Depth-attentional Features for Single-image Rain Removal

Xiaowei Hu[1], Chi-Wing Fu[1,*], Lei Zhu[2,1,*], and Pheng-Ann Heng[1,2]
[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2] Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

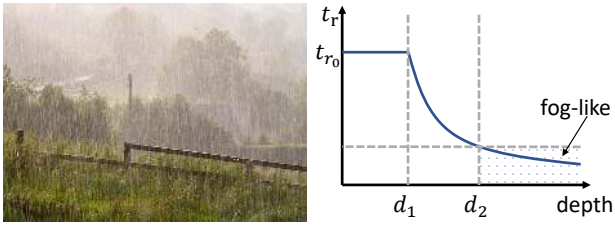
Abstract

Rain is a common weather phenomenon, where object visibility varies with depth from the camera and objects far-away are visually blocked more by fog than by rain streaks. Existing methods and datasets for rain removal, however, ignore these physical properties, thereby limiting the rain removal efficiency on real photos. In this work, we first analyze the visual effects of rain subject to scene depth and formulate a rain imaging model collectively with rain streaks and fog; by then, we prepare a new dataset called RainCityscapes with rain streaks and fog on real outdoor photos. Furthermore, we design an end-to-end deep neural network, where we train it to learn depth-attentional features via a depth-guided attention mechanism, and regress a residual map to produce the rain-free image output. We performed various experiments to visually and quantitatively compare our method with several state-of-the-art methods to demonstrate its superiority over the others.

1. Introduction

Rain is a common weather phenomenon, but its presence could greatly affect the visibility of objects and scene in the captured photos. Hence, it would interfere and degrade the performance of many computer vision and image processing tasks, *e.g.*, object detection [17] and tracking [40] for surveillance [2], autonomous driving [20], and driver assistance [33]. To this end, rain removal has long been a fundamental problem in computer vision research.

Rain removal is, however, a very challenging task, since we have to remove the rain, and at the same time, recover the occluded objects and scene. In nature, the occlusion is caused not only by the rain streaks but also by the fog that comes with the rain. Moreover, the scene visibility *spatially varies* in the image space, since *objects closer to the camera are affected mainly by the rain streaks, while objects far away are affected more heavily by the fog*; see Figure 1(a) for a real photo example. This phenomenon is also depicted in a model by Garg and Nayar [13] (see Figure 1(b)), which describes the intensity of rain streaks and their transformation into the fog as a function of the scene depth.



(a) input real photo (b) rain visibility & depth

Figure 1: (a) An example real photo that demonstrates the scene visibility variation with depth, and the presence of rain streaks and fog; and (b) a plot of rain streak intensity (t_r) against scene depth (d) based on the model in [13].

In the literature on single-image rain removal, existing methods focused on removing the rain streaks by adopting various image priors [5, 19, 22, 23, 31, 32, 41, 52], or by exploiting a deep convolutional neural network (CNN) to learn a mapping between the training images with and without rain streaks [8, 9, 28, 30, 47, 51, 50]. While the state-of-the-art methods can already produce satisfactory results on various synthetic datasets [47, 51, 50], they focus mainly on removing rain streaks and ignore the physical properties of rain; hence, they mostly fail to remove rain and fog altogether. Moreover, existing datasets for rain removal contain only rain streaks, while some of the images are indoor rather than outdoor, thereby also limiting the development of rain removal methods for real photos; see Figure 2.

In this paper, we first analyze the physical properties of rain and formulate a rain imaging process with rain streaks and fog. By then, we prepare a new dataset for rain removal with scene depth information, and further design a new neural network for single-image rain removal by learning depth-attentional features in a depth-guided manner. In summary, this work has the following contributions:

- First, we design an end-to-end neural network, where we formulate a depth-guided attention mechanism to

*Co-corresponding authors

(a) input real photo[1] (b) DID-MDN [50] (c) RESCAN [30] (d) RESCAN [30]+DCPDN [49] (e) our result

Figure 2: Visual comparison of single-image rain removal results on a real photo (a). Results in (b) and (c) are produced by two state-of-the-art rain removal methods, while the result in (d) is produced by further applying a state-of-the-art haze removal method to (b). Comparing (e) with (b) to (d), our method clearly can better remove the rain in the real photo.

learn depth-attentional features and regress a residual map based on the attention weights to remove rain streaks and fog in the input rain image.

- Second, we formulate the rain imaging process based on the visual effects of rain subject to scene depth, implement the formulation to synthesize rain streaks and fog, and prepare a new dataset for rain removal.
- Third, we perform various experiments to evaluate our network and dataset. Results show that our network quantitatively and qualitatively outperforms existing works on both synthetic images and real photos.

2. Related Work

Early methods [1, 19, 22, 32, 41] remove rain streaks in images by designing hand-crafted priors based on low-level image statistics. Barnum *et al*. [1] combined the streak model and rain characteristics to detect and remove rain streaks in frequency space. Since the rain streaks usually have similar and repeated patterns, Chen *et al*. [5] created a low-rank prior based on the rain streak appearance for rain removal. Li *et al*. [31] adopted patch priors based on Gaussian mixture model for the background and rain layers to remove rain streaks. Zhu *et al*. [52] estimated the dominated rain direction and formulated a bi-layer joint optimization to iteratively separate rain streaks from the background.

However, the hand-crafted priors limit the capability to describe and remove rain. Such limitation is overcome by deep learning methods, which automatically learn the features via a convolutional neural network (CNN). Fu *et al*. [8] learned the mapping function between the rain-free and rain layers from the training data, while Yang *et al*. [47] created a multi-task network to jointly detect and remove rain. Later, inspired by the deep residual network for image recognition [16], residual-learning-based networks are developed for rain removal by predicting the residual, i.e., the difference between rain and rain-free images. Fu *et al*. [9] exploited a priori knowledge to formulate a base layer and detail layer from the input image, and then learned the residual from the detail layer through a deep network. Li *et al*. [30] formulated a contextual dilated network with squeeze-and-excitation blocks to iteratively predict the stage-wise residual. Zhang *et al*. [50] developed a residual-aware classifier to determine the rain density and stacked several densely-connected networks to estimate the residual accordingly.

Overall, the state-of-the-art methods focus on images mainly with rain streaks, as limited by the existing datasets. In this work, we not only prepare a new dataset for rain removal based on a realistic rain model with scene depth, rain streaks and fog, but also develop a new depth-attentional feature network to learn the scene depth and take it as guidance to remove rain streaks and fog in the input images.

Other related works. Garg and Nayar [12] developed an image-based rain generation algorithm by considering the scene depth and light sources. Other than images, several methods [10, 38] have been developed to remove rain in videos. Since we focus this work on single-image rain removal, we refer readers to [42] for a detailed survey.

Apart from rain, several recent works started to explore and develop deep learning methods for removing various forms of weather-related artifacts in images, *e.g*., snow [36], raindrops [34, 48], and haze [3, 27, 29, 35, 46, 49]. Since haze is relevant to the fog component in our rain model, we also compare our results with the results produced from the state-of-the-art dehaze networks; see Section 5.

3. Formulation and Dataset

3.1. Rain Model

According to Garg and Nayar [13], the visual intensity of a rain streak depends on the scene depth d from the camera to the underlying scene objects behind the rain. Denoting t_r as the visual intensity of rain streaks and t_{r_0} as the maximum t_r in the model, we have the following cases:

[1] Courtesy of photographer Mac99 (Getty Images No. 182715405)

(a) rain-free image I(x) (b) rain layer R(x) (c) fog layer A(x) (d) rain image O(x)

Figure 3: Two sets of example images in our dataset "RainCityscapes."

- For scene objects close to the camera ($d \leq d_1$), its associated image region will be dominated by rain streaks with little fog, i.e., $t_r = t_{r_0}$, where $d_1 = 2fa$, f is focal length, and a is raindrop radius; see [13] for details;
- For scene objects far from the camera ($d \geq d_2 \gg d_1$), its associated image region will be dominated by fog with little rain streaks, i.e., t_r tends to zero as d increases.
- As d increases from d_1 to d_2, the intensity of rain streaks will drop while the intensity of fog will rise; see the plot in Figure 1(b), which summarizes the variations.

3.2. Our Formulation of Rain Images

In this work, we consider a rain image as a composition of a rain-free image, a rain layer, and a fog layer, and formulate the observed rain image $O(x)$ at pixel x as:

$$O(x) = I(x)\,(\,1 - R(x) - A(x)\,) + R(x) + A_0 A(x)\,, \quad (1)$$

where $I(x)$ denotes the rain-free image with the clear scene radiance; $R(x) \in [0, 1]$ denotes the rain layer; A_0 is the atmospheric light, which is assumed to be a global constant following [37]; and $A(x) \in [0, 1]$ represents the fog layer; see Figure 3 for examples of $I(x)$, $R(x)$, $A(x)$, and $O(x)$.

For both $R(x)$ and $A(x)$, a large value indicates a high intensity of rain streak or fog, while a zero value means no rain streak or no fog. Hence, $(\,1 - R(x) - A(x)\,)$ is multiplied with $I(x)$ in the first term of the formulation, since the scene visibility reduces with $R(x)+A(x)$. Note also that we follow [37] and do not use A_0 on the $I(x)$ term, since $I(x)$ is already affected by the atmospheric light. Furthermore, we model $R(x)$ and $A(x)$ as follows:

- For the rain layer $R(x)$, we model it in two parts:

$$R(x) = R_{\text{pattern}}(x) * t_r(x)\,, \quad (2)$$

where $R_{\text{pattern}}(x) \in [0, 1]$ is an intensity image of uniformly-distributed rain streaks in the image space; $t_r(x)$ is the rain streak intensity map, which depends on the scene depth $d(x)$ according to the rain model described in Section 3.1; and $*$ represents a pixel-wise multiplication. In detail, $t_r(x)$ is modeled as

$$t_r(x) = e^{-\alpha \max(d_1,\, d(x))}\,. \quad (3)$$

where α is an attenuation coefficient that controls the rain streak intensity. Moreover, t_{r_0} (which is the maximum rain streak intensity) equals to $e^{-\alpha d_1}$, whereas $t_r(x)$ starts with t_{r_0} and gradually drops to zero after $d(x)$ goes beyond d_1; see again Figure 1(b).

- Unlike rain, the visual intensity of fog increases exponentially with the scene depth, according to the standard optical model [25] that simulates the image degradation process. Hence, we model the fog layer $A(x)$ as

$$A(x) = 1 - e^{-\beta d(x)}\,, \quad (4)$$

where β is an attenuation coefficient that controls the thickness of fog, and a larger β indicates a thicker fog, and vice versa. Lastly, note also that we assume a homogeneous atmosphere in the scene, so both the rain and fog transmissions depend on d, as described by the exponential formulations in Eqs. (3) and (4).

3.3. Our RainCityscapes Dataset

To capture a pair of real photos with and without rain for training is almost impossible, since the scene objects may move and the environment lighting and camera exposure may change. Hence, existing datasets [47, 51, 50] for rain removal were typically prepared by synthetically adding a 2D layer of rain streaks on photos, where recent deep networks are simply trained on them to remove rain. Clearly, the physical rain model is ignored, so existing methods tend to fail for real photos; see Figure 2 for examples.

In this work, we revisit the problem of single-image rain removal, where we first prepare a new dataset with rain and fog based on the formulation in Section 3.2. To do so, we adopt the photos in the Cityscapes dataset [6] as our rain-free images, and use the camera parameters and scene depth

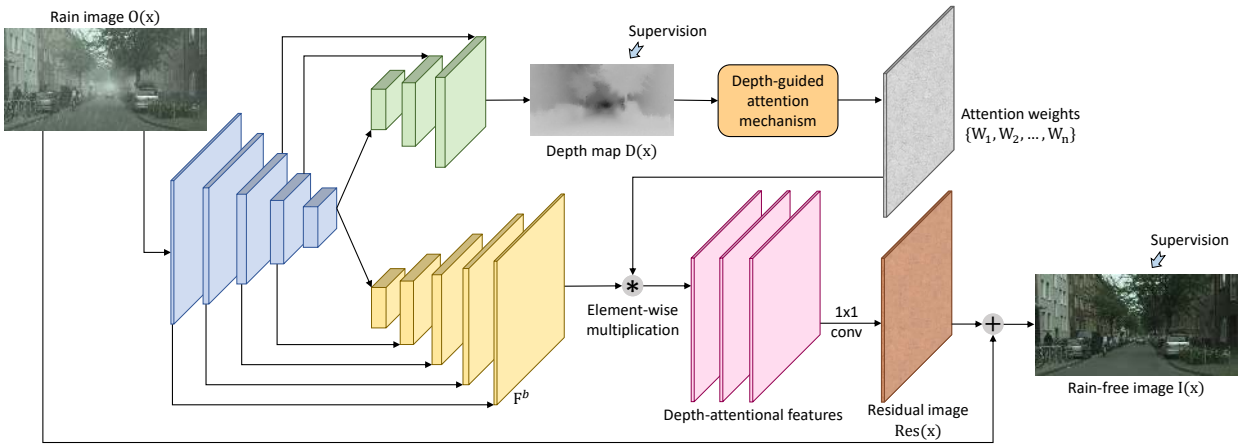


Figure 4: The schematic illustration of DAF-Net: (i) a convolutional neural network (in blue) for extracting multi-resolution features from the input; (ii) a decoder branch (in green) for predicting the depth map; (iii) the depth-guided attention mechanism (in orange) for learning the attention weights; (iv) another decoder branch (in yellow) for producing the depth-attentional features together with the attention weights; and (v) lastly, we use a set of group convolutions [45] on the depth-attentional features (in pink) to predict a residual map, and add it to the input to produce the output rain-free image. In the figure, we depict feature maps as blocks, where thicker blocks have more feature channels.

information in the dataset to synthesize rain and fog on the photos. We name our dataset "RainCityscapes" after the Cityscapes dataset; compared with previous datasets, our dataset are all outdoor photos, each with a depth map, and the rain images exhibit different degrees of rain and fog.

To prepare the dataset, we first picked 262 training images and 33 testing images from the training and validation sets of Cityscape as our rain-free images, where the weather is overcast without obvious shadow and the depth map is plausible. Then, we used a depth denoising method [37] to refine the depth maps of the picked images, and generated the rain streak intensity map $t_r(x)$ and fog layer $A(x)$ from each depth map using Eqs. (3) & (4). Here, we used three sets of parameters $\{(0.02, 0.01, 0.005), (0.01, 0.005, 0.01), (0.03, 0.015, 0.002)\}$ for attenuation coefficients α and β, and raindrop radius a, to simulate different degrees of rain and fog. Next, we used a guided filtering method [14] with the rain-free image as the guidance to smooth $t_r(x)$ and $A(x)$, employed the rain patches in [31] to synthesize the rain streak patterns R_{pattern} in Eq. (2), and then generated the observed rain images using Eq. (1). Altogether, our RainCityscapes dataset has $9,432$ training images and $1,188$ testing images; see Figure 3 for some examples.

3.4. Limitations on the rain imaging process

The rain imaging process assumes that the rain and fog layers are uniformly-distributed and independent. However, in the real world, the visual effects of rain and fog are correlated with the rain intensity [44]; the rain appearance depends on the camera parameters [11] (e.g., exposure time); and the intensity changes are more complex in a volume of rain [13]. Also, the camera ego-motion could disperse the rain distribution and cause extra motion blur in the image space. Though our rain model is an approximation and lacks an optic model, the synthesized images indeed help improve the results compared to previous works and data, which ignore the rain property we explored; see Section 5 for the qualitative and quantitative comparison results.

4. Methodology

Figure 4 shows the architecture of our deep network with the depth-attentional features (named "DAF-Net") for single-image rain removal. It is an end-to-end network that takes a rain image as input, predicts a depth map, and then produces a rain-free image as the output.

In summary, the network first leverages a convolutional neural network (CNN) to extract low-level details and high-level semantics from the input image and produce feature maps in varying resolutions. Then it employs two decoder branches, each progressively upsampling a feature map and combining it with the CNN feature map in the same resolution to produce a new feature map; see the polygonal lines among the feature maps in the blue, green, and yellow blocks in Figure 4. In the top decoder branch, we further regress a depth map (see Section 4.1) and learn a set of attention weights via the depth-guided attention mechanism. In the bottom decode branch, we first generate the final (highest resolution) feature map and then combine it with the attention weights from the top branch to produce the depth-attentional features (see Section 4.2). Lastly, we apply a set of group convolutions [45] on these features, predict a residual map, and add it to the input image to produce the output rain-free image.

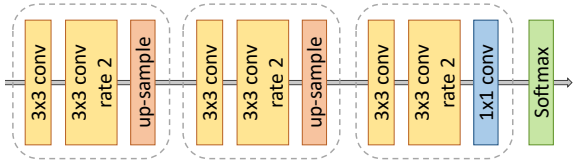


Figure 5: The detailed structure in the depth-guided attention mechanism: "$N \times N$ conv" denotes a convolution operation with a kernel size of $N \times N$, while "rate 2" denotes a dilated convolution [4] with a dilation rate of two.

4.1. Regress the Depth Map

In the top decoder branch shown in Figure 4, when the width of the upsampled feature map reaches a quarter of the input, we add a supervision signal and regress a depth map about the input image. Note that a depth map of lower resolution is sufficient to serve as the guidance for learning the attention weights, so we regress a quarter-width depth map to reduce the computation and memory overhead.

Typically, scene depth has huge spatial ranges. Hence, rather than directly regressing raw depth values in the network, we regress the logarithm of the depth values by transforming the depth values in the supervision signal (input depth map in the training dataset) as follows:

$$D(x) \;=\; e^{-0.1 d(x)} \,, \tag{5}$$

where $d(x)$ denotes the scene depth at pixel x (following our formulation in Section 3) and $D(x)$ is the supervision signal in the network. Therefore, the regressed depth map in the network is, in fact, a map of logarithmic depth values. Note also that this strategy matches our formulation of scene depth, which is in fact stored and processed in logarithmic scale; see Eqs. (3) and (4) in Section 3.

4.2. Depth-attentional Features

As described earlier, the visual effect of rain streaks and fog in an observed rain image depends on the scene depth; hence, the rain streaks and fog, as well as the rain removal process, are depth-dependent. Therefore, we first regress a depth map in our network, and take it as a guidance to learn a set of attention weights. Then, we can use these weights to integrate the feature maps from the bottom decoder branch in our network to form a residual map of rain streaks and fog. Further, we add the residual map to the rain image to produce the output rain-free image; see Figure 4.

To effectively construct the residual map from the convolutional feature maps, we formulate the *depth-guided attention mechanism* to learn the attention weights from the regressed depth map $D(x)$. Figure 5 shows the detailed structure of the mechanism, where we first adopt three convolutional blocks to process $D(x)$ with a ReLU non-linear operation [26] after each 3×3 convolution layer.

The output of the last convolutional block is a set of unnormalized attention weights $\{A_1, A_2, ..., A_n\}$. In general, each weight corresponds to a certain type of rain streaks and fog. Then, we apply the Softmax function (Eq. (7)) to normalize the weights, and generate the attention weights $\{W_1, W_2, ..., W_n\}$, each associated with a certain group of rain streaks and fog:

$$\{A_1, A_2, ..., A_n\} \;=\; \mathcal{B}_{conv}\,(\,D\,;\,\theta\,)\,, \text{ and} \tag{6}$$

$$w_{x,c} \;=\; \frac{e^{a_{x,c}}}{\sum_{c=1}^{n} e^{a_{x,c}}}\,, \tag{7}$$

where $\mathcal{B}_{conv}$ denotes the three convolutional blocks shown in Figure 5; it takes $D(x)$ as input and learns a set of parameters θ to produce the unnormalized attention weights $\{A_1, A_2, ..., A_n\}$; $a_{x,c} \in A_c$ denotes the weight at channel c of pixel x in A_c ($c = \{1, 2, ..., n\}$); and $w_{x,c} \in W_c$ denotes the resulting attention weight, which is obtained by normalizing the $\{a_{x,c}\}_{c=1}^{n}$ via a softmax; see Eq (7).

The feature map F^b of the highest resolution produced from the bottom decoder branch (see the yellow blocks in Figure 4) has 256 feature channels. Next, we divide it into n submaps over the 256 channels, so each submap F_i^b ($i = 1, 2, ..., n$) has $\frac{256}{n}$ channels and has the same resolution as the original feature map F^b; in practice, we set n as 64. Then, we multiply W_c with each feature channel of the c-th submap F_c^b in an element-wise manner to produce the depth-attentional features.

Right now, we prepare the depth-attentional features in n separate parts. Hence, we can then perform group convolutions [45] in n groups individually on each part of the depth-attentional features to enhance the expressiveness of the features. By adopting the group convolutions, the features in each group are only responsible for removing a certain kind of rain streaks and fog with a small intra-class variance. Finally, we merge all the features from different groups using a 1×1 convolution to produce the residual map $Res(x)$, to which we add the input rain image $O(x)$ to produce the output rain-free image $I(x)$.

4.3. Training and Testing Strategies

Loss function. We train the network by minimizing the following loss function L over the pixels in the output rain-free image $I(x)$ and the pixels in the depth map $D(x)$:

$$\begin{aligned} L \;=\; & \omega_i \sum_{x \in \mathcal{X}} \sum_{l \in \{\mathcal{R}, \mathcal{G}, \mathcal{B}\}} |\, I(x)_l \;-\; \bar{I}(x)_l \,|^2 \\ & +\; \omega_d \sum_{x \in \mathcal{X}_4} |\, D(x) \;-\; \bar{D}(x) \,|^2 \,, \end{aligned} \tag{8}$$

where ω_i and ω_d are weights; $\mathcal{X}$ and $\mathcal{X}_4$ denote the image domains of the output image and depth map, respectively; $I(x)_l$ and $\bar{I}(x)_l$ denote the predicted and ground truth values, respectively, in the l-th RGB color channels of pixel

(a) input real photo (b) DID-MDN [50] (c) RESCAN [30] (d) RESCAN [30]+DCPDN [49] (e) our results

Figure 6: More visual comparison results on real photos; see also Figure 2. Again, the results in (b) and (c) are produced by two state-of-the-art rain removal methods, while those in (d) are produced by further applying a state-of-the-art haze removal method to (b). Comparing our results (e) with (b) to (d), our method can again better remove the rain in the real photos.

Table 1: User study results. Mean ratings (from 1 (fake) to 10 (real)) given by the participants on the various datasets.

dataset	rating (mean & standard dev.)
real rain photo	8.93 ± 1.66
RainCityscapes (ours)	6.38 ± 2.52
Rain800 [51]	3.69 ± 2.58
DID-MDN [50]	2.90 ± 2.39
Rain100H [47]	1.46 ± 1.18

x in $\mathcal{X}$; $D(x)$ and $\bar{D}(x)$ denote the predicted and ground truth depth values, respectively, at pixel x; and the values of $I(x)_l$, $\bar{I}(x)_l$, $D(x)$, and $\bar{D}(x)$ are normalized into $[0, 1]$. Note that the size of the rain-free image $I(x)$ is the same as the input image, but the size of the depth map $D(x)$ is only $\frac{1}{16}$ of the input, but still, we put both weights ω_i and ω_d in Eq. (8) empirically as one.

Training parameters. We took the weights of the VGG network [39] trained on ImageNet [7] to initialize the weights in the encoder part of our network, and applied the method in [15] to initialize the weights in the other network parts. Moreover, we employed Adam [24] to optimize the network with the first momentum value of 0.9, the second momentum value of 0.99, and a weight decay of 5×10^{-4}. This optimization strategy can adaptively adjust the learning rate for individual network parameters: higher learning rate for frequently-updated parameters, and vice versa. We set the basic learning rate as 10^{-5}, reduced it by a factor of 0.316 after $70,000$ iterations, and stopped the learning after $100,000$ iterations. Lastly, we trained our network on a single NVidia Titan Xp GPU with a mini-batch size of one without data argumentation. The network was implemented based on CF-Caffe [18, 21]. The training took around 11.5 hours on the training set of RainCityscapes.

Inference. In testing, we feed a rain image as input to the network and obtain the predicted rain-free image in an end-to-end manner. On average, our DAF-Net takes only around 0.09 seconds to process a 256×512 image.

5. Experimental Results

5.1. RainCityscapes Dataset

We conducted a user study to evaluate the quality (i.e., how realistic) of our dataset as compared with three existing rain removal datasets and real photos with rain. To do so, we first collected 50 images: (i) ten real photos downloaded from the Internet by keyword search with "heavy rain photo," (ii) another ten rain images randomly selected from our RainCityscapes dataset, and (iii) thirty rain images from three recent datasets for rain removal (Rain800 [51],

DID-MDN [50], and Rain100H [47]) with ten images randomly selected from each dataset. Second, we recruited 34 participants: 15 females and 19 males, aged from 16 to 30 with mean 24.5. Then, we presented the 50 images to each participant in random order, and asked each of them to rate how real each image is in a scale from 1 (fake) to 10 (real). Therefore, we obtained 340 ratings (34 participants $\times$ 10 images per category) altogether for each category: real photo, our dataset, and the other three datasets.

Table 1 reports the results, showing that the ratings on our dataset are far closer to the ratings on the real photos compared with the other three datasets. This clearly shows that our dataset has more realistic rain images compared to the others and that our method is able to synthesize realistic rain on photos; see again Figure 3. However, our ratings still lag behind the real photos; at the end of the user study, some participants reported that for our rain images, they saw no water splashing on the ground like the real photos.

5.2. Comparisons using Real Photos

Comparing with state-of-the-art rain removal methods. First of all, we downloaded 129 photos from the Internet by using keyword search with "heavy rain photo" (from which we randomly pick ten photos to form the real photo dataset employed in the user study; see Section 5.1). Then, we applied our network to produce rain-free images. Additionally, we applied the following state-of-the-art methods to remove rain in the real photos: DID-MDN [50], RESCAN [30], JOB [52], GMMLP [31], and DSC [32]. To conduct a fair comparison, for deep-learning-based methods DID-MDN and RESCAN, we obtained rain-free image results by using their implementations with the released training models, which were trained on their own datasets. For other methods JOB, GMMLP and DSC, we downloaded and applied their public code with recommended parameters to generate the rain-free image results.

Figures 2 & 6 show our comparison results, where the first column shows the input real photos with rain, while the second, third, and fifth columns show the rain-free photos produced by DID-MDN, RESCAN, and our method, respectively. From these results, we can see that existing methods tend to fail for large and small rain streaks in the rain photos and miss out the fog that comes with the rain. In contrast, our method was designed with a more realistic rain model, thus capable of removing rain streaks as well as the fog in the input rain photos.

Comparing with rain + haze removal. Observing that existing rain removal methods tend to miss out the fog that comes with the rain (see again the second and third columns in Figures 2 & 6), we are thus motivated to try a state-of-the-art haze removal method, i.e., DCPDN [49], to post process their rain-removal results. Similar to the rain removal methods, we used the public implementation and released training model of DCPDN to perform the dehazing.

The fourth column of Figures 2 & 6 show the rain+haze removal results using RESCAN and then DCPDN, as compared to our results in the fifth column. Clearly, further removing the haze reduces the fog, but then, the rain streaks (which were not removed) would become more obvious.

Table 2: Comparison with the state-of-the-arts using the PSNR and SSIM on the test set of RainCityscapes.

method		PSNR	SSIM
DAF-Net (ours)		**30.06**	**0.9530**
rain removal	DID-MDN [50]	28.43	0.9349
	RESCAN [30]	24.49	0.8852
	JOB [52]	15.10	0.7592
	GMMLP [31]	17.80	0.8169
	DSC [32]	16.25	0.7746
haze removal	DCPDN [49]	28.52	0.9277
	AOD-Net [27]	20.40	0.8243

5.3. Comparisons using RainCityscapes

Next, we quantitatively compare the performance of different methods on the RainCityscapes dataset using the available rain-free images as ground truths.

Evaluation metrics. We adopted the peak signal to noise ratio (PSNR) and structural similarity (SSIM) index to quantitatively evaluate the rain removal results from various methods as compared with the rain-free images as the ground truths; see [43] for the definitions of PSNR and SSIM. Although not perfectly true, a larger PSNR or SSIM generally indicates a better result.

Comparison results. Table 2 shows the comparison results with the state-of-the-art rain removal methods, s.t. DID-MDN [50], RESCAN [30], JOB [52], GMMLP [31], and DSC [32]. Among them, DID-MDN [50] and RESCAN [30] use deep neural networks to recover the background by learning the mapping function between the rain and rain-free images from the training data, while others employ the hand-crafted priors to remove rain streaks. Moreover, since our network is also able to remove fog in the rain images, we further compared our method with two state-of-the-art haze removal methods, *i.e.*, DCPDN [49] and AOD-Net [27]; they are both deep-learning-based methods, so we re-trained their models for rain removal.

For a fair comparison, we re-trained all the deep-learning based models on the training set of RainCityscapes and tested them as well as other methods on the testing set of RainCityscapes. The results are reported in Table 2, where our DAF-Net performs favorably against all the others in terms of both PSNR and SSIM. It shows that our network with the learned depth-attentional features has a strong capability to remove the rain streaks and fog in a depth-dependent manner. We illustrate the visual comparison re-

Figure 7: Comparison results produced from our method against those from the state-of-the-art methods on RainCityscapes.

Table 3: Comparison with the state-of-the-arts using the PSNR and SSIM on the Rain100H dataset [47].

method	PSNR	SSIM
DAF-Net (ours)	**28.44**	**0.8740**
DID-MDN [50]	25.00	0.7543
RESCAN [30]	26.45	0.8458
JBO [52]	16.09	0.5149
GMMLP [31]	14.26	0.5444
DSC [32]	15.66	0.4225

Table 4: Evaluation on the DAF-Net. Basic model is DAF-Net without the depth-attentional features (DAF); see Eq. (6) and Section 4.2 for the definition and details of n.

method	n	PSNR	SSIM
basic	-	28.56	0.9457
DAF-Net	16	29.90	0.9527
	32	29.94	0.9528
	64	**30.06**	**0.9530**
	128	29.93	0.9524

sults in Figure 7, where our method can clearly remove the rain streaks and fog, while others tend to produce artifacts on the images or fail to remove large rain streaks, which are also revealed in the corresponding numerical values.

5.4. Comparisons using the Rain100H dataset

Besides RainCityscapes, we compared our network with other methods using the recent Rain100H dataset [47] we downloaded from its project website. Since depth map is not available in this data, we assume a constant depth value of 0.5 on the whole image, i.e., we simply ignore the depth and add rain streaks as a 2D overlay on the rain-free input photos. Table 3 reports the comparison results, where our method also outperforms the other rain removal methods.

5.5. Evaluation on Network Design

Component analysis. We performed an ablation study on our RainCityscapes dataset to evaluate the effectiveness of the depth-attentional features (DAF). The first row of Table 4 shows the results from a basic model, which is built by removing the top decoder branch in Figure 4, so the network only takes the feature map from the bottom decoder branch to generate the rain-free images without the DAF. Comparing the first and fourth rows in Table 4, we can see that our full network with the DAF can produce rain-free images that are more faithful to the ground truths.

Architecture analysis. Inside our network, we empirically determine the value of n, which corresponds to the number of attention weights (n) in the depth-attentional features (DAF); see Section 4.2. Conceptually, a large n means we learn more independent depth levels for the rain streak and fog pattern with less intra-class variance; however, the trade-off is to reduce the number of feature channels accordingly in each level. Table 4 presents the results, showing that the best performance is achieved when n is 64. Hence, we set n as 64 in the network.

6. Conclusion

In this work, we explore the visual effects of rain subject to scene depth and formulate a rain imaging model with rain streaks and fog. Based on the model and Cityscapes dataset, we synthesize more realistic rain images with ground-truth rain-free photos, and prepare the new RainCityscapes dataset for rain removal. Further, we formulate an end-to-end neural network, design the depth-guided attention mechanism, and train the network to learn the depth-attentional features to remove rain streaks and fog in the input rain image. In the end, we test our network on real photos and various datasets, and compare it with the state-of-the-art methods to demonstrate its superiority qualitatively and quantitatively. In the future, we plan to further explore the potential of our depth-attentional features for removing other weather-related artifacts and investigate high-level semantic scene understanding from the rain images.

Acknowledgments

This work was supported by the National Basic Program of China, 973 Program (Project no. 2015CB351706), the Shenzhen Science and Technology Program (Project no. JCYJ20170413162617606), the Hong Kong Research Grants Council (Project no. CUHK 14225616 & CUHK 14203416), and the CUHK Direct Grant for Research 2018/2019. Xiaowei Hu is funded by the Hong Kong Ph.D. Fellowship.

References

[1] P. C. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 86(2-3):256, 2010. 2

[2] N. Buch, S. A. Velastin, and J. Orwell. A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):920–939, 2011. 1

[3] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. DehazeNet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016. 2

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 5

[5] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*, pages 1968–1975, 2013. 1, 2

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 3

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6

[8] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017. 1, 2

[9] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *CVPR*, pages 1715–1723, 2017. 1, 2

[10] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *CVPR*, volume 1, pages I–I, 2004. 2

[11] K. Garg and S. K. Nayar. When does a camera see rain? In *ICCV*, pages 1067–1074, 2005. 4

[12] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM Trans. on Graphics (SIGGRAPH)*, volume 25, pages 996–1002. ACM, 2006. 2

[13] K. Garg and S. K. Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007. 1, 2, 3, 4

[14] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):1397–1409, 2013. 4

[15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *CVPR*, pages 1026–1034, 2015. 6

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[17] X. Hu, X. Xu, Y. Xiao, H. Chen, S. He, J. Qin, and P.-A. Heng. SINet: A scale-insensitive convolutional neural network for fast vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):1010–1019, 2019. 1

[18] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P.-A. Heng. Direction-aware spatial context features for shadow detection. In *CVPR*, pages 7454–7462, 2018. 6

[19] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on Multimedia*, 16(1):83–93, 2014. 1, 2

[20] J. Janai, F. Güney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *arXiv preprint arXiv:1704.05519*, 2017. 1

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6

[22] L.-W. Kang, C.-W. Lin, and Y.-H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742, 2012. 1, 2

[23] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *ICIP*, pages 914–917, 2013. 1

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[25] H. Koschmieder. Theorie der horizontalen sichtweite. *Beitrage zur Physik der freien Atmosphare*, pages 33–53, 1924. 3

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 5

[27] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. AOD-Net: All-in-one dehazing network. In *ICCV*, pages 4770–4778, 2017. 2, 7

[28] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin. Non-locally enhanced encoder-decoder network for single image de-raining. *arXiv preprint arXiv:1808.01491*, 2018. 1

[29] R. Li, J. Pan, Z. Li, and J. Tang. Single image dehazing via conditional generative adversarial network. In *CVPR*, 2018. 2

[30] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, pages 262–277, 2018. 1, 2, 6, 7, 8

[31] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016. 1, 2, 4, 7, 8

[32] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015. 1, 2, 7, 8

[33] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, 2006. 1

[34] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*, 2018. 2

[35] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, and M.-H. Yang. Gated fusion network for single image dehazing. In *CVPR*, pages 3253–3261, 2018. 2

[36] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang. Video desnowing and deraining based on matrix decomposition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[37] C. Sakaridis, D. Dai, and L. Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126:973–992, 2018. 3, 4

[38] V. Santhaseelan and V. K. Asari. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, 112(1):71–89, 2015. 2

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[40] S. Sivaraman and M. M. Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1773–1795, 2013. 1

[41] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang. Exploiting image structural similarity for single image rain removal. In *ICIP*, pages 4482–4486, 2014. 1, 2

[42] A. K. Tripathi and S. Mukhopadhyay. Removal of rain from videos: a review. *Signal, Image and Video Processing*, 8(8):1421–1430, 2014. 2

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 7

[44] Y. Weber, V. Jolivet, G. Gilet, and D. Ghazanfarpour. A multiscale model for rain rendering in real-time. *Computers & Graphics*, 50:61–70, 2015. 4

[45] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 4, 5

[46] D. Yang and J. Sun. Proximal Dehaze-Net: A prior learning-based deep network for single image dehazing. In *ECCV*, pages 702–717, 2018. 2

[47] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *CVPR*, pages 1357–1366, 2017. 1, 2, 3, 6, 7, 8

[48] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Adherent raindrop modeling, detectionand removal in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1721–1733, 2016. 2

[49] H. Zhang and V. M. Patel. Densely connected pyramid dehazing network. In *CVPR*, 2018. 2, 6, 7, 8

[50] H. Zhang and V. M. Patel. Density-aware single image deraining using a multi-stream dense network. In *CVPR*, pages 695–704, 2018. 1, 2, 3, 6, 7, 8

[51] H. Zhang, V. Sindagi, and V. M. Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017. 1, 3, 6

[52] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng. Joint bi-layer optimization for single-image rain streak removal. In *ICCV*, pages 2526–2534, 2017. 1, 2, 7, 8