Blur Interpolation Transformer for Real-World Motion from Blur

Zhihang Zhong[1,2] Mingdeng Cao[1] Xiang Ji[1,] Yinqiang Zheng[1] Imari Sato[1,2]
[1]The University of Tokyo, Japan [2]National Institute of Informatics, Japan
zhong@is.s.u-tokyo.ac.jp {cmd,jixiang}@g.ecc.u-tokyo.ac.jp
yqzheng@ai.u-tokyo.ac.jp imarik@nii.ac.jp

Abstract

This paper studies the challenging problem of recovering motion from blur, also known as joint deblurring and interpolation or blur temporal super-resolution. The challenges are twofold: 1) the current methods still leave considerable room for improvement in terms of visual quality even on the synthetic dataset, and 2) poor generalization to real-world data. To this end, we propose a blur interpolation transformer (BiT) to effectively unravel the underlying temporal correlation encoded in blur. Based on multi-scale residual Swin transformer blocks, we introduce dual-end temporal supervision and temporally symmetric ensembling strategies to generate effective features for time-varying motion rendering. In addition, we design a hybrid camera system to collect the first real-world dataset of one-to-many blur-sharp video pairs. Experimental results show that BiT has a significant gain over the state-of-the-art methods on the public dataset Adobe240. Besides, the proposed real-world dataset effectively helps the model generalize well to real blurry scenarios. Code and data are available at https://github.com/zzh-tech/BiT.

1. Introduction

Aside from time-lapse photography, motion blur is usually one of the most undesirable artifacts during photo shooting. Many works have been devoted to studying how to recover sharp details from the blur, and great progress has been made. Recently, starting from Jin *et al*. [9], the community has focused on the more challenging task of recovering high-frame-rate sharp videos from blurred images, which can be collectively termed joint deblurring and interpolation [37, 38] or blur temporal super-resolution [26, 33–35]. This joint task can serve various applications, such as video visual perception enhancement, slow motion generation [26], and fast moving object analysis [33–35]. For brevity, we will refer to this task as blur interpolation.

Recent works [7, 8, 37] demonstrate that the joint approach outperforms schemes that cascade separate deblurring and video frame interpolation methods. Most joint approaches follow the center-frame interpolation pipeline, which means that they can only generate latent frames for middle moments in a recursive manner. DeMFI [26] breaks this constraint by combining self-induced feature-flow-based warping and pixel-flow-based warping to synthesize latent sharp frame at arbitrary time t. However, even on synthetic data, the performance of current methods is still far from satisfactory for human perception. We find that the potential temporal correlation in blur has been underutilized, which allows huge space for performance improvement of the blur interpolation algorithm. In addition, blur interpolation suffers from the generalization issue because there is no real-world dataset to support model training.

The goal of this work is to resolve the above two issues. In light of the complex distribution of time-dependent reconstruction and temporal symmetry property, we propose dual-end temporal supervision (DTS) and temporally symmetric ensembling (TSE) strategies to enhance the shared temporal features of blur interpolation transformer (BiT) for time-varying motion reconstruction. In addition, a multi-scale residual Swin transformer block (MS-RSTB) is introduced to empower the model with the ability to effectively handle the blur in different scales and to fuse information from adjacent frames. Due to our design, BiT achieves state-of-the-art on the public benchmark performance even without optical flow-based warping operations. Meanwhile, to provide a real-world benchmark to the community, we further design an accurate hybrid camera system following [32, 51] to capture a dataset (RBI) containing time-aligned low-frame-rate blurred and high-frame-rate sharp video pairs. Thanks to RBI, the real data generalization problem of blur interpolation can be greatly alleviated, and a more reasonable evaluation platform becomes available. With these improvements, our model presents impressive arbitrary blur interpolation performance, and we show an example of extracting 30 frames of sharp motion from the blurred image in Fig. 1 for reference.

Our contributions can be summarized as follows: 1) We propose a novel transformer-based model, BiT, for arbitrary

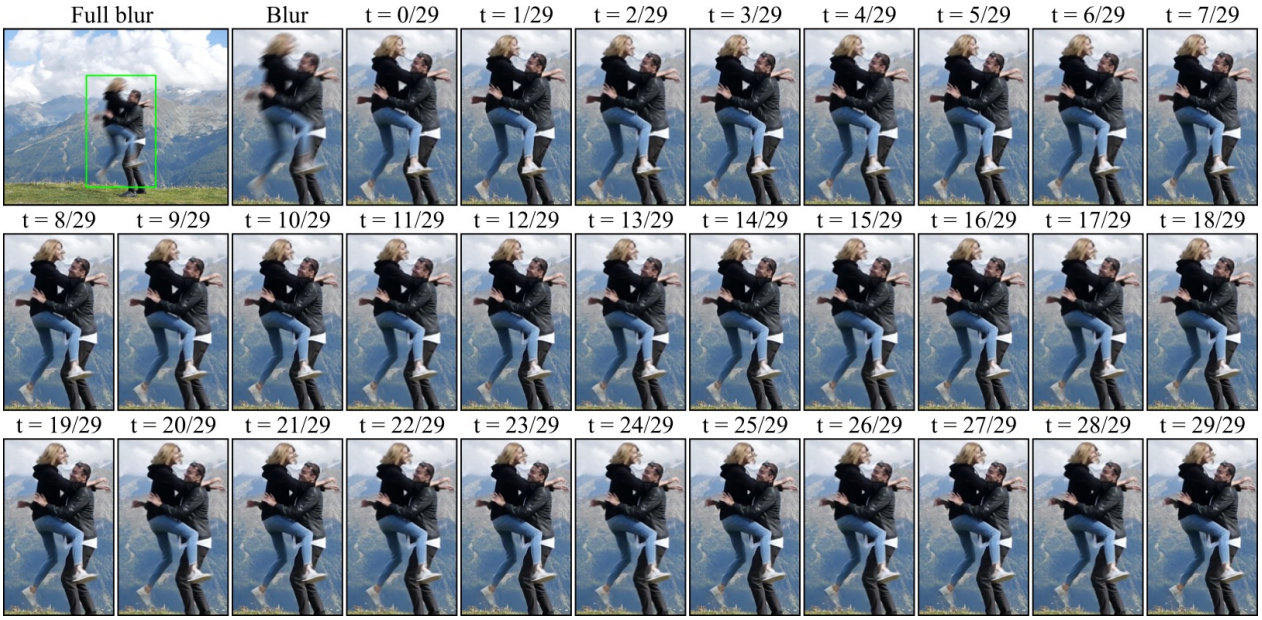


Figure 1. **Arbitrary blur interpolation by BiT.** This is an example of generating 30 sharp frames from blurred image using BiT.

time motion from blur reconstruction. BiT outperforms prior art quantitatively and qualitatively with faster speed. 2) We present and verify two successful strategies including dual-end temporal supervision and temporally symmetric ensembling to enhance the shared temporal features for arbitrary time motion reconstruction. 3) To the best of our knowledge, we provide the first real-world dataset for general blur interpolation tasks. We verify the validity of this real dataset and its meaningfulness to the community by extensive experiments.

2. Related works

2.1. General deblurring

The technological paradigm of general deblurring has experienced a shift from blur kernel estimation by traditional methods such as [10, 16, 31, 47] to direct sharp image regression by deep learning methods such as [24, 42, 43, 45]. Various general network architectures, including CNNs, RNNs, and GANs have been explored in-depth for deblurring. Nah *et al*. [24] and Tao *et al*. [43] verify the effectiveness of the multi-scale (coarse-to-fine) CNNs structure for deblurring, while Zamir *et al*. [49] prove the efficacy of multi-stage progressive strategy for deblurring. [11, 25, 46, 51, 55] customize their RNN structures to better exploit the long-term temporal correlation of blurry video. Wang *et al*. [45] adopt deformable convolution [56] to align the neighboring blurry frames to boost deblurring performance. Pan *et al*. [27] and Son *et al*. [41] explicitly utilize optical flow for more accurate motion compensation. Moreover, GANs are explored by [13, 14] to deblur images with the goal of better human perception. Recently, transformer [18] has made a splash in the low-level vision tasks. Restormer [48], RVRT [19], and VDTR [4] are proposed to demonstrate the great performance of transformer structures in the general deblurring tasks.

2.2. Blur interpolation

The aim of blur interpolation goes beyond traditional deblurring, which focuses on a one-to-one mapping between blurred and sharp images. Instead, it involves utilizing the temporal information present in motion blur to reconstruct a motion sequence. There are similar tasks that utilize the partial temporal information in rolling shutter distortion to extract video clips, such as [50] and [6]. Jin *et al*. [9] are the first to exploit blur interpolation, extracting a sharp video clip from only one single blurry image. However, blur interpolation from singe image faces the fundamental problem of directional ambiguity. Considering the freedom of each individual and uniform blurred region, the solution space of blur interpolation will be exponential. Therefore, Jin *et al*. propose a pairwise order-invariant loss to alleviate the fundamental directional ambiguity and help the model converge to a single solution. Then, Purohit *et al*. [30] utilize a motion representation, which is learned from videos by a self-supervised strategy, to further tackle the directional ambiguity. After that, Argaw *et al*. [2] leverage a spatial transformer network with multiple independent branches and a transformation consistency loss to simultaneously estimate the motion of middle time and other times within the exposure time. Zhong *et al*. [53] explicitly account for such directional ambiguity by introducing a motion guidance rep-

resentation. The motion guidelines enable their approach to produce multiple plausible solutions from the same blurred image, rather than just one as was the case before.

Taking blurry video as input [1, 8, 26, 37, 38] for blur interpolation, directional ambiguity can be largely avoided based on the motion cues of adjacent frames. Specifically, Jin *et al*. [8] present a cascaded scheme of deblurring-first and interpolation-later for this setting. To mitigate the accumulated errors introduced in the cascaded scheme, Shen *et al*. [37, 38] propose a pyramid recurrent framework to estimate the latent sharp sequence without explicitly distinguishing the deblurring stage and interpolation stage. Argaw *et al*. [1] implement blur interpolation by initially estimating the optical flow, and then predicting a motion sequence by warping the decoded features to the corresponding time points. Recently, Oh *et al*. [26] propose DeMFI framework, which combine flow-guided attentive-correlation-based feature bolstering module and recursive boosting techniques to convert lower-frame-rate blurred videos to higher-frame-rate sharp videos with state-of-the-art performance. There are also some works specialized to implement blur interpolation for fast-moving objects such as [33–35]. Given a pre-estimated background and a blurred image with a fast-moving object, they project the object representation to a latent space, and can render the object to a specified time tick within the exposure time. While Pan *et al*. [28] and Lin *et al*. [20] use an additional event camera as an aid to accomplish this task. Our approach further pushes the performance of this task on generic scenarios with dual-end temporal supervision and temporally symmetric ensembling strategies as well as a stronger backbone.

2.3. Deblurring dataset

In the early stage, the research community applied various blur kernels to synthesize blurred images with uniform motion, such as [15, 17, 36, 39]. A blurred image I_b can be described as the convolution between a sharp image I_s and a blurred kernel K with optional Gaussian noise N:

$$I_b = K * I_s + N. \tag{1}$$

Then, to deal with more realistic situation with spatially varying blur, researchers adopt a scheme of averaging consecutive frames of a high-frame-rate sharp video to synthesize blurred and sharp image/video pairs. Based on this basic pipeline, Su *et al*. [42] synthesize a dataset named DVD and additionally interpolates between sharp frames using optical flow to reduce ghosting artifacts in the synthesized blurry video. While Nah *et al*. [24] synthesize a dataset named GOPRO by applying an inverse gamma correction before averaging to reduce the effect of nonlinear transformations. Later, Nah *et al*. [23] combine the strengths of DVD and GOPRO to create a larger and more diverse synthetic deblurring dataset dubbed REDS. Regarding the more challenging blur interpolation task, previous works like [1, 8, 26, 37, 38] also use discrete frames to create datasets of one-to-many blur-sharp pairs.

Models trained on synthetic data suffer from the persistent problem of being difficult to generalize to real-world data. Thus, many researchers have started to use hybrid camera systems to collect real-world datasets for low-level vision tasks [5, 32, 51, 54]. Rim *et al*. [32] and Zhong *et al*. [51, 54] build hybrid camera systems based on beam-splitter to collect real image deblurring dataset (RealBlur) and real video deblurring datasets (BSD and BS-RSCD), respectively. Inspired by the success of real-world datasets, we customize a hybrid system to collect a real dataset (RBI) of time-aligned low-frame-rate and high-frame-rate blur-sharp video pairs. We believe RBI can benefit the community to better benchmark blur interpolation algorithms.

3. Blur interpolation transformer

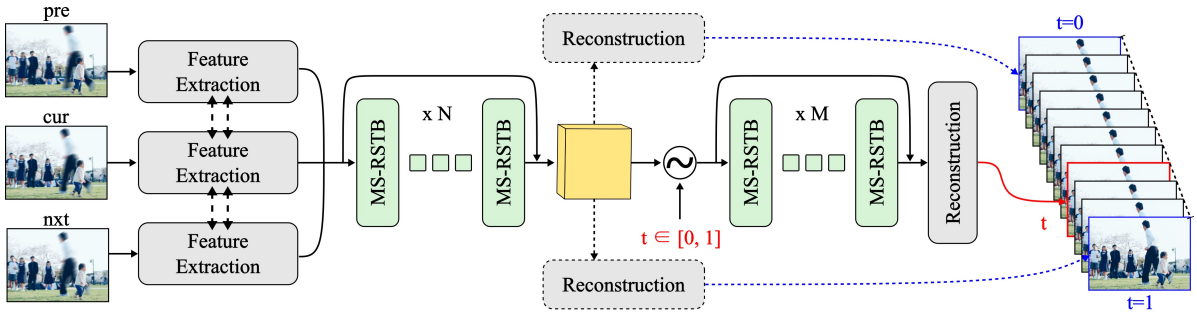
The overview architecture of our blur interpolating transformer (BiT) is shown in Fig. 2 (a). BiT focuses on interpolating a sharp motion $\hat{I}_s^t$ for a blurred image given an arbitrary t during the exposure time. Regarding the directional ambiguity in this problem, considering that modern cameras have short exposure times and that the relative speed between camera and scene is not very fast, we follow the implicit assumption of previous works [8, 26, 38, 53] that there is no ambiguity when the input is video. Thus, we also use neighboring frames as auxiliary inputs to get rid of this issue. The inference process of the target model $\mathcal{F}$ can be described as follows:

$$\hat{I}_s^t = \mathcal{F}\left(\mathbf{I_b}, t\right), \tag{2}$$

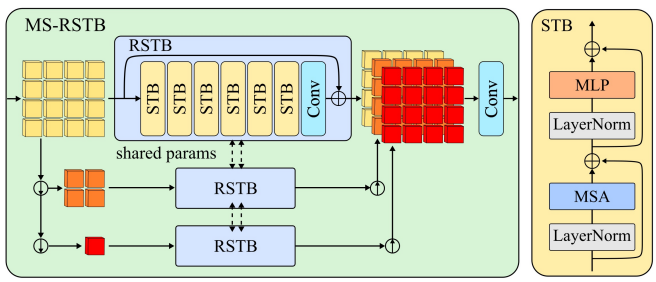
where $\mathbf{I_b} = \{I_b^{pre}, I_b^{cur}, I_b^{nxt}\}$, denoting the input set of previous, current, and next blurred images. t represents a specific time point during exposure of I_b^{cur} with a normalized value range of $t \in [0, 1]$. Apart from the lightweight reconstruction layer $\mathcal{F}_R$, the model is divided into two stages, including a shared temporal feature extraction stage $\mathcal{F}_N$ and an arbitrary motion rendering stage $\mathcal{F}_M$. $\mathcal{F}_N$ consists of a shared down-sampling convolutional block for shallow feature extraction and followed by N blocks of multi-scale residual Swin transformer blocks (MS-RSTB). The $\mathcal{F}_M$ consists of t encoding module and M MS-RSTBs. Then, the inference process Eq. 2 can be reformulated as:

$$\hat{I}_s^t = \mathcal{F}_\mathcal{R}\left(\mathcal{F}_\mathcal{M}\left(\mathcal{F}_\mathcal{N}\left(\mathbf{I_b}\right), t\right)\right). \tag{3}$$

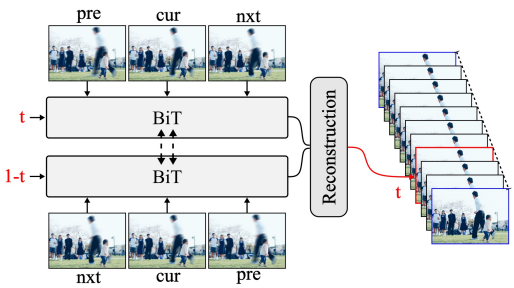
The fact that only the latter part $\mathcal{F}_M$ needs to be repeated when performing multiple time inferences can optimize the network multiplexing efficiency. Extracting well-formed shared temporal features is the key to achieving arbitrary motion rendering under discrete-time supervision. We then present the module and training strategies introduced for



(a) Overview of blur intra-interpolation transformer



(b) Multi-scale residual Swin transformer block



(c) Temporally symmetric ensembling

Figure 2. **Overview of BiT.** (a) is the proposed blur interpolation transformer that takes three consecutive blurry images as inputs to generate shared temporal features. Then, shared features and a normalized t are incorporated to render a sharp motion for a specified moment in the exposure time. Besides, there is an additional reconstruction layer with twice as many channels as the final reconstruction layer for reconstructing the results from the shared features for dual-end time points. (b) is the details of multi-scale residual Swin transformer block. A RSTB module [18] is shared to tackle blur and fuse neighboring features in a coarse-to-fine manner. (c) is temporally symmetric ensembling strategy. A pre-trained BiT with a new reconstruction layer is adopted to fuse the reconstruction features in forward (t) and reverse (1-t) orders.

improving the performance of arbitrary motion rendering from blur, one by one.

Multi-scale residual Swin transformer block. Inspired by the powerful modelling ability of residual Swin transformer block (RSTB) [18] for image restoration task, a new and efficient backbone block is constructed by introducing the classic coarse-to-fine multi-scale structure. The proposed MS-RSTB reuses one RSTB to process interpolated features at different scales, and then a convolutional layer is used to fuse the features to the same shape as the input features, as illustrated in Fig. 2 (b). The original input feature of shape $\mathbb{R}^{C\times H\times W}$ is interpolated to the shape $\mathbb{R}^{C\times H/r^{s-1}\times W/r^{s-1}}$ regarding to the scale level $s \in \{1, \cdots, S\}$ and the rescale ratio r. The computational complexity of MS-RSTB is increased as follows:

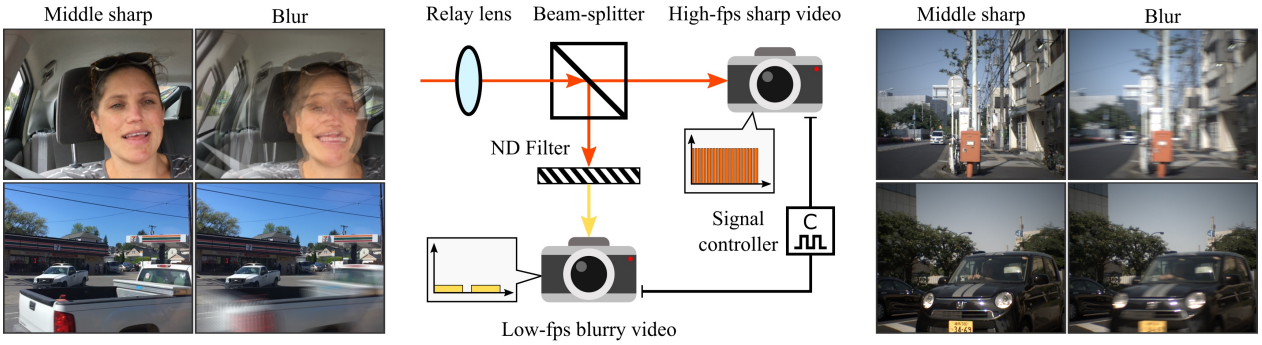
$$\Omega\left(\text{MS-RSTB}\right) \approx \frac{1-(1/r)^{2S}}{1-(1/r)^{2}}\Omega\left(\text{RSTB}\right), \tag{4}$$

where we set $S = 3$ and $r = 2$ so that there is an additional computational cost less than $1/3$. Since the window size of RSTB is fixed, the multi-scale features allow the self-attentive mechanism to be applied from global to local. This facilitates the handling of different scales of blur and the fusion of informative features from adjacent frames without prior knowledge of the range of motion.

Dual-end temporal supervision. A key observation for time-varying motion rendering from blur is that the difficulty increases from the middle moment to both sides with greater temporal difference. One insight is that if, without any temporal cues, the model is able to extract qualified features to render the most extreme moments of motion, such learned features are well-formed in respect to the varying t to better render motions at other moments. Thus, we propose a simple yet effective learning strategy, called dual-end temporal supervision (DTS), to underpin and spread the shared temporal features. Specifically, the shared temporal features, *i.e.*, the yellow cube in Fig. 2 (a), are forced to restore the motions of the two end time points using an additional lightweight reconstruction layer $\mathcal{F}_{\mathcal{R}}^{\mathcal{D}}$ without any t encoding:

$$\left\{\hat{I}_s^t \mid t = 0, 1\right\} = \mathcal{F}_{\mathcal{R}}^{\mathcal{D}}\left(\mathcal{F}_{\mathcal{N}}\left(\mathbf{I_b}\right)\right). \tag{5}$$

Note that this extra reconstruction layer will be discarded in the test mode. DTS acts as anchors for the boundaries, mak-



(a) Samples of Adobe240 (b) Hybrid-camera system (c) Samples of RBI

Figure 3. **Real blur interpolation dataset.** (a) are samples from synthetic dataset Adobe240 with unnatural spikes or steps in the blur trajectory. (b) is the schematic of our hybrid camera system. (c) are samples from our real-world dataset RBI with natural blur.

ing the shared temporal features more conducive to motion rendering in a continuous-time manner.

Temporally symmetric ensembling. Another insight into the arbitrary time motion rendering from blur arises from the consistency of the results from temporally forward and reverse inputs. Intuitively, the rendered motion at t of $\mathbf{I_b}$ can also be represented as the rendered motion at $1-t$ of the temporally inverse blurred inputs $\mathbf{I_b^{inv}} = \{I_b^{nxt}, I_b^{cur}, I_b^{pre}\}$. Thus, given a pre-trained BiT, we can further enhance it by fusing forward and inverse complementary features, named temporally symmetric ensembling (TSE). As shown in Fig. 2 (c), during the fine-tuning process, the last reconstruction layer is replaced by a new layer $\mathcal{F}_{\mathcal{R}}^{\mathcal{T}}$ that can accept twice the number of input channels. The inference process with TSE strategy is as follows:

$$\hat{I}_s^t = \mathcal{F}_{\mathcal{R}}^{\mathcal{T}}\left(\mathcal{F}_{\mathcal{M}}\left(\mathcal{F}_{\mathcal{N}}\left(\mathbf{I_b}\right), t\right), \mathcal{F}_{\mathcal{M}}\left(\mathcal{F}_{\mathcal{N}}\left(\mathbf{I_b^{inv}}\right), 1-t\right)\right). \tag{6}$$

Thanks to the proposed MS-RSTB and the designed temporal optimization strategies, taking the L1 loss to supervise reconstruction results is sufficient to ensure an excellent performance of the model. The total loss terms are as follows:

$$\mathcal{L} = \mathcal{L}_1\left(\hat{I}_s^t, I_s^t\right) + \lambda\left(\mathcal{L}_1\left(\hat{I}_s^0, I_s^0\right) + \mathcal{L}_1\left(\hat{I}_s^1, I_s^1\right)\right). \tag{7}$$

In the training phase, t is only randomly sampled from the available discrete time points of the corresponding dataset. However, in the test phase, t can take any continuous value between 0 and 1. As for the t encoding, we expand the spatial size of t to the same size as the shared features. Then, we merge it into the channel dimension of the shared features, followed by a linear layer for feature fusion. Empirically, we find that such simple encoding can provide good performance, slightly better than the widely used frequency encoding such as [22].

4. Real-world blur interpolation dataset

Limitation of synthetic dataset. First, let us briefly review the synthetic pipeline of previous works. Taking Adobe240 from [37, 38] as an example, a sliding window with size of M frames is used to average the sharp images. The blurred image can be generated as follows:

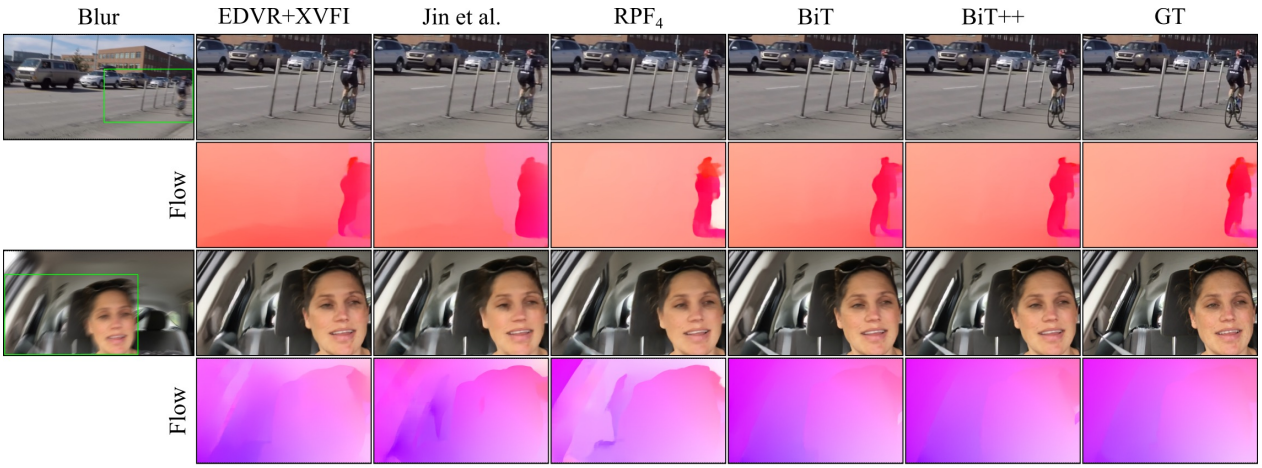
$$I_b = \frac{1}{M}\sum_{m=1}^{M}\left(I_s^m\right), \tag{8}$$

where $M = 11$. The samples of Adobe240 are illustrated in Fig. 3 (a). We can observe the unnatural spikes or steps in the blur trajectory due to the discrete averaging process. Therefore, synthetic datasets like Adobe240 may not reflect the actual difficulty of the blur interpolation task. In addition, the gap between synthetic blur and real blur may lead to generalization problems for models trained on synthetic datasets.

Hybrid camera system. Building a real-world dataset for the blur interpolation task becomes an urgent need. Blur should occur naturally in the form of signal accumulation, as follows:

$$I_b = \int_0^{\tau} S(t)dt, \tag{9}$$

where τ denotes the exposure time, and $S(t)$ denotes the signal captured by the camera sensor at time t. To this end, we design a hybrid camera system as illustrated in Fig. 3 (b). Specifically, two BITRAN CS-700C cameras are physically aligned to the beam-splitter by laser calibration. During shooting, the light is split in half and goes into the camera with high and low frame rate modes. The low-frame-rate camera adopts long exposure scheme to capture the blurred video. Besides, a ND filter with about 10% transmittance is installed before the low-frame-rate camera to ensure photometric balance between the blurred frames and the corresponding sharp frames from the high-frame-rate camera.



(a) Comparison on Adobe240 dataset

(b) Comparison on RBI dataset

Figure 4. **Qualitative comparisons on Adobe240 dataset and real-world dataset RBI.**

(a) Test samples from RBI

(b) Test samples from Adobe240

Figure 5. **Cross-validation between synthetic dataset Adobe240 [38] and real-world dataset RBI.** BiT++(Adobe240) and BiT++(RBI) represent the model trained on Adobe240 and RBI. The model trained on synthetic data will cause artifacts when tested on real data.

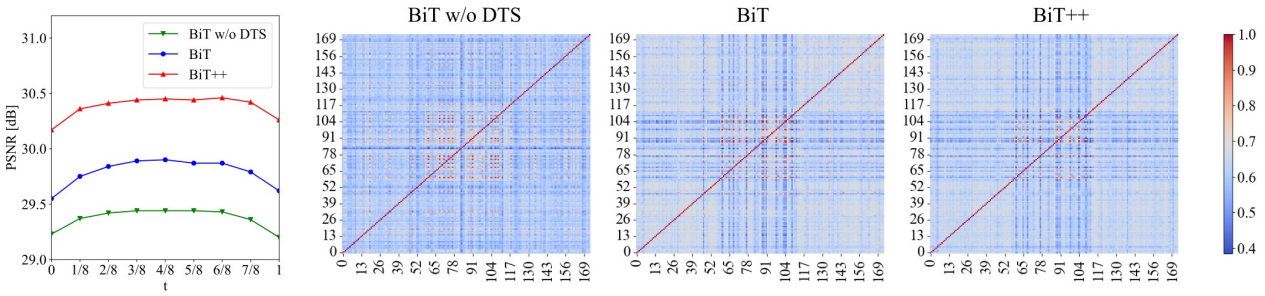
RBI dataset. We use this customized hybrid camera system to collect 55 video pairs as real-world blur interpolation (RBI) dataset. The frame-rate of blurred video and the corresponding sharp video are 25 fps and 500 fps, respectively. The exposure time of blurred image is 18 ms, while the exposure time of sharp image is nearly 2 ms. This means that there are 9 sharp frames corresponding to one blurred frame, and 11 sharp frames exist in the readout deadtime between adjacent blurred frames. The image size is 640×480. We shoot videos of normal urban scenes with various motion modes, including ego-motion, object motion, and both. In addition to blur interpolation, RBI can serve as a dataset to measure the performance of blur synthesis algorithms, such as [3].

5. Experiments

We optimize the loss using AdamW [21] in the PyTorch framework [29]. λ is empirically set as 0.5. In both the initial training or fine-tuning phase, the learning rate is scheduled by the cosine scheduler from 1×10^{-4} to 1×10^{-6}. Common data augmentation operations, including flipping, rotation, and cropping of size 256×256, are used. We set $M = 3$ and $N = 3$ as the default BiT settings for $\mathcal{F}_{\mathcal{N}}$ and

Table 1. **Comparison with the state-of-the-arts on synthetic dataset Adobe240 and our real-world dataset RBI.** **Red** denotes the best performance, and blue denotes the second best performance. Runtime is calculated uniformly using images from the Adobe240 dataset with size of 640×352 on a single RTX2080 Ti GPU.

	Adobe240		RBI		Runtime		
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	1x [s] ↓	60x [s] ↓	Params [M] ↓
EDVR [45]+XVFI [40]	33.19	0.934	28.17	0.847	0.294	17.64	29.2
Jin *et al*. [8]	32.47	0.924	27.73	0.853	0.250	15.00	10.8
RPF_4 [38]	33.32	0.935	28.55	0.872	0.746	44.76	11.4
DeMFI [26]	34.34	0.945	29.03	0.895	0.513	30.78	**7.41**
BiT	34.34	0.948	29.90	0.900	**0.203**	**5.76**	11.3
BiT++	**34.97**	**0.954**	**30.45**	**0.908**	0.395	11.64	11.3



(a) Time-varying performance (b) Channel similarity analysis of shared temporal features

Figure 6. **Ablation studies for temporal strategies.** (a) shows time-varying performance. (b) is CKA analysis of shared temporal features. This visualization is created based on the test data of RBI.

$\mathcal{F}_{\mathcal{M}}$. The number of heads and channel size of self-attention is set to 6 and 174. Regarding the partitioning of the dataset, Adobe240 is the same as previous work [37, 38]; while 50 videos of RBI are used for training and the remaining 5 videos are used for testing. We train BiT on Adobe240 with a batch size of 32 for 800 epochs, and finetune it with TSE strategy for another 400 epochs, on 8 NVIDIA Tesla V100 GPUs. Since the size of RBI is smaller than Adobe240, we double the number of epochs and reduce the batch to 8 to get more iterations for training. Regarding the other models, we retrain them on each dataset for a fair comparison. In addition to this section, we encourage readers to refer to the appendices for more details regarding the proposed RBI dataset, as well as supplementary ablation studies and experiments.

Quantitative and qualitative results. We compare our method with previous state-of-the-arts including Jin *et al*. [8], RPF_4 [38], DeMFI [26], and a cascaded method consists of deblurring model EDVR [45] and interpolation model XVFI [40]. Since the Adobe240 dataset has no readout deadtime, we follow the 2x temporal super-resolution setting of this dataset to compare with other methods. While on the RBI dataset, we compare the middle deblurred images, because in the real case, there is a readout deadtime.

Quantitative results are shown in Table 1. We name our model with the TSE strategy as BiT++ and the one without is BiT. BiT++ can outperform the prior art on Adobe240 and on RBI by a large margin. Besides, the more time points are derived from the same input, the faster our model becomes. BiT achieves 60 inferences in 5.76 seconds, while maintaining favorable performance. Qualitative results are shown in Fig. 4. We can see that the predictions of BiT and BiT++ are closer to the groundtruth with clearer details on both Adobe240 and RBI. We further utilize RAFT [44] to estimate the optical flow between two adjacent predicted frames on Adobe240, as illustrated in Fig. 4 (a). The optical flow of our results is also closer to the groundtruth, which indicates better motion consistency.

Dataset cross-evaluation. To demonstrate the need for a real dataset, we conduct experiments on cross-evaluation between the synthetic dataset Adobe240 and the real-world dataset RBI. First, we show the results of RBI samples predicted by independent BiT++ models trained on Adobe240 and RBI in Fig. 5 (a). We can observe severe artifacts in the results of the model trained on Adobe240. Conversely, testing on synthetic data shown in Fig. 5 (b), we find that

Table 2. **Ablation studies.** BiT w/o MS denotes BiT using single-scale RSTB module. BiT w/o DTS denotes BiT without dual-end temporal supervision. BiT+ denotes BiT that has the same training epochs as BiT++.

	Adobe240					RBI				
	BiT w/o MS	BiT w/o DTS	BiT	BiT+	BiT++	BiT w/o MS	BiT w/o DTS	BiT	BiT+	BiT++
PSNR ↑	33.96	34.10	34.34	34.52	**34.97**	29.40	29.44	29.90	29.99	**30.45**
SSIM ↑	0.944	0.946	0.948	0.946	**0.954**	0.893	0.894	0.900	0.901	**0.908**

Table 3. **Effect of # of MS-RSTB.** The performance is evaluated on Adobe240 using BiT.

	$N, M = 0, 6$	$N, M = 1, 5$	$N, M = 2, 4$	$N, M = 3, 3$	$N, M = 4, 2$	$N, M = 5, 1$	$N, M = 6, 0$
PSNR ↑	34.08	34.09	34.18	**34.34**	34.30	34.05	27.13
SSIM ↑	0.947	0.942	0.943	**0.948**	**0.948**	0.944	0.832
60x Runtime [s] ↓	11.34	9.36	7.98	5.76	4.02	2.16	**0.36**

the model trained on RBI does not introduce artifacts, even if it could not remove the synthetic blur. This experiment demonstrates the risks of training a model on a synthetic dataset for the blur interpolation task, which is consistent with the findings of previous work [52].

Ablation studies. In order to verify the validity of the proposed new module and strategies, we perform relevant ablation experiments. We show the results of BiT with only single-scale RSTB (denoted as BiT w/o MS), BiT without DTS (denoted as BiT w/o DTS), and BiT with the same total training epochs as BiT++ (denoted as BiT+) in Table. 2. The MS-RSTB, DTS, and TSE can bring 0.38dB, 0.24dB, and 0.45dB gain on Adobe240, as well as 0.50dB, 0.46dB, and 0.46dB gain on RBI, respectively. We also present the curves of time-varying performance of ablated models on RBI, as illustrated in Fig.6 (a). The full model BiT++ improves the performance of all time points within the exposure time.

To further explain the benefits from our temporal feature enhancement strategies, we use the central kernel alignment (CKA) [12] to measure the channel-wise similarity of the extracted shared temporal features, as shown in Fig. 6 (b). The modified CKA calculation process is as follows:

$$\mathrm{CKA}(i,j) = \frac{\mathrm{HSIC}(G(F_i), G(F_j))}{\sqrt{\mathrm{HSIC}(G(F_i), G(F_i))\mathrm{HSIC}(G(F_j), G(F_j))}}, \tag{10}$$

where i and j are the channel indices of extracted shared temporal feature $F \in \mathbb{R}^{C\times H\times D}$. HSIC and G are the functions to calculate the Hilbert-Schmidt independence criterion and Gram matrices, respectively. We find that after applying the temporal feature enhancement strategy including DTS and TSE, the shared features show a significant functional stratification in the channel dimension. In particular, the channel features of BiT++ are clustered into three rectangular blocks. We speculate that the first and third feature blocks aggregate common features shared by different time inferences, while the middle feature block aggregates more differentiated features ready to be retrieved accordingly based on the given time query.

Finally, we investigate the optimal distribution of MS-RSTB in Table 3 with a constant total number of 6. By sharing MS-RSTBs before shared temporal features at different inference points, temporal feature computation for multiple time points requires only a single computation of the N MS-RSTB blocks, enabling faster inference with larger values of N. Our analysis indicates that increasing N to 4 or 5 offers a substantial speedup with a negligible decline in performance compared to the default $N = 3$ setting.

6. Conclusion, limitation, and future work

We propose a novel and efficient model, BiT, to realize arbitrary time blur interpolation with state-of-the-art performance. In addition, we present a real-world dataset RBI that enables the first real-world benchmark for blur interpolation task. However, current limited discrete supervision may not be sufficient to cope with very fast motions. Besides, to better cope with different real-world situations, our dataset needs to be expanded to include video pairs with different devices and different exposure parameters. We believe that the reversed process, *i.e.*, learning to synthesize real blur using successive sharp frames from RBI, is also an interesting and valuable direction for the future.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers 20H05953, 22H00529, 20H05951, and JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2108.

References

[1] Dawit Mureja Argaw, Junsik Kim, Francois Rameau, and In So Kweon. Motion-blurred video interpolation and extrapolation. In *AAAI Conference on Artificial Intelligence*, 2021. 3

[2] Dawit Mureja Argaw, Junsik Kim, Francois Rameau, Chaoning Zhang, and In So Kweon. Restoration of video frames from a single blurred image with motion understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710, 2021. 2

[3] Tim Brooks and Jonathan T Barron. Learning to synthesize motion blur. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6840–6848, 2019. 6

[4] Mingden Cao, Yanbo Fan, Yong Zhang, Jue Wang, and Yujiu Yang. Vdtr: Video deblurring with transformer. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 2

[5] Mingdeng Cao, Zhihang Zhong, Jiahao Wang, Yinqiang Zheng, and Yujiu Yang. Learning adaptive warping for real-world rolling shutter correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17785–17793, 2022. 3

[6] Bin Fan and Yuchao Dai. Inverting a rolling shutter camera: bring rolling shutter images to high framerate global shutter video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4228–4237, 2021. 2

[7] Akash Gupta, Abhishek Aich, and Amit K Roy-Chowdhury. Alanet: Adaptive latent attention network forjoint video deblurring and interpolation. *arXiv preprint arXiv:2009.01005*, 2020. 1

[8] Meiguang Jin, Zhe Hu, and Paolo Favaro. Learning to extract flawless slow motion from blurry videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8112–8121, 2019. 1, 3, 7

[9] Meiguang Jin, Givi Meishvili, and Paolo Favaro. Learning to extract a video sequence from a single motion-blurred image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6334–6342, 2018. 1, 2

[10] Tae Hyun Kim and Kyoung Mu Lee. Generalized video deblurring for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5426–5434, 2015. 2

[11] Tae Hyun Kim, Kyoung Mu Lee, Bernhard Scholkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4038–4047, 2017. 2

[12] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 8

[13] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 2

[14] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019. 2

[15] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *CVPR*, pages 1701–1709, 2016. 3

[16] Anat Levin. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems*, pages 841–848, 2007. 2

[17] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971. IEEE, 2009. 3

[18] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 2, 4

[19] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool. Recurrent video restoration transformer with guided deformable attention. *arXiv preprint arXiv:2206.02146*, 2022. 2

[20] Songnan Lin, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy Ren. Learning event-driven video deblurring and interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 695–710. Springer, 2020. 3

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 5

[23] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Mu Kyoung Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[24] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 2, 3

[25] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8102–8111, 2019. 2

[26] Jihyong Oh and Munchurl Kim. Demfi: Deep joint deblurring and multi-frame interpolation with flow-guided attentive correlation and recursive boosting. *arXiv preprint arXiv:2111.09985*, 2021. 1, 3, 7

[27] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3043–3051, 2020. 2

[28] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6820–6829, 2019. 3

[29] 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. *arXiv preprint arXiv:1912.01703*, 2019. 6

[30] Kuldeep Purohit, Anshul Shah, and AN Rajagopalan. Bringing alive blurred moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2019. 2

[31] Wenqi Ren, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1077–1085, 2017. 2

[32] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European Conference on Computer Vision*, pages 184–201. Springer, 2020. 1, 3

[33] Denys Rozumnyi, Martin R Oswald, Vittorio Ferrari, Jiri Matas, and Marc Pollefeys. Defmo: Deblurring and shape recovery of fast moving objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3456–3465, 2021. 1, 3

[34] Denys Rozumnyi, Martin R Oswald, Vittorio Ferrari, and Marc Pollefeys. Shape from blur: Recovering textured 3d shape and motion of fast moving objects. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3

[35] Denys Rozumnyi, Martin R Oswald, Vittorio Ferrari, and Marc Pollefeys. Motion-from-blur: 3d shape and motion estimation of motion-blurred objects in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15990–15999, 2022. 1, 3

[36] Uwe Schmidt, Carsten Rother, Sebastian Nowozin, Jeremy Jancsary, and Stefan Roth. Discriminative non-blind deblurring. In *CVPR*, pages 604–611, 2013. 3

[37] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Blurry video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5114–5123, 2020. 1, 3, 5, 7

[38] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Video frame interpolation and enhancement via pyramid recurrent framework. *IEEE Transactions on Image Processing*, 30:277–292, 2020. 1, 3, 5, 6, 7

[39] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, and Ming-Hsuan Yang. Deep semantic face deblurring. In *CVPR*, pages 8260–8269, 2018. 3

[40] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: Extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14489–14498, 2021. 7

[41] Hyeongseok Son, Junyong Lee, Jonghyeop Lee, Sunghyun Cho, and Seungyong Lee. Recurrent video deblurring with blur-invariant motion estimation and pixel volumes. *ACM Transactions on Graphics (TOG)*, 40(5):1–18, 2021. 2

[42] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 2, 3

[43] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018. 2

[44] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 7

[45] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2, 7

[46] Yusheng Wang, Yunfan Lu, Ye Gao, Lin Wang, Zhihang Zhong, Yinqiang Zheng, and Atsushi Yamashita. Efficient video deblurring guided by motion magnitude. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*, pages 413–429. Springer, 2022. 2

[47] Jonas Wulff and Michael Julian Black. Modeling blurred video with layers. In *European Conference on Computer Vision*, pages 236–252. Springer, 2014. 2

[48] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2022. 2

[49] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14821–14831, 2021. 2

[50] Zhihang Zhong, Mingdeng Cao, Xiao Sun, Zhirong Wu, Zhongyi Zhou, Yinqiang Zheng, Stephen Lin, and Imari Sato. Bringing rolling shutter images alive with dual reversed distortion. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 233–249. Springer, 2022. 2

[51] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 1, 2, 3

[52] Zhihang Zhong, Ye Gao, Yinqiang Zheng, Bo Zheng, and Imari Sato. Real-world video deblurring: A benchmark dataset and an efficient recurrent neural network. *International Journal of Computer Vision*, pages 1–18, 2022. 8

[53] Zhihang Zhong, Xiao Sun, Zhirong Wu, Yinqiang Zheng, Stephen Lin, and Imari Sato. Animation from blur: Multi-modal blur decomposition with motion guidance. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*, pages 599–615. Springer, 2022. 2, 3

[54] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *CVPR*, pages 9219–9228, 2021. 3

[55] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2482–2491, 2019. 2

[56] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 2