

Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems

Supplementary Material

Tim Meinhardt¹

tim.meinhardt@in.tum.de

Michael Moeller²

michael.moeller@uni-siegen.de

Caner Hazirbas¹

hazirbas@cs.tum.edu

Daniel Cremers¹

cremers@tum.de

Technical University of Munich¹

University of Siegen²

Abstract

The supplementary material contains the proof of Remark 3.1 as well as some additional information about the numerical experiments that contribute to the understanding of the main paper. We present detailed qualitative and quantitative evaluation results for each of our two (demosaicking and deconvolution) exemplary linear inverse image reconstruction problems. These results include parameter values obtained with our grid search, reconstruction PSNR values and images.

Proof of Remark 3.1

For the sake of readability let us restate the remark and the four algorithms with the proximal operators of the regularization R replaced by an arbitrary continuous function \mathcal{G} .

PG

$$u^{k+1} = \mathcal{G}(u^k - \tau A^* \nabla H_f(Au^k)). \quad (1)$$

ADMM

$$u^{k+1} = \text{prox}_{\frac{1}{\gamma}(H_f \circ A)} \left(v^{k+1} - \frac{1}{\gamma} y^k \right), \quad (2)$$

$$v^{k+1} = \mathcal{G} \left(u^k + \frac{1}{\gamma} y^k \right), \quad (3)$$

$$y^{k+1} = y^k + \gamma(u^{k+1} - v^{k+1}), \quad (4)$$

PDHG1

$$z^{k+1} = z^k + \gamma A \bar{u}^k - \gamma \text{prox}_{\frac{1}{\gamma} H_f} \left(\frac{1}{\gamma} z^k + A \bar{u}^k \right), \quad (5)$$

$$y^{k+1} = y^k + \gamma \bar{u}^k - \gamma \mathcal{G} \left(\frac{1}{\gamma} y^k + \bar{u}^k \right), \quad (6)$$

$$u^{k+1} = u^k - \tau A^T z^{k+1} - \tau y^{k+1}, \quad (7)$$

$$\bar{u}^{k+1} = u^{k+1} + \theta(u^{k+1} - u^k), \quad (8)$$

PDHG2

$$y^{k+1} = y^k + \gamma \bar{u}^k - \gamma \mathcal{G} \left(\frac{1}{\gamma} y^k + \bar{u}^k \right), \quad (9)$$

$$u^{k+1} = \text{prox}_{\tau(H_f \circ A)}(u^k - \tau y^{k+1}), \quad (10)$$

$$\bar{u}^{k+1} = u^{k+1} + \theta(u^{k+1} - u^k). \quad (11)$$

Remark (Remark 3.1 in main Paper). Consider replacing the proximal operator of R in the PG, ADMM, PDHG1, and PDHG2 methods by an arbitrary continuous function \mathcal{G} . Then the fixed-point equations of all four resulting algorithmic schemes are equivalent, and yield

$$u_* = \mathcal{G}(u_* - t A^T \nabla H_f(Au_*)) \quad (12)$$

with $* \in \{PG, ADMM, PDHG1, PDHG2\}$ and $t = \tau$ for PG and PDHG2, and $t = \frac{1}{\gamma}$ for ADMM and PDHG1.

Proof. For the PG-based algorithmic scheme the statement follows immediately as (12) coincides with the update equation (1).

At fixed-points of the ADMM-based scheme, it follows from Equation (4) that $u_{ADMM} = v$. The optimality condition for Equation (2) therefore becomes $y = -A^T \nabla H_f(Au_{ADMM})$, such that Equation (3) shows the fixed-point Equation (12) for the ADMM-based scheme. Vice versa, for any given element u^0 meeting Equation (12) one initializes $y^0 = -A^T \nabla H_f(Au^0)$, and $v^0 = u^0$ to obtain a fixed-point of the ADMM-based scheme.

At fixed-points of the PDHG1-based scheme (again variables without superscripts denoting the fixed-point), it follows from Equation (7) that $y = -A^T z$. The optimality condition for Equation (5) yields

$$0 = Au - \frac{1}{\gamma} z - Au + \frac{1}{\gamma} \nabla H_f(Au), \quad (13)$$

$$\Rightarrow z = \nabla H_f(Au), \quad (14)$$

and inserting the resulting identity $y = -A^T \nabla H_f(Au)$ into Equation (6) shows that any fixed-point of the PDHG1-based scheme meets Equation (12). For a given fixed-point u^0 meeting Equation (12) the choices $\bar{u}^0 = u^0$, $z^0 = \nabla H_f(Au^0)$, $y^0 = -A^T \nabla H_f(Au^0)$ yield a fixed-point of the PDHG1-based algorithmic scheme.

Finally, for the PDHG2-based scheme Equation (10) yields $y = -A^T \nabla H_f(Au)$, such that Equation (10) yields the fixed-point Equation (12). Again, initializing $\bar{u}^0 = u^0$ with the fixed-point and setting $y^0 = -A^T \nabla H_f(Au^0)$ results in a fixed-point of the PDHG2-based scheme and therefore yields the assertion. \square

Remark. We would like to point out that the PDHG2 algorithm is closely related to ADMM: In fact, with an over-relaxation on the variable y , a reversed update order of u and y , and $\tau = \frac{1}{\gamma}$, $\theta = 1$, it is equivalent to the above ADMM algorithm in the convex case with proximity operators, see e.g. [1], Section 5.3. Interestingly, one can show that this result still remains valid for our algorithmic schemes above in which the proximity operator has been replaced by a neural network.

Evaluation

Demosaicking

We evaluated the effectiveness of our approach on noise free demosaicking of 18 Bayer filtered images of the *McMaster* color image dataset, [5]. For visualization purposes Figure 1 presents demosaicking results obtained with our approach applying the fixed denoising network trained on noise with standard deviation $\sigma = 0.02$. The images include a magnified area of the residual error which illustrates the varying demosaicking performance on differently structured parts of the image. In completion of Figure 4 of the main paper Table 1 contains a comprehensive list of channel-wise PSNR values for each of the 18 color images. The superior reconstruction of the green channel can be attributed to its dominance in the *RGGB* filter pattern. For a full comparison of our results with the state-of-the-art methods mentioned in the main paper we refer to the supplementary material of [3] and [2].

Deconvolution

Our experimental setup consists of the five (*a - e*) deconvolution experiments proposed in [4]. These experiments corrupt 11 standard test images with different blur kernels and Gaussian noise levels. Figure 2 shows the corresponding dataset as well as exemplary deconvolution results obtained by our approach using the fixed network trained on noise with standard deviation $\sigma = 0.02$. The corresponding PSNR values as well as our FlexISP* results are presented in Table 3. A detail explanation of FlexISP*, our reimplementation of [3], can be found in the main paper. To illustrate the robustness with respect to the choice of network we also included the results for networks trained on different σ . For a comprehensive comparison with the methods mentioned in the paper we again refer to the supplementary material of [3]. For the sake of reproducibility Table 2 includes the results of our grid search for the data fidelity parameter α as well as for the regularization parameter β_{TV} for multiple networks.

Table 1: Channel-wise PSNRs in [dB] for each Bayer filtered image of the *McMaster* color image dataset. Our method applies the fixed denoising network trained on $\sigma = 0.02$.

Image	Channel	Reconstruction PSNR in [dB]	
		FlexISP*	Ours
1	R	28.52	29.09
	G	31.55	32.04
	B	26.71	27.01
2	R	33.86	34.69
	G	38.39	39.30
	B	32.18	32.85
3	R	32.31	34.33
	G	35.56	36.83
	B	29.80	30.81
4	R	35.77	38.55
	G	39.90	41.08
	B	32.92	34.47
5	R	34.68	35.31
	G	37.30	37.71
	B	30.67	31.65
6	R	37.12	39.38
	G	41.69	43.09
	B	34.40	36.44
7	R	35.35	35.89
	G	38.31	38.62
	B	33.55	33.85
8	R	35.95	38.42
	G	40.35	41.80
	B	35.56	37.18
9	R	34.76	36.78
	G	40.74	41.81
	B	35.78	36.86
10	R	37.31	37.57
	G	41.61	41.54
	B	36.62	36.90
11	R	38.71	39.92
	G	41.23	42.19
	B	37.90	38.54
12	R	37.96	38.46
	G	40.52	41.60
	B	35.56	37.22
13	R	40.49	42.46
	G	44.74	45.46
	B	37.84	38.68
14	R	38.07	39.13
	G	42.65	43.06
	B	35.88	36.25
15	R	36.77	37.26
	G	42.34	42.58
	B	38.42	38.90
16	R	32.48	34.16
	G	34.05	35.19
	B	32.61	32.65
17	R	31.84	33.37
	G	36.57	37.40
	B	31.77	32.30
18	R	32.78	34.02
	G	36.15	36.92
	B	34.17	35.09
AVG	R	35.26	36.60
	G	39.09	39.90
	B	34.02	34.87
AVG	RGB	36.12	37.12

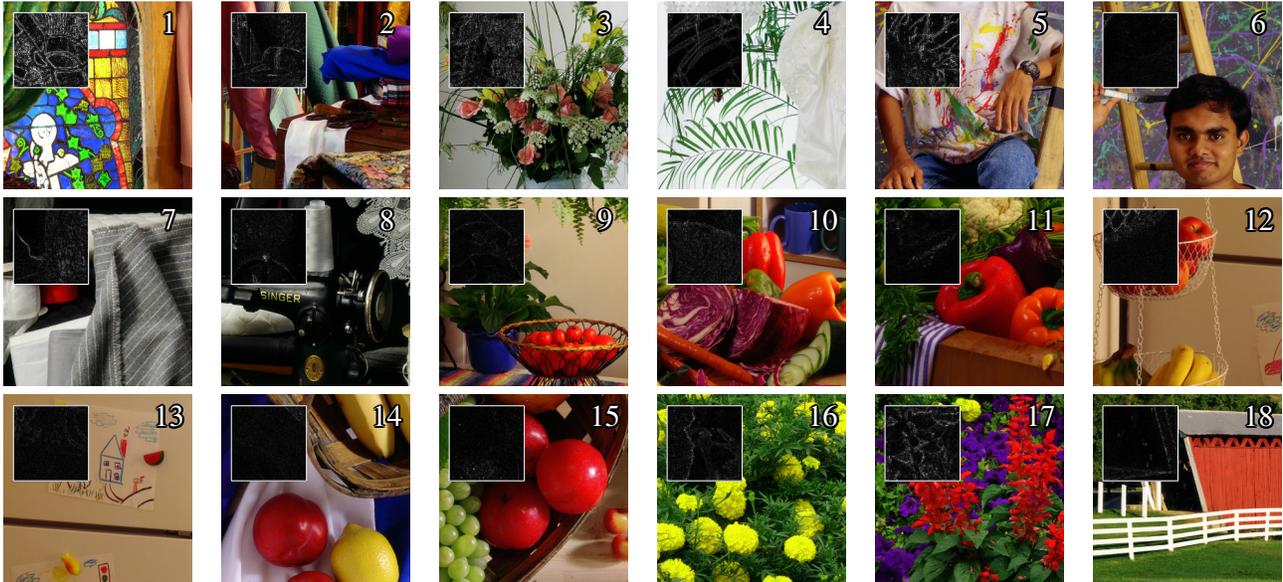


Figure 1: We demosaicked 18 images of the *McMaster* color image dataset applying our approach with the fixed denoising network. To illustrate the remaining reconstruction error we added magnified residual images. To avoid boundary effects the images were cropped by 5 pixels.

Table 2: The optimal deblurring values for the data fidelity parameter α as well as for the regularization parameter β_{TV} with respect to our method applying denoising networks trained on different noise standard deviations σ . All values were obtained by performing an extensive grid search of the parameter space. Following Proposition 3.2 we set the dual step size of the PDHG algorithm to $\gamma = 1.0$ and determined the primal step size τ from $\tau\gamma < c$ with c being the squared norm of the involved linear operator.

σ	Experiment <i>a</i>		Experiment <i>b</i>		Experiment <i>c</i>		Experiment <i>d</i>		Experiment <i>e</i>	
	α	β_{TV}								
0.01	1	0.00	25	0.00	40	0.05	250	0.01	10	0.00
0.02	2	0.00	75	0.00	4	0.00	73	0.00	23	0.00
0.03	5	0.00	149	0.00	7	0.00	107	0.00	43	0.00
0.04	7	0.00	200	0.00	10	0.00	140	0.00	64	0.00
0.05	11	0.01	160	0.01	13	0.00	200	0.00	93	0.00
0.06	13	0.00	200	0.01	17	0.00	240	0.00	120	0.00
0.07	16	0.00	424	0.00	24	0.00	272	0.00	150	0.00
0.08	23	0.00	467	0.00	34	0.00	467	0.00	200	0.00
0.09	24	0.00	300	0.01	36	0.00	600	0.00	267	0.00
0.20	100	0.00	800	0.03	150	0.00	2400	0.00	480	0.10



Figure 2: Our deconvolution dataset based on the experiments introduced in [4]. Each image (128×128 pixels) is shown in its corrupted as well as by our approach reconstructed version. The deblurring was performed using the fixed denoising network trained on $\sigma = 0.02$. To avoid boundary effects the images were cropped by 12 pixels. For visualization purposes we show enlarged versions of the different blur kernels.

Table 3: Imagewise PSNRs in [dB] for each of our 5 (*a - e*) deconvolution experiments for FlexISP* and multiple versions of our approach using denoising networks trained on different σ . Our application independent approach applied a network trained on $\sigma = 0.02$.

Method		Reconstruction PSNR in [dB]										
		<i>Barbara</i>	<i>Boat</i>	<i>Cameraman</i>	<i>Couple</i>	<i>Fingerprint</i>	<i>Hill</i>	<i>House</i>	<i>Lena</i>	<i>Man</i>	<i>Montage</i>	<i>Peppers</i>
Experiment <i>a</i>	FlexISP* [3]	25.93	24.44	23.65	24.16	17.43	25.83	26.93	25.05	24.90	22.84	26.41
	Ours	26.27	24.41	23.78	24.15	17.41	25.89	27.35	25.34	25.02	23.00	26.99
	Ours, $\sigma=0.01$	25.97	24.34	23.40	24.13	17.41	25.78	26.53	24.95	24.88	22.89	26.49
	Ours, $\sigma=0.04$	26.19	24.48	23.93	24.26	17.43	25.95	27.38	25.42	25.12	22.97	27.06
	Ours, $\sigma=0.06$	26.32	24.46	23.97	24.27	17.44	25.98	27.56	25.51	25.13	23.02	27.12
	Ours, $\sigma=0.20$	26.27	24.42	23.99	24.27	17.44	26.03	27.03	25.60	25.17	23.06	27.04
Experiment <i>b</i>	FlexISP* [3]	29.14	26.62	26.00	26.55	17.81	28.70	30.99	27.90	27.38	24.47	29.72
	Ours	29.38	26.74	26.26	26.70	17.86	28.81	31.43	28.27	27.58	24.70	30.13
	Ours, $\sigma=0.01$	29.36	26.66	26.05	26.64	17.82	28.87	31.24	28.17	27.60	24.55	30.19
	Ours, $\sigma=0.04$	29.40	26.70	26.28	26.71	17.85	28.82	31.52	28.40	27.64	24.66	30.14
	Ours, $\sigma=0.06$	29.52	26.79	26.37	26.74	17.83	28.91	31.39	28.37	27.74	24.62	30.27
	Ours, $\sigma=0.20$	29.49	26.77	26.32	26.70	17.84	28.86	31.60	28.39	27.72	24.51	30.24
Experiment <i>c</i>	FlexISP* [3]	23.24	22.11	21.01	22.04	17.04	23.05	23.57	22.57	22.43	21.38	23.47
	Ours	23.12	22.01	20.85	21.93	17.02	23.12	22.77	22.43	22.49	21.22	23.19
	Ours, $\sigma=0.01$	22.49	21.77	20.96	21.75	17.07	22.83	22.64	22.02	22.27	20.91	22.51
	Ours, $\sigma=0.04$	23.03	22.18	21.20	21.89	17.03	23.12	23.26	22.64	22.41	21.43	23.29
	Ours, $\sigma=0.06$	23.02	22.23	21.27	21.98	17.06	23.18	23.62	22.51	22.49	21.40	23.56
	Ours, $\sigma=0.20$	23.15	22.20	21.19	21.93	17.09	23.12	23.50	22.28	22.53	21.33	23.45
Experiment <i>d</i>	FlexISP* [3]	23.13	22.92	21.92	22.87	17.44	23.88	24.95	22.57	22.33	22.19	23.59
	Ours	22.48	22.45	20.89	22.69	17.38	23.53	23.37	22.22	21.97	21.64	22.90
	Ours, $\sigma=0.01$	21.81	22.08	20.71	22.40	17.25	22.98	23.01	21.52	21.62	21.30	22.03
	Ours, $\sigma=0.04$	22.97	22.66	21.78	22.77	17.37	23.78	24.91	22.51	22.23	22.07	23.33
	Ours, $\sigma=0.06$	23.21	22.71	21.83	22.81	17.39	23.87	25.57	22.71	22.39	22.19	23.70
	Ours, $\sigma=0.20$	23.19	22.76	21.85	22.81	17.37	23.87	25.48	22.63	22.39	22.64	23.66
Experiment <i>e</i>	FlexISP* [3]	30.60	28.54	29.19	28.27	23.59	29.31	32.65	29.93	28.49	30.63	31.13
	Ours	31.67	29.24	30.84	28.85	23.42	29.69	33.38	30.80	29.15	32.45	32.36
	Ours, $\sigma=0.01$	29.86	28.69	29.14	28.02	22.19	29.28	31.28	29.36	28.35	29.70	30.65
	Ours, $\sigma=0.04$	31.75	29.51	30.99	28.88	24.20	29.80	33.65	30.93	29.37	32.49	32.28
	Ours, $\sigma=0.06$	31.73	29.48	30.80	28.85	24.14	29.75	33.37	30.91	29.40	32.22	32.21
	Ours, $\sigma=0.20$	31.42	29.28	30.50	28.78	23.80	29.57	33.06	30.73	29.24	31.29	31.94

References

- [1] A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25, 2016. 2
- [2] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 2016. 3
- [3] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajk, D. Reddy, O. Gallo, J. L. abd Wolfgang Heidrich, K. Egiazarian, J. Kautz, and K. Pulli. Flexisp: A flexible camera image processing framework. *ACM Special Interest Group on Computer Graphics (SIGGRAPH)*, 2014. 3, 6
- [4] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3, 5
- [5] L. Zhang, X. Wu, A. Buades, and X. Li. Color Demosaicking by Local Directional Interpolation and Nonlocal Adaptive Thresholding. *Journal of Electronic Imaging*, 2011. 3