Rain Removal from Light Field Images with 4D Convolution and Multi-scale Gaussian Process

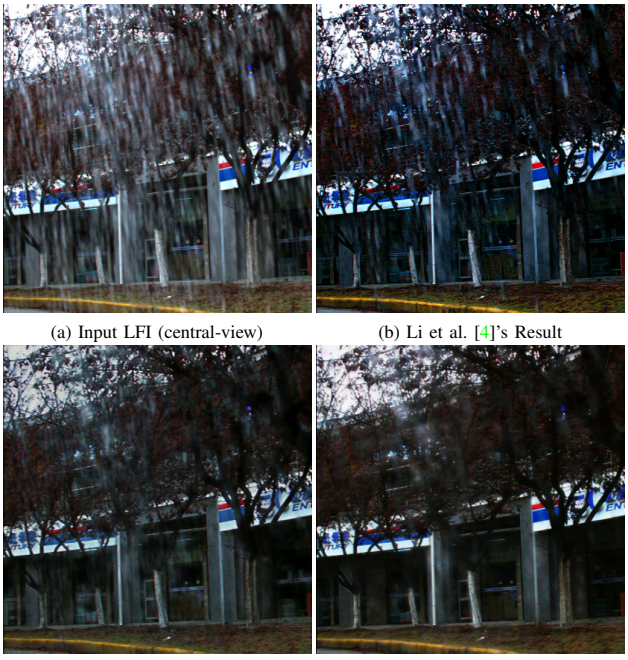
Tao Yan*, *Member, IEEE*, Mingyue Li, Bin Li, Yang Yang, Rynson W.H. Lau, *Senior Member, IEEE*

***Abstract*—Existing deraining methods mainly focus on a single input image. However, with just a single input image, it is extremely difficult to accurately detect and remove rain streaks, in order to restore a rain-free image. In contrast, a light field image (LFI) embeds abundant 3D structure and texture information of the target scene by recording the direction and position of each incident ray via a plenoptic camera, which has emerged as a popular device in the computer vision and graphics research communities. However, making full use of the abundant information available from LFIs, such as 2D array of sub-views and the disparity map of each sub-view, for effective rain removal is still a challenging problem. In this paper, we propose a novel network, 4D-MGP-SRRNet, for rain streak removal from LFIs. Our method takes as input all sub-views of a rainy LFI. In order to make full use of the LFI, we adopt 4D convolutional layers to build the proposed rain steak removal network to simultaneously process all sub-views of the LFI. In the proposed network, the rain detection model, MGPDNet, with a novel Multi-scale Self-guided Gaussian Process (MSGP) module is proposed to detect high-resolution rain streaks from all sub-views of the input LFI at multi-scales. Semi-supervised learning is introduced for MSGP to accurately detect rain streaks by training on both virtual-world rainy LFIs and real-world rainy LFIs at multi-scales via calculating pseudo ground truths for real-world rain streaks. We then feed all sub-views subtracting the predicted rain streaks into a 4D convolution-based Depth Estimation Residual Network (DERNet) to estimate the depth maps, which are later converted into fog maps. Finally, all sub-views concatenated with the corresponding rain streaks and fog maps are fed into a powerful rainy LFI restoring model based on the adversarial recurrent neural network to progressively eliminate rain streaks and recover the rain-free LFI. Extensive quantitative and qualitative evaluations conducted on both synthetic LFIs and real-world LFIs demonstrate the effectiveness of our proposed method.**

***Index Terms*—Light field images, rain removal, 4D Convolution, semi-supervised learning, gaussian process.**

I. Introduction

SEVERE weather conditions, such as raining, snowing and fogging, can degrade the quality of outdoor captured images/videos and the performances of computer vision systems. Rain streaks are semi-transparent and have various sizes, directions and even appearances (e.g., strips and fog/mist) depending on their distances from the camera. Thus, to accurately detect rain streaks and eliminate them for a clean rain-free image is a nontrivial task. To solve this problem, researchers have proposed many rain streak removal methods based on classical optimization methods or deep neural networks.



(a) Input LFI (central-view) (b) Li et al. [4]'s Result

(c) Yasarla et al. [5]'s Result (d) Our Result

Fig. 1. Recovered real-world de-rained LFIs (only center views are shown) obtained by two state-of-the-art methods [4], [5] and our method.

An LFI can be decoded into a set of sub-views of different perspectives. Accurate disparity maps can also be inferred from LFIs. Thus, we believe that rain removal on LFIs [1] and stereo (Dual-Pixel) images [2] ([3]) may produce much higher quality de-rained results than those on single images, by properly exploring and exploiting multi-views and depth maps. Leveraging the advantages of plenoptic cameras, LFIs could also be used to improve the performances of a variety of computer vision applications, such as surveillance and autonomous navigation systems. Meanwhile, we observed that the same rain streaks always occluded different areas of the background in different sub-views, which means that the rain-free LFI could be well restored from a rainy LFI by fully exploiting the abundant information from an LFI [1], as shown in Fig. 1 and 2. In this paper, we propose a novel neural framework for removing rain streaks from a rainy LFI to recover a rain-free LFI.

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

Tao Yan, Mingyue Li and Bin Li are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China.

Yang Yang is with the Department of Computer Science, Jiangsu University, Zhenjiang, China.

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

Corresponding author is Tao Yan, E-mail: yantao.ustc@gmail.com.

[1]Our dataset and code will appear at https://github.com/YT3DVision/4D-MGP-SRRNet.

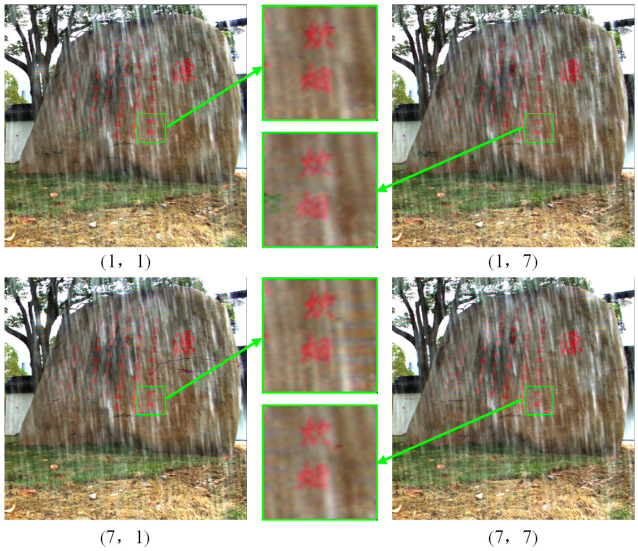


Fig. 2. Sub-views of a real-world rainy LFI. It can be seen that the same rain streaks appear at different positions and occlude different regions of the background in different sub-views.

Traditional image rain streak removal methods [6]–[9] usually perform poorly under complex rain conditions. Recently, deep-learning-based methods [4], [5], [10]–[21] [20]–[27] have shown significant advantages. Although these methods can successfully remove most of the rain streaks, there are always some tiny or large rain streaks that cannot be clearly removed. Generally speaking, these methods have three main limitations. First, none of them can accurately detect various types of real-world rain streaks [28] that have different directions/shapes and varying degrees of opacity, although accurately detecting rain streaks is a crucial step in the rain streak removal task. In view of the fact that rain streaks are continuous and translucent stripes, which resemble the shape of small blood vessels, accurate rain streak detection is a challenging task. Second, distant dense rain streaks are more like fog/mist. Existing methods still have certain limitations in eliminating such dense rain streaks. Third, due to the differences between synthetic rainy images and real-world rainy images, most algorithms trained on synthetic images do not work well on real scenes. An example is shown in Fig. 1.

Recently, Wei et al. [29] and Ding et al. [1] propose using the Gaussian Mixture Model (GMM) to construct semi-supervised learning methods for modeling rain streaks in real-world images by approximating a feature vector of real-world rain streaks as the weighted average of a set of feature vectors of the rain streaks in synthetic virtual-world images. The consistency between the two kinds of rain streaks is ensured by minimizing the Kullback-Leibler (KL) divergence between their distributions. Both methods use real-world images to generalize the rain streak detection ability from the virtual-world domain to the real-world domain. However, in the early stages of training, the learned GMM parameters are usually inaccurate. Thus, it may not be appropriate to minimize the KL divergence in order to reduce the difference between the two distributions, as it would lead to sub-optimal performances. To overcome this problem, Yasarla et al. [5] propose using the Gaussian Process (GP) for rain streak detection, which is a non-parametric model to predict real-world rain streaks based on virtual-world rain streaks. It achieves better performances than those of [29]. To improve the generalization of our network for deraining on real-world rainy LFIs, we also adopt GP to realize semi-supervised learning by supervising the feature vector of real-world rain streaks as a weighted average of the feature vectors of synthetic rain streaks. On the other hand, to overcome the problem of insufficient utilization of information available from an LFI (i.e., only utilizing an array of sub-views in the same row/column of an LFI, called a 3D EPI, each time) of the former method [1], we introduce 4D convolution to make full use of all sub-views of an LFI for effective and efficient LFI deraining.

In this paper, we propose a novel CNN-based method for rain streaks removal from LFIs. The architecture of our method is shown in Fig. 3. Our network takes all sub-views of a rainy LFI as input. 4D convolutional layers that can simultaneously process all sub-views are introduced for the proposed network. In addition, semi-supervised learning with the MSGP (Multi-scale Self-guided GP) module is also introduced to improve the effectiveness and generalization of our network for rain streak detection and rain-free LFI recovery, by training on both synthetic LFIs and real-world LFIs.

Our proposed network consists of three sub-networks. First, a Multi-scale Gaussian Process based Dense Network (MGPDNet) is constructed to extract high-resolution rain streaks from all sub-views. A 4D Depth Estimation Residual Network (DERNet) is then introduced to estimate the depth maps for all sub-views subtracting the estimated rain streaks. The predicted depth maps are later converted into fog maps, which can be used to remove distant dense rain streaks like fog/mist. Finally, all input rainy sub-views concatenated with the corresponding rain streaks and the fog maps are fed into a Recurrent Neural Network with Adversarial Training (RNNAT) to progressively remove the rain streaks and recover the rain-free sub-views. The main contributions of our work can be summarized as:

- We propose a novel 4D Convolution and Multi-scale Gaussian Process based Semi-supervised Rain Removal Network (4D-MGP-SRRNet), which takes all sub-views of a rainy LFI as input to accurately detect rain streaks and recover a rain-free LFI.
- We propose a Multi-scale Self-guided GP (MSGP) module based semi-supervised learning strategy for accurate rain streak detection and rain-free LFI recovery.
- We propose a new rainy LFI dataset, Rainy LFI with Motion Blur (RLMB), which includes 400 sets of synthetic rainy LFIs with ground truth rain-free LFIs and 200 sets of real-world rainy LFIs. Most importantly, motion blur is first carefully considered in realistic rain streak generation for rainy LFI synthesis and rain streak detection/removal from LFIs.

II. Related Work

In the past few years, with the popularity of deep learning, great progress has been achieved for rain removal from images [28]. In this section, we review works related to rain streak removal from images, videos and LFIs.

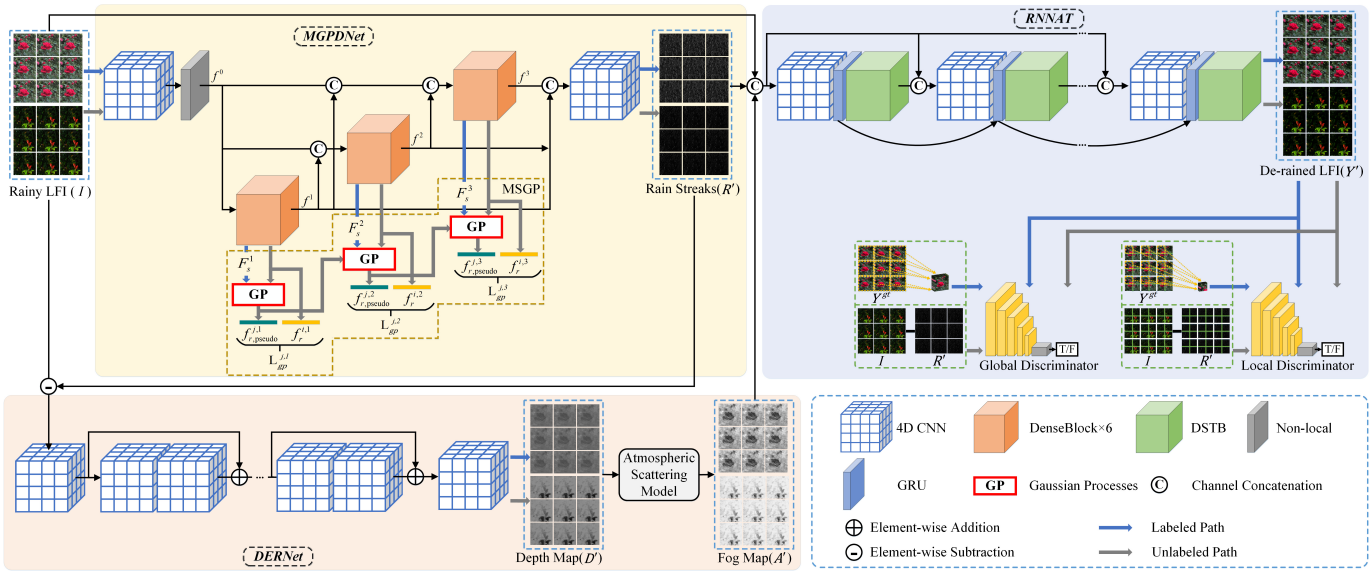


Fig. 3. Network architecture of our proposed 4D-MGP-SRRNet, which contains three parts (i.e., sub-networks). First, the *Multi-scale Gaussian Process Dense Network (MGPDNet)* takes as input all sub-views of a rainy LFI, (i.e., a 5D data of cropped patch stacks), and works in a semi-supervised learning manner on multi-scale features to detect high-resolution rain streaks. Second, estimated rain streaks subtracted from sub-views are fed into the 4D *Depth Estimation Residual Network (DERNet)* to estimate the depth maps. An atmospheric scattering model is also adopted to convert the predicted depth maps into fog maps. Finally, all rainy sub-views concatenated with the corresponding rain streaks and fog maps estimated above are fed into the *Recurrent Neural Network with Adversarial Training (RNNAT)* to progressively removes rain streaks and recover the rain-free LFI.

A. Rain Streak Removal From Images

Traditional deraining methods usually separate a rainy image into rain-streak layer and rain-free layer based on image priors and classical optimization models such as low-rank model [6], sparse coding [7], Gaussian Mixture Model [8]. These methods are time-consuming and their performances are unsatisfactory on complex real-world rainy images. Recently, deep-learning-based rain removal methods [4], [5], [10]–[12], [14], [16], [17], [19], [22], [29]–[47] [20]–[25], [27] have attracted much attention and shown impressive performances.

Zhang et al. [10] propose a residual-aware rain-density classifier to measure the density of rain streaks, and a multi-stream densely connected deraining network to extract features of rain streaks with different scales and shapes. Jiang et al. [36] propose a multi-scale progressive fusion network for collaborative representation of rain streaks. Although these two methods that rely on multi-scale networks to remove rain streaks can restore the details of the rain-free background excellently, their performances in completely removing rain streaks are ordinary. Recently, Zhang et al. [41] propose a conditional GAN-based framework for image deraining. Wang et al. [40] propose a new physical rainy image model to explicitly model the degradation of the haze-like effect on rainy images. Wang et al. [42] propose a kernel-guided convolutional neural network to explore and exploit motion blur and line pattern appearances of rain streaks.

RNN-based methods have also been proposed to progressively remove rain streaks by decomposing rain streaks into multiple rain streak layers with different sizes, directions and densities. Li et al. [32] propose the first representative deep recurrent network for image deraining, which uses contextual dilated networks to acquire large receptive field and recurrent neural networks to decompose the rain removal task into multiple stages. Yang et al. [15], [30] propose a recurrent multi-task deep learning network that learns the binary rain streak map, the appearance of rain streaks, and the rain-free background in each recurrence. Zamir et al. [38] propose a multi-stage architecture for progressive image restoration, which balances spatial details and high-level contextualized information.

Memory neural networks and attention blocks are also used to remove rain streaks from images [12], [14], [16], [33], [37], [44]. Ren et al. [14] propose the bilateral recurrent network, in which two recurrent networks having recurrent layers (convolutional LSTM) in each stage are coupled to simultaneously obtain rain streak layer and clean background layer. Bilateral LSTMs are proposed to propagate deep features across stages and bring interplay between the two networks. Zhu et al. [12] propose a gated non-local deep residual learning framework to remove rain from images. Similarly, Lin et al. [37] propose a network based on sequential dual attention blocks for rain streak removal. Li et al. [33] propose a non-locally enhanced encoder-decoder network for image deraining, which consists of a series of non-locally enhanced dense blocks and adopts pooling indices guided decoding scheme. Jiang et al. [44] propose an attention-guided deraining network to model multiple rain streak layers under the multiple network stages. Ahn et al. [16] propose a rain streak removal network using the maximum color channel selection. Most recently, Fu et al. [17] propose the first GNN-based model for iteratively removing rain streaks.

Depth map estimation for rain streak removal is also studied [4], [11], [39]. Hu et al. [11] investigate visual effects of rain streaks subjected to scene depth, and propose a rainy

image generation model that is composed of rain streaks and fog. A deep neural network is proposed to estimate depth-attentional features by leveraging the depth-guided attention mechanism and regressing on a residual map to restore the de-rained image. However, since the depth maps predicted from rainy images are unreliable, it may seriously affect the quality of de-rained results. Recently, Hu et al. [39] improve the network speed for it to run in real time. Li et al. [4] propose a neural network with a physics-based backbone to separate the entangled rain streaks and rain accumulation, and a depth-guided GAN refinement step to eliminate heavy rain steaks and fog. However, the network fails to eliminate distant rain streaks, especially under heavy rain conditions.

Recently, several works [5], [19], [29], [34], [35], [43] [27] improve the deraining performances on real-world scenes by adopting semi-/un-supervised learning and/or real-world rainy images. Wang et al. [34] propose a semi-automatic method to incorporate temporal priors and human supervision to generate high-quality clean de-rained images from each input sequence of real rainy images. Wei et al. [29] propose a deraining neural network to solve the domain adaption problem of transferring from synthetic images to real-world images by designing a semi-supervised learning strategy. Yasarla et al. [5] propose a semi-supervised learning framework based on the Gaussian Process, which enables the network to use synthetic datasets for learning and unlabeled real-world images for fine-tuning. Zhu et al. [35] propose the first unsupervised learning rain streak removal method, which alleviates the popular paired training constraints by introducing a physical model that explicitly learns a recovered image and corresponding rain streaks from the differentiable programming perspective. It is worth noting that CycleGAN [31] is a commonly used network for unsupervised raindrop removal [43] and rain streak removal [19]. Luo et al. [43] propose a weakly supervised learning based raindrop removal network, which can be trained with both pairwise and unpaired samples. Wei et al. [19] explore the unsupervised single-image rain removal problem using unpaired data, and proposed an unsupervised network called DerainCycleGAN for rain streak removal and generation. Huang et al. [45] propose an encoder-decoder network augmented with a self-supervised memory module in the bottleneck, which enables the network to exploit the properties of rain streaks from both synthetic and real data. Wang et al. [27] propose a task transfer learning mechanism to learn favorable representations from real data through task transfer to improve deraining generalization to real scenes. Notably, Li et al. [48] propose an Unsupervised and Untrained Single-Image Dehazing Neural Network (YOLY), which performs image dehazing by only using the information contained in the observed single hazy image and avoids training itself on an image set with ground-truth.

These semi-supervised/unsupervised-based single-image deraining methods help improve the performances on various real-world images. Unfortunately, the de-rained images generated by these methods still have some large/tiny rain streaks left, especially for input images of challenging rain scenarios.

More recently, along with the Transformer model mitigating the shortcomings of CNNs, such as limited receptive field and inadaptability to input content, several state-of-the-art networks [20], [21], [23]–[25], [46], [47] have been proposed for image restoration (deraing, denoising, super-resolution, motion deblurring, defocus deblurring). Wang et al. [23] propose Uformer with a U-shaped structure formed by locally-enhanced window (LeWin) Transformer blocks that performs nonoverlapping window-based self-attention instead of global self-attention. However, Uformer can only enable communications between adjacent windows. Liu et al. [47] propose the first Swin Transformer-based network called SwinIR for image super-resolution and image denoising. The Residual Swin Transformer block (RSTB) composing of several Swin Transformer layers together with a residual connection is proposed for deep feature extraction in this network. Valanarasu et al. [24] propose TransWeather, which is a Transformer-based end-to-end model with just a single encoder and a decoder for restoring images degraded by any weather conditions. Specifically, intra-patch transformer blocks within Transformer encoder are proposed to enhance attention inside the patches, and Transformer decoder with learnable weather type embeddings is introduced to learn the weather degradation type and uses that information to restore clean images. The latest works [20], [21] are all Swin Transformer-based networks. Zamir et al. [21] propose an efficient hierarchical architecture called Restormer for high-resolution image restoration by making several key designs in building blocks, such as multi-head attention and feed-forward network (FFN). Xiao et al. [20] propose an effective and efficient Transformer-based encoder-decoder architecture for single image deraining, with each encoder/decoder block constructed by several window-based transformer modules that capture local relationships within the local window and a spatial transformer module that complements the locality by modeling cross-window dependencies. In addition, Li et al. [25] propose a contrastive-learning-based All-In-One image restoration network, AirNet, which could recover images from unknown corruption types and levels. Unlike the networks [46], [49] that treat multiple degradations as a multi-task learning problem with multiple input and output heads, all these networks [20], [21], [23]–[25], [47] are single pass network, which does not differentiate different corruption types and ratios.

B. Rain Streak Removal From Frame Sequences

Video deraining methods [26], [50]–[55] exploited spatial and temporal redundancies in frame sequences. They are important in practical applications.

Li et al. [50] propose a multi-scale convolutional sparse coding model, which uses multiple convolution filters convolved on the sparse feature map, to deliver repetitive local patterns of rain streaks, and further uses multi-scale filters to represent rain stripes of different scales. Liu et al. [51] build a joint recurrent deraining network. The network seamlessly integrates rain degradation classification, spatial texture appearances-based rain streaks removal, and temporal coherence-based background detail reconstruction. Yang et al. [52] propose a two-stage recurrent framework built with dual-level flow regularizations to perform the inverse recovery process of the

rain synthesis model for video deraining. Yang et al. [53] propose a two-stage self-learned framework for deraining based on both temporal correlation and consistency. Li et al. [54] propose an online multi-scale convolutional sparse coding model constructed for encoding dynamic rain/snow and background motions with temporal variations.

Zhang et al. [26] propose an end-to-end video deraining framework, called ESTINet, which takes the advantage of deep residual networks and convolutional long short-term memory. It can capture the spatial features and temporal correlations among successive frames at a small computational cost. Yang et al. [55] proposed an augmented Self-Learned Deraining Network called SLDNet+ to remove both rain streaks and rain accumulation by utilizing temporal correlation, consistency, and rain-related priors. Yan et al. [56] propose a two-stage (i.e., the single image module and the multiple frame module) video-based raindrop removal network, which is based on temporal correlation.

Video rain removal methods usually assume that the locations of rain steak across neighboring frames are uncorrelated, which is not like the motion of background layers. Although rain streak removal from videos has been carefully studied, the problem of unclean rain removal still exists and the performance evaluated on each single frame is worse than those of rain streak removal methods for single images.

C. Rain Streak Removal From LFIs

Rain removal methods for LFIs are still in their infancy. The earlier method [57] proposes to first align all sub-views with the central sub-view. Robust principal component analysis is then applied to decompose each image of a set of deformed sub-views into low-rank data and sparse data. Finally, a dark view image is introduced to estimate non-rainfall disparity edges from sparse data whose remaining part is considered as rain, and the non-rainfall disparity edges are restored back to a low-rank image to generate the rain-free LFI. Ding et al. [1] proposed a GAN-based architecture for removing rain streaks from 3D EPI of a rainy LFI. It first estimates depth maps, and then simultaneously detects rain streaks and restores rain-free sub-views by exploiting the correlation between rain streaks and the clean background layer. However, it can only exploit the sub-views in the same row/column of a rainy LFI for rain streak removal at each time.

In conclusion, since these methods have difficulties in accurately predicting various rain streaks from challenging real-world LFIs, they may not be able to recover high-quality rain-free LFIs from various rainy LFIs. To address this limitation, we propose a novel and more effective deep-learning-based method to eliminate rain streaks from rainy LFIs.

III. Rain Imaging Model and Rainy LFI Dataset

The widely used rain imaging model [10] [58] regards a rainy image as a linear combination of the rain-free background and the rain streak/drop layer. Yang et al. [30] discovered that in heavy rain, rain streaks of various shapes and directions would overlap with each other. Thus, they proposed a rain imaging model consisting of multiple layers of rain streaks and a global atmospheric light caused by atmospheric veiling effect as well as blur. Later, Hu et al. [11] found that visual effects of rain subject to scene depth and proposed another rainy image generation model, i.e., a rainy image was a composition of a rain-free image, a rain streak layer and a fog layer, in which both the rain streak and fog transmissions depend on the depth.

In complex real-world scenes, high-speed falling rain always produces motion blur in the captured images. This phenomenon will seriously hinder the rain streaks removal task. Wang et al. [42] proposed a rain streak observation model based on the motion blur of rain streaks, and proposed a neural network to learn the angle and the length of the motion blur kernel from the detail layer of a rainy image. The motion blur kernel is then stretched into a degradation map for rain streak prediction. However, none of the existing rainy image generation models or datasets have taken motion blur of rain streaks into account.

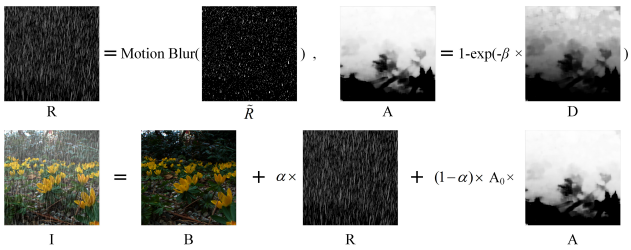


Fig. 4. Our rainy LFI generation process. $\tilde{R}$ represents the static rain streaks without the motion blur effect. R denotes the rain streak layer after adding the motion blur effect. I is the rainy image, and B is the rain-free layer. D is the depth map of B and A is the fog/mist layer.

We carefully take into account motion blur of rain streaks and propose a novel rainy image generation model, which is much closer to real situation, as shown in Fig. 4. Comparing with [42], our rainy image generation model is an improved additive degradation model with considering the fog/mist effect defined as:

$$I = B + \alpha \mathrm{M_b}(R) + (1-\alpha) A_0 A, \tag{1}$$

where I denotes the rainy image. B is the rain-free image coming from the real-world LFI dataset [59] captured by a Lytro Illum camera. R is the rain streak layer. $\mathrm{M_b}(\cdot)$ represents a motion blur operation. A_0 represents the global atmospheric light, whose value is assumed to be a constant [11]. A denotes the fog/mist layer. α is a constant parameter used to adjust the relative intensity between the rain-streak layer and fog layer, empirically set as 0.6 in our experiment.

The particle system of Blender [60] is adopted to simulate 3D rain streaks for our rainy LFI generation. A large number of rain streaks with various shapes, directions and densities are generated. At the same time, the simulated rain streaks are added with a motion blur effect [61], which is supplied by the Motion Blur function of the Render module of Blender.

The visual intensity of fog increases exponentially with the scene depth [11], i.e., degrades linearly with the transmission map T [62]. Specifically, the fog layer can be modeled as:

$$A = 1 - T = 1 - e^{-\beta D}, \tag{2}$$

where D refers to the depth map. β is empirically set as 1.8 to control the thickness of the fog. Larger β means thicker fog.

Since the LFI dataset [59] does not provide ground-truth disparity maps for each LFI, the depth estimation method [63] is adopted to estimate acceptable depth maps for our rainy LFI generation. Finally, the rain-free LFIs, rain-streak layers and fog layers are combined as defined in Eq. 1 to generate the vivid rainy LFIs. A total number of 400 synthetic rainy LFIs and their ground truth rain-free LFIs are generated. In addition, we captured 200 real-world LFIs with a Lytro ILLUM camera to construct a real-world rainy LFI subset for learning the proposed LFI rain removal network.

IV. LFI Rain Detection and Removal Network

We propose the 4D-MGP-SRRNet to detect rain streaks and recover the rain-free LFI from a rainy LFI, as shown in Fig. 3. The network consists of three parts: the rain streak detection network MGPDNet, the depth estimation sub-network DERNet, and the recurrent rain streaks removal sub-network RNNAT. To make full use of all sub-views of a rainy LFI, 4D convolutions [64] actually implemented using 3D convolutions are chosen to construct our rain removal network 4D-MGP-SRRNet. Specifically, 4D convolutional layers are used to explore and exploit all sub-views of an LFI in more meaningful 5D feature space by learning the complicated interactions of spatial and angular representations in our network.

4D convolution can surpass 3D convolution and 2D convolution in many video processing [64] and LFI processing tasks. 4D convolution [64] is originally proposed to model both short- and long-term spatio-temporal representations simultaneously, and overcome short-term spatio-temporal representations modeled by 3D convolution from videos of RGB frames. We fund that such 4D convolution can be used to properly and effectively model the interaction of sub-views intra and inter *3D-EPIs* of an LFI, as shown in Fig. 5, by just considering each *3D-EPI* as an *action unit* of video-level representation for action recognition [64]. For LFI processing, 3D convolution can only model and exploit the interaction of sub-views within a *3D-EPI*, and 2D convolution can not properly model the structural relationship between all sub-views of an LFI for exploring and exploiting as much meaningful information as possible. Therefore, by utilizing 4D convolutional layers to build our network, our network can effectively explore and exploit all sub-views of an LFI for rain streak detection and removal. ReLU activation [65] is followed each 4D convolutional layer in our network.

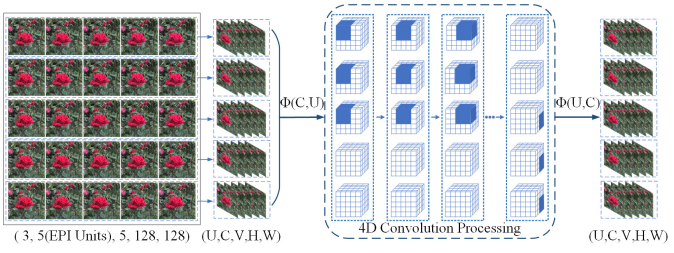


Fig. 5. 4D Convolution operates on an LFI. Φ() means permuting two specified dimensions of the input data.

In our network, for an input rainy LFI with the spatial resolution of $[512, 512]$ and the angular resolution of $[5, 5]$, we crop each sub-view to small patches with the resolution of $[128, 128]$. Then, corresponding patches cropped from the same position of sub-views in the same row are arranged as a *3D-EPI unit* [64]. In this way, a total of 5 *3D-EPI units* can be obtained to form 5D data with the size of $5(\textit{3D-EPI units}) \times 3 \times 5 \times 128 \times 128$, where 3 refers to the channels of sub-views. As shown in Fig. 5, the channel dimension permutes with the *3D-EPI unit* dimension before the 5D data is fed into a 4D convolutional layer, and then the output is permuted reversely to 3D form for subsequent 3D convolutional layers [64]. At the beginning of MGPDNet shown in Fig. 3, a non-local block [66] followed the 4D convolutional layer is used to exploit self-attention for rough intra- and inter-sub-views rain streak features.

A. Rain Streak Detection

Leveraging the densely-connected block [8], [67], we construct rain streak detection subnetwork MGPDNet shown in Fig. 3 to detect rain streaks over different scales of features via three different branches (from bottom to top) that consist of several 4D dense blocks with convolution kernels of $3 \times 3 \times 3 \times 3$, $5 \times 5 \times 5 \times 5$, and $7 \times 7 \times 7 \times 7$, respectively. MGPDNet uses novel progressive fusion instead of direct fusion to integrate the extracted features. The rain streak detection process can be expressed as:

$$\begin{aligned} f^1 &= \mathrm{Dense}_1(f^0), \\ f^2 &= \mathrm{Dense}_2([f^0, f^1]), \\ f^3 &= \mathrm{Dense}_3([f^0, f^1, f^2]), \\ R' &= \mathcal{F}_{\mathrm{conv}}([f^1, f^2, f^3]), \end{aligned} \tag{3}$$

where f^0 denotes the rain streak features extracted by a 4D convolution block from the input LFI I and then processed by a non-local block [66] to enhance the features of rain streaks. Specifically, the non-local block is used to make self-attention for rain streak features by exploiting their correlation intra- and inter-sub-views. $\mathrm{Dense}(\cdot)$ represents the convolution operation of dense blocks. f^1, f^2 and f^3 are extracted features from different branches. $\mathcal{F}_{\mathrm{conv}}(\cdot)$ is introduced to produce the final rain streaks by merging the rain streak features detected on multi-scales, and R' refers to the predicted rain streaks.

In our MGPDNet, we use GP models to jointly model the distribution of synthetic data features and real-world data features on multi-scale dense block branches for supervising unlabeled real-world rain streak detection and improving the generalization of our method.

Gaussian Process: A Gaussian Process (GP) is a collection of random variables, any finite number of which have consistent joint Gaussian distributions [68]–[70]. A GP $\mathrm{g}(\cdot)$ is fully specified by its mean function $\mathrm{m}(\cdot)$ and covariance function $\mathrm{K}(\cdot)$ as:

$$\mathrm{g}(x) \sim \mathcal{GP}(\mathrm{m}(x), \mathrm{K}(x, x')), \tag{4}$$

where x and x' are the possible inputs that index the GP and

$$\begin{aligned} \mathrm{m(x)} &= \mathbb{E}[\mathrm{g}(x)], &(5)\\ \mathrm{K}(x,x') &= \mathbb{E}[(\mathrm{g}(x)-\mathrm{m}(x))(\mathrm{g}(x')-\mathrm{m}(x'))]. &(6)\end{aligned}$$

Assume that the training sample $\mathrm{g}(x^i)$ and the unseen test sample $\mathrm{g}(x_*^j)$, where $i=\{1,...,n\}$ and $j=\{1,...,r\}$, conform to the Gaussian distribution. Therefore, the formula for conditioning a joint Gaussian distribution is defined as:

$$\begin{bmatrix} \mathrm{g}(X) \\ \mathrm{g}(X_*) \end{bmatrix} \sim \mathbb{N}\left(\begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{bmatrix} \right), \tag{7}$$

where $\mu=\mathrm{m}(X)$, $\mu_*=\mathrm{m}(X_*)$. $\Sigma=\mathrm{K}(X,X)$, $\Sigma_*=\mathrm{K}(X,X_*)$ and $\Sigma_{**}=\mathrm{K}(X_*,X_*)$, which denotes the covariances evaluated at all pairs of training and/or testing points.

When modeling real situations (e.g., real-world rainy LFIs), we always access noisy samples, i.e., $\mathrm{y}(x^i)=\mathrm{g}(x^i)+\epsilon^i$ and $\mathrm{y}(x_*^j)=\mathrm{g}(x_*^j)+\epsilon^j$, where ϵ^i and ϵ^j are independent Gaussian noise $\mathcal{N}(0,\sigma_\epsilon^2)$ [70]. The conditional distribution of $g_*=\mathrm{g}(X_*)$ given $g=\mathrm{g}(X)$ can then be expressed as:

$$\begin{aligned} y_*|y \sim \mathcal{N}(&\mu_*+\Sigma_*^T(\Sigma+\sigma_\epsilon^2 I_n)^{-1}(y-\mu), (\Sigma_{**}+\sigma_\epsilon^2 I_r) \\ &-\Sigma_*^T(\Sigma+\sigma_\epsilon^2 I_n)^{-1}\Sigma_*), \end{aligned} \tag{8}$$

where I_n and I_r are the identity matrices with sizes n and r, respectively.

For the rest of this section, we simplify the notation X_* to x'_* to indicate that we try to predict the resulted features $\mathrm{y}(x'_*)$ for feature vector x'_*, given the training data set including inputs X and corresponding ground-truth/output $\mathrm{g}(X)$. In data preprocessing of GP, the mean value is always subtracted from the training/testing samples, which means $\mu=0$ and $\mu_*=0$.

Multi-scale Self-guided Gaussian Process for Rain Streak Detection: Leveraging the GP, we design the semi-supervised learning process for our MGPDNet. The seminal work [5] introduced GP to model the latent features extracted by the encoder. Our semi-supervised process models synthetic LFI features and real-world LFI features extracted by 4D dense blocks of multi-scale branches with different kernel sizes, as shown in Fig. 3 and 6. The calculated mean of the conditional distribution is treated as the pseudo ground truth of the predicted latent feature vector of a real-world LFI for supervising the feature extraction of the dense blocks. In our multi-scale GP, the extracted latent feature vectors of the synthetic LFI form matrix $F_s^k=\{f_s^{i,k}\}_{i=1}^{N_l}$ for the k^{th} scale/branch, where N_l refers to the total number of training patches cropped from our synthetic rainy LFIs.

Since the inversion of a $N\times N$ matrix is computationally nontrivial while $N>1000$, GP approach is suited to small and medium-size data sets [70]. Therefore, for efficient calculation, we adopt the cosine similarity measure to select N_n nearest labeled latent vectors for one unlabeled vector of the corresponding real-world LFI, instead of utilizing all vectors in F_s^k. N_n is set as 16, which is the same as the channel size of the input features extracted for every pixel of the rainy LFI sub-views. The input feature size for the GP model in each branch is 1×49152 ($3\times 128\times 128$) according to the spatial size $[128,128]$ for every sub-view.

It is worth emphasizing that 4D convolutions exploiting features and correlations across all sub-views of the input 5D LFI patch can obtain the predicted features with abundant information for rain streak detection. 4D convolutions of MGPDNet not only retain the spatial sizes of the predicted features for all sub-views of the input 5D data, but also maintain the number of channels. In this way, large predicted features for each sub-view of the input patch are fed into the following GP in each scale/branch. It should be noted that the distribution of rain streak features in different sub-views of LFI is similar. Therefore, for computational efficiency, GPs in our MSGP only process the rain streak features of the central sub-view of an input patch.

In our MSGP, for modeling features of rain streaks, ground truth for the features $f_s^{i,k}$ extracted from the i^{th} synthetic LFI (patch) at scale k is itself, $f_s^{i,k}$. The pseudo ground truth for the j^{th} testing features $f_r^{j,k}$ in the k^{th} dense block branch is defined as $f_{r,pseudo}^{j,k}=\Sigma_*^T(\Sigma+\sigma_\epsilon^2 I_n)^{-1}F_s^k$ (according to Eq. 8), which also have the size 1×49152.

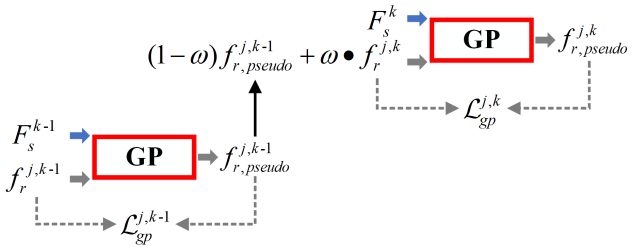


Fig. 6. The working process of the k^{th} GP of the Multi-scale Self-guided GP (MSGP).

In order to fuse the predicted features across different scales and speed up the convergence of multi-scale GP models, we propose a self-guided scheme, as shown in Fig. 6. During training, the weighted average of the pseudo ground truth features obtained by the GP in the previous ($(k-1)^{th}$) branch and the features produced by the dense blocks in the current (k^{th}) branch are taken as the input of the current GP. This means that the pseudo ground truth obtained by the GP in the previous scale (or called branch) is used to guide the modeling of the GP in the subsequent scale (branch). The weighted average of these two features can fuse features across different scales for more accurate detection of rain streaks for the patches cropped from a real-world rainy LFI. Obviously, this multi-scale GP structure is conducive to the rapid convergence of the network. The MSGP module of the rain streak detection sub-network MGPDNet models more regular rain streaks with fewer parameters. Therefore, it is feasible to detect rain streaks more effectively than other state-of-the-art methods. Fig. 7 demonstrates that our MSGP with a self-guided scheme converges more quickly and stably than one without the self-guided scheme.

Specifically, the training process of MGPDNet that leverages a multi-scale GP module contains two stages. The first stage is a fully-supervised learning process, in which the ground-truth rain streaks of synthetic rainy LFIs are used to supervise MGPDNet to extract rain streak features. The second stage is an unsupervised process, in which pseudo ground-truth rain streaks for each real-world rainy LFI are produced by the

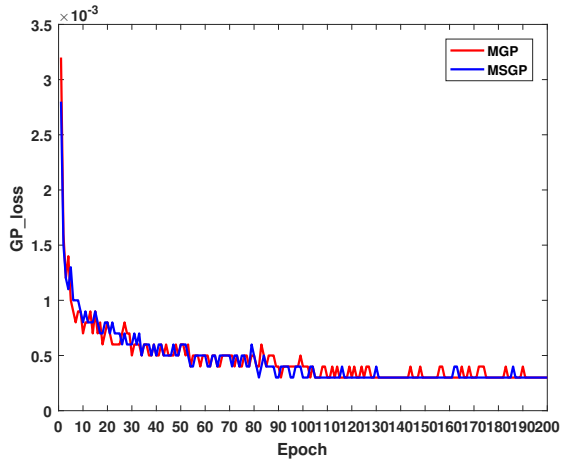


Fig. 7. Convergence process comparison for the MSGP with and without the self-guided scheme. Loss refers to $\mathcal{L}_{gp}$.

pre-trained MGPDNet via the multi-scale GP module. Real-world rainy LFIs taken for training can update the parameters of MGPDNet and improve its generalization to real-world rainy LFIs.

Synthetic rainy LFIs training phase: In this phase, MGPDNet learns network parameters by utilizing synthetic LFIs to extract the rain streaks. L_1 (mean absolute error, MAE) loss and L_2 (mean square error, MSE) loss are common losses for training networks. However, the MSE loss is sensitive to abnormal points, which is not conducive to network training. MAE loss is less sensitive to abnormal points, but it is not differentiable at the target point. Smooth L_1 loss [71] perfectly avoids the defects of MSE and MAE, since its gradient value remains to be 1 even though the absolute error is larger than 1, which is defined as:

$$\mathrm{Smooth}_{\mathrm{L}_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} . \tag{9}$$

In the early stage of training, smooth L_1 loss uses L_1 loss to stabilize the gradient and converge quickly. In the later stage of training, smooth L_1 loss adopts L_2 loss to make the network gradually converge to the optimal solution.

The *smooth L_1 loss* and perceptual loss are both adopted in our supervised loss function.

$$\mathcal{L}_s = \mathrm{Smooth}_{\mathrm{L}_1}(R'_s - R^{gt}_s) + \lambda_p \|\mathrm{VGG}(R'_s) - \mathrm{VGG}(R^{gt}_s)\|_2^2, \tag{10}$$

where λ_p is a constant weighting parameter. R'_s is the predicted rain streaks for a synthetic rainy LFI, and R^{gt}_s is the ground truth. $\mathrm{VGG}(\cdot)$ represents the pre-trained VGG-16 [72].

Besides minimizing the above loss function, all the intermediate feature vectors $f^{i,k}_s$ for the central sub-view of the i^{th} 5D LFI patch cropped from the synthetic rainy LFI in the k^{th} dense block branch are stored in the matrix $F^k_s = \{f^{i,k}_s\}_{i=1}^{N_l}$ during the training process.

Real-world rainy LFIs training phase: After the synthetic rainy LFI training phase, our collected real-world rainy LFIs are used to update the network weights of MGPDNet, which can improve the generalization ability of MGPDNet for rain streak detection on real-world LFIs.

Since it is hard to utilize all synthetic data features to model the pseudo ground truth for a real sample [70], only N_n nearest features from the synthetic data are chosen to form $F^k_{s,n}$. However, it is still non-trivial to accurately detect rain streaks, and utilizing synthetic data to approximate real data may predict false pseudo ground truth. Hence, we minimize the variance $\Sigma^{j,k}_{r,n}$ calculated by $f^{j,k}_r$ and $F^k_{s,n}$, and maximize the variance $\Sigma^{j,k}_{r,f}$ calculated by $f^{j,k}_r$ and N_f farthest synthetic features $F^k_{s,f}$, in order to ensure that $F^k_{s,f}$ are dissimilar to the unlabeled $f^{j,k}_r$ and it does not affect the prediction of GP [5]. Specifically, according to Eq. 8, $\Sigma^{j,k}_{r,n}$ can be written as:

$$\begin{aligned}\Sigma^{j,k}_{r,n} =& (\mathrm{K}(f^{j,k}_r, f^{j,k}_r) + \sigma^2_\epsilon) - \mathrm{K}(f^{j,k}_r, F^k_{s,n})[\mathrm{K}(F^k_{s,n}, F^k_{s,n}) \\ &+ \sigma^2_\epsilon I_n]^{-1}\mathrm{K}(F^k_{s,n}, f^{j,k}_r).\end{aligned} \tag{11}$$

The definition of $\Sigma^{i,k}_{r,f}$ is analogous to $\Sigma^{i,k}_{r,n}$. Similarly, the pseudo ground truth for the real data feature $f^{j,k}_r$ is:

$$f^{j,k}_{r,pseudo} = \mu^{j,k}_r = \mathrm{K}(F^k_{s,n}, f^{j,k}_r)^T(\mathrm{K}(F^k_{s,n}, F^k_{s,n}) + \sigma^2_\epsilon)^{-1}F^k_{s,n} \tag{12}$$

which is used to guide the feature extraction in each 4D dense block branch.

The loss function during the unsupervised training of our MGPDNet is defined as: is:

$$\mathcal{L}_r = \frac{1}{N_u N_b}\sum_{j=1}^{N_u}\sum_{k=1}^{N_b}\mathcal{L}^{j,k}_{gp}, \tag{13}$$

where N_b is the number of 4D dense block branches, and N_u refers to the total number of patches cropped from the real-world rainy LFIs of our RLMB dataset. GPs in our MSGP convert features into vectors for computation convenience, which may result in a lack of global supervision. To ensure the matching of global features, we also introduce perceptual loss to the original GP loss. $\mathcal{L}^{j,k}_{gp}$ is defined as:

$$\begin{aligned}\mathcal{L}^{j,k}_{gp} =& \lambda_{GP}(\|f^{j,k}_r - f^{j,k}_{r,pseudo}\|_2 + \log\Sigma^{j,k}_{r,n} + \log(1 - \Sigma^{j,k}_{r,f})) \\ &+ \lambda_{p,real}(\|\mathrm{VGG}(f^{j,k}_r) - \mathrm{VGG}(f^{j,k}_{r,pseudo})\|_2^2),\end{aligned} \tag{14}$$

where j and k are used to indicate the j^{th} real-world LFI patch and its features $f^{j,k}_r$ extracted from the k^{th} dense block branch, respectively.

Thus, the overall loss function used to train our MGPDNet is defined as:

$$\mathcal{L}_{rain} = \mathcal{L}_s + \mathcal{L}_r. \tag{15}$$

B. Depth Estimation

Due to the occlusion of rain streaks with various shapes and densities, it is difficult to obtain depth maps for the input rainy LFI. In order to alleviate this problem, we subtract the predicted rain streaks from the rainy LFI, and then feed the resulting sub-views to DERNet to estimate accurate depth maps. The depth estimation function for synthetic/real-world rainy LFIs is defined as:

$$D' = \mathcal{F}_{\mathrm{DERNet}}(I - R'), \tag{16}$$

where D' denotes the estimated depth maps, and R' is the predicted rain streaks. $\mathcal{F}_{\mathrm{DERNet}}(\cdot)$ represents the convolution operation of DERNet.

DERNet is trained on the synthetic LFI with ground truth depth maps coming from the synthetic LFI dataset of [1] and a part of the real-world LFI dataset [59] with estimated depth maps produced by [63]. Once obtaining the depth maps, we apply Eq. 2 to convert the estimated depth maps to fog maps, A', in order to assist the subsequent RNNAT to recover rain-free LFIs with consideration of the fog/mist effect.

C. Rain-free Image Recovery

The generator of our RNNAT adopts a recurrent neural network structure to remove rain streaks. It first uses 4D convolution to extract features of all sub-views of an LFI. Then, the Gate Recurrent Unit (GRU) retains and selectively transfers features to the subsequent DSTB shown in Fig. 8. A DSTB is composed of several repeated blocks, each consisting of a Swin Transformer Block (STB) [47], [73] and several 4D Convolution layers in a densely connected manner. The DSTB combines the advantages of CNN and Transformer. The self-attention mechanism within STB can expand the receptive field and capture local and global dependence, and the dense connections can concatenate different levels of features for recovering the rain-free LFI.

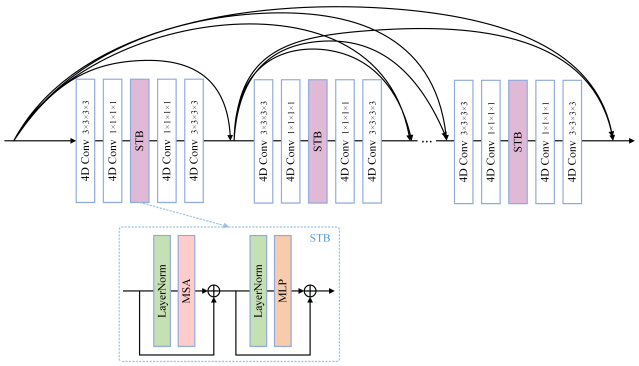


Fig. 8. Architecture of the Dense Swin Transformer Block (DSTB).

The rain removal process of RNNAT is expressed as:

$$Y' = \mathcal{F}_{\mathrm{RNNAT}}([I, R', A']), \tag{17}$$

where Y' denotes the de-rained sub-views produced by RNNAT. A' denotes the fog maps predicted by DERNet. $\mathcal{F}_{RNNAT}(\cdot)$ represents the function of RNNAT.

In order to strengthen the supervision for the network, both the L_1 loss and the perceptual loss are adopted in the generator:

$$\mathcal{L}_g = \|Y' - Y^{gt}\|_1 + \lambda_{p,g}\|\mathrm{VGG}(Y') - \mathrm{VGG}(Y^{gt})\|_2^2, \tag{18}$$

where Y' and Y^{gt} indicate the de-rained sub-views and the ground-truth rain-free sub-views, respectively.

Global-local discriminator is adopted to guide the generator to generate much realistic de-rained sub-views and ensure consistency between the de-rained LFIs and the ground truths for the synthetic LFIs (or pseudo ground truths obtained by subtracting the estimated rain streaks from the input rainy LFIs for real-world rainy scenes). The loss function for the global discriminator is defined as:

$$\mathcal{L}_g^{gan} = -\log(\mathrm{D_g}(Y^{gt})) - \log(1 - \mathrm{D_g}(Y')), \tag{19}$$

where $\mathrm{D_g}(\cdot)$ represents the convolution operation of the global discriminator for rain streak removal.

The local discriminator is introduced to overcome the situation in some scenes, rain streaks can be clearly removed in some regions but not in other local regions, e.g., rain streaks in distant regions. Indeed, our local discriminator, $\mathrm{D_l}(\cdot)$, is an extended PatchGAN [74] working together with the global GAN, much like [75]. Its input is stacked local patches (3D patches) cropped from the same position of all sub-views at a time. The loss function for the local discriminator is defined in the same way as the global discriminator:

$$\mathcal{L}_l^{gan} = \frac{1}{N_p}\sum_{p=1}^{N_p}(-\log(\mathrm{D_l}(Y_p^{gt})) - \log(1 - \mathrm{D_l}(Y_p'))), \tag{20}$$

where Y_p^{gt} and Y_p' refer to the 3D local patches with the spatial resolution $[64, 64]$ cropped from the Y^{gt} and Y', respectively. N_p refers to the number of 3D local patches and is set to 4.

Thus, the loss function for training RNNAT is defined as:

$$\mathcal{L}_{derain} = \mathcal{L}_g + \lambda_{gan}(\mathcal{L}_g^{gan} + \mathcal{L}_l^{gan}), \tag{21}$$

where λ_{gan} is a constant weighting parameter.

Concerning real-world rainy LFIs without ground truths, we subtract the estimated rain streaks from the rainy sub-views as pseudo ground truth for rain-free LFI recovery. Thus, both synthetic LFIs and real-world LFIs can be used to train our RNNAT. In this way, both the performance and generalization of our proposed method can be improved.

The overall loss function for training 4D-MGP-SRRNet is:

$$\mathcal{L}_{total} = \mathcal{L}_{rain} + \mathcal{L}_{derain}. \tag{22}$$

V. Experimental Results and Discussion

A. Network Training Details

Our proposed LFI dataset, called RLMB, includes 400 synthetic rainy LFIs and 200 real-world rainy LFIs. It is divided into a training set of 300 synthetic rainy LFIs and 100 real-world rainy LFIs, and a test set of the remaining images. Each LFI contains 9×9 sub-views. Our proposed 4D-MGP-SRRNet is implemented with PyTorch on a PC with two NVIDIA GeForce RTX 3090 GPUs. Adam optimization is adopted to train our network with a learning rate set to 0.0002 for a total of 200 epochs. We reduce the learning rate by a factor of 0.5 at every 80 epochs. The constant hyper-parameters in Eq. 10, Eq. 14, Eq. 18 and Eq. 21 are set as $\lambda_p = 0.04$, $\lambda_{p,real} = 0.04$, $\lambda_{GP} = 0.015$, $\lambda_{p,g} = 0.04$ and $\lambda_{gan} = 0.01$. The ω in the MSGP is empirically set as 0.5.

In each iteration, the rainy LFI is first fed into MGPDNet to detect rain streaks. The residuals of the rainy LFI and the obtained rain streak map are then fed into DERNet, which directly loads the pre-trained parameters to estimate the depth map. The depth map is converted to a fog map later. Finally, the rainy LFI concatenated with the corresponding rain streaks and fog maps are fed into RNNAT for rain removal. Our MGPDNet and RNNAT are learning together with the parameters of DERNet frozen.

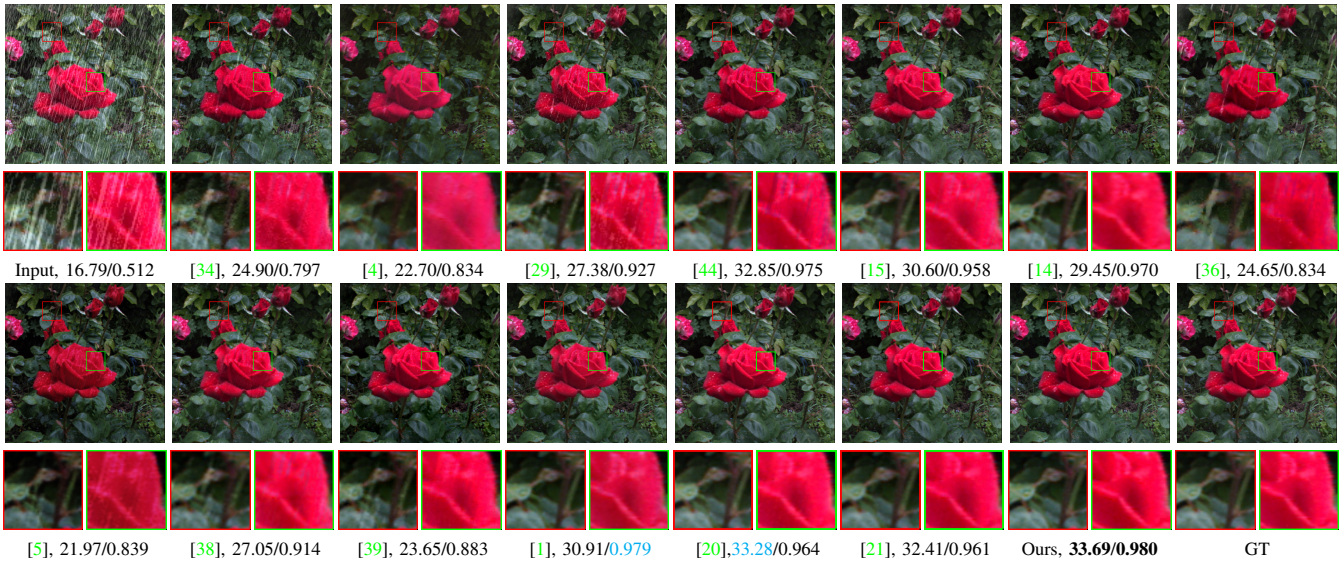


Fig. 9. Comparison of deraining methods on synthetic LFIs. The de-rained center sub-view generated by each method is evaluated on PSNR/SSIM. The best value is highlighted in **bold**, and the second-best value is colored in cyan.

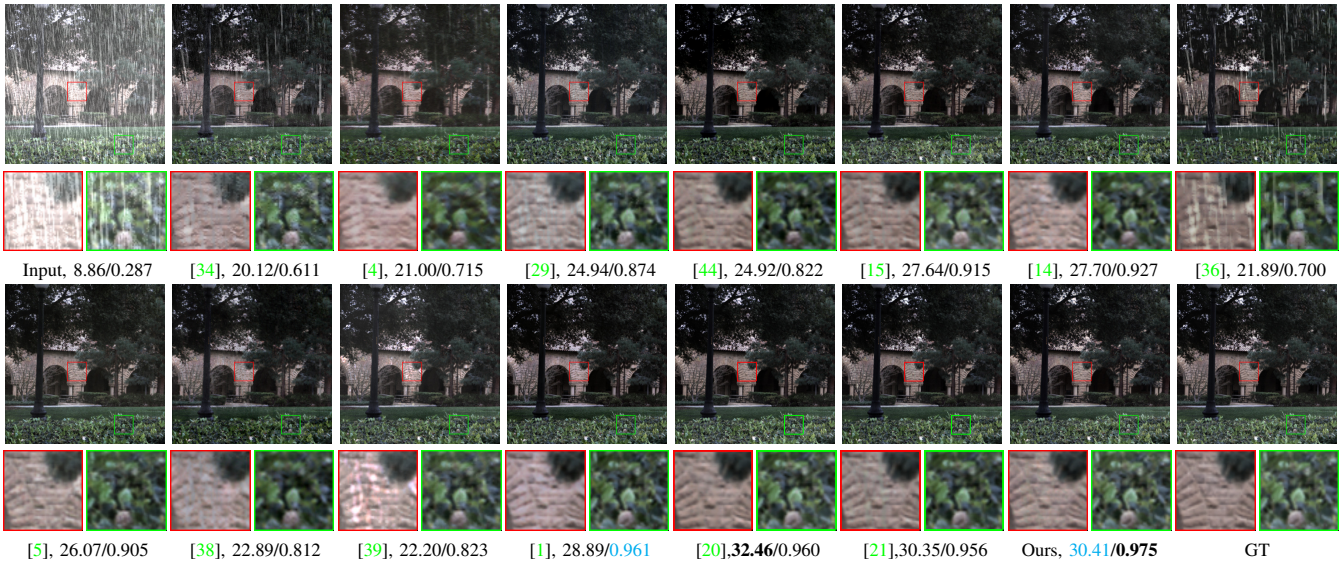


Fig. 10. Comparison of deraining methods on synthetic LFIs. The de-rained center sub-view generated by each method is evaluated on PSNR/SSIM. The best value is highlighted in **bold**, and the second-best value is colored in cyan.

B. Quantitative Evaluation

We conduct quantitative evaluation on the synthetic rainy LFIs. We compared our method with competing methods [1], [4], [5], [14], [15], [29], [34], [36], [38], [39], [44] [20], [21] with their source code re-trained on our generated rainy LFIs.

Two most widely used metrics, peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM), are adopted to conduct our quantitative evaluation. The average PSNR and SSIM values calculated from all 100 test synthetic LFIs are shown in Tab. I, which demonstrates that the de-rained center sub-views recovered by our network with the global-local discriminator are much better (more than 1.30db on PSNR and 0.019 on SSIM) than that produced by other methods [1], [4], [5], [14], [15], [29], [34], [36], [38], [39], [44], and our network with the global-local discriminator outperforms itself with only the global discriminator. It's worth noting that the latest Transformer-based methods [20], [21] perform comparably to our network on the test sub-set of our RLMB dataset. Specifically, on PSNR, the methods [20], [21] obtain the best and the second-best values, while our network with/without local discriminator obtain the third-best and fourth-best values. On SSIM metric, our network with/without local discriminator obtain the best and the third-best values, while the methods [20], [21] obtain the second-best and fourth-best values.

Experimental results for a close shot and a distant shot of real-world-like LFIs are shown in Fig. 9 and 10. Some methods [4], [29], [36], [38] cannot effectively remove rain or haze in most scenes, and only perform better in a small part of scenes. One method [44] removes fog/mist very well. Although it can sometimes obtain the second-best de-rained results, its performance for removing rain is not good enough. Although method [39] obtains satisfactory effects for the rain and fog removal, a lot of blur is introduced to the de-rained

TABLE I
MEAN PSNR/SSIM COMPARISON OF DERAINING METHODS EVALUATED ON THE TESTING SET OF THE SYNTHETIC RAINY LFIS. THE LAST TWO COLUMNS SHOW THE RESULTS OBTAINED BY OUR NETWORK WITH THE GLOBAL DISCRIMINATOR AND GLOBAL-LOCAL DISCRIMINATOR, RESPECTIVELY.

Methods	Wang [34]	Li [4]	Wei [29]	Jiang [44]	Yang [15]	Ren [14]	Jiang [36]	Yasarla [5]	Zamir [38]	Hu [39]	Ding [1]	Xiao [20]	Zamir [21]	Ours-GD	Ours
PSNR	24.59	26.63	24.53	26.40	27.93	28.59	21.38	24.69	24.27	28.18	28.46	**31.52**	30.50	29.54	29.89
SSIM	0.826	0.935	0.821	0.904	0.901	0.938	0.734	0.860	0.848	0.905	0.940	0.956	0.945	0.951	**0.959**

images, such as in Fig. 10. Methods [5], [14], [15], [34] can remove most of the rain streaks. However, when they encounter some very tiny rain streaks, they cannot clearly remove them, as shown in Fig. 9 and 10. The residue of fog or rain streaks and the introduction of blur will obscure the details of the background. LFI rain removal method [1] always performs second best in various challenging scenes. The performances of the latest method [20], [21] is comparable with the former LFI rain removal method [1] even our proposed 4D-MGP-SRRNet on synthetic scenes. Our method can clearly remove various rain streaks and fog, especially in challenging scenes, as shown in Fig. 10. In conclusion, our method is able to remove rain streaks and fog clearly from LFIs captured in various synthetic challenging scenes, whether it is close or long shots.

In addition, we compare our proposed method with several state-of-the-art video rain streak removal methods [26], [50], [51], [54], as shown in Fig. 11 (Fig. 19 of the **Supplemental Material**). All these competing methods take all sub-views of an LFI as input (frame sequence) for rain streak removal. Fig. 11 demonstrates that our proposed method considerably outperforms the video rain streaks removal methods. The results demonstrate that methods [50], [51], [54] have limited effects on rain streak removal from synthetic LFIs, especially for the first three synthetic scenes and the last real-world scene. In addition, none of them is able to remove fog/mist from the exhibited rainy LFIs as our proposed method. The method [26] can clearly remove rain streaks from the synthetic scenes (first three rows), but cannot clearly remove rain streaks from real-world scenes (last two rows) like our proposed network.

C. Qualitative Evaluation

We conduct qualitative evaluation on the carefully collected challenging dataset containing 100 real-world rainy LFIs. Since the process of method [4] for 2D image rain streak removal is similar to that of our rain streak removal from LFIs, and method [14] performs very well on synthetic rainy LFIs, we especially compare our method with these two methods [4], [14] and the semi-supervised methods [1], [5], [29] on several challenging real-world LFIs, as shown in Fig. 12. The results demonstrate that our method with the global-local discriminator performs much better than the compared methods, and achieves state-of-the-art performance. Rain streaks and fog in the rain-free center sub-view restored by our method are almost completely removed. Method [4] removes part of the rain streaks, but introduces a lot of noise. The performance of method [14] is inferior to that of method [29]. It cannot clearly remove the rain streaks and introduced a lot of noise to the de-rained images. Method [5] performs much better and removes most of the rain streaks. Method [1] removes most of the rain streaks in the first-row and third-row scenes, but it introduces blur in the de-rained result in the second-row scenes. Further, it cannot remove large and complex rain streaks with motion blur in the first-row and fourth-row scenes. In Fig. 12, the two latest methods [20], [21] can not clearly remove even obvious rain streaks, which means that their performances on challenging real-world scenes are not as good as that on various synthetic scenes. In conclusion, results produced by these methods are not as good as the de-rained results obtained by our method, which nearly removes all complex rain streaks and without introducing any blur. It is worth noting that the de-rained images produced by our method may be a little darkened in some scenes such as the third and last two scenes of Fig. 12, and some rain streaks may still be there, such as the enlarged patch of the first row of Fig. 12.

Fig. 13 shows rain streaks estimated by our method and existing methods [1], [4], [9], [44], accompanying with their corresponding PSNR/SSIM values. Method [9] performs poorly in the two scenes. Although the rain streaks estimated by method [4] seem acceptable, they miss some obvious rain streaks and cannot detect shapes/boundaries of tiny/large rain streaks. Method [44] incorrectly takes many background details as rain streaks. Method [1] obtains more accurate rain streaks on synthetic scenes, while its effect is not good on real-world scenes (Fig. 14). In contrast, our MGPDNet obtains more accurate rain streaks than the other methods. Accurate rain streak detection provides critical shape and position information of rain streaks for the subsequent rain removal network RNNAT to restore clean de-rained LFI well.

We also compare depth maps obtained by our method and the deep learning-based methods [1], [4], [76], as shown in Fig. 15. Our method is able to estimate satisfactory depth maps for rainy LFIs for the deraining task, and its performance is only a little worse than the state-of-the-art LFI depth estimation method [76] on the synthetic rainy LFI (the first row of Fig. 15). In conclusion, our DERNet can obtain satisfactory depth maps for all sub-views of a rainy LFI.

More experimental results for real-world scenes can be found at our project page: https://github.com/YT3DVision/4D-MGP-SRRNet.

D. Ablation Study

MGPDNet and RNNAT: We conduct ablation experiments to evaluate the effectiveness of MGPDNet and RNNAT in our network. Tab. II shows the ablation study of MGPDNet with/without non-local and MSGP, and RNNAT with/without depth maps estimated by DERNet for LFI deraining. In

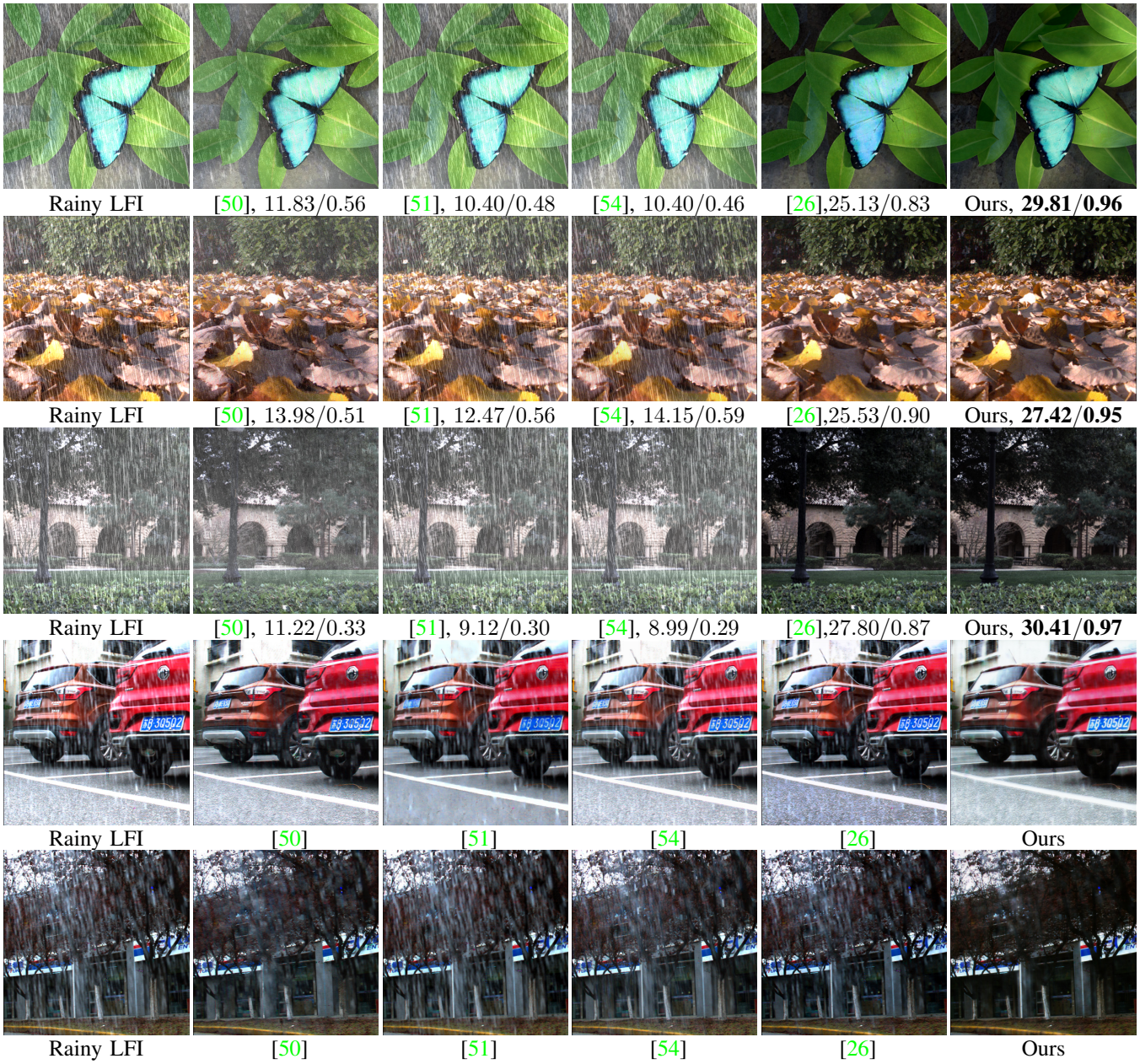


Fig. 11. Comparsion of our method with state-of-the-art video rain streaks removal methods [50], [51], [54] [26]. All these four methods take all sub-views of an LFI as input (frame sequence) for rain streak removal. For rows from top to bottom, the top three rows show three synthetic rainy scenes/LFIs, and the last two rows exhibit two real-world rainy scenes/LFIs. For columns from left to right, rainy LFI and de-rained images (central-view) obtained by the methods [50], [51], [54] [26] and our method, respectively, are exhibited.

addition, we evaluate the performance of our network while replacing the basic block DSTB of RNNAT with DenseBlock.

Fig. 16 shows the estimated rain streaks and de-rained center-view produced by our network with or without MSGP, with simple MGP, respectively. It demonstrates the effectiveness/role of MSGP in 4D-MGP-SRRNet.

2D/3D/4D Convolutions: We conduct an experiment to evaluate the deraining effect of our network using 2D/3D/4D convolutions. The quantitative analysis is shown in Tab. III and Fig. 17. Fig. 17 shows that our 4D-MGP-SRRNet using 2D convolutions incorrectly takes a lot of background details as rain streaks and cannot clearly remove the rain streaks. Our 4D-MGP-SRRNet using 3D convolutions can only extract part of the rain streaks and the rain removal effect is also inferior to our 4D-MGP-SRRNet adopting 4D convolutions.

TABLE II
ABLATION STUDY ON THE BASIC MODULES OF OUR PROPOSED NETWORK. DB MEANS DENSE BLOCK. RB MEANS RESIDUAL BLOCK.

MGPDNet		RNNAT			Metric	
Non-local	MSGP	Depth	DSTB	DB	PSNR	SSIM
√	√	√		√	27.90	0.924
√	√		√		28.32	0.927
√		√	√		29.12	0.932
	√	√	√		29.35	0.950
√	√	√	√		**29.89**	**0.959**

(a) Input (b) Li [4] (c) Ren [14] (d) Wei [29] (e) Yasarla [5] (h) Ding [1] (f) Xiao [20] (g) Zamir [21] (i) Ours-GD (j) Ours

Fig. 12. Comparison of the state-of-the-art deraining methods and our method on the challenging real-world LFIs coming from RLMB. The last two columns show the de-rained sub-views obtained by our network with the global discriminator and global-local discriminator, respectively.

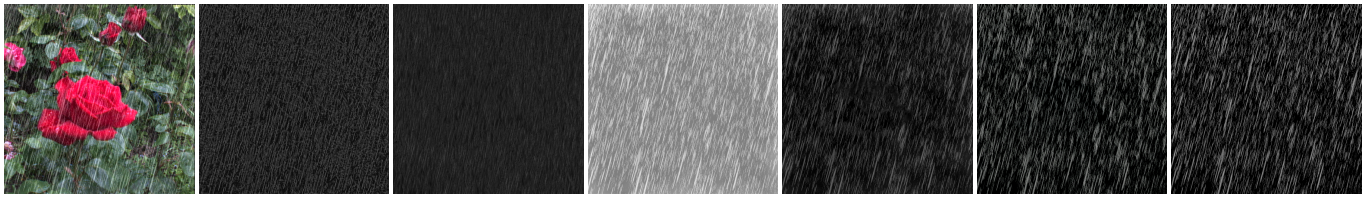
TABLE III
COMPARISON OF OUR NETWORK DERAINING WITH 2D/3D/4D CONVOLUTIONS TESTED ON SYNTHETIC LFIS COMING FROM RLMB.

	Net-2D conv	Net-3D conv	Net-4D conv
PSNR	27.82	29.06	**29.89**
SSIM	0.884	0.908	**0.959**

Tab. IV reports the average tested time for each LFI coming from the test subset of our RLMB dataset, model parameters and GFLOPs of the competing methods and our network. From the second row, it can be seen that the amount of model parameters of our network is smaller than the state-of-the-art Transformer-based single image rain streak removal methods[20,21].

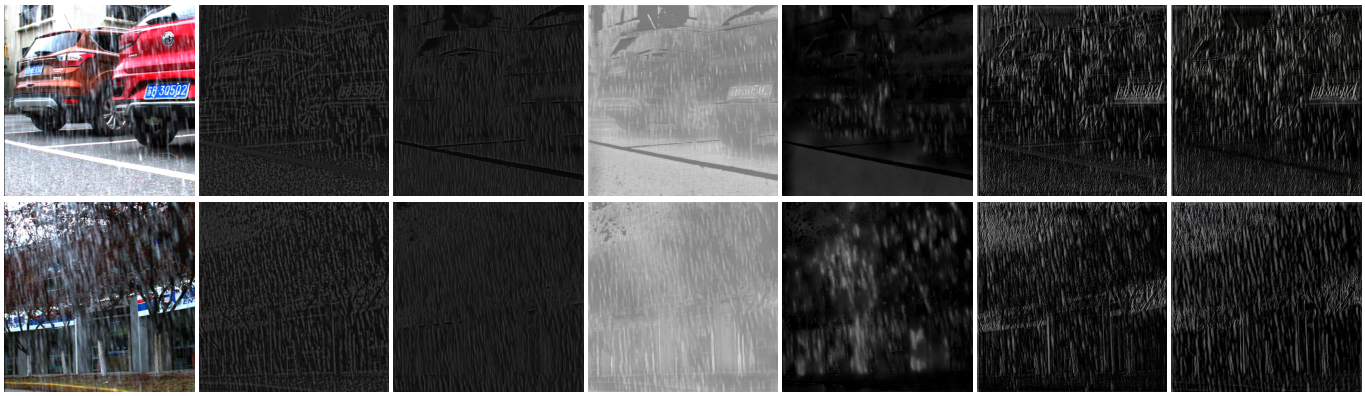
VI. CONCLUSION

In this paper, we have proposed a progressive network called 4D-MGP-SRRNet for detecting and removing rain streaks from LFIs. By leveraging 4D convolution, our network can make full use of all sub-views of LFIs to exploit abundant textural and structural information embedding in LFIs. The MSGP module is proposed for accurate semi-supervised learning-based rain streak detection, which improves the generalization and performance of our network for real-world LFIs. More accurate depth maps are predicted from the results of rain streaks subtracted from rainy sub-views for



Input [9], 19.78/0.689 [4], 15.93/0.130 [44], 6.50/0.350 [1], 22.03/0.789 Ours, **23.68/0.847** GT

Fig. 13. Comparisons of rain streaks detected on the central sub-view of synthetic rainy LFIs by different methods.

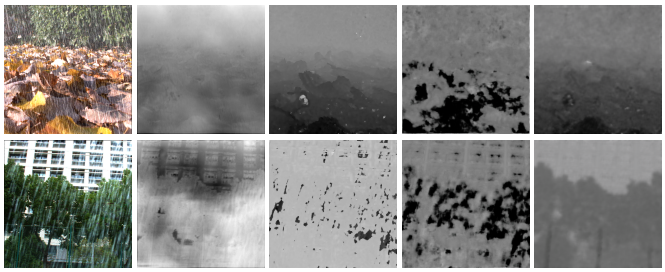


Input Li [9] Li [4] Wang [44] Ding [1] Ours-GD Ours

Fig. 14. Comparison of real-world rain streaks estimated by the state-of-the-art methods [1], [4], [9], [34] and our method. The last two columns show the rain streaks obtained by our method with only global discriminator and global-local discriminator, respectively.

TABLE IV
TIME-COMPLEXITY, PARAMS-COMPLEXITY AND GFLOPS FOR DERAINING METHODS EVALUATED ON THE TEST SET OF SYNTHETIC RAINY LFIS.

Methods	Wang [34]	Li [4]	Wei [29]	Jiang [44]	Yang [15]	Ren [14]	Jiang [36]	Yasarla [5]	Zamir [38]	Hu [39]	Ding [1]	Xiao [20]	Zamir [21]	Ours
Ave.inf.time (s)	1.12	0.35	1.54	0.30	**0.12**	0.38	2.16	0.28	0.30	0.18	0.36	0.85	0.39	0.48
Params(M)	2.10	40.60	**0.07**	0.98	4.17	0.41	0.28	2.62	3.64	4.03	13.24	16.39	26.10	14.91
GFLOPs	9.06	50.08	**1.89**	10.07	68.18	24.56	151.36	5.32	426.8	4.93	35.97	14.48	35.29	47.32



Input Li [4] Tsai [76] Ding [1] Ours

Fig. 15. Comparison of depth maps estimated by our method and the state-of-the-art methods [1], [4], [76] on a synthetic rainy LFI (first row) and a real-world rainy LFI (second row).

fog estimation, and de-rained sub-views are produced by the powerful RNNAT. We also proposed a new rainy LFI dataset RLMB, which consists of both synthetic and real-world rainy LFIs. Extensive experiments on both synthetic and real-world rainy LFIs demonstrate that our proposed method has great superiority over other state-of-the-art methods. However, since it is difficult to collect abundant real-world rainy LFIs, our proposed 4D-MGP-SRRNet is learned on a relatively small size real-world LFI dataset. In addition, the computation cost for rain removal from LFIs is higher than that from regular images.

REFERENCES

[1] Y. Ding, M. Li, T. Yan, F. Zhang, Y. Liu, and R. W. Lau, "Rain streak removal from light field images," *IEEE TCSVT*, 2021.

[2] K. Zhang, W. Luo, Y. Yu, W. Ren, F. Zhao, C. Li, L. Ma, W. Liu, and H. Li, "Beyond monocular deraining: Parallel stereo deraining network via semantic prior," *IJCV*, pp. 1–16, 2022.

[3] Y. Li, Y. Monno, and M. Okutomi, "Dual-pixel raindrop removal," *arXiv:2210.13321*, 2022.

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

[5] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Syn2real transfer learning for image deraining using gaussian processes," in *Proc. CVPR*, 2020.

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

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

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

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

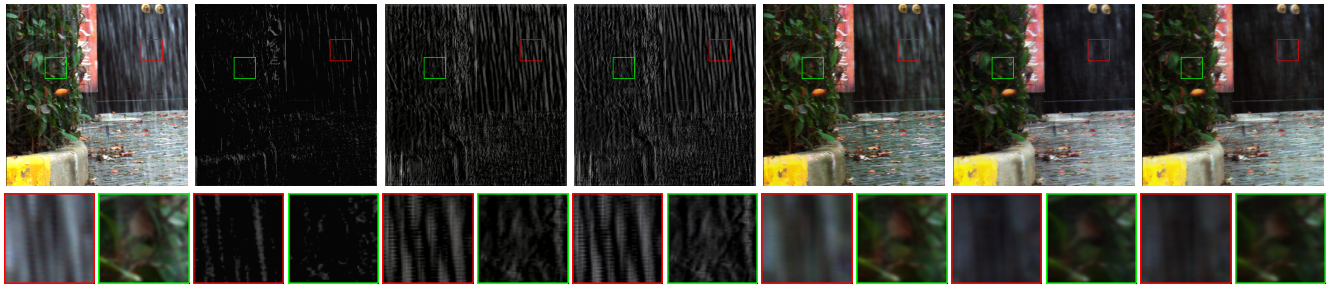
[10] H. Zhang and V. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. CVPR*, 2018.

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

[12] 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.

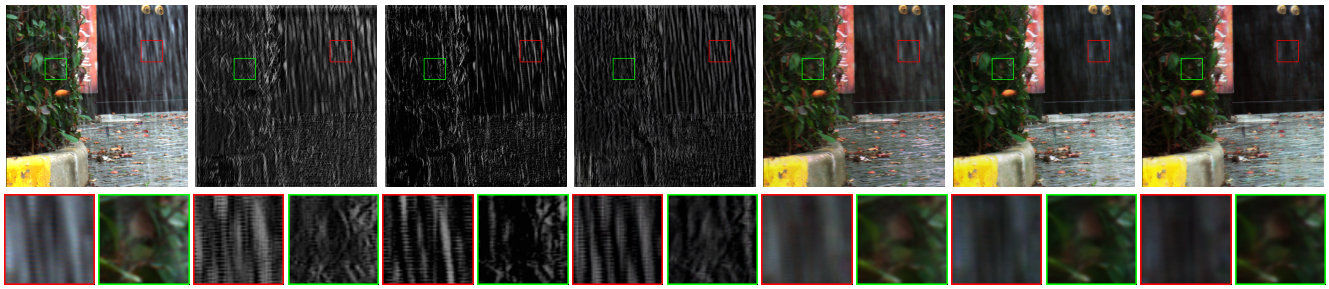
[13] Y. Que, S. Li, and H. J. Lee, "Attentive composite residual network for robust rain removal from single images," *IEEE TMM*, 2020.

[14] 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.



(a) Input (b) Rain w/o MSGP (c) Rain w/ MGP (d) Rain w/ MSGP (e) Output w/o MSGP (e) Output w/ MGP (f) Output w/ MSGP

Fig. 16. Ablation Study of our 4D-MGP-SRRNet on MSGP tested on a real-world rainy LFI.



(a) Input (b) Rain w/ 2D (c) Rain w/ 3D (d) Rain w/ 4D (e) Output w/ 2D (e) Output w/ 3D (f) Output w/ 4D

Fig. 17. Ablation Study of our 4D-MGP-SRRNet using 2D/3D/4D convolution on a real-world rainy LFI.

[15] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE TPAMI*, vol. 42, no. 6, pp. 1377–1393, 2020.

[16] N. Ahn, S. Y. Jo, and S.-J. Kang, "Eagnet: Elementwise attentive gating network-based single image de-raining with rain simplification," *IEEE TCSVT*, 2021.

[17] X. Fu, Q. Qi, Z.-J. Zha, X. Ding, F. Wu, and J. Paisley, "Successive graph convolutional network for image de-raining," *IJCV*, vol. 129, no. 5, pp. 1691–1711, 2021.

[18] H. Huang, A. Yu, Z. Chai, R. He, and T. Tan, "Selective wavelet attention learning for single image deraining," *IJCV*, pp. 1–19, 2021.

[19] Y. Wei, Z. Zhang, Y. Wang, M. Xu, Y. Yang, S. Yan, and M. Wang, "Deraincyclegan: Rain attentive cyclegan for single image deraining and rainmaking," *IEEE TIP*, 2021.

[20] J. Xiao, X. Fu, A. Liu, F. Wu, and Z.-J. Zha, "Image de-raining transformer," *IEEE TPAMI*, 2022.

[21] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. CVPR*, pp. 5728–5739, 2022.

[22] Y. Gou, B. Li, Z. Liu, S. Yang, and X. Peng, "Clearer: Multi-scale neural architecture search for image restoration," *NeurIPS*, vol. 33, pp. 17129–17140, 2020.

[23] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general u-shaped transformer for image restoration," in *Proc. CVPR*, pp. 17683–17693, 2022.

[24] J. M. J. Valanarasu, R. Yasarla, and V. Patel, "Transweather: Transformer-based restoration of images degraded by adverse weather conditions," in *Proc. CVPR*, pp. 2353–2363, 2022.

[25] B. Li, X. Liu, P. Hu, Z. Wu, J. Lv, and X. Peng, "All-in-one image restoration for unknown corruption," in *Proc. CVPR*, pp. 17452–17462, 2022.

[26] K. Zhang, D. Li, W. Luo, W. Ren, and W. Liu, "Enhanced spatio-temporal interaction learning for video deraining: A faster and better framework," *IEEE TPAMI*, 2022.

[27] Y. Wang, C. Ma, and J. Liu, "Removing rain streaks via task transfer learning," *arXiv:2208.13133*, 2022.

[28] S. Li, W. Ren, F. Wang, I. B. Araujo, E. K. Tokuda, R. H. Junior, R. M. Cesar-Jr, Z. Wang, and X. Cao, "A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives," *IJCV*, vol. 129, no. 4, pp. 1301–1322, 2021.

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

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

[31] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV*, 2017.

[32] 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.

[33] 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 Multimedia*, 2018.

[34] 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. CVPR*, 2019.

[35] H. Zhu, X. Peng, J. T. Zhou, S. Yang, V. Chanderasekh, L. Li, and J.-H. Lim, "Singe image rain removal with unpaired information: A differentiable programming perspective," in *Proc. AAAI*, 2019.

[36] 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. CVPR*, 2020.

[37] C.-Y. Lin, Z. Tao, A.-S. Xu, L.-W. Kang, and F. Akhyar, "Sequential dual attention network for rain streak removal in a single image," *IEEE TIP*, vol. 29, pp. 9250–9265, 2020.

[38] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proc. CVPR*, 2021.

[39] X. Hu, L. Zhu, T. Wang, C.-W. Fu, and P.-A. Heng, "Single-image real-time rain removal based on depth-guided non-local features," *IEEE TIP*, vol. 30, pp. 1759–1770, 2021.

[40] Y. Wang, D. Gong, J. Yang, Q. Shi, A. van den Hengel, D. Xie, and B. Zeng, "Deep single image deraining via modeling haze-like effect," *IEEE TMM*, 2020.

[41] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE TCSVT*, 2020.

[42] Y.-T. Wang, X.-L. Zhao, T.-X. Jiang, L.-J. Deng, Y. Chang, and T.-Z. Huang, "Rain streaks removal for single image via kernel-guided convolutional neural network," *IEEE TNNLS*, 2021.

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

[44] K. Jiang, Z. Wang, P. Yi, C. Chen, Z. Han, T. Lu, B. Huang, and J. Jiang, "Decomposition makes better rain removal: An improved attention-guided deraining network," *IEEE TCSVT*, 2020.

[45] H. Huang, A. Yu, and R. He, "Memory oriented transfer learning for semi-supervised image deraining," in *Proc. CVPR*, 2021.

[46] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proc. CVPR*, pp. 12299–12310, 2021.

[47] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and

B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. ICCV*, 2021.
[48] B. Li, Y. Gou, S. Gu, J. Z. Liu, J. T. Zhou, and X. Peng, "You only look yourself: Unsupervised and untrained single image dehazing neural network," *IJCV*, vol. 129, no. 5, pp. 1754–1767, 2021.
[49] R. Li, R. T. Tan, and L.-F. Cheong, "All in one bad weather removal using architectural search," in *Proc. CVPR*, pp. 3175–3185, 2020.
[50] 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. CVPR*, 2018.
[51] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *Proc. CVPR*, 2018.
[52] W. Yang, J. Liu, and J. Feng, "Frame-consistent recurrent video deraining with dual-level flow," in *Proc. CVPR*, 2019.
[53] W. Yang, R. T. Tan, S. Wang, and J. Liu, "Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence," in *Proc. CVPR*, 2020.
[54] M. Li, X. Cao, Q. Zhao, L. Zhang, and D. Meng, "Online rain/snow removal from surveillance videos," *IEEE TIP*, vol. 30, pp. 2029–2044, 2021.
[55] W. Yang, R. T. Tan, S. Wang, A. C. Kot, and J. Liu, "Learning to remove rain in video with self-supervision," *IEEE TPAMI*, 2022.
[56] W. Yan, L. Xu, W. Yang, and R. Tan, "Feature-aligned video raindrop removal with temporal constraints," *IEEE TIP*, vol. 31, pp. 3440–3448, 2022.
[57] C. Tan, J. Chen, and L. Chau, "Edge-preserving rain removal for light field images based on rpca," in *Proc. DSP*, 2017.
[58] 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. CVPR*, pp. 2482–2491, 2018.
[59] "Stanford lytro light field archive," Accessed 26 March 2022. http://lightfield.stanford.edu/lfs.html.
[60] "Blender," Accessed 12 January 2022. https://www.blender.org/.
[61] M. Potmesil and I. Chakravarty, "Modeling motion blur in computer-generated images," *ACM SIGGRAPH*, vol. 17, no. 3, pp. 389–399, 1983.
[62] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE TPAMI*, vol. 33, no. 12, pp. 2341–2353, 2010.
[63] 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. CVPR*, 2015.
[64] S. Zhang, S. Guo, W. Huang, M. R. Scott, and L. Wang, "V4d: 4d convolutional neural networks for video-level representation learning," in *Proc. ICLR*, 2020.
[65] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.
[66] H. Bai, S. Cheng, J. Tang, and J. Pan, "Learning a cascaded non-local residual network for super-resolving blurry images," in *Proc. CVPR*, 2021.
[67] G. Huang, Z. Liu, L. Van Der Maaten, and K. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017.
[68] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*, pp. 63–71, 2003.
[69] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT Press, 2006.
[70] R. Murray-Smith and A. Girard, "Gaussian process priors with arma noise models," in *Irish Signals and Systems Conference*, 2001.
[71] R. Girshick, "Fast r-cnn," in *Proc. ICCV*, 2015.
[72] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.
[73] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proc. ICCV*, 2021.
[74] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. CVPR*, 2017.
[75] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," *arXiv:1803.07422*, 2018.
[76] 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*, 2020.