Learning Fully Convolutional Networks for Iterative Non-blind Deconvolution Supplemental Material

Anonymous CVPR submission

Paper ID 1464

1. Overview

In this supplemental material, we provide the back propagation of the deconvolution module in Section 2. Section 3 shows the effect of the proposed FCNN in non-blind deconvolution. We further train our network using estimated kernels by state-of-the-art deblurring methods and show that our method can also work well on both ground truth kernels and estimated kernels for non-blind deconvolution in Section 4. In Section 5, we show that our method can be applied to the non-blind deconvolition where the blur is Gaussian. In the end, we show more visual results of deblurring synthetic and real blurry images.

2. Back Propagation of Deconvolution Module

The forward propagation of deconvolution module is

$$x = \mathcal{F}^{-1}\left(\frac{\gamma\overline{\mathcal{F}(k)}\mathcal{F}(y) + \sum_{l=h,w}\overline{\mathcal{F}(p_l)}\mathcal{F}(z_l)}{\gamma\overline{\mathcal{F}(k)}\mathcal{F}(k) + \sum_{l=h,w}\overline{\mathcal{F}(p_l)}\mathcal{F}(p_l)}\right). \tag{1}$$

The input of deconvolution module is output of the FCNN (*i.e.*, z_l) and the output is deconvoluted sharp image x. To derive the gradient of (1) in the back propagation step, we use matrix-vector form to rewrite it by

$$\mathbf{x} = \mathbf{A}\mathbf{B} + \sum_{l=h,w}\mathbf{A}\mathbf{C}_l\mathbf{z}_l, \tag{2}$$

where $\mathbf{A} = \mathbf{S}\mathrm{diag}\left(\gamma\overline{\mathcal{F}(k)}\mathcal{F}(k) + \sum_{l=h,w}\overline{\mathcal{F}(p_l)}\mathcal{F}(p_l)\right)^{-1}$, $\mathbf{B}$ is the vector form of $\gamma\overline{\mathcal{F}(k)}\mathcal{F}(y)$ and $\mathbf{C}_l = \mathrm{diag}(\overline{\mathcal{F}(p_l)})\mathbf{F}$. $\mathrm{diag}(\bullet)$ is an operator that transforms a vector into a diagonal matrix, $\mathbf{F}$ and $\mathbf{S}$ are the matrix multiplication forms of FFT and IFFT, $\mathbf{x}$ and $\mathbf{z}_l$ are the vector forms of x and z_l,

Based on (2), the model of deconvolution module in the back propagation is

$$\begin{aligned}\mathbf{\Delta}_{z_l} =& \mathbf{C}_l^\top A^\top \mathbf{\Delta}_x \\ =& \mathbf{F}\mathrm{diag}\left(\overline{\mathcal{F}(p_l)}\right)\mathrm{diag}\left(\gamma\overline{\mathcal{F}(k)}\mathcal{F}(k) + \sum_{l=h,w}\overline{\mathcal{F}(p_l)}\mathcal{F}(p_l)\right)^{-1}\mathbf{S}\mathbf{\Delta}_x,\end{aligned} \tag{3}$$

where $\mathbf{\Delta}_{z_l}$ and $\mathbf{\Delta}_x$ is the vector form of Δ_{z_l} and Δ_x defined in the manuscript. The computation of (3) can be achieved by FFTs

$$\Delta_{z_l} = \mathcal{F}\left(\frac{\overline{\mathcal{F}(p_l)}\mathcal{F}^{-1}(\Delta_x)}{\gamma\overline{\mathcal{F}(k)}\mathcal{F}(k) + \sum_{l=h,w}\overline{\mathcal{F}(p_l)}\mathcal{F}(p_l)}\right), \tag{4}$$

To compute the update of hyper-parameter γ, we rewrite (1) as

$$\mathbf{x} = \mathbf{S}\frac{\gamma\mathbf{D} + \mathbf{E}}{\gamma\mathbf{G} + \mathbf{H}}, \tag{5}$$

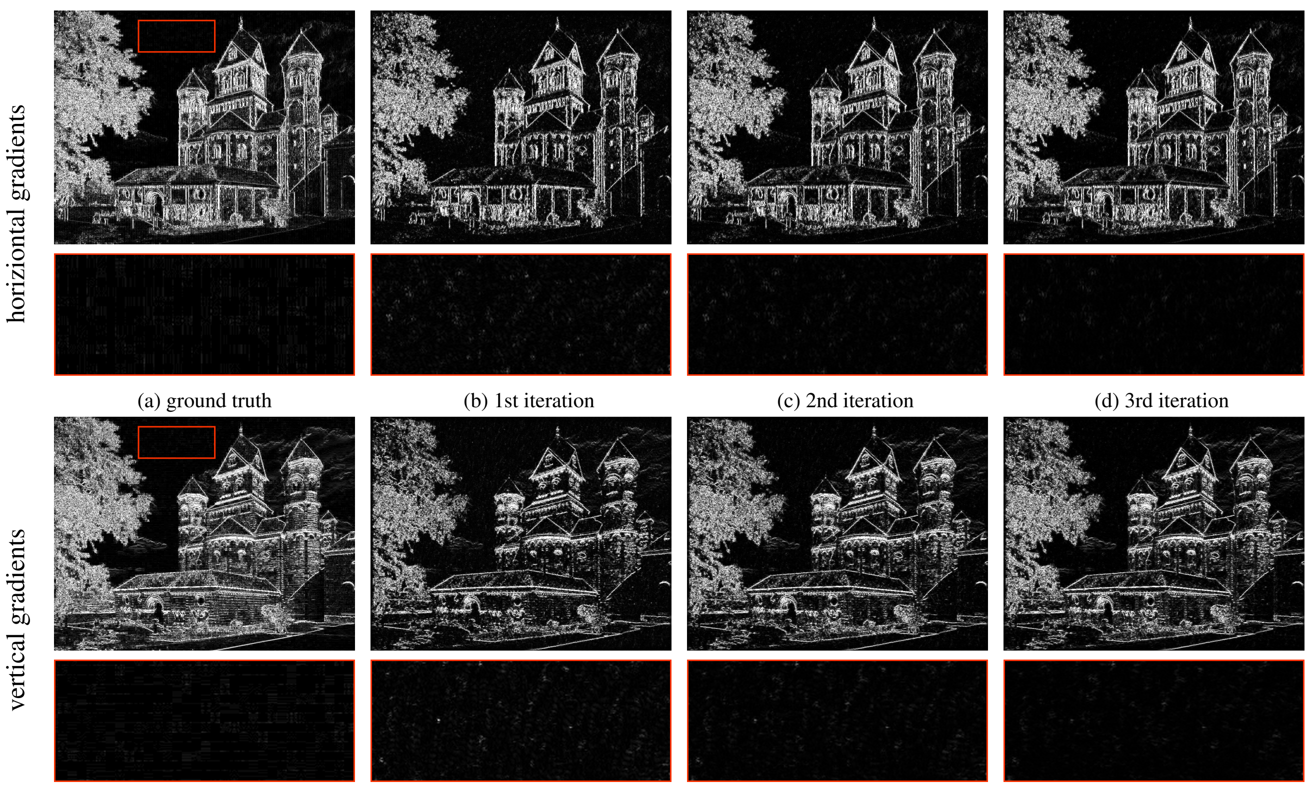
where $\mathbf{D}$, $\mathbf{H}$, $\mathbf{E}$ and $\mathbf{G}$ denote the vector forms of D, H, E and G, respectively, in which $D = \overline{\mathcal{F}(k)}\mathcal{F}(y)$, $E = \sum_{l=h,w} \overline{\mathcal{F}(p_l)}\mathcal{F}(z_l)$, $G = \overline{\mathcal{F}(k)}\mathcal{F}(k)$ and $H = \sum_{l=h,w} \overline{\mathcal{F}(p_l)}\mathcal{F}(p_l)$.

So the update of γ is

$$\begin{aligned}\Delta_\gamma =& \frac{\partial \mathbf{x}^\top}{\partial \gamma}\mathbf{\Delta}_x \\ =& \left(\mathbf{S}\frac{\partial \frac{\gamma\mathbf{D}+\mathbf{E}}{\gamma\mathbf{G}+\mathbf{H}}}{\partial\gamma}\right)^\top \mathbf{\Delta}_x \\ =& \left(\frac{\mathbf{DH}-\mathbf{EG}}{(\gamma\mathbf{G}+\mathbf{H})^2}\right)^\top \mathbf{S}\mathbf{\Delta}_x \\ =& \left(\frac{\mathbf{DH}-\mathbf{EG}}{(\gamma\mathbf{G}+\mathbf{H})^2}\right)^\top \mathcal{F}^{-1}(\mathbf{\Delta}_x),\end{aligned} \tag{6}$$

3. Effect of FCNN in the Proposed Iterative Deconvolution

In Section 4.1 of the manuscript, we compare the deconvolution outputs of different iterations. In this supplemental material, we further discuss its effect of FCNN by quantitative results. Figure 1 shows some intermediate results generated by FCNN, which demonstrates that the FCNN is able to remove noise and preserve fine details.



(a) ground truth (b) 1st iteration (c) 2nd iteration (d) 3rd iteration

(e) ground truth (f) 1st iteration (g) 2nd iteration (h) 3rd iteration

Figure 1. FCNN gradient denoising from different iterations with 1% noise. With the iterations, the noise is removed and fine structures are preserved, which demonstrates the effect of FCNN in the iterative non-blind deconvolution.

4. Training with Estimated Kernels

In the submitted paper, our network is trained with the ground truth kernels. However it cannot work well if kernels are not accurately estimated such as [2, 10].

To overcome this problem, we train the proposed network with estimated kernels in this section. We randomly crop 20000 patches from [1], half of the patches use the ground truth kernels and the rest patches use estimated kernels in the training. We use [2] to estimate kernels from blurred images with 1% noise. We test different non-blind deblurring methods with 1% noise with ground truth kernels as well as kernels estimated from [5, 4, 9, 10].

As shown in Table 1, our network (that is trained with estimated kernels) is comparable with [11] and performs better than other methods when using the estimated kernels by [5, 4, 9, 10]. Figure 2 shows that our network is able to preserve structures and remove noise. In addition, our network (that is trained with estimated kernels) is able to partially remove ringing artifacts while the network trains with ground truth kernels cannot.

Table 1. Average PSNR and SSIM for 1% noise. The model that is trained with ground truth blur kernels and estimated blur kernels outperforms the model which is trained using only ground truth blur kernels when testing with estimated kernels. "Ours with GT" means that the model is trained with ground truth blur kernels. "Our with GT&E" means that the model is trained with ground truth blur kernels and estimated blur kernels. The best performance is marked in red and the second best is marked in blue.

blur kernel	ground truth	Pan dark channel [5]	Pan robust [4]	Xu L0 [9]	Zhong noise [10]
HL [3]	31.57/0.87	29.81/0.85	29.94/0.84	29.35/0.84	26.56/0.78
EPLL [11]	**33.00/0.89**	30.48/0.87	**30.61/0.87**	**29.96/0.86**	26.44/0.79
MLP [7]	31.82/0.86	27.93/0.77	28.76/0.80	27.53/0.76	23.09/0.59
CSF [6]	31.93/0.87	30.04/0.86	30.22/0.86	29.56/0.85	26.79/0.79
Ours with GT	32.82/0.90	30.09/0.87	30.39/0.87	29.58/0.86	26.04/0.78
Our with GT&E	32.14/0.88	**30.52/0.86**	30.44/0.86	29.84/0.85	**27.03/0.79**

5. Gaussian Blur

In this section, we show that the proposed method is able to deal with non-blind deconvolution when the blur is Gaussian. We note that [7] provides MLP model trained from Gaussian blur with 4% noise. For fairness. we also compare different non-blind deblurring methods with ground truth kernels and 4% noise and the 80 clean sharp images are also from [8]. We synthesize blurred images with Gaussian kernel and add Gaussian noise. Different from [7], where the network is trained with fixed Gaussian kernels, the standard deviations of Gaussian kernel are randomly sampled from 1 to 3.5 when training our network.

Table 2 shows that the proposed network outperforms other methods.

Table 2. Average PSNR and SSIM for 4% noises with Gaussian blur. The standard deviation of the blur kernel is either 1.6 or 3. Our three iterations network outperforms other methods in this scenario.

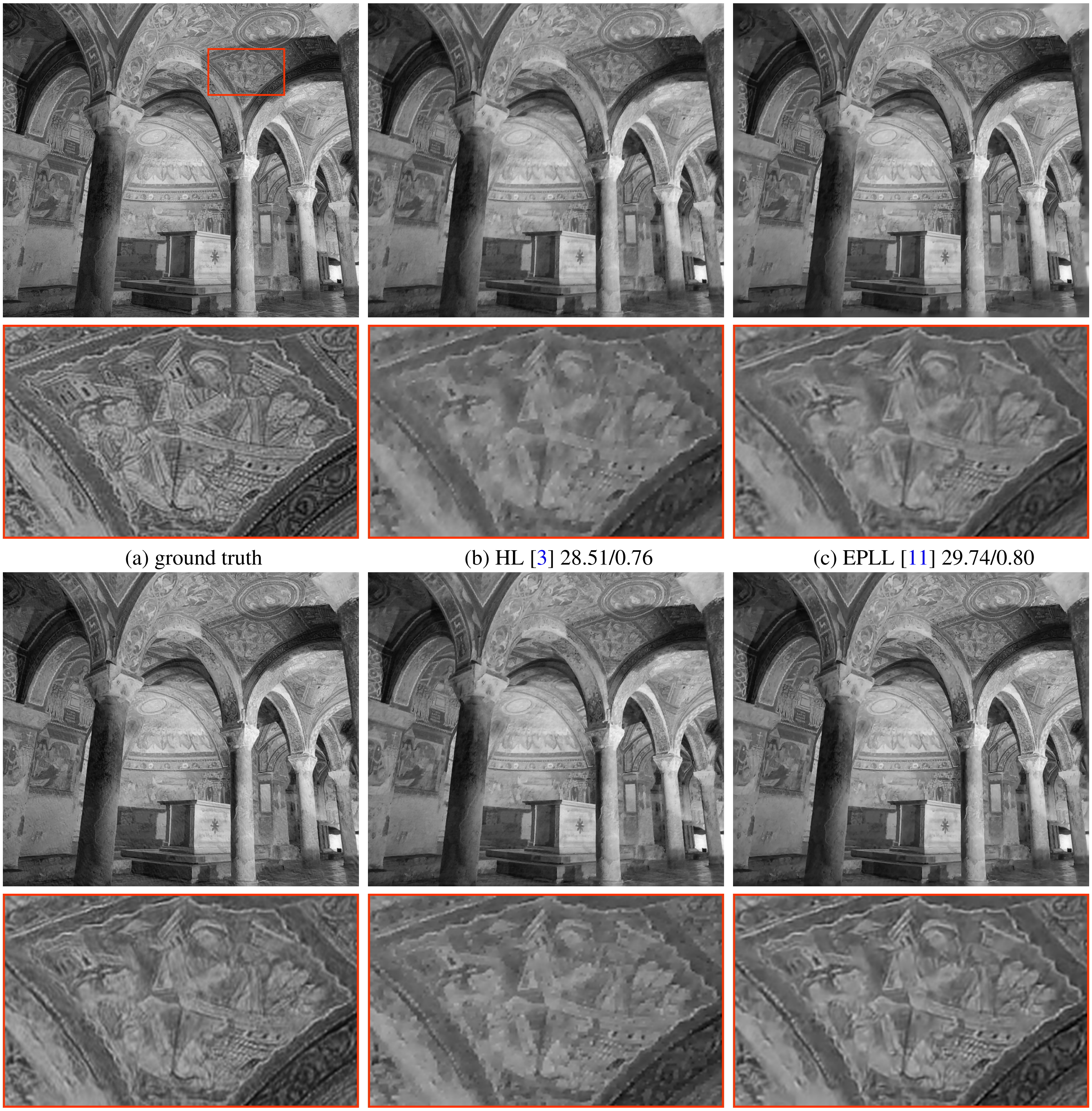
	HL [3]	EPLL [11]	MLP [7]	Ours
PSNR/SSIM	26.67/0.70	27.54/0.73	27.54/0.71	**27.74/0.74**

(a) HL [3] 29.31/0.85 (b) EPLL [11] 28.71/0.86 (c) MLP [7] 26.10/0.76

(d) CSF [6] 29.23/0.86 (e) Ours with GT 28.36/0.86 (f) Our with GT&E 29.62/0.86

Figure 2. Comparison of different non-blind deblurring methods with kernels estimated from Zhong [10] under 1% noise. PSNR and SSIM are shown in the figure. The results generated by our method have higher PSNR and SSIM values. Our network trained with ground truth and estimated kernels is able to remove noise and preserve structures and outperforms the network that is trained only ground truth kernels. “Ours with GT” means that the model is trained with ground truth blur kernels. “Our with GT&E” means that the model is trained with ground truth blur kernels and estimated blur kernels.

6. More Results of Synthetic Blurry Images with Ground Truth Kernels

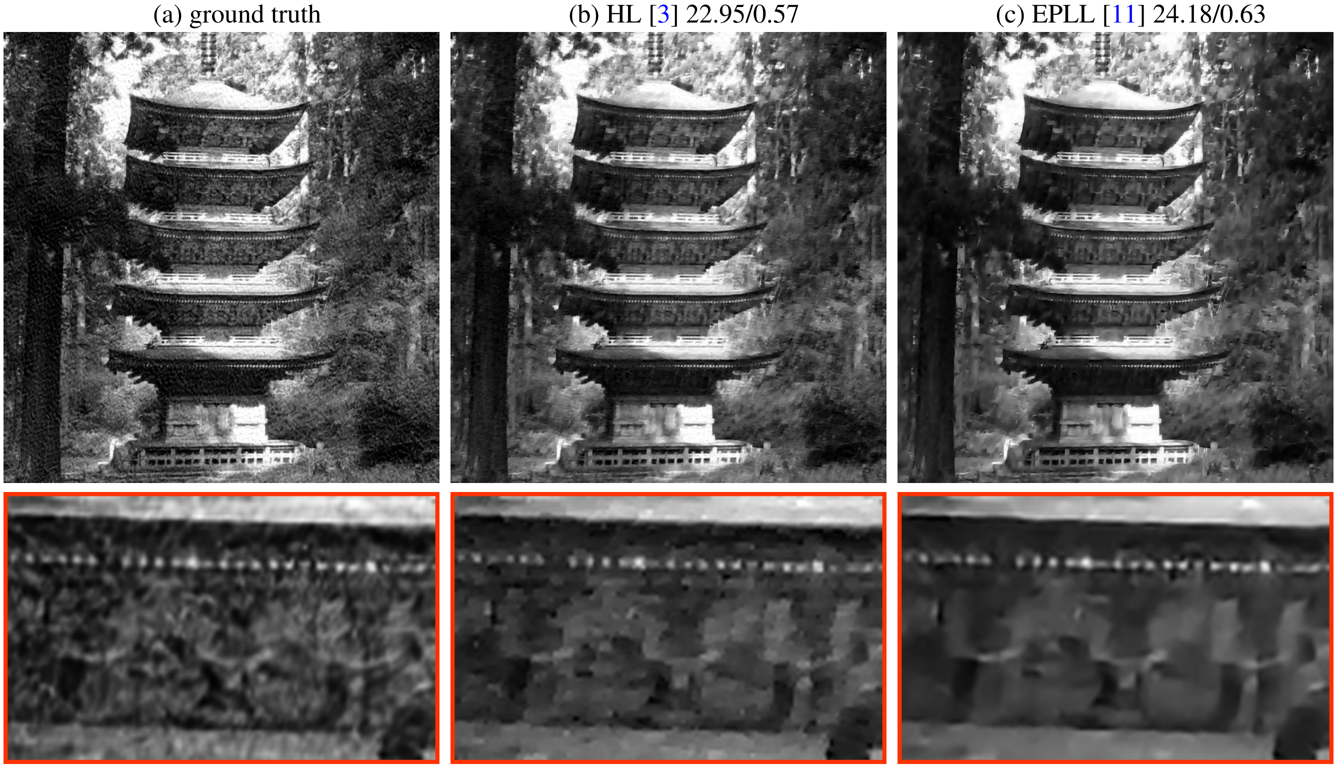


(a) ground truth (b) HL [3] 28.51/0.76 (c) EPLL [11] 29.74/0.80

(d) MLP [7] 29.75/0.79 (e) CSF [6] 28.94/0.78 (f) ours 29.77/0.81

Figure 3. Comparison of different non-blind deblurring methods with ground truth kernel under 1% noise. PSNR and SSIM are shown in the figure.

(a) ground truth (b) HL [3] 22.95/0.57 (c) EPLL [11] 24.18/0.63



(d) MLP [7] 24.07/0.63 (e) CSF [6] 24.71/0.68 (f) ours 25.04/0.68

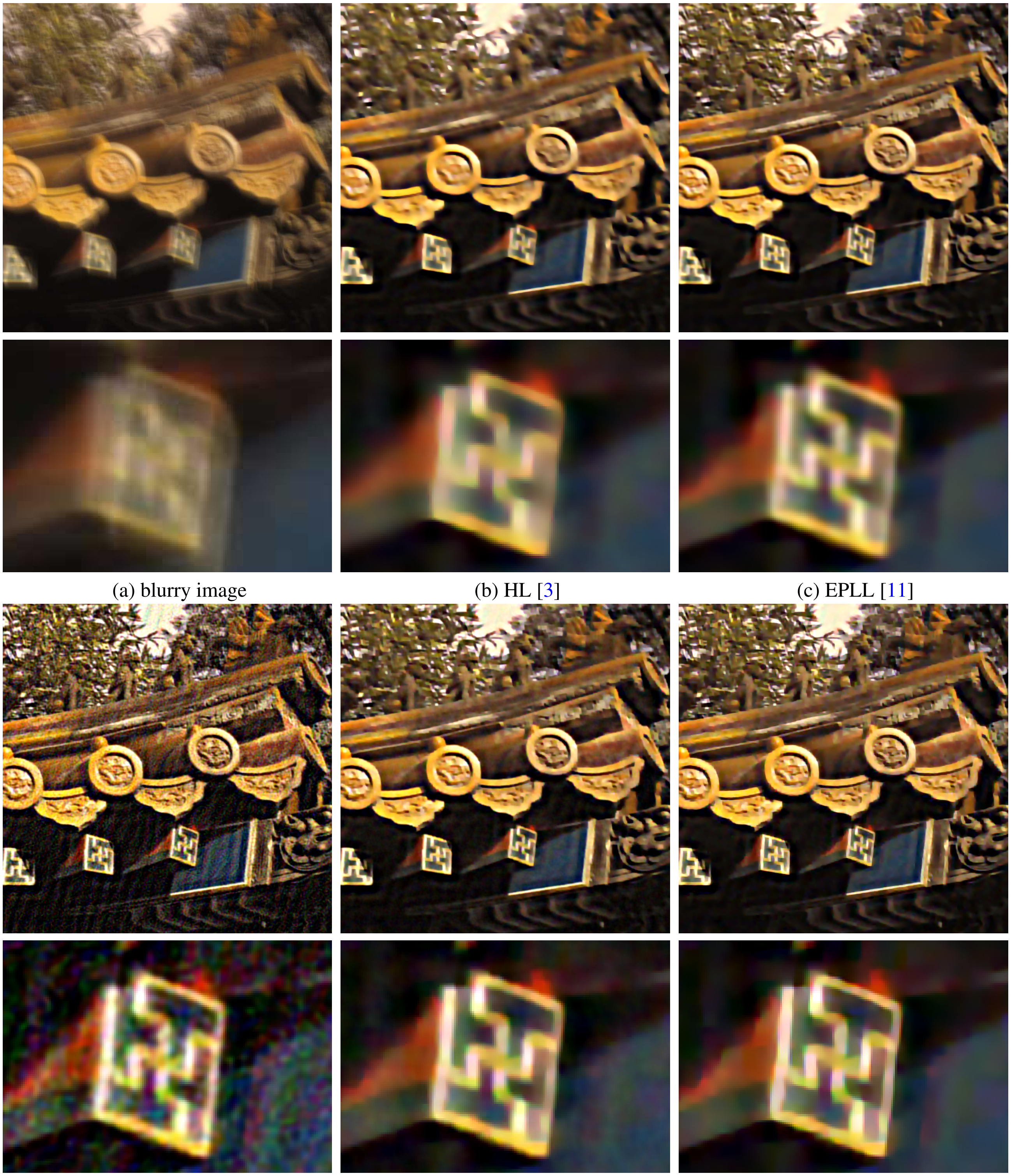
Figure 4. Comparison of different non-blind deblurring methods with ground truth kernel under 3% noise. PSNR and SSIM are shown in the figure.

(a) ground truth (b) HL [3] 25.25/0.67 (c) EPLL [11] 25.94/0.70

(d) MLP [7] 23.82/0.44 (e) CSF [6] 26.18/0.65 (f) ours 26.40/0.71

Figure 5. Comparison of different non-blind deblurring methods with ground truth kernel under 5% noise. PSNR and SSIM are shown in the figure.

7. More Results of Real Blurry Image



(a) blurry image (b) HL [3] (c) EPLL [11]

(d) MLP [7] (e) CSF [6] (f) ours

Figure 6. Comparison of different non-blind deblurring methods for real motion blurry image with 3% noise.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011. 3

[2] S. Cho and S. Lee. Fast motion deblurring. In *ACM TOG (Proc. SIGGRAPH Asia)*, 2009. 2, 3

[3] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 2009. 3, 4, 5, 6, 7, 8

[4] J. Pan, Z. Lin, Z. Su, and M.-H. Yang. Robust kernel estimation with outliers handling for image deblurring. In *CVPR*, 2016. 3

[5] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *CVPR*, 2016. 3

[6] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *CVPR*, 2014. 3, 4, 5, 6, 7, 8

[7] C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *CVPR*, 2013. 3, 4, 5, 6, 7, 8

[8] L. Sun and J. Hays. Super-resolution from internet-scale scene matching. In *ICCP*. 3

[9] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013. 3

[10] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *CVPR*, 2013. 2, 3, 4

[11] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011. 3, 4, 5, 6, 7, 8