. The aforementioned condition becomes crucial in such circumstances.

4.2 Utilizing Spatial and Temporal Information

While pixel information can be readily transformed into a constraint, the process is not trivial while using spatial and temporal neighbors. The effectiveness of both spatial and temporal information to remove rain is illustrated in Fig. 8.

From a variety of experiments, it is inferred that spatial and temporal information cannot be used independently to arrive at a feasible solution. The main challenge during the course of reconstruction is to avoid deterioration in video quality due to presence of local motion in the video. The loss of quality is particularly evident along the edges. However, it is observed that the regions that have a motion component associated with it are present in subsequent frames at a different location with the same intensity pattern. Therefore, those regions can be completely registered from frame to frame causing minimal registration error. However, the same is not the case for rain streaks. The pixels containing rain

cannot be registered accurately because of its rapid change in spatio-temporal domain. In light of the observations made, the following statement was set as the guiding principle in the design of the optimal solution.

Optimization criteria 2: *The intensity of replacement pixel should minimize the registration error with respect to the preceding frame containing no rain.*

In mathematical terms, the criteria can be used to estimate the intensity of the reconstructed scene $I_{rr}(x, y, t)$ as in (22).

$$I_{rr}(x, y, t) = \operatorname{argmin}_{I' \in Q} \{I' - I_{rr}(x - u, y - v, t - 1)\} \tag{22}$$

where $Q = \{I(x, y, t), I_k(x, y, t)\}$, (x, y) is the co-ordinates of a pixel in image space, t represents the time instant, $I(x, y, t)$ is the intensity of the pixel which is affected by rain, $I_k(x, y, t)$ is the estimate from the temporal neighbors. In this research, $I_k(x, y, t)$ is computed as the median of k temporal neighbors of the pixel at (x, y, t) . The optical flow velocity of the pixel is denoted by (u, v) . By incorporating registration between frames to compensate for the rain affected pixels, the temporal smoothness can be increased.

4.3 Key Observation

The main difference between rain streaks and the movement of objects is the continuity in movement. Rain streaks that appear in one frame are not present in the next frame. The edges or other features of the object remain in the frame, albeit at a different location. During reconstruction, a registration method is required that is resilient to the presence of rain streaks in order to apply the criterion mentioned in (22). In this scenario, consider a neighborhood around a rain affected pixel. It can be observed that the neighborhood is not completely occluded by the streak. If the streak is very near to the camera and covers a large area, it would appear blurred. In the case of sharp high intensity streaks, the breadth is comparatively lesser. Based on this idea, experiments were performed in trying to track local phase components from one frame to the next which led to the following key observation.

Observation: *The computation of optical flow velocity using local phase information is more resilient to the presence of rain streaks in comparison with velocities computed using intensity information.*

In intensity based optical flow, the robustness of the computation of velocities depend on the size of the window being considered. If the rain streaks are sufficiently large (closer to the camera), the computation of flow velocities will be error-prone since the intensity patch could match to an entirely different region containing rain. However, the same does not hold true for local phase based flow computation. During the computation of local phase, the dominant structure in a local

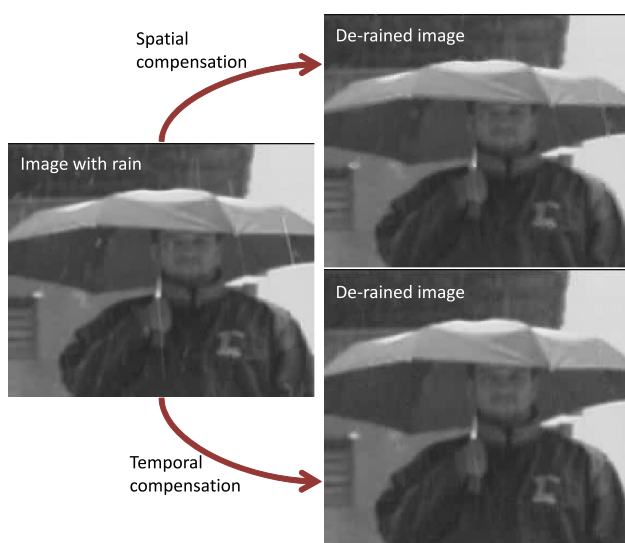
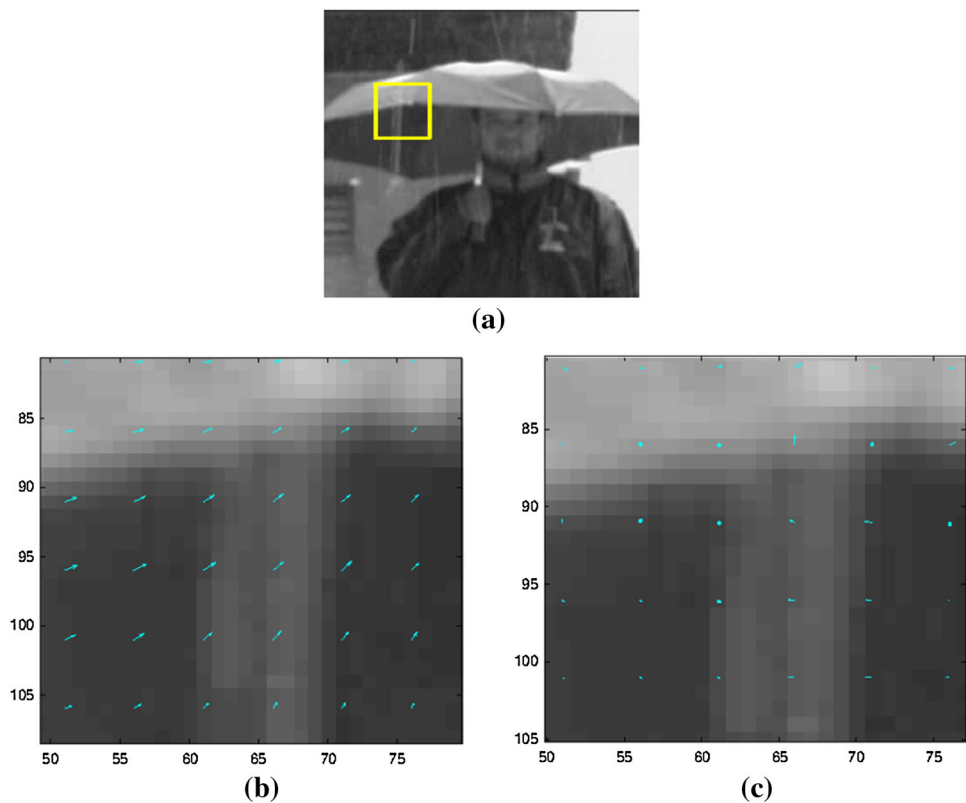


Fig. 8 Illustration of the effect of temporal and spatial compensation methods to remove rain

Fig. 9 Significance of phase based optical flow. **a** Original image with region of interest marked in yellow. **b** Optical flow components using phase based optical flow. **c** Optical flow vectors based on intensity



neighborhood is considered for the process. This neighborhood is defined indirectly by the parameters of the band pass filter. Therefore, when optical flow components are computed for a window of phase information, the net effect is that the local structure is being matched in its entirety with that of the corresponding frame. This would cause the optical flow components to be unaffected by the presence of rain. The difference in optical flow velocities using intensity and phase is illustrated as shown in Fig. 9.

It can be observed that the velocity patterns for phase based optical flow remain the same in the region of rain streak, while that of the intensity based optical flow is completely changed.

Local phase information can be computed in multiple ways, the most prominent method of which is the use of steerable wavelet filters. However, most of the methods are only capable of generating isotropic representation of phase. In order to extract phase information from images using an anisotropic model, the monogenic signal representation by Felsberg and Sommer (2001) is used.

4.4 Monogenic Signal Representation

The analytic signal model in signal processing had enabled the extraction of phase information from signals for the one dimensional case. The isotropic extension of the analytic signal model to multiple dimensions is called the monogenic signal (Felsberg and Sommer (2001)). The implicit assumption

is that the 2D signal is composed of i1D signals. An intrinsic one dimensional (i1D) signal is a signal that requires only one independent variable for its representation. Therefore, if a 2D signal were to be represented using a single i1D signal, the amplitude, phase and orientation of that i1D signal is assumed to be the local amplitude, local phase and local orientation of the 2D signal. Thus an image, $I(\mathbf{x})$ can be represented as shown in (23).

$$I(\mathbf{x}) = A(\mathbf{x}) \cos \varphi(\mathbf{x}) \quad (23)$$

where $\mathbf{x} = (x, y)$ is the spatial co-ordinates of the signal I , $A(\mathbf{x})$ is the local amplitude and $\varphi(\mathbf{x})$ is the local phase of the i1D signal. The i1D signal is oriented along the local orientation, $\theta(\mathbf{x})$.

In order to extract the i1D signal from the 2D signal, a band pass filter along with the Riesz Transform needs to be used. The combination of the band pass filter with the Riesz Transform is called the spherical quadrature filter (SQF). Local amplitude, local phase and local orientation are estimated from the response of even and a pair of odd spherical quadrature filters (SQF). The even SQF is the band pass filter, which in this research is a log-Gabor filter. In the frequency domain, the transfer function of the odd pair of SQF is computed as the product of a band-pass filter and a pair of Riesz kernels. The spatial and frequency domain representation of the pair of Riesz kernels is given in (24) and (25) respectively.

$$h_1(x, y) = \frac{x}{2\pi|\mathbf{x}|^3}, \quad h_2(x, y) = \frac{y}{2\pi|\mathbf{x}|^3}, \quad \mathbf{x} = (x, y) \in \mathbb{R}^2 \tag{24}$$

$$H_1(\omega_1, \omega_2) = -\frac{i\omega_1}{|\omega|}, \quad H_2(\omega_1, \omega_2) = -\frac{i\omega_2}{|\omega|}, \quad \omega = (\omega_1, \omega_2) \tag{25}$$

where (h_1, h_2) represents the pair of Riesz kernels in spatial domain with the corresponding frequency domain representations as (H_1, H_2) and $\omega = (\omega_1, \omega_2)$ are the frequency components.

The odd set of SQFs can then be represented as in (26).

$$\begin{aligned} G_{o1}(\omega) &= -\frac{i\omega_1}{|\omega|} G_e(\omega) \\ G_{o2}(\omega) &= -\frac{i\omega_2}{|\omega|} G_e(\omega) \end{aligned} \tag{26}$$

where $G_e(\omega)$ is the transfer function of the log Gabor filter given in (1).

In the spatial domain, the original signal $I(\mathbf{x})$ is convolved with the transfer function of even and odd pair of SQFs as shown in (27) to obtain the components of the monogenic signal representation $(f(\mathbf{x}), f_1(\mathbf{x}), f_2(\mathbf{x}))$.

$$\begin{aligned} f(\mathbf{x}) &= I(\mathbf{x}) * g_e(\mathbf{x}) \\ f_1(\mathbf{x}) &= I(\mathbf{x}) * g_{o1}(\mathbf{x}) \\ f_2(\mathbf{x}) &= I(\mathbf{x}) * g_{o2}(\mathbf{x}) \end{aligned} \tag{27}$$

where ‘*’ represents the 2D convolution, $g_e(\mathbf{x}), g_{o1}(\mathbf{x})$ and $g_{o2}(\mathbf{x})$ are the spatial domain representations of $G_e(\omega), G_{o1}(\omega)$ and $G_{o2}(\omega)$ respectively.

The local amplitude $A(\mathbf{x})$, local phase $\varphi(\mathbf{x})$ and local orientation $\theta(\mathbf{x})$ can then be computed as shown in (28), (29) and (30) respectively.

$$A(\mathbf{x}) = \sqrt{f^2(\mathbf{x}) + f_1^2(\mathbf{x}) + f_2^2(\mathbf{x})} \tag{28}$$

$$\varphi(\mathbf{x}) = \arctan\left(\frac{\sqrt{f_1^2(\mathbf{x}) + f_2^2(\mathbf{x})}}{f(\mathbf{x})}\right), \quad \varphi \in [0, \pi) \tag{29}$$

$$\theta(\mathbf{x}) = \arctan\left(\frac{f_2(\mathbf{x})}{f_1(\mathbf{x})}\right), \quad \theta \in [0, \pi) \tag{30}$$

The local phase along with the local orientation can be considered to be a vector with magnitude given by the local phase φ and an angle given by the local orientation θ . This vector is called the phase vector $\Phi(\mathbf{x})$ and can be represented as in (31).

$$\Phi(\mathbf{x}) = \varphi(\mathbf{x})\eta(\mathbf{x}) \tag{31}$$

where $\eta(\mathbf{x}) = [\cos(\theta(\mathbf{x})), \sin(\theta(\mathbf{x}))]^T$ is the unit vector along the angle given by local orientation, $\theta(\mathbf{x})$. The phase

vector $\Phi(\mathbf{x})$ can be represented in terms of its components as $\Phi(\mathbf{x}) = [\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x})]^T$.

In this research, the phase vector Φ is used to represent regions of the image. In the monogenic signal model, assumption is that a local image region is comprised of intrinsic 1D signals. Phase vector represents the phase and orientation of the intrinsic 1D signal in a local neighborhood. In terms of physical interpretation, the local phase captures the structural information of the object and the local orientation sheds light on the geometric information of the object. The contrast information in the image is given by the local amplitude.

Robust estimation of orientation: The estimation for local orientation could be affected by noise to a very large degree. Unser et al. (2009) proposed to have a least square estimate of the orientation based on the local neighborhood. The robust estimate is obtained by maximizing the directional Hilbert transform of the function over a neighborhood as represented by the optimization function in (32).

$$\bar{\theta}(\mathbf{x}) = \arg \max_{\theta \in [-\pi, \pi]} \int_{\mathbb{R}^2} v_\sigma(\mathbf{x}' - \mathbf{x}) |\mathcal{H}_\theta\{f(\mathbf{x}')\}| d\mathbf{x}' \tag{32}$$

where \mathbf{x}' is any pixel in the local neighborhood, v_σ is a Gaussian kernel and σ^2 is its variance, $\mathcal{H}_\theta(\cdot)$ is the directional Hilbert transform represented in the frequency domain as in (33).

$$\mathcal{H}_\theta(\omega) = \frac{\omega_1 \cos(\theta) + \omega_2 \sin(\theta)}{|\omega|} \tag{33}$$

where ω_1 and ω_2 are the components of angular frequency, ω .

4.5 Optical Flow using Phase Information

The optical flow technique has been adopted from the method by Alessandrini et al. (2013). The authors present a multi-scale computation technique based on the flow of phase vectors from one frame to the next. In classical optical flow computation, brightness of an image region is assumed to be constant. In this case, the phase vector in a particular region is assumed to be constant. One of the main advantages of using phase is to reduce the dependence on variation in intensity due to changes in illumination.

In mathematical terms, the phase constancy assumption can be expressed as in (34).

$$\Phi(\mathbf{x}, t + 1) = \Phi(\mathbf{x} - \mathbf{d}(\mathbf{x}), t) \tag{34}$$

where $\mathbf{d}(\mathbf{x}) = [u(\mathbf{x}), v(\mathbf{x})]$ represents the displacement made by the pixel at \mathbf{x} , and (u, v) is the optical flow velocity in x and y directions. If the displacement is assumed to be small,

then (34) can be approximated using the Taylor series expansion as in (35).

$$\Phi(\mathbf{x} - \mathbf{d}(\mathbf{x}), t) \approx \Phi(\mathbf{x}, t) - \mathbf{J}(\mathbf{x}, t)\mathbf{d}(\mathbf{x}) \quad (35)$$

where \mathbf{J} is the Jacobian matrix of Φ . If a local neighborhood around a point is defined, then the local displacement \mathbf{d} can be assumed to be similar for all the pixels in the neighborhood window, w . Using this assumption, a group of linear equations can be defined leading to the following expression (36).

$$\langle \mathbf{J} \rangle_w \mathbf{d} = -\langle \Phi_t \rangle_w, \quad (36)$$

$$\text{where } \mathbf{J}(\mathbf{x}, t) = \begin{bmatrix} \Phi_{1x}(\mathbf{x}, t) & \Phi_{1y}(\mathbf{x}, t) \\ \Phi_{2x}(\mathbf{x}, t) & \Phi_{2y}(\mathbf{x}, t) \end{bmatrix}$$

$$[\Phi_1, \Phi_2] = \varphi(\mathbf{x})[\cos(\theta), \sin(\theta)]$$

In (36), $\Phi_{1x} = \partial\Phi_1/\partial x$, $\Phi_{1y} = \partial\Phi_1/\partial y$, $\Phi_{2x} = \partial\Phi_2/\partial x$, and $\Phi_{2y} = \partial\Phi_2/\partial y$. Φ_t is the temporal derivative of Φ and can be expressed as in (37).

$$\Phi_t(\mathbf{x}, t) = \Phi(\mathbf{x}, t+1) - \Phi(\mathbf{x}, t) \quad (37)$$

Then the temporal derivative of the phase vector can be expressed in terms of SQFs as in (38), which was derived in Felsberg (2007).

$$\Phi_t = \frac{f_t f_{R_{t+1}} - f_{R_t} f_{t+1}}{|f_t f_{R_{t+1}} - f_{R_t} f_{t+1}|} \arctan\left(\frac{|f_t f_{R_{t+1}} - f_{R_t} f_{t+1}|}{f_t f_{t+1} - f_{R_t}^T f_{R_{t+1}}}\right) \quad (38)$$

where f_t is $f(\mathbf{x})$ at frame t , f_{R_t} is $f_R(\mathbf{x})$ at frame t , and $f_R(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$.

The aforementioned model for optical flow by Felsberg (2007) considered only translation of pixels. In Alessandrini et al. (2013), the case of affine flow was considered instead of a constant motion constraint that was dependent on the window size w . The affine model for a window w centered at $(x_0, y_0) = (0, 0)$ can be expressed as in (39).

$$\mathbf{d}(\mathbf{x}) = \mathbf{A}(\mathbf{x})\mathbf{u} \quad (39)$$

$$\text{where } \mathbf{A} = \begin{bmatrix} 1 & 0 & x & y & 0 & 0 \\ 0 & 1 & 0 & 0 & x & y \end{bmatrix}$$

$$\mathbf{u} = [d_{10}, d_{20}, d_{1x}, d_{1y}, d_{2x}, d_{2y}]^T$$

d_{10} and d_{20} is the displacement of the center of the window. The other components are partial derivatives as given by $d_{1x} = \partial d_1/\partial x$, $d_{1y} = \partial d_1/\partial y$, $d_{2x} = \partial d_2/\partial x$ and $d_{2y} = \partial d_2/\partial y$.

By combining (39) with (36) and multiplying with \mathbf{A}^T , we get the equation as in (40). Equation (40) is

the equivalent of the Lucas Kanade algorithm using monogenic phase vectors.

$$\langle \mathbf{M} \rangle_w \mathbf{u} = \langle \mathbf{b} \rangle_w \quad (40)$$

where $\mathbf{M} = \mathbf{A}^T \mathbf{J} \mathbf{A}$

$$\mathbf{b} = -\mathbf{A}^T \Phi_t$$

From further simplification of (40), the following expressions for \mathbf{M} and \mathbf{b} can be derived.

$$\mathbf{M} = \begin{bmatrix} \Phi_{1x} & \Phi_{1y} & x\Phi_{1x} & y\Phi_{1x} & x\Phi_{1y} & y\Phi_{1y} \\ \Phi_{2x} & \Phi_{2y} & x\Phi_{2x} & y\Phi_{2x} & x\Phi_{2y} & y\Phi_{2y} \\ x\Phi_{1x} & x\Phi_{1y} & x^2\Phi_{1x} & xy\Phi_{1x} & x^2\Phi_{1y} & xy\Phi_{1y} \\ y\Phi_{1x} & y\Phi_{1y} & xy\Phi_{1x} & y^2\Phi_{1x} & xy\Phi_{1y} & y^2\Phi_{1y} \\ x\Phi_{2x} & x\Phi_{2y} & x^2\Phi_{2x} & xy\Phi_{2x} & x^2\Phi_{2y} & xy\Phi_{2y} \\ y\Phi_{2x} & y\Phi_{2y} & xy\Phi_{2x} & y^2\Phi_{2x} & xy\Phi_{2y} & y^2\Phi_{2y} \end{bmatrix} \quad (41)$$

$$\mathbf{b} = -[\Phi_{1t} \ \Phi_{2t} \ x\Phi_{1t} \ x\Phi_{2t} \ y\Phi_{1t} \ y\Phi_{2t}]$$

where $\Phi_{1x} = \partial\Phi_1/\partial x$, $\Phi_{1y} = \partial\Phi_1/\partial y$, $\Phi_{2x} = \partial\Phi_2/\partial x$, $\Phi_{2y} = \partial\Phi_2/\partial y$, $\Phi_{1t} = \partial\Phi_1/\partial t$ and $\Phi_{2t} = \partial\Phi_2/\partial t$.

The next step is to decide on what would be an ideal size for the window w . In order to overcome the difficulties of too low or too high a window size, a multiscale approach for calculation of flow is followed. The basic idea is that the solution for \mathbf{u} is computed at multiple scales. The value \mathbf{u} for which a measure of residual error is minimum is considered to best estimate of optical flow. In order to compute a dense flow field, bicubic interpolation is used. The expression for residual error is given as in (42).

$$\text{Residual error} = \|\mathbf{M}\mathbf{u}^n - \mathbf{b}\|/|w| \quad (42)$$

where n represents the scale.

4.6 Algorithm for Scene Reconstruction

This section provides a detailed explanation of the algorithm for reconstruction. As mentioned earlier, the background intensity of any pixel in a frame can be estimated using the information from the spatial and temporal neighbors.

The aim of the reconstruction algorithm is to replace the rain affected pixel with the actual intensity of the background pixel. As mentioned earlier, the temporal neighbors provide very good estimate provided no motion is associated with the pixel. Therefore, the default replacement pixel is the temporal neighbor that does not contain rain. In this research, the median of k temporal neighbors is considered to be an optimal estimate.

The crucial part of the algorithm is to decide when a candidate pixel has to be replaced by the intensity from temporal neighbors. Here, the second optimization criteria based on

registering the current frame to the previous rain removed frame is utilized. The intensity of the reconstructed pixel is selected such that it reduces the difference with the registered pixel in the previous frame containing no rain. However, the pixel intensity remains unaltered if the intensity of the replacement pixel is greater than the intensity of the candidate rain pixel. That step is in accordance with the first optimization criteria. These steps are presented in Algorithm 1.

Algorithm 1 Algorithm for reconstruction

Initialization: Compute the median for first k frames.
for $t = k + 1$ to N **do**
 Compute optical flow (u, v) for every pixel in frame $I(x, y, t)$ to frame $I_{rr}(x, y, t - 1)$
for every candidate rain pixel at location (x, y) in the frame at time t **do**
 Find the registered pixel in previous rain removed frame,
 $I_{reg} = I_{rr}(x - u, y - v, t - 1)$
 Find registration error, $r = I(x, y, t) - I_{reg}$
if $r < 0$ **then**
 $I_{rr}(x, y, t) = I(x, y, t)$
else
 $I_{rr}(x, y, t) = I_k(x, y, t)$
end if
end for
end for

The algorithm starts with an initialization process. During the initialization, it is assumed that there is no local motion component associated with the video apart from rain streaks. Rain streaks are removed from the first k frames using a temporal median filter on the pixels that are detected to be part of rain streaks. Further processing is done on the remaining frames of video. The total number of frames is denoted as N . The next step is to compute the optical flow vector from the current frame $I(x, y, t)$ to the previous frame that does not contain rain. A pixel-wise registration error is computed. If the registration error is greater than zero, the pixel is assumed to be part of rain. The rain affected pixel is then replaced by the estimate of the pixel intensity from its temporal neighbors, $I_k(x, y, t)$. $I_k(x, y, t)$ is computed as the median of k temporal neighbors of the rain affected pixel. The size of the neighborhood forms a parameter for the algorithm. In the case of rapid changes in the scene, only one temporal neighbor can be considered for compensation. The rain removed intensity is denoted as $I_{rr}(x, y, t)$.

5 Experimental Results

The previous sections described the framework for detection of rain streaks and a method to reconstruct video to remove rain streaks. The end goal of this process has two perspectives: (1) increase the quality of video in terms of visual perception of the actual scene, and (2) increase the

effectiveness of other higher intelligence operations like object/face/pedestrian detection and recognition.

This section presents details of various experiments performed on different kinds of videos that contain rain. Videos from previously published research have been used to properly evaluate the performance of the proposed technique. The selected videos contain rain in varying complexities along with a variety of scene content.

5.1 Evaluation Strategies

The following techniques are adopted to verify the performance of the algorithm to remove rain.

1. **Qualitative analysis** - One of the main objectives of the algorithm to remove rain is to increase the quality of the video in terms of visual perception. This would demand some constraints on the output like no edge artifacts and better temporal smoothness. This is based on the visual assessment of the reconstructed video with respect to the original video with rain.
2. **Quantitative analysis** - The main technique for quantitative evaluation is the computation of no-reference image quality measure on the videos where rain is removed. A quality measure based on natural scene statistics called Blind Image Quality Index (BIQI) by [Moorthy and Bovik \(2010\)](#) is used in this research. The method uses wavelet analysis to extract features from the image, try to detect the kind of distortion present in the image using a trained classifier and then predict the distortion score using support vector regression. The prediction is based on the images that have been used to train the system. In the case of BIQI, a higher score indicates more distortion and therefore, a lower quality. The final score for image quality is on a scale of 0-100 with 100 indicating an image with the most distortion. In the context of removing rain from video, the reconstruction process should be capable of reducing spatial distortions.

In the work by [Barnum et al. \(2010\)](#), rain streaks are considered to be a two dimensional Gaussians. It was illustrated in the work about how the assumption would be a good basis to detect and remove rain streaks in video. Following the same assumption, it can be assumed that the presence of rain streaks cause Gaussian blurring on images. The amount of Gaussian blur can be estimated using BIQI. However, the technique would be a good measure of the quality of reconstructed video only in the presence of a large number of rain streaks.

In some cases, advanced image analysis algorithms can also be used for evaluation. In one of the videos, a person is walking towards the camera in rain. In this case, a face detection algorithm by [Viola and Jones \(2004\)](#) is employed to evaluate the performance of the algorithm.

5.2 Results and Discussion

This section provides experimental results along with some discussion. The performance of the proposed framework to remove rain is compared with the results of other state of the art research present in literature.

5.2.1 Removing Rain from Static Video

Static videos are videos where the only dynamic component in a scene is rain. Since the intensity variations are completely due to rain, the difference image computation and phase congruency calculations result in all the rain streaks in the frame. In this section, video provided by Zhang et al. (2006) is used for experimental evaluation. A qualitative comparison of the result using the proposed algorithm with the results of Zhang et al. (2006) is shown in Fig. 10.

It can be observed from Fig. 10 that the result using the proposed technique is similar to the result obtained by Zhang et al. (2006).

In terms of visual quality assessment by manual inspection of the reconstructed video, it was observed that the result by Zhang et al. (2006) tends to have lesser dynamic content in comparison with the output of the proposed technique. Thus the visual quality is higher for the output of Zhang et al. (2006). This difference can be attributed to the number of frames that are used to reconstruct a scene. In the method by Zhang et al. (2006), presence of rain streaks is detected from the construction of a temporal histogram for every pixel in the scene. The histogram is constructed for all the frames in the video. The intensity for scene reconstruction is also estimated from the histogram. The proposed technique uses only five neighboring frames for reconstruction. It is also observed that the proposed algorithm removes all the significant streaks with lesser number of frames for reconstruction.

The results of the proposed technique is compared with that of Zhang et al. (2006) quantitatively in terms of the dis-

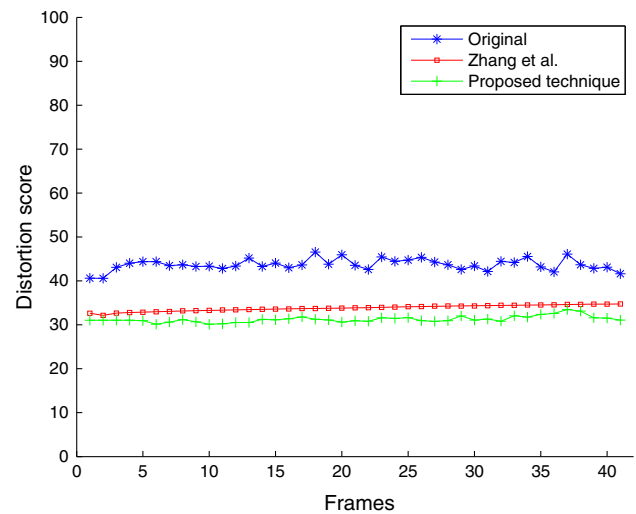


Fig. 11 Quantitative comparison for static videos containing rain. Comparison of quality measures of reconstructed video from proposed method with the output of method by Zhang et al. (2006)

tortion score for individual frames. The comparison is shown in Fig. 11.

The distortion score of the output of Zhang et al. (2006) is more than that of the proposed technique. This is due to the fact that the reconstructed scene from Zhang et al. (2006) suffers from blur induced by the rain streak removal process. The variation in distortion score of the output of Zhang et al. (2006) is very less. This can be attributed to the large number of frames used for reconstruction. The proposed technique used five temporal neighbors to remove rain from the video.

5.2.2 Removing Rain from Video with Dynamic Texture

The effectiveness of rain streak detection can be evaluated effectively when the algorithm is tested on video with dynamic textures. In this test, video provided by Garg and Nayar (2004) is used, in which the scene consists of rain falling on a pool. Rainfall causes variations in the pool that are

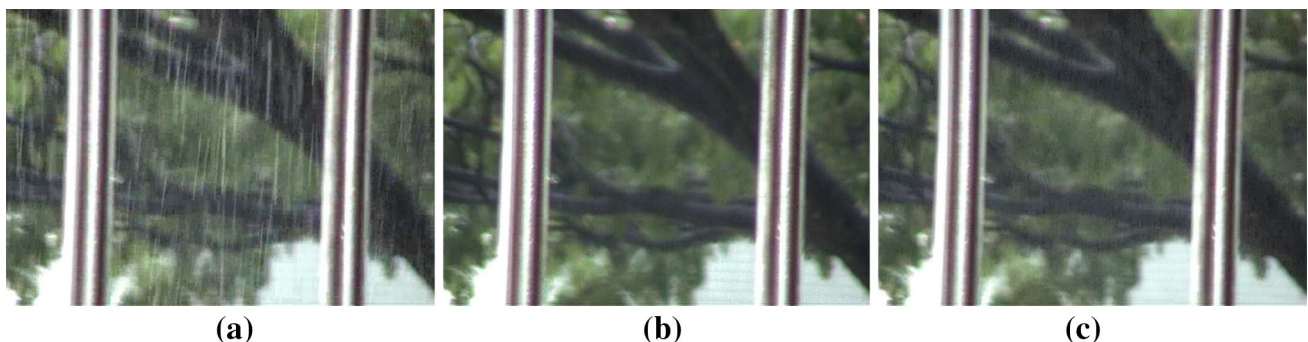


Fig. 10 Qualitative comparison for static video containing rain: **a** original frame with rain, **b** frame with rain removed using the method by Zhang et al. (2006), **c** frame with rain removed using proposed technique

to be segmented out from the rain streaks detected, thereby preserving the characteristics of the dynamic texture in video, which is the appearance of the pool. A qualitative comparison of the output of the proposed technique with the output of Garg and Nayar (2004) is shown in Fig. 12.

On visual examination of the resultant videos, it was observed that the output of the proposed technique is as good as the output from Garg and Nayar (2004) in preserving the dynamic content in the scene while removing rain streaks from the video.

A quantitative comparison of algorithms in terms of the distortion score to measure image quality is shown in Fig. 13. In this experiment, quality assessment is performed on the region containing rain streaks and not the pool region. It can be observed that the performance of the proposed technique is better for most frames in comparison with the technique by Garg and Nayar (2004). The method by Garg and Nayar

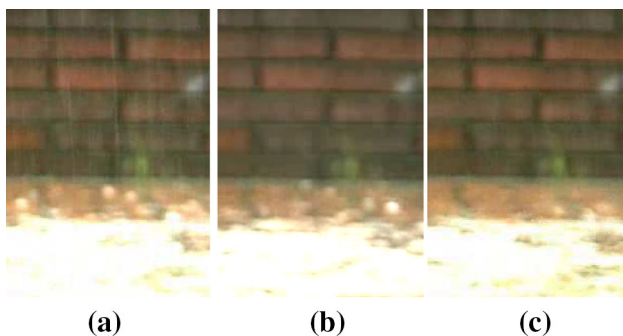


Fig. 12 Qualitative comparison for videos with dynamic texture: **a** original frame with rain, **b** frame with rain removed using method by Garg and Nayar (2004), **c** frame with rain removed using proposed technique

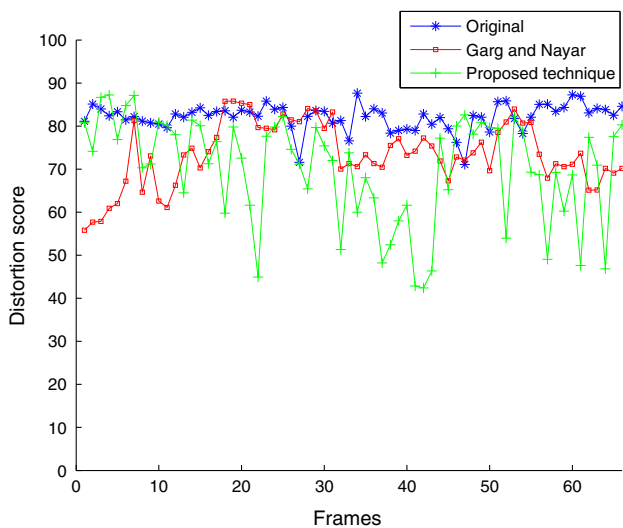


Fig. 13 Quantitative comparison for videos with dynamic texture. Comparison of quality measure of reconstructed video from proposed method with the output of method by Garg and Nayar (2004)

(2004) reduces the rain content in the scene. However, it is not successful in removing the dynamic content due to rain entirely. This could be due to the strict photometric constraints that are applied in the procedure for detection. The significant dips in distortion score for the result of proposed technique is due to the absence of any significant rain component in that particular frame or in its neighborhood.

5.2.3 Removing Rain from a Video Containing Global Motion

The presence of global motion component increases the complexity of the processing to remove rain. The proposed technique consists of a step to determine the difference between consecutive frames. In order to prevent an increase in incorrect rain affect pixels, phase correlation is used to align the frames. The technique for phase correlation mentioned earlier assumes that the movement of camera from one frame to the next is purely translational. This assumption might not be true for all videos. However, the reconstruction technique is robust enough to handle such variations. The experiment presented in this sub-section verifies the robustness of the algorithm to remove rain when camera is moving. The key challenge of the reconstruction algorithm would be to preserve edge information.

In this set of experiments, a video provided by Barnum et al. (2010) is used. The video is of a man sitting in rain. Every frame contains vertical edges that need to be preserved along with the outline of the person. A sample frame where rain is removed is shown in Fig. 14.

In the magnified section of the reconstructed frame, it can be observed that the rain streaks are absent. It can also be observed in the complete image that the vertical edges on the building and the outline of the person are completely preserved in the reconstructed frame when compared with the frame from original video.

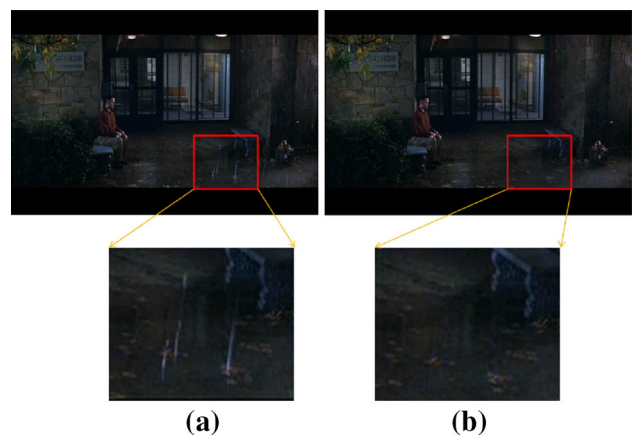


Fig. 14 Sample frame where rain is removed from video when camera is moving. **a** The original frame with rain, **b** the frame with rain removed



Fig. 15 Qualitative comparison for video with global motion. **a** Original frame with rain, **b** frame with rain removed by Barnum et al. (2010), **c** frame with rain removed using proposed technique

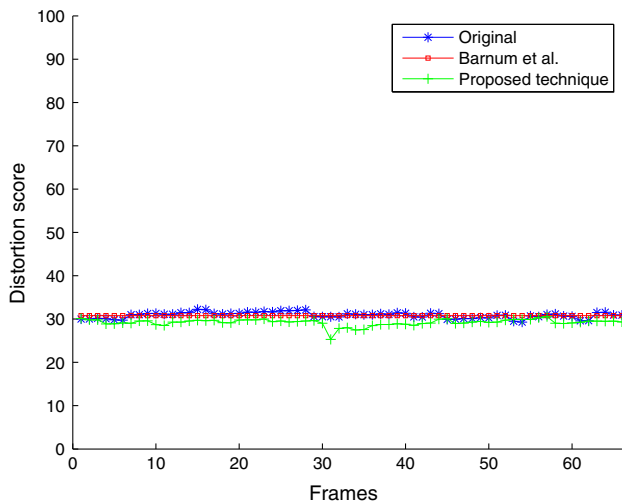


Fig. 16 Quantitative comparison on video with global motion. Comparison of quality measure of reconstructed video from proposed method with the quality measure of the original video with rain and the result by Barnum et al. (2010)

The output from the proposed technique is compared with the output from Barnum et al. (2010) in Fig. 15.

It can be observed that the results of the method by Barnum et al. (2010) and that of the proposed technique are similar in terms of visual quality. However, it was observed that the amount of rain that was removed by the method of Barnum et al. (2010) was lesser in comparison with the result of the proposed technique. This is due to the limitation of the frequency based model by Barnum et al. (2010) to detect streaks that cause lesser intensity variations.

Quantitative analysis is performed on the reconstructed video based on the distortion score of individual frames. The results are shown in Fig. 16.

The complete reconstructed video by Barnum et al. (2010) was not available. Therefore, the distortion score is calculated for the available frame and that value is assumed to be a mean representation of all the frames in the reconstructed video. Since there are no significant changes in the scene content and

the duration of video considered for evaluation is very less, this assumption is considered to be feasible. It is observed that the output of the proposed technique causes a decrease in distortion. The distortion score of the result by Barnum et al. (2010) tends to remain the same as that of the original video. One of the main reasons is that the amount of rain in the frame is very less. Therefore, the changes in the scene are not significant enough to cause a large variation in the distortion score.

5.2.4 Removing Rain from Video Containing Moving Objects

The most challenging part of removing rain is in the case of videos that contain objects which move between frames. The presence of local motion component is not canceled during alignment of frame using phase correlation. Such local motion causes an increase in false detections along the edges of the objects that move. Since the framework for detection of rain streaks did not address the problem of local motion, this experiment is a verification of the effectiveness of the algorithm for reconstruction. The video provided by Garg and Nayar (2004) is used for testing and analysis in this experiment. The video consists of a person, holding an umbrella, moving towards the camera in rain. There are multiple frames where the face of the person is occluded by rain streaks. There are two objects with motion components in this scene - the person and the umbrella. Therefore, edges with varying velocities are present in the same scene and need to be preserved during reconstruction.

A qualitative comparison of the results using the proposed technique with the original video and the reconstructed video using the method from Garg and Nayar (2004) is shown in Fig. 17.

It can be observed that the method in Garg and Nayar (2004) removes rain without causing artifacts along edges. It can also be observed that the proposed technique preserves the edges as well while removing all the rain streaks.

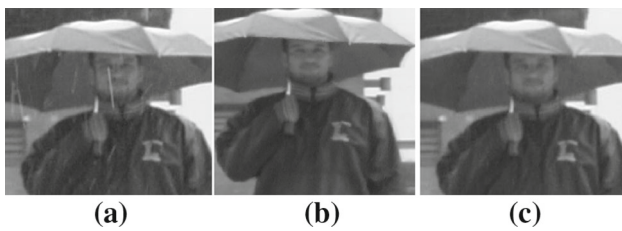


Fig. 17 Qualitative comparison for video containing moving objects. **a** original frame with rain, **b** frame with rain removed in Garg and Nayar (2004), **c** frame with rain removed using proposed technique

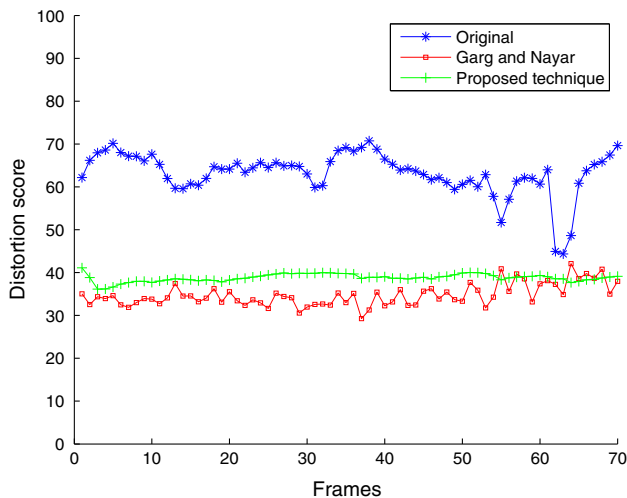


Fig. 18 Quantitative comparison on video with moving objects in the scene. Comparison of image quality measure of the reconstructed video using proposed technique with that of the original video and result of Garg and Nayar (2004)

In terms of quantitative analysis, the measures of image quality are compared for the reconstructed video using proposed technique, the original video and the result by Garg and Nayar (2004). The comparison is shown in Fig. 18.

It can be observed that the proposed technique reconstructed a video of much better quality than the original video. The quality in terms of the distortion score of the reconstructed video is not as good as the result by Garg and Nayar (2004). The reason for the better performance of the method by Garg and Nayar (2004) is that there is not a lot of depth in the scene. The technique by Garg and Nayar (2004) is robust when the streaks are completely in focus and narrow and there is not a lot of depth in the scene, which is the case in the video used for testing.

An automatic algorithm to detect faces by Viola and Jones (2004) is applied on the reconstructed video to further analyze the performance of the proposed algorithm. It needed to be verified whether the lesser quality in comparison with Garg and Nayar (2004) was affecting the performance of feature extraction for advanced image analysis. A sample frame

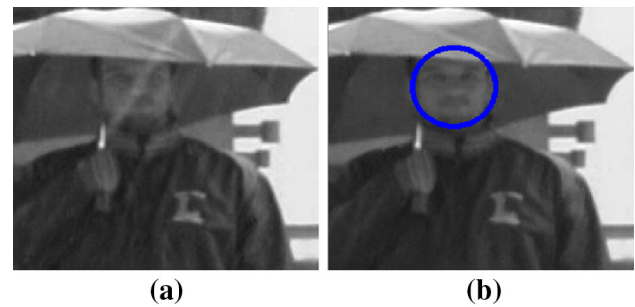


Fig. 19 Improvement in face detection when rain is removed: **a** result of detection on original frame with rain, **b** result of detection when rain is removed using proposed algorithm

Table 1 Quantitative evaluation of rain removal based on face detection

	Original video	Result using Garg and Nayar (2004)	Result using proposed algorithm
No. of frames in which face is detected	82	90	90

Comparison using the performance of algorithm for detecting faces on the video with moving person. The numbers are obtained for zero false detections

showing the result of algorithm for face detection is shown in Fig. 19.

Figure 19a shows the result of the algorithm for face detection on a frame from original video where a rain streak obstructs the view of the face. It can be observed that the algorithm fails to detect the face. In Fig. 19b, the result of application of the algorithm for face detection on the reconstructed frame using the proposed technique is shown. It is observed that the face is detected, mainly because the rain streak has been completely removed.

The algorithm for face detection was applied on the resultant video from Garg and Nayar (2004) as well. The combined results are tabulated in Table 1. The number of frames where face is detected is more for the video with rain removed than the original video with rain. The number of detections of face is the same for the reconstructed video using proposed algorithm and the rain removal algorithm from Garg and Nayar (2004). This proves that even though the proposed algorithm has lesser image quality than the method by Garg and Nayar (2004) in terms of the distortion score, the proposed technique is efficient in retaining all the necessary features for advanced image analysis.

5.2.5 More Qualitative Results

Apart from the scenarios mentioned before, a qualitative analysis was done on some other random videos as well. One of the examples is shown in Fig. 20. The video is

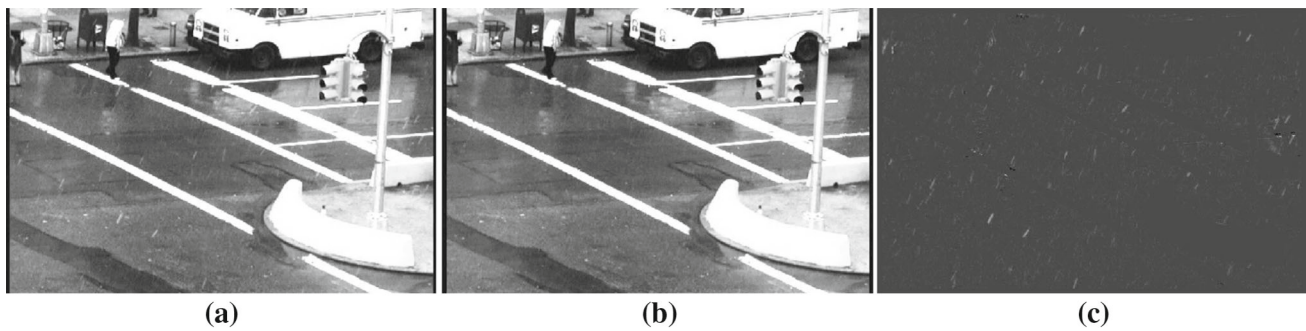


Fig. 20 Removing rain on a scene with moving pedestrian: **a** original frame with rain, **b** frame with rain removed using proposed algorithm, and **c** rain component in the scene that was removed

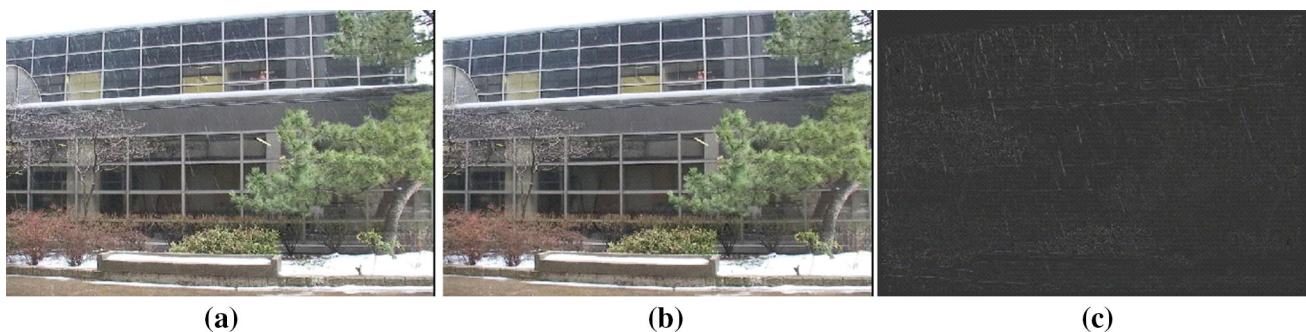


Fig. 21 Removing rain on a scene with vertical edges: **a** original frame with rain, **b** frame with rain removed using proposed algorithm, and **c** rain component in the scene that was removed

of a street with a pedestrian moving and rain is present. The challenge is to preserve the shape of the pedestrian while removing rain from the video. From the removed rain component, it is observed that no part of the pedestrian is altered.

Another example is of a video containing rain and the camera capturing the video is moving considerably. Vertical edges of high intensity are present in the frame that makes the rain removal process particularly challenging. The result of removing rain on the video is shown in Fig. 21.

In the scene with rain removed, there is no distortion to the vertical edges in the scene. It is also observed that rain streaks have been completely removed from the frame as well.

6 Conclusion

In this paper, a novel framework to remove rain from videos was designed. The framework consisted of two parts—(1) framework for detection based on phase congruency features and (2) reconstruction of scene using optical flow estimation from local phase information. The effectiveness of the entire system was tested by applying the same on videos containing rain with varying complexities. The underlying ideas, for the algorithm to detect rain streaks, are derived from the spatio-temporal and chromatic properties of rain.

The framework to detect the location of rain streaks in frame starts with the computation of difference between two consecutive frames. This results in three difference images corresponding to the red, green and blue components. Phase congruency features are extracted from the difference images to get a set of candidate rain pixel. The directional property of rain is incorporated in the design of the feature extraction process. Chromatic constraint based on the strength of phase congruency features is applied to eliminate false detections. In order to check the effectiveness of the proposed framework for rain streak detection, temporal neighbors were used to compensate for the rain affected pixels and the scene was reconstructed. It was observed that the framework was successful in removing rain streaks where there is no motion component associated with the scene, apart from the rain streaks. The framework was modified to eliminate the presence of false detections arising due to global motion like movement of the camera. Phase correlation was used as the technique to align frames prior to the process of detection of streaks. The second part of this research focused on reconstructing the scene after detecting rain streaks. The main challenge was to make the technique robust enough to account for movement of objects present in the scene. A key observation was made that optical flow from phase information is mostly resilient to the presence of rain streaks. Based on this idea, optimization criteria were designed for robust reconstruc-

tion of the scene. The minimization function was the registration error between neighboring frames. Detailed comparisons were made with state of the art techniques qualitatively and quantitatively. The proposed technique was found to be as good as the current state of the art techniques and in some cases, even better. Quantitative evaluation was done using a no-reference image quality index called Blind Image Quality Index (BIQI).

The algorithm is being improved to include large displacement optical flow for better reconstruction of videos containing large and rapid movements. Techniques for improving the temporal consistency are also being designed. Going into the future, it would be of great value if the algorithm could be made to perform in real-time. In the age of autonomous navigation, it would be really beneficial for post-processing algorithms if the video input is devoid of rain. The current algorithm would have to be ported to a suitable implementation platform for enhanced performance.

References

- Alessandrini, M., Basarab, A., Liebgott, H., & Bernard, O. (2013). Myocardial motion estimation from medical images using the monogenic signal. *IEEE Transactions on Image Processing*, 22(3), 1084–1095.
- Barnum, P. C., Narasimhan, S., & Kanade, T. (2010). Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 86(2–3), 256–274.
- Bossu, J., Hautière, N., & Tarel, J. P. (2011). Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International Journal of Comput Vision*, 93(3), 348–367.
- Brewer, N. (2008). Using the shape characteristics of rain to identify and remove rain from video. *Structural, syntactic, and statistical pattern recognition* (pp. 451–458). Berlin: Springer.
- Felsberg, M. (2007). Optical flow estimation from monogenic phase. *Complex motion* (pp. 1–13). Berlin: Springer.
- Felsberg, M., & Sommer, G. (2001). The monogenic signal. *IEEE Transactions on Signal Processing*, 49(12), 3136–3144. doi:10.1109/78.969520.
- Field, D. J., et al. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *The Journal of the Optical Society of America A*, 4(12), 2379–2394.
- Garg, K., & Nayar, S. (2004). Detection and removal of rain from videos. In *CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004*, (vol. 1, pp. 1-528–I-535).
- Garg, K., & Nayar, S. (2007). Vision and rain. *International Journal of Computer Vision*, 75(1), 3–27.
- Garg, K., & Nayar, S. K. (2003). *Photometric model of a rain drop*. Technical Report, New York: Columbia University.
- Hase, H., Miyake, K., & Yoneda, M. (1999). Real-time snowfall noise elimination. In *Proceedings of International Conference on Image Processing, ICIP 99* (vol. 2, pp. 406–409).
- Huang, D. A., Kang, L. W., Yang, M. C., Lin, C. W., & Wang, Y. C. (2012). Context-aware single image rain removal. In *2012 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 164–169).
- Kang, L. W., Lin, C. W., & Fu, Y. H. (2012a). Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4), 1742–1755.
- Kang, L. W., Lin, C. W., Lin, C. T. & Lin, Y. C. (2012b). Self-learning-based rain streak removal for image/video. In *2012 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1871–1874).
- Kim, J. H., Lee, C., Sim, J. Y., & Kim, C. S. (2013). Single-image deraining using an adaptive nonlocal means filter. In *2013 20th IEEE International Conference on Image Processing (ICIP)* (pp. 914–917).
- Kovesi, P. (1999). Image features from phase congruency. *VIDERE: Journal of Computer Vision Research*, 1(3), 1–26.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110.
- Mechler, F., Reich, D. S., & Victor, J. D. (2002). Detection and discrimination of relative spatial phase by v1 neurons. *The Journal of Neuroscience*, 22, 6129–6157.
- Moorthy, A. K., & Bovik, A. C. (2010). A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5), 513–516.
- Morrone, M., & Owens, R. (1987). Feature detection from local energy. *Pattern Recognition Letters*, 6(5), 303–313.
- Narasimhan, S., & Nayar, S. (2002). Vision and the atmosphere. *International Journal of Computer Vision*, 48(3), 233–254.
- Oppenheim, A. V., & Lim, J. S. (1981). The importance of phase in signals. *Proceedings of the IEEE*, 69(5), 529–541.
- Park, W. J., & Lee, K. H. (2008). Rain removal using kalman filter in video. In *International Conference on Smart Manufacturing Application, ICSMA 2008* (pp. 494–497).
- Reddy, B., & Chatterji, B. N. (1996). An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8), 1266–1271.
- Santhaseelan, V., & Asari, V. K. (2011). Phase congruency based technique for the removal of rain from video. *Image analysis and recognition* (pp. 30–39). Berlin: Springer.
- Santhaseelan, V., & Asari, V. K. (2012). A phase space approach for detection and removal of rain in video. In *IS&T/SPIE Electronic Imaging*, (pp. 830,114–830,114).
- Shen, M., Xue, P. (2011). A fast algorithm for rain detection and removal from videos. In *2011 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1–6).
- Starik, S., & Werman, M. (2003). Simulation of rain in videos. *ICCV texture workshop*, 2, 406–409.
- Tripathi, A., & Mukhopadhyay, S. (2011). A probabilistic approach for detection and removal of rain from videos. *IETE Journal of Research*, 57(1), 82–91.
- Tripathi, A., & Mukhopadhyay, S. (2012a). Removal of rain from videos: A review. *Signal, Image and Video Processing*, 1(8), 1–10.
- Tripathi, A., & Mukhopadhyay, S. (2012b). Video post processing: low-latency spatiotemporal approach for detection and removal of rain. *IET Image Processing*, 6(2), 181–196.
- Unser, M., Sage, D., & Ville, D. V. D. (2009). Multiresolution monogenic signal analysis using the riesz-laplace wavelet transform. *IEEE Transactions on Image Processing*, 18(11), 2402–2418.
- Venkatesh, S., & Owens, R. (1989). An energy feature detection scheme. In *IEEE International Conference on Image Processing: Conference Proceedings ICIP'89*, Sep 5–8 1989, Singapore: IEEE.
- Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2), 137–154.
- Wadhwa, N., Rubinstein, M., Durand, F. & Freeman, W. T. (2013). Phase-based video motion processing. In *Proceedings SIGGRAPH on ACM Transactions Graph* (vol. 32(4), pp. 80).
- Xue, X., Jin, X., Zhang, C. & Goto, S. (2012). Motion robust rain detection and removal from videos. In *2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSp)* (pp. 170–174).
- Zhang, X., Li, H., Qi, Y., Leow, W. K. & Ng, T. K. (2006). Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE International Conference on Multimedia and Expo* (pp. 461–464).