Video Waterdrop Removal via Spatio-Temporal Fusion in Driving Scenes

Qiang Wen, Yue Wu, and Qifeng Chen

***Abstract*— The waterdrops on windshields during driving can cause severe visual obstructions, which may lead to car accidents. Meanwhile, the waterdrops can also degrade the performance of a computer vision system in autonomous driving. To address these issues, we propose an attention-based framework that fuses the spatio-temporal representations from multiple frames to restore visual information occluded by waterdrops. Due to the lack of training data for video waterdrop removal, we propose a large-scale synthetic dataset with simulated waterdrops in complex driving scenes on rainy days. To improve the generality of our proposed method, we adopt a cross-modality training strategy that combines synthetic videos and real-world images. Extensive experiments show that our proposed method can generalize well and achieve the best waterdrop removal performance in complex real-world driving scenes.**

I. INTRODUCTION

With the increasing applications of visual perception methods [1], [2], [3], [4] in robotics and autonomous driving, the robustness of these methods has become more and more important in their visual system. However, the performance of current vision methods significantly degrades in rainy weather in autonomous driving scenarios, since raindrops on the windshield or camera lens cause inevitable visual obstruction, as mentioned in Porav et al. [5]. Therefore, removing waterdrops from videos on rainy days is highly important for self-driving cars and various robot applications.

Although many researchers [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [11], [17] propose dedicated frameworks to remove rain streaks, video waterdrop removal receive much less attention and is still an open question. Since there exists a significant geometric gap between waterdrops and rain streaks, the dedicated methods proposed to remove rain streaks cannot perform well on the waterdrop removal task as mentioned in [18]. As shown in Fig. 1, directly applying a state-of-the-art rain streak removal method [19] to remove waterdrops will not yield satisfactory results. Instead of being like the line shape of rain streaks, waterdrops tend to be ellipses scattered on windshields. And each waterdrop usually occupies a larger area than a rain streak, thus being more difficult to deal with.

To tackle the waterdrop removal problem, some researchers [20], [21], [22], [23], [24], [5] propose specialized frameworks to remove waterdrops from a single image. However, they still cannot handle complex driving videos, which are very common in Autonomous Driving. On the one hand, their datasets are usually collected by sprinkling waterdrops on the glass, thus there is a large domain gap between these data and real driving scenes. On the other hand, their methods lack temporal information utilization, which is necessary for video tasks.

Fig. 1: Real-world driving videos with waterdrops.

The lack of paired real-world training data for video waterdrop removal limits the performance of learning-based methods in the real world. It is almost impossible to collect perfectly aligned driving videos with and without waterdrops. To address this issue, we propose a large-scale synthetic video waterdrop dataset with paired data for training. To promote our proposed method to generalize to real driving scenes, we adopt a cross-modality training strategy that jointly trains our model on our large-scale synthetic video dataset and a small-scale real-world image dataset proposed by [21]. We also propose a spatio-temporal fusion-based framework to restore the background information under the regions occupied by sparse waterdrops and even the streaks of waterdrops, as shown in Fig. 1. With the dedicated framework and cross-modality strategy, the proposed method achieves the best-performing video waterdrop removal in real driving scenes.

Our contributions can be summarized as follows.

1) We propose a dedicated framework for video waterdrop removal in complex driving scenes. Our framework is based on spatio-temporal fusion that uses a self-attention mechanism to exploit both spatial and temporal information for restoring clean video frames.
2) We are the first to propose a large-scale synthetic video dataset for the waterdrop removal task. We design a video waterdrop synthesis algorithm based on Hao *et al*. [24] to generate synthetic data.
3) To tackle the domain gap between the synthetic data and real driving scenes, we propose a cross-modality strategy for jointly training the proposed method on our

Qiang Wen (qwenab@connect.ust.hk), Yue Wu (ywudg@connect.ust.hk), and Qifeng Chen (cqf@ust.hk) are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.

synthetic video data and the real-world image data [21].

4) The extensive evaluations demonstrate that our waterdrop removal method significantly outperforms previous works on our synthetic dataset and real driving scenes quantitatively and qualitatively.

II. Related Work

A. Single-Image Methods

Although there are many methods focusing on image deraining, they mainly attempt to remove rain streaks instead of waterdrops from images. To remove watedrops from images, some learning-based methods have been proposed in recent years. Eigen *et al*. [20] propose the first CNN-based method to remove the waterdrops from a single degraded image. Due to the over-shallow network architecture design, their method shows poor removal performance for large and dense waterdrops. Qian *et al*. [21] propose a generative adversarial network (GAN) based method. Their method generates an attention map for each input image. However, with the simple concatenation of the input image and its corresponding attention map, their method still only focuses on the local spatial information instead of the global one. Quan *et al*. [22] propose a shape-driven attention and channel re-calibration to locate and process waterdrops. However, this method is still limited by the local attention mechanism that does not sufficiently explore the long-range but helpful information from the whole single image. Hao *et al*. [24] propose a waterdrop synthesis algorithm for the single image. With their synthetic image dataset, they train a deep network for waterdrop detection and removal. However, their method shows poor performance in real driving scenes. Quan *et al*. [18] propose to utilize the neural architecture search method [25] to generate a complementary network to tackle rain streak removal and waterdrop removal jointly. But their method is still too weak to remove waterdrops from real driving scenes.

B. Multi-Image Methods

Due to limited clues in a single image for clean background information recovery, some researchers [26], [23], [27], [28] propose to take multiple degraded images as input to reconstruct the clean ones. You *et al*. [26] propose to calculate the dense motion change to detect waterdrops in each frame. However, their method needs to retrieve similar but clean information over nearly one hundred frames to restore the regions under waterdrops. Shi *et al*. [23] propose a dedicated framework for the stereo waterdrop removal task. Their method relies heavily on the disparity map in each stereo pair, which is not available in monocular driving videos. A coarse-to-fine method is proposed by Liu *et al*. [27] to tackle obstruction removals, such as reflection removal and fence removal. Their method can be extended to waterdrop removal task. Based on the flow estimation, they propose to warp nearby frames to provide helpful information for recovery. Most recently, Alletto *et al*. [28] propose a spatio-temporal de-raining model to tackle single-image and video waterdrop removal. They also propose a waterdrop synthesis algorithm to simulate driving scenes during rain. Although the abovementioned two methods can remove sparse waterdrops from multiple images, they fail to estimate a correct optical flow when there are numerous waterdrops over a sequence of frames. Considering the bottleneck of flow estimation, we propose to utilize a temporal attention block that directly provides effective information from nearby frames to restore clean background information.

III. Approach

Given a sequence of frames with waterdrops $\left\{\tilde{F}_t\right\}_{t=1}^{T}$, where T is the sequence length, our goal is to remove the waterdrops in these frames and recover the clean ones $\left\{\hat{F}_t\right\}_{t=1}^{T}$. To tackle waterdrop removal, we consider two types of occlusions: the partial occlusion and the complete occlusion, as shown in Fig. 3. There is still some meaningful background information covered by the waterdrop in the partial occlusion, while the information in the complete occlusion is totally lost for background restoration. To address this problem, we propose a *pixel attention block* to pixel-wisely re-weight the intermediate feature of each input frame. The goal of the pixel attention block is to enhance the background information in the partial occlusion and suppress the meaningless information in the complete occlusion.

After re-weighting each pixel in the feature, we need to refine the pixels in partially occluded regions and fill completely occluded regions with meaningful values. We treat this processing as an inpainting task [30], [31]. Inspired by [31], we adopt a self-attention mechanism to refine and fill regions degraded by waterdrops. Unlike previous solely CNN-based works which are limited to exploiting local information, we utilize the self-attention mechanism to restore background information by fusing global information across the whole spatial dimension. Based on such a self-attention mechanism, we propose a *spatial attention block* to refine re-weighted features from the pixel attention block.

Although the *spatial attention block* can restore most regions in features, some regions that are still degraded can be restored by utilizing features from nearby frames. Hence, we propose a *temporal attention block* to fully exploit the valid information from nearby frames. Similarly, but differently to the self-attention mechanism in the *spatial attention block*, the *temporal attention block* refines multiple features simultaneously by fusing the global information across the temporal dimension. Through the above spatio-temporal fusion, the proposed method can reconstruct cleaned frames accurately.

A. Pixel Attention Block (PAB)

Given a sequence of frames within waterdrops $\left\{\tilde{F}_t\right\}_{t=1}^{T}, \tilde{F}_t \in \mathbb{R}^{H\times W\times 3}$, we first feed them into the encoder and obtain a sequence of features $\{f_t\}_{t=1}^{T}, f_t \in \mathbb{R}^{h\times w\times c}$, where $h = \frac{H}{4}$, $w = \frac{W}{4}$ and c denotes the channel number. After encoding, each feature is fed into the pixel attention block to obtain its

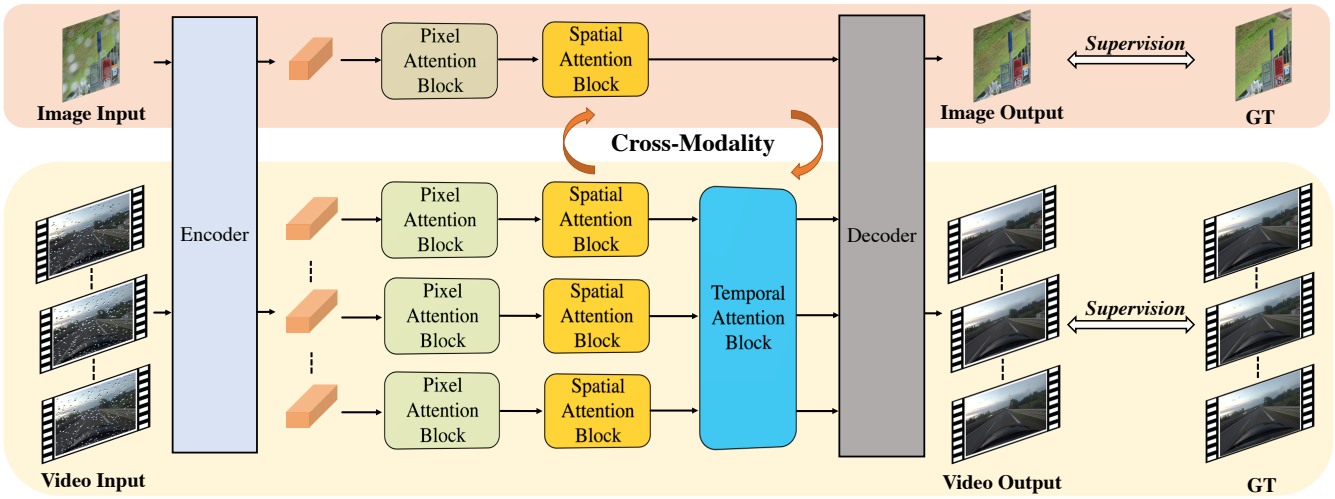

Fig. 2: Overview of the proposed method. We propose a framework with spatial-temporal fusion, based on a self-attention mechanism [29], to reconstruct each feature across spatial and temporal dimensions, respectively. Meanwhile, we also adopt a cross-modality strategy to train our model on our proposed synthetic video data and the real-world image data [21] jointly.

(a) Partial occlusion (b) Complete occlusion

Fig. 3: Two types of occlusions caused by waterdrops. Examples are from the dataset [21].

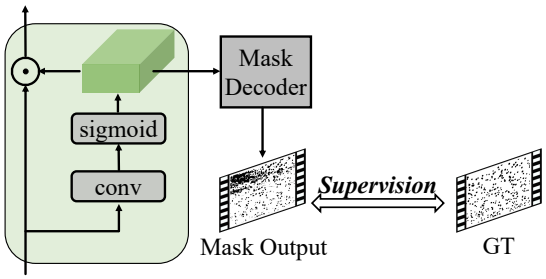

Fig. 4: The detail of the pixel attention block in the proposed method.

corresponding pixel-wise confidence map a_t, as shown in Fig. 4:

$$a_t = \sigma(\Theta_{PA}(f_t)), \tag{1}$$

where $1 \leq t \leq T$, σ denotes the sigmoid activation. To obtain a pixel-wise confidence map for each feature, we adopt a similar block in SENet [32] but we discard the global average pooling layer and replace the fully connected layer with the convolutional layer Θ_{PA}. Since each waterdrop only occupies a narrow region across the spatial dimension in each feature channel, this encourages us to re-weight each pixel by a pixel-wise confidence map a_t instead of a common weight as in [32]:

$$f_t^{'} = a_t \odot f_t, \tag{2}$$

where $\odot$ denotes a pixel-wise multiplication operation. During re-weighting, the value of each element in the pixel-wise confidence map is expected to be 0 for the complete occlusion, 1 for the clean background region.

B. Spatio-Temporal Fusion

1) Spatial Attention Block (SAB): After re-weighting, we need to refine the pixels in the partial occlusion and fill the complete occlusion with meaningful pixel values. To address this, we utilize a self-attention mechanism [29] to reconstruct each feature by fusing patches with a predicted attention map:

$$q_t = \Psi_q(f_t^{'}), \quad k_t = \Psi_k(f_t^{'}), \quad v_t = \Psi_v(f_t^{'}), \tag{3}$$

where $\Psi_q(\cdot)$, $\Psi_k(\cdot)$ and $\Psi_v(\cdot)$ denote the feature embedding for query, key and value, which consists of the 2D convolutional layers with 1×1 kernel size. We extract $N = h \times w$ patches with size $1 \times 1 \times c$ from the query, key and value features, respectively. To calculate an attention map, we reshape each patch of query and key features into a 1-D vector, so the similarity between each query patch p^q and each key patch p^k is

$$s_{i,j} = \frac{p_i^q \cdot (p_j^k)^T}{\sqrt{1 \times 1 \times c}}, \tag{4}$$

where $1 \leq i, j \leq N$, p_i^q and p_i^k denote the i-th query patch and j-th key patch, respectively, $\cdot$ denotes a matrix multiplication operation. As it is mentioned in [29], the similarity value normalized by the dimension of each vector can avoid a small gradient caused by a subsequent softmax function. Finally, the attention map is

$$a_{i,j} = \frac{\exp(s_{i,j})}{\sum_{n=1}^{N} \exp(s_{i,n})}. \tag{5}$$

With the attention map, the output p_i^o for each query patch p_i^q is a weighted fusion of corresponding value patches p^v:

$$p_i^o = \sum_{j=1}^{N} a_{i,j} p_j^v. \tag{6}$$

After receiving these output patches, we reshape them into the size $h \times w \times c$ to obtain the spatially refined feature.

2) Temporal Attention Block (TAB): In the temporal attention block, we extract multi-scale patches for queries, keys, and values from multiple input frames features as in [31]. For the waterdrop removal task, the large patches are marvelous for semantic-level reconstruction, and the small ones encourage texture-level reconstruction. For the trade-off between the computation cost and recovering performance, we split each feature into two parts with size $h \times w \times \frac{c}{2}$ and extract different-size patches ($2 \times 2 \times \frac{c}{2}$ and $8 \times 8 \times \frac{c}{2}$) from them. After fusing patches with a self-attention mechanism, we reshape fused patches into $h \times w \times c$ features to receive temporally refined features.

After the spatio-temporal fusion, we feed each temporally refined feature into the decoder to obtain cleaned frames $\left\{\hat{F}_t\right\}_{t=1}^{T}$.

C. Training

1) Cross-modality Training Strategy: Although the proposed method only trained on our synthetic dataset can generalize well to real driving scenes, training it jointly on our synthetic video data and the real-world image data [21] as shown in Fig. 2 improves the generalization performance of our method. As mentioned in [21], the dataset [21] only consists of degraded images with corresponding clean images, and there is a slight spatial misalignment in each image pair, so we need to utilize these image data carefully.

2) Loss Functions for Frame Sequences:

a) Mask Loss: In the pixel attention block, to encourage the network to predict the pixel-wise attention map precisely as much as possible, we feed the predicted pixel-wise attention map into the mask decoder to receive a waterdrop mask for each input frame, the mask loss is

$$\mathcal{L}_{mask}^{F} = \frac{1}{T}\sum_{t=1}^{T} BCE(\hat{M}_t, M_t), \tag{7}$$

where $BCE(\cdot)$ denotes binary cross entropy loss function, $\hat{M}_t$ denotes the predicted mask and M_t denotes the ground-truth mask from our synthetic dataset.

b) Reconstruction Loss: For the final cleaned frame reconstruction, we add a pixel-wise loss between the network outputs and the ground truths:

$$\mathcal{L}_{recons}^{F} = \frac{1}{T}\sum_{t=1}^{T} \left\| \hat{F}_t - F_t \right\|_2^2, \tag{8}$$

where $\tilde{F}_t$ denotes the cleaned frame output.

c) Temporal Loss: To guarantee the temporal consistency among network outputs, we adopt the conditional video discriminator [33] to calculate the temporal loss (TL) for network outputs:

$$\mathcal{L}_{TL}^{F} = E_{\widehat{x}\sim P_{\tilde{\mathbf{F}}_\mathbf{T}}(\widehat{x})}[logD_V(\widehat{x})], \tag{9}$$

where $P(\cdot)$ denotes the data distribution, $\tilde{\mathbf{F}}_\mathbf{T}$ denotes the concatenation of all the network outputs, D_V denotes the discriminator whose loss function is

$$\mathcal{L}_D = E_{x\sim P_{\mathbf{F}_\mathbf{T}(x)}}[logD_V(x)] + E_{\widehat{x}\sim P_{\tilde{\mathbf{F}}_\mathbf{T}(\widehat{x})}}[1 - logD_V(\widehat{x})], \tag{10}$$

where $\mathbf{F}_\mathbf{T}$ denotes the concatenation of all the ground-truth frames from our synthetic dataset.

The total loss functions for each frame sequence are concluded as follows:

$$\mathcal{L}_{total}^{F} = \lambda_1^F \cdot \mathcal{L}_{mask}^{F} + +\lambda_2^F \cdot \mathcal{L}_{recons}^{F} + \lambda_3^F \cdot \mathcal{L}_{TL}^{F}, \tag{11}$$

where the weights for different losses are set as $\lambda_1^F = 10$, $\lambda_2^F = 25$, and $\lambda_3^F = 5$.

3) Loss Functions for Images: Considering the spatial misalignment between the input image $\tilde{I}$ and its clean ground truth I, we only adopt a feature matching loss between the network output and the ground truth:

$$\mathcal{L}_{feat}^{I} = \sum_{l} \lambda_l \left\| \Phi_l(\hat{I}) - \Phi_l(I) \right\|_1, \tag{12}$$

where $\hat{I}$ denotes the network output, Φ_l denotes the layer l in the VGG-16 network [34] where we select the layers $\mathrm{conv1_2}$, $\mathrm{conv2_2}$, $\mathrm{conv3_2}$, $\mathrm{conv4_2}$ and $\mathrm{conv5_2}$, $\{\lambda_l\}$ denotes the weights for the layers we select. Besides, since there is no need to exploit the temporal information for image reconstruction, we discard the temporal attention block during attention-based fusion.

IV. Synthetic Video Data Generation

Due to the lack of paired data for multi-image/video methods training, we propose a synthetic waterdrop dataset for driving scenes. To the best of our knowledge, there are only three image waterdrop synthesis works [24], [28], [5] which cannot be utilized to generate waterdrops for videos directly. This encourages us to extend such algorithms to video waterdrop synthesis by considering the followings:

1) For a driving video, each waterdrop remains at the same position over a sequence of frames. With the wind or the intense movement of the car, there will be a tiny shift in each waterdrop.

2) With the evaporation of waterdrops, some completely occluded regions turn to be partially occluded along a sequence of frames.

Based on these observations, we design a video waterdrop synthesis algorithm based on [24]. For a sequence of clean frames, we generate 150 to 400 waterdrops in the first frame. For each next frame, we add a shift along a random direction to each waterdrop. Meanwhile, we linearly enlarge the blur kernel size to make waterdrops progressively blurry over frames, which can simulate the evaporation of waterdrops. During synthesizing, we set the length of each sequence as 5, the shift value as 1 pixel, and the blur kernel size from 3 to 20 pixels.

To collect clean driving videos for data synthesis, we choose the DR(eye)VE Dataset [35] which contains 51

Fig. 5: Qualitative comparison on real driving scenes collected from the Internet. The results show that our method can remove various-shaped waterdrops in multiple kinds of weather. The last column shows the proposed method still presents a satisfying performance in the extremely challenging case.

videos with multiple scenes and weathers, varying from the morning, evening, night, sunny, cloudy, countryside, downtown, and highway.

To summarize, we propose a large-scale synthetic dataset with numerous triplets $(F, \tilde{F}, M)$. In each triplet, there are one clean frame F, one frame $\tilde{F}$ with synthetic waterdrops and one binary mask M for waterdrops. Totally, we have 67500 triplets from 45 videos for training and 600 triplets from 6 videos for testing.

V. Experiments

A. Implementation Details

We train the network by minimizing $(\mathcal{L}^F_{total} + \mathcal{L}^I_{feat})$ on synthetic data and real data jointly. To be specific, for every 1000 training iterations, we train the network on synthetic data for 900 iterations while on real-world data for 100 iterations. We set the proportion as $9:1$, the best one we select from $5:1$, $7:1$, $9:1$ and $11:1$ by evaluations. Totally, we train the network for 75000 iterations on both datasets using the ADAM optimizer [36] with $lr = 0.0001$ and $(\beta_1, \beta_2) = (0.9, 0.999)$. We set the length T of each frame sequence as 5, and channel number c as 256 during training.

B. Comparison to State-of-the-Art

As abovementioned in Sec. II-B, there is only one video waterdrop removal method proposed by Alletto *et al*. [28]. However, the lack of source code and real-scene evaluations impedes us from making a fair comparison with them. To evaluate our model and make fair comparisons, we select the most recent and state-of-the-art methods for comparisons:

1) AttentGAN [21], Quan *et al*. [22], CCN [18] are learning-based methods for single-image waterdrop removal.
2) Vid2Vid [33] is a recent and typical video-to-video translation method that is regarded as the spatio-temporal baseline in evaluation.
3) LSTO [27] is a method for multi-image obstruction removal. This method can be extended to video waterdrop removal.

1) Qualitative Evaluation: We present some qualitative comparisons in Fig. 5, evaluated on real driving scenes which are collected from the Internet (without ground truth). For

TABLE I: Quantitative comparison on our proposed synthetic dataset.

	CCN [18]	Vid2Vid [33]	**Ours**
PSNR ↑	26.2878	27.5029	**29.5789**
MS-SSIM ↑	0.9220	0.9439	**0.9627**
LPIPS ↓	0.1896	0.1231	**0.0848**
E_{warp} [37] ↓	0.0828	0.0811	**0.0799**

various waterdrop cases, image methods show poor performance on video waterdrop removal, especially CCN [18], since the network architecture dedicated to the single-image task is too weak to handle such complex driving scenes. Although Vid2Vid [33] shows satisfying removing performance on the sparse and small waterdrops as shown in Fig. 5 column 4, this CNN-based method fails to address streak case with large waterdrops and it cannot restore the background information under the complete occlusions, as shown in Fig. 5 column 1. LSTO [27] can remove most waterdrops in real driving scenes, while this multi-image method relies heavily on flow estimation, which may fail when there are many waterdrops along a sequence of frames. Furthermore, this method is too time-consuming, which costs 60 seconds for each frame of size 960×512, while our method only costs 0.5 seconds for the same one. Based on our pixel attention block and spatio-temporal fusion, the proposed method shows outstanding waterdrop removal performance in sparse cases and streak cases. Even under dark circumstances and heavy rain, the proposed method still presents satisfying results in such extreme cases, as shown in Fig. 5 columns 2 and 5.

2) Quantitative Evaluation:

a) PSNR and MS-SSIM: Due to the lack of paired data of real driving scenes, we evaluate the different methods on the testing set of our synthetic dataset. Since the lack of training codes for other methods, we only compare the proposed method with CCN [18] and Vid2Vid [33], which are re-trained on the proposed dataset with the same cross-modality training strategy. As shown in Table I, we compute the PSNR and MS-SSIM [38] between the result cleaned frames of different methods and ground-truth clean frames. The proposed method shows strong performance over previous works on synthetic data.

b) User Study: We also conduct a user study for the results of different methods on real driving scenes. For each evaluation, we compare our method to the other five methods. Each user is presented with a driving-scene frame degraded by waterdrops and six cleaned frames from different methods. The user needs to choose the cleaned one which has better visual quality. There are 25 driving-scene frames presented in the comparisons and 30 users in this user study. The results are shown in Table II. There is nearly 77.84 percent of users prefer our results, which means our method outperforms others significantly.

c) Temporal Consistency: To evaluate the quality of results across the temporal dimension, we adopt a similar evaluation metric proposed by Lei *et al.* [37]. For each cleaned frame $\tilde{F}_t$, we calculate the warping error with $\tilde{F}_{t-1}$, $\tilde{F}_{t-3}$ and $\tilde{F}_{t-5}$ for considering multi-scale temporal consistency. As shown in Table III, our model without the temporal attention block suffers heavily from temporal inconsistency, which means the temporal attention block not only restores the regions with the clues from nearby frames but also keeps the network outputs temporally consistent.

TABLE II: User study for video waterdrop removal task.

AttentGAN [21]	Quan *et al.* [22]	CCN [18]	Vid2Vid [33]	LSTO [27]	Ours
4.54%	10.23%	0%	6.82%	0.56%	**77.84**%

TABLE III: Ablation study on our proposed synthetic dataset. The best and second-best scores are indicated in **red** and **blue**.

	Ours-noPAB	Ours-noSAB	Ours-noTAB	Ours-noCMS	**Ours (full)**
PSNR ↑	27.6397	28.2355	28.0135	**30.1052**	**29.5789**
MS-SSIM ↑	0.9456	0.9530	0.9492	**0.9625**	**0.9627**
LPIPS ↓	0.1209	0.1318	0.1363	**0.0740**	**0.0848**
E_{warp} [37] ↓	0.0815	0.0810	0.0831	**0.0800**	**0.0799**

(a) Input Frame (b) Ours-noCMS (c) Ours (full)

Fig. 6: Qualitative comparison on real driving scenes for our ablation study.

C. Ablation Study

To better analyze each block or cross-modality training strategy that contributes to the final performance of our network, we remove or replace each block one by one and re-train each modified network on the same datasets. As shown in Table. III, each attention block shows a positive effect for removing and recovering performance. Besides, although the model trained without cross-modality strategy (Ours-noCMS), which is only trained on our synthetic dataset, performs the best in quantitative evaluation in Table III, our full model shows better performance on real driving scenes, as shown in Fig. 6. The cleaned frame maintains the original color in good condition with our cross-modality training strategy, and some tiny waterdrops disappear.

VI. Conclusion

We present a new method with spatio-temporal fusion for video waterdrop removal. To train our proposed method, we build a large-scale synthetic waterdrop dataset for complex driving scenes. With an appropriate cross-modality training strategy, the proposed method shows an impressive waterdrop removal performance in real driving scenes. Furthermore, we are the first one to compare the proposed method with others on numerous synthetic data and real driving data. Both quantitative and qualitative evaluations show the proposed method is better than others.

REFERENCES

[1] T. Bruls, W. Maddern, A. A. Morye, and P. Newman, "Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1863–1870.

[2] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4644–4651.

[3] V. R. Kumar, S. Yogamani, H. Rashed, G. Sitsu, C. Witt, I. Leang, S. Milz, and P. Mäder, "Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2830–2837, 2021.

[4] Z. Zou, R. Zhang, S. Shen, G. Pandey, P. Chakravarty, A. Parchami, and H. X. Liu, "Real-time full-stack traffic scene perception for autonomous driving with roadside cameras," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 890–896.

[5] H. Porav, T. Bruls, and P. Newman, "I can see clearly now: Image restoration via de-raining," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7087–7093.

[6] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017.

[7] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE transactions on circuits and systems for video technology*, vol. 30, no. 11, pp. 3943–3956, 2019.

[8] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, "Hinet: Half instance normalization network for image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 182–192.

[9] R. Li, L.-F. Cheong, and R. T. Tan, "Heavy rain image restoration: Integrating physics model and conditional adversarial learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1633–1642.

[10] D. Chen, M. He, Q. Fan, J. Liao, L. Zhang, D. Hou, L. Yuan, and G. Hua, "Gated context aggregation network for image dehazing and deraining," in *2019 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2019, pp. 1375–1383.

[11] H. Wang, Q. Xie, Q. Zhao, and D. Meng, "A model-driven deep neural network for single image rain removal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3103–3112.

[12] Q. Yi, J. Li, Q. Dai, F. Fang, G. Zhang, and T. Zeng, "Structure-preserving deraining with residue channel prior guidance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4238–4247.

[13] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 821–14 831.

[14] C. Chen and H. Li, "Robust representation learning with feedback for single image deraining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7742–7751.

[15] H. Wang, Z. Yue, Q. Xie, Q. Zhao, Y. Zheng, and D. Meng, "From rain generation to rain removal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 791–14 801.

[16] W.-T. Chen, H.-Y. Fang, J.-J. Ding, and S.-Y. Kuo, "Pmhld: patch map-based hybrid learning dehazenet for single image haze removal," *IEEE Transactions on Image Processing*, vol. 29, pp. 6773–6788, 2020.

[17] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8346–8355.

[18] R. Quan, X. Yu, Y. Liang, and Y. Yang, "Removing raindrops and rain streaks in one go," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9147–9156.

[19] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *CVPR*, 2022.

[20] 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*, 2013, pp. 633–640.

[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*, 2018, pp. 2482–2491.

[22] Y. Quan, S. Deng, Y. Chen, and H. Ji, "Deep learning for seeing through window with raindrops," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2463–2471.

[23] Z. Shi, N. Fan, D.-Y. Yeung, and Q. Chen, "Stereo waterdrop removal with row-wise dilated attention," *IROS*, 2021.

[24] Z. Hao, S. You, Y. Li, K. Li, and F. Lu, "Learning from synthetic photorealistic raindrop for single image raindrop removal," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[25] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *International Conference on Learning Representations*, 2018.

[26] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi, "Adherent raindrop detection and removal in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1035–1042.

[27] Y.-L. Liu, W.-S. Lai, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Learning to see through obstructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 215–14 224.

[28] S. Alletto, C. Carlin, L. Rigazio, Y. Ishii, and S. Tsukizawa, "Adherent raindrop removal with self-supervised attention maps and spatio-temporal generative adversarial networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[30] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," *arXiv preprint arXiv:1801.07892*, 2018.

[31] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial-temporal transformations for video inpainting," in *European Conference on Computer Vision*, 2020, pp. 528–543.

[32] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[33] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, N. Yakovenko, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *NeurIPS*, 2018.

[34] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[35] 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.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[37] C. Lei, Y. Xing, and Q. Chen, "Blind video temporal consistency via deep video prior," in *Advances in Neural Information Processing Systems*, 2020.

[38] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.