Benchmarking Denoising Algorithms with Real Photographs
– Supplemental Material –

Tobias Plötz Stefan Roth
Department of Computer Science, TU Darmstadt

Preface. In this supplemental material we give a proof for $\mathcal{A}(y_n)$ and $R(y_n)$ being linearly uncorrelated. We, furthermore, give additional details on our novel heteroscedastic Tobit regression model (derivation, log-likelihood and its gradient) and highlight the importance of considering clipping of the noisy observations. Finally, we show additional results from our denoising benchmark.

A. Linear Correlation of $\mathcal{A}(y_n)$ and $R(y_n)$

Proposition 1. *The debiased image $\mathcal{A}(y_n)$ and the debiased residual image $R(y_n)$ are linearly uncorrelated.*

Proof. First, we note that the expectation of $R(y_n)$ given $\mathcal{A}(y_n)$ is zero

$$\begin{aligned}
&\mathbb{E}\left[R(y_n) \mid \mathcal{A}(y_n)\right] \\
&\quad = \mathbb{E}\left[\mathcal{A}(y_n) - x_n \mid \mathcal{A}(y_n)\right] && \text{(15a)}\\
&\quad = \mathbb{E}\left[\mathcal{A}(y_n) \mid \mathcal{A}(y_n)\right] - \mathbb{E}\left[x_n \mid \mathcal{A}(y_n)\right] && \text{(15b)}\\
&\quad = \mathcal{A}(y_n) - \mathbb{E}\left[x_n \mid y_n\right] && \text{(15c)}\\
&\quad = 0, && \text{(15d)}
\end{aligned}$$

where the third equality follows from the fact that $\mathcal{A}(\cdot)$ is invertible [11]. Next, we observe that for two random variables X and Y, the expectation of X is zero if $\mathbb{E}\left[X \mid y = Y\right] = 0$ for all y:

$$\mathbb{E}\left[X\right] = \mathbb{E}_Y\left[\mathbb{E}_{X|Y}\left[X\right]\right] = \mathbb{E}_Y\left[0\right] = 0. \tag{16}$$

We now show that two random variables X and Y have zero covariance if $\mathbb{E}\left[X \mid y = Y\right] = 0$ for all y:

$$\begin{aligned}
\mathrm{Cov}(X,Y) &= \mathbb{E}\big[(X - \mathbb{E}X)(Y - \mathbb{E}Y)\big] && \text{(17)}\\
&= \mathbb{E}\big[X(Y - \mathbb{E}Y)\big] && \text{(18)}\\
&= \mathbb{E}\left[XY\right] - \mathbb{E}X \cdot \mathbb{E}Y && \text{(19)}\\
&= \mathbb{E}_Y\left[\mathbb{E}_{X|Y}\left[XY\right]\right] && \text{(20)}\\
&= \mathbb{E}_Y\left[Y \cdot \mathbb{E}_{X|Y}\left[X\right]\right] && \text{(21)}\\
&= \mathbb{E}_Y\left[Y \cdot 0\right] && \text{(22)}\\
&= 0. && \text{(23)}
\end{aligned}$$

From the definition of the linear correlation coefficient it follows that zero covariance between two random variables implies that they are linearly uncorrelated. □

B. Heteroscedastic Tobit Regression

We now derive the log-likelihood and its gradient of the proposed heteroscedastic Tobit regression model (Eqs. 8–9b in the paper). Moreover, we detail the approximation of the noise term of Eqs. (7a) – (7b) of the main paper. For clarity, we denote $\alpha(x_r) = \boldsymbol{\alpha}^\top \mathbf{x} \doteq \tilde{x}$, where $\mathbf{x} = [x_r, 1]^\top$.

B.1. Log-likelihood

Before deriving the log-likelihood of Eq. (8), let us first look at the theoretical case of unclipped intensities x'_n in the high-ISO image:

$$x'_n = \tilde{x} + \epsilon_{r,n}(\tilde{x}). \tag{24}$$

Following Eq. (9a) of the main paper, the conditional distribution of x'_n given the intensities in $\tilde{x}$ is given as a heteroscedastic Gaussian:

$$p(x'_n \mid x_r) = \mathcal{N}\big(x'_n \mid \tilde{x},\ \sigma_{r,n}(\tilde{x})\big). \tag{25}$$

We now consider the clipped noisy signal x_n. To derive its conditional distribution in case that x_n is clipped, we replace the Gaussian PDF with Dirac deltas weighted by the probability mass of all possible values x'_n that would be clipped to x_n. Hence, the conditional distribution is given by a case distinction on whether x_n is unclipped, clipped from below, or from above, respectively. Precisely, we can write

$$\mathcal{T}(x_n \mid x_r) = \begin{cases} \mathcal{N}\big(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big), & \text{if } 0 < x_n < 1 \\ \delta(x_n) \cdot \int_{-\infty}^{0} \mathcal{N}\big(x'_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\, \mathrm{d}x'_n, & \text{if } x_n \leq 0 \\ \delta(1 - x_n) \cdot \int_{1}^{\infty} \mathcal{N}\big(x'_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\, \mathrm{d}x'_n, & \text{if } x_n \geq 1. \end{cases} \tag{26}$$

It is easy to check that $\mathcal{T}(x_n \mid x_r)$ indeed is a valid probability distribution. Obviously, $\mathcal{T}(x_n \mid x_r) \geq 0$ and

$$\int_{\mathbb{R}} \mathcal{T}(x_n \mid x_r)\,\mathrm{d}x_n = \int_{-\infty}^{0} \mathcal{T}(x_n \mid x_r)\,\mathrm{d}x_n \tag{27a}$$
$$+ \int_{0}^{1} \mathcal{T}(x_n \mid x_r)\,\mathrm{d}x_n + \int_{1}^{\infty} \mathcal{T}(x_n \mid x_r)\,\mathrm{d}x_n$$
$$= \int_{-\infty}^{0} \mathcal{N}\big(x'_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\,\mathrm{d}x'_n \tag{27b}$$
$$+ \int_{0}^{1} \mathcal{N}\big(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\,\mathrm{d}x_n$$
$$+ \int_{1}^{\infty} \mathcal{N}\big(x'_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\,\mathrm{d}x'_n$$
$$= 1. \tag{27c}$$

By denoting the cumulative distribution function of a standard normal distribution as $\Psi(z) = \int_{-\infty}^{z} \mathcal{N}(z' \mid 0, 1)\,\mathrm{d}z'$ and by noting that $\Psi\big(\frac{z-\mu}{\sigma}\big) = \int_{-\infty}^{z} \mathcal{N}(z' \mid \mu, \sigma)\,\mathrm{d}z'$, we can write the log-likelihood of $\mathcal{T}(x_n \mid x_r)$ up to constants as

$$\log \mathcal{T}(x_n \mid x_r) = \begin{cases} -\log \sigma_{r,n}(\tilde{x}) - \frac{(x_n - \tilde{x})^2}{2\sigma_{r,n}(\tilde{x})^2}, \\ \qquad\qquad \text{if } 0 < x_n < 1 \\ \delta(x_n) \cdot \log \Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right), \\ \qquad\qquad \text{if } x_n \leq 0 \\ \delta(1 - x_n) \cdot \log\left(1 - \Psi\left(\frac{1-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)\right), \\ \qquad\qquad \text{if } x_n \geq 1. \end{cases} \tag{28}$$

For constant $\sigma_{r,n}(\tilde{x}) = \sigma_{r,n}$ (*i.e.*, stationary noise) this is the log-likelihood of Tobit regression with clipping at 0 from below and at 1 from above [35]. In our special case, we use a non-constant link function for the standard deviation, *i.e.*

$$\sigma_{r,n}(\tilde{x}) = \sqrt{\beta_1^{r,n}\tilde{x} + \beta_2^{r,n}} \tag{29a}$$
$$= \sqrt{(\beta_1^{r} + \beta_1^{n})\tilde{x} + \beta_2^{r} + \beta_2^{n}} \tag{29b}$$

in order to define our heteroscedastic Tobit regression model.

To estimate its parameters, we minimize the negative log-likelihood of all data points

$$(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) = \underset{\boldsymbol{\alpha}, \boldsymbol{\beta}^{r,n}}{\arg\min} \sum_i -\log \mathcal{T}\big(x_n^{(i)} \mid x_r^{(i)}\big). \tag{30}$$

B.2. Log-likelihood Gradient

It is useful to first derive the partial derivatives of terms of the form ${}^{(c-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}$ for some constant c w.r.t. all variables. The partial derivatives can be shown to be given as:

$$\frac{\partial {}^{(c-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_1^{r,n}} = -\frac{1}{2}\big(c - \boldsymbol{\alpha}^\top \mathbf{x}\big) \cdot \big(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}\big)^{-3/2} \cdot \boldsymbol{\alpha}^\top \mathbf{x} \tag{31a}$$

$$\frac{\partial {}^{(c-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_2^{r,n}} = -\frac{1}{2}\big(c - \boldsymbol{\alpha}^\top \mathbf{x}\big) \cdot \big(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}\big)^{-3/2} \tag{31b}$$

$$\frac{\partial {}^{(c-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \boldsymbol{\alpha}} = -\mathbf{x} \cdot \big(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}\big)^{-1/2} - \frac{1}{2}\big(c - \boldsymbol{\alpha}^\top \mathbf{x}\big) \cdot \big(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}\big)^{-3/2} \cdot \beta_1^{r,n}\mathbf{x}. \tag{31c}$$

That allows to derive the partial derivatives for all three cases of the log-likelihood function. For the first case they are given as

$$\frac{\partial \log \mathcal{N}\big(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)}{\partial \beta_1^{r,n}} = -\frac{1}{2}\big(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}\big)^{-1} \cdot \boldsymbol{\alpha}^\top \mathbf{x} - \frac{x_n - \boldsymbol{\alpha}^\top \mathbf{x}}{\sqrt{\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}}} \cdot \frac{\partial {}^{(x_n-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_1^{r,n}} \tag{32a}$$

$$\frac{\partial \log \mathcal{N}\big(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)}{\partial \beta_2^{r,n}} = -\frac{1}{2}\big(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}\big)^{-1} - \frac{x_n - \boldsymbol{\alpha}^\top \mathbf{x}}{\sqrt{\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}}} \cdot \frac{\partial {}^{(x_n-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_2^{r,n}} \tag{32b}$$

$$\frac{\partial \log \mathcal{N}\big(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)}{\partial \boldsymbol{\alpha}} = -\frac{1}{2}\big(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}\big)^{-1} \cdot \beta_1^{r,n}\mathbf{x} - \frac{x_n - \boldsymbol{\alpha}^\top \mathbf{x}}{\sqrt{\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{x} + \beta_2^{r,n}}} \cdot \frac{\partial {}^{(x_n-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \boldsymbol{\alpha}}. \tag{32c}$$

To compute the last term of each equation we employ Eqs. (31a) – (31c). For the second case the gradient of the

Applied on	Evaluated on	WNNM	KSVD	EPLL	NCSR	BM3D	MLP	TNRD	FoE
RAW	RAW	0.971	0.968	0.968	0.853	**0.972**	0.939	0.963	0.967
RAW	sRGB	**0.933**	0.919	0.931	0.713	**0.933**	0.886	0.894	0.907
RAW+VST	RAW	**0.974**	0.972	0.973	0.969	**0.974**	0.963	0.961	0.955
RAW+VST	sRGB	**0.935**	0.931	0.927	0.924	0.932	0.916	0.892	0.914
sRGB	sRGB	0.866	**0.900**	0.829	0.834	0.855	0.838	0.708	0.887

Table 4. Mean SSIM [36] of the denoising methods tested on our benchmark dataset. We apply denoising either on linear raw intensities, after a variance stabilizing transformation (VST), or after conversion to the sRGB space. Likewise, we evaluate the result either in linear raw space or in sRGB space. The noisy images have a SSIM of 0.863 (linear raw) and 0.710 (sRGB).

log-likelihood is given by

$$\frac{\partial \log \Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)}{\partial \beta_1^{r,n}} = \frac{\mathcal{N}\big(0 \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)}{\Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)} \cdot \frac{\partial {}^{(-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_1^{r,n}} \tag{33a}$$

$$\frac{\partial \log \Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)}{\partial \beta_2^{r,n}} = \frac{\mathcal{N}\big(0 \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)}{\Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)} \cdot \frac{\partial {}^{(-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_2^{r,n}} \tag{33b}$$

$$\frac{\partial \log \Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)}{\partial \boldsymbol{\alpha}} = \frac{\mathcal{N}\big(0 \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)}{\Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)} \cdot \frac{\partial {}^{(-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \boldsymbol{\alpha}}, \tag{33c}$$

again employing Eqs. (31a) – (31c). The third case works analogously. In practice, we optimize for $\boldsymbol{\beta}' = \log \boldsymbol{\beta}^{r,n}$ to ensure that $\boldsymbol{\beta}^{r,n}$ is positive. Furthermore, we exclude pixels near image edges [13] from the regression and truncate the log-likelihood to be robust to outliers, *i.e.* we set the gradients to zero for pixels with $\log \mathcal{T}(x_n^i \mid x_r^i) < -10$.

When estimating the $\boldsymbol{\alpha}$ parameter for the image pairs in our dataset, we use previously recorded noise parameters $\boldsymbol{\beta}$. These were obtained from running our full Tobit regression on controlled images showing a color checker, see Fig. 6.

B.3. Approximation of Noise Term

Here, we quantify the error that is induced by approximating the noise term in Eqs. (7a) – (7b) of the main paper. Specifically, we approximate the variance of the Gaussian noise by

$$\alpha_1^2 \left(\beta_1^r x_r + \beta_2^r\right) \approx \beta_1^r (\alpha_1 x_r + \alpha_2) + \beta_2^r. \tag{34}$$

Obviously, the left-hand side would converge to the right-hand side as $\alpha_1 \to 1$ and $\alpha_2 \to 0$, if the ISO value and exposure time were set with perfect accuracy. In practice, however, this is not the case. We now evaluate the practical impact of our approximation. With denoting $\beta(x_r) = \beta_1 x_r + \beta_2$, let

$$\sigma(x_r) = \sqrt{\alpha_1^2 \beta^r(x_r) + \beta^n(\alpha(x_r))} \tag{35}$$

be the true noise level function and

$$\hat{\sigma}(x_r) = \sqrt{\beta^{r,n}(\alpha(x_r))} \tag{36}$$

be the approximated noise level function. We compute the normalized root mean squared error Φ [27] between the true and the approximated noise level function, assuming a uniform distribution over pixel intensities

$$\Phi(\sigma, \hat{\sigma}) = \int_0^1 \frac{\left(\sigma(x_r) - \hat{\sigma}(x_r)\right)^2}{\sigma(x_r)} \, \mathrm{d}x_r. \tag{37}$$

The average normalized RMSE on our dataset is $1.4 \cdot 10^{-4}$, meaning that on average approximating the noise standard deviation introduces a relative error of 0.014%. This is insignificant compared to the overall estimation accuracy (see Sec. 5 of the paper).

Figure 6. Test scene used for noise parameter calibration.

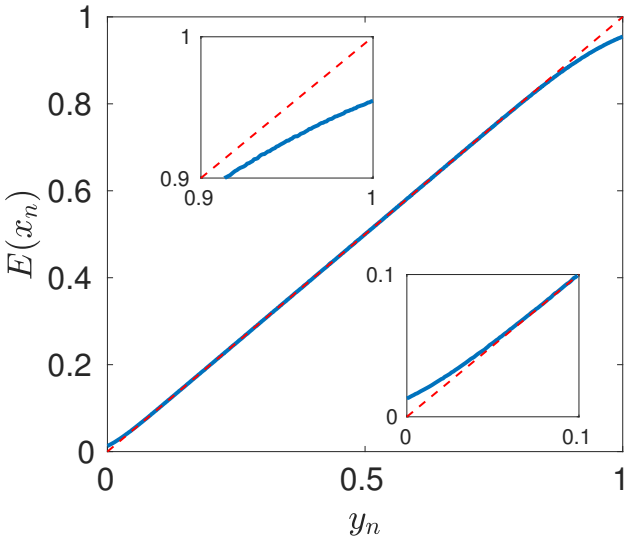


Figure 7. Noise-free intensities (red dashed line) *vs*. mean of clipped noisy intensities (blue solid line).

C. Bias from Clipping

Figure 7 plots the noise-free image intensities y_n against the average of clipped noisy observations x_n for the noise level function of the Nexus 6P at ISO 6400. We can see that the mean of the clipped observations strongly deviates from the true noise-free intensities near the clipping boundaries, also see [11]. Due to this bias introduced by clipping the signal, we can not recover the noise free signal by simply averaging noisy observations spatially or temporally. Hence, we perform the smoothing operation of our low-frequency residual correction in the debiased domain, *c.f*. Eqs. (11) and (12).

D. Simulation of Poisson-Gaussian Noise

For our experiments on synthetic data (Sec. 5 of main paper) we apply Poisson-Gaussian noise to the noise-free images (Eq. 13). To demonstrate that Eq. (13) is sensible let x'_n again be the unclipped noisy signal. According to the heteroscedastic Gaussian noise model, *c.f*. Eqs. (2a) and (2b), the mean and variance of x'_n are given by

$$\mathbb{E}(x'_n) = y_n, \tag{38}$$

$$\mathrm{Var}(x'_n) = \beta_1^n y_n + \beta_2^n. \tag{39}$$

Let now z_n be the unclipped simulated noisy signal of Eq. (13):

$$z_n \sim \beta_1^n \cdot \mathcal{P}\left(y_n/\beta_1^n\right) + \mathcal{N}\left(0, \sqrt{\beta_2^n}\right) \tag{40}$$

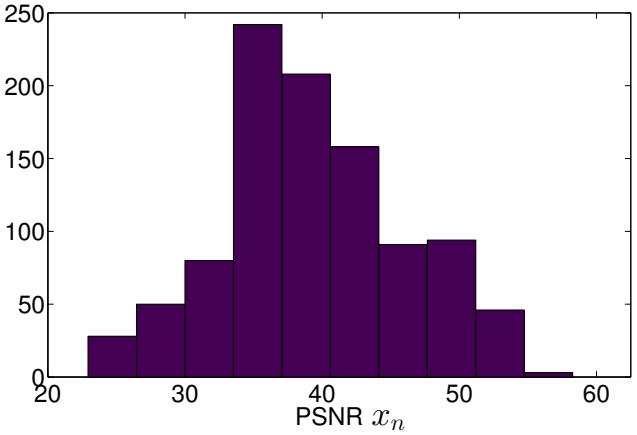


Figure 8. Histogram of PSNR values (in dB) of the crops of the noisy test images.

According to the properties of the Poisson distribution, the mean and variance of z_n are given by

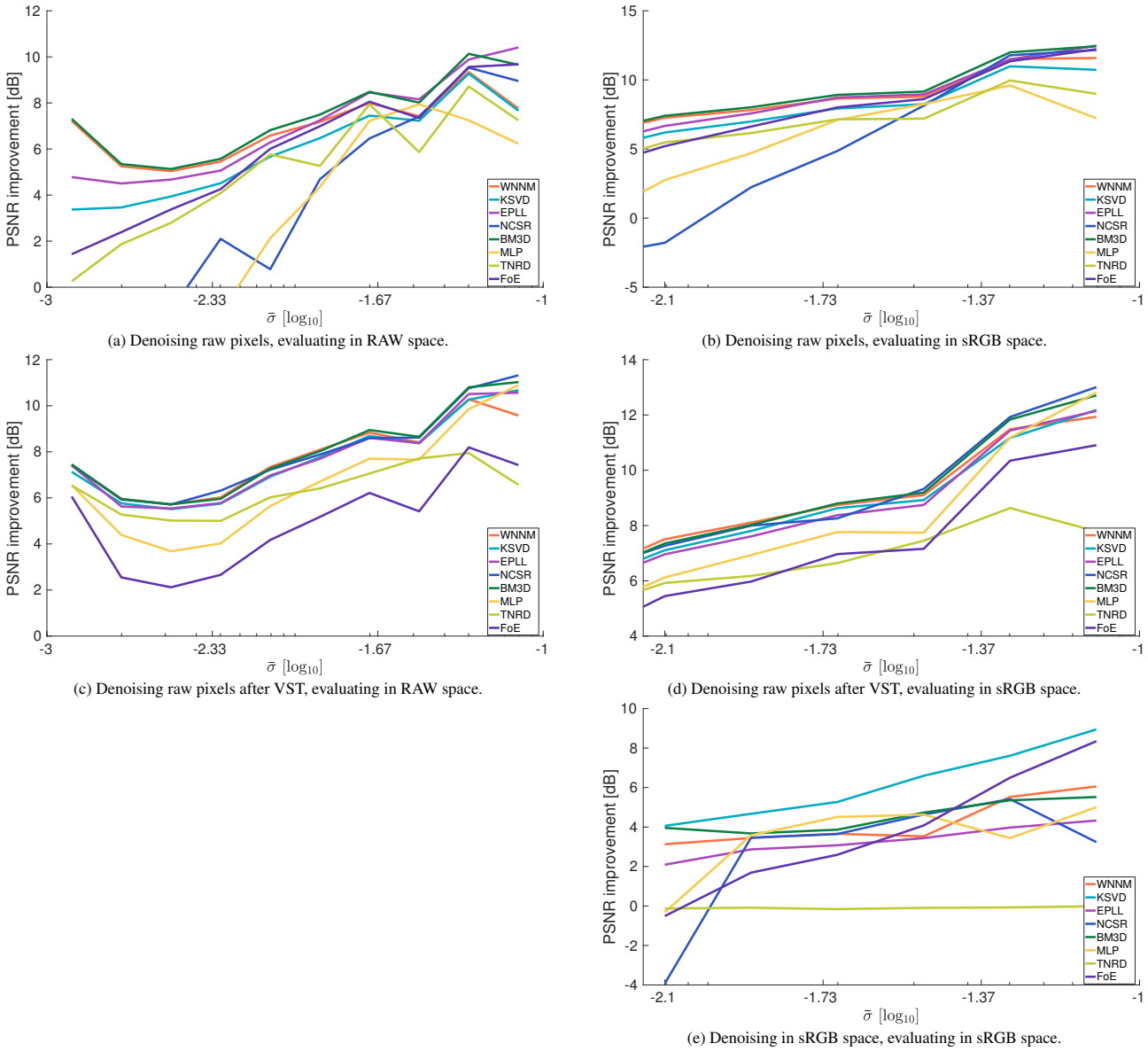
$$\mathbb{E}(z_n) = \beta_1^n \frac{y_n}{\beta_1^n} + 0 = y_n, \tag{41}$$

$$\mathrm{Var}(z_n) = (\beta_1^n)^2 \frac{y_n}{\beta_1^n} + \beta_2^n = \beta_1^n y_n + \beta_2^n. \tag{42}$$

We can see that the two first moments of x'_n and z_n match and hence z_n provides a good simulation of the noise as given by the noise level function $\boldsymbol{\beta}^n$. The same holds for the simulation of the reference image.

E. Additional Results

Finally, we give a few more results obtained on our novel DND benchmark dataset. First, Fig. 8 shows a histogram of the PSNR values of the crops of the noisy test images in linear raw space. As we can see, our dataset covers a wide range of noise levels for the noisy images, hence allowing to benchmark denoising algorithms across many different situations. Note that the mean PSNR of the noisy images (39.38 dB) is significantly below the PSNR of the reference images (52.76 dB, from the estimated noise level function). Consequently, the ground truth accuracy of our benchmark far exceeds the performance of state-of-the-art denoising techniques (*c.f*. Table 3), thus providing significant headroom even for future improvement in denoising techniques. Figure 9 shows denoising results aggregated for different noise levels. The top-performing methods overall achieve consistent results across almost all noise levels. We can furthermore observe that NCSR has severe problems in denoising images affected by weak intensity-dependent noise. When applying the variance stabilizing transformation, NCSR shows a more competitive performance. For MLP we observe that performance on RAW



(a) Denoising raw pixels, evaluating in RAW space.

(b) Denoising raw pixels, evaluating in sRGB space.

(c) Denoising raw pixels after VST, evaluating in RAW space.

(d) Denoising raw pixels after VST, evaluating in sRGB space.

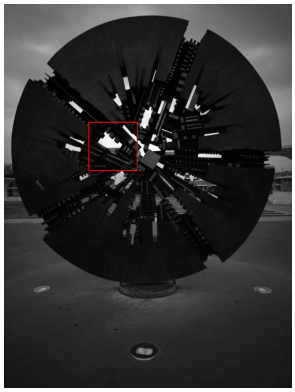
(e) Denoising in sRGB space, evaluating in sRGB space.

Figure 9. Denoising performance by noise level $\bar{\sigma}$.

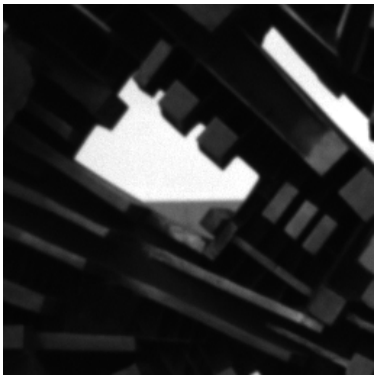
denoising peaks for $\hat{\sigma}$ close to the noise level used for training, *i.e.* $\sigma_{\text{train}} \approx 10^{-1.41}$. For removing noise with a different noise level, MLP does not generalize well.

Table 4 provides SSIM [36] results for the tested methods on our benchmark. Generally, the conclusions made in Sec. 6 of the paper based on PSNR values generalize to the SSIM results. As we can see, BM3D and WNNM show the best performance and their scores differ only marginally. Overall, we observe that SSIM scores are high across all methods.

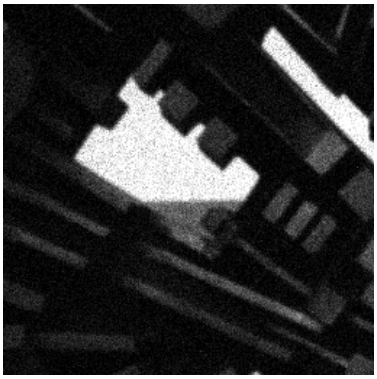
Finally, Figures 10 – 13 show denoising results of the tested algorithms for one crop of two different images in our database. The results were obtained from denoising raw intensities after the variance stabilizing transformation. We display the denoised images both in linear raw space (red channel only) and in sRGB space after our camera processing pipeline, *c.f.* Sec. 6 of the paper. On Figs. 10 and 12 we can see that many methods oversmooth fine structures (*e.g.*, MLP and FoE), while TNRD undersmoothes and fails to remove a significant part of the noise. When looking at the results in sRGB space (Figs. 11 and 13), we can see that denoising introduces visually apparent color artifacts for all methods. Moreover, the noise is clearly spatio-chromatically correlated in sRGB space.



(a) Full-sized reference



(b) Reference



(c) Noisy (26.59 dB)

(d) WNNM (37.02 dB)

(e) KSVD (35.87 dB)

(f) EPLL (35.50 dB)

(g) NCSR (37.80 dB)

(h) BM3D (36.81 dB)

(i) MLP (37.61 dB)

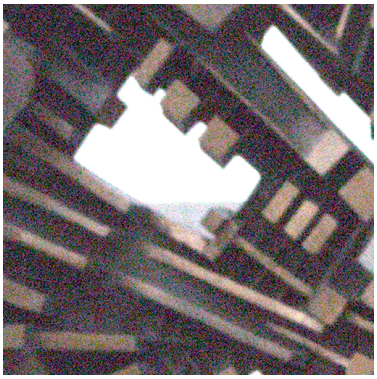
(j) TNRD (32.38 dB)

(k) FoE (32.39 dB)

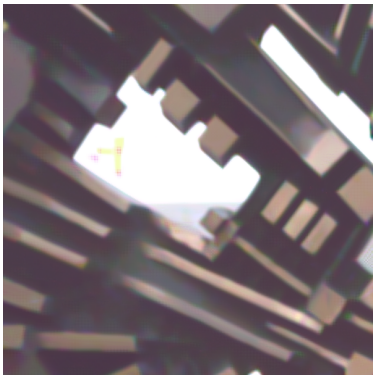
Figure 10. Example denoising result (red channel only) with PSNR values, displayed in linear raw space.

(a) Full-sized reference

(b) Reference



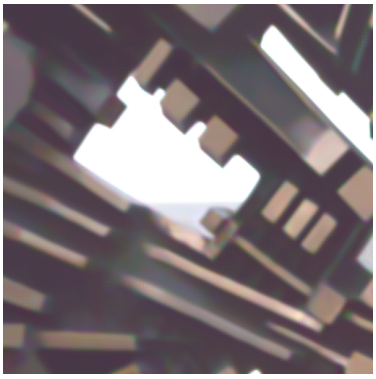
(c) Noisy (19.17 dB)



(d) WNNM (30.85 dB)

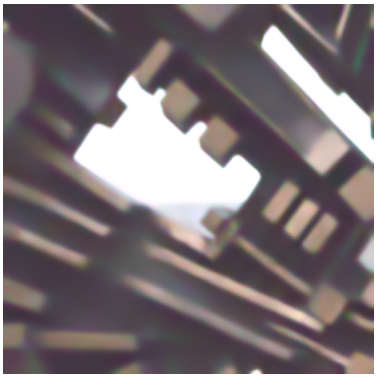
(e) KSVD (29.76 dB)

(f) EPLL (30.26 dB)

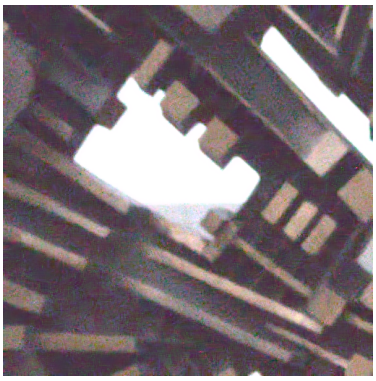


(g) NCSR (31.90 dB)

(h) BM3D (31.36 dB)



(i) MLP (31.38 dB)



(j) TNRD (26.51 dB)

(k) FoE (27.81 dB)

Figure 11. Example denoising result (red channel only) with PSNR values, displayed in sRGB space.

(a) Full-sized reference

(b) Reference

(c) Noisy (36.34 dB)

(d) WNNM (45.46 dB)

(e) KSVD (45.04 dB)

(f) EPLL (45.54 dB)

(g) NCSR (45.51 dB)

(h) BM3D (45.78 dB)

(i) MLP (44.58 dB)

(j) TNRD (40.44 dB)

(k) FoE (42.50 dB)

Figure 12. Example denoising result (red channel only) with PSNR values, displayed in linear raw space. Intensities of crops are uniformly scaled for better display.

(a) Full-sized reference

(b) Reference

(c) Noisy (23.74 dB)

(d) KSVD (35.21 dB)

(e) WNNM (34.66 dB)

(f) EPLL (34.83 dB)

(g) NCSR (35.23 dB)

(h) BM3D (35.37 dB)

(i) MLP (35.43 dB)

(j) TNRD (31.94 dB)

(k) FoE (34.00 dB)

Figure 13. Example denoising result with PSNR values, displayed in sRGB space.