Dual-Pixel Raindrop Removal

Yizhou Li
yli@ok.sc.e.titech.ac.jp

Yusuke Monno
ymonno@ok.sc.e.titech.ac.jp

Masatoshi Okutomi
mxo@ctrl.titech.ac.jp

Tokyo Institute of Technology,
Tokyo, Japan

Abstract

Removing raindrops in images has been addressed as a significant task for various computer vision applications. In this paper, we propose the first method using a Dual-Pixel (DP) sensor to better address the raindrop removal. Our key observation is that raindrops attached to a glass window yield noticeable disparities in DP’s left-half and right-half images, while almost no disparity exists for in-focus backgrounds. Therefore, DP disparities can be utilized for robust raindrop detection. The DP disparities also bring the advantage that the occluded background regions by raindrops are shifted between the left-half and the right-half images. Therefore, fusing the information from the left-half and the right-half images can lead to more accurate background texture recovery. Based on the above motivation, we propose a DP Raindrop Removal Network (DPRRN) consisting of DP raindrop detection and DP fused raindrop removal. To efficiently generate a large amount of training data, we also propose a novel pipeline to add synthetic raindrops to real-world background DP images. Experimental results on synthetic and real-world datasets demonstrate that our DPRRN outperforms existing state-of-the-art methods, especially showing better robustness to real-world situations. Our source code and datasets are available at http://www.ok.sc.e.titech.ac.jp/res/SIR/.

1 Introduction

Raindrops are typically attached to a glass window or a windshield and refract the light from the scene similar to a fish-eye lens. Therefore, raindrops in images eliminate the original background textures on raindrop-covered regions, which greatly reduces the image visibility and may disturb many computer vision tasks, e.g., object detection [18], video surveillance [21], and autonomous driving [6]. To avoid these disadvantages, removing raindrops in images has been treated as one of the important low-level vision tasks.

Recently, many deep-learning-based methods have been proposed to address single image deraining, including rain streak removal [8, 16, 17, 29, 34, 35, 38] and raindrop removal [11, 26, 28, 32, 33, 43]. Regarding the raindrop removal, representative methods include raindrop-mask-guided methods [11, 26, 32, 33], an edge-guided method [28], and a Laplacian-pyramid-based method [43], as well as general image restoration frameworks [20, 39, 40] validated to be effective on the raindrop removal. Despite the fact that
these state-of-the-art single image raindrop removal methods achieve good performance on synthetic datasets, they often show degraded and limited performance on real-world data, as experimentally pointed out in the survey paper of [15], because of the domain gap between synthetic training data and real-world data. One inherent challenge of the single image raindrop removal is that raindrop detection and removal highly rely on raindrop appearance (e.g., raindrop textures and shapes) observed in a single image. Therefore, they inevitably fail when there are large appearance gaps of raindrops between the synthetic training data and the real-world data at an application phase.

In this paper, we focus on a Dual-Pixel (DP) sensor to better address the raindrop removal task. A DP sensor has been adopted in some consumer digital cameras (e.g., Canon 5D Mark IV and Canon EOS R5) and smartphones (e.g., Google Pixel series and Samsung Galaxy series), as it can be implemented without significantly increasing the cost. As shown in Fig. 1(a), a DP sensor divides each sensing pixel into two halves with left and right photodiodes, by which two individual images called left-half and right-half images can be captured. The summation of these two images, which we call a combined image, corresponds to the image captured by a regular sensor. As shown in Fig. 1(b), for an in-focus scene point within the Depth-of-Field (DoF), the light rays reach the same pixel. In contrast, for an out-of-focus scene point outside the DoF, the light rays reach different pixels, leading to an intensity shift on the horizontal axis between the left-half and the right-half images, which is called a DP disparity. The studies [3, 5, 25, 36] show that the DP disparity can be modeled as different Point Spread Functions (PSFs) of the left-half and the right-half images as shown in Fig. 1(b), where the left-half and the right-half images can be generated from an all-in-focus image by convolution using different disk blur kernels corresponding to their varied PSFs. Recent studies have also demonstrated that the DP disparity is useful for a wide range of applications, such as defocus deblurring [3, 5, 36], autofocusing [2, 4, 13], depth estimation [9, 25, 42], joint deblurring and depth estimation [22], and reflection removal [24]. However, to the best of our knowledge, there is no existing study that adopts a DP sensor for the raindrop removal. Although some of recent rain streak or raindrop removal studies exploit multi-view observations in a stereo or light-field setup [7, 37, 41], the DP sensor has the advantage that it can capture two perfectly aligned and synchronized images in one shot.

A key observation to motivate us to adopt a DP sensor is that, in practical applications requiring raindrop removal such as autonomous driving, the camera is usually background-focused, while the raindrops which are closer to the camera are out-of-focus. Therefore, as shown in Fig. 1(d), the raindrop regions exhibit noticeable DP disparities while other background regions show almost no DP disparity. This suggests that we can utilize DP disparities, instead of solely depending on the raindrop appearance, to detect raindrop locations. Since the DP disparities for raindrops bring slightly-shifted raindrop positions in the left-half and the right-half images, the non-raindrop-covered visible background textures in those images are also shifted. This suggests that we can exploit this redundancy to better restore the background details by fusing the information from the leaf-half and the right-half images.

With the above motivation, we propose DP Raindrop Removal Network (DPRRN), which consists of two main parts: DP raindrop detection and DP fused raindrop removal. Firstly, the DP raindrop detection part robustly predicts the raindrop masks on the left-half and the right-half images, according to the DP disparities of raindrops. Then, the DP fused raindrop removal part removes the raindrops with the predicted raindrop masks as guide, where we intermediately remove the raindrops from the left-half and the right-half images, respectively, and then fuse and refine those results to obtain the final raindrop removal result as a combined image, i.e., an image captured by a regular sensor. The networks for both parts are
Figure 1: (a) In a DP sensor (Google Pixel 4), each green pixel is split into two halves. (b) The light rays from an in-focus point reach the same pixel, while the light rays from an out-of-focus point reach different pixels in the left-half and the right-half images, producing a DP disparity. The corresponding point spread functions (PSFs), which are used to model the DP disparity, are shown in the right. (c) Raindrops are usually out-of-focus and exhibit DP disparities, while in-focus backgrounds show almost no DP disparity. This motivates us to exploit the DP disparity for raindrop detection and removal.

trained in an end-to-end manner.

Since existing raindrop removal datasets are not applicable to a DP sensor, we built the first DP raindrop removal dataset. To efficiently generate a large amount of training data, we present a novel pipeline to add synthetic raindrops to real-world background DP images. We captured 613 real-world scenes and generated 2,452 pairs of training data with synthetic raindrops. In addition, to test the generalization performance in real-world situations, we also collected a real-world dataset containing 82 DP image pairs with and without real raindrops, which contains raindrop patterns and background textures that are unseen in the training data. Main contributions of this work are summarized as follows:

• We propose the first DP raindrop removal network and dataset. Our network effectively utilizes DP disparities on raindrops for raindrop detection and removal.

• We experimentally demonstrate that our DP raindrop removal network achieves state-of-the-art performance and shows better robustness to real-world situations.

2 DP Raindrop Removal Datasets

2.1 Synthetic-Raindrop Dataset

It is laborious to collect a large number of real-world aligned image pairs with and without real raindrops. It is also hard to accurately simulate DP sensors’ inherent characteristics, such as uneven vignetting for the left-half and the right-half images. Therefore, to construct a training dataset for our network, we take a hybrid approach, which only synthesizes raindrops and adds them to real background DP images, which can be easily captured using a real DP camera as usual. In all our experiments, we used Google Pixel 4. Since Google Pixel 4 adopts the DP sensor architecture only to the green pixels, we used green-channel images for our dataset construction, though our dataset construction and network architecture can be easily extended to the RGB color domain.
Figure 2: Proposed DP synthetic-raindrop dataset generation: (a) The overall flow for the left-half image, which consists of raindrop generation, raindrop blurring, and alpha blending. (b) A pin-hole model to generate all-in-focus raindrops. (c) The raindrop blurring process with DP PSF modeling. The similar process is conducted for the right-half image.

Figure 2(a) shows the proposed data generation pipeline, where the processes for the left-half image are presented. The processes for the right-half image can similarly be performed. The pipeline consists of three main parts: all-in-focus raindrop generation, raindrop blurring, and alpha blending of synthetic raindrops and a real background.

The first part generates all-in-focus raindrops based on the raindrop refraction model of [36]. For simplicity, as shown in Fig. 2(b), we assume that the whole background has a constant depth $d_{scene}$ and the raindrops (glass) has another smaller constant depth $d$. We add all-in-focus raindrops to the real-world background DP image $B_l$ with ray tracing to generate the image with all-in-focus raindrops $I_l^{AIF}$. The locations of the raindrops are determined by a randomly assigned binary raindrop mask $M^{AIF}_l(x) = 1$ means that the pixel $x$ is a part of a raindrop region and $M^{AIF}_l(x) = 0$ means that the pixel is in background regions. We set the background depth $d_{scene} = 10m$ and randomly change the raindrop depth $d$ between $15cm$ and $25cm$.

The second part simulates DP disparities for the raindrops. For this purpose, we apply the spatially-varying DP PSF modeling in [36], where in total $6 \times 8$ PSF kernel shapes, denoted as $H^i_l$ for the left-half image, are calibrated for each sub-patch $i$, as shown in Fig. 2(c). To determine the scale of PSF kernels, we calculated the Circle of Confusion (CoC) radius $r = \frac{q}{2} \times \frac{s'}{2} \times \frac{d-s}{q}$, which corresponds to the PSF radius, where $q$ is the aperture diameter, $s'$ is the distance between the lens and the sensor, $s$ is the focus distance, and $d$ is the assumed raindrop depth. Specifically, using a thin lens model, $q$ is calculated as $q = \frac{f}{F}$ according to Google Pixel 4’s focal length $f$ and F-stop $F$, and $s'$ is calculated as $s' = \frac{fs}{s-f}$, where the focus distance is set as $s = d_{scene}$ because the camera is assumed to be background-focused. According to the calculated CoC radius, we re-scale the PSF kernels to derive $H^i_l$ in correct PSF size for the real camera.

To blur the all-in-focus raindrops with the above PSF kernels, the raindrops $R^i_l^{AIF}$ are extracted from $I_l^{AIF}$ as $R^i_l^{AIF} = M^{AIF}_l \otimes I_l^{AIF}$, where $\otimes$ denotes pixel-wise multiplication. Then, the raindrops $R^i_l^{AIF}$ are divided into $6 \times 8$ patches and convolved with the PSF kernels.

Thus, the raindrop image with DP PSF modeling is

$$I_l^{DP} = (1 - M^{AIF}_l) \cdot I_l^{AIF} + M^{AIF}_l \cdot R^i_l^{AIF}.$$
in a patch-by-patch manner as $\tilde{R}_{l}^{blur|i} = H_{i}^{l} \ast R_{AiFR}^{i} | l$, where $\ast$ denotes the convolution operation. The sub-patches are then stitched to form the blurred raindrop image $\tilde{R}_{l}^{blur}$. Similarly, the binary raindrop mask $M_{AiFR}$ is also blurred to generate the blurred raindrop mask $\tilde{M}_{l}^{blur}$, which is used as the weight for the alpha-blending in the next step.

The final part performs the alpha blending of the synthetic raindrops and the real-world background as $I_{l} = \tilde{M}_{l}^{blur} \otimes \tilde{R}_{l}^{blur} + (1 - \tilde{M}_{l}^{blur}) \otimes B_{l}$ to generate the left-half image $I_{l}$ with smooth and natural transitions from the raindrops to the background. The similar process is conducted to generate the right-half image $I_{r}$, where the corresponding right-half background image $B_{r}$ and DP PSF kernels $H_{r}$ are used for the same all-in-focus raindrop mask $M_{AiFR}$. For the combined images corresponding to the images captured by a regular sensor, we generate them by averaging the left and the right images as $I_{c} = \frac{I_{l} + I_{r}}{2}$ and $B_{c} = \frac{B_{l} + B_{r}}{2}$.

For each real-world background DP image pair, we generate 4 different synthetic-raindrop image pairs by randomly changing the raindrop depth and mask. Each data contains the generated images of $I_{l}, I_{r}, I_{c}$ and the ground-truth background images of $B_{l}, B_{r}, B_{c}$. We captured 613 scenes and generated 2,452 image pairs in total, among which 1,960 pairs are used for training and the rest 492 pairs are used for testing. Two samples are shown in Fig. 3(a), where noticeable DP disparities exist in the raindrop regions.

### 2.2 Real-World Dataset

To evaluate the generalization performance, we also collected 82 real-world DP image pairs with and without real raindrops. To minimize the impact of background refraction changes, we only used one glass panel and fixed it at a position of 15-25cm from the camera. Then, we first captured a DP image pair with the clean panel as the ground-truth images. Then, we sprayed water onto the panel to generate real raindrops and captured another DP image pair as the input images. To reduce the impact of motion, we used a solid tripod to fix the smartphone and the panel, and remotely controlled the smartphone to shoot images. As shown in Fig. 3(b), we carefully collected 82 pairs of high-quality and well-aligned DP images with and without real raindrops, containing varied and complex raindrop patterns.
3 Proposed DP Raindrop Removal Network

Figure 4 shows the overview of our proposed DP Raindrop Removal Network (DPRRN). We input the pair of the left-half and the right-half images \( \{I_l, I_r\} \) and estimate the clean background image \( \hat{B}_c \) as close as the ground truth \( B_c \), where the final output \( \hat{B}_c \) is a combined image as captured by a regular sensor. Our network consists of two parts: (i) DP raindrop detection, where we conduct raindrop detection on \( I_l \) and \( I_r \), respectively, and (ii) DP fused raindrop removal, where we remove raindrops on \( I_l \) and \( I_r \), respectively, whose results are subsequently fused and refined for the final raindrop removal result. Each network is constructed using UNet \([31]\) with residual blocks \([12]\), for which we refer to the supplementary material. All the networks are trained in an end-to-end manner, as detailed below.

### 3.1 DP Raindrop Detection

In DP raindrop detection, the input images \( I_l \) and \( I_r \) are channel-wise concatenated and sent to two different networks \( G_{\text{mask}}^l \) and \( G_{\text{mask}}^r \) to predict two raindrop masks \( \hat{M}_l \) and \( \hat{M}_r \) depicting the raindrop locations of \( I_l \) and \( I_r \), respectively. For the left-half mask \( \hat{M}_l \), the concatenated \( I_l \) and \( I_r \) are sent to \( G_{\text{mask}}^l \) to predict a pixel-wise soft raindrop mask \( \hat{M}_l \) through Sigmoid activation function at the last layer. In this process, \( G_{\text{mask}}^l \) performs raindrop localization guided with the DP disparities on the raindrops. Similarly, for the right-half mask \( \hat{M}_r \), another network \( G_{\text{mask}}^r \) is utilized to predict a pixel-wise soft raindrop mask \( \hat{M}_r \) through Sigmoid activation function at the last layer. In this process, \( G_{\text{mask}}^r \) performs raindrop localization guided with the DP disparities on the raindrops. Similarly, for the right-half mask \( \hat{M}_r \), another network \( G_{\text{mask}}^r \) is utilized to predict a pixel-wise soft raindrop mask \( \hat{M}_r \) through Sigmoid activation function at the last layer. Here, \( G_{\text{mask}}^l \) and \( G_{\text{mask}}^r \) are two individual networks that do not share the parameters because the disparity of \( I_r \) to \( I_l \) and the disparity of \( I_l \) to \( I_r \) show inversed shift directions and we have found using two individual networks for them achieves more robust performance.

As for the loss function on the raindrop detection, we use the Binary Cross Entropy (BCE) \([10]\) losses as

\[
L_{\text{Mask}} = \text{BCE}(\hat{M}_l, M_l) + \text{BCE}(\hat{M}_r, M_r),
\]

where \( M_l \) and \( M_r \) are the ground-truth binary raindrop masks, where the value one means that the corresponding pixel is a part of the raindrop regions of \( I_l \) and \( I_r \), respectively.

### 3.2 DP Fused Raindrop Removal

In DP fused raindrop removal, we first conduct raindrop removal of DP images \( I_l \) and \( I_r \) separately, with the guide of the estimated masks \( \hat{M}_l \) and \( \hat{M}_r \) as raindrop location clues.
For the rain removal network $G_l^{dlr}$, we input the left-half image $I_l$ concatenated with the mask $\hat{M}_l$ to derive the raindrop residuals $\hat{R}_l$, from which the left-half background image is predicted as $\hat{B}_l = I_l - \hat{R}_l$. The raindrop removal of the right-half image $I_r$ is performed in the same manner using another network $G_r^{dlr}$ to predict the right-half background image $\hat{B}_r$. Here, $G_l^{dlr}$ and $G_r^{dlr}$ are also two individual networks without sharing the parameters to address the different blur kernels applied in $I_l$ and $I_r$.

In the above processes, the background DP images $\hat{B}_l$ and $\hat{B}_r$ are predicted through $G_l^{dlr}$ and $G_r^{dlr}$ basically in a single image input manner. However, as the raindrop locations and the visible background information in $I_l$ and $I_r$ are slightly different according to the DP disparities, the restored background details in $\hat{B}_l$ and $\hat{B}_r$ are varied as well. This suggests the potential to derive a more accurate restoration result by fusing and refining the results of $\hat{B}_l$ and $\hat{B}_r$.

We then apply another network $G_{fuse}$ for fusing the raindrop removal results $\hat{B}_l$ and $\hat{B}_r$. Because our target is to predict a clean background image as captured by a regular sensor, we calculate the corresponding mask as $\hat{M}_c = \max(\hat{M}_l, \hat{M}_r)$ to derive a pixel-wise soft raindrop mask to the combined image $I_c$, where $\max$ denotes the pixel-wise maximum operation. We also calculate the initial combined background image as $\hat{B}_c^{init} = \frac{\hat{B}_l + \hat{B}_r}{2}$. Then, the channel-wise concatenation of $\hat{M}_c, \hat{B}_l, \hat{B}_r, \hat{B}_c^{init}$ is sent to $G_{fuse}$ to derive the residuals $\hat{R}_c^{refine}$ in terms of $\hat{B}_c^{init}$, from which the final output background image is derived as $\hat{B}_c = \hat{B}_c^{init} - \hat{R}_c^{refine}$.

The DP fused raindrop removal is optimized using negative SSIM losses [29, 30] to maximize SSIMs between the raindrop removal results and the ground-truth clean background images as

$$L_{dlr} = -SSIM(\hat{B}_l, B_l) - SSIM(\hat{B}_r, B_r) - SSIM(\hat{B}_c, B_c),$$

which is the simple summation of three losses for each raindrop removal output.

Combined with the DP raindrop detection and the DP fused raindrop removal, the whole DPRRN is trained using the simple summation of the mask detection loss and the raindrop removal loss as

$$L = L_{mask} + L_{dlr}.$$ 

### 4 Experimental Results

#### 4.1 Implementation Details

Our DPRRN is implemented using Pytorch [23] and trained on a single NVIDIA RTX3090 GPU. During the training, we use randomly cropped $480 \times 120$ patches with the batch size set to 12. To ensure that the shift directions of raindrops in the DP image pair do not change, we do not apply random flipping for data augmentation. ReLU [14] is set as the activation function. RAdam [19] is used as the optimizer to train the network with an initial learning rate of $1e^{-3}$. We adopt two-stage training, where we first train DP raindrop detection with $L_{mask}$ only for 100 epochs without changing the learning rate and then use all the losses $L = L_{mask} + L_{dlr}$ to train the whole DPRRN in end-to-end for another 400 epochs, during which the learning rate is decayed by multiplying 0.2 at 120, 240, and 360 epochs.
Table 1: Quantitative comparison on our datasets. (Red: rank 1st; Blue: rank 2nd)

<table>
<thead>
<tr>
<th>Input Image Types</th>
<th>Methods</th>
<th>Synthetic</th>
<th>Real-World</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td>Regular</td>
<td>AttGAN (CVPR2018) [26]</td>
<td>34.55</td>
<td>0.9614</td>
</tr>
<tr>
<td></td>
<td>CCN (CVPR2021) [27]</td>
<td>38.88</td>
<td>0.9790</td>
</tr>
<tr>
<td></td>
<td>MPRNet (CVPR2021) [39]</td>
<td>41.02</td>
<td>0.9828</td>
</tr>
<tr>
<td></td>
<td>Restormer (CVPR2022) [40]</td>
<td>41.94</td>
<td>0.9845</td>
</tr>
<tr>
<td></td>
<td>DGUNet (CVPR2022) [20]</td>
<td>41.32</td>
<td>0.9835</td>
</tr>
<tr>
<td></td>
<td>RDPD+ (ICCV2021) [5]</td>
<td>40.51</td>
<td>0.9809</td>
</tr>
<tr>
<td></td>
<td>DPRRRN−RD (Ours)</td>
<td>40.93</td>
<td>0.9824</td>
</tr>
<tr>
<td></td>
<td>DPRRRN (Ours)</td>
<td>42.70</td>
<td>0.9867</td>
</tr>
</tbody>
</table>

Figure 5: Qualitative comparison on the synthetic-raindrop dataset.

4.2 Comparison with State-of-the-Art Methods

**Compared Methods:** We compared AttGAN [26] and CCN [27], which are the methods designed for single image raindrop removal. Since AttGAN has not released the official training code, we referred to the past practice and used a third-party repository [1] to train AttGAN on our dataset. CCN is proposed to remove rain streaks and raindrops in one go. Since our current target is raindrop removal only, we took the raindrop removal part of CCN for training and comparison. We also compared general image restoration methods, MPRNet [39], DGUNet [20], and Restormer [40], which are very recent state-of-the-art methods. Although these methods are not specifically designed for raindrop removal, they all contain comparative experiments on raindrop removal and demonstrate their effectiveness. For all the above single image methods, the input is a regular single image, which is calculated as $I_c = \frac{1}{2}I_l + \frac{1}{2}I_r$, and the output target is a clean background image in a regular sensor domain, which is supervised by the ground-truth background image calculated as $B_c = \frac{B_l + B_r}{2}$. We used their default parameter settings to train the networks.

As a DP-based method, a recent DP-based defocus deblurring method RDPD+ [5] was compared to validate the effectiveness of our network design utilizing DP images. The input to both RDPD+ [5] and proposed DPRRN is a DP image pair $I_l$ and $I_r$, and the final output is also supervised by a clean background image $B_c$ in a regular sensor domain. Since RDPD+ does not have specific design for defocusing deblurring, it can be directly trained on our DP synthetic-raindrop dataset. As an ablation study, we also compared our DPRRN without the DP raindrop detection part, which is denoted as DPRRN−RD, where the only DP fused raindrop removal part was trained without the guide of the raindrop masks.

**Results on Synthetic-Raindrop Dataset:** We here show the results, where both the training and the testing were performed on the synthetic-raindrop dataset as detailed in Sec. 2.1. The second-right part of Table 1 shows the quantitative results with PSNR and SSIM. The results show that our DPRRN achieves the best performance on the synthetic data. Compared
with the past single image methods, our DPRRN achieves PSNR and SSIM improvements of 0.76dB and 0.0022, respectively. Compared with the DP-based method of RDPD+, our DPRRN also achieves PSNR and SSIM improvements with the large margins of 2.19dB and 0.0043, respectively. In comparison of DPRRN$^{-\text{RD}}$ and DPRRN, we can confirm that the DP raindrop detection part significantly contributes to the performance improvement. These results validate that our DPRRN architecture can effectively exploit DP information to better remove raindrops. Figure 5 shows the qualitative results on the synthetic dataset. For the raindrops in the red box, our DPRRN can restore the thin details of window frames, while the other methods exhibit over-smoothing results. For the raindrops in the green box, our DPRRN can recover sharper details around the branches compared with the other methods.

**Results on Real-World Dataset:** We next show the results, where the networks were trained using the synthetic-raindrop dataset and then tested on the real-world dataset as detailed in Sec. 2.2. The rightmost part of Table 1 shows PSNR and SSIM results, demonstrating that our DPRRN exhibits large PSNR and SSIM margins of 1.29dB and 0.0137 compared with the best-performed existing methods, respectively. These margins on the real-world data are larger than those on the synthetic data (i.e., 0.76dB and 0.0022), demonstrating higher generalization ability of our DPRRN. Although MPRNet, Restormer, and DGUnet show better results than DPRRN$^{-\text{RD}}$ on the synthetic data, they show worse results than DPRRN$^{-\text{RD}}$ on the real-world dataset. This indicates that, even without the raindrop detection part, our DPRRN shows better robustness to real-world situations.

Figure 6 shows the qualitative results on the real-world dataset. The existing methods have a large number of remaining raindrops, while our DPRRN is able to remove those raindrops and restore clean and sharp background details. Especially for the red and the green box regions, our DPRRN is the only method to successfully detect and remove the long flowing raindrops, even though such a complex raindrop pattern is not included in the training data. This further validates the strong robustness of our DPRRN. More results on both the synthetic and the real-world datasets can be seen in our supplementary material.
5 Conclusion

In this paper, we have proposed the first DP-based raindrop removal network, named DPRRN, consisting of DP raindrop detection and DP fused raindrop removal. In DPRRN, we have utilized DP disparities existing in raindrop regions for robust raindrop detection and fine background detail recovery. We have also constructed both synthetic and real-world DP raindrop removal datasets to train and test our DPRRN. Using those datasets, we have experimentally demonstrated that our DPRRN outperforms existing state-of-the-art methods, showing better generalization ability to real-world raindrops. One of our future work is to jointly address rain streaks and raindrops to further boost the real-world applicability.

Acknowledgement: This work was supported by JST SPRING, Grant Number JPMJSP2106.

References


