Rain Streak Removal From Light Field Images

Yuyang Ding, Mingyue Li, Tao Yan*, Fan Zhang, Yuan Liu, and Rynson W.H. Lau

Raining is a common weather condition, and may seriously degrade the performances of outdoor computer vision systems, such as surveillance and autonomous navigation. Rain streaks may exhibit diverse appearances in the captured images, depending on their distances from the camera. For example, sparse rain streaks near the camera lens may appear as continuous and translucent strips, while distant densely accumulated rain streaks are more like fog and mist. Existing rain removal methods are mainly based on a single input image. However, on a single image, it is difficult to estimate a reliable depth map for rain removal. A light field image (LFI) records abundant structural and texture information of the target scene by capturing multi-perspective sub-aperture views with a single exposure. With a LFI, it is easier to estimate the depth maps, and rain streak locations across sub-aperture views are highly correlated. We observe that rain streaks usually have different slops and/or chromaic values, compared with the background scene, along the epipolar plane images (EPIs) of an LFI. Thus, we propose to make use of 3D EPIs to detect rain streaks and restore the background. To this end, we propose a novel GAN architecture to remove rain streaks from an LFI. Our method takes as input a 3D EPI, i.e., a stacked of sub-aperture views along the same row of a rainy LFI. It first estimates the disparity maps for the 3D EPI by utilizing an autoencoder based depth estimation sub-network. The disparity maps concatenated with the input sub-aperture views are then fed into a non-local residual block, and two branched autoencoder sub-networks are used to extract rain-streaks and recover rain-free sub-aperture views. Extensive experiments conducted on both synthetic real-world-like LFIs and real-world LFIs demonstrate the effectiveness of our method.

***Index Terms*—Light field image, rain streak removal, deraining**

I. Introduction

RAIN removal has been a popular research topic in the computer vision community. It has attracted a significant attention from researchers in the past decade. Raindrops attached to the camera len usually cause refraction and distort the background scene. Nearby rain streaks may appear semi-transparent, and often introduce a certain degree of opaque/blur to the objects behind. Distant dense rain streaks behave more like mist/veiling, and obscure the background scene [1] [2] [3] [4]. However, removing these various types of rain streaks from a single view or multi-view image/video is not a trivial task.

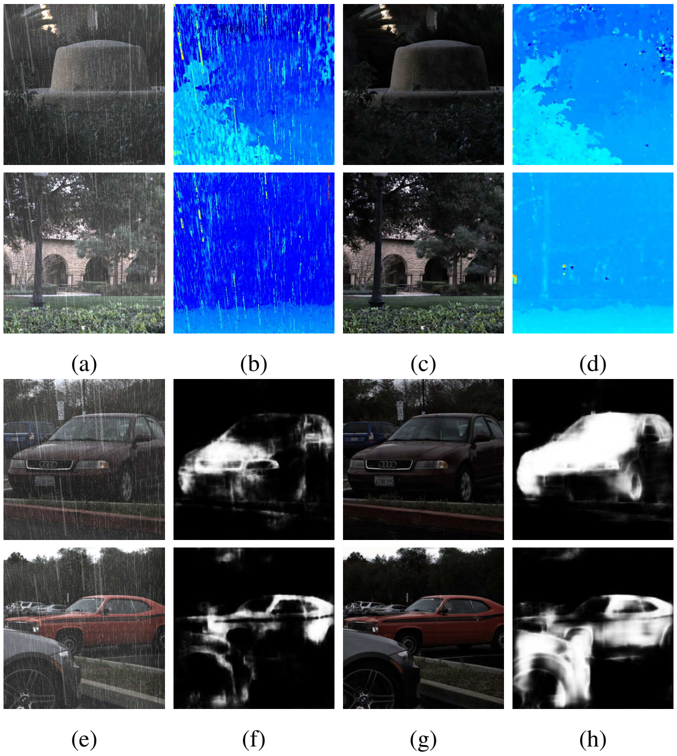


Fig. 1: Results of depth/disparity estimation and saliency detection on rainy LFIs and our derained images. (a) is the central subaperture view of the original rainy LFI, while (c) is our de-rained image. (b) and (d) show the disparity maps produced by method [5] on the rainy and de-rained images in (a) and (c), respectively. Similarly, (e) and (g) are another pair of central subaperture view of the rainy LFI and our de-rained image, while (f) and (h) are the saliency detection results by method [6] on (e) and (g).

Light field cameras [7] [8] are becoming an important type of imaging device in both VR/AR and autonomous navigation systems. In computer vision, light field denotes a simplified four-dimensional parametric representation of the original seven-dimension optical radiation field of a 3D space. As raining can degrade the quality of light-field images (LFIs) and affect vision tasks on LFIs, such as depth estimation [5] and saliency detection [6], it is important to restore rain-free LFIs from rainy LFIs, as shown in Fig. 1. A similar research problem is reflection removal on LFIs [9] [10] [11], which exploit multi-perspective subaperture views of an LFI for accurate reflection removal. In this work, we investigate the problem of rain streak removal from a single LFI.

As shown in Fig. 2, the angular and spatial coordinates of an LFI can be represented as (s, t) and (x, y), respectively. The

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61902151), and in part by the Natural Science Foundation of Jiangsu Province (Grant No. BK20170197).

Yuyang Ding, Mingyue Li, Tao Yan, Fan Zhang and Yuan Liu are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Jiangsu, China.

Rynson W.H. Lau is with the Department of Computer Science, City University of Hong Kong, Hong Kong SAR.

Tao Yan is the corresponding author. Rynson W.H. Lau leads this project.

input of our method is a 3D EPI consisting of a set of sub-aperture views in the same row/column, which can be extracted from a 4D LFI by fixing the angular dimension (e.g., $t = t^*$) as follows:

$$V_{t^*} = L(x, y, s, t^*). \tag{1}$$

Given that rain streak appearances on an image are highly correlated to the content of the disparity map, an accurate disparity map can be useful for rain steaks removal. However, on a single image, it is difficult to estimate an accurate disparity map for rain removal, and rain streak locations across different frames of a video may be uncorrelated. In contrast, on an LFI, it is easier to estimate accurate disparity maps, and the rain streak locations across sub-aperture views are spatially correlated. As shown in Fig. 2, the slopes of rain streaks in a 3D epipolar plane image (EPI, i.e., a row of sub-aperture views of an LFI) are different from that of the background scene. Hence, we propose to extract rain streaks and recover a rain-free LFI from a rainy LFI by fully exploiting the spatial and angular information of an input LFI.

In this paper, we proposed a depth/disparity map guided GAN architecture to remove rain streaks from rainy LFIs. Our proposed network includes a generator part and a discriminator part, as shown in Fig. 3. The generator part first estimates the disparity maps using an auto-encoder based *Disparity Estimation Network (DE-Net)*. It then concatenates the estimated disparity maps with the input 3D EPI, and feed them to a U-shaped *Background Recovery Network (BR-Net)* to extract the rain-streak image and restore the rain-free image. In the discriminator part, the restored rain-free sub-aperture views are further fused with the estimated rain-streak image to generate a rainy LFI. The discriminator part contains two discriminators, *Discriminator 1* and *Discriminator 2*. *Discriminator 1* is introduced to identify the recovered rain-free LFI and the ground-truth rain-free LFI. It guides the generator to generate more accurate rain-free LFIs. *Discriminator 2*, on the other hand, is used to identify the reproduced rainy LFI and the input rainy LFI, in order to guide the generator to generate more accurate rain-free as well as rain-streak LFIs. Finally, a rainy light-field dataset, consisting of both synthetic real-world-like LFIs and real-world LFIs, is constructed for training and evaluation under semi-supervised learning.

Our main contributions of this work are summarized as:

- We propose the first powerful end-to-end deep learning model for rain-streak removal on a single rain LFI, by decompsing it into rain-free LFI and rain-streak LFI with mist/veiling.
- We have conducted extensive experiments to evaluate the proposed model.
- We construct the first rainy light-field dataset, consisting of both synthetic real-world-like LFIs and real-world LFIs for training and evaluation.

II. Related Work

Rain removal methods can be broadly divided into single-image based methods and video (or multi-image) based methods. In this section, we mainly review rain removal methods for singe images and frame sequences, and we also discuss LFI-based deraining and other-related works.

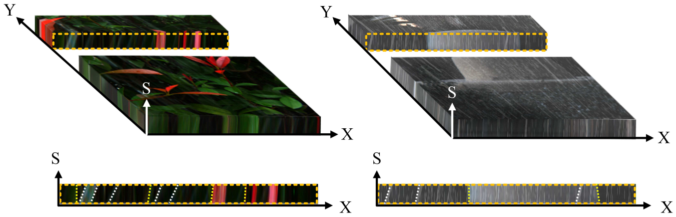


Fig. 2: 3D EPI of an LFI. The first row shows 3D EPIs, and the second shows 2D EPIs, i.e., a slice cut from the 3D EPI. Normally, rain streaks are much closer to the camera len than the background scene. Hence, the slopes of rain streaks on an EPI are always smaller than that of the background, as shown by the white dotted lines and yellow dotted lines, respectively.

A. Single-Image Rain Streak Removal Methods

Traditional methods [1] [2] [12] [13] [14] [15] usually explored physical characteristics of rain streaks by utilizing handcrafted image features with classical optimization or machine learning methods. Many models have been developed to capture the intrinsic differences between the rain signal and normal textures based on spatial redundancy, e.g., generalized low rank model [14], sparse coding [15], discriminative sparse coding [13], Gaussian mixture model [12], and layer decomposition [16]. Li et al. [17] modeled the input noisy image as a superposition of coarse structure and fine details, and proposed a new discriminative dictionary learning method to separate coarse-scale structures and fine details. However, classical optimization-based methods are difficult to obtain immaculate rain-free images, especially when images contain heavy rain streaks.

On the other hand, with the power of learning from data, the deep-learning-based rain removal methods [18] [19] [3] [20] [21] [22] [23] [24] [25] [26] [4] [27] have achieved significant progress in the last few years, and substantially outperform traditional methods. Fu et al. [18] proposed *DerainNet* to learn the relationship between rainy and clean image detail layers. The input rainy image is first separated into a low-passed base layer and a high-passed detail layer. It then estimates the de-rained layer from the detail layer and enhanced the de-rained layer and the base layer to produce a rain-free image. However, the method may fail in removing heavy rain. Yang et al. [22] proposed a rain image model that comprises of a rain streak accumulation layer (i.e., mist/veiling) and multiple rain-streaks layers. A recurrent rain detection and removal architecture is proposed to remove rain streaks and clean up rain accumulation progressively. A multi-task contextualized dilated convolution model was proposed to obtain binary rain streak map, rain streak appearance, and rain-free background in each recurrence. Qian et al. [19] proposed an attentive generative adversarial network for raindrop removal. It adopts an attentive generative network using adversarial training to inject visual attention into both the generative and adversarial networks. While the visual attention

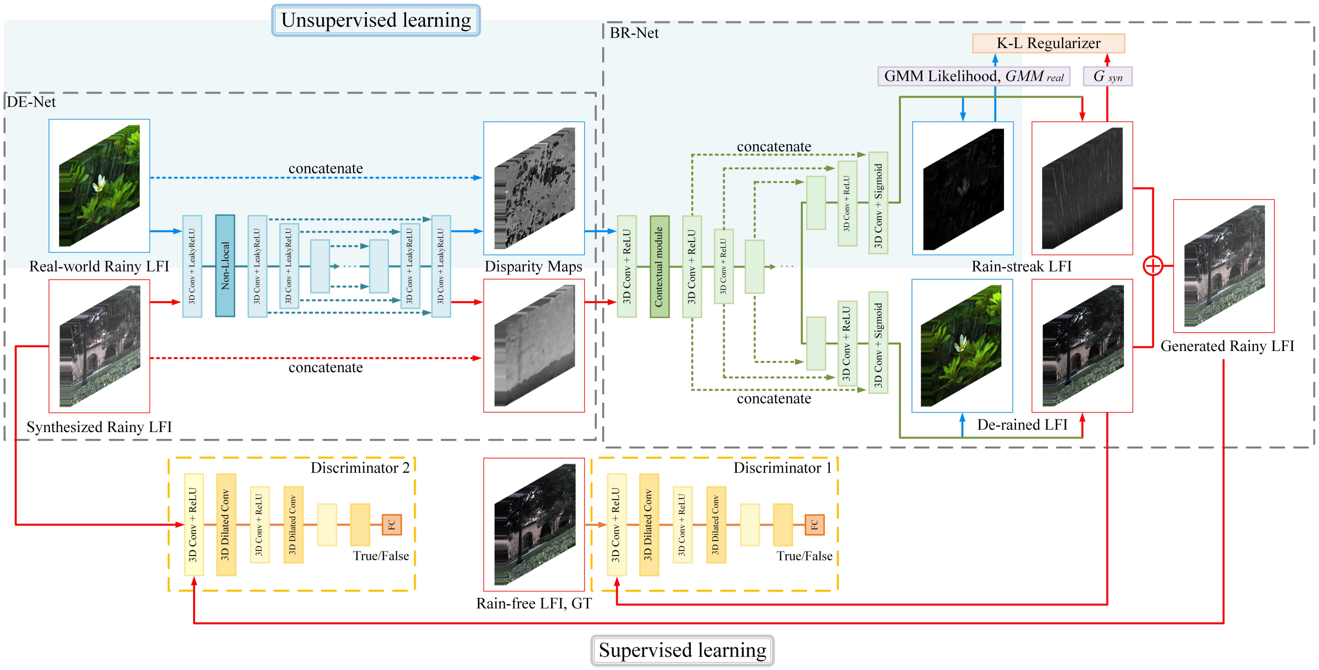


Fig. 3: Our proposed rain streak removal network. Blocks inside red boxes represent the synthetic real-world-like rainy LFI and its results, and blocks inside blue boxes represent the real-world LFI and its results. The red solid arrows represent the supervised learning procedure, while the blue solid arrows represent the unsupervised learning progress. Our proposed deraining network contains two sub-networks. First, the *Disparity Estimation Network (DE-Net)* estimates the disparity maps for the input LFI. Second, the estimated disparity maps are concatenated with the input LFI and fed into the *Background Recovery Network (BR-Net)* to extract the rain-streak LFI and recover the rain-free LFI. Two discriminators are used to identify the results of the synthetic rainy LFI. *Discriminator 1* is used to authenticate the generated rain-free LFI. *Discriminator 2* is used to authenticate the generated rainy LFI, which is reproduced by combining the extracted rain-streak LFI and the recovered rain-free LFI. For real-world rainy LFIs without ground-truth rain steak images, a Gaussian mixture model (GMM), GMM_{real}, is adopted to represent the real-world rain streak distribution, and the Kullback-Leibler divergence between the Gaussian G_{syn} learned from the synthetic rain streaks and GMM_{real} is minimized.

of the generator would pay more attention to the raindrop regions and their surrounding structures, the discriminator could assess local consistency of the restored rain-free regions. Li et al. [20] proposed a non-locally enhanced encoder-decoder network for rain streak removal. It is composed of a series of non-locally enhanced dense blocks [28] [29] to exploit hierarchical features from all convolutional layers as well as to capture the long-distance dependency and structural information. Zhang et al. [23] proposed a density-aware multi-stream dense convolutional architecture for joint rain steak density estimation and removal. A residual-aware rain-density classifier was proposed to determine the density-level of a given rainy image, and a multi-stream densely connected deraining network was proposed to remove rain streaks from the rainy images guided by the estimated rain-density information. Li et al. [24] proposed a multiple stages rain remove method based on the squeeze-and-excitation block and the dilated convolutional neural network.

Recently, Ren et al. [21] proposed a simpler baseline network for rain streak removal in multiple stages, where a shallow ResNet is deployed at each stage. Hu et al. [3] and Li at al. [4] incorporated depth maps to improve the quality of the output rain-free images. Hu et al. [3] analyzed the visual effects of rain subject to scene depth and established a synthetic model composed of rain streaks and fog. A robust network was proposed to learn depth-attentional features via a depth-guided attention mechanism, and regress a residual map to produce a rain-free image. However, since the depth map estimated from a single rainy image is unreliable, affecting the quality of the results. Li et al. [4] proposed a 2-stage CNN (i.e., a physics-based backbone followed by a depth-guided GAN refinement) to remove heavy rain steaks and rain veiling effect. In the physics-based stage, a streak-aware decomposition module helped decompose the entangled rain streaks and rain accumulation. In the refinement stage, a depth-guided conditional GAN helped recover the background details. However, this method cannot remove distant rain streaks, especially in heavy rain. Wei et al. [25] proposed a rain streak removal method to solve the domain adaption problem of transferring from synthetic rainy images to real-world rainy images by adopting semi-supervised learning, which utilized both supervised synthetic samples and unsupervised samples for training.

Most recently, Jiang et al. [27] proposed a multi-scale progressive fusion network (MSPFN) to exploit multi-scale rain information using a pyramid framework. Ren et al. [30]

proposed a simple yet effective single recurrent network (SRN) for image deraining, in which the composition pattern of two layers can be exploited and propagated in multiple stages via LSTM. A bilateral recurrent network (BRN) is also proposed to model the interplay between rain streak and background image layers. Zhang et al. [26] proposed a conditional GAN-based network for rain streak removal. Densely connected blocks [29] are used to construct the generator sub-network to enable steady gradient flow, and a multi-scale discriminator is proposed to leverage features from multi-scales to authenticate the de-rained image. Luo et al. [31] proposed a weakly supervised raindrop removal adversarial network, which needs only a collection of images with image-level annotations indicating the presence/absence of raindrops for training. A raindrop detector in trained to indicate raindrop regions. An attention-based generative network is then used for raindrop removal. Zhu et al. [32] proposed a gated non-local deep residual learning framework for image deraining, which can avoid over-deraining or under-deraining caused by global residual learning. However, the proposed method is still trained and evaluated on synthetic rainy images, and cannot handle real-world rainy images well.

However, all aforementioned deep-learning based rain removal methods take a single 2D image as input. They cannot be directly applied on LFIs to take advantage of the additional information provide by LFIs. In contrast, our method is specially designed for rain steak removal on LFIs and has two main advantages. First, our method can restore accurate disparity maps from input LFIs. Second, it can recover clear and sharp rain-free LFIs by exploring abundant color, texture and depth/disparity information from the input LFIs.

B. Video Rain Streak Removal Methods

Garg and Nayar [1] [2] proposed the seminal work of video rain modeling. Recent video deraining methods [33] [34] [35] [36] exploited spatial and temporal redundancies in frame sequences.

Li et al. [33] proposed a multiscale convolutional sparse coding (MS-CSC) model for video rain streak removal. It first utilizes multiple convolutional filters convolved on the sparse feature maps to detect repetitive shape characteristic of rain streaks sparsely scattered over different positions of the video, and then utilize multiscale filters to represent different scales of rain streaks. Liu et al. [34] proposed a recurrent neural network to integrate rain degradation classification, spatial texture appearance based rain removal and temporal coherence based background detail reconstruction. Yang et al. [35] proposed a two-stage recurrent network built with dual-level flow regularizations to perform the inverse recovery process of the rain synthesis model for video deraining. Yang et al. [36] proposed a two-stage self-learned de-raining network to removal rain streaks based on both temporal correlation and consistency.

Video deraining methods always assume that the rain steak locations across frames are uncorrelated. In contrast, rain streak locations in different subaperture views of an LFI are highly correlated. However, unlike video frame sequences, there are no temporal redundancy can be exploited in LFIs. Thus, LFI rain streak removal is different from video rain streak removal.

C. Light Field Image Deraining and Related Tasks

There are some works on LFIs that are related to deraining or other tasks. Wu et al. [37] presented a comprehensive overview and discussion of various aspects of research on LFIs, including depth estimation [38] [39] [40], super-resolution [41] [42], and editing and inpainting [43]

Two optimization-based methods [44] [45] were proposed for rain streak removal on LFIs. Tan et al. [44] first aligned all subaperture views to the central subaperture view, and a robust PCA is then adopted to decompose each image of the set of warped subaperture views into a low-rank image and a sparse image. Finally, a dark view image is introduced to estimate the non-rain disparity edges from a 3D EPI, which are converted back to the low-rank image to produce the final rain-free LFI. Yang et al. [45] first estimated disparity maps from the input LFI and detected raindrop regions as a binary mask using the disparity maps. The image refocusing on the far regions is then enhanced by a high-pass filter. Finally, existing deep learning-based image inpainting methods [46] [47] are directly adopted to complete the masked raindrop LFI.

The most related task to rain removal on LFIs is snow removal and reflection removal. Tao et al. [48] proposed a deep-learning-based network for snowflake removal from LFIs. It takes 3D EPI of an snowy LFI as input, from which a 3D residual network is introduced to detect snowflakes. Then, an encoderdecoder-based LF image restoration network is proposed to restore the clean background image. Ni et al. [9] proposed a single-LFI reflection removal method. Through focus manipulation and analyzing the EPI slop, it labeled edges in the background and reflection layers, respectively. The labeled edges are then used to separate the initial background and reflection layers. Finally, by enforcing the input EPIs to be the sum of the background and reflection EPIs, refined separation results can be obtained through an alternating optimization between layer update and disparity update. Li et al. [10] [11] noted that if an LFI was formed by the superimposition of two LFI layers of different disparities, the EPI strong gradient points of both images would be at different positions of the combined EPI, and the gradient values would be preserved. They proposed a general sandwich model to describe the depth range of the background and reflection images, and then a traditional optimization based method [10] and a deep-learning-based method [11] to solve the reflection removal problem of LFIs.

All these methods are either based on classical optimization techniques or just utilizing CNNs for de-rained LFI inpainting [45]. They are not able to accurately detect all rain streaks on a LFI, clearly remove rain streaks and recover sharp rain-free LFI from the input LFI. In this paper, we propose a deep-learning based method for rain streak removal on LFIs.

III. A Rainy Real-world-like LFI Dataset

Adding one-/multi-layers of rain streaks to a clean background image [12] [22] is a common way to generate real-

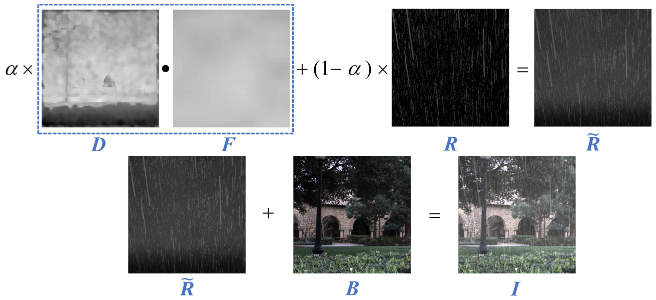


Fig. 4: Our real-world-like rainy LFI generation procedure. D is the disparity map of rain-free background LFI B, and are normalized to $[0, 1]$. F is the LFI of mist/veiling. R is the input rain-streak LFI. $\tilde{R}$ is the synthesized rain-streak LFI with mist/veiling. B is the rain-free background LFI. I is the synthesized output rainy LFI. α is a constant parameter.

world-like rainy images. A rainy image is usually considered as a linear/non-linear combination of a rain-free background layer and rain-streak layer(s) [14] [19] [3] [20] [22] [23] [26]:

$$I = B + R, \tag{2}$$

where I, B and R denote the rainy image, rain-free background image and rain-streak image, respectively. The goal of rain streak removal is to detect R and recover B from I, which is an ill-posed problem. However, a rainy image dataset created in this way may be different from the actual real-world rainy images, which means that a rain removal network trained on such a dataset may not produce the desired rain-free images given real-world rainy images.

In our method, various synthetic rain streaks simulated with the particle system of Blender [49] were superimposed to real-world LFIs to generate authentic rainy LFIs, or called real-world-like LFIs. The real-world LFI dataset [50] captured by a Lytro Illum camera is utilized to generate real-world-like rainy LFIs. However, for real-world rain-free/backgound LFIs, there are no ground-truth disparity maps that could be used directly. Hence, we adopt a state-of-the-art method [38] to obtain acceptable disparity maps as the ground truth. The disparity maps for each real-world LFI can then be used to generate rain and mist/veiling effects. 3D rain streaks with various shapes, directions and densities are generated using the particle system of Blender [49], and the rain steak LFIs are then rendered with Blender [49]. Finally, the above rendered LFIs of rain-streaks and real-world LFIs [50] are combined to generate the real-world-like rainy LFIs, as shown in Fig. 4. In this way, we obtained more than 80 rainy LFIs for training and evaluation.

In our rainy LFI generation model, $\tilde{R}$ is a linear combination of the semi-transparent rain streaks and mist/veiling simulated with the disparity maps. Specifically, the formula for generating real-world-like rainy LFIs can be described as:

$$\begin{aligned} \tilde{R} &= \alpha D F + (1-\alpha) R, && (3)\\ I &= B + \tilde{R}, && (4) \end{aligned}$$

where D is the disparity map of the input background rain-free LFI B, and is normalized to $[0, 1]$. F is the LFI of mist. R is the input rain-streak LFI. $\tilde{R}$ is the synthesized rain-streak LFI with mist/veiling effect. B is the rain-free background LFI. I is the synthesized output rainy LFI. α is a constant parameter (set as 0.3).

IV. LFI Rain Streak Removal Network

In this paper, we propose an adversarial network for rain-streak removal from a single LFI, as shown in Fig. 3. The generator part of our network contains two sub-networks, one for disparity/depth map estimation, and another for rain streak extraction and rain-free LFI recovery. Similarly, the discriminator part also contains two discriminators. One is used to discriminate the de-rained LFI and the ground truth. Another one is used to identify the generated rainy LFI fused from the de-rained/rain-streak images and the input LFI. Specifically, in the generator part of our network, a 3D EPI of the input rainy LFI is taken as the input of the first sub-network to estimate the disparity maps. The stacked disparity maps are then concatenated with the input 3D EPI as the input to the second U-shaped sub-network to detect the rain streaks and recover the desired rain-free image with two separately decoding branches. In particular, the 3D convolution layer is adopted as a basic layer to construct our neural network to process the input 3D EPI, which is able to extract meaningful information from multiple subaperture views at the same time.

A. Disparity Map Estimation Network (DE-Net)

Inspired by [3], the mathematical formulation of a rainy image is closely related to the depth map of the background image. However, it is difficult to estimate an accurate depth/disparity map for a single rainy image due to two reasons. First, the available contextual information conveyed by a single view is not enough to estimate a highly accurate depth/disparity map. Second, the interference of rain streaks makes it is difficult to estimate an accurate depth/disparity map for the background image obscured by rain streaks. Due to the abundant spatial and angular information recorded in an LFI, more accurate depth/disparity maps can be obtained than that from a single image/video.

Our 3D auto-encoder based DE-Net takes a 3D EPI as input. First, a 3D convolution layer with a filter size of $3 \times 3 \times 3$ and channel dimension 32 is used to extract high dimensional local features from the input 3D EPI. Inspired by the non-locally enhanced auto-encoder network [20], a 3D convolution non-local model [28] is then introduced to extract global responses/features from the high dimensional local features to enlarge the perceptual region of our network for disparity estimation.

The nonlocal block is adopted to overcome the limitation of the small receptive field of each 3D convolutional layer in our method. For example, if the nonlocal block is not used, the receptive region of the first 3D convolutional layer of the network is $3 \times 3 \times 3$, and that of the deepest layer is $22 \times 22 \times 9$. If we want to increase the perceptual region, we have to increase the number of 3D convolution layers. In contrast, the response/feature map obtained by the nonlocal block can obtain both global contextual information and self-attention information of the input image, both of which can help our method detect rain streaks and recover rain-free images accurately.

The 3D convolutional DE-Net is introduced to estimate the disparity maps for the input 3D EPI. The number of encoder channels increases from 64 to 1024. The decoder part is symmetrical with the encoder part, and the corresponding layers are concatenated by skip connections to reduce the lack of high-frequency information. The last layer outputs the restored disparity maps for the input subaperture views. Each convolution layer has a convolutional kernel size of $3 \times 3 \times 3$. We chose LeakyRelu as the activation function. A batch normalization layer is added to each convolutional layer to avoid the vanishing and exploding of gradients. The architecture of the DE-Net is shown in the Supplemental. It is worth noting that due to the limited number of real-world-like rainy LFIs available, both the synthetic real-world-like rainy LFIs discussed in Sec. III and some synthetic rainy LFIs with ground-truth disparity (rendered by Blender [49] in 3D virtual worlds) are utilized to train our DE-Net to produce accurate disparity maps.

We choose L_1 to measure the difference between the estimated disparity maps and the ground true as follows:

$$L_d = \|V_d' - F_d(V_{t^*})\|_1, \tag{5}$$

where V_{t^*} is the stacked subaperture views (i.e., 3D EPI with fixed angular coordinate t^*). V_d' denotes the ground-truth disparity maps. F_d represents the function of our disparity estimation sub-network.

B. Background Recovery Network (BR-Net)

With the disparity maps obtained by DE-Net, we concatenate them with the input 3D EPI as input to the subsequent *Background Recovery Network (BR-Net)* to extract rain streaks and recover a rain-free LFI, as shown in Fig. 3.

In the encoder part of BR-Net, we introduce a contextual module after a 3D convolutional layer, as shown in Fig. 5. The contextual module contains three parallel dilated convolutional paths, each of which contains three dilated convolutional layers with dilation rates of 1, 2, and 3. In this way, a larger receptive field and the characteristics of rain streaks with different sizes and densities could be obtained. The feature maps are then passed to two separate decoding branches. One is used to estimate the rain-streak images, and another one is used to recover the rain-free subaperture views. The structure of the generator part is shown in the Supplemental.

We adopt L_1 norm as the loss function of BR-Net. Since 3D convolution [51] [52] may loose high-frequency details and introduce blurriness, we add a perceptual loss and a style loss to the objective function to enforce the sharpness of the resulting rain-free images. However, we find that noise may be introduced to the generated image content. Thus, we introduce a total variation loss to address this problem. The loss function is defined as follows:

$$L_{g_1} = \|V_{rv} - V_{rv}'\|_1, \tag{6}$$

where V_{rv} are the recovered rain-free subaperture views with a fixed angular coordinate t^*, and V_{rv}' is the corresponding ground truth.

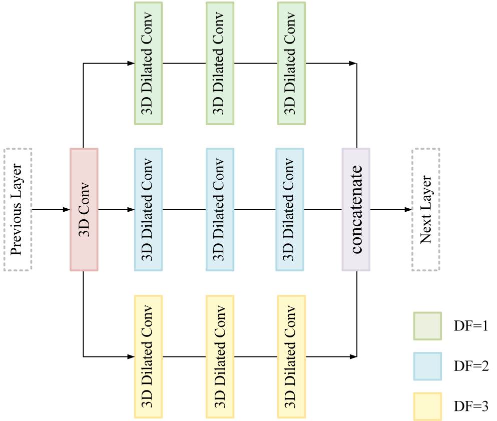


Fig. 5: The proposed contextual module in our method.

We use pre-trained VGG-16 [53] to project a selected feature map to a higher-level feature space for calculating the perceptual loss. The $pool_1$, $pool_2$ and $pool_3$ layers are chosen to calculate the loss of the feature map. The perceptual loss is defined as:

$$L_{perc_1} = \sum_{p=1}^{P} \sum_{s=1}^{N} ||\psi_{t^*,s}^{gt} - \psi_{t^*,s}^{out}||_1, \tag{7}$$

where $\psi_{t^*,s}^{out}$ is the feature map selected from a layer of VGG network. The inputs are the recovered rain-free subaperture views with a fixed angular coordinate t^*. N is the number of subaperture views in the input 3D EPI and set as 9. P is the number of selected *pool* layers.

The definition of the style loss is similar to the perceptual loss. We first compute the autocorrelation with the Gram matrix [54] G on each feature map. The style loss is then:

$$L_{style_1} = \sum_{p=1}^{P} \sum_{s=1}^{N} ||G(\psi_{t^*,s}^{gt}) - G(\psi_{t^*,s}^{out})||_1, \tag{8}$$

The total variation loss is used to ensure smoothness between neighboring pixels, and is defined as:

$$L_{tv_1} = \sum_{i,j} \sum_{s=1}^{N} \frac{||V_{t^*,s}^{i,j+1} - V_{t^*,s}^{i,j}||_1}{N} + \sum_{i,j} \sum_{s=1}^{N} \frac{||V_{t^*,s}^{i+1,j} - V_{t^*,s}^{i,j}||_1}{N}. \tag{9}$$

Similarly, the energy terms L_{g_2}, L_{perc_2}, L_{style_2}, L_{tv_2} are introduced to extract rain-streaks, which is similar to the energy terms defined to recover rain-free subaperture views.

As discussed above, our proposed network is trained on the synthetic rainy LFIs with both ground-truth rain streaks and derained LFIs, which tends to make the network biased toward learning the specific patterns of synthetic rain streaks. Thus, our network may not be able to generalize to real-world rain streaks of rainy LFIs, which may have more complex rain steaks with multiple directions and layers influenced by illuminations and wind. Inspired by the seminal work [25], we adopt the proposed semi-supervised learning scheme to our method to solve such a domain adaption problem [25] [55] by utilizing both synthetic rainy LFIs and real-world rainy LFIs for training, as shown in Fig 3.

Since the Gaussian mixture model (GMM) can be used to approximate the stochastic distribution of rain streaks, we represent real-world rain streaks as a Gaussian mixture model, GMM_{real}, as follows:

$$\mathcal{R} \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathcal{R} |\, \mu_k, \textstyle\sum_k), \tag{10}$$

where π_k, μ_k and $\sum_k$ represent the mixing coefficient, mean and variance of the Gaussian distribution, respectively.

The negative log likelihood function imposed on real-world rain streaks can be written as follows:

$$L_{real}(\mathcal{R}; \textstyle\prod, \sum) = - \sum\limits_{n=1}^{N} \log \sum\limits_{k=1}^{K} \pi_k \mathcal{N}(\mathcal{R} |\, 0, \sum_k). \tag{11}$$

In order to reduce the discrepancy between the synthetic rain streak data domain and the real-world rain streak data domain, Kullback-Leibler divergence between the Gaussian G_{syn} learned from synthetic rain streaks and the aforementioned Gaussian mixture model GMM_{real} learned from real rain streaks is minimized as:

$$L_{KL} = \min_{k} D_{KL}(G_{syn} \,\| GMM_{real}^{k}), \tag{12}$$

where GMM_{real}^{k} is the kth component of GMM_{real}.

Finally, the complete objective loss function of the generator is defined as:

$$\begin{aligned} L_{total} &= w_0(L_{g_1} + L_{g_2}) + w_1(L_{perc_1} + L_{perc_2}) \\ &+ w_2(L_{style_1} + L_{style_2}) + L_{tv_1} + L_{tv_2} \\ &+ w_3 L_d + w_4 L_{real} + w_5 L_{KL}, \end{aligned} \tag{13}$$

where w_0, w_1, w_2, w_3, w_4 and w_5 are constant hyper-parameters.

C. Discriminative Network

We propose two discriminative networks. While *Discriminator 1* is used to authenticate rain-free image, *Discriminator 2* is used to authenticate the generated rainy subaperture views (combined from the derained subaperture views and extracted rain-streaks images).

The generated rainy LFI combined from the derained LFI and rain-steak images can be defined as:

$$V_{rg} = V_{rv} + V_{rs}, \tag{14}$$

where V_{rv} is defined in Eq. 6. V_{rs} is the estimated rain-streak images. V_{rg} is the generated rainy images.

Discriminator 1 authenticates the generated rain-free images. It contains 6 3D convolutional layers, each of which has a perceptual region size of $3 \times 3 \times 3$. In order to reduce the memory cost of the 3D convolution layers, a max-pooling layer after every two 3D convolutional layers is adopted to reduce the number of parameters. A dilated convolution is also used to increase the size of the receptive field. Finally, a fully connected layer is added to authenticate the input subaperture views. Parameters of the discriminator are listed in the Supplemental. The loss function of *Discriminator 1* is defined as:

$$L_{dis1} = -\log(D(V'_{rv})) - \log(1 - D(V_{rv})), \tag{15}$$

where V_{rv} is the recovered rain-free subaperture views, and V'_{rv} is the ground truth defined in Eq. 6.

The network architecture of *Discriminator 2* is the same as *Discriminator 1*, and the loss function of *Discriminator 2* is defined as:

$$L_{dis2} = -\log(D(V_{t*}) - \log(1 - D(V_{rg})), \tag{16}$$

where V_{t^*} is the input 3D EPI of the ground truth rainy LFI.

When the outputs of *Discriminator 1* on the de-rained LFIs tend to be false, *Discriminator 1* shares parameters directly with *Discriminator 2* to optimize the generator. When *Discriminator 1* cannot distinguish the de-rained LFIs from the ground truth, *Discriminator 2* will optimize the generator via identifying the generated rainy LFIs. When the outputs of *Discriminator 2* on the generated rainy LFIs tend to be false, the rain-streak image generator branch still needs to be optimized. When fixing *Discriminator 1* and *Discriminator 2*, BR-net is trained to optimize the parameters in order to produce samples that cannot be distinguished from the ground truth. Then, the outputs of *Discriminator 1* and *Discriminator 2* on the generated samples will more likely be true.

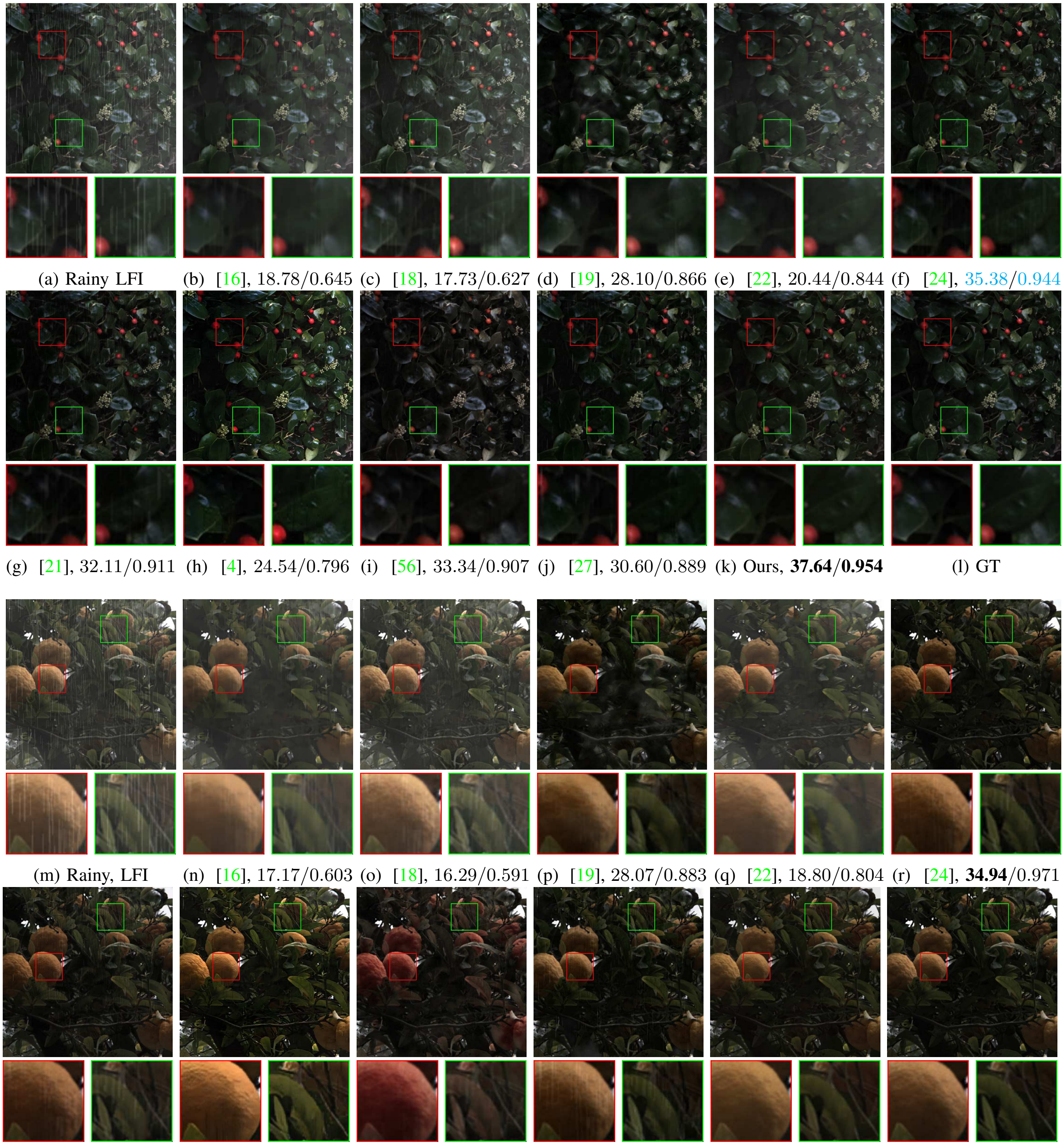
V. Experimental Results And Discussion

A. Training Details

Our LFIs have an angular resolution of 9×9, and spatial resolution of 512×512. Due to the memory size limitation of the NVIDIA RTX Titan GPU, a rainy LFI is cut into small patches of 128×128, and the stride size is 64. To evaluate the effectiveness of our method, twelve synthetic real-world-like rainy LFIs are tested, as shown in Fig. 6, 7, 11 and 12. More results can be found in the Supplemental. In addition, we have also captured some real-world rainy LFIs to evaluate our method, as shown in Fig. 8. They are captured in various outdoor raining scenes, such as plants (Fig. 6 and 8), houses (Fig. 11), cars (Fig. 8 and 12). These LFIs contain both close-up scenes (Fig. 6, 7 and 8) and distant scenes (Fig. 11 and 12), with diverse disparity information.

Our network is implemented using Keras 1.0.5, and Python 3.6.0, and then trained on a PC with an i7-7700K CPU, 24GB memory size, and an NVIDIA RTX TITAN GPU with 24GB memory. The epoch is set as 50, and the batch size is set as 4. Adam optimization is adopted to train our network. The *initial learning rate*, $beta1$, $beta2$ and $epsilon$ are set as 0.0001, 0.9, 0.999 and $1.0e{-}8$, respectively. The parameters of the generator network are updated on each sample, but the parameters of tje discriminator are adjusted on each set of 5 samples. The weight parameters of the generator and discriminator networks are set as 10 and 1, respectively. The constant hyperparameters in Eq. 13 are set as $w_0 = 10$, $w_1 = 0.5$, $w_2 = 0.5$, $w_3 = 10$, $w_4 = 0.5$ and $w_5 = 1e - 9$.

The parameters of our network are first updated by training on the synthesized rainy LFI. The real-world rainy LFI is then fed into the network to obtain the predicted rain-free image



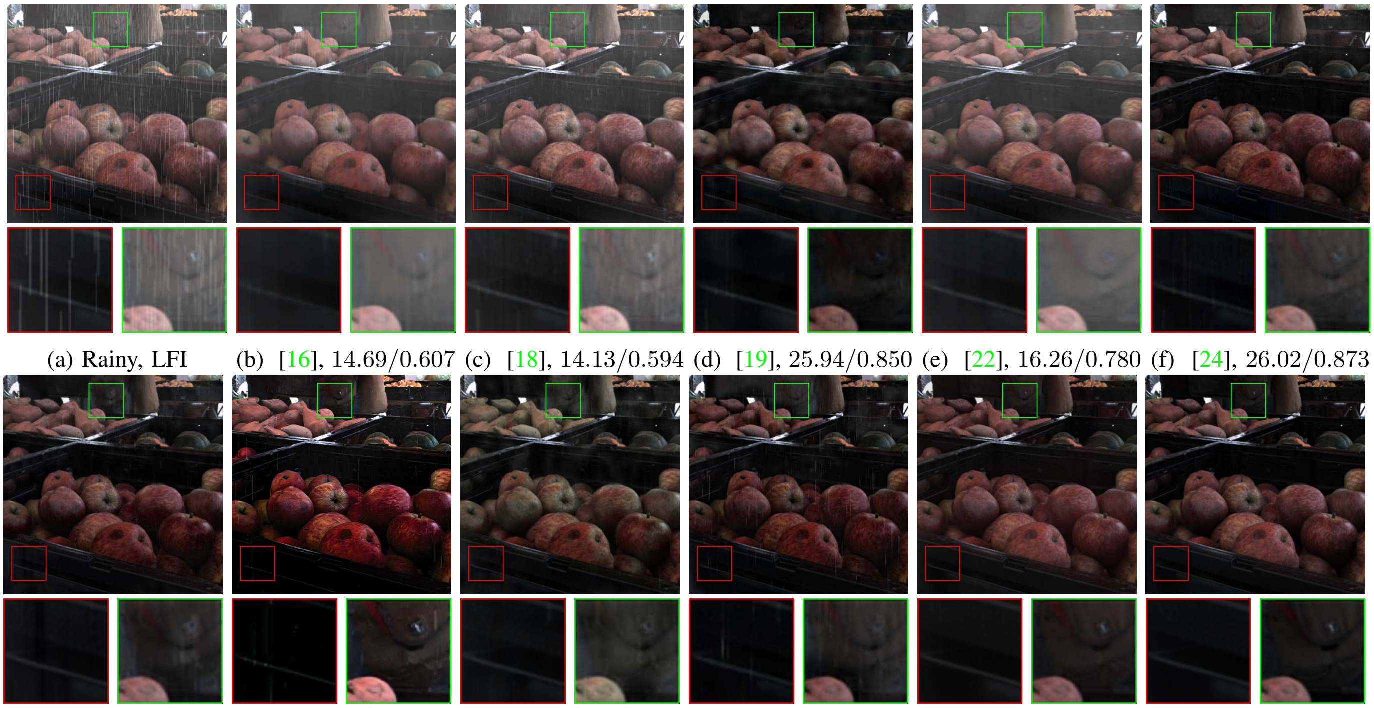
(a) Rainy LFI (b) [16], 18.78/0.645 (c) [18], 17.73/0.627 (d) [19], 28.10/0.866 (e) [22], 20.44/0.844 (f) [24], 35.38/0.944

(g) [21], 32.11/0.911 (h) [4], 24.54/0.796 (i) [56], 33.34/0.907 (j) [27], 30.60/0.889 (k) Ours, **37.64**/**0.954** (l) GT

(m) Rainy, LFI (n) [16], 17.17/0.603 (o) [18], 16.29/0.591 (p) [19], 28.07/0.883 (q) [22], 18.80/0.804 (r) [24], **34.94**/0.971

(s) [21], 30.36/0.923 (t) [4], 25.87/0.850 (u) [56], 29.99/0.908 (v) [27], 32.19/0.929 (w) Ours, 34.33/**0.955** (x) GT

Fig. 6: Comparison of different methods for rain streaks removal on LFIs. The de-rained images of each method are marked with their PSNR/SSIM values. The best values are marked in **bold**, and the second best are marked in cyan.



(a) Rainy, LFI (b) [16], 14.69/0.607 (c) [18], 14.13/0.594 (d) [19], 25.94/0.850 (e) [22], 16.26/0.780 (f) [24], 26.02/0.873

(g) [21], 26.48/0.891 (h) [4], 22.20/0.824 (i) [56], 26.65/0.873 (j) [27], 25.90/0.856 (k) Ours, **33.21**/**0.907** (l) GT

Fig. 7: Comparison of different methods for rain streak removal on a LFI. The de-rained image of each method is marked with its PSNR/SSIM values. The best values are marked in **bold**, and the second best are marked in cyan.

and rain-streak image. The Gaussian mixture model is used to simulate the distribution of real-world rain streaks, and the expectation maximization (EM) algorithm is used to iteratively solve the Gaussian mixture model.

B. *Quantitative Evaluation*

We compare our method with the state-of-the-art classical optimization-based rain removal methods [16] and deep learning-based methods [18] [22] [19] [24] [21] [4] [27] for single 2D images. Two typical metrics, Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity Index (SSIM), are adopted for evaluation of our method and the other methods on the central subaperture view of each LFI.

Due to the learning ability of deep learning, deep-learning based rain removal methods [18] [22] [19] [24] [21] [4] [56] [27] [32] usually perform much better than methods based on classical optimization, e.g., [16], in almost all cases.

The PSNR and SSIM values of the de-rained central subaperture views obtained by our method are higher than those obtained by the other methods, especially in close-up scenes with complex structures, as shown in Fig. 6 and 7, and distant scenes with fog/mist, as shown in Fig. 11 and 12. Our method has better PSNR performances on most of the challenging examples. For few scenes as shown in Fig. 6 and 11, it seems that the PSNR values obtained by our method are lower than those obtained by [24]. In these cases, results of our method are the second best, and the gap between our results and the best ones is minor.

We can also see that the SSIM performances of our method match with those of the state-of-the-art method [24]. Our method and [24] perform much better than the other methods [16] [18] [19] [22] [21] [56] [27]. Although the SSIM values obtained by [24] on some scenes, as shown in Fig. 11 and 12, are higher than those of ours, our derained results appear to remove rain streaks better visually, as shown in Fig. 11. To make things worse, [24] introduces geometric distortion to the pillar marked with a green box in Fig. 11r.

To demonstrate the effect of the non-local model in *DE-Net*, we have also conducted an ablation study, as shown in Table I. The PSNR values of the de-rained images obtained by our method with the non-local model are obviously higher than those without the non-local model. Similarly, the SSIM values of the derained images produced by our method are higher than those generated by our method without the non-local model in most cases. Hence, the non-local model is very useful and necessary in our proposed network. Table II shows the ablation study on the hyper-parameters of our proposed network, in which each value represents the average result of all tested 12 rainy LFIs.

Metric	Our method	Fig. 6	Fig. 7	Fig. 11	Fig. 12
PSNR	without NL	31.48	30.60	25.97	30.52
	with NL	**35.98**	**33.21**	**33.15**	**34.87**
SSIM	without NL	0.939	0.903	**0.941**	0.940
	with NL	**0.954**	**0.907**	0.940	**0.948**

TABLE I: Ablation study of our method with/without the non-local model in *DE-Net*.

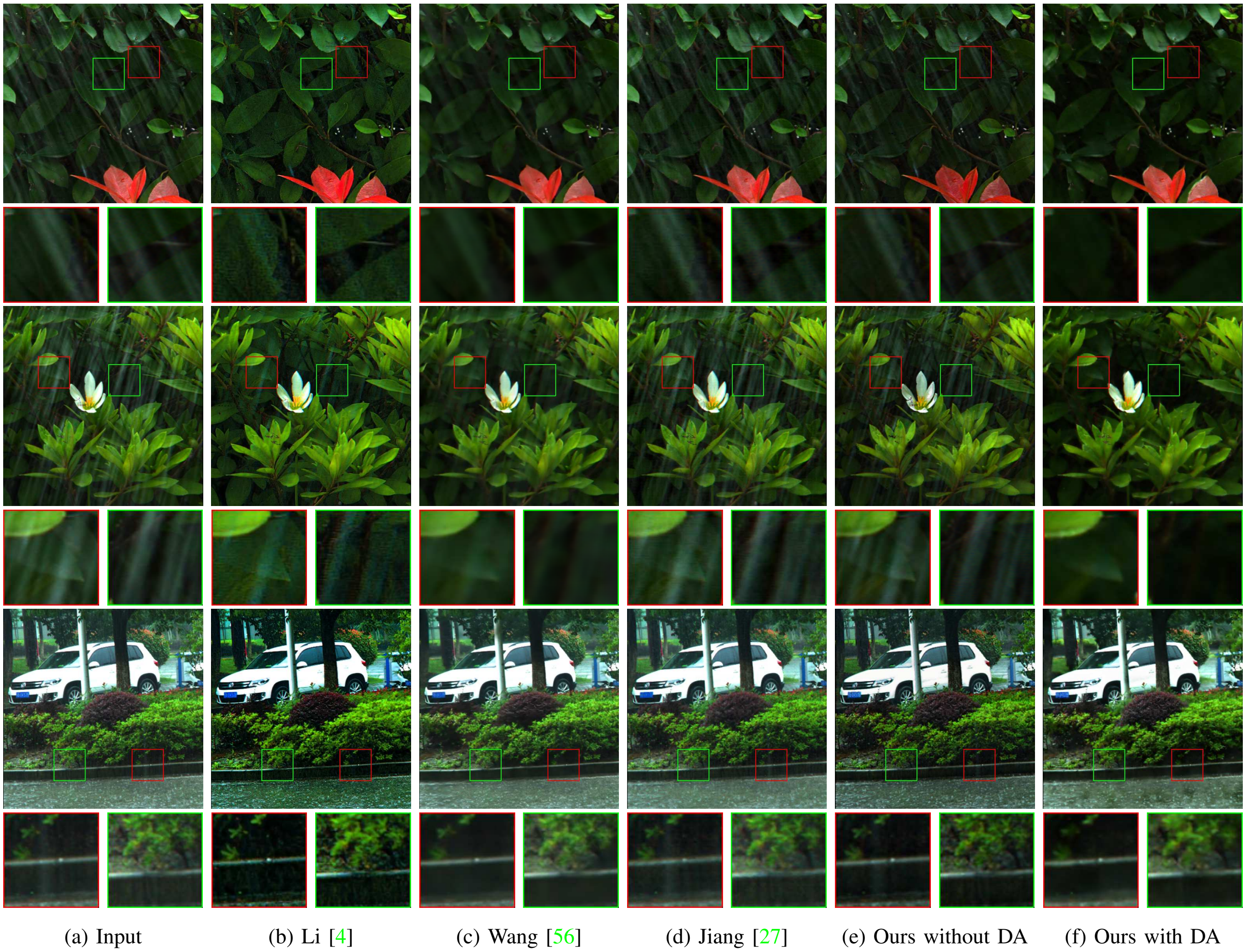


Fig. 8: Comparison of different rain-streak removal methods on real-world LFIs. (e) are the results of our method without domain adaption (DA) semi-supervised learning, and (f) are the results of our method with DA semi-supervised learning.

w_1	0			0.5			1		
w_2	0	0.5	1	0	0.5	1	0	0.5	1
PSNR	30.34	33.15	30.13	32.21	**34.23**	33.75	29.87	31.65	32.13
SSIM	0.898	0.918	0.901	0.913	**0.936**	0.926	0.885	0.931	0.921

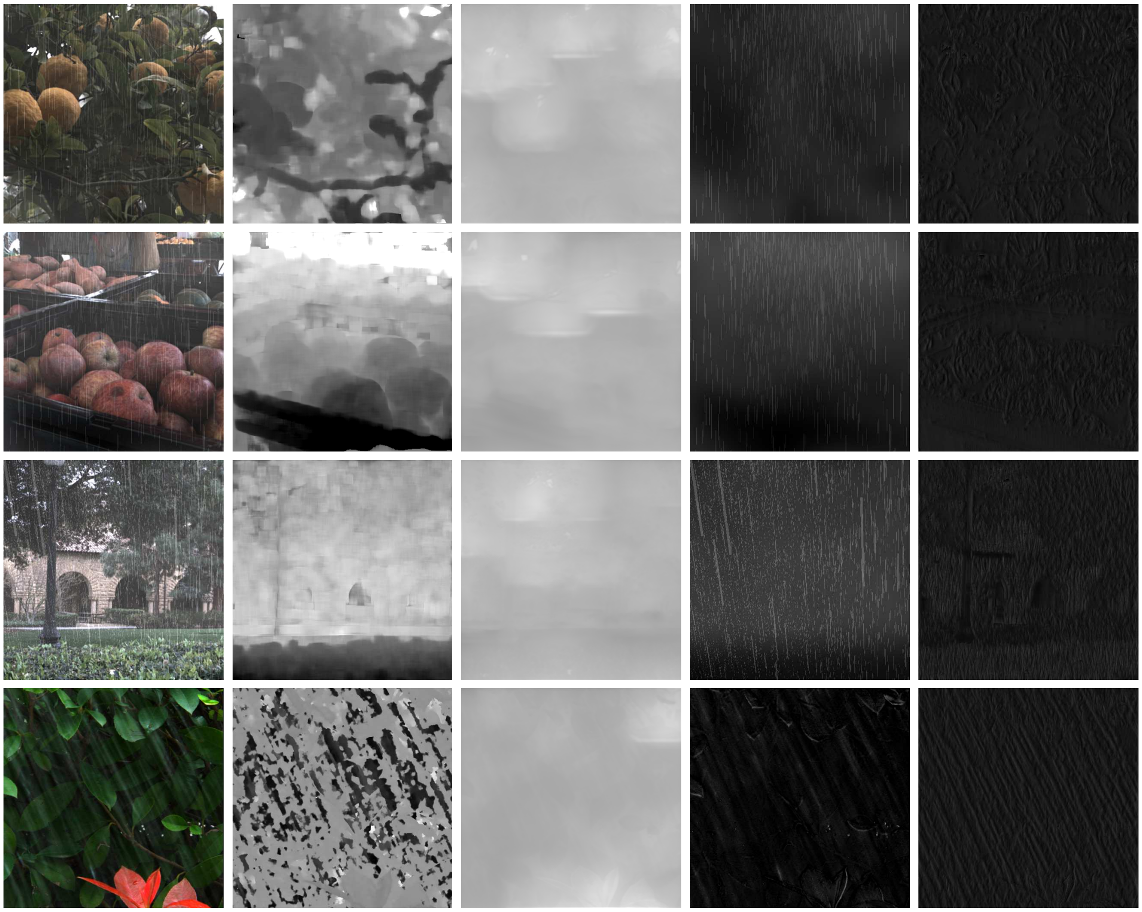
TABLE II: Ablation study on the basic components and the hyper-parameters of our proposed network.

We also compare our proposed method with the state-of-the-art video rain steak removal methods [33] [34] on 3D EPIs, as shown in Fig. 13. We can see that the video rain steak removal methods [33] [34] cannot produce acceptable results. They have low PSNR and SSIM values, introduce blur to the image details and fail to remove mist/veiling in the derained results. This shows that video rain streak removal methods cannot be used directly for rain streak removal on LFIs.

C. Qualitative Evaluation

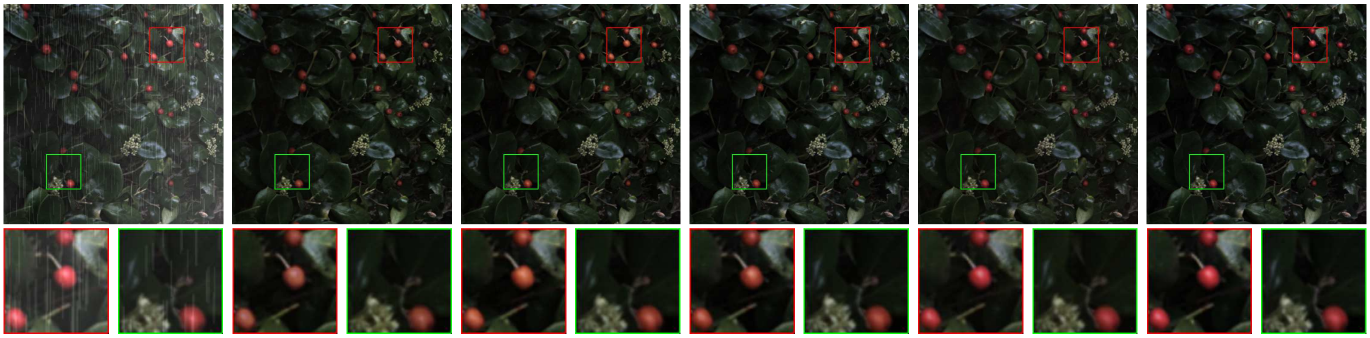
Although method [16] can remove some tiny rain streaks well, it can neither successfully remove large rain streaks nor tiny rain streaks completely, and it blurs the de-rained images. There are still some rain streaks obviously visible in the resulting images shown in Fig. 6, 7, 11 and 12. In addition, since it cannot remove fog/mist in the de-rained images, its de-rained images appear brighter than the ground truth.

The deep learning-based methods [18] [19] [22] generally perform well in most cases. However, they still have some flaws. Method [18] does not perform well in handling rainy images with small rain streaks or large and dense rain streaks, as shown in Fig. 6, 7, 11 and 12. Meanwhile, it also fails to remove fog/mist in the derained images, as shown in Fig. 11 and 12. The reason may be that as [18] adopts a filter to divide the input image into high-frequency layer and low-frequency layer and only the high-frequency layer is processed to remove rain streaks, the low-frequency layer that contains large rain



(a) Input (b) Our disparity maps (c) Depth maps of [4] (d) Our rain streaks (e) Rain streaks of [4]

Fig. 9: Comparisons of disparity maps and rain streaks obtained by our method and the method [4].



(a) Input (b) L_1 (c) L_1 + L_{style} (d) L_1 + L_{perc} (e) Ours (f) GT

Fig. 10: Ablation study of our method.

streaks and/or fog/mist is not processed.

The performance of [19] is much better than those of methods [16] [18]. However, it fails to effectively remove large rain streak, as shown in Fig. 12. The reason may be that as it is specially designed to deal with rain drop removal, it does not work very well for rain streak removal.

Method [22] can successfully remove rain streaks in Fig. 6 and 7. However, it fails to deal with heavy and dense raining scenes, as shown in Fig. 12. It also fails to deal with fog/mist, causing the resulting images to be brighter than the ground truth, and it always blurs the recovered images, as shown in Fig. 6, 7, 11 and 12.

Method [24] performs very well on rain streak removal and fog/mist removal. In the scenes shown in Fig. 11, its performance is comparable to our method. However, in some examples shown in Fig. 6 and 7, it fails to detect and remove all tiny rain streaks. In addition, it usually darkens the whole derained images as shown in Fig. 11 and 12. The network structure of [21] is similar to [24]. Its performance is also similar to [24].

Due to the introduction of depth/disparity information, the dehazing effect of method [4] is remarkable. However, this method causes the derained images to be sharp and oversaturated, as shown in Fig. 6, 7 and 12. In addition, for heavy rain images, the rain removal effect is unacceptable, as shown in Fig. 11 and 12.

Although method [56] can remove most of the rain streaks well, it fails to obtain correct colors after training on our dataset. Method [27] fails to remove large rain streaks well, as shown in Fig. 6, 11 and 12. The colors and brightness of their results appear to be similar to those of [24] [21] [4] and ours. Their PSNR and SSIM values are also good, as shown in Fig. 6 and 7. However, the colors of the zoom-in patches for method [56] in Fig. 6 and 7 still appear a bit different and unnatural, and the zoom-in patches for both [56] [27] still contain a lot of blurs and unremoved rain strips.

Our proposed method takes a 3D EPI as input to detect rain streaks and recover the disoccluded regions previously covered by rain streaks. It is by exploiting the abundant information embedded in LFIs. Our method can properly preserve the geometric/textural details of the recovered LFIs than all the other compared methods, as shown in the regions highlighted by red/green boxes in Fig. 11. Hence, it always performs better than the other methods.

We have also evaluated the rain removal effect on real-world LFIs, as shown in Fig. 8. We can see that our method removes rain streaks more cleanly than the other methods [4] [56] [27] in these challenging examples. Our method under supervised training without domain adaption from synthetic rain streaks to real-world rain streaks has limited performances. However, its performances under semi-supervised training by utilizing both synthetic LFIs and real-world LFIs are greatly improved.

Fig. 9 shows the disparity maps and rain streaks obtained by our method and those by method [4]. Our method produces more accurate disparity maps and rain streaks than [4]. However, since the perspective viewpoint, light and contrast of each subaperture view may be slightly different, as shown in regions highlighted by the red/green boxes in Fig. 14, our network is slightly more complex in order to handle all these issues. In addition, since the input 3D EPI is larger than a single image, the memory cost and processing costs of our method are higher than the other state-of-the-art methods.

Fig. 10 shows an ablation study of our method with/without various energy terms defined in the objective energy function in the synthetic scene and virtual scene with greater disparities. We can see that the results of our method with all energy terms used produce more realistic results than the other alternatives, in which there are blur and/or chromatic aberration introduced into the resulting images.

VI. Conclusion and Future Work

In this paper, we have proposed a novel adversarial network for rain streak removal from LFIs. Our method is the first method specifically proposed for rain streak removal on LFIs. We have carefully created a synthetic real-world-like rainy LFI dataset for training and evaluation of our proposed method. Our method takes 3D EPIs of LFIs as input. It first estimates the disparity maps for each subaperture views, and then detects rain steaks and recovers derained subaperture views. Our method is trained in a semi-supervised manner to overcome the domain adaption from synthetic LFIs to real-wrold LFIs. Fog/mist embedded in the rainy LFIs can also be clearly removed together with rain streaks. Our method takes full advantages of the textural and structural information of the input 3D EPI to detect and remove rain streaks from LFIs. Extensive experiment demonstrates that our method outperforms state-of-the-art methods designed for rain streak removal on 2D images/videos.

References

[1] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE CVPR*, 2004.

[2] K. Garg and S. K. Nayar, "Vision and rain," *IJCV*, vol. 75, p. 327, October 2007.

[3] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, "Depth-attentional features for single-image rain removal," in *Proc. IEEE CVPR*, 2019.

[4] R. Li, L. Cheong, and R. T. Tan, "Heavy rain image restoration: Integrating physics model and conditional adversarial learning," in *Proc. IEEE CVPR*, 2019.

[5] Y.-J. Tsai, Y.-L. Liu, M. Ouhyoung, and Y.-Y. Chuang, "Attention-based view selection networks for light-field disparity estimation," in *Proc. AAAI*, vol. 34, pp. 12095–12103, 2020.

[6] Y. Piao, Z. Rong, M. Zhang, X. Li, and H. Lu, "Deep light-field-driven saliency detection from a single view," in *Proc. IJCAI*, 2019.

[7] "Raytrix," Accessed 31 July 2020. https://raytrix.de/.

[8] "Lytro," Accessed 31 July 2020. https://en.wikipedia.org/wiki/Lytro.

[9] Y. Ni, J. Chen, and L.-P. Chau, "Reflection removal on single light field capture using focus manipulation," *IEEE TCI*, vol. 4, no. 4, pp. 562–572, 2018.

[10] T. Li, D. P. K. Lun, Y. Chan, and Budianto, "Robust reflection removal based on light field imaging," *IEEE TIP*, vol. 28, no. 4, pp. 1798–1812, 2019.

[11] T. Li, Y. H. Chan, and D. P. K. Lun, "Improved multiple-image-based reflection removal algorithm using deep neural networks," *IEEE TIP*, vol. 30, pp. 68–79, 2021.

[12] L. Yu, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *IEEE CVPR*, 2016.

[13] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. IEEE ICCV*, 2015.

[14] Y. L. Chen and C. T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proc. IEEE ICCV*, 2013.

[15] L. Kang, C. Lin, and Y. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE TIP*, vol. 21, pp. 1742–1755, April 2012.

[16] Y. Li and M. S. Brown, "Single image layer separation using relative smoothness," in *Proc. IEEE CVPR*, 2014.

[17] H. Li, Y. Wang, Z. Yang, R. Wang, X. Li, and D. Tao, "Discriminative dictionary learning-based multiple component decomposition for detail-preserving noisy image fusion," *IEEE TIM*, vol. 69, no. 4, pp. 1082–1102, 2020.

[18] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE TIP*, vol. 26, pp. 2944–2956, June 2017.

[19] 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 CVPR*, pp. 2482–2491, 2018.

[20] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin, "Non-locally enhanced encoder-decoder network for single image de-raining," in *Proc. ACM MM*, 2018.

[21] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: a better and simpler baseline," in *Proc. IEEE CVPR*, 2019.

[22] W. Yang, R. T. Tan, J. Feng, J. Liu, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. IEEE CVPR*, 2017.

[23] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE CVPR*, 2018.

[24] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. ECCV*, 2018.

[25] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proc. IEEE CVPR*, 2019.

[26] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE TCSVT*, vol. 30, no. 11, pp. 3943–3956, 2020.

[27] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *Proc. IEEE CVPR*, 2020.

[28] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE CVPR*, 2018.

[29] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE CVPR*, 2017.

[30] D. Ren, W. Shang, P. Zhu, Q. Hu, D. Meng, and W. Zuo, "Single image deraining using bilateral recurrent network," *IEEE TIP*, vol. 29, pp. 6852–6863, 2020.

[31] W. Luo, J. Lai, and X. Xie, "Weakly supervised learning for raindrop removal on a single image," *IEEE TCSVT*, 2020.

[32] L. Zhu, Z. Deng, X. Hu, H. Xie, X. Xu, J. Qin, and P.-A. Heng, "Learning gated non-local residual for single-image rain streak removal," *IEEE TCSVT*, 2020.

[33] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng, "Video rain streak removal by multiscale convolutional sparse coding," in *Proc. IEEE CVPR*, 2018.

[34] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *Proc. IEEE CVPR*, 2018.

[35] W. Yang, J. Liu, and J. Feng, "Frame-consistent recurrent video deraining with dual-level flow," in *Proc. IEEE CVPR*, 2019.

[36] W. Yang, R. T. Tan, S. Wang, and J. Liu, "Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence," in *Proc. IEEE CVPR*, 2020.

[37] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 926–954, 2017.

[38] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. So Kweon, "Accurate depth map estimation from a lenslet light field camera," in *Proc. IEEE CVPR*, 2015.

[39] H. Zhu, Q. Wang, and J. Yu, "Occlusion-model guided antiocclusion depth estimation in light field," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 965–978, 2017.

[40] J. Y. Lee and R. Park, "Complex-valued disparity: Unified depth model of depth from stereo, depth from focus, and depth from defocus based on the light field gradient," *IEEE TPAMI*, pp. 1–1, 2019.

[41] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field reconstruction using deep convolutional network on epi," in *Proc. IEEE CVPR*, 2017.

[42] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Light field reconstruction using shearlet transform," *IEEE TPAMI*, vol. 40, no. 1, pp. 133–147, 2018.

[43] M. Le Pendu, X. Jiang, and C. Guillemot, "Light field inpainting propagation via low rank matrix completion," *IEEE TIP*, vol. 27, no. 4, pp. 1981–1993, 2018.

[44] C. Tan, J. Chen, and L. Chau, "Edge-preserving rain removal for light field images based on rpca," in *Proc. DSP*, 2017.

[45] T. Yang, X. Chang, H. Su, N. Crombez, Y. Ruichek, T. Krajnik, and Z. Yan, "Raindrop removal with light field image using image inpainting," *IEEE Access*, vol. 8, pp. 58416–58426, 2020.

[46] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edgeconnect: Generative image inpainting with adversarial edge learning," *arXiv:1901.00212*, 2019.

[47] X. Hong, P. Xiong, R. Ji, and H. Fan, "Deep fusion network for image completion," in *Proc. ACM MM*, 2019.

[48] T. Yan, Y. Ding, F. Zhang, N. Xie, W. Liu, Z. Wu, and Y. Liu, "Snow removal from light field images," *IEEE Access*, vol. 7, pp. 164203–164215, 2019.

[49] "Blender," Accessed 31 July 2020. https://www.blender.org/.

[50] "Stanford lytro light field archive," Accessed 31 July 2020. http://lightfields.stanford.edu/LF2016.html.

[51] S. Heber, W. Yu, and T. Pock, "Neural epi-volume networks for shape from light field," in *Proc. ICCV*, 2017.

[52] Y. Wang, F. Liu, Z. Wang, G. Hou, Z. Sun, and T. Tan, "End-to-end view synthesis for light field imaging with pseudo 4dcnn," in *Proc. ECCV*, 2018.

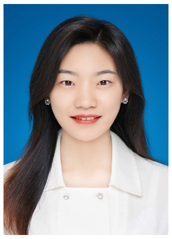
[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.

[54] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*, 2016.

[55] T. Dapeng, C. Jun, Y. Zhengtao, Y. Kun, and W. Lizhen, "Domain-weighted majority voting for crowdsourcing," *IEEE TNNLS*, vol. 30, no. 1, pp. 163–174, 2018.

[56] 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 CVPR*, 2019.

Yuyang Ding received his bachelor degree in Software Engineering from Shanxi University, China, in 2016. He is now pursuing his MS degree in Jiangnan University. His research interests include computer vision and image processing.



Mingyue Li received her bachelor degree in Engineering Management from Jilin University, China, in 2019. She is now pursuing her MS degree in Jiangnan University. Her research interests include computer vision and image processing.

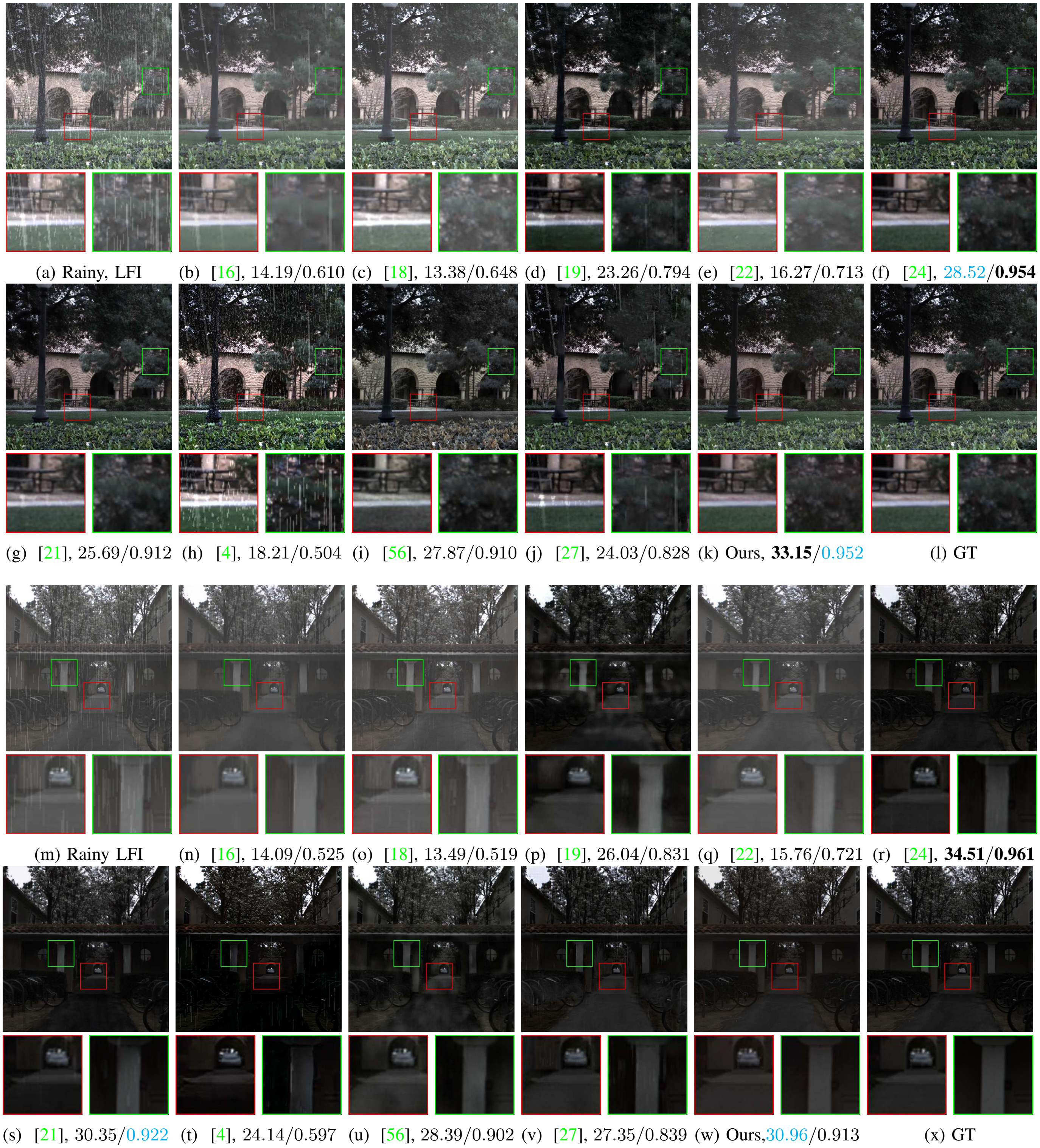
Tao Yan received his bachelor degree in computer science and technology from Northwest University in China, and dual Ph.D. degrees in computer science from City University of Hong Kong (CityU) and University of Science and Technology of China (USTC). He is now an associate professor with the School of Artificial Intelligence and Computer Science, Jiangnan University in China. His research interests include computer vision and image editing.

Fan Zhang received her bachelor degree in Digital Media Technique from Jiangnan University, China, in 2020. She is now pursuing her MS degree in Beijing University of Posts and Telecommunications. Her research interests include computer vision and image processing.

Yuan Liu is currently a professor with the School of Artificial Intelligence and Computer Science, Jiangnan University in China. His research interests include network security and artificial intelligence.

Rynson W.H. Lau received his Ph.D. degree from University of Cambridge. He was on the faculty of Durham University and is now with City University of Hong Kong.

Rynson serves on the Editorial Boards of Computer Graphics Forum, and Computer Animation and Virtual Worlds. He has served as the Guest Editor of a number of journal special issues, including ACM Trans. on Internet Technology, IEEE Trans. on Multimedia, IEEE Trans. on Visualization and Computer Graphics, and IEEE Computer Graphics & Applications. Rynson's research interests include computer graphics and computer vision.



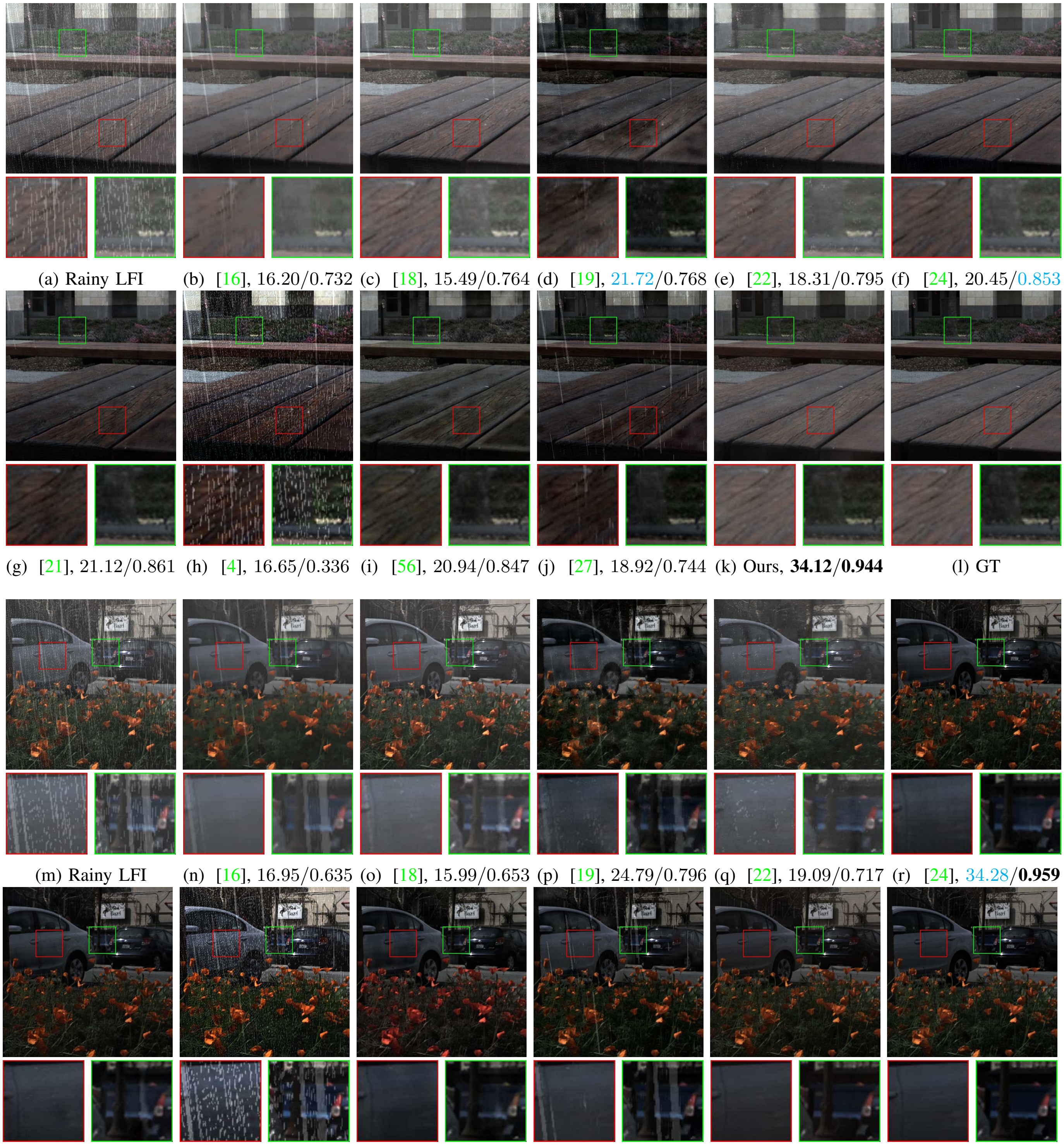
(a) Rainy, LFI (b) [16], 14.19/0.610 (c) [18], 13.38/0.648 (d) [19], 23.26/0.794 (e) [22], 16.27/0.713 (f) [24], 28.52/**0.954**

(g) [21], 25.69/0.912 (h) [4], 18.21/0.504 (i) [56], 27.87/0.910 (j) [27], 24.03/0.828 (k) Ours, **33.15**/0.952 (l) GT

(m) Rainy LFI (n) [16], 14.09/0.525 (o) [18], 13.49/0.519 (p) [19], 26.04/0.831 (q) [22], 15.76/0.721 (r) [24], **34.51**/**0.961**

(s) [21], 30.35/0.922 (t) [4], 24.14/0.597 (u) [56], 28.39/0.902 (v) [27], 27.35/0.839 (w) Ours,30.96/0.913 (x) GT

Fig. 11: Comparison of different rain streak removal methods on LFIs. The derained images of each method are shown with their PSNR/SSIM values. The best values are marked in **bold**, and the second best are marked in cyan.



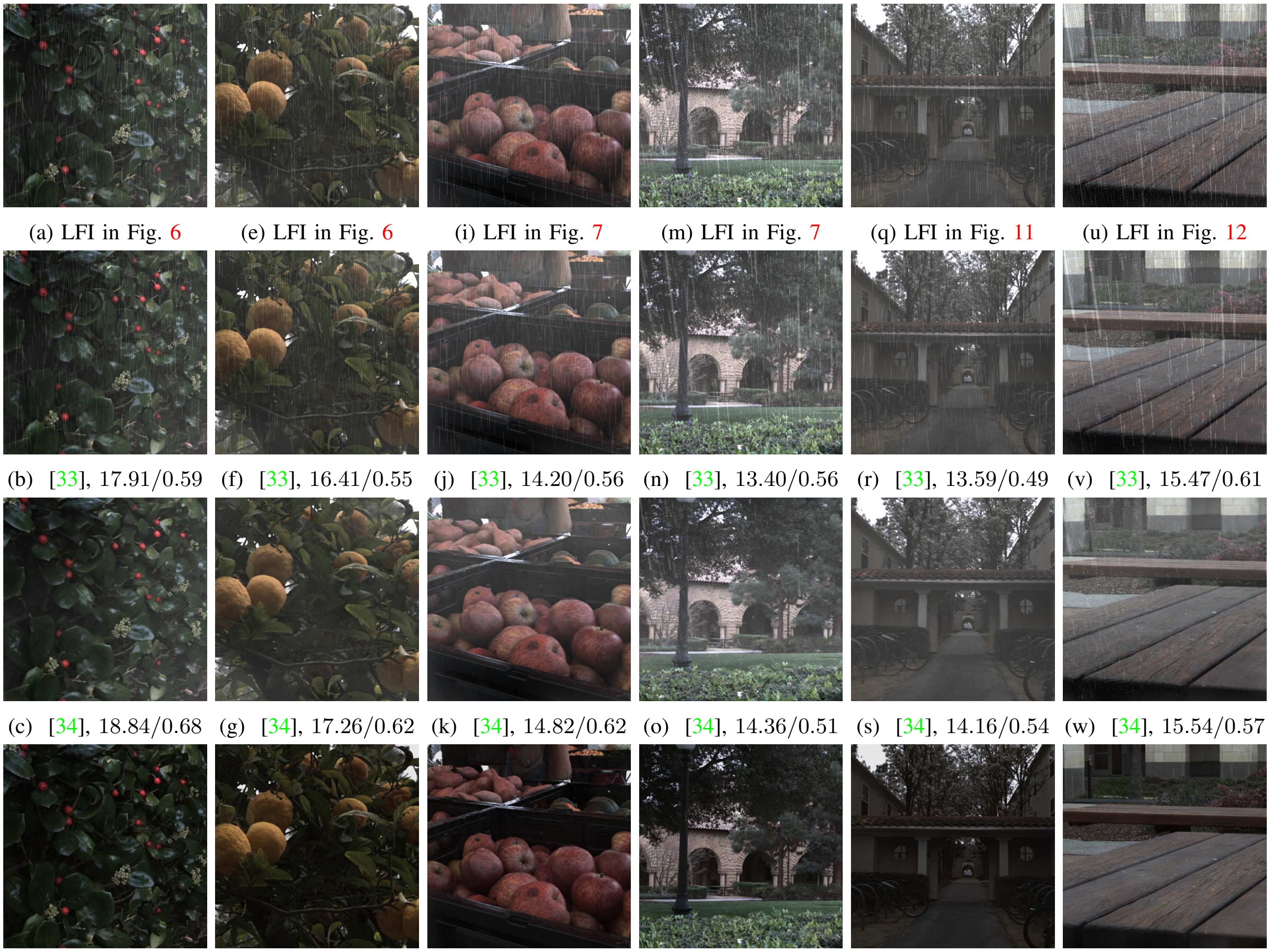
(a) Rainy LFI (b) [16], 16.20/0.732 (c) [18], 15.49/0.764 (d) [19], 21.72/0.768 (e) [22], 18.31/0.795 (f) [24], 20.45/0.853

(g) [21], 21.12/0.861 (h) [4], 16.65/0.336 (i) [56], 20.94/0.847 (j) [27], 18.92/0.744 (k) Ours, **34.12**/**0.944** (l) GT

(m) Rainy LFI (n) [16], 16.95/0.635 (o) [18], 15.99/0.653 (p) [19], 24.79/0.796 (q) [22], 19.09/0.717 (r) [24], 34.28/**0.959**

(s) [21], 30.55/0.923 (t) [4], 18.43/0.462 (u) [56], 31.22/0.929 (v) [27], 26.61/0.843 (w) Ours, **35.62**/0.952 (x) GT

Fig. 12: Comparison of different rain streak removal methods on LFIs. The derained images of each method are shown with their PSNR/SSIM values. The best values are marked in **bold**, and the second best are marked in cyan.



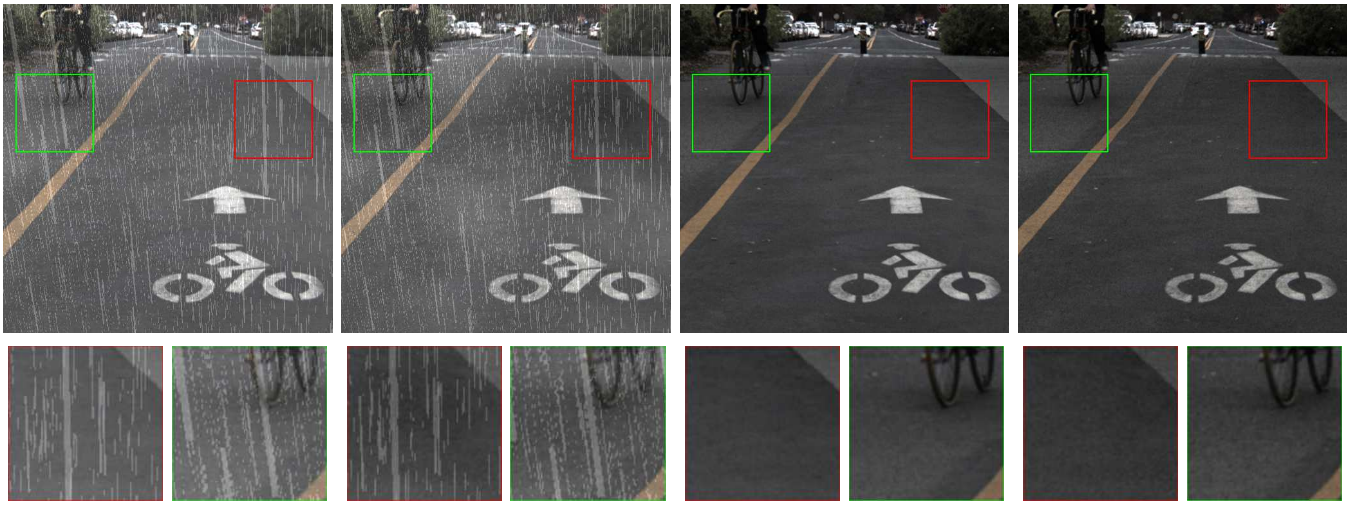
(a) LFI in Fig. 6 (e) LFI in Fig. 6 (i) LFI in Fig. 7 (m) LFI in Fig. 7 (q) LFI in Fig. 11 (u) LFI in Fig. 12

(b) [33], 17.91/0.59 (f) [33], 16.41/0.55 (j) [33], 14.20/0.56 (n) [33], 13.40/0.56 (r) [33], 13.59/0.49 (v) [33], 15.47/0.61

(c) [34], 18.84/0.68 (g) [34], 17.26/0.62 (k) [34], 14.82/0.62 (o) [34], 14.36/0.51 (s) [34], 14.16/0.54 (w) [34], 15.54/0.57

(d) ours, **37.64/0.95** (h) ours, **34.33/0.95** (l) ours, **33.21/0.91** (p) ours, **35.35/0.96** (t) ours, **30.96/0.91** (x) Ours, **31.96/0.92**

Fig. 13: Comparing our method with state-of-the-art video rain streak removal methods [33] [34], taking 3D EPIs of rainy LFIs as input. Derained subaperture views produced by each method are shown with PSNR/SSIM values.



(a) Rainy sub-view(0,0) (b) Rainy sub-view (0,8) (c) De-rained sub-view (0,0) (d) De-rained sub-view (0,8)

Fig. 14: Rainy/original subviews and derained subviews of one real-world rainy LFI. Note that the intensity of the same regions in different subviews may be different.