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 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, the DP disparities can be utilized for robust raindrop detection. The DP disparities also bring the advantage that the occluded background regions by raindrops are slightly 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 constructed synthetic and real-world datasets demonstrate that our DPRRN outperforms existing state-of-the-art methods, especially showing better robustness to real-world situations.

Index Terms—Raindrop removal, dual-pixel sensor, deep learning.

I. 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 [1], video surveillance [2], and autonomous driving [3]. 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 [4], [5], [6], [7], [8], [9], [10] and raindrop removal [11], [12], [13], [14], [15], [16]. Regarding the raindrop removal, representative methods include raindrop-mask-guided methods [11], [13], [15], [16], an edge-guided method [12], and a Laplacian-pyramid-based method [14], as well as general image restoration frameworks [17], [18], [19]. Despite the fact that these state-of-the-art single-image 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 [20], because of the domain gaps between synthetic training data and real-world data. One inherent limitation of 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 in real situations when there are large appearance gaps between the raindrops in the synthetic training data and the real-world testing data.

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 manufacturing cost. As shown in Fig. 1(a), a DP sensor divides each 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.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
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. First, 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 a 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 result as a combined image, i.e., an image captured by a regular sensor. The networks for both parts are 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. Our dataset considers two common setups for raindrop removal: a vertical setup (e.g., building windows) and a tilted setup (e.g., car windshields). To efficiently generate a large amount of training data, we conducted a novel pipeline to add synthetic raindrops to real-world background DP images based on spatially-varying point spread function (PSF) modeling for a DP sensor [24], [25], [26], [28]. In addition, to test the generalization performance in real-world situations, we collected a real-waterdrop dataset that has ground-truth clean images and rainy images with imitated raindrops by a sprayer. The real-waterdrop dataset is for both quantitative and qualitative evaluation, which contains raindrop patterns and background textures that are unseen in the synthetic training data. Additionally, to further validate real-world robustness, we collected a real-car dataset for qualitative evaluation consisting of real rainy images featuring real raindrops, captured inside a moving car on a rainy day. The main contributions of this work are summarized as follows:

- We propose the first DP raindrop removal network. Our network contains a synthetic-raindrop dataset for network training and a real-waterdrop dataset for real-world evaluation quantitatively and qualitatively. Both datasets consider two common setups for glass windows: vertical and tilted setups. We also provide a real-car dataset captured in a real car on a rainy day for further verification.
- We experimentally demonstrate that our DP raindrop removal network achieves state-of-the-art performance quantitatively and qualitatively. Remarkably, the proposed network shows substantially better robustness to real-world situations compared with previous state-of-the-art methods.

A preliminary version of this work was published in BMVC2022 conference proceedings [32], which exclusively focused on the vertical setup of the glass window. In this extended version, we have broadened the scope of our study by addressing another common configuration for the glass window, namely the tilted setup, which is prevalent in everyday automobiles. To this end, we restructured the synthetic-raindrop dataset generation pipeline to account for the spatially-varying glass depth. Furthermore, we have reacquired the real-waterdrop dataset for both the vertical and the tilted setups, utilizing new waterdrop spraying techniques to improve the quality and the realism of the imitated raindrops. We have also collected a new real-car dataset to facilitate the evaluation of raindrop removal in actual driving scenarios. Using those new datasets, we have conducted a more comprehensive ablation study and experimental comparison, where two stereo-camera-based methods are newly included as the closest methods to ours.

II. RELATED WORK

A. Single-Image Deraining

Rain, including rain streaks and raindrops, negatively impacts images and videos by reducing visibility and degrading textures.
Consequently, various efforts have been made to restore clean backgrounds from rainy images for both rain streaks [4], [5], [6], [7], [8], [9], [10] and raindrops [11], [12], [13], [14], [15], [16], [33], [34], [35]. Current single-image raindrop removal methods fall into two categories: prior-based and deep-learning-based methods. Prior-based methods [33], [34], [35] focus on raindrop detection using human-defined priors, where detected raindrop regions are wiped out and inpainted. On the other hand, deep-learning-based methods [11], [12], [13], [14], [15], [16] learn direct mappings from rainy to clean images, demonstrating higher performance on various benchmarks. However, single-image learning-based methods struggle with high dependence on raindrop appearance, leading to limited robustness to real-world data with various sizes and shapes of the raindrops.

A prevalent approach to improving deep-learning-based raindrop removal is to detect the raindrop regions first and use them as a guide to the subsequent removal network. The method [11] predicts a raindrop mask using a long short-term memory network. The method [12] detects the raindrops with an edge map and an image torque operator. Instead of a raindrop mask, the method [15] predicts a blurring level map with the value from −1 to 1, which indicates the blurring level of the raindrops. The motivation behind these methods is similar to our method. However, they still solely rely on raindrop appearance for raindrop detection, resulting in limited robustness. In contrast, our method additionally employs DP disparity, which is a consistent physical property across synthetic and real-world data to make the raindrop detection more effective.

Regarding raindrop removal datasets, synthetic datasets have been built by the method [16], [36], which use ray tracing for raindrop rendering given the background image and depth. However, the method [36] assumes a pinhole camera model without blurring of raindrop regions, which is not sufficiently realistic. Although the method [16] adds blurs to the raindrops, it uses a constant blur size for the whole image, without considering that the raindrop depth may vary when the glass panel is tilted. Alternatively, the study [11] constructed a real-world dataset with paired rainy and clean background images for network training, captured by spraying waterdrops onto a glass panel. Although this real-world dataset is valuable, it is not directly applicable to our DP-based method, as we require DP raindrop images with two photodiodes.

### B. Image Deraining Using Multi-View Observations

Although no existing study has explored the use of a DP sensor for image deraining, some recent learning-based methods have utilized multi-view observations in a stereo camera setup [37], [38] or a light-field camera setup [39], [40] targeting for rain streaks [37], [40] or raindrops [38], [39].

In a stereo camera setup, the method [37] exploits the differences of degraded regions by rain streaks in the left and the right stereo images due to varying spatial locations of rain streaks for the left and the right views. This method also combines the stereo observations and extracted semantic information to enhance its performance. Similarly, the method [38] utilizes stereo observations for rain/waterdrop removal and introduces a novel row-wise dilated attention module to increase the receptive field for efficient information propagation between stereo images. It also incorporates an attention consistency loss to enhance left-right stereo consistency. Even though our DP-based method handles the left-half and the right-half DP images as stereo images, the DP sensor has the advantage that it is a compact all-in-one sensor without the necessity of any geometric calibration. Moreover, as introduced in Fig. 1, since the DP disparities are more focused on the raindrops than stereo disparities, where apparent disparities exist also in background regions, the DP disparities are particularly beneficial for robust raindrop detection and removal.

In a light-field camera setup, the study [40] observes that rain streaks exhibit different slopes and/or chromatic values compared to background scenes along the epipolar plane images and thus uses them to detect rain streaks and restore the background. The method [39] uses a depth map estimated from the light field image to identify raindrop regions, as they exhibit smaller depths compared to background areas. Although a light-field camera has the benefit of providing multi-view observations of more than two views, it makes the entire system complicated. Also, a light-field camera often faces low-resolution issues.

### C. Applications of Dual-Pixel Sensor

As introduced in Section I, the most important feature of a DP sensor is the DP disparity existing in out-of-focus regions. Recent studies have demonstrated that the DP disparity is useful for a wide range of applications, such as autofocusing [21], [22], [23], defocus deblurring [24], [25], [26], depth estimation [27], [28], [29], [30], and reflection removal [31]. For autofocusing, the DP disparity facilitates phase detection across the entire image for fast autofocus. For defocus deblurring, the DP image pair provide a light field of two views, which assists in better reducing defocus blur. For depth estimation, the DP disparity is estimated from the DP image pair and converted to the depth, which is related to the distance from the camera’s focus distance. For reflection removal, the DP disparity is used to detect reflection components while leveraging shifted background contents. Although the reflection removal method of [31] shows a similar motivation to our method, to the best of our knowledge, no existing study employs a DP sensor for raindrop removal.

Another class of studies for a DP sensor is the modeling of DP sensor optics. The studies [24], [25], [26], [28] indicate that the DP disparity can be modeled by different PSFs for the left-half and the right-half images, where the DP disparity can be synthesized from an all-in-focus image by convoluting it with disk blur PSF kernels varied for the left-half and right-half images. We adopt the PSF modeling of [26] to generate a training dataset for our raindrop removal task.

### III. PROPOSED DATASETS

#### A. Synthetic-Raindrop Dataset

1) **Motivation and Overview:** It is impractical to collect a large number of real-world aligned image pairs with and without

---

*Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.*
real raindrops. It is also hard to accurately simulate a real DP sensor’s inherent characteristics, such as uneven vignetting for the left-half and the right-half images. Therefore, to efficiently construct a training dataset for our DP-based network, we propose a hybrid approach that only synthesizes raindrops and adds them to real background DP images. As the background images without raindrops can be easily captured using a real DP camera, the amount of data can be easily enlarged. In all our experiments, we used Google Pixel 4. Since Google Pixel adopts the DP sensor architecture only to the green pixels, we used green-channel images, though our dataset construction and network architecture can be easily extended to the RGB color domain.

Fig. 2 shows the proposed data generation pipeline, where the processes for the left-half image are presented. The same processes are applied to the right-half image. The pipeline consists of three main parts: all-in-focus raindrop generation, raindrop blurring, and alpha blending. In our study, we used the real specifications of Google Pixel 4 for generating synthetic raindrops. Concretely, the focal length was set at 4.41 mm, equivalent to 27 mm for a full-frame camera. The sensor width is 5.64 mm and the F-stop was fixed at 1.7. We maintained the focus distance at 10m. To enhance training and testing efficiency, the images were scaled to achieve a sensor horizontal resolution of 800 pixels. Each part of the pipeline for the left-half image generation is detailed below.

2) All-in-Focus Raindrop Generation: The first part generates all-in-focus (AiF) raindrops based on the raindrop refraction model of [16]. As shown in Fig. 3, we consider two common setups for glass: vertical and tilted setups. For simplicity, we assume that the whole background has a constant depth $d_{\text{scene}}$.

We define the glass depth $d_0$ as the distance between the optical center and the intersection point of the glass and the optical axis. The tilt angle of the glass is denoted as $\theta$ for the tilted setup. We add AiF raindrops to the real-world background DP image $B_l$ with ray tracing to generate the image with AiF raindrops $I^\text{AiF}_l$. The sizes and locations of the raindrops on the glass are randomly determined in the 3D space. As a result, the corresponding 2D binary raindrop mask $M^\text{AiF}$ is obtained, where $M^\text{AiF}(x) = 1$ means that the pixel $x$ is part of a raindrop region and $M^\text{AiF}(x) = 0$ means that the pixel is in the background. We set the background depth as $d_{\text{scene}} = 10$ m. For the vertical setup, we set the glass depth as $d_0 \in [15 \text{ cm}, 25 \text{ cm}]$, according to our previous work [32]. For the tilted setup, we set the tilt angle as $\theta \in [30^\circ, 45^\circ]$ and the glass depth as $d_0 \in [5 \text{ cm}, 25 \text{ cm}]$ to adapt to assumed tilted situations evaluated in this paper.

3) Raindrop Blurring: The second part simulates DP disparities for AiF raindrops based on a thin lens model. For this purpose, we apply a spatially-varying DP PSF modeling in [26], where in total 6 $\times$ 8 PSF kernel shapes, denoted as $H_l$ for the left-half image, are calibrated for each sub-patch $i$ of Google Pixel 4 sensor, as shown in Fig. 4(b). However, the kernels in [26] are calibrated only for front-focus scenarios. Applying these kernels directly in our back-focus assumption would lead to an opposite disparity direction in our DP raindrop generation. To address this, we swap the left and the right PSF kernels. This adjustment aligns the disparity direction in our synthetically generated images with that observed in real back-focus DP images. To determine the scale of each PSF kernel, we calculate the circle of confusion (CoC) size of a real camera as

$$CoC = q \times \frac{s'}{s} \times \frac{d - s}{d},$$

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 depth of the scene point to calculate the CoC size (see Fig. 4(a) for illustration). The calculated CoC size corresponds to the radius of each PSF kernel.

As we consider the raindrop blurring, we focus on the raindrop depth, i.e., the glass depth for CoC size calculation. For the vertical setup of Fig. 3, we simply set a constant depth $d =$
$d_0$ for calculating the CoC size of all the PSF kernels. For the tilted setup of Fig. 3, the glass depth varies with the raindrop location. Because calculating the glass depth separately for each raindrop pixel is slow and inefficient, we split the image into six rows, which correspond to the number of the PSF kernels in the vertical direction, and assume that the raindrop pixels on the same row share the same depth value. With this assumption, we only need to calculate the depth values for each row as $d_1, d_2, \ldots, d_6$. Specifically, as illustrated in Fig. 4(a), we can calculate these depth values as

$$d_i = \begin{cases} 
  d_0/(1 + \tan(\frac{s}{r})/\tan \theta), & \text{if } i = 1, \\
  d_0/(1 - \tan(\frac{s}{r})/\tan \theta), & \text{if } i = 6, \\
  d_1 + (i - 1) \times \frac{(d_6 - d_1)}{5}, & \text{if } i = 2, 3, 4, 5,
\end{cases}$$

where $\tau$ is the sensor’s vertical field-of-view (FOV) which is 30° in our experiments. $\theta$ is the tilt angle of the glass. Then, the CoC size for each depth is calculated as

$$CoC_i = q \times \frac{s'}{s} \times \frac{d_i - s}{d_i}. \tag{3}$$

As for the other parameters, using the same thin lens model, $q$ is calculated as $q = \frac{f}{\tau}$, according to Google Pixel 4’s focal length $f$ and F-number $F$, and the lens-to-sensor distance $s'$ is calculated as $s' = \frac{fs}{r}$, where the focus distance is set as $s = d_{\text{scene}}$ because the camera is assumed to be background-focused. According to the calculated CoC size, we re-scale each PSF kernel to derive $H_i$ to the correct PSF size for the real camera. As shown in Fig. 4(b), for the tilted setup, the size of each re-scaled PSF kernel becomes gradually smaller from top to bottom, as the depth becomes gradually closer to the focus distance $s$.

We here further clarify the disparity level with current glass depth settings. Specifically, the DP disparity can be computed as the absolute difference in the geometric centers of the left and the right PSF kernels as

$$\text{disp} = |GC(H_l)_x - GC(H_r)_x|, \tag{4}$$

where $GC(H)$ calculates the geometric center of the kernel $H$ as

$$GC(H)_x = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} H(x, y)x, \tag{5}$$

where $w$ and $h$ denote the width and the height of the kernel $H$. Using the above definition of DP disparity, Fig. 5 shows varying disparity levels in relation to glass depth. As mentioned in Section III-A2, in our experiments, we set glass depths at the range of [15, 25] cm for the vertical setup. According to Fig. 5, the disparity range for the vertical setup results from 6 px (5cm glass depth) to 1.2 px (25 cm glass depth). As for the tilted setup, we set glass depths at the range of [5, 25] cm and the tilted angles at the range of [30, 45] (degrees). In the furthest case, the glass depth results in 47 cm, which can occur in the case of the maximum 25 cm glass depth and the minimum 30-degree tilted angle. In this furthest case, the resultant disparity is 0.6 px.

To blur the AiF raindrops with the above PSF kernels, the AiF raindrops $R_i^{AiFR}$ are extracted from $I_i^{AiFR}$ as

$$R_i^{AiFR} = M^{AiFR} \otimes I_i^{AiFR}, \tag{6}$$

where $\otimes$ denotes pixel-wise multiplication. Then, as shown in Fig. 4(b)–(d), the AiF raindrops $R_i^{AiFR}$ are divided into $6 \times 8$
sub-patches and convolved with the PSF kernels in a patch-by-patch manner as

\[ \hat{R}_l^{blur|i} = H_l^i \ast I_{AiFR|i} \]

(7)

where \( \ast \) denotes the convolution operation. The sub-patches are then stitched to form the blurred raindrop image \( \hat{R}_l^{blur} \). Similarly, the binary raindrop mask \( M_{AiFR} \) is also blurred to generate the blurred raindrop mask \( \hat{M}_l^{blur} \), which is used as the weight for the alpha-blending in the next step.

4) Alpha Blending: The final part performs the alpha blending of the synthetic raindrops and the real-world background as

\[ I_l = \hat{M}_l^{blur} \otimes \hat{R}_l^{blur} + (1 - \hat{M}_l^{blur}) \otimes B_l \]

(8)

to generate the left-half image \( I_l \) with smooth and natural transitions from the raindrops to the background. The same process is conducted to generate the right-half image \( I_r \), where the right-half background image \( B_r \) and the PSF kernels for the right-half image \( H_r^i \) are used for the same AiFR raindrop mask \( M_{AiFR} \). The differences of the PSF kernels between \( H_l^i \) and \( H_r^i \) (see [26] for details) derive the DP disparities for the raindrops. For the combined images corresponding to the images captured by a regular sensor, we generate them by averaging the left-half and the right-half 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, we generated four different synthetic-raindrop data by randomly changing the glass depth \( d_0 \) and the glass angle \( \theta \) only for the tilted setup) within the assumed ranges explained in Section III-A2. The raindrop size and location are also randomly changed. Each data contains the generated images of \( I_l, I_r, I_c \) and the ground-truth background images of \( B_l, B_r, B_c \). A total of 613 background images were captured, resulting in 2,452 scenes for each of the vertical and the tilted setups. Of the 2,452 scenes for each setup, 1,960 scenes were used for training, and the remaining 492 scenes were used for testing, where the vertical and the tilted setups were treated independently in this paper. Representative samples from the dataset are shown in the first row of Fig. 6, where DP disparities can be observed in the raindrop regions.

B. Real-World Dataset

1) Real-Waterdrop Dataset: To assess generalization performance in real-world situations both quantitatively and qualitatively, it is necessary to collect real-world rainy images with corresponding ground-truth clean images. However, obtaining such paired images on a real rainy day is impractical. To address this issue, we used a sprayer to generate waterdrops onto a glass panel, mimicking real raindrops. To minimize the impact of background refraction changes, we only used one glass panel and fixed it at a certain position 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 waterdrops and captured another DP image pair as the input images. To prevent the production of mist-like minor raindrops that lack realism, in contrast to our previous study [32] where a mist sprayer was used, we employed a watering can and allowed waterdrops to fall from a specific height to simulate more realistic raindrops. Also, 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. In the data acquisition process, we varied both the glass depth and the glass angle randomly for every few capture, ensuring that the resultant dataset possesses adequate diversity for comprehensive analysis. As shown in the second row of Fig. 6, we carefully collected high-quality and well-aligned DP images with and without real waterdrops for both vertical and tilted setups, featuring varied and complex waterdrop patterns. We collected 125 scenes for the vertical setup and 130 scenes for the tilted setup, which are used only for evaluation.

2) Real-Car Dataset: Despite utilizing a simulated real-waterdrop dataset, the resulting captured images are not entirely representative of true rainy images with inclement weather conditions. To demonstrate the proposed method’s robustness in actual real-world driving scenarios, an additional dataset of 94 rainy images was collected through the use of a real car on a rainy day. Although this dataset does not include any accompanying ground-truth data, it can be utilized for qualitative comparison. To collect the aforementioned dataset, a Google Pixel 4 smartphone was set focused on the far background and
moved around behind the car windshield to capture different patterns of the raindrops on the windshield surface. The images were captured in the car type of Toyota JPN TAXI, which features a glass with an approximate tilt angle of 40 degrees. This dataset provides a more realistic and challenging platform for evaluating the proposed methods’ performance in actual driving conditions. Some representative samples are shown in Fig. 7.

IV. PROPOSED DP RAINDROP REMOVAL NETWORK

Fig. 8 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 \( B_c \) as close as the ground truth \( B_c \), where the final output \( B_c \) is a combined image as captured by a regular sensor. Our DPRRN 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 result. Each network is constructed using UNet [41] with residual blocks [42], for which we refer to the supplementary material, available online, in detail. All the networks are trained in an end-to-end manner, as detailed below.

A. DP Raindrop Detection

In DP raindrop detection, the input images \( I_l \) and \( I_r \) are channel-wise concatenated and sent to two individual networks \( G^l_{mask} \) and \( G^r_{mask} \) to predict two raindrop masks \( M_l \) and \( M_r \) depicting the raindrop locations of \( I_l \) and \( I_r \), respectively. For the left-half mask \( M_l \), the concatenated \( I_l \) and \( I_r \) are sent to \( G^l_{mask} \) to predict a pixel-wise soft raindrop mask \( \hat{M}_l \) through Sigmoid activation function at the last layer. In this process, \( G^l_{mask} \) performs raindrop localization guided by the DP disparities on the raindrops. Similarly, for the right-half mask \( M_r \), another network \( G^r_{mask} \) is utilized to predict a pixel-wise soft raindrop mask \( \hat{M}_r \). Here, \( G^l_{mask} \) and \( G^r_{mask} \) are two individual networks that do not share the parameters because the disparity of \( I_l \) to \( I_r \) and the disparity of \( I_l \) to \( I_r \) show inverted 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) [43] losses as

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

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 raindrops in \( I_l \) and \( I_r \), respectively.

B. 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 removal network \( G^l_{derain} \), 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 \( G^r_{derain} \) to predict the right-half background image \( \hat{B}_r \). Here, \( G^l_{derain} \) and \( G^r_{derain} \) 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 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 \( \hat{B}_c = \max (\hat{M}_l, \hat{M}_r) \) 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 \( \hat{B}_c^{init} = \hat{B}_l^{init} + \hat{B}_r^{init} \). Then, the channel-wise concatenation of \( \hat{M}_l, \hat{B}_l, \hat{B}_r, \hat{B}_c^{init} \) is sent to \( G^{fuse} \) to derive the residuals \( R_c^{fuse \_fine} \) in terms of \( \hat{B}_c^{init} \), from which the final output background image is derived as \( B_c = R_c^{fuse \_fine} + \hat{B}_c^{init} \).

The DP fused raindrop removal is optimized using negative structural similarity index measure (SSIM) losses [7, 44] to maximize SSIMs between the raindrop removal results and the ground-truth clean background images as

\[
L_{derain} = -SSIM \left( \hat{B}_l, B_l \right) - SSIM \left( \hat{B}_r, B_r \right) - SSIM \left( \hat{B}_c, B_c \right)
\]

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

\[ L = L_{\text{mask}} + L_{\text{derain}}. \]  

(11)

V. EXPERIMENTAL RESULTS

A. Implementation Details

Our DPRRN is implemented using Pytorch [45]. During the training, we use randomly cropped 480 × 120 patches with the batch size set to 12. To ensure that the shift directions of raindrops in DP image pairs do not change, we do not apply random flipping for data augmentation. RAdam [46] is used as the optimizer to train the network with an initial learning rate of $1 \times 10^{-3}$. We adopt two-stage training, where we first train DP raindrop detection with only the mask loss $L_{\text{mask}}$ for 100 epochs without changing the learning rate and then use the total loss $L = L_{\text{mask}} + L_{\text{derain}}$ 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. Using a single NVIDIA RTX 3090 GPU, it takes approximately 20 hours for the training and 0.0527 seconds for a single inference with 528 × 400 image size.

B. Ablation Study

The proposed DPRRN leverages the advantages of DP disparities through the introduction of DP raindrop detection and DP fused raindrop removal. To assess the effectiveness of each proposed component, an ablation study was conducted on the synthetic-raindrop dataset and the real-world waterdrop dataset.

Table I shows the results, where four methods are compared: BASELINE, which is the network built on regular sensor data, where raindrop detection and removal are conducted by taking the combined image $I_\text{c}$ as the input; DPRRN$^{-\text{RD}}$, which is the network built by excluding the DP raindrop detection part of DPRRN (i.e., not using the raindrop masks for DP fused raindrop removal); DPRRN$^{-\text{FRR}}$, which is the network built by removing the fusion part of DPRRN (i.e., $\hat{B}_\text{init}$ corresponds to the final output); DPRRN, which is the full proposed network. Each network was independently trained for the comparison.

Table I demonstrates that all DP-based methods exhibit superior performance compared to the regular-sensor-based BASELINE, confirming the effectiveness of using the DP sensor. DPRRN outperforms DPRRN$^{-\text{RD}}$ by a large margin on both the synthetic and the real-world datasets, validating that the DP raindrop detection utilizing DP disparities significantly improves the performance. DPRRN also demonstrates improvement over DPRRN$^{-\text{FRR}}$ to a certain extent, highlighting the positive impact of the DP fused raindrop removal in refining the result by incorporating additional background information brought by shifted raindrops.

C. Comparison With State-of-the-Art Methods

1) Compared Methods: We compared AttGAN [11] and CCN [36], 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 [47] 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 extracted the raindrop removal part of CCN for training and comparison. We also compared general image restoration methods, MPRNet [17], Restormer [18], and DGUNet [19]. Although these methods are not specifically designed for raindrop removal, their papers contain comparative experiments on rain streak removal and other image restoration tasks (e.g., image deblurring and image denoising) and demonstrate their effectiveness. For all the above single-image methods, the input is a regular single image, which is calculated as \( I_r = \frac{L+I_l}{2} \), and the output is a restored background image in a regular sensor domain, which is supervised by the ground-truth background image calculated as \( B_r = \frac{B_l + B_{r0}}{2} \). We used default parameters to train each network.

Two recent stereo-camera-based methods, StereoWaterdrop [38] and EPRRNet [37], were also included in the comparison, as they have a similar motivation to our method. Although our dataset is not a strict stereo image dataset, these methods can be retrained on our dataset by treating a DP image pair \( I_l \) and \( I_r \) as a stereo image pair. Because StereoWaterdrop uses the disparity map for computing losses, we generated the disparity map using GANet [48] as the original StereoWaterdrop. Because EPRRNet exploits semantic labels during the training, we followed the original implementation using a pre-trained semantic segmentation network to obtain the semantic labels on our dataset. Both methods generate two deraining results for \( I_l \) and \( I_r \) individually, which were supervised by \( B_l \) and \( B_r \), respectively. For the comparison, the two derained images were averaged to derive the final result in a regular sensor domain.

As a DP-based method, a recent DP-based defocus deblurring method of RDPD+ [25] was compared to validate the effectiveness of our network design utilizing a DP image. The input to both RDPD+ and our DPRRN is a DP image pair \( I_l \) and \( I_r \), and the final output was supervised by a clean background image \( B_r \) in a regular sensor domain. Since RDPD+ does not have any specific design for defocusing deblurring, it can be directly trained on our dataset.

2) Results on Vertical Setup. Quantitative Results: The training was performed solely on the synthetic-raindrop dataset and the testing was conducted on both the synthetic-raindrop and the real-world waterdrop datasets. Table II summarizes the peak signal-to-noise ratio (PSNR) and the SSIM results under the vertical setup. The results indicate that our DPRRN consistently surpasses the other state-of-the-art methods on both datasets. Compared to the best regular-sensor-based methods (i.e., Restomer for the synthetic and MPRNet for the real world), DPRRN achieves PSNR and SSIM improvements of 0.43 dB and 0.0055, respectively, for the synthetic dataset. For the real-world dataset, more substantial margins of 1.68 dB and 0.0128 are exhibited, which emphasizes the superior robustness of our DP-based method to real-world situations.

Furthermore, compared to the stereo-based EPRRNet, DPRRN achieves the improvements of 1.60 dB in PSNR and 0.0110 in SSIM on the synthetic dataset, as well as 1.82 dB and 0.0158 improvements on the real-world dataset. Also, compared to another DP-based RDPD+, our DPRRN obtains the margins of 2.25 dB and 0.0138 on the synthetic dataset, and 0.66 dB and 0.0102 on the real-world dataset. All these results strengthen the advantage of our network design, which effectively leverages the information provided by the DP disparities for raindrop detection and removal.

Qualitative Results: The qualitative results on the synthetic dataset in Fig. 9 demonstrate that our DPRRN effectively removes the raindrops and restores the structural details of the Ferris wheel and the roller coaster. Conversely, the other methods exhibit insufficient raindrop removal and incomplete restoration results. The results on the real-world dataset in Fig. 10 further highlight the robustness of DPRRN, as it is the only method that successfully removes the raindrops for the whole image, including the stripe-shaped raindrops in the red box and the bright raindrops in the green box, which are raindrop types not included in the synthetic training data. The regular-sensor-based methods fail to detect and remove the stripe-shaped raindrops and show over-smoothed results in the green box. EPRRNet and RDPD+ only succeed in removing the stripe-shaped raindrops but still fail to remove the bright raindrops in the green box. These results validate the robustness of our DPRRN in real-world situations.

3) Results on Tilted Setup. Quantitative Results: Table III summarizes the PSNR and the SSIM results under the tilted setup. Similar to the case of the vertical setup, the results indicate that our DPRRN outperforms all the other methods on both datasets. Specifically, compared to each second-best method highlighted in blue color, DPRRN achieves the PSNR and the SSIM improvements of 0.51 dB and 0.0029 on the synthetic

---

**TABLE II**

| Method Types | Methods | Synthetic (Vertical) | Real-World (Vertical) |
|--------------|---------|----------------------|----------------------|
|              | PSNR    | SSIM     | PSNR    | SSIM     |
| Regular      | AtGAN (CVPR2018) [11] | 30.22 | 0.9038 | 29.55 | 0.9014 |
|              | CCN (CVPR2021) [36] | 35.07 | 0.9515 | 32.13 | 0.9273 |
|              | MPRNet (CVPR2021) [17] | 36.76 | 0.9615 | 33.16 | 0.9393 |
|              | Restormer (CVPR2022) [18] | 37.53 | 0.9638 | 32.11 | 0.9304 |
|              | DGUNet (CVPR2022) [19] | 37.19 | 0.9633 | 31.99 | 0.9304 |
| Stereo       | StereoWaterdrop (IROS2021) [38] | 35.55 | 0.9410 | 32.52 | 0.9260 |
|              | EPRRNet (IJCVC2022) [37] | 36.36 | 0.9583 | 33.02 | 0.9363 |
| Dual Pixel   | RDPD+ (ICCV2021) [25] | 35.71 | 0.9555 | 34.18 | 0.9419 |
|              | DPRRN (Ours) | 37.96 | 0.9693 | 34.84 | 0.9521 |
Fig. 9. Qualitative comparison on the synthetic-raindrop dataset (vertical).

Fig. 10. Qualitative comparison on the real-waterdrop dataset (vertical).

| Method Types | Methods | Synthetic (Tilted) | Real-World (Tilted) |
|--------------|---------|--------------------|--------------------|
| Regular      | AttGAN (CVPR2018) [11] | 31.85 0.9872 | 30.99 0.9120 |
|              | CCN (CVPR2021) [36] | 35.70 0.9668 | 31.03 0.9170 |
|              | MPRNet (CVPR2021) [17] | 38.27 0.9749 | **34.69 0.9372** |
|              | Restormer (CVPR2022) [18] | **38.95 0.9761** | 33.89 0.9348 |
|              | DGUNet (CVPR2022) [19] | 38.89 0.9764 | 34.42 0.9371 |
| Stereo       | StereoWaterdrop (ICROS2021) [38] | 34.97 0.9545 | 32.52 0.9187 |
|              | EPRRRNet (IJCV2022) [37] | 37.81 0.9734 | 32.36 0.9286 |
| Dual Pixel   | RDPD+ (ICCV2021) [23] | 36.89 0.9688 | 33.98 0.9308 |
|              | DPPRN (Ours) | **39.46 0.9793** | **35.26 0.9393** |

Red: rank 1st; Blue: rank 2nd.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
dataset, and 0.57 dB and 0.0021 on the real-world dataset. These results validate the effectiveness of our DPRRN even for the tilted setup in both aspects of the DP sensor utilization and the network design.

**Qualitative Results:** The qualitative results on the synthetic dataset are depicted in Fig. 11. The results reveal that, for the raindrops located within the red box, our DPRRN is capable of restoring the intricate details of the billboard, unlike the other methods which tend to produce noticeable artifacts and over-smoothed results. Similarly, for the raindrops within the green box, DPRRN can accurately recover the window structures, while the other methods present distorted results, such as stretching the window incorrectly. For the results on the real-world dataset in Fig. 12, the performance of our DPRRN is further highlighted, as DPRRN is the only method that is able to sufficiently eliminate the raindrops from the entire image, while the other methods tend to ignore many raindrops, leading to under-deraining.

**4) Results on Real-Car Dataset:** To demonstrate the practical utility of our DPRRN in real situations, we provide qualitative comparisons with the other methods in the real-car dataset, which was taken in a real car on a rainy day. We directly employ the networks trained on the synthetic-raindrop dataset (tilted setup) for testing. Fig. 13 depicts two examples. For the top sample, the results show that DPRRN outperforms the other methods, successfully removing all the raindrops across the entire image, including dense and bright ones in the red box and small ones in the green box. The other regular-sensor-based methods and the stereo-based methods fail to effectively remove the raindrops. The result of the DP-based RDPD+ method is...
relatively closer to ours, but the overall quality of the image produced by DPRRN is clearer and sharper.

The bottom sample was taken from a closer distance from the glass, resulting in raindrops with huge size and blur. Despite this challenge, DPRRN effectively removes all the raindrops and restores the building details in the red box. It also successfully eliminates the raindrops in the green box, where the raindrops are brighter and have a more significant impact on the background. In contrast, all the other methods fail to detect and remove the small and minor raindrops, leading to noticeable under-deraining.

More results on all the datasets can be seen in the supplementary materials, available online.

D. Effects of PSF Calibration

We initially used the front-focus PSF kernels from [26] and achieved high real-world generalization performance. As our target is the back-focus situation, we swapped the front-focus kernels of [26] to ensure the correct disparity direction for our target, as described in Section III-A3. However, as mentioned in the supplementary material of [26], the front-focus and the back-focus kernels may show different shapes, resulting in different blur and disparity patterns. Therefore, we followed [26] to calibrate the PSF kernels for the back-focus situation using the same 6 × 8 disk pattern as [26] and the optimization method proposed in [49]. For the back-focus setup, we placed a Google Pixel 4 smartphone 15 cm away from the display and set the focus distance to 10 m.

Using these calibrated back-focus PSF kernels, we re-rendered the synthetic-raindrop dataset (tilted) with the same method mentioned in Section III-A while maintaining the same raindrop locations, glass depths, and tilted angles. This ensures a fair comparison by changing only the PSF kernels to confirm the effect of the PSF calibration. We then re-trained the proposed DPRRN using this re-rendered dataset and evaluated
TABLE IV
QUANTITATIVE COMPARISON WITH DIFFERENT PSF KERNELS ON
REAL-WORLD WATERDROP DATASET (TILTED)

| Methods                        | Real-World (Tilted) |
|--------------------------------|---------------------|
|                                | PSNR | SSIM  |
| DPRRN (Swapped [26]’s Front-focus Kernels) | 35.26 | 0.9393 |
| DPRRN* (Calibrated Back-focus Kernels)       | 35.30 | 0.9403 |

Fig. 14. Qualitative comparison with different PSF kernels on real-car dataset. DPRRN and DPRRN* denote the networks trained with the swapped front-focus PSF kernels of [26] and the calibrated back-focus PSF kernels, respectively.

its performance on the real-world waterdrop dataset (tilted) and the real-car dataset. As shown in Table IV, the quantitative performance on the real-world waterdrop dataset (tilted) slightly improved. Qualitative comparisons in Fig. 14 demonstrate improved real-world generalization performance with the re-trained DPRRN (marked as DPRRN*), likely due to improved consistency between the synthetic and the real-world blur and disparity patterns. In conclusion, while reusing the PSF kernels from [26] achieved satisfactory results, the calibrated back-focus PSF kernels further enhanced real-world raindrop removal performance.

E. Results of Boosting Downstream Task

Our ultimate goal in conducting raindrop removal is to ensure the stable execution of downstream tasks in real-world applications. To demonstrate the ability of our DPRRN to achieve this goal, we exemplify the performance of an object detection algorithm using Google API [50]. As depicted in Fig. 15, the object detection results before and after the raindrop removal reveal that the object detection algorithm fails to detect the cars when applied to the raindrop-degraded image, whereas the algorithm functions properly when applied to the derained image by DPRRN. This highlights the practical potential of DPRRN.

VI. CONCLUSION AND DISCUSSION

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 recovery. To train and test our DPRRN, we have constructed both synthetic and real-world DP raindrop removal datasets, which address vertical and tilted setups that commonly appear in real-world situations. Using the constructed datasets, we have experimentally demonstrated that our DPRRN outperforms existing state-of-the-art methods, showing better generalization ability to real-world raindrops.

As for the limitations of our method, we have primarily focused on raindrop removal, while rain streak removal is also crucial for practical applications, as rain streaks and raindrops often coexist. Moreover, we have treated the tilted and the vertical setups independently. Our method may encounter difficulties when a single network is tasked with handling both setups simultaneously. Also, our current method is tailored to Google Pixel 4 smartphone model and its PSF kernels. Applying our method to other DP sensors would necessitate calibrating the PSF kernels for each device, which could be labor-intensive.

As potential directions for future research, we plan to address both rain streaks and raindrops within a unified framework. By simultaneously tackling these two weather phenomena, we aim...
to increase the real-world applicability of our method, ultimately improving performance in diverse weather conditions and offering a more robust solution for various vision-based applications. Furthermore, exploring a unified network architecture capable of handling different glass setups is worth investigating. Meanwhile, devising an efficient synthetic PSF kernel generation or automatic PSF calibration could also be beneficial.

REFERENCES

[1] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2117–2125.

[2] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, “Efficient deep CNN-based fire detection and localization in video surveillance applications,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 49, no. 7, pp. 1419–1434, Jul. 2019.

[3] C. Chen, A. Seff, A. Korinhauser, and J. Xiao, “DeepDriving: Learning affordance for direct perception in autonomous driving,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 2722–2730.

[4] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3855–3863.

[5] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Deep joint rain detection and removal from a single image,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1357–1366.

[6] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, “Recurrent squeeze-and-excitation context aggregation net for single image deraining,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 254–269.

[7] D. Ren, W. Zou, Q. Hu, P. Zhu, and D. Meng, “Progressive image deraining networks: A better and simpler baseline,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3937–3946.

[8] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” in Proc. IEEE Int. Conf. Comput. Vis., 2020, pp. 3103–3112.

[9] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, “Spatial attentive single-image deraining with a high quality real rain dataset,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 12 270–12 279.

[10] Y. Li, Y. Monno, and M. Okutomi, “Single image deraining with rain embedding consistency and layered LSTM,” in Proc. IEEE Winter Conf. Comput. Vis. Appl. Comput. Vis., 2022, pp. 4060–4069.

[11] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, “Attentive generative adversarial network for raindrop removal from a single image,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 2482–2491.

[12] Y. Quan, S. Deng, Y. Chen, and H. Ji, “Deep learning for seeing through window with raindrops,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 2463–2471.

[13] M. Shao, L. Li, H. Wang, and D. Meng, “Selective generative adversarial network for raindrop removal from a single image,” Neurocomputing, vol. 426, pp. 265–273, 2021.

[14] S. Zini and M. Buzzelli, “Laplacian encoder-decoder network for raindrop removal,” Pattern Recognit. Lett., vol. 158, pp. 24–33, 2022.

[15] M.-W. Shao, L. Li, D.-Y. Meng, and W.-M. Zuo, “Uncertainty guided defocus deblurring using multi-scale feature extraction and phase correlation matching,” Sensors, vol. 15, no. 3, pp. 5747–5762, 2015.

[16] A. Abuolaim and M. S. Brown, “Defocus deblurring using dual-pixel data,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 111–126.

[17] A. Abuolaim, M. Delbracio, D. Kelly, M. S. Brown, and P. Milanfar, “Learning to reduce defocus blur by realistically modeling dual-pixel data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 2289–2298.

[18] S. Xin et al., “Defocus map estimation and deburring from a single dual-pixel image,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 2228–2238.

[19] R. Garg, N. Wadhwa, S. Ansari, and J. T. Barron, “Learning single camera depth estimation using dual-pixels,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 7628–7637.

[20] A. Punnappurath, A. Abuolaim, M. Afifi, and M. S. Brown, “Modeling defocus-disparity in dual-pixel sensors,” in Proc. IEEE Int. Conf. Comput. Photography, 2020, pp. 1–12.

[21] Y. Zhang, N. Wadhwa, S. Orts-Escalon, C. Hâne, S. Fanello, and R. Garg, “Du2Net: Learning depth estimation from dual-cameras and dual-pixels,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 582–598.

[22] L. Pan, S. Chowdhury, R. Hartley, M. Liu, H. Zhang, and H. Li, “Dual pixel exploration: Simultaneous depth estimation and image restoration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 4340–4349.

[23] A. Punnappurath and M. S. Brown, “Reflection removal using a dual-pixel sensor,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1556–1565.

[24] Y. Li, Y. Monno, and M. Okutomi, “Dual-pixel raindrop removal,” in Proc. Brit. Mach. Vis. Conf. 2022, Art. no. 0439.

[25] J. C. Halimeh and M. Roser, “Raindrop detection on car windshield using geometric-photometric environment construction and intensity-based correlation,” in Proc. IEEE Intell. Veh. Symp., 2009, pp. 610–615.

[26] M. Roser, J. Kurz, and A. Geiger, “Realistic modeling of water droplets for monocular adherent raindrop recognition using Bezier curves,” in Proc. Asian Conf. Comput. Vis., 2010, pp. 235–244.

[27] Q. Wu, W. Zhang, and B. V. Kumar, “Raindrop detection and removal using salient visual features,” in Proc. IEEE Int. Conf. Image Process., 2012, pp. 941–944.

[28] R. Quan, X. Yu, Y. Liang, and Y. Yang, “Removing raindrops and rain streaks in one go,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 9147–9156.

[29] K. Zhang et al., “Beyond monocular deraining: Parallel stereo deraining network via semantic prior,” Int. J. Comput. Vis., vol. 130, pp. 1754–1769, 2022.

[30] Z. Shi, N. Fan, D.-Y. Yeung, and Q. Chen, “Stereo waterdrop removal with row-wise dilated attention,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2021, pp. 3829–3836.

[31] T. Yang et al., “Raindrop removal with light field image using image inpainting,” IEEE Access, vol. 8, pp. 51 416–51 426, 2020.

[32] Y. Ding, M. Li, T. Yan, F. Zhang, Y. Liu, and R. W. Lau, “Rain streak removal from light field images,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 2, pp. 467–482, Feb. 2022.

[33] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv., 2015, pp. 234–241.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[35] I. J. Good, “Rational decisions,” in Breakthroughs in Statistics. Berlin, Germany: Springer, 1992, pp. 365–377.

[36] D. Ren, W. Shang, P. Zhu, Q. Hu, D. Meng, and W. Zuo, “Single image deraining using bilateral recurrent network,” IEEE Trans. Image Process., vol. 29, pp. 6852–6863, May 19, 2020.

[37] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 8026–8037.
L. Liu et al., “On the variance of the adaptive learning rate and beyond,” 2019, arXiv: 1908.03265.

F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, “GA-Net: Guided aggregation net for end-to-end stereo matching,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 185–194.

F. Mannan and M. S. Langer, “Blur calibration for depth from defocus,” in Proc. 13th Conf. Comput. Robot Vis., 2016, pp. 281–288.

Yizhou Li (Student Member, IEEE) received the BEng degree from the School of Computer Science, Sichuan University, in 2019, and the MEng degree from the Department of Systems and Control Engineering, Tokyo Institute of Technology, in 2021. He is currently working toward the PhD degree with the Department of Systems and Control Engineering, Tokyo Institute of Technology. His research interests include low-level vision and 3D reconstruction.

Yusuke Monno (Member, IEEE) received the BE, ME, and PhD degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 2010, 2011, and 2014, respectively. From 2013 to 2014, he joined the Image and Visual Representation Group, École Polytechnique Fédérale de Lausanne as a research internship student. He is currently an assistant professor with the Department of Systems and Control Engineering, School of Engineering, Tokyo Institute of Technology. He received various awards including the NF Foundation R&D Encouragement Award in 2015, the Inoue Research Award for Young Scientists in 2016, the IEEE Signal Processing Society Japan Best Paper Award in 2017, the Telecom System Technology Award in 2020, and the ICIP Best Student Paper Award in 2022. His research interests include both theoretical and practical aspects of image processing and computer vision.

Masatoshi Okutomi (Member, IEEE) received the BEng degree from the Department of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo, Japan, in 1981, the MEng degree from the Department of Control Engineering, Tokyo Institute of Technology, in 1983, and the DrEng degree from the Tokyo Institute of Technology, in 1993, for his research on stereo vision. He joined the Canon Research Center, Canon Inc., Tokyo, in 1983. From 1987 to 1990, he was a visiting research scientist with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. Since 1994, he has been with Tokyo Institute of Technology, where he is currently a professor with the Department of Systems and Control Engineering, School of Engineering.