Let's See Clearly: Contaminant Artifact Removal for Moving Cameras

Xiaoyu Li[1] Bo Zhang[2] Jing Liao[3] Pedro V. Sander[1]

[1]The Hong Kong University of Science and Technology
[2]Microsoft Research Asia [3]City University of Hong Kong

Figure 1: Contaminant removal for video frames captured by dirty lens camera. Contaminants on the lens, *e.g.*, dust, dirt and moisture, cause spatially variant photography artifacts (first row). Our method restores these contaminant artifacts by leveraging the spatio-temporal consistency from multiple frames (second row). Please refer to our *supplementary material* for video results.

Abstract

Contaminants such as dust, dirt and moisture adhering to the camera lens can greatly affect the quality and clarity of the resulting image or video. In this paper, we propose a video restoration method to automatically remove these contaminants and produce a clean video. Our approach first seeks to detect attention maps that indicate the regions that need to be restored. In order to leverage the corresponding clean pixels from adjacent frames, we propose a flow completion module to hallucinate the flow of the background scene to the attention regions degraded by the contaminants. Guided by the attention maps and completed flows, we propose a recurrent technique to restore the input frame by fetching clean pixels from adjacent frames. Finally, a multi-frame processing stage is used to further process the entire video sequence in order to enforce temporal consistency. The entire network is trained on a synthetic dataset that approximates the physical lighting properties of contaminant artifacts. This new dataset and our novel framework lead to our method that is able to address different contaminants and outperforms competitive restoration approaches both qualitatively and quantitatively.

1. Introduction

As imaging devices have become ubiquitous, the ability to take photographs and videos everywhere and anytime has increased significantly. Mobile cameras, action cameras, surveillance cameras, and the sensors of autonomous driving cars are often exposed to the harsh environment in which contaminants will cause deterioration of image quality. Figure 1 shows some examples of dirty lens artifacts, where the visibility of the scene radiance is partially affected by the absorption and reflection of the contaminants along the light path [17]. These undesired artifacts are not only aesthetically disturbing, but also bring difficulty for subsequent computer vision tasks. Although one can physically clean the lens sporadically, doing this frequently is by no means a handy solution and sometimes infeasible for real-time situations.

Since the contaminants adhere to the lens surface and thereby lie out of focus, their imaging effect can be modeled by a low-frequency light modulation [17], *i.e.*, the dirty lens artifacts appear diffuse and semi-transparent with the high-frequency textures of the background scene partially preserved. This makes image or video inpainting methods [6, 55, 19, 46, 54] inadequate for our task as they completely ignore the underlying structures and the hallucinated content. Albeit visually plausible, they may deviate significantly from the real scene. Furthermore, these works assume the completion regions are prescribed by a user-given mask, whereas our task automatically identifies the degradation region, which is inferred from camera motion.

This work is more closely related to single image artifact

removal for raindrops [11, 18, 32, 33], reflection [3, 12, 45, 57] and thin obstructions [29]. These works typically adopt learning approaches, utilizing the spatial prior of natural images to restore the spatial variant degradation. Nonetheless, the artifact removal for a single image is inherently ill-posed, and the learned spatial prior often fails to generalize to scenes with domain gaps. To solve this, multi-frame approaches [2, 28, 47] decouple the occlusion and background scene by leveraging the fact that there exists motion difference between the two layers, and the pixels occluded in one frame are likely to be revealed in other frames. In particular, the recent learning-based approach [28] achieves remarkable quality in removing unwanted reflection and obstructions. However, this method only considers a fixed number of adjacent frames as input, which should be varied depending on the magnitude of the motion and obstruction size, whereas our recurrent scheme supports an arbitrary number of adjacent frames for restoration until convergence.

In this work, we propose a learning-based framework tailored for removing the contaminant artifacts of moving cameras. To this end, we first train the network to automatically spot the contaminant artifacts which are usually prominent in the flow maps of a video with a moving camera. As opposed to layer decomposition, we only focus on the background motion, of which the degraded region by the contaminants is hallucinated and softly blended by our flow completion network, depending on how much of the background is occluded.

In order to leverage information spanning an arbitrary number of frames, the restoration for each frame is recurrent. That is, to restore one frame, we recurrently feed the adjacent frames one by one. Guided by the completed background flow, the pixels within the artifact region can be progressively restored by referring to the corresponding clean pixels from other frames. So far the restoration operates on each input frame individually, utilizing only the information of their adjacent frames. To produce the temporally consistent result for the whole video, we propose another multi-frame processing stage, in which we follow the same pipeline again but this time using the restored results from the last recurrent stage as input.

We train the entire framework in a supervised fashion. To achieve this, we propose a synthetic dataset that follows the imaging physics of contaminant artifacts. Extensive experiments prove that the proposed model can generalize to real dirty lens videos (as shown in Figure 1), outperforming strong baselines both qualitatively and quantitatively. Our contributions can be summarized as follows:

- We propose the first deep learning approach to specifically address the contaminant artifacts for moving cameras. The proposed method performs better than general restoration methods on real videos.
- A physics-inspired synthetic dataset is proposed to mimic real contaminant artifacts.
- We propose a flow completion module to effectively hallucinate the background motion given the partially visible structure clue within the degraded region.
- The proposed recurrent scheme not only helps leverage the multiple adjacent frame information to restore individual frames but can also be reused to refine such frame-wise output and ultimately yield temporally coherent video results.

2. Related Work

Camera artifact removal. The pioneer work [17] proposes a physics-based method to remove the dirty lens artifact, yet the point-wise restoration they propose cannot handle complex artifacts. Following works, on the other hand, merely focus on the contaminant detection [1, 8, 43, 49], but do not study how to give a clean image with contaminant removal. Indeed, the artifacts region can be restored with a follow-up content completion [29, 25, 54, 30], yet this will totally neglect the underlying structure within the degraded region. In comparison, we jointly consider the artifact localization and restoration in a single framework that utilizes the partially visible structures as much as possible. Notably, lens flare or glare is another common lens artifact that plagues the photography in which the scene is also partially obstructed. Nonetheless, existing solutions [4, 34, 44] focus on single image restoration which is inherently ill-posed, whereas our method explicitly utilizes multi-frame information captured by moving cameras.

Adherent raindrop removal. A number of methods have been proposed to address raindrops attached to glass windows, windscreens, *et al*., mostly for single image [11, 32, 18, 33]. Several methods have been proposed to remove raindrops from video [37, 48, 51, 52, 53]. However, after detecting the raindrops using the spatial-temporal information, these methods rely on off-the-shelf video inpainting techniques to restore the video, which does not fully utilize the partially visible details within the raindrops. Besides, both the raindrop detection and restoration are optimized separately. Recently, Liu *et al*. [28] present a learning approach for removing unwanted obstructions which also include semi-transparent raindrops. Instead of formulating the problem as layer decomposition, we only consider the scene motion and use a recurrent scheme to consider information from an arbitrary number of adjacent frames. Besides, our method does not require time-consuming online optimization as post-processing for handling real-world sequences.

Video-based restoration. Video-based restoration such as video inpainting, video denoising and video deblur uti-

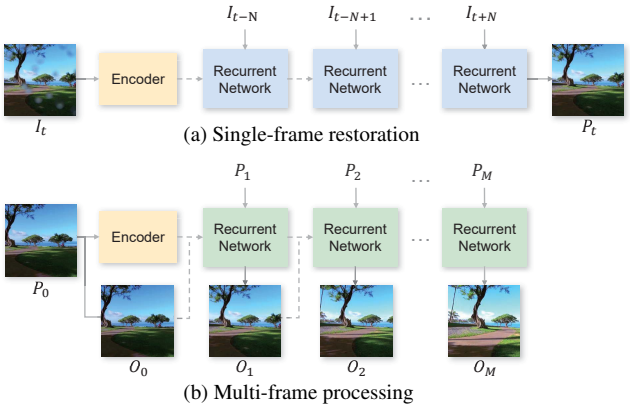


Figure 2: Overview of our two-stage recurrent network for contaminant removal. (a) In the single-frame restoration stage, frame I_t is progressively restored by feeding its adjacent frames one by one. (b) The multi-frame processing stage takes the frame-wise results $\{P_t\}$ as input and recurrently processes them to produce a temporal coherent result $\{O_t\}$.

lizes spatial-temporal information for restoration. One typical application is rain streak removal. While prior approaches rely on hand-crafted features [5, 13, 14, 15, 16, 38, 56], recent prevalent methods resort to deep neural networks [7, 24]. Although some of them also employ recurrent scheme [26, 27, 50] to leverage temporal information, additional modules like flow completion, multi-frame processing have been uniquely considered for our problem.

3. Method

Figure 2 illustrates the proposed two-stage recurrent framework. Given an input frame I_t suffering from contaminant artifacts, we first gradually restore the degraded region by iteratively utilizing the adjacent frames $I_{t-N\leq k\leq t+N}$ that may reveal some new clean pixels under the camera motion. This is achieved by aligning the frames with the hallucinated flow. This way, we obtain frame-wise intermediate outputs $\{P_t\}$, which are further fed into the multi-frame processing stage and yield the frames $\{O_t\}$ that consider the temporal consistency relative to the outputs at an earlier time. Next, we introduce a synthetic dataset that realistically emulates the contaminant artifacts for training (Section 3.1). Then we elaborate on the details of the single-frame restoration (Section 3.2) and the multi-frame processing (Section 3.3), respectively.

3.1. Dataset Construction

It is challenging to obtain large quantities of well-aligned video pairs of real scenes, so we synthesize a training dataset that covers realistic and diverse contaminant artifacts. To this end, we render images following the physics model [17] about how the contaminants affect the image irradiance. Specifically, we use Blender [10] for the rendering. We collect a large number of moving camera videos as source frames, which serve as the scene textures representing our scene. Between the scene and the camera, we added a glass layer with an index of refraction set to 1 to simulate the contaminants. We model the contaminants with randomly deformed particles adhered to the glass layer in order to mimic diverse shapes so that our method can handle various real world situations. The material of the contaminants is a mixture of different shaders: the glass shader adds some refraction, the emission shader contributes some radiance so as to emulate the scattering due to the lens dirt, and the transparent shader models the light attenuation caused by the contaminants. By stochastically varying the parameters of these shaders, we are able to simulate the effect of common contaminant materials. For sequential frames in the video, the parameters for generating random contaminants are the same for consistency. But for frames from different video sequences, we use random parameters and synthesize them independently. Figure 3 shows examples of our rendered images. The synthetic samples closely mimic the real contaminated images and have a large variation to cover common situations in real photos.

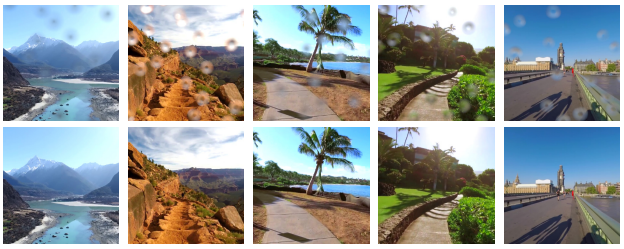
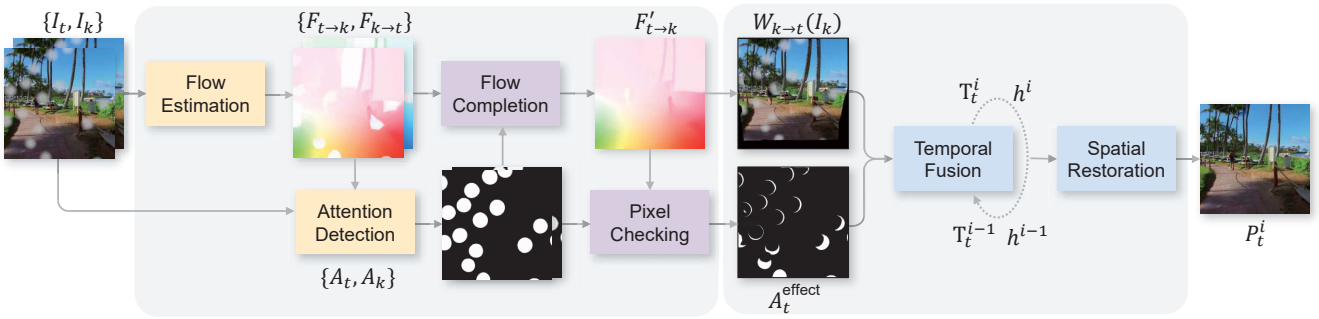


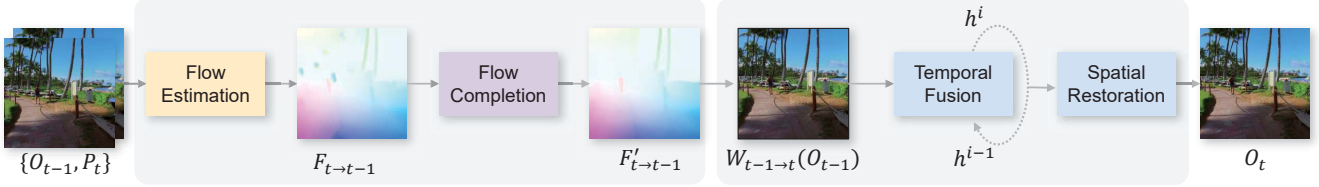
Figure 3: Samples from our synthetic dataset. The first row shows images with contaminant artifacts and the second row shows the corresponding ground truth images.

3.2. Single-frame Restoration

In this stage, we aim to remove the artifacts from frame I_t by recurrently referring to the adjacent frames $\{I_k\}$. The procedures for the single-frame restoration are depicted in Figure 4 (a). We first estimate the bidirectional flows $\{F_{t\to k}, F_{k\to t}\}$ between the two frames $\{I_t, I_k\}$ and detect the attention maps $\{A_t, A_k\}$ to localize the degraded region based on the flows. Guided by the attention maps, we complete the background motion by a flow completion module so that we can warp the reference frame towards the input accordingly. The pixel checking module validates whether the pixel in the warped reference $\mathcal{W}_{k\to t}(I_k)$ can be used to restore the corresponding contaminated pixels in I_t. Next, a recurrent temporal fusion module updates the restored result from T_t^{i-1} to T_t^i by leveraging the effective clean pixels from $\mathcal{W}_{k\to t}(I_k)$ as well as the recurrent hidden state h^{i-1} that comes from the last iteration $(i-1)$. Finally, the hid-



(a) Single-frame restoration



(b) Multi-frame processing

Figure 4: Overview of the recurrent pipeline for (a) single-frame restoration and (b) multi-frame processing.

den representation is decoded to the image output P_t^i with a spatial restoration module.

Flow estimation & attention detection. We first estimate the optical flows $\{F_{t\to k}, F_{k\to t}\}$ between the input I_t and its adjacent frame I_k using the off-the-shelf RAFT model [42]. As shown in Figure 4 (a), the contaminants become prominent in the estimated flow. Therefore we could utilize it to help predict the attention map that indicates the degraded region. Specifically, we adopt an U-Net [36] to estimate the attention map A_t for I_t using the information of flow $F_{t\to k}$ assisted with the frame I_t. The network is trained with a binary cross entropy (BCE) loss between A_t and the ground truth A_t^{gt}:

$$\mathcal{L}_{\text{att}} = -\frac{1}{HW}\sum_{p} A_t^{\text{gt}} \log A_t + (1 - A_t^{\text{gt}}) \log(1 - A_t) \quad (1)$$

where p indexes the pixels and HW is the image resolution. Similarly, the attention map A_k of I_k can be estimated using the inverse flow $F_{k\to t}$ and frame I_k. Here, a higher value for A_t indicates a higher possibility of being occluded by the contaminants.

Flow completion & pixel checking. Due to the motion difference between the fast-moving background and relatively static contaminants, the pixels degraded in one frame could be revealed in adjacent frames. Hence, we could utilize this fact to restore the video frames. In order to leverage the corresponding clean pixels from the reference frames, we need to hallucinate the flow of the background scene that is unreliably estimated within the degraded region. To this end, we propose a flow completion module whose effect is shown in Figure 5: the pixels within the degraded region can

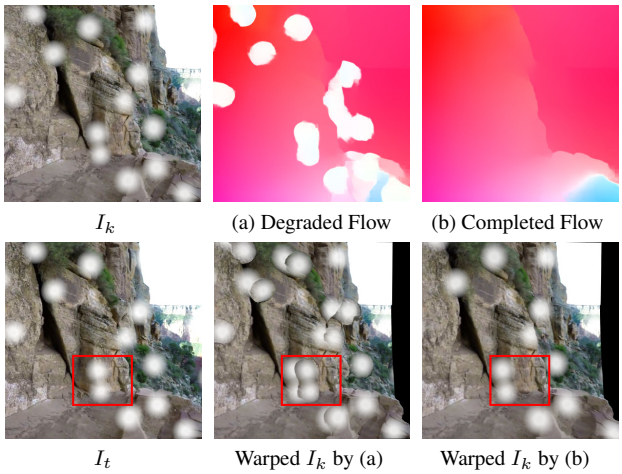


Figure 5: The effect of our flow completion module. The middle of the marked region is still degraded with the warping $\mathcal{W}_{k\to t}(I_k)$ using the degraded flow but corrected filled in with the completed flow.

only be correctly filled with clean pixels according to the completed flow. Note that our flow completion module not only corrects degraded regions but automatically resolves the flow errors at both clean and degraded regions.

Notably, the estimated flow within the degraded region may not be unreliable all the time in that the flow module may leverage partially visible structures and induce correct flow estimation. As such, the flow completion should flexibly hallucinate the flow depending on how much the background structures are visible. Therefore, we propose a feature fusion layer that dynamically fuses the features of two branches: the input and the flow hallucinated from the

scratch, according to a fusion weight map α:

$$f_{\text{out}} = f_{\text{in}} \odot \alpha + \mathcal{G}_l(f_{\text{in}}) \odot (1 - \alpha) \tag{2}$$

where f_{in} and f_{out} are the input and output features respectively, $\mathcal{G}_l$ denotes the mapping function, and $\odot$ is the Hadamard operator. The fusion weight map is obtained from the input feature map with layers $\mathcal{G}_\alpha$ followed by a sigmoid squashing function:

$$\alpha = \text{sigmoid}(\mathcal{G}_\alpha(f_{\text{in}})) \tag{3}$$

Larger values in α denote that the degraded pixel is more visible, so the flow completion is more confident to the flow input and resorts less to the hallucinated values.

The flow completion module has an autoencoder architecture whose encoder consists of six such fusion layers, whereas we place four fusion layers with dilation at the bottleneck. The decoder, on the other hand, adopts the upsampling module as [42], *i.e.*, learning a weighted combination of a local 3×3 neighbors at coarse resolution, which we find beneficial to produce a sharper flow estimation compared with the traditional bilinear upsampling. We enforce the flow completion with the $\mathcal{L}_1$ loss between the output $F'_{t\to k}$ and the ground truth $F^{\text{gt}}_{t\to k}$,

$$\mathcal{L}_{\text{flow}} = \left\| F'_{t\to k} - F^{\text{gt}}_{t\to k} \right\|_1. \tag{4}$$

Having localized the artifact in both frames and obtained the background flow, we can determine which pixel of I_k is useful to restore the degraded pixels for the current frame. We identify these effective pixels in the $\mathcal{W}_{k\to t}(I_k)$ by computing the following map:

$$A^{\text{effect}}_t = (1 - \mathcal{W}_{k\to t}(A_k)) \odot A_t, \tag{5}$$

which we use to guide the following restoration modules.

Spatio-temporal restoration. Prior approaches [51, 53, 28] exploit the spatio-temporal information from a fixed number of frames, yet the number of adjacent frames needed for the restoration may vary depending on the magnitude of camera motion and the size of degraded region. In view of this, we propose a recurrent restoration network that provides the flexibility of feeding a varying number of adjacent frames and can thereby leverage long-term temporal information when necessary.

The whole recurrent restoration module consists of two steps: temporal fusion and spatial restoration. The temporal fusion iteratively estimates a sequence of temporarily restored results. In each iteration, the recurrent module produces an intermediate image restoration T^i_t and a hidden state h^i based on the T^{i-1}_t and h^{i-1} of the last iteration. We regard I_t to be the initial restoration result, *i.e.*, $T^0_t = I_t$. The recurrent module adopts a convolutional gated recurrent unit (ConvGRU) [9], and the iteration process can be formulated as follows,

$$\begin{aligned} z_i &= \sigma(\text{Conv}[h^{i-1}, x_i]) \\ r_i &= \sigma(\text{Conv}[h^{i-1}, x_i]) \\ h' &= \tanh(\text{Conv}[r_i \odot h^{i-1}, x_i]) \\ h^i &= (1 - z_i) \odot h^{i-1} + z_i \odot h' \end{aligned} \tag{6}$$

where z_i and r_i are update gate and reset gate respectively, and x_i is the feature of the input which is a concatenation of the frame I_t, the attention map A_t, the warped frame $\mathcal{W}_{k\to t}(I_k)$ and the effective restoration map A^{effect}_t. Once the hidden state is updated by the GRU block, it will pass through three convolutional layers followed by a sigmoid function to predict a blending mask M, which is used to attentively fuse the warping $\mathcal{W}_{k\to t}(I_k)$ and the intermediate prediction T^{i-1}_t:

$$T^i_t = M \odot \mathcal{W}_{k\to t}(I_k) + (1 - M) \odot T^{i-1}_t \tag{7}$$

We enforce such intermediate result by minimizing its mean square error against the ground truth C_t. Note that we compute the loss for all the iterations $2N$ and each iteration is accounted by different factors:

$$\mathcal{L}_{\text{fusion}} = \frac{1}{2N} \sum_{i=1}^{2N} \gamma^{|2N-i|} \left\| T^i_t - C_t \right\|_2^2. \tag{8}$$

In the experiments we empirically use $\gamma = 0.8$.

As more adjacent frame are utilized, the restoration progressively improves. Nonetheless, there may exist scene locations occluded in all the frames, so it still requires to leverage the spatial prior for restoration. We use the contextual autoencoder architecture from [32] for this spatial restoration task, as shown in Figure 4 (a). The network receives the temporal fusion result T^i_t and the hidden state h^i as the input, and learns the spatial restoration by minimizing the perceptual loss [21]:

$$\mathcal{L}_{\text{spatial}} = \frac{1}{2N} \frac{1}{L} \sum_{i=1}^{2N} \sum_{l=1}^{L} \left\| \phi^l(P^i_t) - \phi^l(C_t) \right\|_2^2 \tag{9}$$

where P^i_t denotes the spatial restoration output at ith iteration and ϕ^l is the lth layer of a pretrained VGG model [39]. The spatial restoration module is capable to deal with different levels of degradation during training, complementing the restoration ability of the recurrent fusion.

In summary, we train the entire single-frame restoration network using the following objective function:

$$\mathcal{L}_{\text{single}} = \mathcal{L}_{\text{att}} + \mathcal{L}_{\text{flow}} + \lambda_1 \mathcal{L}_{\text{fusion}} + \lambda_2 \mathcal{L}_{\text{spatial}} \tag{10}$$

where the coefficients λ_1 and λ_2 balance different terms. In the experiments we set $\lambda_1 = 100$ and $\lambda_2 = 10$.

Frame | Input Flow | GT Flow | Conv | GatedConv | FeatFusion | Ours

Figure 6: Example results of different network architectures for flow completion. Our full model with feature fusion layers and the upsampling module can produce a more accurate result with sharper motion boundaries.

Input | GT | w/o Attention Map | w/o Flow Completion | w/o Spatial Restor. | Full Model

Figure 7: Ablation study of single-frame stage. Our full model can generate the results with fewer visible artifacts.

3.3. Multi-frame Processing

By far, the video frames are processed individually, based on the adjacent frames. Hence, the single frame restoration (denoted by Ψ) can be formulated as,

$$P_t = \Psi(I_t|\{I_k\}), k \in [t-N, t+N]. \tag{11}$$

However, the temporal consistency over the entire output sequence cannot be guaranteed due to the nature of frame-by-frame processing. To address this, we propose a multi-frame processing stage as shown in Figure 2 (b) and Figure 4 (b). As opposed to the first stage that keeps refining one frame in different iterations, the multi-frame processing refines different input frames during iterations. Concretely, we feed into the outputs from the first stage in sequence, and let the network adjust P_t based on the earlier output frame O_{t-1}, so the processing becomes:

$$O_t = \Psi(P_t|O_{t-1}), \quad O_0 = P_0. \tag{12}$$

A slight difference of the pipeline is that the attention detection module is no longer needed since the input frames $\{P_t\}$ are already cleaned by the frame-wise processing. Besides, we introduce a temporal loss $\mathcal{L}_{\text{temporal}}$ to enforce the temporal consistency between successive outputs:

$$\mathcal{L}_{\text{temporal}} = \frac{1}{M-1}\sum_{t=2}^{M}\Big\{e^{-\|C_t - \mathcal{W}_{t-1\to t}(C_{t-1})\|_2^2/\mu} \times \big\|O_t - \mathcal{W}_{t-1\to t}(O_{t-1})\big\|_1\Big\} \tag{13}$$

where $\mathcal{W}_{t-1\to t}(C_{t-1})$ and $\mathcal{W}_{t-1\to t}(O_{t-1})$ are the frame warpings using the ground truth flow, M is the length of the video sequence, and we set the exponential coefficient $\mu = 0.02$. The overall loss for training the multi-frame processing is defined as,

$$\mathcal{L}_{\text{multi}} = \mathcal{L}_{\text{flow}} + \lambda_1\mathcal{L}_{\text{fusion}} + \lambda_2\mathcal{L}_{\text{spatial}} + \lambda_3\mathcal{L}_{\text{temporal}}, \tag{14}$$

where the newly introduced weight λ_3 is set by 10.

4. Experiments

4.1. Implementation

Training details. We adopt Adam optimizer [22] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 0.0001, and batch size of 8 images for training. Each image uses five neighboring frames as input, where the middle frame is the one to be restored in the single-frame stage. During training, We randomly crop these images from 384×384 to 256×256

Table 1: Ablation study for the flow completion network.

	Input	Conv	GatedConv	FeatFusion	Ours
EPE ↓	3.17	2.10	1.89	1.72	**1.60**

Table 2: Ablation study for the single-frame stage.

Model	PSNR ↑	SSIM ↑	E_{warp} ↓
w/o Attention Map	33.70	0.976	0.0046
w/o Flow Completion	34.09	0.975	0.0046
w/o Spatial Restoration	29.61	0.953	0.0049
Full Model	**35.37**	**0.980**	**0.0045**

for data augmentation. It first takes 300 epochs to train the single-frame stage. After that, we run the trained single-frame model on the entire dataset to generate the training set for the multi-frame stage, which takes another 50 epochs to converge. Our method is implemented using Pytorch [31]. The entire training takes approximately five days on 8x GeForce RTX 2080Ti GPUs.

Datasets. We render 600 video clip pairs as our training set, where each clip has 30 frames at 6fps and a resolution of 384×384. For the test set, we produce another 30 clip pairs with random rendering parameters to differentiate from the training set. We will use this test set for the quantitative evaluation in ablation studies and comparisons since the ground truth videos are available. For qualitative results, we use a Canon EOS 80D camera to capture the real videos with different contaminants adhered to the lens.

4.2. Ablation Study

Flow completion network. We first conduct an ablation study to demonstrate the effectiveness of our flow completion network. Different architectures with the same attention detection module are used to learn the completed flows. Specifically, we adopt the same encoder-decoder architecture with plain convolution layer (Conv), gated convolution layer (GatedConv) [54] for image inpainting and our feature fusion layer (FeatFusion). Finally, we use the feature fusion layer and replace the decoder with our upsampling module which is our full model for flow completion task (Ours). As shown in Table 1 and Figure 6, our full model achieves the most accurate flow with the lowest endpoint error (EPE) and sharpest motion boundaries.

Effectiveness of the main component. We validate the effectiveness of three main components in the first stage: attention detection, flow completion and spatial restoration. To keep all the other modules intact, we remove the attention detection module by giving attention maps with all zeros to the subsequent modules. For the removal of the flow completion, we directly use the degraded flow without any

Table 3: Quantitative comparison of our approach with other methods.

Method	Type	PSNR ↑	SSIM ↑	E_{warp} ↓
PReNet [35]	Single-image	33.78	0.977	0.0049
AttGAN [32]	Single-image	35.05	**0.980**	0.0047
FastDerain [20]	Video-based	20.80	0.794	0.0075
ObsRemoval [28]	Video-based	29.17	0.952	0.0052
FastDVDnet [41]	Video-based	31.95	0.936	0.0051
Ours (stage one)	Video-based	**35.37**	**0.980**	0.0045
Ours	Video-based	34.98	0.979	**0.0035**

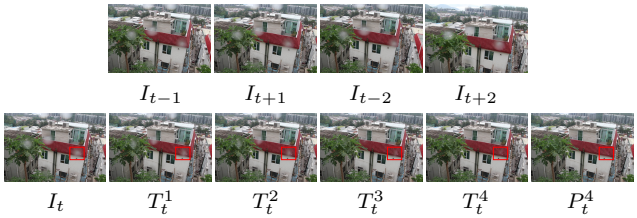


Figure 8: Example results showing the progressive restoration of our recurrent network. The first row is the input neighboring frames, and the second row is the corresponding restoration results.

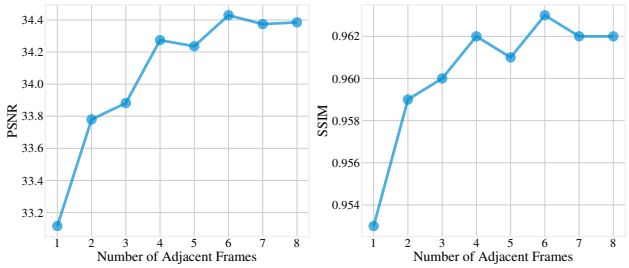


Figure 9: Quality comparison of our approach with different input frames. Our method is able to use more frames for better performance until convergence.

Figure 10: Qualitative comparison with single-image restoration methods on the real images.

processing. In addition to using metrics such as PSNR and SSIM, we also use the warping error (E_{warp}) from [23] to measure the temporal consistency of the results, *i.e.*, we apply the method in [40] to detect the occlusion regions and calculate the consistency between every two consecutive frames excluding these pixels. As shown in Table 2, the best result is achieved in terms of PSNR, SSIM and E_{warp} when the full model is used. Examples from the test set are

Input FastDerain [20] ObsRemoval [28] FastDVDnet [41] Ours (stage one) Ours

Figure 11: Qualitative comparison with video-based restoration methods on the real video frames contaminated by dirt.

shown in Figure 7. One can see that the proposed full model can generate results with fewer visible artifacts.

Recurrent restoration. Owing to the recurrent design of our network, our method is able to restore the frame progressively by iteratively utilizing the adjacent frames as shown in Figure 8. In this example, four iterations of temporal fusion are used to obtain the output. Figure 9 plots the PSNR and SSIM results against the number of input frames or iterations, showing that the optimal number of iterations is approximately six. We could also determine the number of iterations automatically by using the blending mask from the temporal restoration module, which indicates how many pixels are used for restoration at the current frame/iteration. We could therefore stop when it reaches below a given threshold. Since our training dataset includes videos with diverse magnitudes of motion, the network can learn to handle different cases in each iteration. In addition, due to the special design of the recurrent unit, it could generalize to the different number of iterations with satisfactory performance.

4.3. Comparisons

We compare our method with related techniques for single-image and video-based restoration on our test set. Five competitive methods with public source code are included, which are PReNet [35], AttGAN [32], FastDerain [20], ObsRemoval [28] and FastDVDnet [41]. Among them, FastDerain [20] is an optimization-based method and others are learning-based approaches which are retrained on the same training set. All the video-based methods leverage the same number of input frames as our approach for both training and testing during the comparison. These methods focus on different restoration tasks like adherent raindrop removal [32], rain streak removal/deraining [35, 20], obstruction removal [28], and video denoising [41], which could be potentially applied to our task. As shown in Table 3, our single-frame stage outperforms other methods in terms of PSNR and SSIM whereas the full model with the multi-frame processing achieves the lowest warping error. The new temporal loss in the full model significantly improves the temporal consistency albeit the slight drop of PSNR and SSIM. Figure 10 and 11 showcase the results on real scenes for qualitative comparison. Our method generalizes well to the real captured images and demonstrates more visually pleasing results without noticeable contaminant artifacts. Figure 1 shows that our method has the ability to remove various contaminants in the real world and produces high-quality results. We provide additional results in conjunction with the video outputs in the *supplementary material*.

Running time. We evaluate the inference time of all compared methods on the Intel Xeon Gold 6244 machine with an Nvidia GeForce RTX-2080Ti GPU card. The resolution of the input videos is 256×256. The average times to process one frame for the different methods are 0.029s for PReNet [35], 0.025s for AttGAN [32], 1.28s for FastDerain [20], 1.25s for ObsRemoval [28], 0.0088s for FastDVDnet, and 0.88s for our method.

5. Conclusion

We present a novel framework that removes the contaminant artifact for moving cameras. We propose an attention detection module to localize the degraded regions and a flow completion module to recover the background motion for better alignment. Guided by the attention map and the restored flows, we recurrently fuse corresponding clean pixels to the current frame using the reference frames. Ultimately a multi-frame processing stage improves the temporal consistency. Experiments on both synthetic dataset and real scenes verify the effectiveness of each component and prove the quality advantage over prior approaches. We will make the synthetic dataset along with the source code publicly available and hopefully benefit the following works.

Acknowledgements: This work was supported in part by grants from the Hong Kong Research Grants Council (RGC) to HKUST and CityU, including the Early Career Scheme under Grant 9048148 (CityU 21209119).

References

[1] Vivek Akkala, Parth Parikh, BS Mahesh, Ajinkya S Deshmukh, and Swarup Medasani. Lens adhering contaminant detection using spatio-temporal blur. In *2016 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE, 2016. 2

[2] Jean-Baptiste Alayrac, Joao Carreira, and Andrew Zisserman. The visual centrifuge: Model-free layered video representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2457–2466, 2019. 2

[3] Nikolaos Arvanitopoulos, Radhakrishna Achanta, and Sabine Susstrunk. Single image reflection suppression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4498–4506, 2017. 2

[4] CS Asha, Sooraj Kumar Bhat, Deepa Nayak, and Chaithra Bhat. Auto removal of bright spot from images captured against flashing light source. In *2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pages 1–6. IEEE, 2019. 2

[5] Peter C Barnum, Srinivasa Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. *International journal of computer vision*, 86(2-3):256, 2010. 3

[6] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9066–9075, 2019. 1

[7] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. 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*, pages 6286–6295, 2018. 3

[8] Li Chen, Dawei Zhu, Jing Tian, and Jiaxiang Liu. Dust particle detection in traffic surveillance video using motion singularity analysis. *Digital Signal Processing*, 58:127–133, 2016. 2

[9] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 5

[10] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 3

[11] David Eigen, Dilip Krishnan, and Rob 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

[12] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3238–3247, 2017. 2

[13] Kshitiz Garg and Shree K Nayar. Detection and removal of rain from videos. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004. 3

[14] Kshitiz Garg and Shree K Nayar. When does a camera see rain? In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1067–1074. IEEE, 2005. 3

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

[16] Kshitiz Garg and Shree K Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007. 3

[17] Jinwei Gu, Ravi Ramamoorthi, Peter Belhumeur, and Shree Nayar. Removing image artifacts due to dirty camera lenses and thin occluders. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10. ACM, 2009. 1, 2, 3

[18] Zhixiang Hao, Shaodi You, Yu Li, Kunming Li, and Feng Lu. Learning from synthetic photorealistic raindrop for single image raindrop removal. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[19] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016. 1

[20] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Transactions on Image Processing*, 28(4):2089–2102, 2018. 7, 8

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[23] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 7

[24] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6644–6653, 2018. 3

[25] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018. 2

[26] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. D3r-net: Dynamic routing residue recurrent network for video rain removal. *IEEE Transactions on Image Processing*, 28(2):699–712, 2018. 3

[27] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3233–3242, 2018. 3

[28] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Learning to see through obstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14215–14224, 2020. 2, 5, 7, 8

[29] Scott McCloskey, Michael Langer, and Kaleem Siddiqi. Removal of partial occlusion from single images. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):647–654, 2010. 2

[30] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019. 2

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 7

[32] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying 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. 2, 5, 7, 8

[33] Yuhui Quan, Shijie Deng, Yixin Chen, and Hui Ji. Deep learning for seeing through window with raindrops. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2463–2471, 2019. 2

[34] Ramesh Raskar, Amit Agrawal, Cyrus A Wilson, and Ashok Veeraraghavan. Glare aware photography: 4d ray sampling for reducing glare effects of camera lenses. In *ACM SIGGRAPH 2008 papers*, pages 1–10. ACM, 2008. 2

[35] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3937–3946, 2019. 7, 8

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[37] Martin Roser and Andreas Geiger. Video-based raindrop detection for improved image registration. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 570–577. IEEE, 2009. 2

[38] Varun Santhaseelan and Vijayan K Asari. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, 112(1):71–89, 2015. 3

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[40] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *European conference on computer vision*, pages 438–451. Springer, 2010. 7

[41] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1354–1363, 2020. 7, 8

[42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. *arXiv preprint arXiv:2003.12039*, 2020. 4, 5

[43] Michal Uricar, Ganesh Sistu, Hazem Rashed, Antonin Vobecky, Varun Ravi Kumar, Pavel Krizek, Fabian Burger, and Senthil Yogamani. Let's get dirty: Gan based data augmentation for camera lens soiling detection in autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 766–775, 2021. 2

[44] Patricia Vitoria and Coloma Ballester. Automatic flare spot artifact detection and removal in photographs. *Journal of Mathematical Imaging and Vision*, 61(4):515–533, 2019. 2

[45] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8178–8187, 2019. 2

[46] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2019. 1

[47] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T Freeman. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015. 2

[48] Atsushi Yamashita, Isao Fukuchi, and Toru Kaneko. Noises removal from image sequences acquired with moving camera by estimating camera motion from spatio-temporal information. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3794–3801. IEEE, 2009. 2

[49] Ping Yang, Li Chen, Jing Tian, and Xin Xu. Dust particle detection in surveillance video using salient visual descriptors. *Computers & Electrical Engineering*, 62:224–231, 2017. 2

[50] Wenhan Yang, Jiaying Liu, and Jiashi Feng. Frame-consistent recurrent video deraining with dual-level flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1661–1670, 2019. 3

[51] Shaodi You, Robby T Tan, Rei Kawakami, and Katsushi Ikeuchi. Adherent raindrop detection and removal in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1035–1042, 2013. 2, 5

[52] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Raindrop detection and removal from long range trajectories. In *Asian Conference on Computer Vision*, pages 569–585. Springer, 2014. 2

[53] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Adherent raindrop modeling, detectionand removal in video. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1721–1733, 2015. 2, 5

[54] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019. 1, 2, 7

[55] Ruisong Zhang, Weize Quan, Baoyuan Wu, Zhifeng Li, and Dong-Ming Yan. Pixel-wise dense detector for image inpainting. In *Computer Graphics Forum*, volume 39, pages 471–482. Wiley Online Library, 2020. 1

[56] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE international conference on multimedia and expo*, pages 461–464. IEEE, 2006. 3

[57] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4786–4794, 2018. 2