Learning to Restore ssTEM Images from Deformation and Corruption

Anonymous ECCV submission

Paper ID 7

1 Restoration Framework

1.1 Interpolation Module

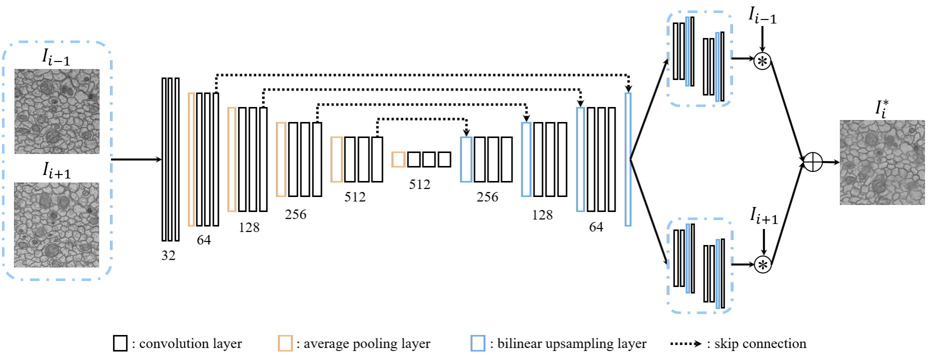

Fig. 1. Network structure of the proposed interpolation module. We concatenate two adjacent images I_{i-1} and I_{i+1} as inputs. The outputs of network are two pairs of 1D kernels, which process inputs in a convolutional manner to generate the interpolated result I_i^*. $\circledast$ denotes convolution. The numbers below each convolution layer denote the channel number.

Our interpolation module is built upon the kernel prediction network (KPN) [1], as shown in Fig. 1. Features from the network are adopted to generate four 1D kernels, *i.e.*, Separable-Conv. We adopt ReLU as the activation function after each convolution layer (Conv). In addition, we use skip connections (Skip-merge) to incorporate features from different layers. Average Pooling (AvePooling) is used to perform downsampling for feature compression and the bilinear interpolation is used for upsampling to recover feature resolution. More details can be found in Table 1.

1.2 Unfolding Module

The structure of unfolding module is listed in Table 2, where a residual variant of U-Net [2] is adopted to implement the optical flow estimation. Each residual

Table 1. Structure of interpolation module.

Blocks	Components	Size of feature maps
input	-	$256 \times 256 \times 6$
down1	3*Conv + AvePooling	$128 \times 128 \times 32$
down2	3*Conv + AvePooling	$64 \times 64 \times 64$
down3	3*Conv + AvePooling	$32 \times 32 \times 128$
down4	3*Conv + AvePooling	$16 \times 16 \times 256$
down5	3(Conv + AvePooling	$8 \times 8 \times 512$
upscaling5	3*Conv + Bilinear + Conv + Skip-merge	$16 \times 16 \times 512$
upscaling4	3*Conv + Bilinear + Conv + Skip-merge	$32 \times 32 \times 256$
upscaling3	3*Conv + Bilinear + Conv + Skip-merge	$64 \times 64 \times 128$
upscaling2	3*Conv + Bilinear + Conv + Skip-merge	$128 \times 128 \times 64$
sub-structure	3*Conv + Bilinear + Conv + Skip-merge	$256 \times 256 \times 51$
output	Separable-Conv	$256 \times 256 \times 1$

Table 2. Structure of unfolding module.

Blocks	Components	Size of feature maps
input	-	$256 \times 256 \times 6$
down1	Conv + Res + Conv + Maxpooling	$128 \times 128 \times 32$
down2	Conv + Res + Conv + Maxpooling	$64 \times 64 \times 64$
down3	Conv + Res + Conv + Maxpooling	$32 \times 32 \times 128$
down4	Conv + Res + Conv + Maxpooling	$16 \times 16 \times 256$
bridge	Conv + Res + Conv	$16 \times 16 \times 512$
upscaling4	Deconv + Skip-merge + Conv + Res + Conv	$32 \times 32 \times 256$
upscaling3	Deconv + Skip-merge + Conv + Res + Conv	$64 \times 64 \times 128$
upscaling2	Deconv + Skip-merge + Conv + Res + Conv	$128 \times 128 \times 64$
upscaling1	Deconv + Skip-merge + Conv + Res + Conv	$256 \times 256 \times 32$
last-layer	Conv	$256 \times 256 \times 2$
output	Warping	$256 \times 256 \times 1$

block (Res) contains three stacked convolution layers and a residual skip connection. Maxpooling layer is used to perform downsampling and deconvolution layer (Deconv) is used to upsample feature maps.

1.3 Fusion Module

As listed in Table 3, our proposed fusion module is implemented by a simplified version of U-Net structure [2]. The skip connection is used to concatenate features from different layers, termed as Skip-concat.

2 Restoration Results

To evaluate the superiority and the generalization capability of our proposed method, we provide supplementary restoration results on *synthetic* (as shown in Figs. 2 and 3) and *realistic* samples (as shown in Figs. 4 and 5).

Table 3. Structure of fusion module.

Blocks	Components	Size of feature maps
input	-	256 × 256 × 6
down1	2*Conv + Maxpooling	128 × 128 × 32
down2	2*Conv + Maxpooling	64 × 64 × 64
down3	2*Conv + Maxpooling	32 × 32 × 128
upscaling3	2*Conv + Deconv	64 × 64 × 128
upscaling2	Skip-concat + 2*Conv + Deconv	128 × 128 × 64
upscaling1	Skip-concat + 2*Conv + Deconv	256 × 256 × 32
output	Skip-concat + 2*Conv	256 × 256 × 1

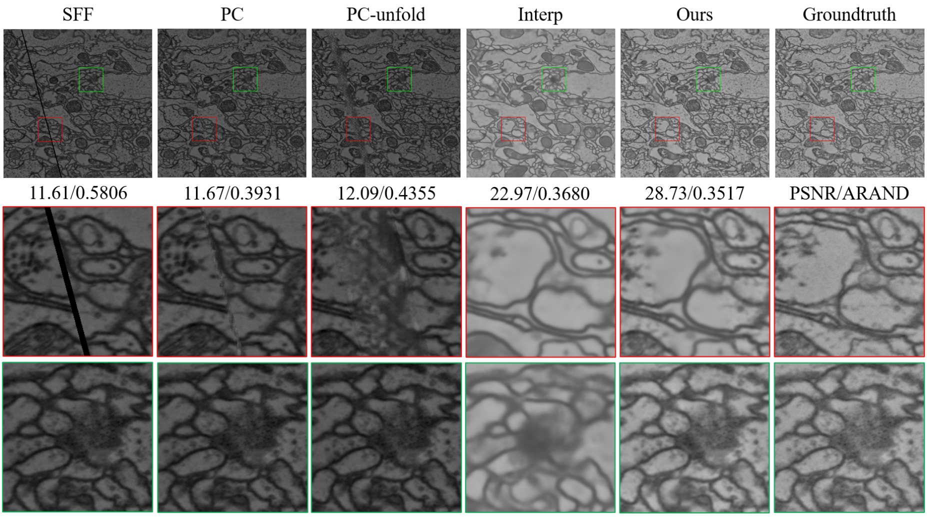

Fig. 2. Visual comparison of restoration results on *synthetic* samples.

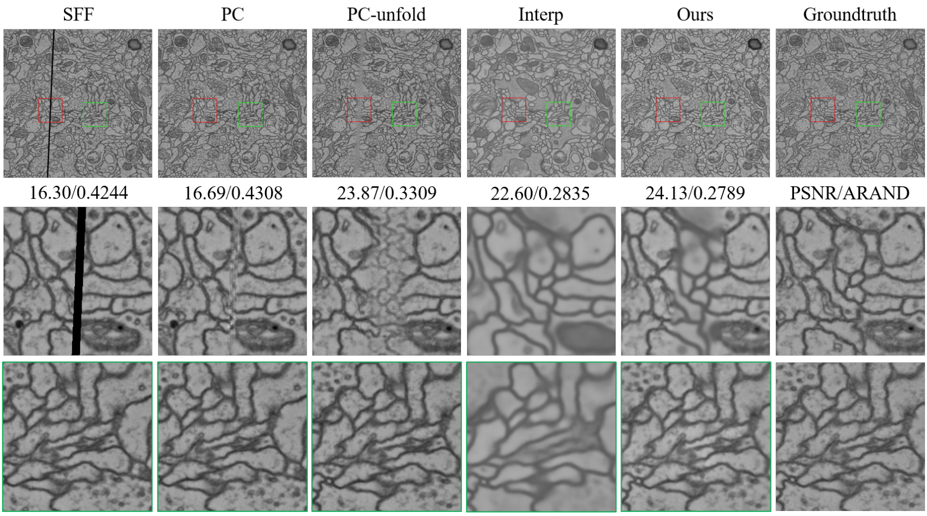

Fig. 3. Visual comparison of restoration results on *synthetic* samples.

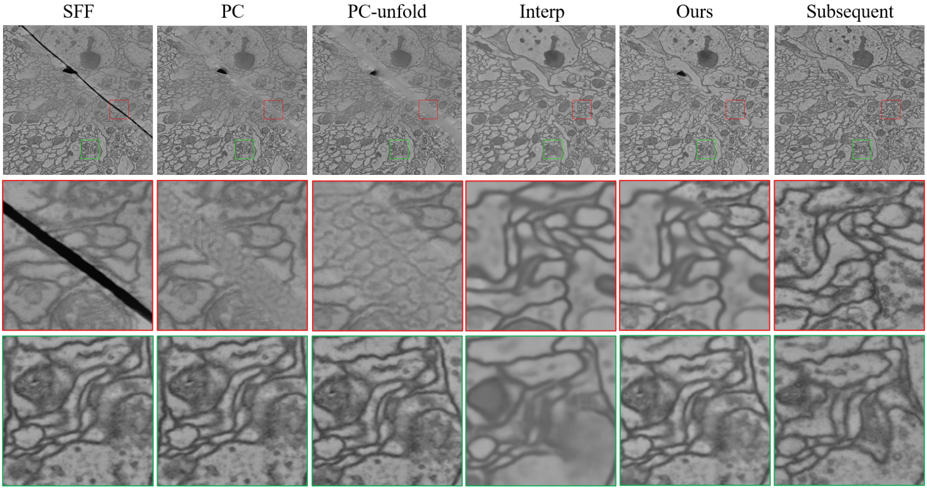

Fig. 4. Visual comparison of restoration results on *realistic* samples.

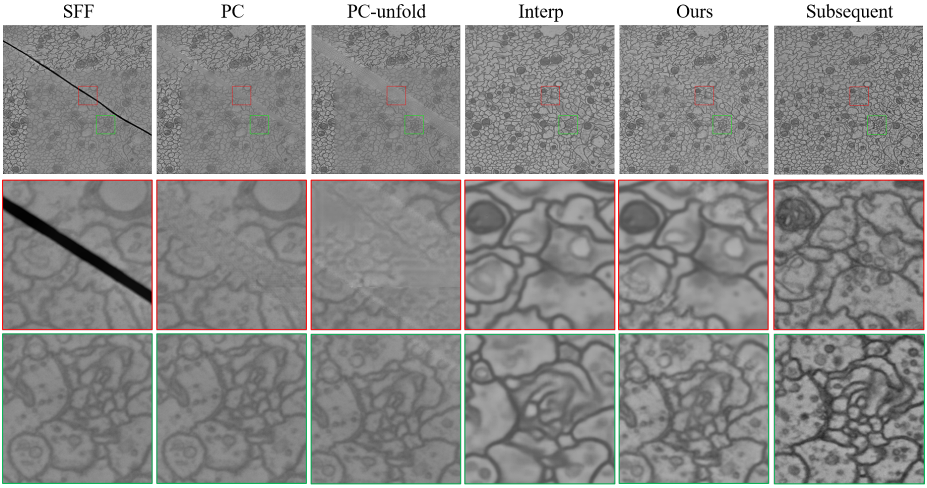

Fig. 5. Visual comparison of restoration results on *realistic* samples.

References

1. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive separable convolution. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 261–270 (2017)
2. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)