

Appendix of “Real-world Deep Local Motion Deblurring”

A. Color Cast Problems in RealBlur Dataset

RealBlur dataset (Rim et al. 2020) contains local degradations of color casts in paired images. The colors of the same area in a pair of sharp-blur images are different. This should be avoided because color casts may reduce the training performance in local deblurring tasks. Fig. 1 provides examples of the color cast problem in RealBlur-J. In Scene 059, both the reference image pair (Blur-1 and GT-1) and the blur-sharp pair (Blur-14 and GT-14) contain color cast locally. Blur-1’s window frame area contains more colorful stripes than GT-1’s window frame area does. The Blur-14’s window area contains more orange elements than the GT-14’s window area does. The same problems occur in Scene 108. After deblurring by DeblurGAN on RealBlur dataset, color casts are more pronounced in the deblurring results. Different from RealBlur dataset, we apply color correction and photometrical alignment in post-precessing in Section 3 (in the main body of our paper). Fig. 2 and Fig. 3 in Appendix D prove that our effort is well-assured.

B. The Authenticity of ReLoBlur Dataset

We prove the authenticity of ReLoBlur dataset by quantitatively and perceptually evaluating the results of our paired image post-processing pipeline, including color correction, photometrical alignments and geometrical alignments.

To evaluate the color correction performances, we capture a standard whiteboard in a standard 6500K reflective lightbox. As shown in Fig. 2, before color correction, there exists non-uniform color cast in each camera and the grayscale values disperse from 200 to 255. After color correction, the photo of the whiteboard presents uniform white color and the grayscale values are more concentrated (230~250).

Results of photometrical alignment are depicted in Fig. 3, which shows that there exists an apparent brightness difference between the original-shot photo captured by Camera B and Camera S. After photometrical alignment, the two photos share almost the same brightness and color.

We quantitatively evaluate the accuracy of color correction and photometric alignment by computing a ΔL function by pixels between the static sharp background (l_1) and blur background (l_2), which is defined as:

$$\Delta L = \frac{\sum_{x=1}^M \sum_{y=1}^N |l_1(x, y) - l_2(x, y)|}{\sum_{x=1}^M \sum_{y=1}^N l_2(x, y)}, \quad (1)$$

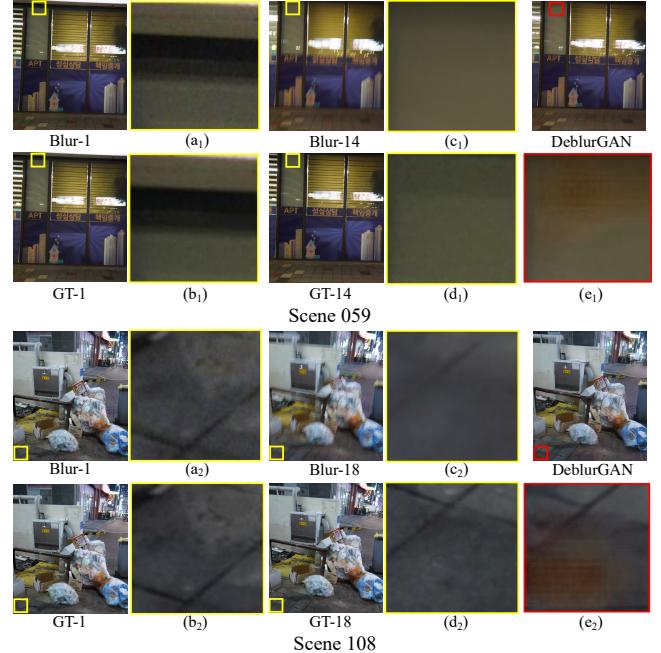
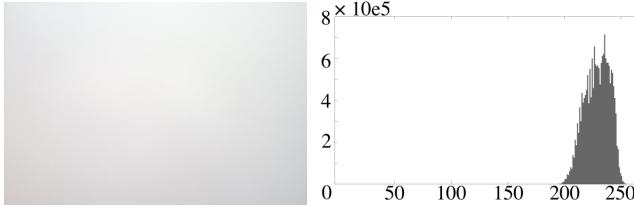


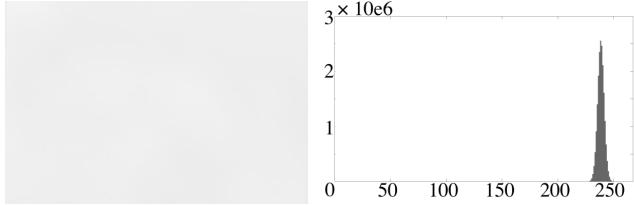
Figure 1: Color cast problems occurred in RealBlur dataset and how they influence deblurring neural network’s training results: (a₁), (a₂), (b₁) and (b₂) are patches of reference image pairs. (c₁), (c₂), (d₁) and (d₂) are patches of blur-sharp image pairs. (e₁) and (e₂) are deblurring results of DeblurGAN on RealBlur-J. The two examples are from Scene 059 and Scene 108 in RealBlur-J.

where $M = 2152$ and $N = 1436$ are the image width and height, respectively. (x, y) denotes pixel location. The average ΔL of the static image pairs of all scenes is 2.870%, which implies that there is no obvious color cast or brightness difference perceptually.

For geometrical alignment evaluation, we compare the PSNR of the static background image pairs before and after geometrical alignment. Fig. 4 shows that there is no obvious misalignment between the sharp and locally blurred image perceptually. As illustrated in Tab. 1, the average PSNR of the static aligned image pairs is 36.79dB, which is 10dB over that of the original-shot image pairs. Both the visual results



(a) before color correction.



(b) after color correction.

Figure 2: Color correction results. The left and right images in each subfigure are whiteboard photos and histograms.

and quantitative results above prove that our paired image post-processing pipeline is well-assured.

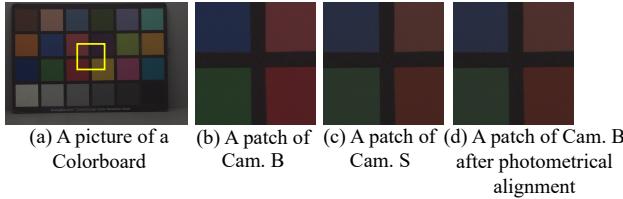


Figure 3: Phometrical alignment results.

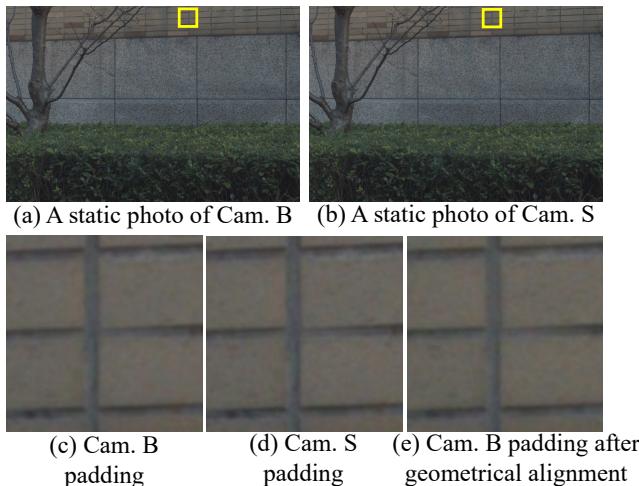


Figure 4: Geometrical alignment results.

C. LBFMG Algorithm

We propose a local blur foreground mask generator, LBFMG, to generate the ground truths for the training of lo-

Table 1: Quantitive results of image post-processing

Post-processing	ΔL	\uparrow PSNR	\uparrow SSIM	Geometrical Error (pixels)
Before	4.550%	26.14	0.5859	3
After	2.870%	36.79	0.9091	1

cal blur mask prediction. LBFMG applies a background subtractor, *BackgroundSubtractorMOG2*¹ to extract the foreground of an image. For the initialization of the background, we utilize static images, which only contain the sharp background without moving objects in the scene. For each local blur mask Msk , the generation procedure is shown in Fig. 5 in the main body of our paper. And more generating details are illustrated in Algorithm 1.

Algorithm 1: LBFMG

- 1: **Load:** *BackgroundSubtractorMOG2* function G , a zero matrix M with the size of 2152×1436 .
 - 2: **Input:** local blurred image B_1 and its corresponding sharp image S_1 , static sharp image S_0 and its corresponding static image B_0 , other image pairs $P = \{(S_k, B_k)\}_{k=2}^{K-1}$ (K : total number of image pairs) of the scene.
 - 3: **Initialize:** put the static image S_0 and B_0 into *BackgroundSubtractorMOG2* to initialize the sharp foreground, $fgmaskS$, and the blur foreground, $fgmaskB$, respectively.
 - 4: **repeat**
 - 5: Sample $k \sim \{2, 3, \dots, K\}$ in order
 - 6: Put a sharp image $S_k \sim P$ into G_1 to update $fgmaskS$
 - 7: Put a local blurred image $B_k \sim P$ into G_2 to update $fgmaskB$
 - 8: **until** $k = K$
 - 9: Put the sharp image S_1 into G_1 to generate $fgmaskS$.
 - 10: Put the local blurred image B_1 into G_2 to generate $fgmaskB$.
 - 11: Calculate M by Equation: $M = (fgmaskS > 1) \text{ OR } (fgmaskB > 1)$.
 - 12: Generate the ground-truth mask of local blurred image B_1 with image erosion and dilation.
 - 13: **Output:** Ground-truth mask Msk_1 .
-

D. Losses and Shift-Invariant Operation

To compensate for the geometrical alignment error (about 1 pixel), we apply a shift-invariant operation when computing the total loss function (Equ. 7 in the main body of our paper), including a mask predicted loss \mathcal{L}_M , and a reconstruction loss \mathcal{L}_{recon} . The shift-invariant operation moves a predicted sharp map $\hat{S} = \mathcal{M}(B)$ for (k, l) pixel, $k, l \in \{-1, 0, 1\}$, where k and l denote the moving pixel towards x and y direction, respectively. “-” means the opposite axis. \mathcal{M} means

¹A function based on Gaussian Mixture-based Background-Foreground Segmentation Algorithm, which can be viewed on https://docs.opencv.org/3.4/d7/d7b/classcv_1_1BackgroundSubtractorMOG2.html

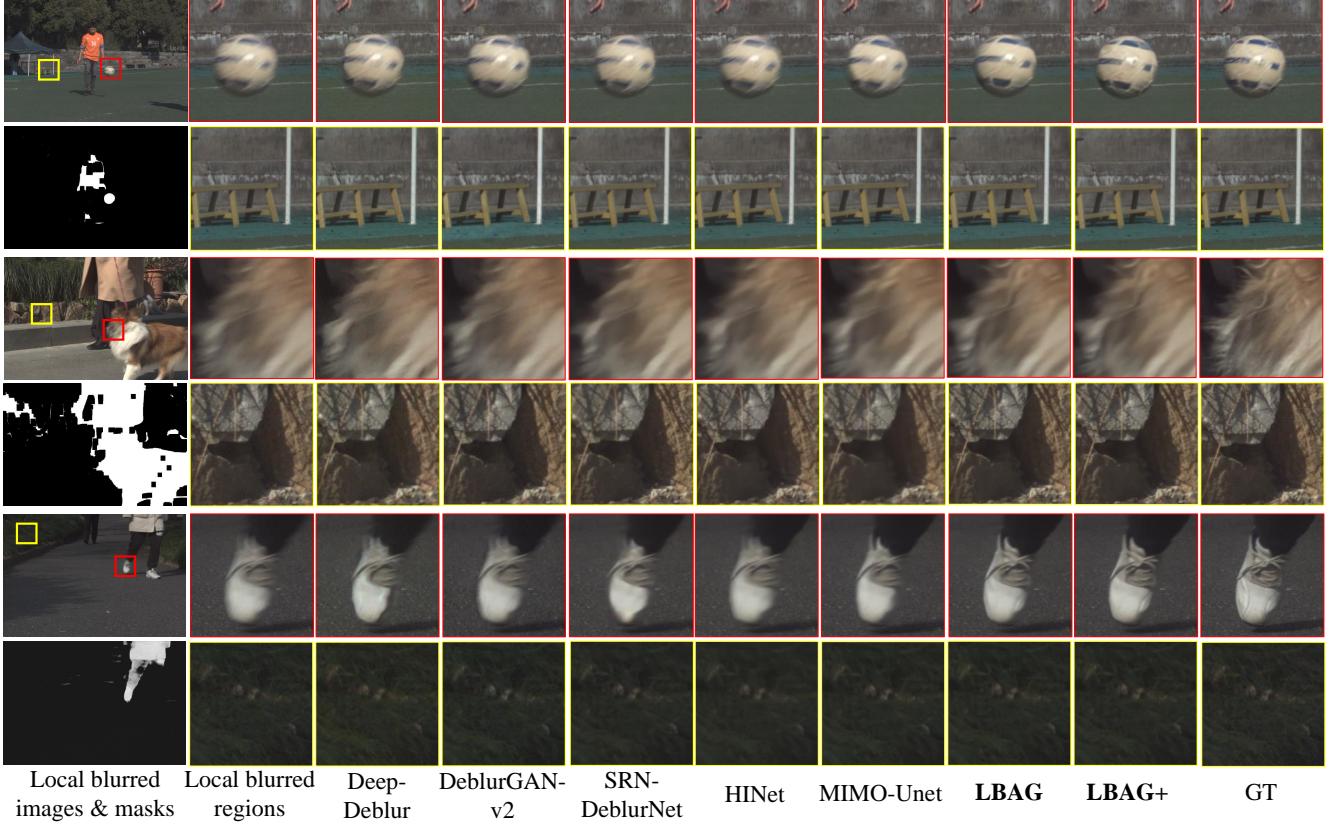


Figure 5: More visual comparisons of different deblurring methods on ReLoBlur dataset: red frames denote locally blurred regions and deblurred regions; yellow frames denote sharp regions and the corresponding regions in deblurred images. The 1st pictures in the 2nd, 4th and 6th row are predicted blur masks by our proposed LBAG.

the proposed local motion deblurring model. B means a locally blurred image. As shown in Fig. 6, we compute the total loss functions on the overlapping area of the ground-truth sharp image S (or the ground-truth mask m) and the shift-invariant predicted sharp map \hat{S} (or the predicted local blur mask \hat{m}). With the shift-invariant operation, the total loss can be also written as:

$$\begin{aligned} \mathcal{L}_{Sh}(\hat{S}, S) = & \lambda_1 \mathcal{L}_{ShM}(\hat{m}, m) + \lambda_2 \mathcal{L}_{ShMAE}(\hat{S}, S) \\ & + \lambda_3 \mathcal{L}_{ShSSIM}(\hat{S}, S) + \lambda_4 \mathcal{L}_{ShMSFR}(\hat{S}, S), \end{aligned} \quad (2)$$

where $\lambda_1 = 0.01$, $\lambda_2 = \lambda_3 = 1$, $\lambda_4 = 0.1$ are constant loss weights. The shift-invariant MSE loss, MAE loss, SSIM loss and MSFR loss follow the derivation below:

$$\mathcal{L}_{ShM}(m, \hat{m}) = \min_{\forall k \in \{-1, 0, 1\}, l \in \{-1, 0, 1\}} (\text{MSE}(m, \text{Shift}_{k,l}(\hat{m}))), \quad (3)$$

$$\mathcal{L}_{ShMAE}(S, \hat{S}) = \min_{\forall k \in \{-1, 0, 1\}, l \in \{-1, 0, 1\}} (\text{MAE}(S, \text{Shift}_{k,l}(\hat{S}))), \quad (4)$$

$$\begin{cases} \mathcal{L}_{SSIM}(S, \hat{S}) = -\frac{1}{N * M} \sum_{x=1}^N \sum_{y=1}^M \text{SSIM}(S(x,y), \hat{S}(x,y)), \\ \mathcal{L}_{ShSSIM}(S, \hat{S}) = \min_{\forall k \in \{-1, 0, 1\}, l \in \{-1, 0, 1\}} (\mathcal{L}_{SSIM}(S, \text{Shift}_{k,l}(\hat{S}))), \end{cases} \quad (5)$$

$$\begin{cases} \mathcal{L}_{MSFR}(S, \hat{S}) = \sum_{k=1}^K \frac{1}{t_k} \|\mathcal{F}(S_k) - \mathcal{F}(\hat{S}_k)\|_1, \\ \mathcal{L}_{ShMSFR}(S, \hat{S}) = \min_{\forall k \in \{-1, 0, 1\}, l \in \{-1, 0, 1\}} (\mathcal{L}_{MSFR}(S, \text{Shift}_{k,l}(\hat{S}))). \end{cases} \quad (6)$$

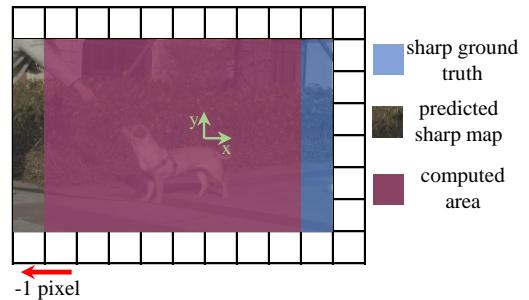


Figure 6: Shift-invariant operation.

E. More Local Deblurring Results

We provide more local deblurring results of comparing the methods on ReLoBlur datasets, as shown in Fig. 5.

F. The Construction of Synthetic Data

Firstly, we select the foreground regions using COCO-Instance Segmentation model² (He et al. 2017). Then, we use *Kornia* filter (Zhang 2019) to blur the selected foreground regions with a 5×5 blur kernel and different moving modes, to simulate local motion blur. The moving modes include translation and rotation.

References

- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.
- Rim, J.; Lee, H.; Won, J.; and Cho, S. 2020. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European Conference on Computer Vision*, 184–201. Springer.
- Zhang, R. 2019. Making convolutional networks shift-invariant again. In *International conference on machine learning*, 7324–7334. PMLR.

²<https://github.com/facebookresearch/detectron2>