

Learning to Push the Limits of Efficient FFT-based Image Deconvolution

– Supplemental Material –

Jakob Kruse¹ Carsten Rother¹ Uwe Schmidt²

¹ TU Dresden ² MPI of Molecular Cell Biology and Genetics, Dresden

1. Details about boundary adjustment comparison

Section 4.3 of the main paper compares our proposed boundary adjustment (BA) strategy (*Our BA*, cf. Eq. 17 and Fig. 2 of the main paper) to the traditional edgetapering method (*ET once*, cf. Eq. 11 of the main paper) and the BA approach (*ET each*) of CSF [3]; these BA strategies are compared within our FDN model, the CSF model, and a standard Wiener filter [5].

Specifically, we use the publicly available code to train different variants of the CSF model on a dataset of the same size as ours, and only adjust the BA strategy. Furthermore, we apply the Wiener filter as defined in Eq. 2 of the main paper, which we can use iteratively with our BA approach by replacing y with $\varphi_t(y, k, x^t)$; we estimate the expected image spectrum n from 3000 clean image patches.

While our BA comparison is depicted visually in Fig. 5 of the main paper, Table 1 also provides the numeric results and additionally includes stages 6 – 10 of our FDN model. As compared to the BA approach of CSF (*ET each*), the results suggest that CSF would also benefit from further stages if used with our BA strategy (cf. 6th column). Remarkably, the performance of the Wiener filter is not even fully saturated after 50 iterations (Wiener⁵⁰) when applied with our BA approach (cf. 3rd column, only every 5th step shown after iteration 10).

Fig. 1 shows an example where our proposed BA strategy yields a substantial improvement in image quality compared to standard edgetapering (*ET once*).

Stage	Wiener		CSF _{5×5} ⁵			FDN _G ¹⁰		
	ET once	Our BA	ET each	ET once	Our BA	ET each	ET once	Our BA
1	31.75	31.75	32.87	32.87	32.87	32.72	32.71	32.71
2		31.93	33.77	33.78	33.89	33.92	33.94	33.99
3		32.06	34.02	33.98	34.13	34.24	34.22	34.40
4		32.16	34.06	34.04	34.26	34.11	34.29	34.54
5		32.25	34.06	34.09	34.32	34.22	34.32	34.64
6		32.32				33.92	34.33	34.73
7		32.38				34.09	34.37	34.79
8		32.43				34.11	34.37	34.86
9		32.48				34.13	34.39	34.87
10		32.52				34.24	34.43	34.98
15		32.68						
20		32.79						
25		32.87						
30		32.93						
35		32.98						
40		33.02						
45		33.05						
50		33.08						

Table 1. Comparison of BA methods for FDN, CSF, and Wiener filter. Shown are average PSNR (dB) on the dataset of Levin *et al.* [1].

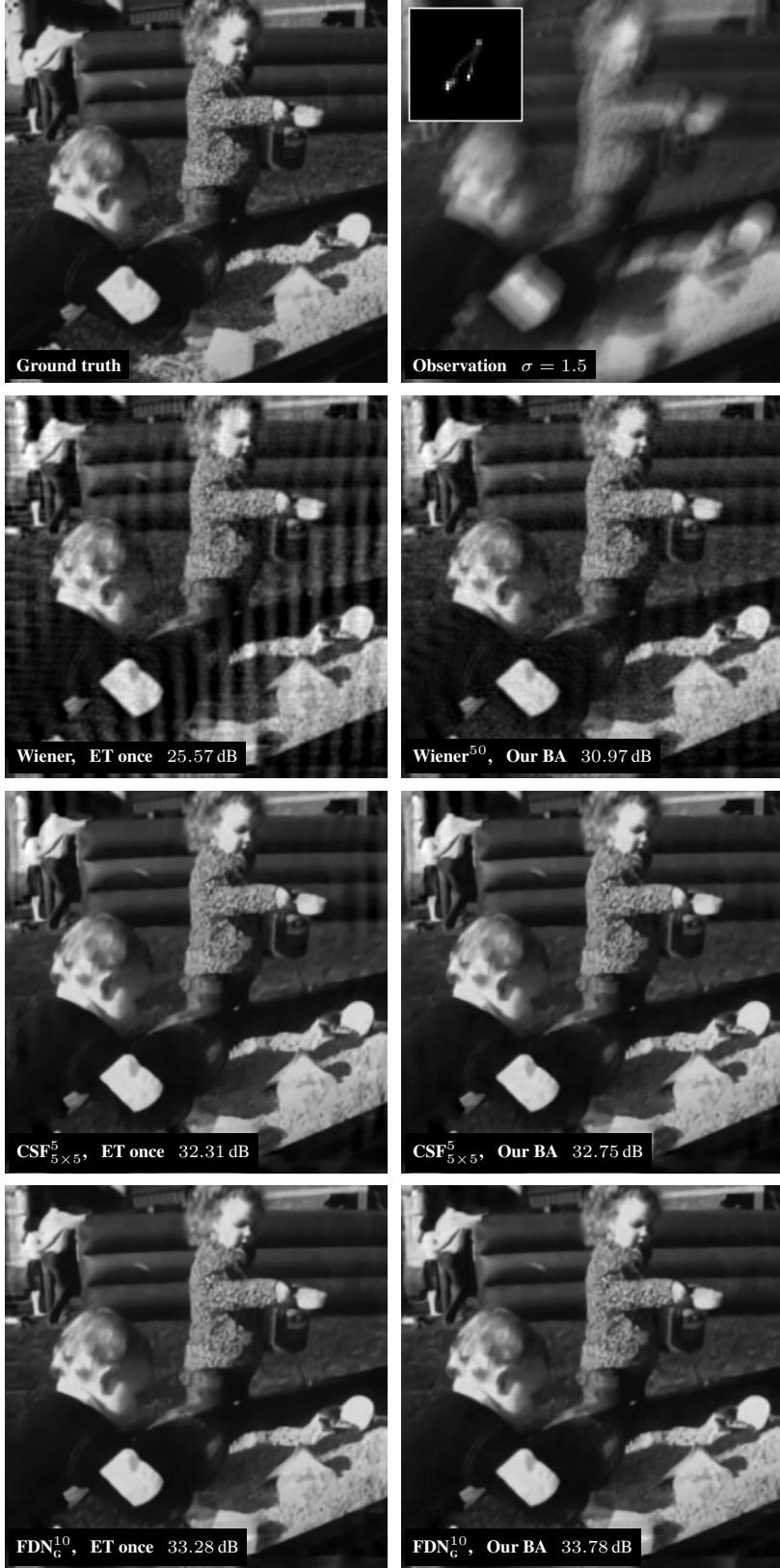


Figure 1. Qualitative comparison of our boundary adjustment strategy (*Our BA*) with common edgetapering (*ET once*) for three deconvolution methods (Wiener filter, CSF, and our FDN) on an example image from the dataset of Levin *et al.* [1]. See text for details.

2. Detailed results and additional models

We provide here the results of our greedily trained model FDN_G^{10} (for $\sigma \in [1.0, 3.0]$) after every of its ten stages, instead of just the final result after stage 10 as shown in Table 1 of the main paper.

Furthermore, we evaluate additional models for different (ranges of) noise levels: Following the common practice of most discriminative deconvolution methods, we train two noise-specialized models (for $\sigma = 1.5$ and $\sigma = 2.55$) to be used for the benchmarks of Levin *et al.* [1] and Sun *et al.* [4], respectively. Furthermore, we train a single model to be used for noise of up to 5% strength, *i.e.* using a much wider range of noise levels during training with $\sigma_{\text{train}} = [0.1, 12.75]$, which should cover most typical blurred photographs.

Detailed results of all our ten-stage models for both benchmarks are shown in Table 2. Fig. 2 additionally illustrates the greedy stage-wise performance for each of the two datasets. As expected, the noise-specialized models are on par or slightly better than our model from the main paper for $\sigma \in [1.0, 3.0]$. However, the difference is typically quite small. The much more versatile model for $\sigma \in [0.1, 12.75]$ has somewhat lower performance, but is still competitive with the state-of-the-art methods EPLL [6] and RTF [2] on both datasets (*cf.* Table 1 of the main paper). Nevertheless, future work should investigate how to increase the performance of noise-versatile models even further.

σ_{train}	DS	Model	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6	Stage 7	Stage 8	Stage 9	Stage 10
1.5	[1]	FDN_G^{10}	32.67	34.02	34.44	34.59	34.73	34.81	34.88	34.94	34.98	35.02
		FDN_T^{10}										35.18
2.55	[4]	FDN_G^{10}	29.78	31.88	32.31	32.41	32.54	32.57	32.62	32.63	32.63	32.62
		FDN_T^{10}										32.59
$[1.0, 3.0]$ (main paper)	[1]	FDN_G^{10}	32.71	33.99	34.40	34.54	34.64	34.73	34.79	34.86	34.87	34.98
		FDN_T^{10}										35.09
	[4]	FDN_G^{10}	29.63	31.69	32.22	32.30	32.43	32.45	32.54	32.55	32.60	32.62
		FDN_T^{10}										32.67
$[0.1, 12.75]$	[1]	FDN_G^{10}	32.59	33.46	34.01	34.24	34.44	34.55	34.69	34.75	34.82	34.84
		FDN_T^{10}										34.86
	[4]	FDN_G^{10}	29.93	31.59	32.15	32.20	32.36	32.32	32.30	32.27	32.31	32.23
		FDN_T^{10}										32.41

Table 2. Detailed results (PSNR in dB) for non-blind deblurring on the two benchmark datasets (DS) of Levin *et al.* [1] and Sun *et al.* [4]. FDN_G^{10} with greedy training and FDN_T^{10} with subsequent finetuning. At test time, used $\sigma = 1.5$ for Levin *et al.* and $\sigma = 2.55$ for Sun *et al.*.

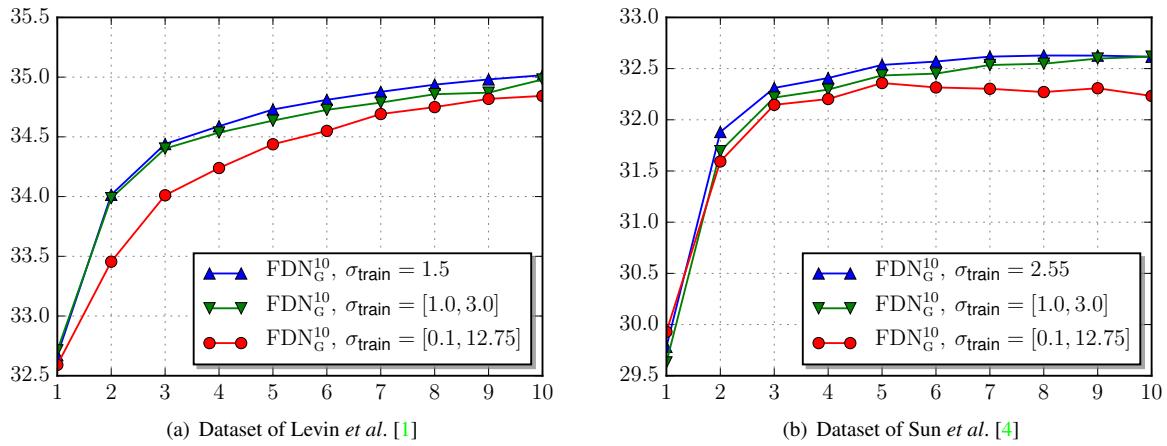


Figure 2. Greedy stage-wise performance (PSNR in dB) on two non-blind deblurring benchmarks for three models each (*cf.* Table 2).

3. Learned CNNs at different stages

Figure 6 (top row) of the main paper shows the output of the CNNs ϕ_t^{CNN} for the first five stages of a learned model. Intuitively, the CNN serves the purpose of modulating smoothness, which is sensitive to the location of edges in the sharp image. Since this is challenging to accomplish based on the observed blurred image, the CNN is less important at the first stage. However, the CNN can and will do a better job when the current estimate of the deblurred image improves after each stage. Nevertheless, Fig. 6 of the paper shows that the output of the CNNs will be very similar after only a few stages.

4. Additional example results

Additional results for example images from the dataset of Sun *et al.* [4] are shown in Figs. 3 to 6.

References

- [1] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, Florida, June 2009.
- [2] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother. Cascades of regression tree fields for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):677–689, Apr. 2016.
- [3] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, June 2014.
- [4] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *Proceedings of IEEE International Conference on Computational Photography*, 2013.
- [5] N. Wiener. *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. John Wiley & Sons, Inc., New York, N. Y., 1949.
- [6] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Proceedings of the Thirteenth IEEE International Conference on Computer Vision*, 2011.

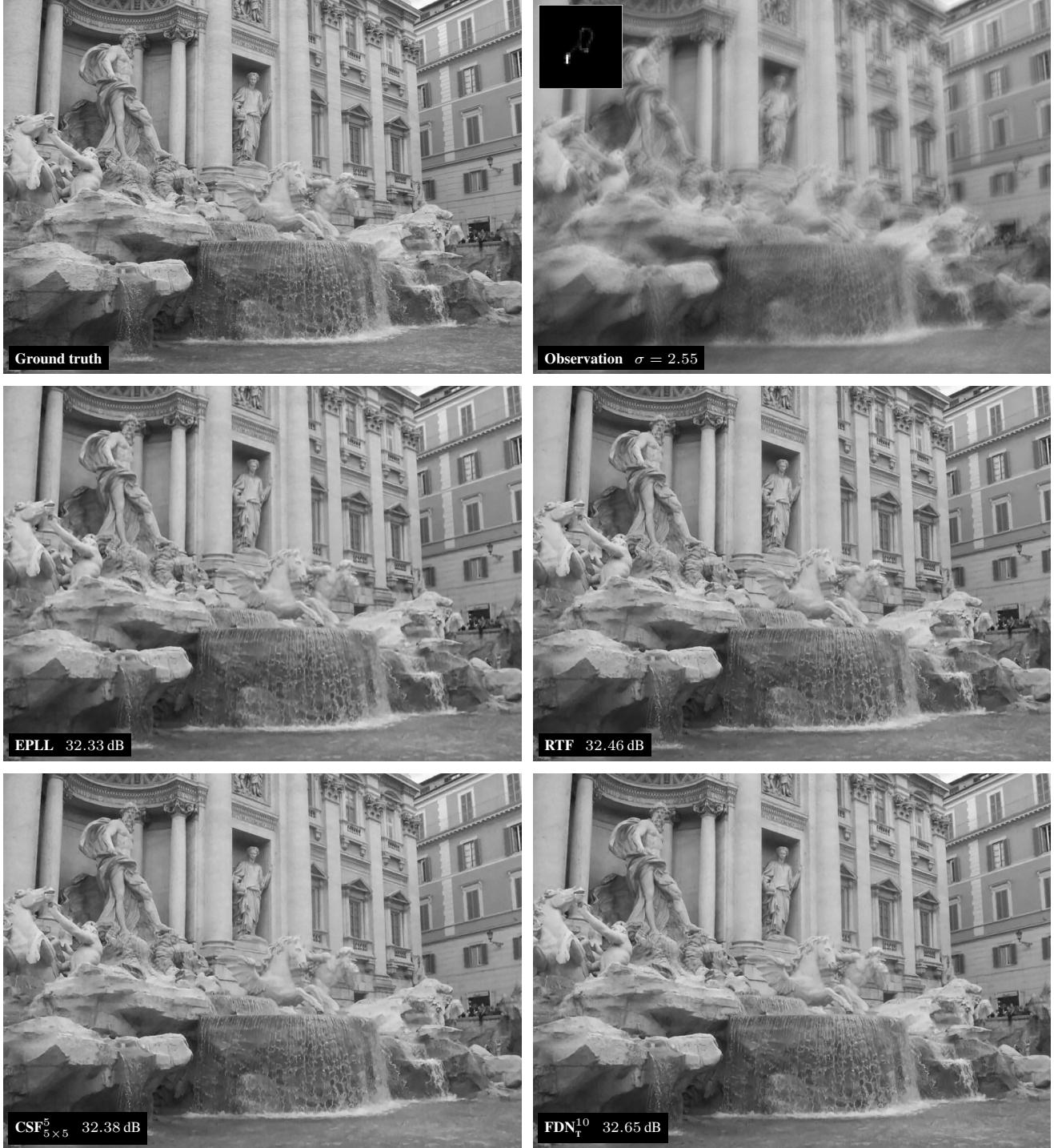


Figure 3. Deblurring example, comparing our $\text{FDN}_{\text{T}}^{10}$ model ($\sigma_{\text{train}} = [1.0, 3.0]$) with EPLL [6], RTF [2], and CSF [3].



Figure 4. Deblurring example, comparing our FDN_T^{10} model ($\sigma_{\text{train}} = [1.0, 3.0]$) with EPLL [6], RTF [2], and CSF [3].

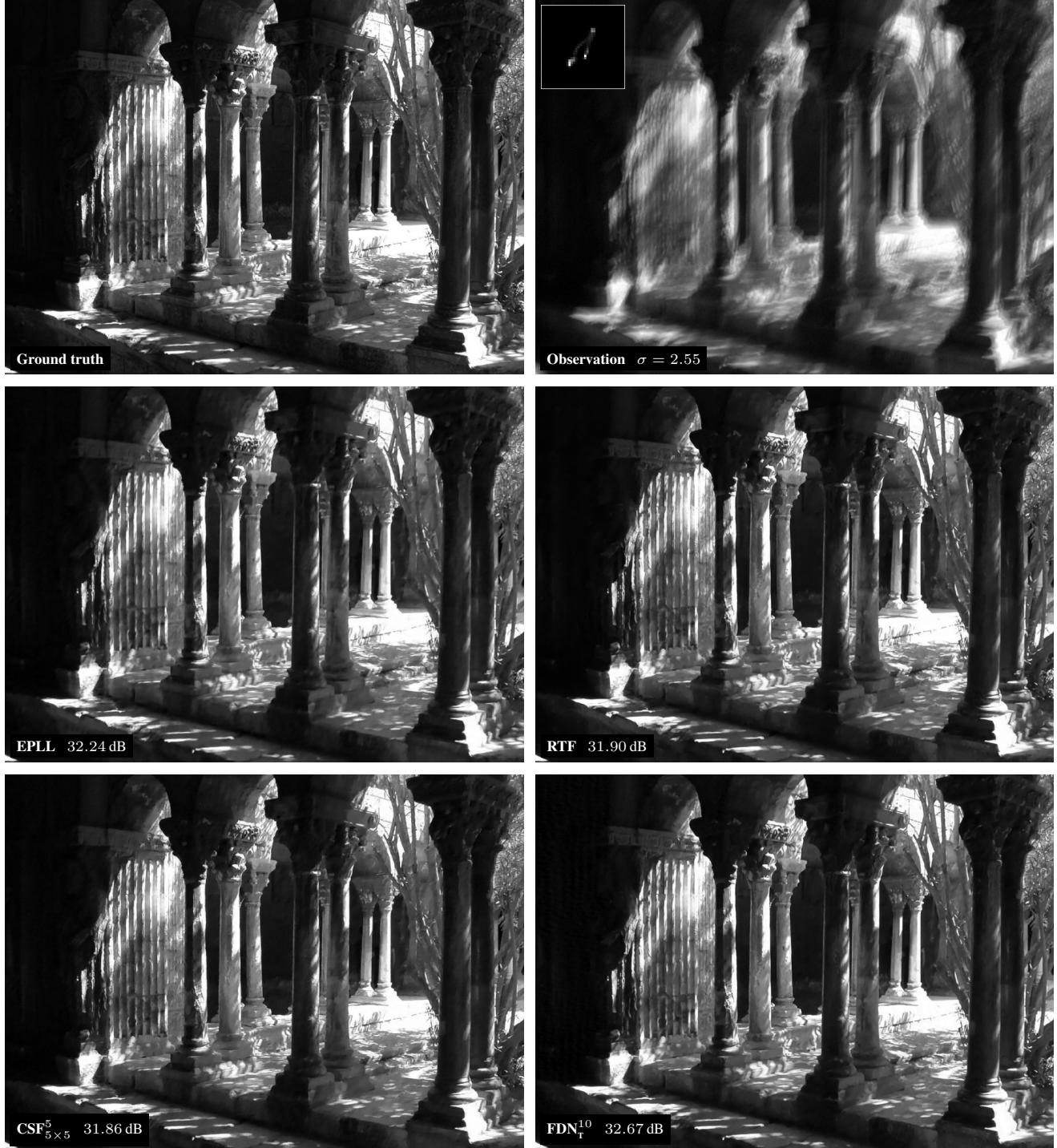


Figure 5. Deblurring example, comparing our FDN_T^{10} model ($\sigma_{train} = [1.0, 3.0]$) with EPLL [6], RTF [2], and CSF [3].

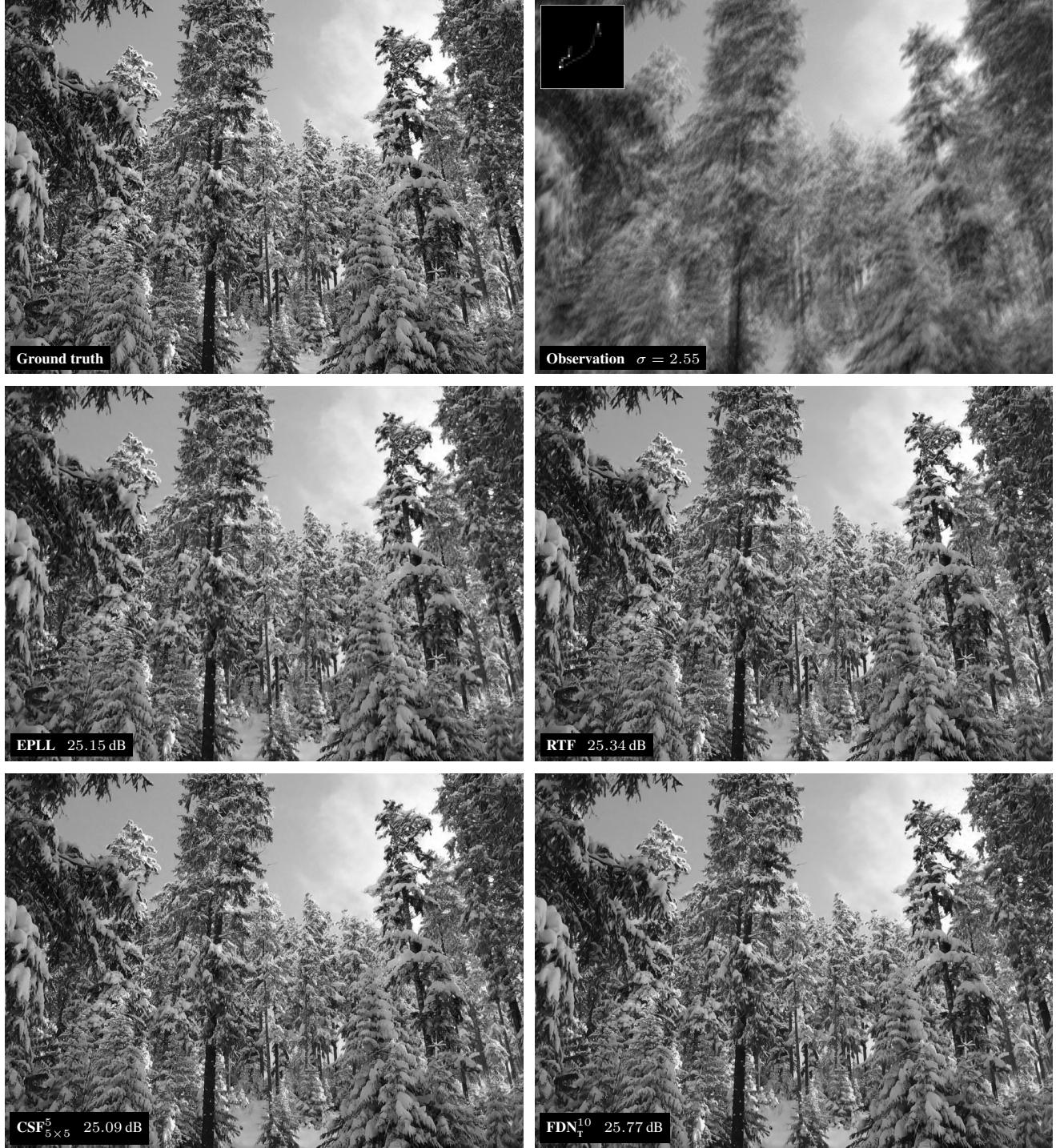


Figure 6. Deblurring example, comparing our FDN_T^{10} model ($\sigma_{train} = [1.0, 3.0]$) with EPLL [6], RTF [2], and CSF [3].