# Blur Interpolation Transformer for Real-World Motion from Blur – Supplementary Material

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

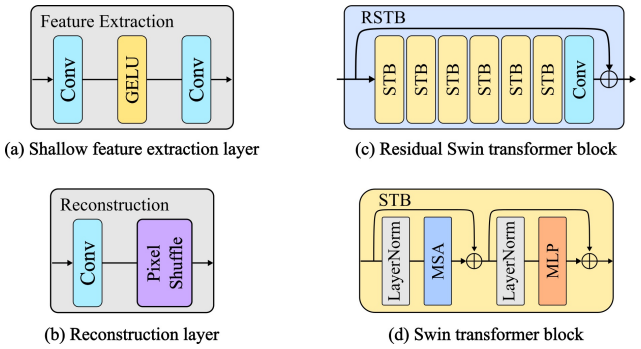



Figure 1. **Structure of sub-modules.** (a) is the structure of shared shallow feature extraction layer. (b) is the structure of reconstruction layer. (c) is the structure of residual Swin transformer block. (d) is the structure of Swin transformer block.

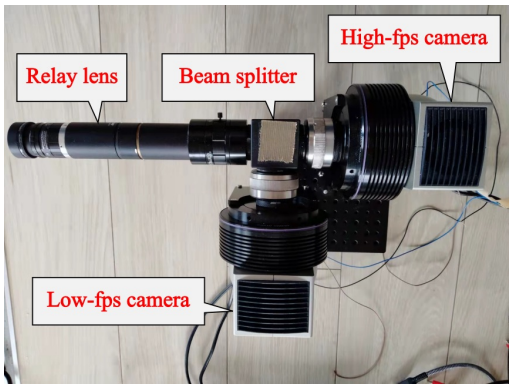



Figure 2. **Hardware of the proposed hybrid camera system.**

## 1. Supplementary details

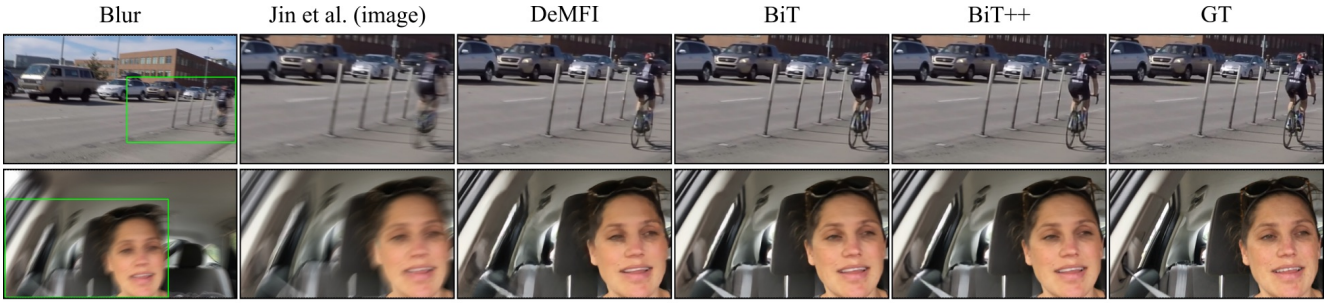
**Blur interpolation transformer (BiT).** First, we add some details of the BiT network structure in this part. The shared shallow feature extraction layer is illustrated in Fig. 1 (a). It consists of two $3 \times 3$ 2d convolution layers with stride equal to 2, and a GELU [2] activation layer between them. The lightweiht reconstruction layer is shown in Fig. 1 (b). It only consists of a $3 \times 3$ 2d convolution layer and a PixelShuffle [9] layer with upscale factor equal to 4. Regarding residual Swin transformer block (RSTB), we borrow the structure from SwinIR [4], as illustrated in Fig. 1 (c). It consists of 6 stacked Swin transformer blocks (STB) and a $3 \times 3$ 2d convolution layer at the end for learning the features in a residual manner. The STB follows the design of [5], as illustrated in Fig. 1 (d). It applies the standard multi-head self-attention mechanism [10] to locally shifted windows. First, the input features are reshaped from $\mathbb{R}^{H \times W \times C}$ to $\mathbb{R}^{\frac{HW}{M^2} \times M^2 \times C}$ by dividing the features into $\frac{HW}{M^2}$ non-overlapping local windows of shape $M \times M$. In our case, we set $M = 8$. Then, self-attention mechanism is applied to the features $F \in \mathbb{R}^{M^2 \times C}$ in each local window. The *query*, *key*, and *value* matrices $Q$, $K$, and $V$ are calculated as follows:

$$Q = FP_Q, \quad K = FP_K, \quad V = FP_V, \tag{1}$$

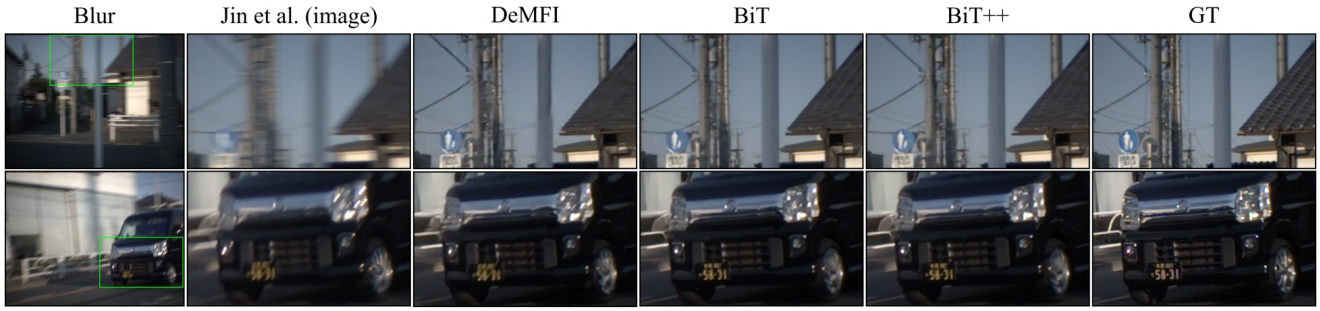
where $P_Q$, $P_K$, and $P_V$ are shared projection matrices across local windows, and $Q$, $K$, and $V$ are projected features with shape $\mathbb{R}^{M^2 \times d}$. The process of self-attention is

Table 1. **Configurations of RBI dataset.**

| | Train | Test |
|---|---|---|
| Video pairs | 50 | 5 |
| Blur frame-rate | 25 | 25 |
| Sharp frame-rate | 500 | 500 |
| Blur exposure time | 18ms | |
| Sharp exposure time | ≈2ms | |
| Total blurred frames | 1250 | 125 |
| Total sharp frames | 25000 | 2500 |
| Resolution | $640 \times 480$ | |
| Camera | BITRAN CS-700C | |




(a) Supplementary comparison on Adobe240 dataset




(b) Supplementary comparison on RBI dataset

Figure 3. **Supplementary comparison on Adobe240 [8] and the real-world RBI dataset.**

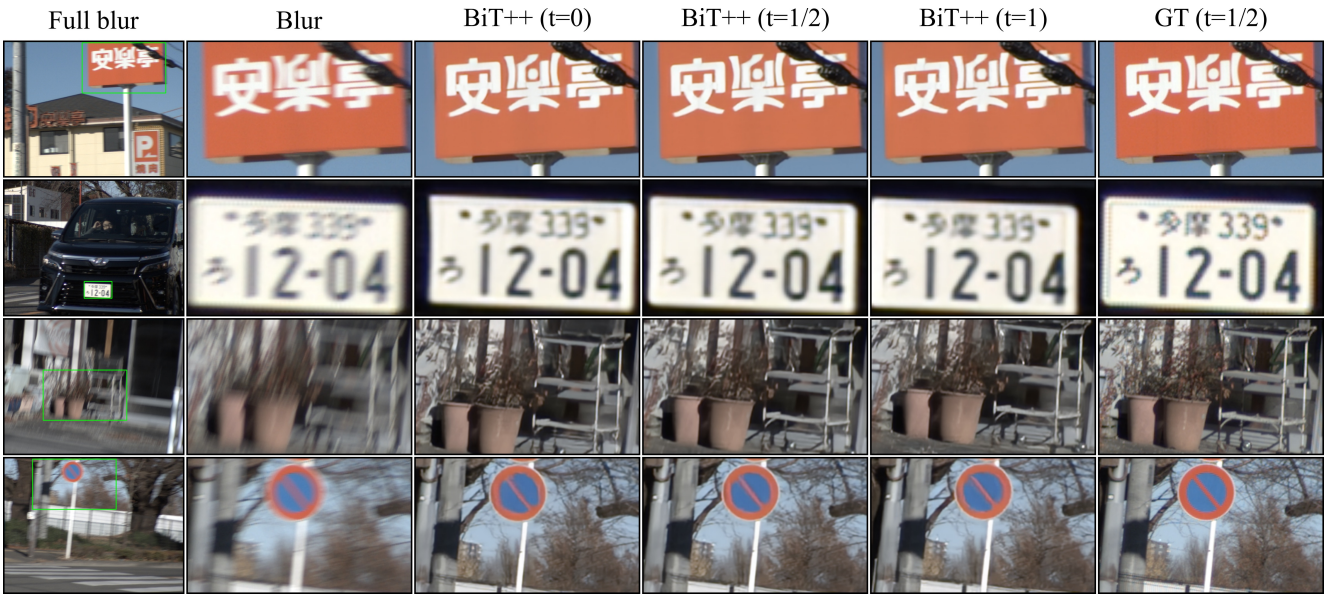



Figure 4. **Cross-validation on BSD dataset.**

Table 2. **Effect of selected indices for DTS strategy.** The modifications are based on BiT.

| | t=1/2 & t=1/2 | t=3/8 & t=5/8 | t=1/4 & t=3/4 | t=1/8 & t=7/8 | t=0 & t=1 |
|---|---|---|---|---|---|
| PSNR ↑ | 29.70 | 29.79 | 29.86 | 29.91 | 29.90 |
| SSIM ↑ | 0.894 | 0.895 | 0.897 | 0.898 | 0.900 |

Table 3. **Additional ablation for DTS.**

| | PSNR ↑ | SSIM ↑ |
|---|---|---|
| BiT (DTS w/ t) | 29.24 | 0.891 |
| BiT | 29.90 | 0.900 |

Table 4. **Additional ablation for TSE.**

| | PSNR ↑ | SSIM ↑ | Parameters [M] ↓ |
|---|---|---|---|
| BiT+ (larger) | 30.12 | 0.902 | 12.049 |
| BiT++ | 30.45 | 0.908 | 11.345 |

Table 5. **Ablation study of t encoding scheme.**

| | BiT (freq.) | BiT | BiT (freq.) | BiT |
|---|---|---|---|---|
| Dataset | Adobe240 | Adobe240 | RBI | RBI |
| PSNR ↑ | 34.27 | 34.34 | 29.85 | 29.90 |
| SSIM ↑ | 0.948 | 0.948 | 0.897 | 0.900 |

Table 6. **Effect of pretraining from Adobe240 to RBI.** The metrics are calculated only using middle predicted results ($t = 0.5$).

| | BiT | Pre-BiT | BiT++ | Pre-BiT++ |
|---|---|---|---|---|
| PSNR ↑ | 29.90 | 30.79 | 30.45 | 31.32 |
| SSIM ↑ | 0.900 | 0.916 | 0.908 | 0.922 |

Table 7. **Comparison with Jin *et al*. [3] that takes single blurred image as input on synthetic dataset Adobe240 and our real-world dataset RBI.** **Red** denotes the best performance, and blue denotes the second best performance.

| | Adobe240 | | RBI | |
|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ |
| Jin *et al*. [3] | 25.03 | 0.776 | 25.27 | 0.814 |
| BiT | 34.34 | 0.948 | 29.90 | 0.900 |
| BiT++ | **34.97** | **0.954** | **30.45** | **0.908** |

described as follows:

$$\text{Attention}\,(Q,K,V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V, \quad (2)$$

where $B$ denotes the learnable relative positional encoding. Besides, multi-head mechanism is adopted to the self-attention (MSA), *i.e*., performing self-attention in parallel on the channel dimensions. We set the number of heads to 6 and the total number of channels to 174. Along with a multi-layer perception (MLP) and LayerNorm operation, the whole process of STB is as follows:

$$F = MSA(LN(F)) + F, \quad (3)$$
$$F = MLP(LN(F)) + F. \quad (4)$$

Besides, shifted window partition is alternated between blocks to achieve cross-window connections, where the shift size is half of the window size $M$.

**Real-world blur interpolation dataset (RBI).** The actual diagram of our hybrid camera is illustrated in the Fig. 2. In addition, the detailed configurations are shown in Table 1. As for geometric alignment, the two cameras are first mechanically aligned assisted with collimated laser beams. Later, a homography correction using standard checker pattern is conducted, so as to reduce the alignment error to less than one pixel. Lens distortion will occur when two lenses are behind the beam-splitter, thus we put the lens in front (only one lens). Even with any distortion, the two cameras are identical, so the effect on learning is limited. There is no post-processing, such as flow-based alignment, but only homography. This real-world dataset can be applied to multiple applications, such as image/video deblurring, blur interpolation, and blur synthesis [1]. By simply modifying the parameters of the hardware, we can obtain a richer and more diverse dataset in the future.

## 2. Supplementary results

**Effect of selected indices for DTS strategy.** We present an ablation study of the indices selected for the DTS strategy in Table 2. t=0 & t=1 represents the default setting for DTS strategy. These ablation studies are conducted on RBI dataset. Experimental results show that it is better to select sharp frames closer to the dual-end of the exposure time as supervision.

**Additional ablation for DTS.** We add an ablation study of supervising dual-end by going through $F_M$ with t as input, namely "BiT (DTS w/ t)", as shown in Table 3. The results support that DTS can make the form of shared features more conducive to arbitrary interpolation.

**Additional ablation for TSE.** In contrast to BiT/BiT+, BiT++ has a larger reconstruction layer with 11.345M parameters, which is a mere 0.67% increase over the 11.270M parameters in BiT/BiT+. We train a larger BiT+ network with 12.049M parameters by increasing the output channels of $F_M$ without TSE, as shown in Table 4. The results support that TSE actually brings extra information more than the effect of more parameters.

**Ablation study of t encoding scheme.** We show the comparison to commonly used frequency encoding in Table 5. As mentioned in the manuscript, we find that simple encoding by concatenating feature channels can provide good enough performance, even slightly better than the widely used frequency encoding.

**Pretraining using Adobe240.** We show the effect of pretraining from Adobe240 [7, 8] to our RBI in Table 6. The BiT initialized with checkpoints of the BiT trained on Adobe240 is denoted as Pre-BiT, and the corresponding full model with temporal symmetry is denoted as Pre-BiT++. The results show that pretraining with Adobe240 can bring benefits to the model. Although the data of Adobe240 is synthetic, its scene diversity is useful to the model.

**Supplementary comparison results.** Due to the directional blurring problem [12], it is very unlikely that the method with one blurred image as input produces results with the same decomposition order as the ground-truth sequence. The poor quantitative performance of Jin *et al*. [3] in Table 7 prove this point. Besides, we supplement the visual results of Jin *et al*. [3] and DeMFI [6] on both Adobe240= [8] and the real-world RBI dataset to further validate the superior performance of our method, as illustrated in Fig. 3.

**Third-part data validation.** To further validate the robustness of the model trained on our real-world dataset, RBI, we test our model on the third-party data from BSD [11]. The qualitative results are shown in Fig. 4. Sharp motion sequence are successfully rendered out of the blurred images. This again highlights the necessity and importance of proposing a real dataset.

**Video demos.** In the end, we provide several video clips compared with previous methods including $\text{RPF}_4$ [7, 8] and DeMFI [6] in https://zzh-tech.github.io/BiT/ for reference. BiT++ upsamples the temporal resolution by a factor of 16, resulting in smooth video clips with clearer details. Further improvement of the long-term temporal consistency (across several blurred input images) will be one of our future directions.

## References


[1] 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. 3

[2] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 1

[3] 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. 3, 4

[4] 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. 1

[5] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1

[6] 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. 4

[7] 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. 4

[8] 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. 2, 4

[9] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 1

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[11] 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. 4

[12] Zhihang Zhong, Xiao Sun, Zhirong Wu, Yinqiang Zheng, Stephen Lin, and Imari Sato. Animation from blur: Multimodal 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. 4