Spatio-Temporal Deformable Attention Network for Video Deblurring

Huicong Zhang[1[0000−0001−5708−0855]], Haozhe Xie[2[0000−0001−9596−5179]], and Hongxun Yao[1[0000−0003−3298−2574]]

[1] Harbin Institute of Technology [2] Tencent AI Lab
https://vilab.hit.edu.cn/projects/stdan

Abstract. The key success factor of the video deblurring methods is to compensate for the blurry pixels of the mid-frame with the sharp pixels of the adjacent video frames. Therefore, mainstream methods align the adjacent frames based on the estimated optical flows and fuse the alignment frames for restoration. However, these methods sometimes generate unsatisfactory results because they rarely consider the blur levels of pixels, which may introduce blurry pixels from video frames. Actually, not all the pixels in the video frames are sharp and beneficial for deblurring. To address this problem, we propose the spatio-temporal deformable attention network (STDANet) for video delurring, which extracts the information of sharp pixels by considering the pixel-wise blur levels of the video frames. Specifically, STDANet is an encoder-decoder network combined with the motion estimator and spatio-temporal deformable attention (STDA) module, where motion estimator predicts coarse optical flows that are used as base offsets to find the corresponding sharp pixels in STDA module. Experimental results indicate that the proposed STDANet performs favorably against state-of-the-art methods on the GoPro, DVD, and BSD datasets.

Keywords: video deblurring, pixel-wise blur levels, spatio-temporal deformable attention

1 Introduction

In the past few years, hand-held image capturing devices, such as smartphones and action cameras, have been pervasive in our daily life. The camera shake and high-speed movement in dynamic scenes often generate undesirable blur in the video. The blurry video significantly reduces the visual quality and degrades performance in many subsequent vision tasks, including tracking [21,9], video stabilization [20], and SLAM [17]. Therefore, it is extremely attractive to develop an effective method to deblur videos for above mentioned human perception and high-level vision tasks.

Unlike image deblurring, video deblurring methods exploit additional information in the temporal domain. The key success factor of the video deblurring methods is to compensate for the blurry pixels of the mid-frame with the

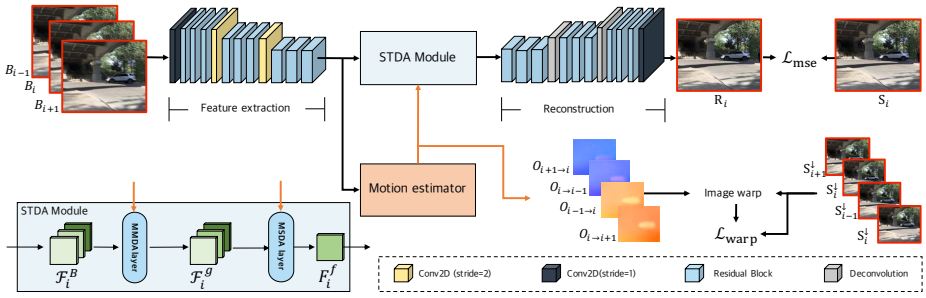


Fig. 1. The overview of STDANet, which takes three adjacent frames as input and restores the sharp mid-frame. Note that $\mathbf{S}_{i-1}^{\downarrow}$, $\mathbf{S}_{i}^{\downarrow}$, and $\mathbf{S}_{i+1}^{\downarrow}$ are the corresponding downsampled ground truth sharp frames of $\mathbf{S}_{i-1}$, $\mathbf{S}_{i}$, and $\mathbf{S}_{i+1}$, respectively.

sharp pixels of the adjacent video frames. Traditional video deblurring methods [12,1,3,38] often model motion blur by optical flow. Then those methods jointly estimate the optical flow and latent frames under the constraints by some hand-crafted priors.

Early deep learning methods [13,30,24,35] directly concatenate the multi-frames features to restore the mid-frame based on the CNN. However, those methods do not take full advantage of the information of the video frames because they explicitly considering the alignment of video frames. The recent mainstream deep learning methods [25,19] align the video frames by optical flows and directly generate the sharp frames by fusing aligned frames. However, they are less effective for the frames whose pixels contain large displacements because they may introduce blurry pixels that are not beneficial for blurring. EDVR [35] computes the pixel-wise similarity in multiple frames and restores the pixels in the mid-frame with high-similarity pixels in the video frames. However, the pixels of high similarity in the adjacent frames are also blurry for the blurry pixels in the mid-frame, which are not beneficial for deblurring.

To solve these issues, we propose spatio-temporal deformable attention network (STDANet), which extracts the information of sharp pixels by considering the pixel-wise blur levels of the video frames. Specifically, STDANet is based on an encoder-decoder network combined with motion estimator and spatio-temporal deformable attention (STDA) module. First, the encoder extracts the multi-features from multiple input frames. Then, the motion estimator predicts coarse optical flows between consecutive video frames given the multi-features generated by the encoder. After that, the estimated optical flows and the extracted features are fed to STDA module to generate the fused features by aggregating the information of the sharp pixels from the extracted multi-features. Different from recent methods [25,19], where the optical flows are used to align the adjacent frames, the optical flows are used as base offsets in the STDA module, which reduces the degradation of deblurring results caused by inaccurate

optical flows. Finally, the decoder restores the sharp mid-frame based on the fused features.

The main contributions are summarized as follows:

- We propose a spatio-temporal deformable attention (STDA) module which aggregates the information of sharp pixels in the input consecutive video frames and eliminates the effects of blurry pixels introduced from input consecutive video frames.
- We present a spatio-temporal deformable attention network (STDANet) equipped with motion estimator and the proposed STDA module, where motion estimator predicts coarse optical flows and provides base offsets to find sharp pixels in adjacent frames.
- We quantitatively and qualitatively evaluate STDANet on the DVD, GoPro, and BSD datasets. The experimental results indicate that STDANet performs favorably against state-of-the-art methods with comparable computational complexity.

2 Related Work

2.1 Single-Image Deblurring

The traditional single image deblurring methods [28,33,15,22,18] assume a uniform blur kernel and design various natural image priors to compensate for the ill-posed blur removal process. However, these methods do not have the ability to handle the non-uniform blur. To solve the non-uniform blur problem, one group of methods [7,6,36,8,40] extends the degree of freedom of the blur model from uniform to non-uniform in a limited way compared to the dense matrix. Another group of methods [2,10,12,11] introduces additional segmentations into blur models or adopt motion estimation-based deblurs.

With the development of deep learning, many CNN-based methods are proposed to solve dynamic scene deblurring. Gong [5] adopt a fully-convolutional deep neural network (FCN) to directly estimate the motion flow from the blurry image and restore the unblurred image from the estimated motion flow. Sun [32] use CNN to estimate the motion blur field. With the emergence of large datasets for single image deblurring, several works [34,23,41,16,26] use CNN to directly generate clear images from blurry images in an end-to-end manner. Nah [23] use a multi-scale method for single image deblurring. However, the parameters between each scale are not shared, which leads to a huge amount of parameters. To solve this problem, SRN [34] introduces a deblur network with skip connections where the parameters are shared in each scale. DeblurGAN-v2 [16] uses an end-to-end generative adversarial network (GAN) for single image motion deblurring and introduces the Feature Pyramid Network into single image deblurring. DMPHN [41] introduces the hierarchical multi-patch (MP) model for deblurring and improves deblur performance. MT-RNN [26] uses an RNN with recursive feature maps for progressive deblurring over iterations.

2.2 Multi-Image Deblurring

Several methods utilize multiple images to solve dynamic scene deblurring from videos. The traditional methods [12,1,3,38] jointly estimate the optical flow and blur kernel to restored frames with the some hand-crafted priors. However, the proposed priors usually lead to complex energy functions which are difficult to solve. In addition, Su [30] align the consecutive frames and then the Convolutional Neural Networks are used to restored images. Kim [13] propose a recurrent neural network to fuse the concatenation of the multi-frames features. Wieschollek [37] develop a recurrent network to recurrently use the features from the previous frame in multiple scales. Wang [35] achieve better alignment performance base on deformable convolution. Zhou [44] use the dynamic filters to align the consecutive frames. Pan [25] introduce a temporal sharpness prior to improve the ability of the deblur network. Zhang [42] develop a adversarial loss and spatial-temporal 3D convolutions to improve latent frame restoration. Recently, ARVo[19] uses self-attention to capture the pixel correlation of the consecutive frames. However, those methods rarely consider the different blur levels of each frame, which make they do not take full advantage of the sharpness pixel information in the video frames.

3 The Proposed Method

The proposed STDANet aims to restore the sharp mid-frame $\mathbf{R}_i$ given three consecutive blurry frames $\mathcal{B}_i = \{\mathbf{B}_k\}_{k=i-1}^{i+1}$. As shown in Figure 1, it contains four components: the feature extraction network, the motion estimator, the STDA module, and the reconstruction network, where the feature extraction network and the reconstruction network follow the encoder-decoder architecture. First, the feature extraction network generates the extracted features $\mathcal{F}_i^b = \{\mathbf{F}_k^b\}_{k=i-1}^{i+1}$ for $\mathcal{B}_i$. Then, the motion estimator predicts the optical flows $\mathcal{O}_i = \{\mathbf{O}_{k\to k+1} | k = i-1, i\} \cup \{\mathbf{O}_{k+1\to k} | k = i-1, i\}$ between the two adjacent frames $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$. Next, the STDA module takes $\mathcal{F}_i^b$, $\mathcal{O}_i$ as input and generates the fused features $\mathbf{F}_i^f$ by aggregating the features of low-blur-level pixels in the consecutive frames. Finally, the reconstruction network restores the sharp frame $\mathbf{R}_i$ for $\mathbf{B}_i$. Except STDANet, we also propose STDANet-Stack, which uses a cascaded strategy [25] to stack STDANet and takes five adjacent blurry frames $\{\mathbf{B}_k\}_{k=i-2}^{i+2}$ as input.

3.1 Motion Estimator

Previous video deblurring methods [25,19] that use optical flows to align two adjacent frames to the mid-frame, which requires accurate optical flows generated by heavyweight neural networks such as PWC-Net [31]. In contrast, optical flows are used as the base offsets in the STDA module, which are more robust to the errors in estimated optical flows. Therefore, we propose the motion estimator that predicts coarse optical flows between two adjacent frames with much

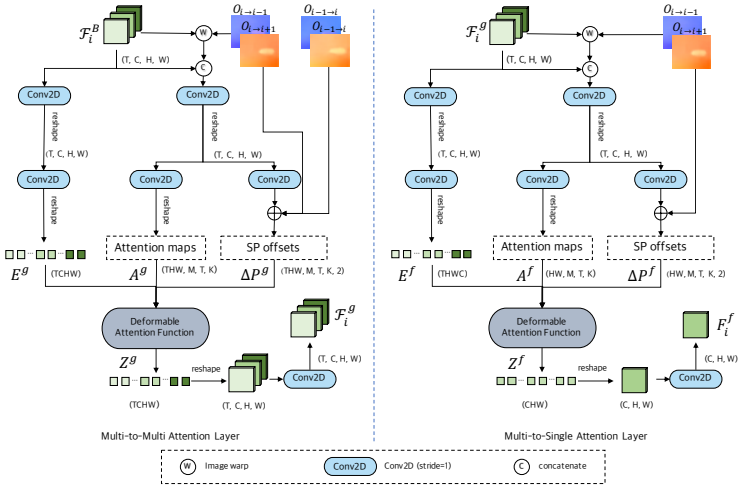


Fig. 2. The detailed network structure of the MMA and MSA layers. Note that "SP Offsets" denotes "the offsets of sampling points".

smaller computational complexity. To accelerate the computational complexity, the motion estimator generates the optical flows that are of 1/4 sizes the input images. Consequently, the motion estimator is 1/70 the size of PWC-Net. Compared to existing methods for optical flow estimation [4,31,39], the motion estimator does not use any time-consuming layers such as correlation layer [4], cost volume layer [31,39].

Specifically, the motion estimator consists of stacked four convolutional layers with kernel sizes of 3 and strides of 1. Given the three adjacent image features $\mathcal{F}_i^b$, the motion estimator generates four optical flows $\mathcal{O}_i = \{\mathbf{O}_{k\to k+1} | k = i-1, i\} \cup \{\mathbf{O}_{k+1\to k} | k = i-1, i\}$, where $\mathbf{O}_{m\to n}$ represents the optical flow from the m-th frame to the n-th frame.

3.2 Spatio-temporal Deformable Attention Module

To extract the information of sharp pixels from consecutive video frames, we propose spatio-temporal deformable attention (STDA) module. As shown in Figure 2, there are two layers in the STDA module that aggregates features in a coarse-to-fine manner, named Multi-to-Multi attention (MMA) layer and Multi-to-Single attention (MSA) layer. Figure 3 gives an illustration how the MMA and MSA layers extract image features of sharp pixels.

Multi-to-Multi Attention Layer The multi-to-multi attention layer takes the image features of three consecutive frames $\mathcal{F}_i^b$ as input and generates the coarse aggregated image features $\mathcal{F}_i^g = \left\{\mathbf{F}_k^g | \mathbf{F}_k^g \in \mathbb{R}^{C\times H\times W}\right\}_{k=i-1}^{i+1}$, where C, H, and W represent the number of channels, height, and width of the image features, respectively.

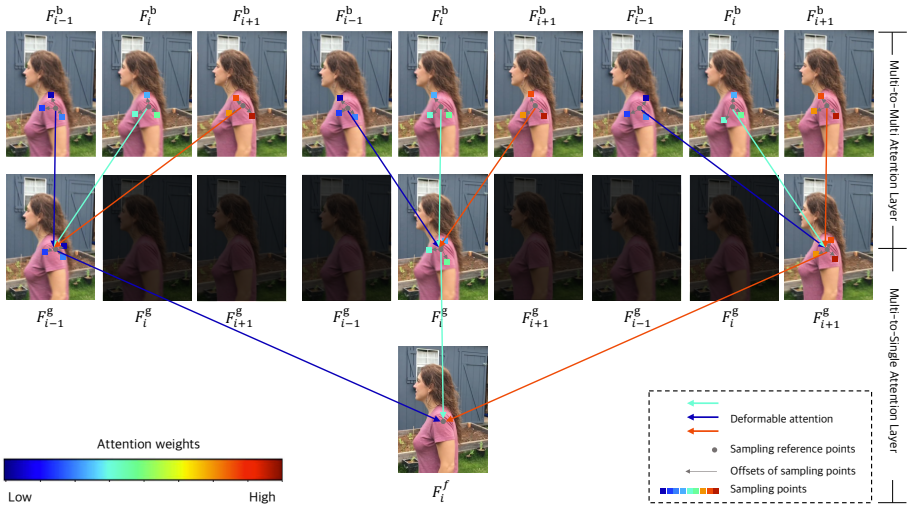


Fig. 3. The illustration of MMA and MSA layers. The colors of the sampling points denotes the corresponding attention weights, where higher attention weights indicate that the sampling points are sharper. First, the MMA layer extracts the information of sharp pixels from multi-features $\mathcal{F}_i^b$ and generates the features of adjacent frames $\mathcal{F}_i^g = \{\mathbf{F}_k^g | \mathbf{F}_k^g\}_{k=i-1}^{i+1}$. Second, the MSA layer generates the fused features $\mathbf{F}_i^f$ by aggregating the information of sharp pixels from $\mathcal{F}_i^g$.

Step 1. The image features $\mathcal{F}_i^b = \left\{\mathbf{F}_k^b | \mathbf{F}_k^b \in \mathbb{R}^{C\times H\times W}\right\}_{k=i-1}^{i+1}$ of adjacent frames are aligned to the mid-frame with the estimated optical flows $\mathcal{O}_i$ and produces $\mathcal{F}_i^w = \left\{\mathbf{F}_k^w | \mathbf{F}_k^w \in \mathbb{R}^{C\times H\times W}\right\}$, where $\mathbf{F}_k^w$ is

$$\mathbf{F}_k^w = \text{Warp}(\mathbf{F}_k^b, \mathbf{O}_{i\to k}), k = i-1, i+1 \tag{1}$$

where "Warp" denotes the backward warp with operation.

Step 2. The concatenated features $\mathbf{F}_i^c \in \mathbb{R}^{(2T-1)\times C\times H\times W}$ are generated by concatenating the aligned features $\mathcal{F}_i^w$ and image features $\mathcal{F}_i^b$, where T denotes the number of frames in the sliding window.

Step 3. Given $\mathbf{F}_i^c$ and $\mathcal{F}_i^b$ as input, the attention maps $\mathbf{A}^g \in \mathbb{R}^{Q\times M\times T\times K}$, the offsets of sampling points $\Delta\mathbf{P}^g \in \mathbb{R}^{Q\times M\times T\times K\times 2}$, and the flatten features $\mathbf{E}^g \in \mathbb{R}^{THWC}$ are generated, where $Q = THW$. M, T, and K represent the number of attention heads, the number of sampling points, and the number of frames, respectively. The attention maps $\mathbf{A}^g$ are used to measure the sharpness of the pixels, which are normalized by $\sum_{t=1}^{T}\sum_{k=1}^{K}\mathbf{A}_{mtqk}^g = 1$. The offsets of sampling points $\Delta\mathbf{P}^g$ and estimated optical flows $\mathcal{O}_i$ are used to find the corresponding pixels in the adjacent frames, where $\mathcal{O}_i$ provides the base offsets. $\Delta\mathbf{P}^g$, $\mathbf{A}^g$, and $\mathcal{O}_i$ are generated as following

$$\begin{aligned}\Delta\mathbf{P}^g &= \mathcal{C}^o_{\text{MMA}}(\mathbf{F}^c_i)\\ \mathbf{A}^g &= \mathcal{C}^m_{\text{MMA}}(\mathbf{F}^c_i)\\ \mathbf{E}^g &= \text{Concat}(\mathcal{C}^l_{\text{MMA}}(\mathbf{F}^b_{i-1}), \mathcal{C}^l_{\text{MMA}}(\mathbf{F}^b_i), \mathcal{C}^l_{\text{MMA}}(\mathbf{F}^b_{i+1}))\end{aligned} \tag{2}$$

where "Concat" denotes the concatenation operation. $\mathcal{C}^m_{\text{MMA}}, \mathcal{C}^o_{\text{MMA}}, \mathcal{C}^l_{\text{MMA}}$ represent different convolution layers. The attention map $\mathbf{A}^g$, offsets of sampling points $\Delta\mathbf{P}^g$ and flatten features $\mathbf{E}^g$ are fed to the deformable attention function $\mathcal{D}$ [45] and produces the fused features $\mathbf{Z}^g \in \mathbb{R}^{TCHW}$.

$$\mathbf{Z}^g = \mathcal{D}(\mathbf{A}^g, \phi(\Delta\mathbf{P}^g, \mathcal{O}_i), \mathbf{E}^g), \tag{3}$$

where ϕ represents the operation that adds the estimated optical flows to the offsets of sampling points $\Delta\mathbf{P}^g$. In 3, the optical flows is used as based offsets, which reduces the degradation of deblurring results caused by inaccurate optical flows.

Step 4. $\mathbf{Z}^g$ is reshaped and splitted into $\left\{\mathbf{F}^h_k | \mathbf{F}^h_k \in \mathbb{R}^{C\times H\times W}\right\}^{i+1}_{k=i-1}$. The final fused features $\mathcal{F}^g_i = \left\{\mathbf{F}^g_k | \mathbf{F}^g_k \in \mathbb{R}^{C\times H\times W}\right\}^{i+1}_{k=i-1}$ are generated as following

$$\mathbf{F}^g_k = \mathcal{C}^g_{\text{MMA}}(\mathbf{F}^h_k) \tag{4}$$

where $\mathcal{C}^g_{\text{MMA}}$ denotes a convolutional layer.

Multi-to-Single Attention Layer The multi-to-single attention layer takes the coarse aggregated image features $\mathcal{F}^g_i$ as input and generates the fused features $\mathbf{F}^f_i$ for the mid-frame. Similar to the MMA layer, the MSA layer aggregates information of sharp pixels from the adjacent frames. However, in the MSA layer, the aggregated features are only propagated to the mid-frame. Therefore, in the MSA layer, the fused features $\mathbf{Z}^f \in \mathbb{R}^{CHW}$ is generated as following

$$\mathbf{Z}^f = \mathcal{D}(\mathbf{A}^f, \phi(\Delta\mathbf{P}^f, \{\mathbf{O}_{k\to i} | k = i-1, i+1\}), \mathbf{E}^f) \tag{5}$$

where $\mathbf{A}^f \in \mathbb{R}^{HW\times M\times T\times K}$, $\Delta\mathbf{P}^f \in \mathbb{R}^{HW\times M\times T\times K\times 2}$, and $\mathbf{E}^f \in \mathbb{R}^{TCHW}$ are the attention maps, the offsets of sampling points, and flatten features obtained as in the MMA layer. The fused features $\mathbf{F}^f_i$ is obtained as following

$$\mathbf{F}^f_i = \mathcal{C}^f_{\text{MSA}}(\mathbf{F}^n_i) \tag{6}$$

where $\mathcal{C}^f_{\text{MSA}}$ denotes a convolutional layer. $\mathbf{F}^n_i \in \mathbb{R}^{C\times H\times W}$ is reshaped from $\mathbf{Z}^f$.

3.3 Feature Extraction and Reconstruction Networks

Feature Extraction Network. The feature extraction network generates image features $\mathcal{F}^b_i$ from blurry images $\mathcal{B}_i$. It consists of three convolutional blocks, two of which have a convolution layer with the stride of 2 followed by three

residual blocks with LeakyReLU as the activation function. The first convolutional block has a convolution layer with the stride of 1 followed by three residual blocks with LeakyReLU as the activation function.

Reconstruction Network. The reconstruction network is used to restore the sharp mid-frame $\mathbf{R}_i$ by taking the fused features from STDA module as input. It consists of three convolutional blocks, two of which have one deconvolutional layer with the stride of 2 and three residual blocks with LeakyReLU as the activation function. The last convolutional block has one convolutional layer with the stride of 1 and three residual blocks with LeakyReLU as the activation function.

3.4 Cascaded Progressive Deblurring

Inspired by TSP [25], we propose STDANet-Stack by stacking STDANet in a cascaded manner [25]. It takes five adjacent blurry video frames $\{\mathbf{B}_k\}_{k=i-2}^{i+2}$ as input and restores the sharp mid-frame $\mathbf{R}_i$.

Specifically, STDANet-Stack restores $\mathbf{R}_i$ in two steps. First, it produces $\hat{\mathbf{R}}_{i-1}$ by taking $\{\mathbf{B}_k\}_{k=i-2}^{i}$ as input. Similarly, $\hat{\mathbf{R}}_i$ and $\hat{\mathbf{R}}_{i+1}$ are restored by taking $\{\mathbf{B}_k\}_{k=i-1}^{i+1}$ and $\{\mathbf{B}_k\}_{k=i}^{i+2}$ as inputs, respectively. Next, $\mathbf{R}_i$ is generated by taking $\left\{\hat{\mathbf{R}}_k\right\}_{k=i-1}^{i+1}$ as input.

3.5 Loss Functions

We employ two loss functions to train STDANet and STDFANet-Stack.

MSE Loss represents the distance between the restored frame R and its corresponding ground truth sharp frame S, which is formulated as

$$\mathcal{L}_{mse} = \| \mathbf{R} - \mathbf{S} \|^2 \tag{7}$$

Warp Loss is introduced to train the motion estimator in an unsupervised manner, which is computed as

$$\mathcal{L}_{warp} = \| \mathbf{S}_i^{\downarrow} - \mathrm{Warp}(\mathbf{S}_j^{\downarrow}, \mathbf{O}_{i \to j}) \|^2 \tag{8}$$

where $\mathbf{S}_i^{\downarrow}$ and $\mathbf{S}_j^{\downarrow}$ are the two downsampled frames. $\mathbf{O}_{i \to j}$ represents the estimated optical flow from $\mathbf{S}_j^{\downarrow}$ and $\mathbf{S}_i^{\downarrow}$. "Warp" denotes the backward warp operation.

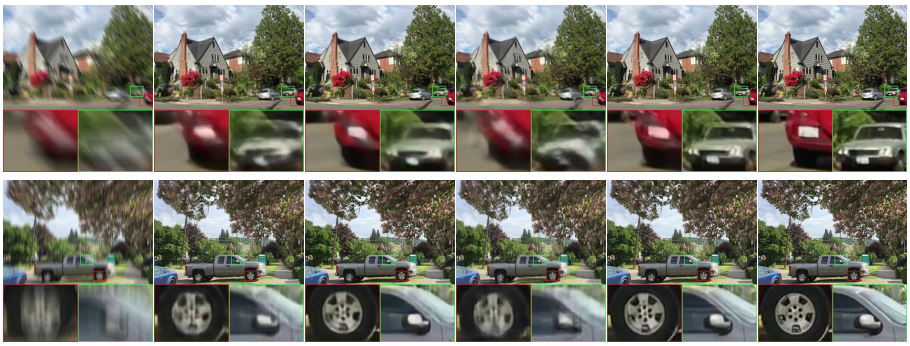
Total Loss are defined as

$$\mathcal{L}_{total} = \mathcal{L}_{mse} + \gamma \mathcal{L}_{warp} \tag{9}$$

where γ controls the weights of the two loss functions.

Table 1. The quantitative results on the DVD dataset. Note that "Ours*" denotes STDANet-Stack.

Method	SRN	IFI-RNN-L	STFAN	EDVR	TSP	PVDNet	ARVo	Ours	**Ours***
PSNR	30.53	31.67	31.24	31.82	32.13	32.31	32.80	32.63	**33.05**
SSIM	0.8940	0.9160	0.9340	0.9160	0.9270	0.9260	0.9352	0.9300	**0.9374**



(a) Input (b) EDVR (c) TSP (d) PVDNet-L (e) Ours (f) GT

Fig. 4. The qualitative results on the DVD dataset. Note that "GT" stands for ground truth.

4 Experiments

4.1 Datasets

DVD. The DVD dataset [30] contains 71 videos (6,708 blurry-sharp pairs), splitting into 61 training videos (5,708 pairs) and 10 testing videos (1,000 pairs).
GoPro. The GoPro dataset [23] contains 3,214 pairs of blurry images and sharp images at 1280×720 resolution, where 2,103 and 1,111 pairs of blurry images and sharp images are used for training and testing, respectively.
BSD. The BSD dataset [43] is a real-world video deblur dataset, which contains three sub-datasets with different sharp exposure time - blurry exposure time.

4.2 Evaluation Metrics

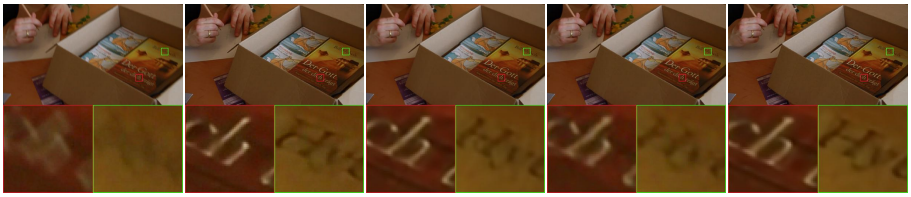
For fair comparisons, STDANet-Stack use same cascaded progressive structure like TSP [25] and ARVo [19]. In the experiments, we use both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) as quantitative evaluation metrics for testing set. Moreover, GMACs (Giga multiply-add operations per second) is used to evaluate the computational complexity.

4.3 Implementation Details

To achieve better trade-off between video deblurring quality and computational efficiency, the M, K, T are set as 4, 12 and 3, respectively. γ is set to 0.05.

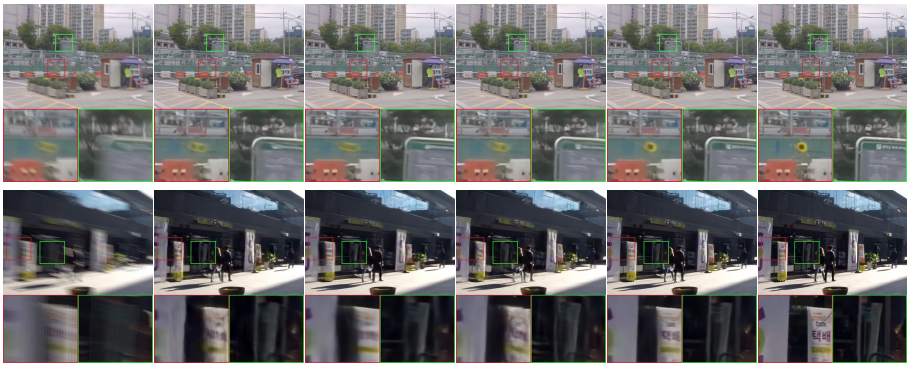
Table 2. The quantitative results on the GoPro dataset. Note that "Ours*" denotes STDANet-Stack.

Method	SRN	IFI-RNN-L	STFAN	EDVR	TSP	PVDNet	PVDNet-L	**Ours**	**Ours***
PSNR	30.61	31.05	28.59	31.54	31.67	31.52	31.98	**32.29**	**32.62**
SSIM	0.9080	0.9110	0.8608	0.9260	0.9279	0.9210	0.9280	**0.9313**	**0.9375**
GMACs	1175	1,425	504	2739	6450	1004	1755	1677	6000



(a) Input (b) EDVR (c) TSP (d) PVDNet (e) Ours

Fig. 5. The qualitative results of real blur images from the DVD dataset. There are no corresponding ground truth for the real blur images.



(a) Input (b) EDVR (c) TSP (d) PVDNet-L (e) Ours (f) GT

Fig. 6. The qualitative results on the GoPro dataset. Note that "GT" stands for ground truth.

The network is implemented with PyTorch [27] [1]. The network is trained with a batch size of 8 on four NVIDIA Geforce RTX 2080 Ti GPUs. The initial learning rate is set to 10^{-4}. The network is optimized using Adam optimizer [14] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We randomly crop the input images into patches with resolutions of 256×256, along with random flipping or rotation during training.

[1] The source code is available at https://github.com/huicongzhang/STDAN

Table 3. The quantitative results on the BSD dataset. Note that "Ours*" denotes STDANet-Stack.

Method	1ms–8ms		2ms–16ms		3ms–24ms	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
IFIRNN	33.00	0.9330	31.53	0.9190	30.89	0.9170
ESTRNN	33.36	0.9370	31.95	0.9250	31.39	0.9260
EDVR	32.79	0.9264	31.99	0.9129	31.53	0.9192
TSP	33.62	0.9419	32.19	0.9285	31.68	0.9266
PVDNet-L	33.93	0.9392	32.46	0.9290	31.87	0.9293
Ours	**34.21**	**0.9446**	**33.13**	**0.9388**	**32.65**	**0.9409**
Ours*	**34.32**	**0.9456**	**33.27**	**0.9420**	**32.83**	**0.9443**

Table 4. The quantitative results on the GoPro dataset in terms of PSNR and SSIM when replacing MMA and MSA layers with the concatenation operation.

MMA			✓	✓
MSA		✓		✓
PSNR	30.12	31.15	31.18	**32.29**
SSIM	0.8950	0.9146	0.9152	**0.9313**

4.4 Experimental Results

The DVD dataset. To evaluate the performance of the proposed method, we compare it with the state-of-the-art methods. Table 1 shows the quantitative results on the DVD dataset [30], where IFI-RNN-L [29] is larger IFI-RNN [24]. The proposed method outperforms the state-of-the-art methods in term of PSNR and SSIM. Compared to the best state-of-the-art method ARVo, the proposed STDANet-Stack improves the PSNR and SSIM by 0.25dB and 0.0022, respectively. Figure 4 shows several examples in the testing set, which indicates that existing state-of-the-art methods are less effective when the inputs contain heavy blur. We further compare the proposed method with state-of-the-art methods on the real blur images from the DVD dataset. Figure 5 shows that the proposed method generates sharper images with more visual details, which demonstrates the superiority of removing the unknown real blur in dynamic scenes robustly.

The GoPro dataset. We compare STDANet to the state-of-the-art video deblurring methods on the GoPro dataset [23]. As show in Table 2, the proposed STDANet and STDANet-Stack perform favorably against the state-of-the-art methods in terms of PSNR and SSIM. Compared to the PVDNet-L [29], STDANet achieves higher PSNR and SSIM with lower computational complexity. STDANet-Stack achieves 0.95dB higher PSNR than TSP [25] with lower computational complexity, where the STDANet-Stack use the same cascaded progressive structure as TSP [25]. As shown in Figure 6, the proposed method restores better image details and structures.

(a) Input (b) EDVR (c) TSP (d) PVDNet-L (e) Ours (f) GT

Fig. 7. The qualitative results on the 2ms-16ms subset of the BSD dataset. Note that "GT" stands for ground truth.

(a) Input (b) (-MMA,-MSA) (c) (-MMA,+MSA) (d) (+MMA,-MSA) (e) (+MMA,+MSA) (f) GT

Fig. 8. The qualitative results when replacing MMA and MSA layers with the concatenation operation. Note that "+" and "-" denote "with" and "without", respectively. "GT" stands for ground truth.

The BSD dataset. We compared the our method to the state-of-the-art methods on BSD dataset [43]. For a fair comparison, the EDVR [35], TSP [25], and PVDNet-L [29] are trained with their open-sourced implementations. In Table 3, our method achieves the best results on all the three subsets in terms of PSNR and SSIM. The qualitative results are shown in Figure 7, which indicate that our method restores much sharper images.

5 Analysis and Discussions

5.1 Effectiveness of the STDA Module

MMA and MSA layers. The STDA Module contains two main components: the MMA and MSA layers, which aggregates information of sharp pixels from adjacent frames. To validate the effectiveness of the STDA module, the MMA and MSA layers are replaced with the concatenate operation. In the concatenate operation, the information from all pixels are introduced from adjacent frames. Table 4 shows the qualitative comparison when the MMA or MSA layer is removed. Specially, when both MMA or MSA layers are removed, the estimated

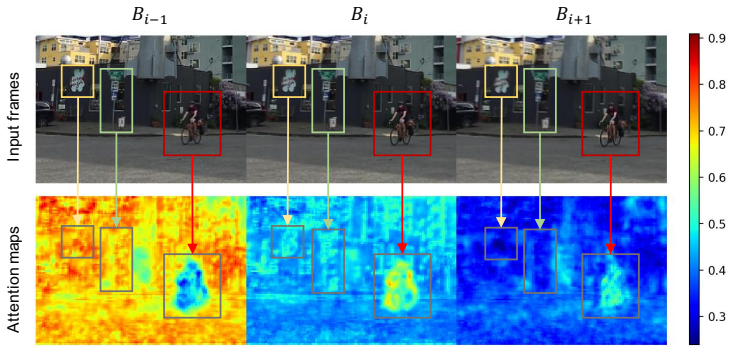


Fig. 9. The visualization of the attention maps in the MSA layer. Sharper pixels have larger attention weights. **(zoom in for best view)**.

Table 5. The quantitative results on the GoPro dataset in terms of PSNR, SSIM, and GMACs with different numbers of sampling points.

#Sampling Points	$K = 1$	$K = 8$	$K = 12$	$K = 16$
PSNR	31.64	32.12	32.29	**32.32**
SSIM	0.9183	0.9288	0.9313	**0.9319**
GMACs	**1520**	1620	1677	1735

optical flows are used to align the features from adjacent frames. The experimental results shows that the networks perform worse without the help of the information of sharp pixels extracted by the MMA and MSA layers. Figure 8 shows the qualitative comparison on the GoPro dataset. The network is less effective to restore sharp details when both MMA and MSA layers are removed. Figure 9 gives the visualization of the attention maps in the MSA layer, which shows that sharper pixels have larger attention weights. For example, the man riding a bicycle (highlighted with a red bounding box) is blurry in B_{i-1}, and thus the corresponding regions are of low weights in the attention maps. In contrast, B_i have larger weights for this region. To conclude, the proposed STDA module effectively aggregates the information of sharp pixels from adjacent frames.

Sampling Points. To investigate the effect of numbers of sampling points in the STDA module, we compare the performance with different numbers of sampling points. As shown in Table 5, larger number of sampling points leads to better restoration results but also heavier computational cost. Specially, the STDA Module degenerates to the temporal attention when $K = 1$, which causes severe degeneration in restoration results. The PSNR only increases 0.03 dB when increasing the number of sampling points from 12 to 16. Therefore, we set $K = 12$ due to the trade-off between the computational cost and restoration performance.

Attention Heads. The number of attention heads is one of the important hyperparameter in the deformable attention function. We also compare the effect

Table 6. The quantitative results on the GoPro dataset in terms of PSNR, SSIM, and GMACs with different numbers of attention heads.

#Attention Heads	$M = 1$	$M = 4$	$M = 8$
PSNR	32.13	32.29	**32.34**
SSIM	0.9294	0.9313	**0.9322**
GMACs	**1548**	1677	1849

Table 7. The quantitative results on the GoPro dataset in terms of PSNR, SSIM, and GMACs with different optical flow estimators.

Estimator	None	PWC-Net	Motion Estimator
PSNR	31.58	**32.36**	32.29
SSIM	0.9176	**0.9326**	0.9313
GMACs	**1632**	2352	1677

with different numbers of attention heads in Table 6. As the number of attention heads increases, the PSNR, SSIM, and GMACs increase. Considering the trade-off between the computational complexity and restoration performance, we choose the number of attention heads $M = 4$.

5.2 Effectiveness of the Motion Estimator

To evaluate the effectiveness of the motion estimator, we compare the video deblur results with different optical flow estimators. As shown in Table 7, removing the optical flow estimator causes considerable degeneration. Although STDANet with PWC-Net [31] archives the best results, it also leads to high computational cost. STDANet with the proposed motion estimator archives the best trade-off between the deblur results and computational complexity.

6 Conclusions

In this paper, we propose STDANet for video deblurring. The main motivation of this work is that not all the pixels in the video frames are sharp and beneficial for deblurring. Therefore, the proposed STDANet extracts the information of sharp pixels by considering the pixel-wise blur levels of the video frames. Different from mainstream video debulr methods that requires accurate optical flows to align two adjacent frames to the mid-frame, the coarse optical flows are estimated by a lightweight motion estimator and are used as the base offsets to find the corresponding sharp pixels in the adjacent frames. Experimental results indicate that the proposed STDANet performs favorably against state-of-the-art methods on the GoPro, DVD, and BSD datasets.

Acknowledgement. This work is supported by the National Key R&D Program of China (No. 2021ZD0110901).

References

1. Bar, L., Berkels, B., Rumpf, M., Sapiro, G.: A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In: ICCV (2007) 2, 4
2. Couzinie-Devy, F., Sun, J., Alahari, K., Ponce, J.: Learning to estimate and remove non-uniform image blur. In: CVPR (2013) 3
3. Dai, S., Wu, Y.: Motion from blur. In: CVPR (2008) 2, 4
4. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: ICCV (2015) 5
5. Gong, D., Yang, J., Liu, L., Zhang, Y., Reid, I.D., Shen, C., van den Hengel, A., Shi, Q.: From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In: CVPR (2017) 3
6. Gupta, A., Joshi, N., Zitnick, C.L., Cohen, M.F., Curless, B.: Single image deblurring using motion density functions. In: ECCV (2010) 3
7. Harmeling, S., Hirsch, M., Schölkopf, B.: Space-variant single-image blind deconvolution for removing camera shake. In: NIPS (2010) 3
8. Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: ICCV (2011) 3
9. Jin, H., Favaro, P., Cipolla, R.: Visual tracking in the presence of motion blur. In: CVPR (2005) 1
10. Kim, T.H., Ahn, B., Lee, K.M.: Dynamic scene deblurring. In: ICCV (2013) 3
11. Kim, T.H., Lee, K.M.: Segmentation-free dynamic scene deblurring. In: CVPR (2014) 3
12. Kim, T.H., Lee, K.M.: Generalized video deblurring for dynamic scenes. In: CVPR (2015) 2, 3, 4
13. Kim, T.H., Lee, K.M., Schölkopf, B., Hirsch, M.: Online video deblurring via dynamic temporal blending network. In: ICCV (2017) 2, 4
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 10
15. Krishnan, D., Tay, T., Fergus, R.: Blind deconvolution using a normalized sparsity measure. In: CVPR (2011) 3
16. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In: ICCV (2019) 3
17. Lee, H.S., Kwon, J., Lee, K.M.: Simultaneous localization, mapping and deblurring. In: ICCV (2011) 1
18. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient marginal likelihood optimization in blind deconvolution. In: CVPR (2011) 3
19. Li, D., Xu, C., Zhang, K., Yu, X., Zhong, Y., Ren, W., Suominen, H., Li, H.: Arvo: Learning all-range volumetric correspondence for video deblurring. In: CVPR (2021) 2, 4, 9
20. Matsushita, Y., Ofek, E., Ge, W., Tang, X., Shum, H.: Full-frame video stabilization with motion inpainting. TPAMI **28**(7), 1150–1163 (2006) 1
21. Mei, C., Reid, I.D.: Modeling and generating complex motion blur for real-time tracking. In: CVPR (2008) 1
22. Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: ECCV (2014) 3
23. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (2017) 3, 9, 11

24. Nah, S., Son, S., Lee, K.M.: Recurrent neural networks with intra-frame iterations for video deblurring. In: CVPR (2019) 2, 11
25. Pan, J., Bai, H., Tang, J.: Cascaded deep video deblurring using temporal sharpness prior. In: CVPR (2020) 2, 4, 8, 9, 11, 12
26. Park, D., Kang, D.U., Kim, J., Chun, S.Y.: Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In: ECCV (2020) 3
27. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019) 10
28. Ren, W., Cao, X., Pan, J., Guo, X., Zuo, W., Yang, M.: Image deblurring via enhanced low-rank prior. TIP (2016) 3
29. Son, H., Lee, J., Lee, J., Cho, S., Lee, S.: Recurrent video deblurring with blur-invariant motion estimation and pixel volumes. ACM Trans. Graph. (2021) 11, 12
30. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: CVPR (2017) 2, 4, 9, 11
31. Sun, D., Yang, X., Liu, M., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR (2018) 4, 5, 14
32. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: CVPR (2015) 3
33. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: ICCP (2013) 3
34. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR (2018) 3
35. Wang, X., Chan, K.C.K., Yu, K., Dong, C., Loy, C.C.: EDVR: video restoration with enhanced deformable convolutional networks. In: CVPR Workshops (2019) 2, 4, 12
36. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. In: CVPR (2010) 3
37. Wieschollek, P., Hirsch, M., Schölkopf, B., Lensch, H.P.A.: Learning blind motion deblurring. In: ICCV (2017) 4
38. Wulff, J., Black, M.J.: Modeling blurred video with layers. In: Fleet, D.J., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV (2014) 2, 4
39. Xu, J., Ranftl, R., Koltun, V.: Accurate optical flow via direct cost volume processing. In: CVPR (2017) 5
40. Xu, L., Zheng, S., Jia, J.: Unnatural L0 sparse representation for natural image deblurring. In: CVPR (2013) 3
41. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: CVPR (2019) 3
42. Zhang, K., Luo, W., Zhong, Y., Ma, L., Liu, W., Li, H.: Adversarial spatio-temporal learning for video deblurring. TIP (2019) 4
43. Zhong, Z., Gao, Y., Zheng, Y., Zheng, B.: Efficient spatio-temporal recurrent neural network for video deblurring. In: ECCV (2020) 9, 12
44. Zhou, S., Zhang, J., Pan, J., Zuo, W., Xie, H., Ren, J.S.J.: Spatio-temporal filter adaptive network for video deblurring. In: ICCV (2019) 4
45. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: ICLR (2021) 7