Supplementary Material: Neural Image Representations for Multi-Image Fusion and Layer Separation

Seonghyeon Nam, Marcus A. Brubaker, and Michael S. Brown

York University
snam0331@gmail.com, {mab,mbrown}@eecs.yorku.ca

1 Implementation Details

We describe our implementation in detail for various multi-image layer separation tasks. In particular, we describe the settings and hyperparameters that we found to give the best qualitative results.

Moiré removal. For g_{θ_T}, we use an MLP with two hidden layers, 256 hidden units, and a ReLU activation function. We also initialize the bias of the output layer to the identity transformation of the input coordinates. For $f^1_{\theta_O}$, we use a SIREN [15] with four hidden layers and 256 hidden units. For $f^2_{\theta_U}$, we use a SIREN with four hidden layers and 128 hidden units. We set λ_{Interf} and λ_{Excl} to 0.001 and 0.002, respectively. At training time, we update the networks for 3,000 iterations using an Adam optimizer [10] with the learning rate of 0.0001. We use the $\frac{1}{4}$ of the size of the input image as the batch size.

Obstruction removal. We use a SIREN with 4 hidden layers and 256 hidden units for g_{θ_T}, $f^1_{\theta_O}$, and $f^2_{\theta_U}$. We set λ_{Interf}, λ_{TVFlow}, and λ_{Excl} to 0.1, 0.02, and 0.001, respectively. We use the same optimizer and learning rate settings as those in the moiré removal. We use 5,000 and the $\frac{1}{32}$ of the size of the input image as the number of iterations and the batch size.

Rain removal. We use a SIREN with 5 hidden layers and 256 hidden units for g_{θ_T}, $f^1_{\theta_O}$, and $f^2_{\theta_U}$. We set λ_{Interf} and λ_{TVFlow} to 0.01 and 0.02, respectively. Similarly, the number of training iterations is set to 5,000, and the batch size is set to $\frac{1}{32}$ of the size of the input image.

2 Additional Experiments

2.1 Learning motion in NIRs vs. conventional motion estimation

In our approach, the motion of a scene is optimized jointly with a layer separation task from scratch. To show the effectiveness of it, we compare our method with a conventional motion estimation. Specifically, we use a homography-based NIR in the task of moiré removal. For a baseline, we replace the MLP for estimating

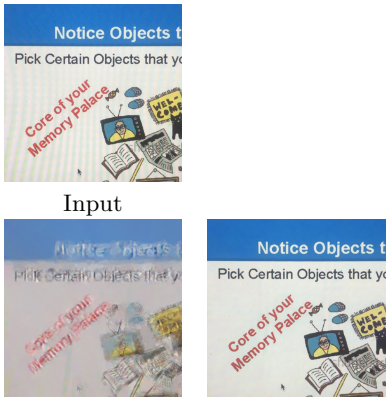

Fig. 1: Comparison with conventional homography estimation. We adopt a homography estimation method [7] in a homography-based NIR.

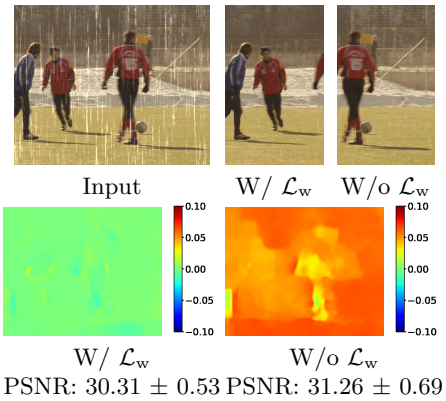

Fig. 2: Analysis on w. We apply an additional loss $\mathcal{L}_w = \sum \|w\|_1$ and compare it with the original model. For PSNRs, we train each model 5 times and aggregates results.

homography matrices with a conventional homography estimation method [4], and use the center frame as a reference. We train both the baseline and the original NIR on the same synthesized dataset. As a result, the baseline achieves an average PSNR of 23.94 and SSIM of 0.7690, while those of the original NIR are 38.68 and 0.9751. Fig. 1 shows a qualitative comparison on real burst images. As can be seen, our result is visually plausible compared to the baseline result. Since the input scene is highly corrupted by interference patterns, it is difficult to estimate the motion of the underlying scene accurately using the conventional motion estimation method. In this case, it is often required to do an additional refinement [12]. On the other hand, our method achieves a better performance by jointly learning the motion and layer separation in a single framework of NIR.

2.2 Further Analysis on w.

In Fig. 2, we additionally conduct an experiment to further understand the space represented by w. Specifically, we explicitly enforce our model to represent most of pixel values at $w = 0$ by adding the regularization $\mathcal{L}_w = \sum \|w\|_1$. We also evaluate the performance by training the same model five times and computing the mean and standard deviation of PSNRs. As can be seen, the model with $\mathcal{L}_w$ uses 0 as the center of w space, while the original model uses an arbitrary value varying according to initialization. However, the output quality is similar, as shown in both qualitative and quantitative results. This finding implies that the smoothness prior, driven by $\mathcal{L}_{\text{TVFlow}}$, is more important to learn multi-image representation than the center of w.

Fig. 3: Additional application to burst image denoising on the images in [14].

Fig. 4: Additional application to joint image demosaicing and burst super-resolution on the images in [3].

2.3 Other Applications

Burst image denoising. We additionally apply our method to burst image denoising. To do this, we cast the problem as a layer separation that is to decompose the signal of images into the underlying scene and noise. In this task, we use an occlusion-free flow-based neural representation as a function of the scene, which is reasonable since the motion of burst images is typically small but may not be planar. We use the same objective as used in moiré removal. As shown in Fig. 3, we evaluate the performance using a burst image denoising dataset in [14]. Our method demonstrates a competitive result compared with a burst image denoising method [14], which indicates that our multi-image fusion based on NIRs is also useful to remove random noise signals.

Joint demosaicing and burst super-resolution. To further understand the effectiveness of our multi-image fusion, we apply our method to a more challenging task: burst image super-resolution. Since our method deals with a real-valued coordinate space, it is technically applicable to a sub-pixel registration in burst

super-resolution. For the experiment, we use a real burst dataset in [3,2] that contains sequences of 14 raw burst images for testing. To reconstruct RGB values from Bayer color filter array (CFA) images, we multiply a channel mask to the 3-channel output of a NIR before comparing it to ground truth. At inference time, we take all channels of RGB output, where missing channels are interpolated. For implementation, we use an occlusion-free flow-based NIR. Fig. 4 shows results of a joint demosaicing and burst super-resolution (x4). The original image is upsampled using bicubic interpolation, and all images are post-processed using the code in [2]. As can be seen, our results show clearer details of images than bicubic results but are still blurry. We conjecture that the sub-pixel alignment may not be accurate since the optimization only relies on the error of pixel intensities. Thus, an interesting direction of follow-up research would be to add more loss functions and regularization to assist with sub-pixel registration. Image noise is also reduced in our results without an explicit noise layer, but using our two-stream architecture would improve the performance.

2.4 Additional Comparisons

Table 1 compares quantitative results of obstruction removal on three controlled sequences in [17]. We use the baseline results reported in [13]. Like other tasks, our method (without supervision) achieves comparable results to the unsupervised method, but does not outperform supervised approaches.

Fig. 5 shows burst images used for the layer separation applications addressed in the main paper, and Fig. 6 to Fig. 9 show additional qualitative results. Particularly, the last example in Fig. 7 shows a failure case. In the example, our method fails to remove the floor of the reflected scene. Since our layer separation relies on the difference of scene motion, our method does not work well when the movement of the two layers is similar. Note that we use an occlusion-aware flow-based NIR for the results in Fig. 8 due to occlusion and disocclusion. Even though we choose one of three models for each application, it is generally desirable to use the best model to fit the problem's setting. For the last example in Fig. 9, we use synthetic images to show our result on a dynamic scene.

Method	Stone		Toy		Hanoi	
	SSIM	NCC	SSIM	NCC	SSIM	NCC
[11]	0.7993	0.9334	0.6877	0.7068	N/A	N/A
[17]	N/A	0.9738	N/A	0.8985	N/A	0.9921
[1]	0.7942	0.9351	0.7569	0.7972	N/A	N/A
[12]	0.8598	0.9632	0.7696	0.9477	0.9238	0.9929
[13]	0.8635	0.9315	0.8494	0.9542	0.9457	0.9938
Ours	0.8617	0.9451	0.7700	0.8136	0.9045	0.9840

Table 1: Quantitative result of obstruction removal on the data in [17]

Fig. 5: Examples of burst images used in our paper. From top to bottom, each row shows burst images for moiré, reflection, fence, and rain removal, respectively.

Fig. 6: Qualitative comparison of moiré removal on real images.

Fig. 7: Qualitative results of reflection removal on real images in [11].

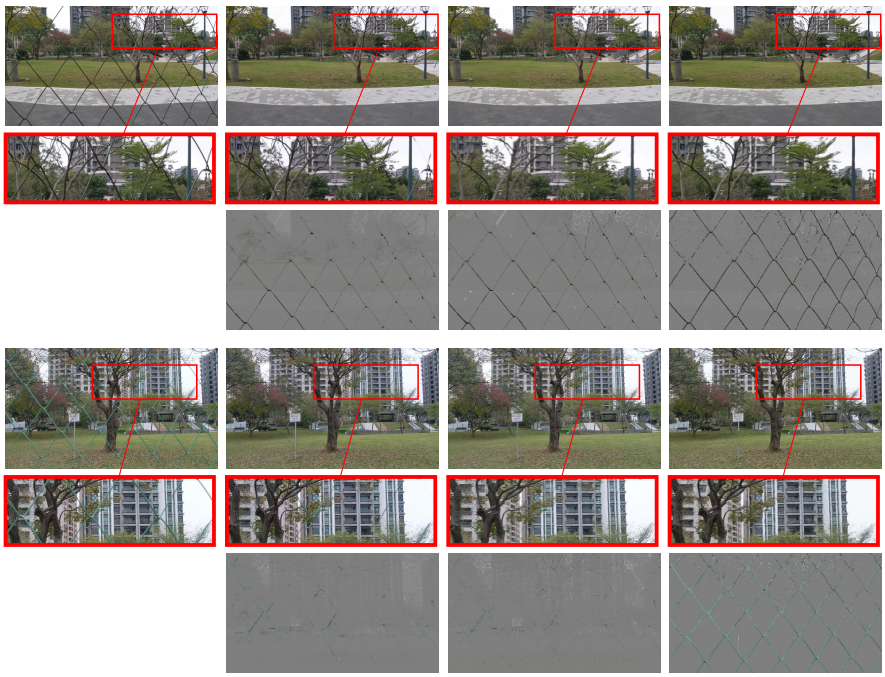

Fig. 8: Qualitative comparison of fence removal on real images in [13].

Fig. 9: Qualitative comparison of rain removal on real images in NTURain [5].

References

1. Alayrac, J.B., Carreira, J., Zisserman, A.: The visual centrifuge: Model-free layered video representations. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2457–2466 (2019) 4, 7
2. Bhat, G., Danelljan, M., Timofte, R.: Ntire 2021 challenge on burst super-resolution: Methods and results. In: IEEE Conf. Comput. Vis. Pattern Recog. Worksh. pp. 613–626 (2021) 4
3. Bhat, G., Danelljan, M., Van Gool, L., Timofte, R.: Deep burst super-resolution. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9209–9218 (2021) 3, 4
4. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. Int. J. Comput. Vis. **74**(1), 59–73 (2007) 2
5. Chen, J., Tan, C.H., Hou, J., Chau, L.P., Li, H.: Robust video content alignment and compensation for rain removal in a cnn framework. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6286–6295 (2018) 9
6. Gandelsman, Y., Shocher, A., Irani, M.: Double-DIP: Unsupervised image decomposition via coupled deep-image-priors. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11026–11035 (2019) 6, 7
7. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003) 2
8. Jiang, T.X., Huang, T.Z., Zhao, X.L., Deng, L.J., Wang, Y.: Fastderain: A novel video rain streak removal method using directional gradient priors. IEEE Trans. Image Process. **28**(4), 2089–2102 (2018) 9
9. Kim, S., Nam, H., Kim, J., Jeong, J.: C3net: Demoireing network attentive in channel, color and concatenation. In: IEEE Conf. Comput. Vis. Pattern Recog. Worksh. pp. 426–427 (2020) 6
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 1
11. Li, Y., Brown, M.S.: Exploiting reflection change for automatic reflection removal. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2432–2439 (2013) 4, 7
12. Liu, Y.L., Lai, W.S., Yang, M.H., Chuang, Y.Y., Huang, J.B.: Learning to see through obstructions. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 14215–14224 (2020) 2, 4, 8
13. Liu, Y.L., Lai, W.S., Yang, M.H., Chuang, Y.Y., Huang, J.B.: Learning to see through obstructions with layered decomposition. arXiv e-prints pp. arXiv–2008 (2020) 4, 7, 8
14. Liu, Z., Yuan, L., Tang, X., Uyttendaele, M., Sun, J.: Fast burst images denoising. ACM Trans. Graph. **33**(6), 1–9 (2014) 3
15. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Adv. Neural Inform. Process. Syst. **33** (2020) 1
16. Xu, D., Chu, Y., Sun, Q.: Moiré pattern removal via attentive fractal network. In: IEEE Conf. Comput. Vis. Pattern Recog. Worksh. pp. 472–473 (2020) 6
17. Xue, T., Rubinstein, M., Liu, C., Freeman, W.T.: A computational approach for obstruction-free photography. ACM Trans. Graph. **34**(4), 1–11 (2015) 4