Rain Streaks Removal Using Discrete Cosine Transform and Diffusion based Technique

Riya, Bhupendra Gupta and Subir Singh Lamba
PDPM Indian Institute of Information Technology, Design & Manufacturing
Jabalpur, MP, India 482005
E-mail: riyashrama41@gmail.com, gupta.bhupendra@gmail.com, subirs@gmail.com

Abstract. Images acquired by the camera are sometimes affected by many environmental conditions such as fog, rain, snow, and clouds, etc., which influence the visible quality of the images and several computer vision applications. In this manuscript, we developed a new rain removal technique. Firstly, an exiting haze removal technique is used so that rain streaks could be visible and after that apply Discrete cosine transform (DCT) and diffusion-based technique to remove rain streaks from the rainy images. The experimental section shows the supremacy of the proposed method over the existing methods.

Keywords: Rain streaks, Discrete cosine transform, Diffusion, Low-frequency component, High-frequency component

1. Introduction

Bad weather conditions affect the performances of many applications in computer vision like image tracking, object detection, and image recognition. Two types of weather conditions classified in the literature are steady and dynamic conditions [1], [2]. Steady bad weather conditions consist of haze, mist, and fog while dynamic bad weather conditions consist of hail, snow, and rain. In the case of steady bad weather conditions, the droplet size is very small to investigate from the camera, while in dynamic bad weather conditions the size of the droplet is extremely large compared to the steady-state. The large size of the droplets are captured by the camera and degrades the image visual quality. Rain is the well-known component of the dynamic condition and has intense reflection to light. Rain is, generally, considered a bright streak that influences the image visual quality. Hence, it becomes inevitable to remove rain from images for better visual quality.

In recent years, many rain removal algorithms are introduced in the literature [3], [4], [5], [6], [7], [8], [9], [10], [11]. In these algorithms, certain numbers of successive frames are required to find the rain-affected pixel. Garg and Nayar introduced a method
[10] to remove rain during the acquisition by adjusting some camera parameters. This method does not work well for fast-moving objects near to the camera and heavy rain. Garg and Nayar in [1], [3] speculated that most of the raindrops affect only a single frame and a small number of raindrops affect more than two successive frames. The intensity difference between the rain-affected pixel in the latest frame and the earlier frame tells about the change of intensity due to the rain. This speculation produces false rain pixel detection. To avoid false rain pixel detection, it is considered that these raindrops obey linear photometric constraints. In photometric constraints, it is assumed that almost all the raindrops possess same size and velocity and it is also considered that all the pixels lying on the same rain streak possess identical irradiance because the brightness of the pixel is less influenced by the background. However, in real life, it is well established that there is a disparity in the size of the raindrops and velocity of the raindrops which violates speculation of the photometric constraints. In heavy or light rain, this method inhibits to distinguish between the rain pixel and moving object pixel [4]. It exhibits that all rain streaks do not always obey the photometric constraints, which leads to the miss detection of rain pixels. 30 successive frames are required to rain removal in this method.

For rain removal, a method relied on temporal and chromatic characterization of the rain drops has been proposed in [4]. Temporal property shows that a specific pixel is not necessarily covered by a raindrop in each frame and chromatic property shows that the effect of a raindrop on the red, green and blue channel is approximately the same. These disparities over the color components are bound by a small threshold. The chromatic property is also followed by a slow-moving object. This method employs all the available frames in rain removal and removing rain effectively from a static background where no moving object is available. Chromatic based properties are also used to remove rain from the rain-affected videos in [5]. This method is unable to detect all rain streaks since the chromatic properties are not fulfilled in practice as detailed earlier. At least three consecutive frames are desired for the proper rain removal. Barnum et al. [7], [12] using frequency information of every frame for rain detection and rain removal. For rain removal, a blurred Gaussian model is used in this method. This model is unable to detect unsharp rain streak.

Bossu et al. suggested a Gaussian model [8] by assuming that the rain streaks are steady in orientation. This method only approximates the important rain streaks which cause to miss detection of some raindrops. The intersection of multiple rain streaks is another reason for miss detection, which also causes unpleasant shapes and orientation. Tripathi et al. [9] advocated a probabilistic based method for rain removal. In this method, the temporal properties of rain are used for rain removal. This method removes rain streaks from the videos effectively without inducing any degrading effect. The probabilistic approach seeks 15 successive frames for rain removal, which is lesser than the existing methods. But its implementation requires a large memory size which delays the result.

Recently, researchers turns their rain removing approach from video to single image
[13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. In [20], [21], [23], nonlocal mean filter and guided filter are used for rain removal. Filtering based methods are fast in implementation and easy to use but rarely give a satisfactory result. Either some rain streaks remain in the resultant image or some image details are considered as rain streaks, which causes blurring in the output image. Some other models build to distinguish rain from the rainy image background [22], [24], [25]. However, these models still cause image blurring because of some image details and edges are considered as rain streaks. Hence, they produce a loss of some details in the image. A two-step processing introduced in [13], [14], [15], in which low pass filter is employed to find high-frequency and low-frequency components. Here, the low-frequency component could be made rain less and the high-frequency part is processed by some descriptors for extracting the image details which is fed back in the processed low-frequency component to get a resultant rain-free image. Deep learning-based methods are proposed for rain removal [16], [17], [26], [27]. These models give the best result by designing significant deep networks.

Although filtering based methods yield significant results still have some drawbacks, either remove some significant details from the image together with rain component or do not remove rain component effectively from the image. This work proposes a novel rain removal method that effectively removes the rain and preserves image details.

2. Proposed method

Here, a new rain removal technique is introduced. Figure 1 depicts the framework of the proposed method. It is observed that the rain images contain some hazy effect, which creates a problem for the proper extraction of rain streaks because some rain streaks do not visible in the hazy effect. For proper extraction of the rain streaks firstly, a haze effect removal technique [28] is applied on the input rainy image so that all rain streaks are visible as one can observe in Figure 2. Afterward, discrete cosine transform (DCT) [29] is applied on the haze-free image to decompose it into low-frequency component \((L_D)\) and high-frequency component \((H_D)\). For a 2-D image, 2-D DCT coefficient \(g\) of size \(m \times n\) is calculated by

\[
g(u, v) = c_u c_v \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Y(i, j) \cos \left( \frac{\pi(2i+1)u}{2m} \right) \times \cos \left( \frac{\pi(2j+1)v}{2n} \right),
\]

where, \(Y\) is the haze removed image, \(u \leq m - 1, v \leq n - 1\) and \(c_u, c_v\) are calculated by

\[
c_u = \begin{cases} \frac{1}{\sqrt{m}}, & u = 0 \\ \frac{2}{\sqrt{m}}, & 1 \leq u \leq m - 1 \end{cases}, \quad c_v = \begin{cases} \frac{1}{\sqrt{n}}, & v = 0 \\ \frac{2}{\sqrt{n}}, & 1 \leq v \leq n - 1 \end{cases},
\]

where, high-frequency component \((H_D)\) contains significant rain streaks and prominent image details. Again, the input rainy image is processed by the diffusion-based PM model [30], which is now a rain-free image with prominent geometrical edges.
Figure 1. Framework of the proposed method.

```
I_t = \text{div}(c(x, y, t)\nabla I) = c(x, y, t)\Delta I + \nabla c \cdot \nabla I, \quad I(x, y, 0) = I_0, \quad (3)
```

where \( I \) is the input image and \( t \) is the number of iterations. In this work, \( t \) is taken as 5. The rain-free image is decomposed into high-frequency component \((H_S)\) and low-frequency component \((L_S)\), where the high-frequency component contains significant image details. To extract rain streaks from the input image, the absolute difference of both of the high-frequency components \((H_D \& H_S)\) is obtained. Maximum rain streaks are obtained from this difference, called rain component \((RC)\).

Now, rain component \((RC)\) is subtracted by the earlier high-frequency component \((H_D)\) from this we get a detail component with significant image detail and added to the rain-free image. At last, a final rain-free image with significant image details is obtained. Figure 3 shows the results obtained by the proposed method.
3. Experimental results and discussion

This section shows the efficiency of the proposed method over the state-of-the-art methods [21], [23]. For experimental analysis, the most common rainy images are used. The rain removal outputs are compared in Figure 4 and Figure 5. In Figure 4, the first column contains the original rainy images, the second and third column contains the results obtained by the Zheng et al. [21] and Ding et al. [23] while the last column contains the results obtained by the proposed method. From Figure 4, it is observed that results obtained by Zheng et al. [21] method is demonstrating the smoothing effect while some rain components are still visible in the results obtained by Ding et al. [23]. The smoothing effect in Zheng et al. [21] method shows in Figure 5, where one can see the image details with the rain component of the image. This shows that there is some loss of image details in the processed images.

In contrary, results obtained by Ding et al. [23] have less detail removal effect but
not effectively removing the rain from the images, which can be seen in Figure 5, while the proposed method effectively removes the rain and preserves more image detail in the processed image. The efficiency of the proposed method can be seen in Figure 4 and Figure 5. The comparison of the proposed method with the state-of-the-art methods shows that the proposed method is outperforming the other methods.

4. Conclusion

This work presented a discrete cosine transform (DCT) and diffusion-based rain removal technique. The results depict that the proposed method efficiently removes the rain component from the image without affecting the image details. It is observed that the existing methods either over smooth the images or affect the image details while the proposed method removes the rain streaks and preserves more image details.

References

[1] Garg K and Nayar S K 2007 *International Journal of Computer Vision* **75** 3–27
[2] Narasimhan S G and Nayar S K 2002 *International Journal of Computer Vision* **48** 233–254
[3] Garg K and Nayar S K 2004 *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol 1* (IEEE) pp 528–535
Figure 5. Rain component obtained by Zheng et al. [21], Ding et al. [23] and the proposed method (better visible in enlarge view).

[4] Zhang X, Li H, Qi Y, Leow W K and Ng T K 2006 2006 IEEE international conference on multimedia and expo (IEEE) pp 461–464
[5] Liu P, Xu J, Liu J and Tang X 2009 Computer and information science 2 53–60
[6] Park W J and Lee K H 2008 2008 International Conference on Smart Manufacturing Application (IEEE) pp 494–497
[7] Barnum P, Kanade T and Narasimhan S 2007 Proceedings of the First International Workshop on Photometric Analysis For Computer Vision-PACV 2007 (INRIA) pp 1–8
[8] Bossu J, Hautiere N and Tarel J P 2011 International journal of computer vision 93 348–367
[9] Tripathi A K and Mukhopadhyay S 2011 IETE Journal of Research 57 82–91
[10] Garg K and Nayar S K 2005 Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1 vol 2 (IEEE) pp 1067–1074
[11] Brewer N and Liu N 2008 Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR) (Springer) pp 451–458
[12] Barnum P C, Narasimhan S and Kanade T 2010 International journal of computer vision 86
[13] Fu Y H, Kang L W, Lin C W and Hsu C T 2011 *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 1453–1456

[14] Kang L W, Lin C W and Fu Y H 2011 *IEEE transactions on image processing* **21** 1742–1755

[15] Chen D Y, Chen C C and Kang L W 2014 *IEEE transactions on circuits and systems for video technology* **24** 1430–1455

[16] Fu X, Huang J, Zeng D, Huang Y, Ding X and Paisley J 2017 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp 3855–3863

[17] Yang W, Tan R T, Feng J, Liu J, Guo Z and Yan S 2017 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp 1357–1366

[18] Huang D A, Kang L W, Yang M C, Lin C W and Wang Y C F 2012 *2012 IEEE International Conference on Multimedia and Expo* (IEEE) pp 164–169

[19] Huang D A, Kang L W, Wang Y C F and Lin C W 2013 *IEEE Transactions on Multimedia* **16** 83–93

[20] Xu J, Zhao W, Liu P and Tang X 2012 *Computer and Information Science* **5** 49

[21] Zheng X, Liao Y, Guo W, Fu X and Ding X 2013 *International Conference on Neural Information Processing* (Springer) pp 258–265

[22] Chen Y L and Hsu C T 2013 *Proceedings of the IEEE International Conference on Computer Vision* pp 1968–1975

[23] Ding X, Chen L, Zheng X, Huang Y and Zeng D 2016 *Multimedia Tools and Applications* **75** 2697–2712

[24] Luo Y, Xu Y and Ji H 2015 *Proceedings of the IEEE International Conference on Computer Vision* pp 3397–3405

[25] Li Y, Tan R T, Guo X, Lu J and Brown M S 2016 *Proceedings of the IEEE conference on computer vision and pattern recognition* pp 2736–2744

[26] Zhang H, Sindagi V and Patel V M 2019 *IEEE transactions on circuits and systems for video technology* **30** 3943–3956

[27] Fu X, Huang J, Ding X, Liao Y and Paisley J 2017 *IEEE Transactions on Image Processing* **26** 2944–2956

[28] Jha D K, Gupta B and Lamba S S 2016 *IET Computer Vision* **10** 331–343

[29] Ochoa-Dominguez H and Rao K R 2019 *Discrete Cosine Transform* (CRC Press)

[30] Perona P and Malik J 1990 *IEEE Transactions on pattern analysis and machine intelligence* **12** 629–639