Rain Removal from Videos using the Temporal-Spatial Statistical Properties

Robin Kalia
Electronics and Telecommunications Research Institute (ETRI),
138 Gajeongdong, Yuseong-gu
305 700, Daejeon, South Korea
robinkalia@etri.re.kr

Amol Jaikar
Electronics and Telecommunications Research Institute (ETRI),
138 Gajeongdong, Yuseong-gu
305 700, Daejeon, South Korea
amol@etri.re.kr

ABSTRACT

Detection and removal of rain streaks from videos has recently become a great and challenging topic of research. This paper discusses a new technique for the removal of rain from videos using the temporal-spatial statistical properties. For this the temporal statistical properties of the pixels affected by rain are made use of, and then an efficient and easy algorithm is implemented which takes care of the effective removal of rain from videos. This technique works very well for videos with still and moving backgrounds involving moving objects with a fixed camera position. For the videos which involve the motion of the camera, the technique works well for a small rate of change of background in the camera frames. Our algorithm does not use variable and conditional parameters like the shape, size, velocity, and spatio-temporal physical model of raindrops, and camera's parameters like the aperture, focal length, and exposure time. The test results quantitatively and qualitatively illustrate that the performance of our algorithm is quite efficient in comparison to the previously existing algorithms which are state of the art techniques used for the purpose of removing rain from videos.

Keywords

Rain Detection, Rain Removal, Image Restoration, Temporal Properties, Pixel Occlusion, Spatial Filtering, Outdoor Vision and Weather.

1. INTRODUCTION

In the present day scenarios, we need to perform real time image processing and computer vision operations on real world objects. However, we have to deal with a large amount of interference and noise effects in the images of real world objects. The most important and prominent effect is that of the weather. The weather effects cause a lot of irritation to human viewers, and also affect the performance of vision algorithms for carrying out tasks like object detection, object recognition, tracking and image segmentation. Our project is based on Content Based Image Retrieval for Photo Automatic Sorting System. We have to carry out tasks like face detection, face recognition, image registration, general object detection and recognition, and key frames extraction in the images and videos of real world objects, which could vary from people to landscapes and architecture. In this case we use small features which operate on the images.

There are two kinds of outdoor weather conditions that we have to deal with: static weather conditions (fog, haze) and dynamic weather conditions (rain and snowfall). The dynamic effects of weather conditions, like the blurring and intensity altering effects of rain streaks on large portion of the images, affect the efficiency of these algorithms. Any vision algorithm which uses small features will be seriously affected by the disruptive effects of the weather conditions. Hence, we aim to reduce these disruptive effects of weather like motion blurring, and restore the image to its original form, to carry out our task of Content Based Image Retrieval with a good performance. In this paper we deal specifically with the dynamic weather conditions involving the removal of rain streaks from videos.

1.1 Related Work

Starik and Werman approached this problem by trying out temporal median filtering on pixels [Starik03a]. The problem with their developed method is that it works in the case of moderate rain conditions on clear day scenes. However, in the case of heavy rain and poor contrast scenarios, their method causes unnecessary blurring of other details in the images while retaining the blurring effects of rain considerably. Figure 1(a) shows the effect of using this approach on a scene with appropriate contrast and heavy rain conditions, and Figure 1(b) illustrates the effect of using this method on a scene with poor contrast and heavy rain conditions. It can be seen that the image becomes a little blurred at the edges, and the rain streaks are still quite clearly evident after the temporal median filtering. Garg and Nayar have tried to remove rain using the spatial and photometric properties [Garg04a]. Their method does not work very well in the case of videos involving heavy rain. They also removed rain from videos for certain conditions by adjusting the camcorder's parameters like the exposure time and aperture [Garg05a]. However, this case does not apply very well in the case of heavy rain conditions. Zhang utilized the temporal and chromatic properties to remove rain [Zhang06a]. This method does not perform real time processing, and the chromatic property they use depends on the experimental frames. Barnum et al use blurred Gaussian to approximate a rain streak for the blurring that it causes [Barnum07a]. This can work for clear rain scenes but in the heavy rain case, a blurred Gaussian is not effectively appropriate to segment rain streaks.

Zhao and Liu implemented the histogram model to detect rain in videos [Zhao08a]. Their method uses K-means clustering, and its effectiveness is appropriate only for videos of stationary scenes taken with a fixed camera position. Park and Lee have used the Kalman filtering method to estimate the intensity of the rain affected pixels [Park08a]. This method performs real time processing of videos but it works only for a fixed camera position and still background which is not practical in real applications. Brewer and Liu model rain streaks based on the shape, velocity, and aspect ratio of rain drops [Brewer08a]. However, the aspect ratio depends heavily upon the camera's exposure time, and for a video with unknown exposure time and heavy rain, the segmentation of rain and non-rain regions is not much effective, and their algorithm does not work very well for heavy rain conditions in a video. Liu and Xu detect rain using the chromatic property, and they develop a discriminant function to eliminate false detections [Liu08a]. Their algorithm considers only videos with stationary background taken by a stationary camera, and it uses the threshold values which have to be estimated depending upon the video in consideration. They improve their method in [Liu09a] by effectively segmenting the rain and moving object pixels to work for any video with better and effective results.

1.2 Our Work

In this paper we discuss the removal of rain using the temporal-spatial statistical properties. The intensities of each pixel are analyzed for the first 15 frames. Using the statistical properties of the pixels, an algorithm is empirically developed to distinguish the rain affected pixels from the other unaffected pixels. This algorithm works for the videos of the static scenes taken by a stationary camera. To distinguish rain affected pixels in the videos with changing background taken by both stationary and moving cameras, further processing of the rain affected pixels is carried out. An empirical distinguishable property of the rain affected pixels is used. This property states that the difference in the intensities of the rain affected pixels in consecutive frames is comparatively lower than the difference in the intensities of the pixels affected by the motion of the object.

Section 2 deals with the detection and the removal of rain. Section 3 deals with the experimental results and the comparison of our algorithm with some of the previously existing algorithms. Section 4 leads to the conclusion where we discuss the benefits and the drawbacks of the proposed method. Section 5 lists the references that we have used for our study.

(a)

(b)

Figure 1: Illustrations of the result of using temporal median based filtering (a) A scene from a video with heavy rain (b) A scene from a video with heavy rain and poor contrast

2. DETECTION AND REMOVAL OF RAIN

2.1 Rain Detection

The temporal statistical properties are used for the detection of rain. We deal with the case of a stationary camera and stationary background, which may or may not contain moving objects. Then we will deal with a more general case involving the motion of camera with a changing background.

We consider the intensities of each pixel for the first 15 frames. We take the average of the intensities for each pixel over the sequential 15 frames. We have considered a few observations reported by Garg and Nayar [Garg04a] as our initial assumptions. These observations are mentioned below.

- The intensity of a pixel shoots to a very high value as compared to its background when it is occluded by a rain drop.
- A pixel is not always covered by rain throughout the video.
- Also a pixel is almost negligibly covered by rain in more than 2 frames consecutively. The case where the pixel is covered by rain drops in more than two frames consecutively has also been accounted for by our rain removal algorithm.

So we take the average intensity of each pixel over the 15 frames. Let us say that for a particular pixel *i* in frame *n*, this specific value is $t_{i,n}^1$. The intensity values higher than this average value $t_{i,n}^1$, are considered and stored separately. Then we take the average of these higher intensities, $t_{i,n}^2$ and take the mid value between this average and the average $t_{i,n}^1$ calculated for all the frames earlier, as the threshold. Let us say that this threshold value is $t_{i,n}^3$ whose value is obtained from equation 2.1a.

$$t_{i,n}^3 = \frac{t_{i,n}^1 + t_{i,n}^2}{2} \tag{2.1a}$$

The intensity values greater than $t_{i,n}^3$ are empirically found out to be affected by rain, and the rest are not generally affected by rain.

We carry out this processing for all the pixels in the next 15 frames and hence forth, till the end of the video. This method will detect the rain in the case where the camera's position is fixed and the background is stationary, without involving the motion of any random object. Next we consider the case involving the motion of a random object in the video with the fixed position of the camera. This algorithm will detect the motion of a random object as false positives. In that case we can use another property of rain affected pixels which is described ahead, to distinguish them from the motion of some random object. So, we refine this algorithm further to remove the false positives.

We have observed experimentally that for a specific frame *n*, and for a particular pixel *i* affected by rain, the difference $\Delta I_{i,n,r}$ between the intensities for the frame *n* and the frame *n-1*, is lower than the difference $\Delta I_{i,n,o}$ between the intensities of the pixels affected by the motion of the random object for the frame *n* and frame *n-1*. This can be expressed mathematically with the equations.

$$\Delta I_{i,n,r} = I_{i,n,r} - I_{i,n-1,r} \tag{2.1b}$$

$$\Delta I_{i,n,o} = I_{i,n,o} - I_{i,n-1,o} \tag{2.1c}$$

$$\Delta I_{i,n,r} < \Delta I_{i,n,o} \tag{2.1d}$$

Here *r* refers to rain affected pixels and *o* refers to the pixels affected by the motion of the object. We have a general observation for these differences in intensity values.

$$\Delta I_{i,n,r} = \alpha(t_{i,n}^2 - t_{i,n}^1) \tag{2.1e}$$

$$\Delta I_{i,n,o} = \beta(t_{i,n}^2 - t_{i,n}^1) \tag{2.1f}$$

It is observed that the value of α lies in the range [0.2 – 0.5] and the value of β is generally greater than 0.6. This summarizes the range for α and β that we get from the videos which we used for our observation. The more precise narrow range for α and β will depend on the experimental video more accurately. Using this property of the rain affected pixels we can reduce the false positives further. This method can effectively eliminate the edges detected by the motion of the object in sequential frames, which are wrongly detected as candidate rain pixels.

Finally in the case involving the motion of the camera, we consider the videos where the background changes at very slow rate so that intensity of the pixels can be considered for the desired number of frames by their proper alignment. The limit for the frame rate of the videos, for which our algorithm seems to produce good results, lies between 10-15 frames per minute. The next section involves the removal of rain from the rain affected candidate pixels.

2.2 Removal of Rain

Here we consider the cases where a candidate pixel is affected once or more in three consecutive frames. In the case where a particular pixel *i* is affected once by rain in frame *n*, the intensity $I_{i,n}^{first}$ is calculated as an average of the intensity of the pixel in the previous frame $I_{i,n-1}$ and the next frame $I_{i,n+1}$, where the pixel is not affected by rain. This concept is taken from Garg and Nayar [Garg04a].

$$I_{i,n}^{first} = \frac{I_{i,n-1} + I_{i,n+1}}{2} \quad (2.2a)$$

If the pixel is affected by rain more than once in consecutive frames, we consider two cases. In one case the pixel is affected twice in three consecutive frames, and in the other one which is very rare in practical situations, a particular pixel is affected in all the three consecutive frames. For the former case, if a pixel *i* is affected in frame *n* and frame *n-1*, the pixel in frame *n-1* is convolved with the spatial 3×3 mask which is illustrated in Figure 2. The justification behind using this spatial filter is based upon an empirical observation which is, it is very improbable for all the pixels in the 3×3 neighborhood of the affected pixel to be covered by rain drops and especially in a streak. Also, the close neighborhood of a pixel generally has an identical intensity background pattern.

The net intensity of the pixel *i*, $I_{i,n}^{second}$ is then computed as the average of the intensity $I_{i,n+1}$ in the *n+1* frame and the intensity after spatial filtering $I_{i,n-1}^{spat.filt.}$ in the *n-1* frame.

$$I_{i,n}^{second} = \frac{I_{i,n-1}^{spat.filt.} + I_{i,n+1}}{2} \quad (2.2b)$$

We will get a similar result if the pixel in the *n+1* frame has been affected instead of the *n-1* frame.

$$I_{i,n}^{second} = \frac{I_{i,n-1} + I_{i,n+1}^{spat.filt.}}{2} \quad (2.2c)$$

Next we consider the final case, which is very rare in practical situations, where a particular pixel is affected consecutively in three frames. This is the case when the rain is very heavy as in the case of a hurricane or storm. The intensity $I_{i,n}^{third}$ of the pixel *i* in frame *n* is then calculated as the average of the intensities $I_{i,n-1}^{spat.filt.}$ and $I_{i,n+1}^{spat.filt.}$ of the pixel in the frames *n-1* and *n+1*, respectively, after it has been spatially filtered using the same 3×3 mask as shown in Figure 2.

$$I_{i,n}^{third} = \frac{I_{i,n-1}^{spat.filt.} + I_{i,n+1}^{spat.filt.}}{2} \quad (2.2d)$$

$$\frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Figure 2: Spatial 3×3 mask

3. EXPERIMENTAL RESULTS

We applied this algorithm on different videos involving heavy rain, changing background, static and dynamically changing positions of camera. We consider these cases one by one to show the effectiveness of our algorithm in different practical scenarios. The results here are shown for the implementation of this technique on the standard videos used by Garg and Nayar [Garg04a], Zhang [Zhang06a], and Park and Lee [Park08a] to facilitate comparison with their methods. We compare our work with their methods since they performed completely independent, unrelated, and pioneering work in this area. The rest of the work done by other people involves the usage of some part of their algorithms to develop and modify their own technique for rain detection and its removal from videos.

These videos were taken from the work done by Garg and Nayar, and Zhang [Garg04a, Garg05a, Garg06a, Zhang06a]. We consider a simple case of a video in which rain is falling heavily in front of a brick wall causing ripples on the ground. Here the background is not changing and the position of the camera is fixed. Figure 3(a) shows the original image frame, and Figure 3(b) shows the same image frame after the application of our algorithm. It is quite clear that the rain streaks have been removed very well. Next we consider the similar case of a scene of a wall with dense rain streaks. Figure 4(a) shows an image frame from the video, and Figure 4(b) shows the same clear image frame obtained after the removal of the rain streaks. The quality of the picture obtained after the application of our algorithm is very good.

(a)

(b)

Figure 3: Video with still background and stationary camera position (a) Original image frame with clearly visible rain streaks (b) The same image frame obtained after the application of our algorithm on the video

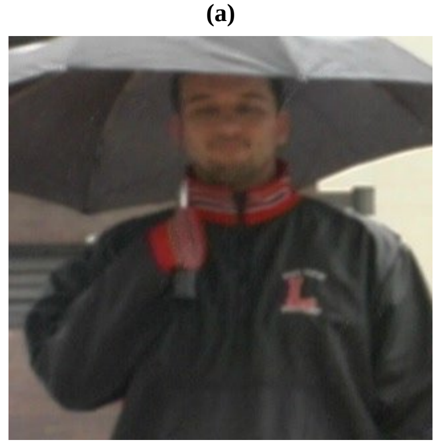
Next we consider a more general case where the camera's position is fixed and the object is moving. Here we consider the case where there are candidate rain pixels in the foreground and the background. Figure 5(a) shows an image frame taken from a video in which a man is moving and the background as well as the camera is fixed in position. Here the rain streaks are very clear. Figure 5(b) shows the same frame after the removal of the rain streaks from the background as well as the foreground containing the moving object. The developed algorithm proves to be really effective in this case.

(a)

(b)

Figure 4: Video with still background and stationary camera position (a) Original image frame with heavy rain streaks (b) Image frame obtained after the application of our algorithm on the video containing the frame shown in figure 3(a)

(a)

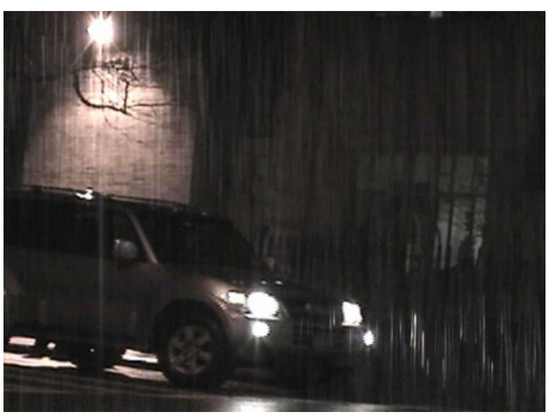


(b)

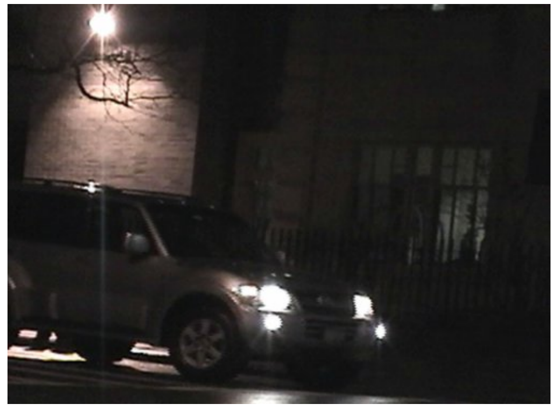
Figure 5: Video with still background and moving

object with stationary camera position (a) The frame shows the image of a moving object with fixed background and stationary camera position containing rain streaks (b) This image shows the same frame after removal of heavy rain streaks using our algorithm

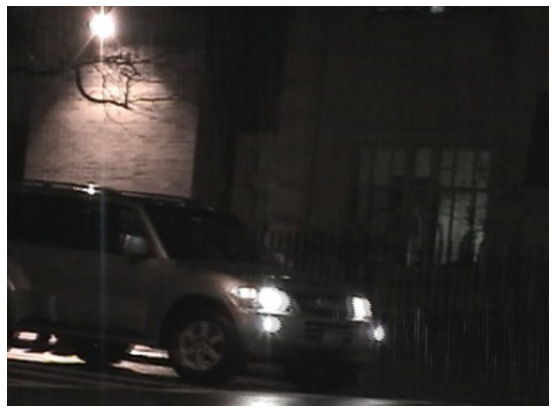
We have done qualitative comparison of our algorithm with the previously existing algorithms developed by Garg and Nayar [Garg04s], Zhang [Zhang06a], and Park and Lee [Park08a]. For this we considered a challenging case where the contrast is dark, the background is changing at a slow rate with heavy rain and the position of the camera is changing. A particular image frame is considered in Figure 6(a). Figure 6(b) shows the same frame with the rain streaks almost completely removed after the application of our algorithm. Figure 6(c) shows the frame after the application of Garg and Nayar's method. It can be seen that their method is not very effective in removing the rain streaks completely in this case. Figure 6(d) shows the image frame after the application of Zhang's method. Here the rain streaks have been removed in a better way as compared to Garg and Nayar's method. However, some rain streaks can still be perceived. Finally, Figure 6(e) shows the result after applying the Kalman filtering process as proposed by Park and Lee. Since Park and Lee's method does not perform well for videos taken by cameras with changing positions, rain streaks are still very evident in Figure 6(e) after the Kalman filtering process. The better clarity in the visual content after the removal of rain from the video using our algorithm can be compared to the results obtained after the application of other methods as shown below.



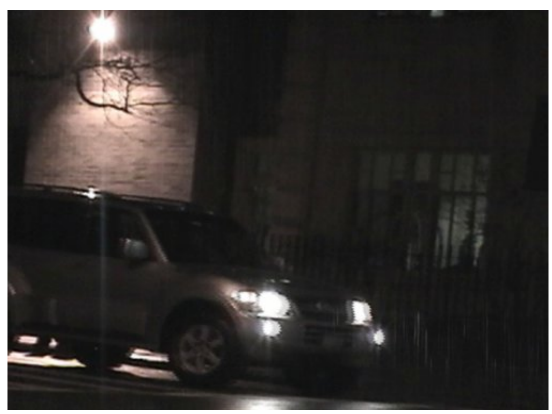
(a) Original Scene



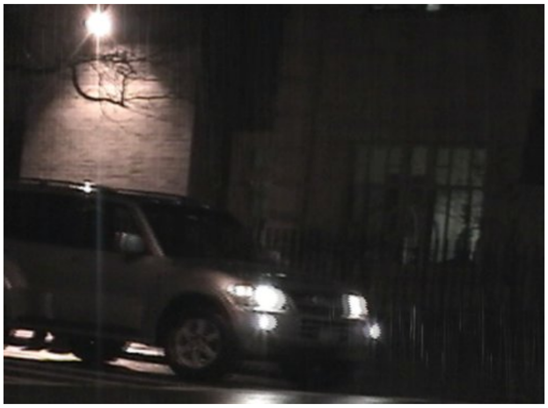
(b) Proposed Method



(c) Garg and Nayar's Method



(d) Zhang's Method



(e) Park and Lee's method

Figure 6: Qualitative comparison of our algorithm with the previously existing algorithms. Image frame is taken from a video having a dark contrast with changing background and dynamically changing camera position (a) Original scene (b) Rain streaks have been almost completely removed with the help of our algorithm (c) This image frame shows the result after the application of Garg and Nayar's method (d) Image frame after the application of Zhang's algorithm (e) This image frame shows the result after the application of Kalman filtering process as implemented by Park and Lee.

As we had mentioned in the introduction section, our purpose for this work is to restore the images which have been considerably weather degraded, to carry out tasks like object recognition, object detection, and image registration using vision algorithms. In this case we detect feature points on the image after which we carry out techniques like SIFT, SURF and MSER. It is observed that we do not get proper points in the images which have been degraded by rain. Hence, we establish the quantitative performance of our method in terms of the detection of proper feature points using the Harris-Affine Detector and the Hessian-Affine Detector [Mikolajczyk04a]. This is a completely new approach in comparison to the previous approaches to quantitatively judge the efficiency of their algorithms. Our method for evaluating quantitative performance is very specific to our objective for carrying out this work. We hope that this kind of evaluation has a potential for further research where the aim of restoring weather degraded images is to carry out content based indexing and retrieval in images and videos.

Table 1 and Table 2 illustrate the performance of Harris-Affine and Hessian-Affine Detectors on the images which have been illustrated in Figure 3, 4 and 5.

Table1: Harris-Affine Detector

Image	Resolution	Correct Number of Points Detected	Total Number of Points Detected	Time Taken
Figure 4(a)	368×288	109	550	1.2s
Figure 4(b)	368×288	114	508	1.183s
Figure 5(a)	320×304	44	239	1.033s
Figure 5(b)	320×304	47	215	1.033s
Figure 6(a)	504×376	115	767	2.333s
Figure 6(b)	504×376	139	710	2.317s

Table2: Hessian-Affine Detector

Image	Resolution	Correct Number of Points Detected	Total Number of Points Detected	Time Taken
Figure 4(a)	368×288	120	219	0.433s
Figure 4(b)	368×288	136	185	0.417s
Figure 5(a)	320×304	56	133	0.333s
Figure 5(b)	320×304	66	124	0.333s
Figure 6(a)	504×376	148	311	0.817s
Figure 6(b)	504×376	153	296	0.800s

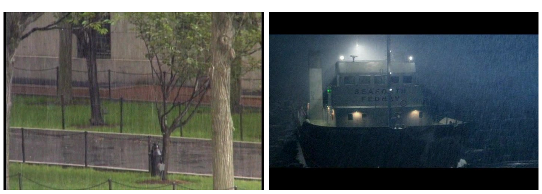
It can be seen that the proper number of points detected using the Harris-Affine and Hessian-Affine detectors is more in the case of restored images in comparison to the weather degraded images, where the improper number of feature points detection along with the time taken for it, is more.

4. CONCLUSIONS

The experimental results show that the proposed algorithm works very well for different scenes with still and moving backgrounds with moving random objects having varying textures and still and moving camera positions. We have implemented the complete setup on MATLAB platform. The results show that the efficiency of our algorithm is comparable to the previously existing algorithms. We have dealt with the particular cases where a pixel is covered by rain in more than one frame which gives us advantage over Garg and Nayar's method where the average of the intensities of the neighboring pixels in the same frame is taken as the intensity of the pixel in that frame.

Our method has a small latency, so it could be used in real time processing applications where latency does not need to be strictly negligible. This gives us an advantage over Zhang's method which considers many frames from the complete video for processing and thus is not suitable for real time processing applications. Also, Park and Lee's method of Kalman filtering process does not apply well for videos involving changing background and changing camera position. Thus, although their method does not involve any significant latency, it considers very specific cases which are not that practical in day to day applications. Our image quality is comparable to other methods for the general case involving slowly changing background, along with the changing camera position. Also, we do not consider the size, velocity, shape, and any physical model of rain streaks, and the external parameters like camera's aperture size, focal length, and exposure time.

On the other hand, there are a few limitations of this method as well. The small latency makes the method unsuitable for real time processing applications requiring no latency at all. Also in the case of videos having image frames with very large resolution, this method would require a lot of storage memory which may make it unsuitable for some specific practical applications. In the case of heavy rain when the spatial filter is convolved two times in two frames for a given pixel position, there is a slight degradation of quality of the video, in terms of a little blurring of the details. Still the video is better in terms of quality after the removal of heavy rain from it. The range of α and β is empirical and in some cases there are misclassifications between the rain affected and the moving object pixels. For videos with very bright background, the rain affected pixels may not be detected properly leading to insufficient removal of rain from the frames. This method cannot deal with videos with very fast changing background and camera position. Figure 7 illustrates two cases where this algorithm is not effective. In Figure 7(a), the camera's frame change rate is very fast; whereas in Figure 7(b) the rain is very heavy along with mist as experienced in a hurricane or storm. Future work focuses upon dealing with these issues.



(a) **(b)**

Figure 7: Scenes from the videos where the performance of the algorithm is not effective (a) A scene from a video where the background is changing at a high frame rate (b) A scene from a video with extremely heavy rainfall which is experienced in a hurricane or a storm.

5. REFERENCES

[Starik03a] S. Starik and M. Werman, "Simulation of Rain in Videos," in International Workshop on Texture Analysis and Synthesis, 2003.

[Mikolajczyk04a] Krystian Mikolajczyk and Cordelia Schmid, "Scale and Affine Interest Point Detectors," in International Journal of Computer Vision 60(1), pp. 63-86, 2004.

[Garg04a] K. Garg and S.K. Nayar, "Detection and Removal of Rain from Videos," in Proc. CVPR, vol. 1, pp. 528-535, 2004.

[Garg05a] K. Garg and S.K. Nayar, "When Does a Camera See Rain?," in Proc. ICCV 2005, vol. 48, no. 3.

[Garg06a] K. Garg and S.K. Nayar, "Photorealistic Rendering of Rain Streaks," in Proceedings of ACM SIGGRAPH 2006.

[Zhang06a] Xiaopeng Zhang and Teck Khim Ng, "Rain Removal in Video by combining Temporal and Chromatic Properties," in Proc. Multimedia and Expo, 2006.

[Barnum07a] Peter Barnum, Takeo Kanade, Srinivasa G Narsimhan, "Spatio-Temporal Frequency Analysis for Removing Rain and Snow from Videos," in PACV Workshop at ICCV 2007.

[Zhao08a] Xudong Zhao, Peng Liu, Jiafeng Liu, and Xialong Tang, "The Application of Histogram on Rain Detection in Video," in Proceedings of the 11th Joint Conference on Information Sciences, 2008.

[Liu08a] Liu Peng, Xu Jing, Liu Jiafeng, Tang Xianglong, Zhao Wei, "A Rain Removal Method Using Chromatic Property for Image Sequence," in Proceedings of the 11th Joint Conference on Information Sciences, 2008.

[Park08a] Wan-Joo Park and Kwae-Hi Lee, "Rain Removal using Kalman Filter in Video," in International Conference on Smart Manufacturing Application, April 9-11, 2008.

[Brewer08a] Nathan Brewer and Nianjin Liu, "Using the Shape Characteristics of Rain to Identify and Remove Rain from Video," in SSPR & SPR '08, Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, LNCS 5342, pp. 451 – 558, 2008.

[Liu09a] Peng Liu, Jing Xu, Jiafeng Liu, Xianglong Tang, "Pixel Based Temporal Analysis Using Chromatic Property for Removing rain from Videos," in Journal of Computer and Information Science, Vol. 2, No. 1, February 2009.