Recovering Raindrop Removal Images under Heavy Rain

Kosuke Matsumoto, Fumihiko Sakaue and Jun Sato
Nagoya Institute of Technology, Japan
{*matsumoto@cv., sakaue@, junsato@*}*nitech.ac.jp*

Keywords: Raindrop Removal, Heavy Rain, GAN, B-spline.

Abstract: In this paper, we propose a new method for removing raindrops in images under heavy rain. When we drive in heavy rain, the raindrops attached to the windshield form a film and our visibility degrades drastically. In such situations, the existing raindrop removal methods cannot recover clear images, since these methods assume that the background scene is visible through the gap between the raindrops, which does not happen anymore in heavy rain. Thus, we in this paper propose a new method for recovering raindrop removal images under heavy rain from sequential images by using conditional GAN. The results of our experiments on real images and synthetic images show that the proposed method outperforms the state-of-the-art raindrop removal method.

1 INTRODUCTION

When driving in heavy rain, the risk of an accident goes up since a large amount of raindrops adhere to the windshield and hinder visibility. In recent years, the realization of autonomous vehicles is expected, but even in the autonomous vehicles that use in-vehicle cameras, keeping visibility in the rain is a big problem, as with human drivers.

Thus, in recent years, some methods for removing raindrops from in-vehicle camera images have been proposed. However, these methods assume that the amount of raindrops attached to the windshield is relatively small hence the background scene can be observed through the gaps between the raindrops. Thus, in the case of heavy rain, when raindrops spread like a film over the entire windshield, these existing methods cannot generate raindrop removal images appropriately.

Therefore, in this paper, we propose a new method for generating raindrop removal images properly as shown in Fig. 1 (b) even when raindrops adhere to the entire glass surface as shown in Fig. 1 (a).

In our method, the raindrops are not considered as occluding objects but are considered as transparent objects that refract incident light in various directions and distort the observed images. Our method is based on end-to-end learning, where the input is a set of sequential images distorted by heavy rain and the output is a clear image with no distortion. We use a conditional GAN framework for training a generator efficiently. By using the property of sequential images under heavy rain, where the background scene does not change drastically while the non-uniform raindrops on the windshield randomly change the light paths to the viewpoint of the camera, our network can recover a clear undistorted image of the background scene from a set of sequential images distorted by heavy rain.

(a) image under heavy rain (b) result of our method

Figure 1: Input image observed under heavy rain and raindrop removal image obtained from our method.

For training our network, we need pairs of raindrop and undistorted images. However, under actual heavy rain, it is very difficult to obtain corresponding images with and without raindrops. Thus, we in this research synthesize training dataset of sequential images captured under heavy rain, and train our network by using the synthetic heavy rain dataset. We show that our network trained on the synthetic heavy rain dataset outperforms the state-of-the-art raindrop removal method proposed by Qian (Qian et al., 2018) in our real image experiments.

Matsumoto, K., Sakaue, F. and Sato, J.
Recovering Raindrop Removal Images under Heavy Rain.
DOI: 10.5220/0009325601280136
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 4: VISAPP*, pages 128-136
ISBN: 978-989-758-402-2; ISSN: 2184-4321

2 RELATED WORK

There are two major causes of poor visibility in rainy weather. The first one is rain streaks that block light passing through 3D space, and the second one is raindrops attached to the windshield that refract light and distort images.

For removing rain streaks in images, many authors proposed image dehazing methods based on rain characteristics (Garg and Nayar, 2004; N. and N., 2008; Chen and Hsu, 2013; Santhaseelan and Asari, 2014), sparse coding (Luo et al., 2015) and deep neural networks (Fu et al., 2017; Li et al., 2017; Yang et al., 2017). However, these methods cannot recover images which are distorted by raindrops on the windshield.

For removing raindrops attached on windows, multiple cameras were often used to obtain the scene information that is hidden by raindrops and invisible at a certain viewpoint (Yamashita et al., 2005; Matsui et al., 2014). The sequential images were also used for obtaining the scene information hidden by raindrops at a certain time instant (Yamashita et al., 2009; Nomoto et al., 2011; You et al., 2016). More recently, the deep learning technique is used for recovering raindrop removal images from a single camera view (Qian et al., 2018).

Although these raindrop removal methods work efficiently under light rain, they no longer work properly under heavy rain. This is because these methods assume that the background scene can be observed through the gaps between the raindrops, that is no longer the case in heavy rain.

Thus, we in this paper consider the raindrops not as occluding objects but as refractive media that distort the observed images, and propose a method for recovering distorted images under heavy rain.

3 RAINDROP REMOVAL USING GENERATIVE ADVERSARIAL NETWORK

In the heavy rain, the entire image is covered with raindrops, and the background scene cannot be observed through the gaps between the raindrops. Thus, in this research, we recover undistorted background scene by using the entire image which is distorted by unknown non-uniform water film.

We assume that the 3D shape of the non-uniform water film changes over time, and we can observe a set of sequential images in which the background scene is distorted by the changing non-uniform water film. We use such distorted sequential images for recovering the undistorted back ground scene image. Although the image distortion is dynamic and non-uniform, we think the distorted images include the background scene information, and hence the undistorted background scene image can be recovered from the distorted sequential images.

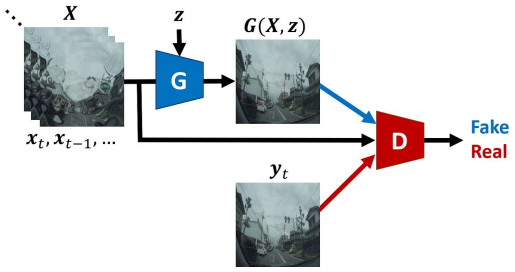


Figure 2: Generative adversarial network (GAN) for generating raindrop removal images. Generator G generates a raindrop removal image $\hat{y}_t$ from a set of sequential distortion images X and noise z. Discriminator D learns to discriminate a false pair $(\hat{y}_t, X)$ and a true pair (y_t, X), and generator G is trained so that it minimizes correct answer of discriminator.

For recovering undistorted scene images from the distorted sequential images, we use conditional GAN (Isola et al., 2017). The network structure of our conditional GAN is shown in Fig. 2. The generator G is a 16-layer convolution-deconvolution network (U-Net) (Ronneberger et al., 2015) and the discriminator D is a 5-layer convolution network.

The input X of the generator is a set of distorted sequential images up to the current time t as follows:

$$X = \{x_t, x_{t-1}, \cdots, x_{t-T+1}\} \tag{1}$$

where, T is the number of time instants in the set of sequential images, X. The generator G generates an undistorted image $\hat{y}_t$ at time t from the set of distorted sequential images X and a random noise vector z as follows:

$$\hat{y}_t = G(X, z) \tag{2}$$

The ground truth of the undistorted image is denoted as y_t. Unlike the standard conditional GAN, our generator G generates a single undistorted image $G(X, z)$ from multiple images X.

The discriminator D is trained so that a ground truth pair $\{X, y_t\}$ is determined to be true and a fake pair $\{X, G(X, z)\}$ is determined to be false. The network is trained, so that the discriminator maximizes the rate of correct judgments and the generator minimizes it. Thus, the training of our conditional GAN can be described as follows:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{3}$$

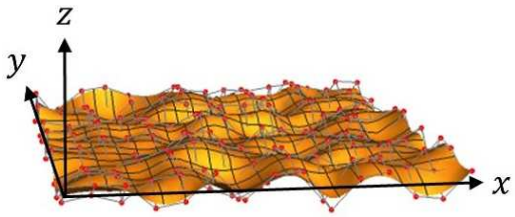


Figure 3: B-spline surface. Red points show control points of B-spline surface.

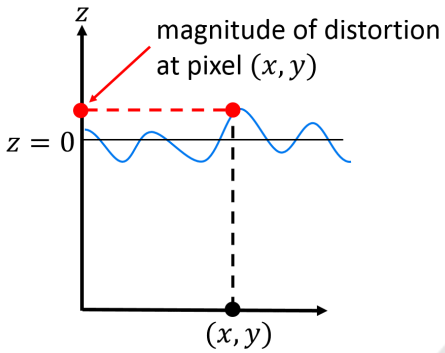


Figure 4: Representation of distortion by using a B-spline surface. The magnitude of distortion at pixel (x,y) is represented by the z coordinate of the B-spline surface at (x,y).

where, $\mathcal{L}_{cGAN}$ is the following adversarial loss:

$$L_{cGAN}(G,D) = E_{X,y_t\sim p_{data}(X,y_t)}[\log D(X,y_t)] + E_{X\sim p_{data}(X),z\sim p_z(z)}[\log(1-D(X,G(X,z)))] \quad (4)$$

and $\mathcal{L}_{L1}$ is an L_1 loss as follows:

$$\mathcal{L}_{L1}(G) = E_{X,y_t\sim p_{data}(X,y_t),z\sim p_z(z)}\|y_t - G(X,z)\|_1 \quad (5)$$

As training progresses, generator G will generate images that make it difficult for the discriminator D to determine authenticity. In other words, a trained generator can generate raindrop removal images that are natural to humans from a series of distorted sequential images.

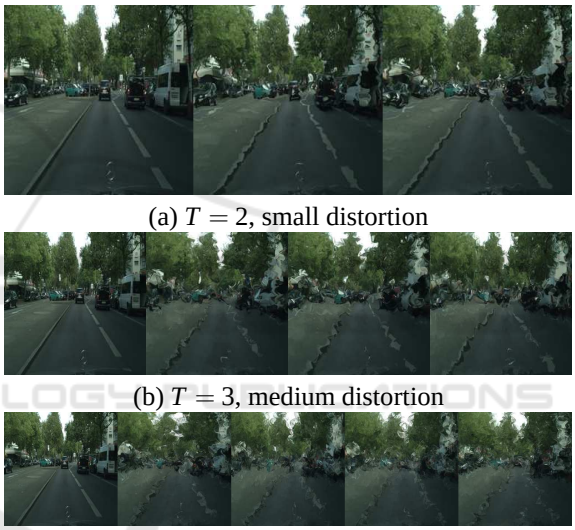
4 DATASET

Since the accuracy of deep learning depends on the dataset, it is important to collect many data for training. However, it is very difficult to obtain various scene data with and without various raindrops, in particular under heavy rain. Thus, in this research, we synthesize images that are distorted by various heavy rains adhere on a windshield and build a dataset for training.

(a) Original images

(b) Raindrop images

Figure 5: Example of raindrop images synthesized by B-spline surfaces.



(a) $T = 2$, small distortion

(b) $T = 3$, medium distortion

(c) $T = 4$, large distortion

Figure 6: Example images in our dataset. (a) shows an original image and two sequential raindrop images in the case of $T = 2$. (b) shows those in the case of $T = 3$ and (c) shows those in the case of $T = 4$ respectively. The magnitude of distortion is small in (a), and those in (b) and (c) are medium and large respectively.

4.1 Representation of Raindrop Distortions

Since we consider situations in which heavy rain generate non-uniform water film on a windshield, the distortion caused by the non-uniform water film is represented by using a B-spline surface. By using the B-spline surface, the surface shape can be controlled by a limited number of control points efficiently. Fig. 3 shows an example of B-spline surface which is generated from the control points shown in red. The B-spline surface $\mathbf{S}(u,v)$is defined by using $K \times K$ control

points $\mathbf{P}_{ij}(i = 1,\cdots,K; j = 1,\cdots K)$ as follows:

$$\mathbf{S}(u,v) = \sum_{i=1}^{K}\sum_{j=1}^{K} N_i^P(u)N_j^P(v)\mathbf{P}_{ij} \tag{6}$$

where, u and v denote parameters in horizontal and vertical axes, and P is the degree of B-spline.

$N_i^P(u)$ and $N_j^P(v)$ are called blending functions and represent the influence of the control points. In the case of a B-spline surface, B-spline basis function is used as the blending function, which is defined by the position of the knot sequence $x_i(i = 1,\cdots,K+P+1)$, as follows:

$$N_i^1(u) = \begin{cases} 1 & (x_i \le u \le x_{i+1}) \\ 0 & (otherwise) \end{cases} \tag{7}$$

$$\begin{aligned} N_i^K(u) &= \frac{u - x_i}{x_{i+k} - x_i} N_i^{K-1}(u) \\ &+ \frac{x_{i+k+1} - u}{x_{i+k+1} - x_{i+1}} N_{i+1}^{K-1}(u) \end{aligned} \tag{8}$$

In this paper, image distortion due to raindrop water surface is represented by the B-spline surface.

4.2 Raindrop Image Synthesis

In case of heavy rain, the entire image is heavily distorted by raindrops. In order to generate a distorted image from an undistorted image, image transformation is performed by using B-spline surfaces. In our B-spline surfaces, x axis represents the horizontal direction of the image, and the y axis represents the vertical direction of the image respectively. The z axis of the B-spline surface represents the displacement of the image point (x,y) caused by the raindrops as shown in Fig. 4. Since we have displacement in x and y axes, the displacement is represented by two independent B-spline surfaces, $\mathbf{S}_x(x,y)$ and $\mathbf{S}_y(x,y)$. Thus, the point coordinates (x,y) of the original image is transformed into (x',y') in the distorted image as follows:

$$\begin{cases} x' = x + x_p(x,y) \\ y' = y + y_p(x,y) \end{cases} \tag{9}$$

where, $x_p(x,y)$ and $y_p(x,y)$ denote the displacement of a point (x,y) in x and y axes, and these are represented by the z coordinate of $\mathbf{S}_x(x,y)$ and $\mathbf{S}_y(x,y)$ respectively.

By varying the control points of the B-spline surface, it is possible to represent distortions caused by various raindrops. Thus, we control the z coordinates of the $K \times K$ control points of $\mathbf{S}_x$ and $\mathbf{S}_y$. More specifically, the image distortion is controlled by a vector $\mathbf{p} = [z_{11}^x,\cdots,z_{KK}^x,z_{11}^y,\cdots,z_{KK}^y]^\top$, which consists of the z coordinates of the $K \times K$ control points of $\mathbf{S}_x$ and $\mathbf{S}_y$. We call it a water surface parameter.

Some example images generated by using the B-spline surface are shown in Fig. 5.

4.3 Sequential Raindrop Image Dataset and Training

By using the method described in section 4.2, we generated a dataset of sequential raindrop images. The sequential images in Cityscapes dataset (Cordts et al., 2016) were distorted by changing the water surface parameter $\mathbf{p}$ randomly. The number of time instants T in sequential images was set to 2, 3 and 4 for evaluating the relationship between the number of time instants T and the accuracy of recovered raindrop removal images.

Furthermore, in order to evaluate the relationship between the magnitude of distortion and the accuracy of undistorted image recovery, the dataset was generated changing the magnitude of image distortion in three patterns of small, medium and large. Thus, the sequential raindrop image dataset was generated with 3 different numbers of time instants and 3 different patterns of distortion, and a total of $3 \times 3 = 9$ datasets were created. Each dataset consists of 325 training sequences and 50 test sequences. Some examples of nine datasets are shown in Fig. 6.

The network explained in section 3 was trained by using the sequential raindrop image datasets. We trained the network with 1000 epoch. Then, the trained generator was used for generating undistorted images from a set of distorted sequential images.

5 EXPERIMENTS

5.1 Synthetic Image Experiments

We next show the experimental results obtained from the proposed method. We first show results from synthetic image experiments.

Fig. 7 shows results under small image distortions, where (a) shows ground truth raindrop removal images and (b) shows input raindrop images distorted by heavy rain. Fig. 7 (c), (d) and (e) show raindrop removal images obtained from the proposed method in the case of $T = 2$, $T = 3$ and $T = 4$ respectively. It can be confirmed that the input images in (b) have distortions whereas the recovered images in (c), (d) and (e) have no distortion.

Fig. 8 shows results under medium distortions. As shown in Fig. 8 (b), we have relatively large distor-

(a) ground truth (b) input image (c) $T = 2$ (d) $T = 3$ (e) $T = 4$

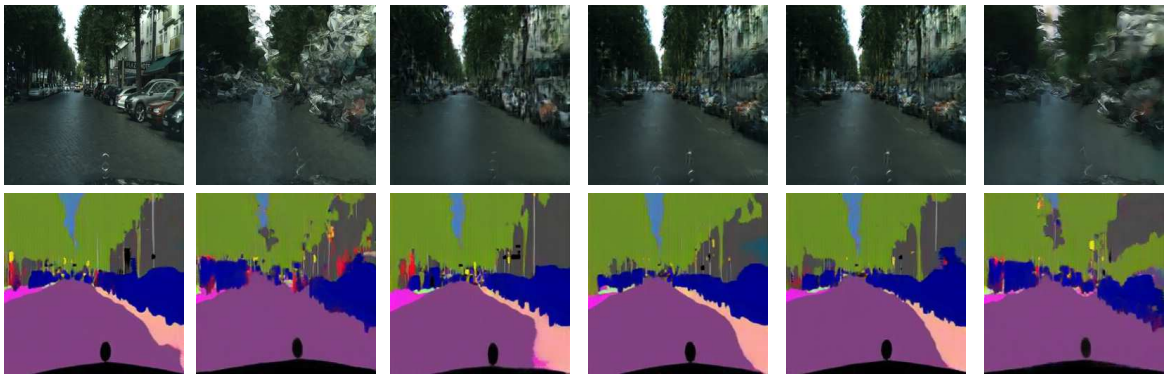
Figure 7: Results under small image distortions. (a) shows ground truth images, (b) shows test input images which are distorted by raindrops, (c), (d) and (e) show raindrop removal images obtained from our method in the case of $T = 2$, $T = 3$ and $T = 4$ respectively.

(a) ground truth (b) input image (c) $T = 2$ (d) $T = 3$ (e) $T = 4$

Figure 8: Results under medium image distortions. See the caption of Fig. 7 for detail explanations.

(a) ground truth (b) input image (c) $T = 2$ (d) $T = 3$ (e) $T = 4$

Figure 9: Results under large image distortions. See the caption of Fig. 7 for detail explanations.



(a) ground truth (b) input image (c) $T = 2$ (d) $T = 3$ (e) $T = 4$ (f) A^2Net

Figure 10: Comparison with the existing method. Columns (a) to (e) are the same as before, and column (f) is the result of the conventional method A^2Net. The first row shows the generated images, and the second row shows the results of segmentation obtained from the standard segmentation network.

Table 1: Segmentation error of the proposed method ($T = 2$, $T = 3$, $T = 4$) and the existing method (A^2Net) under small, medium and large image distortions.

	$T = 2$	$T = 3$	$T = 4$	A^2Net
small	0.0803	0.0776	0.0748	0.1286
medium	0.1456	0.1407	0.1293	0.1773
large	0.1851	0.1508	0.1456	0.2266

Table 2: LPIPS of the proposed method ($T = 2$, $T = 3$, $T = 4$) and the existing method (A^2Net) under small, medium and large image distortions.

	$T = 2$	$T = 3$	$T = 4$	A^2Net
small	0.0322	0.0291	0.0268	0.1990
medium	0.1181	0.1145	0.0927	0.1992
large	0.1179	0.1118	0.1294	0.2312

tions in input raindrop images. Nonetheless, the proposed method reduced the image distortion drastically as shown in Fig. 8 (c), (d) and (e). Also, we find that less distorted images were generated as we increase the number of time instants T in the image sequence.

Fig. 9 shows results under large distortions. As shown in Fig. 9 (b), it is almost impossible to recognize original vehicles and buildings in the input images. Nonetheless, the proposed method recovered fairly good road scene images with vehicles and buildings. Again, image quality increases as we increase the number of time instants T.

5.2 Comparison with Conventional Methods

We next compare our method with the existing state-of-the-art raindrop removal method. In this experiment, we compared our method with A^2Net (Qian et al., 2018) proposed by Qian et al. Their method also uses deep neural network for removing raindrops in images. However, their method assumes that the raindrops in images are not so many, and the background scene is occluded only partially in images. We tested these two methods under heavy rain situations. The first row of Fig. 10 shows the raindrop removal images generated from the proposed method and A^2Net under heavy rain situations. As shown in these images, A^2Net cannot recover raindrop removal images properly, while the proposed method provides us fairly good results.

For comparing the accuracy of raindrop removal images numerically, we computed LPIPS (Zhang et al., 2018), which can measure semantic similarity of two images. Unlike the traditional error metric, such as RMSE and SSIM, it has been confirmed by many authors that LPIPS provides us image similarity that matches the human sense. We computed LPIPS between the ground truth images and raindrop removal images generated by our method and A^2Net. The results are shown in Table 2. As shown in this table, LPIPS of our method decreases as we increase the number of time instants used in our method. We can also find that our method provides us much better raindrop removal images than A^2Net.

For evaluating the visibility of objects in the recovered images, we also evaluated our method by performing the semantic segmentation on the recovered images using a pretrained semantic segmentation network. The semantic segmentation network used in this evaluation is pix2pix (Isola et al., 2017). For comparing the segmentation accuracy numerically, the segmentation error was computed by normalizing the total number of wrongly segmented pixels by the number of total pixels N_p in an image multiplied with

(a) input images

(b) results from the proposed method

(c) results from A^2Net

Figure 11: Results from real image experiment. (a) shows images observed under heavy rain. (b) and (c) show raindrop removal images obtained from images in (a) by using the proposed method and A^2Net respectively. The visibility in (b) is better than that in (a) and (c). For example, the white lane markers and the road guardrails are heavily distorted in (a) and (c), but they look more accurate in (b).

the number of images N_i as follows:

$$error = \frac{1}{N_i \cdot N_p} \sum_{i=1}^{N_i} \sum_{j=1}^{N_p} wrong(i,j) \tag{10}$$

$$wrong(i,j) = \begin{cases} 0: & \text{pixel } j \text{ of image } i \text{ is} \\ & \text{segmented correctly} \\ 1: & \text{others} \end{cases} \tag{11}$$

The second row in Fig. 10 show the results of semantic segmentation performed on raindrop removal images recovered from our method and A^2Net under large image distortions. As shown in Fig. 10 (f), the segmentation result of A^2Net is very different from that of the ground truth in (a). However, that of the proposed method is very close to the ground truth as shown in Fig. 10 (c), (d) and (e). Moreover, it can be seen that the quality of the raindrop removal image improves as we increase the number of time instants used in the proposed method.

The improvement of accuracy in the proposed method can be seen more clear in Table 1, which shows the segmentation error defined in Eq. (10). As we can see in this table, the proposed method outperforms A^2Net in all cases of small, medium and large distortions. We can also find that the accuracy improves as we increase the number of time instants in our method.

From these results, we find that the proposed method is very efficient for recovering images which are distorted by heavy rain.

5.3 Real Image Experiments

Up to now, we tested our method by using synthetic heavy rain images. For evaluating the efficiency of our method, we next apply our method to real heavy rain images.

Fig. 11 (a) shows input images of various road scenes observed under heavy rain. As we can see in these images, the road scenes are heavily distorted due to the rain water film, and driving with these images is very difficult. Fig. 11 (b) shows raindrop removal images obtained from the proposed method, and Fig. 11 (c) shows those obtained from A^2Net.

From these results, we find that both methods cannot remove image distortions perfectly, but the proposed method provides us better views of the road scene than A^2Net.

Since the ground truth undistorted images of these scenes are not available, we cannot evaluate the accuracy of these generated images numerically. However, we can still find that the visibility of the scene in (b) is better than that in (a) and (c). For example, the white lane markers and the road guardrails are heavily distorted in (a) and (c), but they look more accurate in (b).

Although, our method outperforms the existing state-of-the-art method, we also find that our method is not perfect, and we need more improvements. In particular, the heavy rain model used to generate the training dataset needs to be improved. A more accurate heavy rain model would lead to more accurate raindrop removal.

6 CONCLUSIONS

In this paper, we proposed a new method for removing image distortion caused by raindrops under heavy rain.

In heavy rain, raindrops form a non-uniform film on the windshield, and the visibility for a driver degrades drastically. The existing raindrop removal methods cannot recover clear images in such situations, since these methods assume that the background scene is visible through the gap between the raindrops, which does not happen anymore in heavy rain. Thus, we in this paper proposed a new method for recovering raindrops removal images from the series of distorted images. The results of our experiments show that the proposed method outperforms the state-of-the-art raindrop removal method in heavy rain situations.

The proposed method is promising, but challenges remain. In our proposed method, image degradation due to raindrops is considered. However, in actual heavy rains, image degradation due to rain streaks also exists, so it is desirable to expand to a method that improves both of these degradations.

Furthermore, it is also important to use real heavy rain images to train the network for improving the accuracy. Since the ground truth undistorted images are not available under heavy rain, we need to consider unsupervised leaning in the raindrop removal framework.

REFERENCES

Chen, Y. and Hsu, C. (2013). A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proc. of International Conference on Computer Vision*, pages 1968–1975.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fu, X., Huang, J., Ding, X., Liao, Y., and Paisley, J. (2017). Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956.

Garg, K. and Nayar, S. (2004). Detection and removal of rain from videos. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134.

Li, Y., Tan, R., Guo, X., Lu, J., and Brown, M. (2017). Single image rain streak separation using layer priors. *IEEE Transactions on Image Processing*.

Luo, Y., Xu, Y., and Ji, H. (2015). Removing rain from a single image via discriminative sparse coding. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 3397–3405.

Matsui, T., Sakaue, F., and Sato, J. (2014). Raindrop removal by using camera array system. In *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2249–2250. IEEE.

N., B. and N., L. (2008). Using the shape characteristics of rain to identify and remove rain from video. In *Proc. of Joint International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition*, volume LNCS5342, pages pp 451–458.

Nomoto, K., Sakaue, F., and Sato, J. (2011). Raindrop complement based on epipolar geometry and spatiotemporal patches. In *Proc. of International Conference on Computer Vision Theory and Applications*, pages 175–180.

Qian, R., Tan, R. T., Yang, W., Su, J., and Liu, J. (2018). Attentive generative adversarial network for raindrop removal from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Santhaseelan, V. and Asari, V. (2014). Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, pages 1–19.

Yamashita, A., Fukuchi, I., and Kaneko, T. (2009). Noises removal from image sequences acquired with moving camera by estimating camera motion from spatiotemporal information. In *Proc. of International Conference on Intelligent Robots and Systems*, pages 3794–3801.

Yamashita, A., Tanaka, Y., and Kaneko, T. (2005). Removal of adherent waterdrops from images acquired with stereo camera. In *Proc. of International Conference on Intelligent Robots and Systems*, pages 400–405.

Yang, W., Tan, R., Feng, J., Liu, J., Guo, Z., and Yan, S. (2017). Deep joint rain detection and removal from a single image. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*.

You, S., Tan, R., Kawakami, R., Mukaigawa, Y., and Ikeuchi, K. (2016). Adherent raindrop modeling, detectionand removal in video. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1721–1733.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.