Fast Non-blind Deconvolution via Regularized Residual Networks with Long/Short Skip-Connections (Supplemental Material)

Hyeongseok Son
POSTECH

Seungyong Lee
POSTECH

1. Calculating a gradient of the regularization term

For updating weights of a network in the backpropagation procedure, a gradient equation of a regularization term should be defined. For calculating the gradient, we assume each pixel is locally independent. The gradient of the sparse prior at each pixel in the loss layer is then calculated using four image gradients between the pixel and its four neighbors. That is,

$$\begin{aligned}\frac{\partial R(f)}{\partial f_{i,j}} =&\alpha\Bigg(\sum_{k,l\in\{+1\}} w_{i,j} p \left|f_{i,j} - f_{i+k,j+l}\right|^{p-1} \\ &- \sum_{k,l\in\{-1\}} w_{i+k,j+l} p \left|f_{i,j} - f_{i+k,j+l}\right|^{p-1}\Bigg).\end{aligned} \tag{1}$$

Notations are the same as in Eq. (1) in the main paper.

2. Wiener filter parameter estimation

Wiener filter [17] can be represented by

$$\widehat{X} = \left[\frac{1}{V}\frac{|V|^2}{|V|^2+K}\right] \odot Y_1, \tag{2}$$

where $K = \frac{var(n)}{var(x)}$ is the parameter to control the output quality. Other notations are the same as in Eq. (3) in the main paper.

In the real applications which need the non-blind deconvolution, we do not know the variances of both latent image and noise. To avoid an exhaust search for the optimal K, we develop simple algorithms to estimate the variances. The variance of noise is estimated from the difference between the input and the median-filtered images. This simple method works properly except when the noise level is very low, where most regions of images are piecewise smooth. We can assume the variance of the latent image bigger than that of the input image as the image variance is reduced by blurring. We estimate the latent image variance by increasing the input image variance using a simple curve defined as:

$$estimated\_var(x) = \frac{\sqrt{9var(y)}}{8} \tag{3}$$

We use this estimated K for Wiener filter to generate our dataset for training. Although our estimation algorithm for K is very rough, this approach helps us produce deconvolution results with similar artifact behaviors needed for effective training of our network. The same algorithm is used to estimate K when we apply Wiener filter to the input blurry image in the preprocessing step of our deconvolution framework. Sometimes, especially for real blurry images with inaccurate kernels, the estimated K may not produce the best deconvolution results. In that case, a better K can be found by few trials to search from the estimated value.

3. Additional experiments

3.1. Dealing with cropped blurry images

In the main paper, we used a circular convolution, preserving the image size, to make a synthetic blurry image as MLP [15] did. However, some non-blind deconvolution methods [5, 14] used a cropped convolution, which crops image boundary after convolution, in their papers. This different condition could make inconsistent experimental results for other methods. We did an additional experiment using the cropped convolution as shown in Fig. 1 for a pair comparison. Wiener filter may generate ringing artifacts in this case, so we use a padding technique [12] which makes the image boundary circularly smooth for preventing it. After applying it, our method still shows better performance than [5, 14] as shown in Fig. 1.

3.2. Blind deconvolution

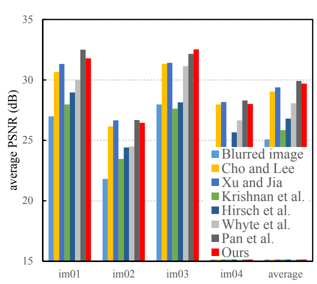
In the middle of the kernel estimation, the initial and intermediate blur kernels are rather inaccurate, so we used a strong regularization parameter ($\alpha = 0.08$) for training our network. We then compared our image deblurring framework with Cho and Lee [3], Levin *et al.*[11], Xu and Jia

(a) input (b) Fortunato [5] (c) CSF [14] (d) ours (e) ground-truth

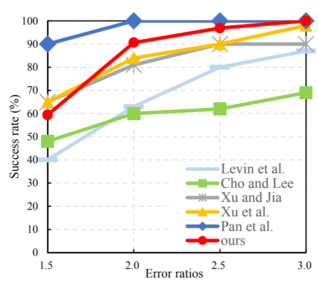
Figure 1. Qualitative comparison with boundary-cropped images. The blurred image was blurred by kernel (a) and (e) in the paper and weak noise (1%). PSNRs of the top rows are 26.99, 26.54, **28.97** dB from (b) to (d). PSNRs of the bottom row are 27.66, 27.35, **30.10** dB from (b) to (d).

(a) ground-truth (b) L_0 (strong) [22] (c) L_0 (weak) [22] (d) dark channel [13] (e) our method

Figure 3. Results of blind deconvolution methods for a noisy image. The top row shows interim results and estimated kernels. The bottom row shows the final result of each method. We took the input image from [21].



(a) Results on benchmark [KHM*12]



(b) Results on benchmark [LWDF09]

Figure 2. Quantitative comparison on two blind deblurring benchmarks [8, 10]. Our method is comparable to the state-of-the-art methods.

[20], Krishnan *et al.*[9], Hirsch *et al.*[7], Whyte *et al.*[16], Xu *et al.*[22], and Pan *et al.*[13] using two image deblurring benchmarks [8, 10]. We set $K = 0.02$ (for [8]) and $K = 0.06$ (for [10]) in Eq. (2) in all experiments. Fig. 2 shows our framework produces results comparable to the state-of-the deblurring methods. Our framework also reduces the processing time from 12.48s of the original version using L0 deconvolution to 8.84s. Our method is a runner-up in the both benchmarks, but our method is much faster than [13], which spends 707.82s in the same environment (a 700×500 image, MATLAB versions).

Our image deblurring framework is especially effective for handling noisy blurred images due to the robustness of our deconvolution method against noise (Fig. 3). For a noisy image, a weak regularization in [22, 13] would amplify noise and strong regularization would produce a too smooth result, prohibiting estimation of an accurate kernel. In our framework, the network can remove strong noise while preserving image structures, enabling accurate kernel estimation even in a noisy image.

3.3. Denoising

Our non-blind deconvolution framework is strongly related to the denoising problem. Denoising methods based

data set	noise (σ)	BM3D [4]	WNNM [6]	PCLR [2]	PGPD [19]	Ours
Set 14	0.039	34.19	**34.48**	34.46	34.23	34.26
	0.12	28.46	28.77	28.70	28.53	**29.12**
	0.20	26.09	26.34	26.31	26.21	**26.76**
	0.27	24.64	24.79	24.78	24.71	**25.39**

Table 1. Quantitative comparison with noisy images. We uses a 10 layer network. Note that $\sigma = 0.039 = \frac{10}{255} = 3.9\%$ in the image range $(0 \sim 1)$.

on deep learning usually utilize auto encoders [18, 1] rather than convolutional networks. We tested whether our network architecture is appropriate for denoising. We used a nested residual network with 10 layers and trained using image pairs with four different noise levels. Table 1 shows that our network also works well for denoising. Our network shows slightly higher performance than other existing methods except at very low noise levels. Note that the noise is amplified by Wiener filter in our deconvolution framework, so this characteristic does not affect our deconvolution performance.

References

[1] F. Agostinelli and M. R. Anderson. Adaptive multi-column deep neural networks with application to robust image denoising. In *Proc. NIPS*, 2013.

[2] F. Chen, L. Zhang, and H. Yu. External patch prior guided internal clustering for image denoising. In *Proc. ICCV*, 2015.

[3] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph*, 28(5):1, 2009.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising with block-matching and 3d filtering. In *Proc. SPIE Electronic Imaging*, 2006.

[5] H. E. Fortunato and M. M. Oliveira. Fast high-quality non-blind deconvolution using sparse adaptive priors. *The Visual Computer*, 30(6-8):661–671, 2014.

[6] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *Proc. CVPR*, 2014.

[7] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In *Proc. ICCV*, 2011.

[8] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *Proc. ECCV*, 2012.

[9] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proc. CVPR*, 2011.

[10] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. CVPR*, 2009.

[11] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proc. CVPR*, 2011.

[12] R. Liu and J. Jia. Reducing boundary artifacts in image deconvolution. In *Proc. ICIP*, 2008.

[13] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *Proc. CVPR*, 2016.

[14] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *Proc. CVPR*, 2014.

[15] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *Proc. CVPR*, 2013.

[16] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *Int'l Journal of Computer Vision*, 98(2):168–186, 2012.

[17] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1964.

[18] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Proc. NIPS*, 2012.

[19] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *Proc. ICCV*, 2015.

[20] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proc. ECCV*, 2010.

[21] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Proc. NIPS*, 2014.

[22] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proc. CVPR*, 2013.