Adherent Raindrop Removal with Self-Supervised Attention Maps and Spatio-Temporal Generative Adversarial Networks

Stefano Alletto*, Casey Carlin*, Luca Rigazio*, Yasunori Ishii**, and Sotaro Tsukizawa**

*Panasonic Beta, Mountain View, CA
**Panasonic AI Solution Center, Osaka, Japan

Abstract

With the rapid increase of outdoor computer vision applications requiring robustness to adverse weather conditions such as automotive and robotics, the loss in image quality that is due to raindrops adherent to the camera lenses is becoming a major concern. In this paper we propose to remove raindrops and improve image quality in the spatio-temporal domain by leveraging the inherent robustness of adopting motion cues and the restorative capabilities of conditional generative adversarial networks. We first propose a competitive single-image baseline capable of estimating the raindrop locations in a self-supervised manner, and then use it to bootstrap our novel spatio-temporal architecture. This shows encouraging performance when compared to both state of the art single-image de-raining methods, and recent video-to-video translation approaches.

1. Introduction

Raindrops adherent to camera lenses can severely degrade image quality and represent a major challenge for vision systems where weather independent reliability is required such as advanced driver-assistance systems, autonomous driving or robotics. Due to their shape, raindrops reflect light rays from a wider area similarly to fish-eye lenses. Moreover, given how close they are to the camera sensor, they are often out of focus.

Despite the similarity of the adherent raindrop removal problem to more common tasks such as image denoising and bad weather visibility enhancing (e.g. fog, haze, rain streaks), removing raindrops from camera lenses has some key differences which often require specifically designed algorithms. Indeed, while denoising, de-fogging or de-hazing can be addressed as a global image transformation problem [33, 6, 27], de-raining requires some cues about where the degradation occurs in the image, often obtained in the form of raindrop detections or attention maps [21, 22, 23, 13]. One of the major issues faced when tackling the raindrop removal problem is the lack of available data, especially in the video domain. The most recent and promising methods require (*clean, rainy*) image pairs, which are inherently hard to obtain in real-world scenarios. Trying to tackle this issue, Qian et al. [21] recently proposed the DeRaindrop dataset, which while being an invaluable contribution, still suffers from its small size and occasional slight misalignments between *clean* and *rainy* images. Most critically, due to the complex requirements of the real world acquisition process, it only features a small variance of raindrop distributions and backgrounds. On the other hand, synthetic data has been used in the past [22], but under very limiting assumptions such as raindrops being sphere sections. As our first contribution, we design a synthetic raindrop generation method that uses computer graphics to superimpose

(a) Rainy (b) De-rained

Figure 1: Examples de-rained with our method. First row: Synthetic raindrops, second row: Real

Corresponding author: stefano.alletto@us.panasonic.com

photo-realistic raindrops over images and videos, simulating the complex interactions between light and droplets. This results in large quantities of data that we empirically show to not only allow experiments on otherwise unavailable datasets, but also improve the generalization capabilities over existing datasets such as [21].

Exploiting this new and large quantity of photo-realistic data, we address the adherent raindrop removal problem by designing a novel neural network that relies on spatio-temporal information for removing raindrops. We build our architecture in two steps: First we design a single-image baseline network that uses attention-based location cues to remove raindrops from the first few frames of a sequence. Then, our spatio-temporal model is initialized using these results and proceeds to de-rain the video on-line and in a spatially and temporally consistent manner thanks to the ability of generative adversarial networks to generate high quality and realistic imagery. Examples of our results can be seen in Fig. 1.

To summarize, in this paper we make the following contributions:

- We design a computer-graphics based method for superimposing photo-realistic raindrops to real-world images capable of generating much broader raindrop distributions than the ones available in current public datasets.
- We propose a competitive single-image de-raining baseline that relies on a novel self-supervised raindrop location estimation process, thus removing the limiting requirement of localization ground truth masks.
- We develop a novel spatio-temporal de-raining model that leverages the temporal robustness of explicitly encoding optical flow information and the image synthesis capabilities of generative adversarial networks. To the best of our knowledge, this is the first attempt at using generative adversarial networks for spatio-temporal raindrop removal.

2. Related Work

Recent methods tackling raindrop removal can be divided in two main categories: single-image and video based.

Single-image removal: Methods that only use single images for detection and removal [26, 18, 4, 25, 21] traditionally relied on hand-crafted image features. For example, Wu et al. [26] analyze color, shape and texture in an image to identify potential regions of interest, and use a saliency-driven approach to obtain the final localization map. Once raindrop locations are obtained and appropriately pruned of false positives via SVM classification, standard image inpainting techniques are used to reconstruct the underlying image. More recently, Eigen et al. [4] tackle the removal of raindrops and other small drop-like (e.g. mud) degradations using one of the first approaches based on a convolutional neural network. Using a fairly shallow model (3 Conv layers) and a standard MSE loss they remove drops but, due to the small capacity of their network and the known issues with MSE-based optimization, results tend to be blurry and cannot cope with different raindrops distributions, in particular bigger and more out of focus drops. The current state of the art results for single-image raindrop removal are obtained by the method by Qian et al. [21]. Not dissimilarly from the use of saliency in [26], they rely on the idea of guiding the removal through an attention-based mechanism. In particular, they use a convolutional LSTM to estimate raindrop locations in the spatial domain, and exploit this information in a GAN framework where both generator (the remover) and discriminator benefit from the attention-based localization. The main drawback of this approach is the way the attention maps are learned: The authors rely on supervised training and ground truth location masks, which are inherently hard to obtain. They automatically compute the location masks via image processing, which often results in poor quality ground truths. Differently from them, we propose a novel method for estimating the localization maps in a completely self-supervised way, both removing the need for hard to obtain ground truth and predicting better location maps thanks to not having to learn from low quality binary masks.

Spatio-temporal removal: On the other hand, researchers have also been using video information to tackle the raindrop detection and removal problem [22, 31, 30, 18], generally showing improved performance when relying on both spatial and temporal information. In particular, Nashashibi et al. [18] rely on blur detection and edge (or lack of thereof) estimation to propose candidate regions for raindrops, and validate them by using spatio-temporal correlation between ROIs. Yamashita et al. [29, 28] propose two similar methods for removing adherent raindrops using multiple images (stereo in [29], a monocular video sequence in [28]). In [29], the authors use stereo-based disparity and correlation for each pixel to detect raindrops and interpolate the corrupted zones between the two views, under the assumption that the parts of the image that are occluded by raindrops in one view will be visible in the other one. In [28], a similar process is applied but in the temporal domain using a monocular video sequence. Differently, You et al. [30] compute optical flow and dense trajectories and exploit motion inconsistencies caused by occluding raindrops. Detection is cast as a labeling problem in a Markov Random Field framework, where the motion consistency is used along with appearance and sharpness consistencies. Finally, removal is performed via trajectory-based video completion, ensuring both spatial and temporal consistency. The

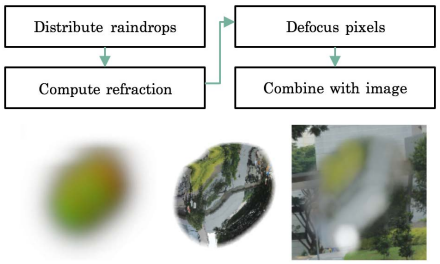

Figure 2: Stages of raindrop generation, from computing refraction direction, to color sampling, to defocus and compositing.

authors later extend their method in [31], where they show improved results by incorporating a deeper analysis of the blurriness of a raindrop and performing video completion dependent from the type of blur and thus information that can be recovered from the raindrop itself.

To the best of our knowledge, no methods relying on modern deep neural networks for spatio-temporal adherent raindrop removal currently exist. This is likely due to two major factors: The lack of video-based datasets of suitable size, and the significantly more demanding hardware requirements of video-based deep learning, for which only recent advancements in GPU technology enabled many applications.

3. Generating Photo-realistic Raindrops

To address the lack of data in the field, in this paper we propose to use computer graphics and leverage the research that has been put into screen-space post-processing techniques to augment large quantities of image data that are already publicly available. Specifically, for generating our raindrops we combine and modify two screen-space effects commonly used in games: Refraction and depth-of-field.

Screen-Space Refraction: The first stage in our raindrop synthesis process is generating the refraction direction buffer. This is an off-screen image which is used to determine, for each pixel of the screen, what portion of the background image should be used to render the raindrop refraction. Accurate modeling of the raindrop surface normal [23] and calculation of the vector of refracted light is possible. However, since we do not have a 3D scene to work with, and therefore cannot perfectly recreate refraction through a raindrop, our goal in this experiment was to reproduce raindrops with subjectively correct appearance. An individual raindrop is an informational texture, where the red and green channels encode the refraction vector, and the blue channel encodes a refraction multiplier, simulating the thickness of the drop.

When we have built our refraction direction buffer, we use a fragment shader to build a second buffer containing the refracted background image called the refraction color buffer. The following steps are executed for each pixel of the destination, in our case, the refraction color buffer:

- Sample the refraction direction buffer at the same position of our current pixel.
- Multiply the direction buffer's red and green channels by the blue channel, and add the resulting vector to the current pixel's position.
- Rescale the alpha channel of the direction buffer from 0.0-1.0 to a narrow range (say, 0.4-0.6) to make the sharp contour of the drop.
- At the new offset position, look up the color of the background and output it as the refracted color, along with the alpha value.

The resulting refraction color buffer already looks like raindrops on a transparent background, but it is virtually impossible for both a distant background and water on the lens to be completely in focus. Hence, we use this reflection color buffer as an input to a second process: Screen space depth of field.

Defocus Blur and Bokeh: Light passing through raindrops will be substantially out of focus if the background is in focus as the drops are virtually guaranteed to be behind the focal plane. Light traveling through a raindrop and hitting the image sensor in these conditions is defocused and spread out over an area referred to as the circle of confusion. Subjectively, low contrast regions do appear blurred, but bright high contrast regions expand into distinct circles referred to by photographers as "bokeh."

The analogous image processing operation is convolving the in-focus image with a disk. Ergo, no blur can accurately reproduce the defocus phenomena. To produce a defocused depth-of-field effect we use an approximation of this disk convolution called Bokeh Splatting [17, 14]. Note that, since we do not have a depth map for our backgrounds, we assume that light defocused by raindrops will have the same sized bokeh shape. To begin, when we build the refraction color buffer, we save the location and color of that pixel to a list of "rays of light," but only if a drop is present (the alpha channel is greater than zero.) Next, we transform our list of pixels into textured squares representing individual bokeh shapes with a geometry shader. This draws tens of thousands of transparent bokeh shapes over the background, each corresponding to a single ray. These shapes use the color we saved from the refraction color buffer, and we make the disks transparency inversely proportional to luminance of their color, so dark refractions have less influence on the final image. To properly reproduce the appearance of a raindrop obstructing the underlying background, these disks are stacked using alpha blending rather than the additive blending, as additive blending cannot remove light from scene.

Figure 3: Transition from in-focus to extreme defocus.

This has one significant drawback: Drawing order matters. In a real image, a defocused bright light in darkness should result in a single bright bokeh on a dark field. Using our technique that light's bokeh could get buried under disks representing dark pixels. The list of pixels is built in the fragment shader, so the order in which pixels are added to the list is nondeterministic. Our solution is to sort the list of pixels by their luminance, so that the disks are rendered dark-to-light, which accurately reproduces the "bloom" effect of bright light and makes the brightest bokeh the most clearly visible. Finally, to achieve a realistic range of focus, we add tunable parameters to our bokeh splatting effect so that defocus is defined by a parameter ranging from fully in focus at 0 to nearly invisibly out of focus at 1. Furthermore, bokeh size increases linearly with defocus while bokeh opacity decreases exponentially with defocus (see Fig. 3).

4. Proposed Method

After introducing our approach for generating high-fidelity synthetic raindrops, in this section we first present our baseline single-image architecture, and then discuss how to extend it to account for temporal information in the raindrop removal pipeline.

4.1. Single-image Raindrop Removal

Our single-image baseline network is built on two major components: A raindrop location estimator and a remover. The location estimator relies on the idea that a rainy image R can be expressed as $R = C + A$ where C is the clean image and A is an additive map that represents the raindrops. Differently from Qian et al. [21], this allows the location map to be learned in a self-supervised manner that does not require ground truth locations (i.e. binary raindrop masks). To estimate this location map our architecture is based on the popular encoder-decoder paradigm [10]. In particular, two strided convolutional layers encode the input image and are followed by a feature extractor that uses residual blocks [7] to propagate information. The final output is obtained by decoding these features with two transposed convolution layers. To train the location estimation without location ground truth, the following objective function is minimized:

$$L_A(R, R') = ||R - R'||_2^2, \; with \; R' = \frac{C + G_A(R)}{2} \quad (1)$$

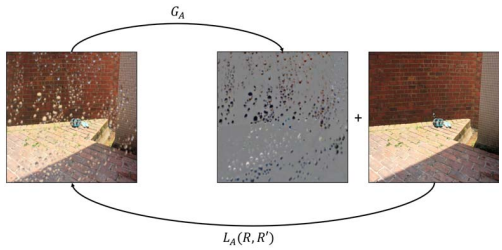

Figure 4: Self-supervised additive map for our location estimation process.

where R' is the reconstructed rainy image obtained by summing the additive location map obtained from the generator network $G_A(R)$ to the clean image. The idea behind this loss function is to exploit the cyclic relationship between rainy and clean images to avoid direct supervision in term of raindrop locations (see Fig. 4 for a graphical example of this process). Since our goal is to learn an additive map A that produces plausible results when added to the clean image, we add a second adversarial term to the loss function [5]. Note that, in place of the popular VGG Perceptual loss [12], in Equation 1 a L2 loss is chosen. This is because the use of the VGG perceptual loss is not suitable to evaluate rainy images due to the lack of raindrops on the images used to train said VGG network, resulting in its features being unreliable. On the other hand, the L2 loss does not suffer from this issue and is thus more suitable in this context.

The location map estimated by this first step is then fed to the second part of the network along with the rainy image R through channel-wise concatenation. Since they both output a three channel image, the raindrop remover shares the same architecture of the location estimator with the exception of the number of input channels which is doubled.

This second part of the network is also trained using a combination of adversarial and content losses. As a content loss we adopt the VGG Perceptual loss [12]. The additive map estimator and the remover are jointly trained optimizing this final objective function:

$$L(R, R', C, C') = \alpha L_A(R, R') + \alpha L_{VGG}(C, C') + \beta L_{Adv}(R, R') + \beta L_{Adv}(C, C') \quad (2)$$

where L_{VGG} is the VGG perceptual loss and L_{Adv} are the two adversarial components. The two weights α and β are empirically set to $10, 1$ respectively.

4.2. Spatio-temporal Raindrop Removal

In this section we present our solution for extending the baseline architecture with temporal information. Following [24], we model the temporal dynamics by making a Markov assumption, where each video frame is generated sequentially based on the previous T frames. Formally,

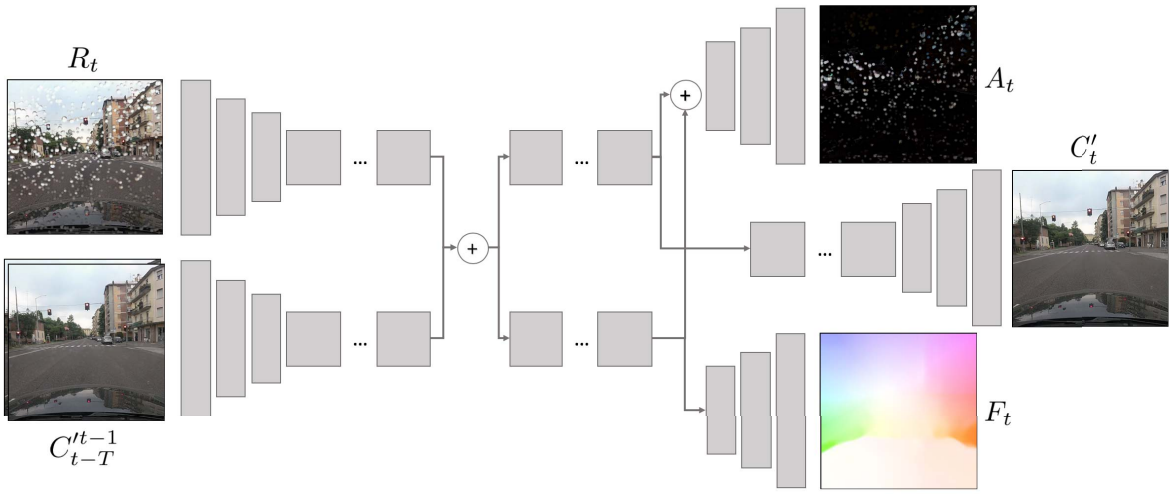

Figure 5: Architecture of our spatio-temporal de-raining model.

this allows us to model the current de-rained frame C'_t as $C'_t = \mathcal{F}(C'^{t-1}_{t-T}, R_t)$ where $\mathcal{F}$ is the video de-raining function implemented by our generator network. The first $T-1$ frames are de-rained using the single-image architecture described in Section 4.1 (we empirically fix $T = 3$). Our preliminary experiments showed no significant gain in performance when increasing the temporal width of the method further (while significantly increasing training time).

Traditional computer vision methods [22, 31, 30] often relied on optical flow features to detect raindrops in images. For this reason, we explicitly include flow information in our generator network and use it to better localize raindrops in the current frame. Joint learning of optical flow is shown to be beneficial to several tasks [19, 3, 34], hence instead of just having optical flow as an additional input, we jointly learn optical flow inside our generator. Ground-truth optical flow is traditionally hard to obtain, and for this specific task a coarse flow is sufficient to guide the learning of temporal features. For these reasons, we follow recent methods estimating flow in a self-supervised manner [24, 1, 11] by optimizing a reconstruction loss on the warping between two adjacent frames. Furthermore, similarly to [24] we also drive the learning of the optical flow through an adversarial objective where the *real* flow between two adjacent frames is obtained using a pretrained FlowNet 2 [9]. We refer to Figure 5 for a graphical explanation of our spatio-temporal generator architecture: First, two encoder branches extract features from the current rainy image R_t and from the previous $T-1$ de-rained images C'^{t-1}_{t-T}. The resulting feature maps are concatenated channel-wise and fed into two new branches: Additive map and optical flow estimation. The optical flow branch is followed by a decoder that outputs the flow between frames t and $t-1$. The additive map estimation feature maps are concatenated channel-wise with the flow features before being decoded into the actual additive map, explicitly injecting strong temporal information in the location estimation. The de-raining branch then proceeds to use the additive map estimation features for producing C'_t. We refer to the additional material for a layer-by-layer breakdown of our generator architecture and details on the discriminator networks used for the adversarial objectives.

To train this spatio-temporal model, we employ a combination of different loss functions.

Optical flow: As stated above, we learn optical flow in a self-supervised manner. Being F_t the flow between frames $t-n$ and t predicted by our flow branch, $\tilde{F}_t$ the flow between the same inputs computed using [9] and W_t the results of warping C_{t-n} with F_t, we train our flow branch by optimizing a combination of the following objectives:

$$L_W = ||C_t - W_t||_2^2 \tag{3}$$

$$L_{F_{Adv}} = E[logD_F(\tilde{F}_t)] + E[log(1 - D_F(F_t))] \tag{4}$$

where D_F is our flow discriminator. To learn more robust temporal features, we progressively increase the size of the temporal interval considered, with $n = 1, \ldots, T$.

Additive location map: following the same insight used for our single-image baseline, for the additive map estimation branch we use Equation 1 as content loss and further add the adversarial term:

$$L_{A_{Adv}} = E[logD_I(R)] + E[log(1 - D_I(R'))] \tag{5}$$

with D_I being our image discriminator.

De-raining: To train our final de-raining branch, we optimize the VGG perceptual loss [12] as content loss and an additional adversarial objective on the de-rained output:

(a) Rainy image (b) Additive map (c) De-rained result

Figure 6: Example of additive map computed by our self-supervised location estimation branch.

$$L_{VGG} = \sum_j \frac{1}{c_j h_h w_j} ||\phi_j(C_t) - \phi_j(C'_t)||_2^2 \quad (6)$$

$$L_{C'_{Adv}} = E[log D_I(\tilde{C}_t)] + E[log(1 - D_I(C'_t))] \quad (7)$$

where $c_j h_h w_j$ are the shapes of the $j-th$ feature map ϕ_j of the pre-trained VGG network and D_I is the same image discriminator used in Eq. 5.

5. Experimental Evaluation

Obtaining real-world video sequences with *clean, rainy* pairs is inherently hard, requiring either a very complex hardware setup or severely constraining the type of scenes acquired. While such a dataset exists for single images [21], large scale video-based raindrop removal is still largely unexplored. For this reason, we choose to rely on a photo-realistic synthetic data generation process to generate training and testing data. Using the technique described in Section 3, we augment videos from the DR(eye)VE dataset [20] by adding photo-realistic raindrops. The reasons for choosing the DR(eye)VE dataset are twofold: First, it currently is the biggest automotive (arguably the prime setting for raindrop removal) dataset providing high-quality videos of driving, featuring more than 500,000 frames; second, its sequences are divided by weather conditions which allows us to synthetically augment *cloudy* sequences obtaining very realistic results. We select 11 sequences out of the DR(eye)VE dataset for training and divide them into 2750 clips 30 frames long, each one with a different synthetic raindrop distribution. For testing, we use 2 more sequences resulting in 500 clips of 30 seconds each.

5.1. Image Quality Evaluation

We begin our evaluation by analyzing the performance of our de-raining approach using image quality assessment metrics such as PSNR and SSIM. While these metrics do not account for any temporal aspect, they are commonly used to evaluate image quality and allow us to provide context for our baseline performance by comparing it to the current state of the art for image de-raining.

Table 1: Image quality assessment evaluation on the DeRaindrop dataset. *Baseline* indicates the results of our single-image network.

	PSNR	SSIM
Rainy	24.20	0.875
Eigen13 [4]	28.59	0.673
Pix2Pix [10]	30.14	0.830
Qian et al. [21]	31.57	0.902
Ours	**31.94**	**0.945**

DeRaindrop Dataset: The first experiment is performed on the DeRaindrop dataset [21]. Following the experimental setup of Qian et al., we use the *test_a* test set for our evaluation. Since the dataset does not provide video sequences but only single images, we evaluate the performance of our single-image baseline and compare it to the current state of the art.

In Table 1 we report the results of evaluating raindrop removal on the DeRaindrop dataset where we compare our baseline network against the current state of the art for single images, the method by Qian et al. To provide a better context, we also report the results of Eigen13 [4] and Pix2Pix [10]. It can be seen that our baseline network outperforms recent single-image de-raining methods on both PSNR and SSIM metrics, with a particularly large gap in the second one. Compared to Qian et al. [21], we argue that our self-supervised location estimation is capable of producing better attention maps. Indeed, Qian et al. rely on a supervised process that is trained with automatically generated ground truths obtained by applying image processing techniques such as thresholding and morphology to the difference between *rainy* and *clean* images, often resulting in imprecise ground truth masks. Slight misalignments between the image pairs, objects moving in the scene or parts of the image not correctly filtered by the thresholding step contribute to reduce the quality of the ground truth, hindering their learning process. On the other hand, thanks to the self-supervised location estimation described in Section 4.1, we do not suffer from this issue and thus we obtain better de-raining performance. Figure 7 reports qualitative examples of this evaluation.

DR(eye)VE dataset: Furthermore, we evaluate the de-raining performance on the DR(eye)VE dataset augmented with synthetic raindrops. Here, we measure the performance of our baseline method and of the full spatio-temporal approach and compare them against the current state of the art for single-image de-raindrop [21]. To establish a spatio-temporal baseline we compare against the recent video-to-video translation method by Wang et al., Vid2Vid [24]. Furthermore, due to the lack of spatio-temporal raindrop removal methods, we evaluate the current state of the art for rain-streak removal [15]. To provide

Figure 7: Qualitative results of de-raining on the DeRaindrop test-set

a fair comparison, these methods are re-trained on the same synthetic training-set used for the training of our spatio-temporal approach. Note that differently from [21], we train their method using ground-truth location masks obtained from our synthetic generation instead of computing them by subtracting rainy and clean frames, a procedure subject to error and noise. As for [15], the same two-stages training approach proposed in the paper is used. Despite holding the current state of the art for spatio-temporal raindrop removal, the methods by You et al. [30, 31] are not included in this evaluation. While obtaining promising results on their very constrained data, these methods are not suitable for real-world raindrop removal and cannot be successfully applied to sequences from the DR(eye)VE dataset. We refer the reader to the supplementary material for a broader discussion on the reason behind the failure of [30, 31].

As Table 2 shows (*Rainy* column) the synthetically augmented DR(eye)VE dataset suffers a more severe degradation compared to the DeRaindrop dataset (PSNR 22.07 vs 24.20, SSIM 0.725 vs 0.875). Furthermore, thanks to the ability to generate different raindrop distributions, a much broader variety of raindrop patterns, sizes and types of focus is present compared to the DeRaindrop dataset where all the images have been acquired by spraying water on a glass. According to both PSNR and SSIM, our spatio-temporal method results in the best image quality by a large margin, followed by our single-image baseline. Despite some similarities between the two tasks and the excellent results it achieves on video rain-streak removal, the poor performance of the method by Liu et al [15] is due to the significantly different temporal dynamics. In fact, video rain-streak removal relies on the assumption of fast-moving rain-streaks and tackles the problem in a way which is more similar to video denoising, which is unsuitable for raindrop removal.

Table 2: Image quality assessment evaluation on the DeRaindrop dataset. *Baseline* indicates the results of our single-image network.

	PSNR	SSIM
Rainy	22.07	0.725
Qian et al. [21]	26.29	0.961
Baseline	29.13	0.967
Vid2Vid [24]	28.64	0.959
Liu et al [15]	23.26	0.938
Ours	**32.44**	**0.974**

5.2. Temporal Consistency

To evaluate the quality of the de-rained video sequences taking into account the temporal aspects of videos, we rely on the same experimental pipeline proposed by Wang et al. [24]. This evaluation procedure is motivated by the two use-cases of a video-based de-raining method: Either the results are destined to a human viewer (e.g. de-raining a back-facing camera on a car for safety or parking), or used for further processing by another network (e.g. to improve object detection while driving in a rainy day). To account for these use-cases, the two metrics used are:

- **Human Preference Score (HPS)** evaluates the visual

Table 3: Evaluation with spatio-temporal metrics.

	Qian et al. [21]	Baseline	Vid2Vid [24]	Ours
HPS	0.12	0.08	0.05	**0.75**
FID	3.25	4.24	8.83	**1.51**

quality of the outputs. Results from the different methods evaluated are shown to people and they are asked to express a preference indicating which one has the best quality both in terms of de-raining performance, overall video quality and temporal consistency of the results. Following [24], we employ a pool of 10 different subjects each one viewing 15 randomized sequences from the de-rained test-set.

- **Fréchet Inception Distance (FID)[8]** is used to measure both visual quality and temporal consistency of a video. Using a pre-trained network as feature extractor, spatio-temporal features are computed from both the clean and the de-rained videos and their statistics are compared. More formally, being $\mu, \tilde{\mu}$ the mean of the computed feature maps for respectively the clean and de-rained videos and $\Sigma, \tilde{\Sigma}$ their covariance matrices, the FID is computed as $||\mu - \tilde{\mu}||^2 + Tr(\Sigma + \tilde{\Sigma} - 2\sqrt{\Sigma\tilde{\Sigma}})$. Here, the popular I3D network [2] is used as inception network (layer *maxpool3d_5a_2x2*).

Table 3 reports the results of this evaluation. Surprisingly, despite both metrics accounting for the temporal consistency of the resulting videos, Vid2Vid performs the worst. The poor results of this temporal baseline are due to the fact that while producing videos that are overall temporally consistent, Vid2Vid often fails to remove raindrops and introduces artifacts in their place. This confirms the observation that raindrop removal requires location information and cannot directly be treated as a image or video translation problem. Besides this, the spatio-temporal evaluation confirms that our method is largely preferred by human observers and also outputs videos that result in activation maps much more similar to the original clean videos.

5.3. Synthetic Data for Improving Domain Specific De-Raining

Here we discuss a second application of the synthetic raindrop generation approach described in Section 3: Domain specific data augmentation. We propose to evaluate the improvement in terms face reconstruction and face detection under real-world heavy raindrops and use our single-image baseline network as de-raining method and the DeRaindrop dataset [21] as data baseline. Note that due to the specific setup in which the dataset has been acquired, no human beings nor faces are included in the training data. We train our network twice with the same training setup except that the second time we only use half of the DeRaindrop dataset, and use as second half a set of images

Figure 8: From left to right: Rainy image, de-rained without domain specific images, de-rained with 50% synthetically augmented CelebA images.

of faces with synthetic rain on them, randomly sampled from the popular CelebA dataset [16]. Using an acquisition rig similar to the one used in [21], we spray water and collect video sequences with real-world raindrops. In each of the collected videos, one and only one face is always present, resulting in an obvious ground truth for detection recall. De-raining the videos with the two networks and applying the popular MTCNN face detector [32], we show an increase in detection recall from the 0.65 of the network trained only on DeRaindrop to 0.76 when using synthetic data augmentation, compared to a recall for the rainy image of 0.56. This confirms the ability of our synthetic generation method to produce photo-realistic raindrops and shows how augmenting a generic dataset with domain specific rainy images can significantly improve the performance. Figure 8 shows qualitative examples of face reconstruction using the two networks. It can be seen how despite neither network achieves perfect de-raining in such challenging conditions, augmenting the training set with synthetic raindrops over faces greatly improves the face reconstruction.

6. Conclusions

In this paper we presented the first deep-learning based method for adherent raindrop removal using video information. Thanks to our baseline's capability of estimating raindrop locations in a self-supervised manner, we remove the requirement for binary ground truth location masks. This not only produces good results with the current datasets, but will also be beneficial in the future when switching to new and real-world datasets. Our spatio-temporal architecture shows results that are often preferred to existing methods both by human observer and by inception scores, validating our pipeline. To foster future research on the topic, we plan to release both the source code of our method and the computer-graphics based raindrop generation tool.

References

[1] S. Alletto, D. Abati, S. Calderara, L. Rigazio, and R. Cucchiara. Self-supervised optical flow estimation by projective bootstrap. *IEEE Transactions on Intelligent Transportation Systems*, 2018. 5

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017. 8

[3] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 686–695. IEEE, 2017. 5

[4] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE international conference on computer vision*, pages 633–640, 2013. 2, 6

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 4

[6] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014. 1

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[8] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 8

[9] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, page 6, 2017. 5

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 4, 6

[11] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016. 5

[12] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 4, 5

[13] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara. Rainy weather recognition from in-vehicle camera images for driver assistance. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 205–210. IEEE, 2005. 1

[14] S. Lee, G. J. Kim, and S. Choi. Real-time depth-of-field rendering using point splatting on per-pixel layers. In *Computer Graphics Forum*, volume 27, pages 1955–1962. Wiley Online Library, 2008. 3

[15] J. Liu, W. Yang, S. Yang, and Z. Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 6, 7

[16] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 8

[17] L. McIntosh, B. E. Riecke, and S. DiPaola. Efficiently simulating the bokeh of polygonal apertures in a post-process depth of field shader. In *Computer Graphics Forum*, volume 31, pages 1810–1822. Wiley Online Library, 2012. 3

[18] F. Nashashibi, R. de Charrette, and A. Lia. Detection of unfocused raindrops on a windscreen using low level image processing. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pages 1410–1415. IEEE, 2010. 2

[19] J. Y.-H. Ng, J. Choi, J. Neumann, and L. S. Davis. Actionflownet: Learning motion representation for action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1616–1624. IEEE, 2018. 5

[20] A. Palazzi, D. Abati, S. Calderara, F. Solera, and R. Cucchiara. Predicting the driver's focus of attention: the dr(eye)ve project. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 6

[21] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2018. 1, 2, 4, 6, 7, 8

[22] M. Roser and A. Geiger. Video-based raindrop detection for improved image registration. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 570–577. IEEE, 2009. 1, 2, 5

[23] M. Roser, J. Kurz, and A. Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *Asian Conference on Computer Vision*, pages 235–244. Springer, 2010. 1, 3

[24] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 4, 5, 6, 7, 8

[25] D. D. Webster and T. P. Breckon. Improved raindrop detection using combined shape and saliency descriptors with scene context isolation. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4376–4380. IEEE, 2015. 2

[26] Q. Wu, W. Zhang, and B. V. Kumar. Raindrop detection and removal using salient visual features. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 941–944. IEEE, 2012. 2

[27] Y. Xu, J. Wen, L. Fei, and Z. Zhang. Review of video and image defogging algorithms and related studies on image restoration and enhancement. *Ieee Access*, 4:165–188, 2016. 1

[28] A. Yamashita, I. Fukuchi, and T. Kaneko. Noises removal from image sequences acquired with moving camera by estimating camera motion from spatio-temporal information. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ*

International Conference on, pages 3794–3801. IEEE, 2009. 2

[29] A. Yamashita, Y. Tanaka, and T. Kaneko. Removal of adherent waterdrops from images acquired with stereo camera. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 400–405. IEEE, 2005. 2

[30] S. You, R. Tan, K. Rei, and K. Ikeuchi. Adherent raindrop modeling, detection and removal in video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1035–1042, 2013. 2, 5, 7

[31] 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, 3, 5, 7

[32] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 8

[33] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 1

[34] Y. Zou, Z. Luo, and J.-B. Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision*, pages 38–55. Springer, 2018. 5