GTAV-NightRain: Photometric Realistic Large-scale Dataset for Night-time Rain Streak Removal

Fan Zhang  
Beijing Institute of Technology

Shaodi You  
University of Amsterdam

Yu Li  
International Digital Economy Academy

Ying Fu  
Beijing Institute of Technology

Figure 1: Comparisons between existing datasets and our GTAV-NightRain dataset. Rain streaks look unrealistic in the first row because the rain is superposed on 2D images of sunny days without interaction with the scene illumination and structure. For night scenes, they are also based on 2D superposing and their rain does not have real interaction with the scenes either. Rain streaks in our dataset spread in 3D space and interact with light sources both geometrically and photoelectrically, making them more realistic. Real images from the Internet are included for comparison.

Abstract

Rain is transparent, which reflects and refracts light in the scene to the camera. In outdoor vision, rain, especially rain streaks degrade visibility and therefore need to be removed. In existing rain streak removal datasets, although density, scale, direction and intensity have been considered, transparency is not fully taken into account. This problem is particularly serious in night scenes, where the appearance of rain largely depends on the interaction with scene illuminations and changes drastically on different positions within the image. This is problematic, because unrealistic dataset causes serious domain bias. In this paper, we propose GTAV-NightRain dataset, which is a large-scale synthetic night-time rain streak removal dataset. Unlike existing datasets, by using 3D computer graphic platform (namely GTA V), we are allowed to infer the three dimensional interaction between rain and illuminations, which insures the photometric realness. Current release of the dataset contains 12,860 HD rainy images and 1,286 corresponding HD ground truth images in diversified night scenes. A systematic benchmark and analysis are provided along with the dataset to inspire further research.

1 Introduction

Rain reflects and refracts light and causes visibility degradation, making it necessary to remove rain streaks from rainy images. Deraining has been studied for decades and deep learning methods [9, 20, 37, 47] gain their popularity recently like in other research areas [18, 16, 23, 24, 3]. But
they mostly rely on synthetic data for training due to practical limitations, where rain streak layer is superposed onto clean images [22, 49, 47]. Synthetic datasets help mitigate the problem of data shortage considering different aspects of rain such as scale, intensity, density and accumulation.

However, they ignore the photometric realness. Because rain reflects and refracts light in the scene to the camera, its appearance can be changed greatly by environmental illuminations. This is particularly problematic in night scenes, because illumination in night scenes cannot be considered as parallel and uniform. Moreover, rain appearance varies drastically depending on its position in the 3D space. Existing datasets do not consider this issue and superpose 2D rain layer onto clean images. In a word, their rain streaks do not interact with the environment and therefore look unrealistic.

In this paper, instead of layer superposing, we propose to generate data by rendering rain streaks in 3D virtual scenes. Specifically, we use GTA V [10] which is a popular synthetic platform for AI in urban scenes [31, 30, 17]. We investigate into the underlying asset files and mechanism related to rain rendering in GTA V. Then we make modifications using game mods to set up a suitable platform for data collection. This allows us to edit the existence of rain in a scene while keeping everything else unchanged. In another word, we can obtain images pairs with (input) and without rain (ground truth).

Practically, in this paper, we first introduce the geometry and photometry of rain streaks. Second, we describe the technical details on how to edit the assets and settings of GTA V so as to collect the data. Third, we present our new dataset GTA V-NightRain, which contains 12,860 rainy images and 1,286 clean images of a diversified night scenes. Finally, we benchmark to analyze the existing methods using our new dataset. The contribution of this paper is concluded as follows:

- The proposed dataset is the first to consider both geometric and photometric realness for rendering airborne rain, especially in night scenes.
- The proposed dataset is large scale with diversified urban scenes and high graphic quality, which minimizes dataset bias.
- A systematical and comprehensive benchmark is provided using the proposed dataset so as to inspire further research.

2 Related work

In this section, we first briefly introduce existing deraining datasets and some SOTA methods which are evaluated on our dataset. Then we also mention some work related to collecting data from GTA V.

Deraining datasets There are many deraining datasets proposed these years focusing on certain rain streak properties, most of which are synthesized following [12]. Li et al. [21] and Yang et al. [40] also reviewed the methods and datasets for rain removal. Li et al. [22] provided 12 images for test (Rain12) following Garg and Shree [11]. Yang et al. [39] synthesized rain100L and rain100H with one and five types of rain streaks for light and heavy rain cases. Fu et al. [9] synthesized 14000 rainy images (Rain14000) with different streak orientations and magnitudes using Photoshop on 1000 clean images from UCID dataset [32], BSD dataset [1] and Google image search and. Zhang and Patel [47] presented a dataset of 12000 images (Rain1200) with rain-density labels for 3 density levels. Zhang et al. [48] synthesized 800 images (Rain800) by following [9, 8]. Wang et al. [37] proposed a semi-automatic method to generate clean image from real rain image sequence and constructed a large-scale dataset (SPA-data) of about 29500 rain/rain-free image pairs. Li et al. [19] proposed NYU-Rain dataset containing 16200 images and Outdoor-Rain dataset including 10500 images with rain accumulation effects rendered with depth information. Similarly, Hu et al. [14] proposed RainCityscapes based on Cityscapes dataset [5] and the rain patches are selected from [22]. Li et al. [20] proposed a large scale dataset (MPID) covering rain streaks, raindrops and rain/mist cases and also provided annotated real images for detection task after deraining.

These datasets mainly focus on the scale, shape, direction and intensities of rain streaks and some may further consider rain accumulation and different density levels. However, photometry property is not considered for rain streak removal as frequently as adherent raindrop removal [13, 42, 44, 43], even though theories on this issue have been well studied [11] that long ago. Thus, we attempt to take the first spite on the dataset containing colored rain steaks caused by environmental illuminations.
Table 1: Comparisons between existing datasets and GTA V-NightRain. Our dataset is not only large in amount and resolution, but also renders rain streak layer in 3D virtual scene considering photometry property rather than superposing 2D rain layer onto clean images omitting interactions with environment. Note that SPA-data still contains around 5 night scenes even though it is not intended for nighttime deraining.

| Dataset          | Number (#train/#test) | Resolution      | Real/Synthetic | Layer Type | Day/Night | Light Interactions |
|------------------|------------------------|-----------------|----------------|------------|-----------|-------------------|
| Rain12           | 12                     | 481 × 321       | Synthetic      | 2D         | Day       | /                 |
| Rain100L         | 1800/200               | 481 × 321       | Synthetic      | 2D         | Day       | /                 |
| Rain100H         | 1800/200               | 481 × 321       | Synthetic      | 2D         | Day       | /                 |
| Rain14000        | 9100/4900              | 512 × 384       | Synthetic      | 2D         | Day       | /                 |
| Rain12000        | 12000/1200             | 1024 × 512      | Synthetic      | 2D         | Day       | /                 |
| Rain800          | 700/100                | 1024 × 384      | Synthetic      | 2D         | Day       | /                 |
| NYU-Rain         | 13500/2700             | 640 × 480       | Synthetic      | 2D         | Day       | /                 |
| Outdoor-Rain     | 9000/1500              | 720 × 480       | Synthetic      | 2D         | Day       | /                 |
| RainCityscapes   | 9432/1188              | 2048 × 1024     | Synthetic      | 2D         | Day       | /                 |
| SPA-data         | 28500/1000             | 512 × 512       | Real           | 3D         | Day&Night | Y                 |
| MPID             | 2400/250               | 512 × 384       | Synthetic&Real | 2D         | Day       | /                 |
| GTA-V-NightRain  | 10000/2860             | 1920 × 1080     | Synthetic      | 3D         | Night     | Y                 |

Deraining methods

Ren et al. [29] proposed a baseline network named PReNet for single image deraining with the simple combination of ResNet and multi-stage recursion and achieved promising performance. Wang et al. [31] proposed a novel SPatial Attentive Network (SPANet) to remove rain streaks in a local-to-global manner. Wang et al. [36] designed a Rain Convolutional Dictionary Network (RCDNet) for better interpretability and integration with physical structure of general rain streaks. Deng et al. [6] proposed Detail-Recovery image Deraining Network (DRDNet) to remove rain streaks and recover lost details. Yi et al. [41] utilized residue channel prior to protect struture information and guide the training of Structure Preserving Deraining Network (SPDNet). Huang et al. [15] proposed a novel Memory-Oriented Semi-Supervised method (MOSS) for single image deraining to learn rain degradations from both labeled synthetic and unlabeled real-world data. Zamir et al. [46] presented MPRNet to progressively learn restoration functions for the degraded inputs. There were also powerful image restoration backbones in recent years which can achieve good performance on deraining, such as Restormer [45], HINet [4] and MAXIM [35].

Dataset using GTA V

GTA V is a popular platform for synthetic datasets. However, few work use it for low-level vision. Richter [31] first presented an approach to rapidly producing pixel-accurate semantic label maps for images synthesized by GTA V and further presented a new benchmark suite [30] for visual perception. Krahenbuhl [17] presented a framework to extract ground truth supervision from video games in real-time. Doan et al. [7] also presented G2D that could be utilized to collect the dataset in GTA V.

3 Prerequisite: geometry and photometry of rain

Unlike most existing dataset, we start from the fundamental modeling so as to ensure realness. In this section, we briefly introduce the necessary modeling of geometry and photometry on rain. Then we discuss why rendered rain in GTA V is a realistic approximation. Finally, we describe how to modify GTA V into a platform for the proposed dataset.

3.1 The shape of raindrop

The shape of raindrop was well studied dozens of years ago and is commonly assumed to be spherical [12]. Raindrop tends to be spherical with small size and becomes equilibrium when the size increases due to the pressure of air. It can be modeled with a 10th order cosine distortion of a sphere [2,12]:

$$r(\theta) = a \left( 1 + \sum_{n=1}^{10} c_n \cos(n\theta) \right),$$  (1)
3.2 Photometry of raindrop

Appearance of rain purely depends on its reflection and refraction. As shown in Fig. 2(a), considering a point B on the surface of the raindrop with a surface normal $\hat{n}$, rays from the scene $\hat{r}$, $\hat{s}$ and $\hat{p}$ are directed towards the imaging sensor via refraction, specular reflection and internal reflection, respectively. Therefore, the radiance $L(\hat{n})$ of point B can be expressed as the sum of the radiance $L_r$ of refracted ray, radiance $L_s$ of specularly reflected ray and radiance $L_p$ of internally reflected ray:

$$L(\hat{n}) = L_r(\hat{n}) + L_s(\hat{n}) + L_p(\hat{n}),$$  \hspace{1cm} (2)

due to limitation of space, we summarize the details in supplemental materials. As concluded in [12], refraction is dominating the appearance.

3.3 Photometry of rain streaks

While stationary raindrop can be regarded as spherical lens, falling raindrop falls fast and can not be exactly captured as it is by a camera with a normal exposure time like 30ms. Motion blur left on the image can greatly change the appearance of recorded raindrop in the scene and these trajectories are commonly called as rain streaks. The intensity of a rain streak depends on the brightness of the raindrop as well as the background scene radiance and integration time of the camera [12].

Assuming that we observe a raining scene with a linear camera with exposure time $T$, the intensity $I_r$ at the pixel effected by a raindrop is a linear combination of the irradiance $E_b$ of the background and the irradiance $E_r$ of the raindrop:

$$I_r = \int_0^T E_r dt + \int_T^T E_b dt.$$  \hspace{1cm} (3)

For a static background, $E_b$ can be assumed to be constant during the exposure time $T$ and thus the above equation can be expressed as:

$$I_r = \tau \bar{E}_r + (T - \tau) E_b,$$  \hspace{1cm} (4)

where $\tau$ is the time a raindrop stays at a pixel and $\bar{E}_r$ is the time-averaged irradiance contributed by the raindrop. For a pixel observing no raindrop, we have $I_b = E_b T$. Therefore, the change of

Figure 2: (a) Illustration on photometry of spherical raindrop [12]. Radiance arriving at imaging sensor consists of refracted, reflected and internally reflected light rays from environmental illumination. (b) The asset of raindrop in GTA V [10] for set1. (c) The asset of raindrop we draw to create set2.
Figure 3: Illustration of photometry of rain streaks spread in 3D scene. Raindrops are responsive to dynamic and non-parallel light sources.

Intensity $\Delta I$ contributed by a rain streak can be expressed as:

$$\Delta I = I_r - I_b = \tau (\bar{E}_r - E_b) = -\beta I_b + \alpha,$$

with $\beta = \frac{\tau}{\bar{E}_r}$, $\alpha = \tau \bar{E}_r$. Based on the numerical bounds and observations provided by [12], we can know that a raindrop produces a positive change in intensity and stays at a pixel for a time far less than the integration time of a typical video camera. The change in intensities of pixels within a rain streak are linearly related to the background intensities $I_b$ and $\beta$ is in the range $0 < \beta < 0.039$, which means $\Delta I$ is mainly contributed by the intensity change caused by rain streak. And this is the reason why white rain streaks are directly superposed onto the image for most existing synthetic datasets.

4 GTAV-NightRain dataset

4.1 Motivation

In real world, it is almost impossible to collect paired data with and without rain while keep everything else pixel-wisely unchanged. Therefore, existing datasets seek solutions using synthetic rains. As shown in Table 1, existing datasets simply overlap an extra rain layer in 2D. However, as described in Section 3 and further illustrated in Figure 3, the appearance of rain streaks change drastically with space due to environmental illuminations, especially near artificial light sources in night scenes because they spread in three dimensional space and illumination in night scenes can not be considered as parallel and uniform.

In real case, spherical raindrops are observed as rain streaks due to motion blur. While with limited rendering frame rate, artifacts occur between intervals of frames for computer game. The rain assets are predefined to be rain-streak-like as shown in Fig 2(b) and avoid this problem. Photometry of rain exists in GTA V and there exist light interactions between environmental illuminations and rain streaks. Raindrops in this game can also refract and reflect lights. Furthermore, these raindrops also spread across the 3D space and interact with light sources.

With all these properties, in this paper, we propose to investigate into this issue by collecting such a dataset containing dynamic and environment responsive rain streaks. Rather than superposing 2D rain streak layer onto clean images like prior work, we acquire data by rendering rain streaks in 3D virtual scenes of GTA V which has been proven to be an effective way to collect synthetic data [31, 30, 17].

4.2 Controllable conditions

Weather types. In GTA V, weather changes following a predefined pseudo-random schedule. There are many weather types which are common in real life such as extra-sunny with no clouds, clear with a few clouds, clouds, rain, snow, foggy and etc. Under different weather, the environment changes accordingly. Sun, clouds, fog and other environment assets also change following predefined schedule. Here we focus on the data collection for deraining, which means we need the rain and rain-free weather like extra-sunny, clear or clouds.

Environmental illuminations. In GTA V, sun is the main light source during the day and changes greatly at different clocks such as dawn, sunset and the noon. While at night, artificial light sources account for the scene illumination, including street lights, billboards and neon lights. Moon can also
Figure 4: Example images in our GTAV-NightRain dataset. *set1*, *set2* and *set3* correspond to splits rendered with rain shape asset by default, ours and **Heavy Rain** mod, respectively. *set2* intend to show the dynamic response to different lighting sources. And *set3* intend to show the dynamic response to both complex lighting and structures. Please view in color and zoom in for details.

contribute a lot when there is no clouds. Here we focus on night scenes where artificial light sources have great effects on the appearance of rain streaks. Thus we collect rainy and rain-free images near these local light sources.

**Rain.** In this game, raindrop shape is predefined to be rain-streak-like as shown in Fig 2 (b). Raindrops are rendered with photometry property, which means rain streaks refract or reflect lights from background. This is important but not taken into account by existing datasets. Moreover, images captured in the game contain rain streaks spread in 3D scenes, which means depth and occlusion are also considered. Rain streaks falling nearby are larger than ones far away. With occlusion, we can capture rain streaks satisfying the space arrangements in the scene by entering buildings. Moreover, the underlying rain streak shape can be substituted and we can test different raindrop shapes in the rainy scene, as shown in 2 (c). With raindrop models more fit to physical one, we can get more realistic rainy images. We can also control the density of rain for diversity.

**Puddles, ripples and splashes.** As common in real world, there will form puddles on the ground as it rains. Ripples also show up when the raindrops hit the puddle and there will be also splashes. Wetness of the environment also changes with time. We can control the existence of them or change their scales with specific parameter setting. To keep everything pixel-wisely unchanged except rain streaks, we disable the puddles, ripples and splashes before collecting data. A side dataset with changes on puddles, ripples and splashes will be provided in future release.

4.3 Dataset setup

After figuring out basic elements and mechanisms in GTA V, we are ready to collect the paired data for deraining. For our dataset, we need to collect rainy images and corresponding rain-free ground truths containing colored rain streaks in night scenes. There are several steps described as follows:

1. To switch weather between *rain* and *clear* for example, we utilize the *SimpleTrainer* mod [33]. We can also use it disable the presence of moving pedestrians, vehicles and make the character controlled invisible. It is also important to adjust the time and freeze the clock to arrange the environment settings.

2. To modify the underlying files, we make use of the *OpenIV* toolkit [34] to manage mods and game files. We can change parameters of weather, lightings, assets and time schedules and also modify underlying assets like rain shape file.

3. To change the appearance of rain streaks and disable puddles, we find *Insane Rain* [25], *Real Rain* [26] and *Better Puddles* [27] fitting our needs. We figure out internal mechanisms on how these mods work and make modifications on our own.

4. The whole environment of GTA V changes following the schedule defined in *timecyle* file, corresponding to different weather. To keep lightings unchanged when switching weather, we should keep parameters in the files all the same.
Table 2: Quantitative results of compared methods on GTA V-NightRain dataset. All the models are trained on our dataset following the default setting by authors. Higher metrics mean better performance for PSNR and SSIM. Lower metrics mean better for LPIPS.

| Method      | Training | Set1       | Set2       | Set3       |
|-------------|----------|------------|------------|------------|
|             | PSNR     | SSIM       | LPIPS↓     | PSNR       | SSIM       | LPIPS↓     | PSNR       | SSIM       | LPIPS↓     |
| Input       |          |            |            |            |            |            | 33.29      | 0.9418     | 0.1531     |
| SPANet      |          |            |            |            |            |            | 35.35      | 0.9642     | 0.0965     |
| PReNet      |          |            |            |            |            |            | 38.73      | 0.9799     | 0.0365     |
| DRDNet      |          |            |            |            |            |            | 35.61      | 0.9595     | 0.1051     |
| RCDNet      |          |            |            |            |            |            | 38.69      | 0.9781     | 0.0404     |
| SPDNet      |          |            |            |            |            |            | 37.56      | 0.9729     | 0.0585     |
| MPRNet      | Set1     | 39.70      | 0.9836     | 0.0287     | 37.59      | 0.9767     | 0.0500     | 33.74      | 0.9521     | 0.0854     |
| SPANet      | Set2     | 34.26      | 0.9555     | 0.1063     | 34.63      | 0.9649     | 0.0832     | 30.12      | 0.9131     | 0.1824     |
| PReNet      |          | 36.62      | 0.9713     | 0.0627     | 38.45      | 0.9789     | 0.0323     | 32.28      | 0.9345     | 0.1156     |
| DRDNet      |          | 35.14      | 0.9564     | 0.1128     | 35.63      | 0.9647     | 0.0894     | 31.62      | 0.9116     | 0.1801     |
| RCDNet      |          | 36.21      | 0.9654     | 0.0780     | 37.31      | 0.9732     | 0.0531     | 31.94      | 0.9233     | 0.1399     |
| SPDNet      |          | 36.15      | 0.9647     | 0.0800     | 37.29      | 0.9729     | 0.0528     | 32.32      | 0.9274     | 0.1350     |
| MPRNet      |          | 37.38      | 0.9759     | 0.0476     | 39.70      | 0.9824     | 0.0232     | 32.94      | 0.9413     | 0.0951     |

Table 3: Model complexity. The FLOPs are calculated given a test image of size 256 × 256.

| Model       | Parameters (M) | FLOPs (G) |
|-------------|----------------|-----------|
| SPANet [37] | 0.28           | 36.3      |
| PReNet [29] | 0.17           | 66.4      |
| RCDNet [36] | 3.16           | 195.1     |
| DRDNet [6]  | 5.23           | 689.8     |
| SPDNet [41] | 3.32           | 96.6      |
| MPRNet [46] | 3.63           | 141.5     |

For more details of modifications, please refer to the supplemental materials. To set up for data collection, we only switch weather between rain and clear/clouds. The only difference is whether rain streaks exist and we keep wetness, cloud type and lightings unchanged. We disable moving pedestrians, vehicles and the character we control to avoid the change in scenes. We disable puddles, ripples and splashes to simplify the task and update the timecycle file to keep the same lightings.

With all settings configured, rainy images and corresponding ground truth can be easily obtained with only a few operations. We collect 10 rainy images with 1 rain-free image for each scene. For a data unit of 10 rainy images, we move the character to proper location such as the front of stores and turn the camera view to proper direction like the neon light. Then we switch the weathers and capture data. In total, we test two rain shapes including the default and the one we draw. We collect 5000/500 rainy images for the training/test sets for each rain shape, making up set1 and set2. We choose an area on purpose to collect the test set and the capture process of training data is finished away there. In addition, we also collect 1860 rainy images to form the set3 using the Insane Rain mod with larger density level covering more scenes and select 186 images from it for the hard case. Example images are provided in Figure 4. The game is rendered in 1080P on a NVIDIA RTX 3090 GPU.

5Benchmark and results

5.1 Experimental setups

Based on our GTA-V-NightRain dataset, we benchmark 6 existing deraining methods, including SPANet [37], PReNet [29], RCDNet [36], DRDNet [6], SPDNet [41] and MPRNet [46]. We follow the default training settings provided by authors and train these models on the training sets of set1 and set2. Then we test them on test sets of three subsets and real nighttime rainy images we acquired from the Internet. Model complexity is listed in Table 3 for comparison and all experiments are performed on a server with 4 NVIDIA GTX 1080Ti GPUs.
5.2 Quantitative results

We adopt commonly used PSNR and SSIM [38] as well as perceptual metrics LPIPS [49] for performance evaluation and quantitative results are listed in Table 2. We can see that MPRNet [37] trained on \( \text{set1} \) and \( \text{set2} \) both achieve the best performance on all test sets in terms of almost all four metrics, showing its excellent deraining ability. SPANet [37] and DRDNet [6] get relatively worse metrics than other methods while PReNet [29], RCDNet [36] and SPDNet [41] behave better than the former two methods. An interesting observation can be found that models with better performance (e.g., MPRNet [46] and PReNet [29] drop by about 2dB in PSNR when tested on different test set) suffers more from domain gap than those with lower performance (e.g., SPANet [37] and DRDNet [6] only drop by less than 0.5dB instead). It implies that deep models have to trade off between generalization ability and task performance, which makes powerful models more dependent on training data. When tested on \( \text{set3} \) which is harder, performance gap between \( \text{set1} \) and \( \text{set2} \) for all methods narrows to a promising extent. It means that models trained on \( \text{set2} \) are able to handle rain streaks with appearance they’ve never met before. This confirms the effectiveness of the rain streak layer drawn by us and the possibility to train more robust models by designing more diverse rain streak layers.

5.3 Qualitative results

Here we only provide visual results on \( \text{set3} \) for the limited space. Please refer to supplemental material for more qualitative results. As shown in Figure 5, SPANet [37] can only remove a few rain streaks while other methods achieve better results, among which MPRNet [46] behaves best. For the first test image, models trained on \( \text{set2} \) can remove more rain than those of \( \text{set1} \). For the second one, models are good at removing different rain streaks, i.e., MPRNet_Set1 removes colored rain streaks near the roof while MPRNet_Set2 tends to remove more rain streaks near the fuel dispenser.
Figure 6: Qualitative results of evaluated methods on real nighttime rainy images from the Internet. The pretrained model PReNet_Real1 is included for comparison. Please view in color and zoom in for details.

It confirms the effectiveness of our rain streak layer and the possibility to integrate real rain layers for data generation. For the last test image, there are still some rain streaks not removed for all methods and we can find that models like PReNet_Set2 and MPR_Set2 tend to remove texts on the blue board.

Besides, we collect real nighttime rainy images from the Internet and existing real deraining datasets for further evaluation. We also include a pretrained model PReNet_Real1 for comparison. As shown in Figure 6, behaviors of all tested models vary for different real images. For the first image containing blur rain streak, PReNet_Set1, PReNet_Set2, RCDNet_Set1 and MPRNet_Set1 remove most part of rain streaks except for the right-up corner. For the second image, DRDNet_Set2 and SPDNet_Set2 outperform other models and remove most rain streaks. For the last rainy image, we can see that PReNet_Set1, DRDNet_Set1 and RCDNet_Set1 remove rain streaks in front of headlights and DRDNet_Set1 achieves the best result. While the pretrained model PReNet_Real1 cannot properly handle these images. These results all prove that nighttime deraining is a challenging task and both models and training data play important role in final performance on real-world rainy images. And models trained on our dataset can indeed handle these nighttime rainy images, confirming the effectiveness of our dataset.

6 Limitations and future work

In this paper, we propose the first nighttime deraining dataset which considers photometry of rain streaks using GTA V. Although rainy images are more similar to real world ones than existing...
deraining datasets, there exist limitations in our current dataset. (I) The default graphic quality of GTA V falls behind recent games which support ray tracing techniques and there is large room for improvements. (II) Due to the rendering mechanism, the game cannot be static at the very moment and we have to disable moving pedestrians and vehicles, limiting the diversity of captured scenes. (III) There inevitably exists domain gap between our synthetic dataset and real-world data thus our dataset on its own is not enough for solving nighttime deraining. However, the development of this dataset still continues and there are a lot to do in future work.

1. It is promising to update the dataset with 4K resolution and good texture mods to further enhance the realness. It is portable to newly developed game with better graphic quality. Better realness in terms of graphic quality and rendering mechanism may both narrow the gap between graphically rendered images and real captured ones.

2. We focus on deraining in night scenes in this work but illuminations also vary in day scenes. It is promising to construct full-time datasets covering both day and night scenes, enabling possible usage in hybrid tasks like low light image enhancement and deraining. Meanwhile, hybrid training of existing datasets and GTAV-NightRain may also do help to achieving better robustness of learning-based models.

3. Many assets like puddles are disabled in this version of dataset but it is promising to construct datasets with more complex rain scenes to train robust models handling more than rain streaks.

4. Utilizing tools by Richter et al. [31] to extract assets from GPU may handle the problem of moving objects and enable more efficient way for data generation.

7 Conclusion

In this paper, we present a deraining dataset called GTAV-NightRain considering the photometry property of rain streak and there are 14146 images in total. Different from existing synthetic datasets, we collect data by rendering rain streaks in 3D virtual scenes in GTA V rather than superposing 2D rain streak layer on clean images. With proper modifications, we turn the game into a good platform to collect paired rainy and rain-free images, where we can easily change the weather and keep other conditions pixel-wisely unchanged. We also evaluate several SOTA deraining methods on our dataset to find what problems they will meet when facing colored rain streaks and give insights on future research interest. Portable to other newly developed games, collecting deraining dataset in such way can be benefited from the advancement of both more realistic rain modeling and better graphic quality of games, which makes the collected data better. We will also extend our dataset to day time scenes to further enrich its diversity and enhance the realness with higher resolution and graphic quality.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. IEEE TPAMI, 33(5):898–916, 2010.
[2] Kenneth V Beard and Catherine Chuang. A new model for the equilibrium shape of raindrops. Journal of the Atmospheric sciences, 44(11):1509–1524, 1987.
[3] Haibo Chen, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, Dongming Lu, et al. Artistic style transfer with internal-external learning and contrastive learning. NeurIPS, 2021.
[4] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In CVPRW, 2021.
[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016.
[6] Sen Deng, Mingqiang Wei, Jun Wang, Yidan Feng, Luming Liang, Haoran Xie, Fu Lee Wang, and Meng Wang. Detail-recovery image deraining via context aggregation networks. In CVPR, 2020.
[7] Anh-Dzung Doan, Abdul Mohsi Jawaid, Thanh-Toan Do, and Tat-Jun Chin. G2d: from gta to data. arXiv preprint arXiv:1806.07381, 2018.
[8] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. IEEE TIP, 26(6):2944–2956, 2017.
[9] Xueyang Fu, Jiabin Huang, Deliu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In CVPR, 2017.
[10] Rockstar Games. Grand theft auto v. https://www.rockstargames.com/gta-v
[11] Kshitiz Garg and Shree K Nayar. Photorealistic rendering of rain streaks. *ACM TOG*, 25(3):996–1002, 2006.

[12] Kshitiz Garg and Shree K Nayar. Vision and rain. *IJCV*, 75(1):3–27, 2007.

[13] Zhixiang Hao, Shaodi You, Yu Li, Kunming Li, and Feng Lu. Learning from synthetic photorealistic raindrop for single image raindrop removal. In *ICCVW*, 2019.

[14] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *CVPR*, 2019.

[15] Huaibo Huang, Aijing Yu, and Ran He. Memory oriented transfer learning for semi-supervised image deraining. In *CVPR*, 2021.

[16] Lei Ke, Xia Li, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Prototypical cross-attention networks for multiple object tracking and segmentation. *NeurIPS*, 2021.

[17] Philipp Krähenbühl. Free supervision from video games. In *CVPR*, 2018.

[18] Jingjing Li, Wei Ji, Qi Bi, Cheng Yan, Miao Zhang, Yongri Piao, Huchuan Lu, et al. Joint semantic mining for weakly supervised rgb-d salient object detection. *NeurIPS*, 2021.

[19] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *CVPR*, 2019.

[20] Siyuan Li, Iago Breno Araujo, Wengqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaoijie Guo, and Xiaochnao Cao. Single image deraining: A comprehensive benchmark analysis. In *CVPR*, 2019.

[21] Siyuan Li, Wengqi Ren, Feng Wang, Iago Breno Araujo, Eric K Tokuda, Roberto Hirata Junior, Zhangyang Wang, Xiaochnao Cao, et al. A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives. *IJCV*, 129(4):1301–1322, 2021.

[22] Yu Li, Robby T Tan, Xiaoijie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016.

[23] Yunan Liu, Shanshan Zhang, Yang Li, and Jian Yang. Learning to adapt via latent domains for adaptive semantic segmentation. *NeurIPS*, 2021.

[24] Yan Liu, Zhijie Zhang, Li Niu, Junjie Chen, and Liqing Zhang. Mixed supervised object detection by transferring mask prior and semantic similarity. *NeurIPS*, 2021.

[25] LUISDACOS. Insane rain mod 1.2. [https://www.gta5-mods.com/misc/insane-rain-mod](https://www.gta5-mods.com/misc/insane-rain-mod). Online; accessed 11-03-2022.

[26] LUISDACOS. Real rain 1.01. [https://www.gta5-mods.com/misc/enhanced-rain-mod](https://www.gta5-mods.com/misc/enhanced-rain-mod). Online; accessed 11-03-2022.

[27] markyparky41. Better puddles 1.3. [https://www.gta5-mods.com/misc/better-puddles](https://www.gta5-mods.com/misc/better-puddles). Online; accessed 11-03-2022.

[28] J.S. Marshall and W.M. Palmer. The distribution of raindrops with sizes. *Journal of Meteorology*, 5:165–166, 1948.

[29] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, 2019.

[30] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *ICCV*, 2017.

[31] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

[32] Gerald Schaefer and Michal Stich. Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 472–480. International Society for Optics and Photonics, 2003.

[33] sjaak327. Simple trainer for gta v 14.1. [https://www.gta5-mods.com/scripts/simple-trainer-for-gtav](https://www.gta5-mods.com/scripts/simple-trainer-for-gtav). Online; accessed 11-03-2022.

[34] OpenIV Team. Openiv 4.0. [https://www.gta5-mods.com/tools/openiv](https://www.gta5-mods.com/tools/openiv). Online; accessed 11-03-2022.

[35] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *CVPR*, 2022.

[36] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, 2020.

[37] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, 2019.

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[39] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *CVPR*, 2017.

[40] Wenhan Yang, Robby T Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. Single image deraining: From model-based to data-driven and beyond. *IEEE TPAMI*, 43(11):4059–4077, 2020.

[41] Qiaosi Yi, Juncheng Li, Qinyan Dai, Faming Fang, Guixu Zhang, and Tieyong Zeng. Structure-preserving deraining with residue channel prior guidance. In *ICCV*, 2021.
[42] Shaodi You, Robby T Tan, Rei Kawakami, and Katsushi Ikeuchi. Adherent raindrop detection and removal in video. In CVPR, 2013.

[43] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Raindrop detection and removal from long range trajectories. In ACCV, 2014.

[44] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Adherent raindrop modeling, detection and removal in video. IEEE TPAMI, 38(9):1721–1733, 2015.

[45] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In CVPR, 2022.

[46] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In CVPR, 2021.

[47] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In CVPR, 2018.

[48] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. IEEE TCSVT, 30(11):3943–3956, 2019.

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In CVPR, 2018.