RainyScape: Unsupervised Rainy Scene Reconstruction using Decoupled Neural Rendering (Supplementary Materials)

Paper ID: 3249

1 DATASET DETAILS

1.1 Existing deraining datasets.

The field of rain removal has seen the development of numerous datasets. To facilitate comparison, we present a summary of commonly utilized datasets in Table 1, which includes information about data resolution, availability of ground truth, and whether the data is real or simulated. The table also provides details on the rainy data simulation method employed, as well as information on whether the data is in video or multi-view format.

Acquiring pairwise real-world datasets for image deraining is a challenging task due to the difficulty in obtaining ground truth rain-free images. Wang et al. [7] tackled this issue by capturing video data of a static scene with rain, analyzing the video's histogram, and designing an algorithm to extract the corresponding rain-free image. While this approach allowed them to generate paired real-scene data, the shooting conditions were highly constrained, requiring static scenes and a stationary camera, which differs significantly from practical applications.

Alternatively, researchers have employed simulation methods to create synthetic deraining datasets, including the use of classic algorithms [4], the development of new algorithms [8], and the utilization of commercial software. Yan et al. [8] synthesized pairs of rain-free light field data by leveraging depth information and considering the effects of rain streaks, motion blur, and fog. *However, their simulation method has inherent limitations that may not accurately reflect real-world rainy scenarios.* Specifically, two main issues arise: 1) Rain is rendered independently of the scene, resulting in a loss of interaction between the rain and the scene, leading to inaccuracies in color and distribution. 2) Setting multiple cameras to render a consistent rain map does not guarantee that the simulated camera parameters match those of real cameras, causing a domain gap between the rain and the background. To better represent such scenarios, we created a rainy scene using Maya software and modeled the scene and rain. We used ray tracing rendering to obtain paired data that more accurately captures the effects of a real-world scene.

1.2 Dataset Description

We construct the dataset designed to provide a comprehensive and diverse set of scenes for the draining task. We generated the dataset by *Maya* [1], a powerful 3D modeling and rendering software that can simulate vivid visual effects of rainy weather. Our dataset contains 10 scenes, each with 50 viewpoints.

1.2.1 View Consistent Raindrops. In real-world scenarios, when one captures a rainy scene with multiple cameras from different viewpoints, the projections of the same raindrop in different captured images should satisfy view consistency, which is the most important characteristic of light field datasets. To ensure the view consistency of raindrops in our dataset, we constructed the 3D model for each raindrop in the scene and rendered raindrops together with other 3D scene objects. Specifically, we first constructed a 3D raindrop field for each scene where raindrops were uniformly distributed in the 3D space and let the raindrop field cover all scene objects captured by different cameras. Then, we generated all view images via the *Arnold for Maya* [1] renderer.

Note that other rainy datasets usually construct rainy images in the 2D image space, i.e., combine the clean image and the rain-streak image based on a simple linear formulation. Although some methods generate the rain-streak image by considering the scene depth, it is still difficult to generate real-world-like rainy images because the generation of rain-steak image is independent of the clean image, and there are no interaction effects, e.g., refraction, reflection, and scattering, between raindrops and light rays. In contrast, our dataset constructs raindrops, scene objects, and light sources in 3D space of *Maya* [1] software which can generate interaction effects between raindrops and light rays during the render process and produce more realistic visual effects of rainy scenes.

Besides, our dataset provides a range of rain densities and directions, as well as different orientations of rain streaks in different scenes. This is important as rain in the real world is often accompanied by wind, causing the orientation of rain streaks to vary. The density of rain streaks is also varied to reflect the different levels of rainfall experienced in the real world.

1.2.2 Scene Objects. To ensure that the dataset is representative of real-world scenarios, our dataset constructs all scenes using outdoor objects such as cars, bridges, roads, buildings, and boats, etc. This is intended to make it easier for supervised deep learning-based deraining methods to generalize to other unseen outdoor rainy scenes.

1.2.3 Camera Distribution. The viewpoints of each scene are distributed roughly on the surface of a sphere, with all cameras oriented toward the center of the sphere. All 50 cameras make up an array, with each row having 10 cameras. This distribution is intended to maximize the overlap rate of the scene content captured by different cameras, providing compensatory and diverse information for use in deraining tasks.

1.3 Data Format

1.3.1 View Images. Each view image is in *.png* format with a size of 1024 × 1024. We named all 50 view images in a zigzag sequence in the camera array, i.e.,

1.3.2 Depth Maps. We provide the ground truth depth map of each view in *.mat* format. The naming format of depth maps is the same as that of view images, i.e., the zigzag sequence in the camera array.

Table 1: Comparison of deraining datasets. The data simulation methods include rain streak synthesis algorithms [4, 8] and commercial software (Photoshop, Adobe, Maya).

Dataset	Data Type	Resolution	Video	Multi-view	Simulation Method
Rain200L/H [9]	*Syn*	481 × 321	×	×	Algorithm [4]
DDN-Data [3]	*Syn*	512 × 384	×	×	Photoshop
DID-Data [10]	*Syn*	586 × 586	×	×	Photoshop
RainCityscapes [5]	*Syn*	2048 × 1024	×	×	Photoshop
RainSynL/H25 [6]	*Syn*	640 × 360	✓	×	Algorithm [4]
Proposed	*Syn*	1024 × 1024	×	✓	Maya
NTURain [2]	*Syn, Real*	640 × 480	✓	×	Adobe
RLMB [8]	*Syn, Real*	5 × 5 × 512 × 512	×	✓	Algorithm [8]
SPA-Data [7]	*Real*	256 × 256 (train) 512 × 512 (test)	×	×	Algorithm [7]

001.png,	002.png,	...	010.png
011.png,	012.png,	...	020.png
...	...	...	...
041.png,	042.png,	...	050.png

1.3.3 Camera Parameters.

1.3.3 Camera Parameters. We provide the ground truth camera parameters of all views in a *.csv* file. The *.csv* file is organized as follows.

- **Camera Name**: the name of each camera with format *cameraShape-i* to indicate the i^{th} camera's parameters.
- **Position X**: the x value of the camera position in the world coordinate.
- **Position Y**: the y value of the camera position in the world coordinate.
- **Position Z**: the z value of the camera position in the world coordinate.
- **Rotation X**: the camera Euler angle degree around the x-axis of the world coordinate system.
- **Rotation Y**: the camera Euler angle degree around the x-axis of the world coordinate system.
- **Rotation Z**: the camera Euler angle degree around the x-axis of the world coordinate system.
- **Focal Length**: the focal length of the camera in *mm*.
- **Horizontal Aperture**: the horizontal size of the camera aperture in *mm*.
- **Vertical Aperture**: the vertical size of the camera aperture in *mm*.

2 ADDITIONAL EXPERIMENTAL RESULTS

To provide a more comprehensive evaluation of our proposed method, we present a **video demo** showcasing experimental results on four test datasets from our proposed dataset. The demo includes our generated results, predicted rain maps, and comparisons with baseline methods, allowing readers to gain a detailed understanding of our method's performance across various scenarios.

To further evaluate the effectiveness and generalizability of our approach in real-world scenarios, we extended the RainyScape-3DGS framework to process video data captured in a natural rainy environment. Using a mobile phone, we recorded an outdoor scene on a rainy day, focusing on traffic lights as the primary foreground objects. This setup allowed us to assess the performance of our method in the presence of dynamic rain and complex scene geometry. The captured video was then cropped to obtain a sequence of 110 frames, each with a resolution of 720 × 720 pixels. Subsequently, we employed Colmap software to estimate the camera parameters necessary for rendering.

It is worth noting that the video data captured in this real-world scenario ***presents additional challenges*** *compared to the view-consistent rain data used in our primary experiments*. The outdoor scene is open and unconstrained, with ***dynamic rain*** that varies in intensity and direction over time. These factors introduce complexities in terms of rain streak appearance, motion, and interaction with the scene, making rain-free scene reconstruction *more difficult*. By successfully processing this real-world dataset, we demonstrate the robustness and adaptability of our method to diverse and challenging rainy conditions encountered in practical applications. The experimental results of the real-world video processing, which are showcased in the demo, yield several key observations:

(1) Our method can be effectively extended to process video data, significantly suppressing the impact of rain on the scene.
(2) The proposed framework successfully decouples high-frequency details from dynamic rain, exhibiting superior preservation of high-frequency details while minimizing the presence of rain streaks compared to baseline methods.
(3) As the camera captures an open scene, distant background objects are convolved with only a small amount of Gaussian kernels, resulting in a lower rendering quality for the background.

These findings highlight the potential of our method to be applied to a wide range of real-world scenarios, even when dealing with videos captured using mobile devices. However, it is important to acknowledge that the rendering quality of distant background objects in open scenes may be compromised due to the limited influence of the Gaussian kernels. Future research could investigate techniques to enhance the background rendering quality in such cases.

3 DISCUSSION AND FUTURE WORK

Our proposed RainyScape demonstrates effectiveness in handling real-world rainy scenarios, but several limitations warrant further discussion and present opportunities for future research. Firstly, the post-processing stage relies on a simple blending approach based on the generated rain map, which heavily depends on the accuracy of the estimated rain map. Secondly, rain streaks close to scene objects may be mistakenly identified as part of the scene, leading to ambiguities in the deraining process. Lastly, during the 3D Gaussian Splatting (3DGS) process, a portion of the rain streaks may be wrapped by Gaussian kernels and treated as part of the scene, resulting in rendered results that still exhibit some residual rain effects.

To address these limitations, several avenues for future work can be explored. Investigating advanced blending techniques, such as learning-based methods or adaptive weighting schemes, could improve the post-processing stage and enhance the overall deraining quality. Incorporating semantic information could provide additional context to disambiguate rain streaks from scene objects. Furthermore, exploring corresponding processing methods during Gaussian kernel splitting or deformation could help suppress the residual rain remaining in the scene after the 3DGS process. By addressing these challenges and exploring the suggested research directions, we believe that the performance and applicability of our method can be further enhanced.

REFERENCES

[1] Autodesk, INC. [n. d.]. *Maya*. https:/autodesk.com/maya

[2] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. 2018. Robust video content alignment and compensation for rain removal in a cnn framework. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6286–6295.

[3] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. 2017. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3855–3863.

[4] Kshitiz Garg and Shree K Nayar. 2006. Photorealistic rendering of rain streaks. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 996–1002.

[5] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. 2019. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 8022–8031.

[6] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. 2018. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3233–3242.

[7] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. 2019. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12270–12279.

[8] Tao Yan, Mingyue Li, Bin Li, Yang Yang, and Rynson W. H. Lau. 2023. Rain Removal From Light Field Images With 4D Convolution and Multi-Scale Gaussian Process. *IEEE Transactions on Image Processing* 32 (2023), 921–936.

[9] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2017. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1357–1366.

[10] He Zhang and Vishal M Patel. 2018. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 695–704.