Graphics and Visual Computing 7 (2022) 200058

Contents lists available at ScienceDirect

Graphics and Visual Computing

journal homepage: www.elsevier.com/locate/gvc

Technical section

Locally-guided neural denoising☆

Lukas Bode [a,*], Sebastian Merzbach [a], Julian Kaltheuner [a], Michael Weinmann [b], Reinhard Klein [a]

[a] University of Bonn, Friedrich-Hirzebruch-Allee 8, Bonn, 53115, Germany
[b] Delft University of Technology, Van Mourik Broekmanweg 6, Delft, 2628 XE, Netherlands

ARTICLE INFO

Article history:
Received 21 July 2022
Received in revised form 20 September 2022
Accepted 22 September 2022
Available online 27 September 2022

Keywords:
Computer graphics
Image processing
Denoising
SVBRDF restoration

ABSTRACT

Noise-like artifacts are common in measured or fitted data across various domains, e.g. photography, geometric reconstructions in terms of point clouds or meshes, as well as reflectance measurements and the respective fitting of commonly used reflectance models to them. State-of-the-art denoising approaches focus on specific noise characteristics usually observed in photography. However, these approaches do not perform well if data is corrupted with location-dependent noise. A typical example is the acquisition of heterogeneous materials, which leads to different noise levels due to different behavior of the components either during acquisition or during reconstruction. We address this problem by first automatically determining location-dependent noise levels in the input data and demonstrate that state-of-the-art denoising algorithms can usually benefit from this guidance with only minor modifications to their loss function or employed regularization mechanisms. To generate this information for guidance, we analyze patchwise variances and subsequently derive per-pixel importance values. We demonstrate the benefits of such locally-guided denoising at the examples of the Deep Image Prior method and the Self2Self method.

1. Introduction

Data containing high levels of noise poses a huge problem for many applications in entertainment, advertisement, and design. Immersive experiences of scenes and objects rely on respective high-fidelity depictions and are significantly impacted by noisy data resulting from the capture or modeling process. Unfortunately, certain types and levels of noise cannot be avoided during data capture. Physical or economic constraints might affect the choice of the sensor or the amount and quality of the data that can be handled while meeting requirements regarding the computational burden for a task. Therefore, methods are typically designed to be robust to noisy data. Whereas certain types of noise including sensor noise can typically be handled robustly, the robustness to other noise types including compression artifacts or missing data is often still lacking and relies on sophisticated denoising methods.

In the field of appearance capture and modeling – which is concerned with creating photo-realistic virtual models that capture details regarding surface geometry and reflectance behavior of their real-world counterparts – noisy 3D measurements, inaccurate calibration and image noise have to be dealt with. Oftentimes, these are corrupted by non-uniform location-dependent

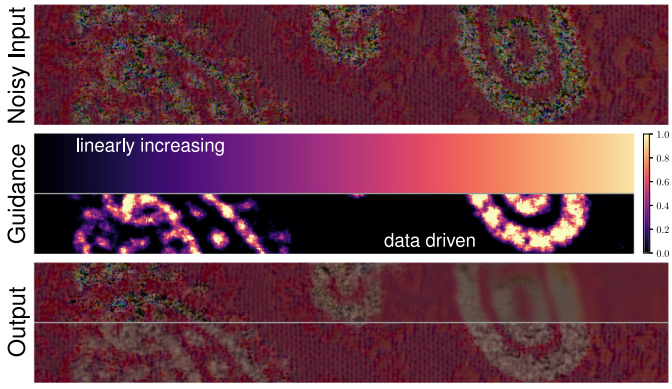

Fig. 1. We present a novel approach to remove spatially concentrated noise from images. Given a noisy input image (top row), a guidance map (middle row) can be used to control the denoising intensity of state-of-the-art denoising algorithms in a spatially-varying manner. We propose a fully automatic way to generate such guidance information by detecting noisy pixels in the input (middle row, bottom half). Hereby, corrupted pixels of the input can be denoised while preserving fine details in others (bottom).

☆ This article was recommended for publication by M. Wimmer.

* Corresponding author.
E-mail addresses: lbode@cs.uni-bonn.de (L. Bode), merzbach@cs.uni-bonn.de (S. Merzbach), kaltheun@cs.uni-bonn.de (J. Kaltheuner), M.Weinmann@tudelft.nl (M. Weinmann), rk@cs.uni-bonn.de (R. Klein).

https://doi.org/10.1016/j.gvc.2022.200058

noise as depicted in Fig. 1. The typical way to handle such data is to apply image restoration algorithms like super-resolution, denoising and inpainting, which aim at recovering an original image x from its corrupted version $\tilde{x}$. This can be stated in terms of the optimization problem

$$\arg\min_x E(x;\tilde{x}) + R(x) \tag{1}$$

with the data term $E(x;\tilde{x})$ and a regularization term $R(x)$. In contrast to the task-specific data term, finding a good prior $R(x)$ is challenging. For a surjective mapping $g : \theta \mapsto x$, Functional (1) corresponds to

$$\arg\min_\theta E(g(\theta);\tilde{x}) + R(g(\theta)). \tag{2}$$

As shown by Ulyanov et al. [1], the choice of a *good* (possibly injective) mapping g allows getting rid of the prior term. By defining $g(\theta)$ as $f_\theta(z)$, where f is given by a deep neural network with parameters θ and using a fixed input z, we obtain

$$\arg\min_\theta E(f_\theta(z);\tilde{x}), \tag{3}$$

which can be solved based on gradient descent, i.e. we optimize the neural network's parameters to finally represent the searched restored version of the image $x^* = f_{\theta^*}(z)$ based on the optimal network weights θ^*. In other words, the underlying inverse problem is regularized by the deep network itself. Other approaches [2,3] combined this approach with additional priors.

These restoration methods like the Deep Image Prior are hard to control, which poses a problem in the above example of data corrupted with location-dependent noise. Applying the Deep Image Prior approach without modification results in either loss of fine details or carrying over large amounts of artifacts.

In this paper, we propose a method to control the training of state-of-the-art learning-based denoising algorithms to enable successful restoration of such data. At the example of the restoration of fitted spatially-varying bidirectional reflectance distribution function (SVBRDF) textures that describe the reflectance behavior of surfaces, we show how characteristic properties of occurring artifacts can be leveraged to guide the optimization. In particular, we introduce spatially varying guidance by means of a per-pixel importance value, which can be calculated in a fully automatic manner by analyzing patchwise variances in the image. Based on two exemplary denoising approaches [1,4], we show how to utilize the per-pixel importance values during the denoising process with only minor modifications to the original algorithms (see Fig. 1). We validate the potential of our approaches in comparison to the respective original algorithms without modifications as well as another learning-based state-of-the-art denoising method [5], where our approaches outperform the baselines in terms of image quality of the restored images.

2. Related work

In the context of model-based optimization for inverse problems such as restoration, denoising, superresolution and deblurring, it is well-known that the typically involved regularization term has a significant influence on the resulting performance. Therefore, lots of effort has been spent on finding good denoiser priors. Total variation [6,7] has been widely applied, but the results may exhibit watercolor-like artifacts. Further approaches include Gaussian mixture models (GMM) [8] and the computationally expensive K-SVD denoiser prior [9]. Furthermore, non-local means [10] as well as block-matching and 3D filtering (BM3D) [11] tend to oversmooth irregular structures for images that do not exhibit self-similarities. As leveraging the correlation between different color channels by jointly handling them has been shown to lead to better performance in comparison to the independent handling of color channels [12], several works focused on color priors (e.g. [13–15]). Popular techniques such as CBM3D [13] rely on first decorrelating the image into a luminance-chrominance color space and subsequently applying the gray BM3D method for each transformed color channel separately. However, the resulting luminance-chrominance color channels still remain correlated [16], which indicates that it might be beneficial to jointly handle RGB channels.

Instead of the aforementioned hand-designed approaches, recent work particularly focused on learning-based methods to find the respective color image priors capturing characteristics of the given data. The learned deep CNN denoiser prior by Zhang et al. [17] benefits from the parallelization of the inference on the GPU and exploits the prior modeling capacity offered by deep architectures. Building on this work, the denoising algorithm by Yang et al. [18] utilizes ensemble learning to improve on the results, while Quan et al. [19] designed a complex-valued CNN to leverage insights from classical image recovery algorithms. However, the approach involves a training on a large dataset of thousands of clean/noisy image pairs. Despite relying on an image dataset for training as well, Recorrupted-to-Recorrupted [20] lifted the requirement for clean images in the dataset by proposing to learn a mapping of corrupted images to other corrupted images following the same noise distribution but with the noise being independent of the noise in the input image. Consequently, the clean image can be found by the averaging of multiple corrupted images. In contrast, the untrained approach by Ulyanov et al. [1] on Deep Image Priors (DIPs) shows that low-level statistics of a single input image can be sufficiently captured by the structure of a single DIP generator network. Invariance to adversarial perturbations and the suppression of non-robust image features are particularly achieved in the early iterations [21] after which overfitting starts to occur. To avoid the need for early stopping, i.e. finding a suitable number of iterations where the image prior does not overfit to noise characteristics or artifacts, other works rely on Bayesian approaches [22,23] or under-parametrization based on deep decoder approaches [24] to prevent the overfitting and reach a stable convergence behavior. Further work on DIPs focused on optimizing the underlying network architecture as part of the denoising process [25,26]. The potential of such deep priors have also been demonstrated for hyperspectral image denoising [27,28] and even for surface reconstruction [29–31].

A similar approach, which in contrast to DIP is not relying on early stopping, has been introduced with Self2Self by Quan et al. [4]. Instead of finding a mapping from fixed noise to the input image, they use a similar U-Net architecture to find a mapping from a noisy input image to a clean image directly. Regularization is handled by employing a Bernoulli input masking scheme as well as dropout in the decoder layers.

More recently, CVF-SID [5] has been proposed as an approach for self-supervised single image denoising by disentangling clean image, signal-dependent noise and signal-independent noise in an end-to-end fashion. In the field of nonblind image deconvolution, Chen et al. [32] introduced a spatially-adaptive dropout scheme to handle the solution ambiguity introduced by the deblurring problem. While also assuming the input image to be corrupted by Gaussian white noise, they rely on the assumption of the noise being uniformly distributed over the image and independent of the image signal in order to denoise the image during the deblurring process.

3. Methodology

The goal of our work is to widen the range of problems, commonly used learning-based self-supervised single image denoising algorithms can be successfully applied to. While not being the only use-case for our work, we are specifically targeting the problem of denoising images with an arbitrary number of channels corrupted with location-dependent noise instead of noise being uniformly distributed over the whole image. Current state-of-the-art algorithms tend to either introduce additional blurriness in originally clean pixels or are not capable of sufficiently removing the noise from the image.

We first propose the calculation of importance images based on an estimated per-pixel noise level. Subsequently, we present exemplary minor adjustments to the Deep Image Prior (DIP) as well as the Self2Self (S2S) method in order to guide their denoising process according to the importance values.

3.1. Inference of a guidance map

We propose the guidance of image processing operations like denoising based on a guidance map in terms of a per-pixel importance value $m(x, y)$ for pixel (x, y), where values close to 1 indicate that the pixel of the input image should be preserved in the denoised image while pixels with importance close to 0 should be denoised as much as possible as they are assumed to have a low signal-to-noise ratio. Note, that this importance is directly related to the noise level of a pixel via

$$m(x, y) = 1 - n(x, y), \tag{4}$$

where $n(x, y)$ is the noise level for pixel (x, y). While calculating the true noise level from the image signal is an underconstrained problem, for the purpose of the guidance map it suffices to find a rough estimate of it as we can rely on the natural regularization capabilities of the underlying denoising algorithms. If working with RGB images, noise level estimates are calculated independently for all channels. The maximum over the noise levels of all channels is calculated before the remapping step described in Section 3.1.3. For the remainder of this section, we assume to be working with grayscale images for notational simplicity.

3.1.1. Variance-based noise level estimation

Building on the assumption that noisy regions usually have a high variance, the naive way would be to estimate the per-pixel noise level as patchwise variance of the respective pixel neighborhood. The variance for such a neighborhood $\mathcal{N}(x, y) \subseteq \mathcal{I}$ is defined as

$$n_{\text{var}}(x, y) = \frac{1}{|\mathcal{N}(x, y)|} \sum_{(x', y') \in \mathcal{N}(x, y)} (\mathcal{I}(x', y') - \mu(\mathcal{N}(x, y)))^2 \tag{5}$$

and the mean over an arbitrary set of pixels $\mathcal{P}$ is defined as

$$\mu(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{(x', y') \in \mathcal{P}} \mathcal{I}(x', y'). \tag{6}$$

However, this noise level estimate is prone to erroneously high values at discontinuities in the input image which we typically want to preserve in the denoised image making this method applicable only for very smooth images.

3.1.2. SVD-based noise level estimation

To alleviate the aforementioned problem and allow for better adaptation to local noise characteristics, we apply a local noise level estimation. For this purpose, we propose to split the pixel neighborhood $\mathcal{N}(x, y)$ into two disjoint subsets $\mathcal{N}_{\text{lower}}(x, y)$ and $\mathcal{N}_{\text{upper}}(x, y)$ depending on whether the respective pixel is below or above the patch mean $\mu(\mathcal{N}(x, y))$, such that

$$\mathcal{N}(x, y) = \mathcal{N}_{\text{lower}}(x, y) \cup \mathcal{N}_{\text{upper}}(x, y) \tag{7}$$

and

$$\mathcal{N}_{\text{lower}}(x, y) \cap \mathcal{N}_{\text{upper}}(x, y) = \emptyset. \tag{8}$$

Subsequently, pixels of both subsets are lifted into $\mathbb{R}^3$

$$\mathcal{N}^3_{\{\text{lower,upper}\}}(x, y) = \left\{ \begin{pmatrix} x' \\ y' \\ \mathcal{I}(x', y') \end{pmatrix} \middle| (x', y') \in \mathcal{N}_{\{\text{lower,upper}\}}(x, y) \right\}, \tag{9}$$

where $\cdot_{\{a,b\}}$ combines equations for $\cdot_a$ and $\cdot_b$ for notational simplicity. Afterwards, covariance matrices $M_{\mathcal{N}^3_{\{\text{lower,upper}\}}}$ can be constructed to perform an singular value decomposition (SVD) (dependence on the pixel (x, y) omitted for notational simplicity)

$$M_{\mathcal{N}^3_{\{\text{lower,upper}\}}} = U_{\{\text{lower,upper}\}} \Sigma_{\{\text{lower,upper}\}} V^T_{\{\text{lower,upper}\}}, \tag{10}$$

where $\sigma_{\{\text{lower,upper}\},i} = \Sigma_{\{\text{lower,upper}\},ii}$, i.e. the diagonal entries of matrices $\Sigma_{\{\text{lower,upper}\}}$, are the singular values. We are looking for the smallest singular value

$$\sigma_{\{\text{lower,upper}\},\min} = \min_{i \in \{1,2,3\}} \sigma_{\{\text{lower,upper}\},i} \tag{11}$$

of each subset as this value can be interpreted as the variance, and therefore the amount of noise, of the subset in normal direction of a plane fitted to the respective pixels. As this analysis is conducted for both subsets of pixels individually, the approach is robust against image discontinuities in contrast to relying on the patchwise variance directly. Additionally, due to the SVD, smooth color gradients are not detected as noise either. These two partial noise level estimates can be reduced to an estimate for the whole patch by choosing an appropriate reduction operator. Experiments have shown that the results are best using the minimum of σ_{lower} and σ_{upper}. We argue, that an additional robustness against detecting high-frequency details in the image as noise is more important than additional accuracy in estimating the noise level. The noise level can therefore be estimated as

$$n_{\text{svd}}(x, y) = \min_{s \in \{\text{lower,upper}\}} \sigma_{s,\min}(x, y). \tag{12}$$

3.1.3. Remapping

Despite noisy pixels having usually higher estimated noise level values $n_{\{\text{var,svd}\}}(x, y)$, we can still observe significant values for clean image pixels as well. Non-zero noise level estimates might prevent the full overfitting of the denoising network to clean pixels and thus can introduce unwanted blurriness for respective pixels. To avoid this, we apply a remapping technique to generate the final guidance images.

We observed that the square root of the estimated noise levels, i.e. the standard deviation of pixel values, for clean pixels roughly follows a Gaussian distribution as depicted in Fig. 2. By calculating a histogram over the noise levels of all pixels, we find the bin with the highest pixel count as this is assumed to be the peak of the distribution with mean value $\sqrt{n_{\text{peak}}}$. Remapping our estimated noise level values using $(2\sqrt{n_{\text{peak}}})^2 = 4n_{\text{peak}}$ as lower bound and the 95th percentile $n_{0.95}$ as upper bound and clamping to $[0, 1]$ finally yields robust guidance images

$$m_{\{\text{var,svd}\}}(x, y) = 1 - \frac{n_{\{\text{var,svd}\}} - 4n_{\text{peak}}}{n_{0.95} - 4n_{\text{peak}}}. \tag{13}$$

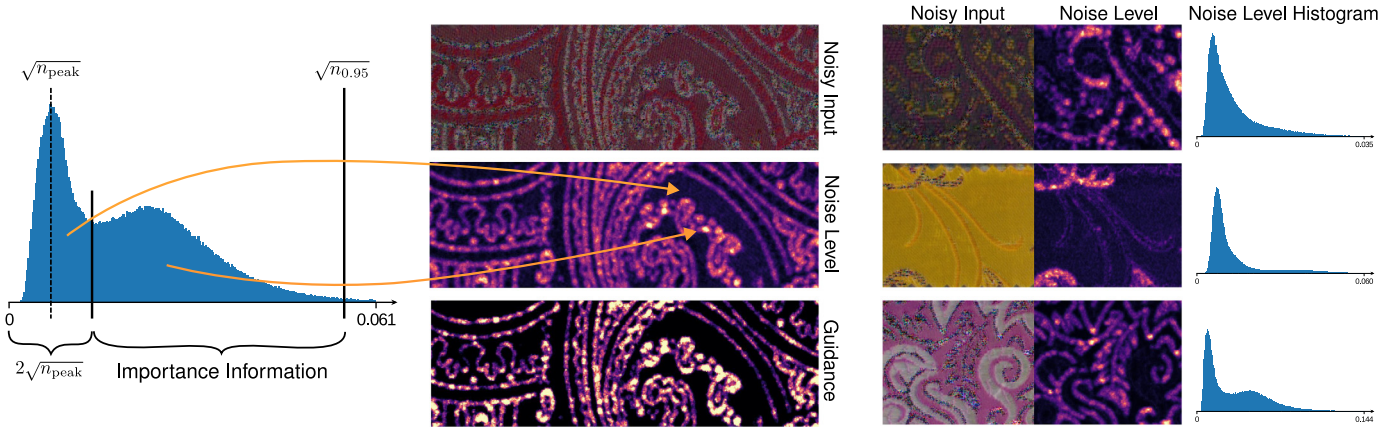

Fig. 2. Remapping procedure: Square root of estimated noise levels, i.e. the standard deviation of pixel values, roughly follows a Gaussian distribution for noise-free pixels. By finding the peak of the distribution, the respective pixels can be discarded (left) and the remaining part up to the 95th percentile is remapped to [0, 1], resulting in a robust guidance map (middle). This holds for other examples as well (right).

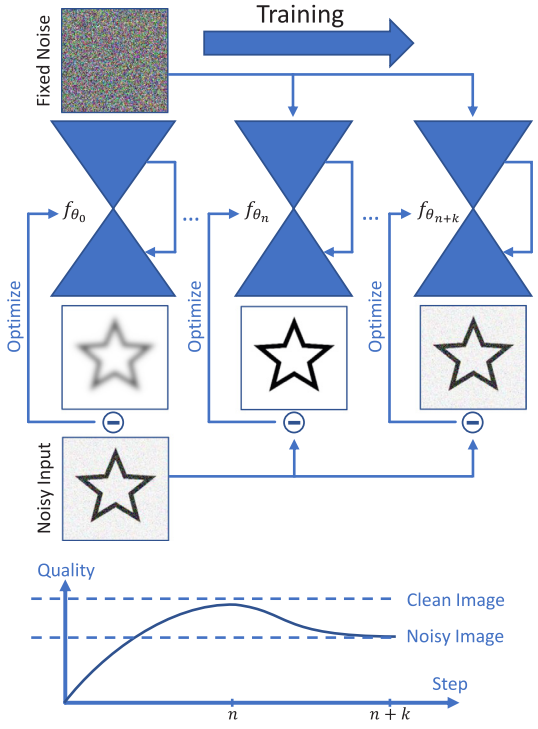

Fig. 3. Original DIP: Based on a noisy input image, a modified U-Net [33] is trained to map a fixed noise image to the noisy input image itself. Over the course of the training, natural image content is learned first due to the inherent regularization capabilities of the network. Early stopping is applied to stop the training process as soon as maximum quality is reached. If the network is trained further, the network output will finally converge to the actual noisy input image.

3.2. Guided Deep Image Prior

The generated guidance images can easily be used in state-of-the-art denoising algorithms, such as DIP [1]. This approach uses a neural network as natural prior for image restoration tasks including denoising. As depicted in Fig. 3, the input for the network is a fixed noise image with 32 channels consisting of uniformly distributed random numbers in the range [0, 0.1]. Iteratively, the network learns a mapping from the noise image to the noisy input image by minimizing an $\mathcal{L}_2$-loss. During this training process, the network learns the more natural low-frequency components of the noisy input image first, while the high-frequency components are only learned in later stages. Hence, by interrupting the training process before the network learns to reconstruct the unwanted noise, we can consider the network output as denoised image. For further regularization, random noise drawn from the normal distribution $N(0, 1/30)$ is added to the network input in each step to further regularize the training process.

As in the original approach, the output of the network is smoothed over multiple iterations with an exponential weight according to

$$\mathcal{I}^i = 0.01\hat{\mathcal{I}}^i + 0.99\mathcal{I}^{i-1}, \tag{14}$$

where $\mathcal{I}^i$ is the output image in iteration i and $\hat{\mathcal{I}}^i$ is the actual network prediction. This way, artifacts accidentally produced by the trained network are mostly smoothed out resulting in more accurate restorations.

Where not stated differently, we are using the same hyperparameters as the original approach in the denoising setting. In particular, we thus configure the network to have an encoder and a decoder each consisting of five double convolution layers with 128 filters. Each double convolution also contains batch normalizations and LeakyReLU activation functions. Reflection padding is used as it is described to work best by the authors [1].

Using the standard $\mathcal{L}_2$-loss as proposed by Ulyanov et al. [1] results in missing fine details in clean parts while the artifacts are already being learned by the network in corrupted ones and therefore being carried over to the output image. Since the artifacts are potentially restricted to some parts of the noisy input image due to systematic reasons, we propose a guided loss function to have further control over the restoration process.

We use guidance image described in Section 3.1 for this purpose. Depending on a pixel's importance we stop the training process early by weighting down the loss induced by the respective pixel according to a weight $w^i_{\text{dec}}(x, y)$. This weight depends on the current iteration number i reducing the respective pixels contribution to the loss over time. The resulting loss term is

$$\mathcal{L}^i_{\text{dec}} = \frac{1}{|\mathcal{I}|} \sum_{(x,y)\in\mathcal{I}} ((\hat{\mathcal{I}}^i(x, y) - \mathcal{I}(x, y)) \cdot w^i_{\text{dec}}(x, y))^2. \tag{15}$$

The decay weight $w^i_{\text{dec}}(x, y)$ is chosen to have an exponential fall-off after an initial warm-up phase with full contribution to the loss (dependence on the pixel (x, y) omitted for notational simplicity):

$$w^i_{\text{dec}} = \begin{cases} 1 & i < \kappa_w \\ (0.9 + 0.1\ m)^{(i-\kappa_w)\cdot\kappa_r} \cdot (1 - \kappa_c) + \kappa_c & \text{else,} \end{cases} \tag{16}$$

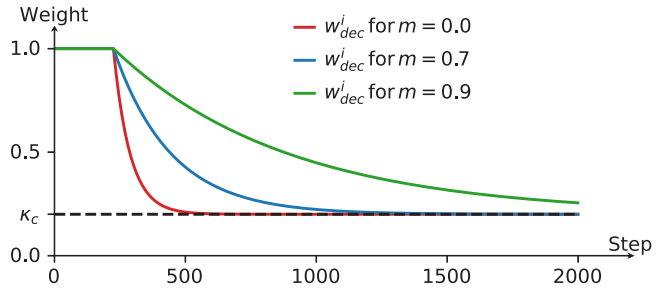

Fig. 4. Decaying per-pixel weight for DIP enables to locally control the denoising effect. After an initial warm-up phase (here $\kappa_w = 225$) with full denoising intensity, the weight decays exponentially (here $\kappa_r = 0.15$) and converges against a fixed lower bound (here $\kappa_c = 0.2$ for visualization purposes but set to $\kappa_c = 0.02$ during all experiments).

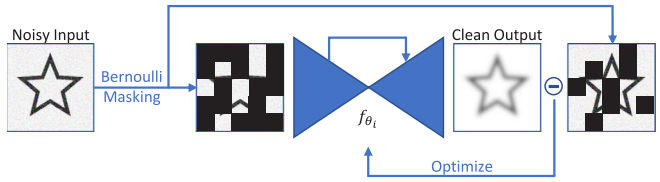

Fig. 5. Original S2S: A modified U-Net [33] is trained as an autoencoder to map a noisy input image to itself. During the training, two different regularization mechanisms are applied: First, Bernoulli masking is performed to split the noisy image into a subset of pixels used as network input and the other pixels being used as target image in the loss function. This way, the loss is calculated only on pixels unseen by the network in the respective iteration. Second, dropout in the decoder layers of the network further help to prevent the autoencoder to learn the noise.

where κ_w specifies the number of initial warm-up iterations without any decay of the weight, while κ_r controls the decay rate and κ_c specifies a lower bound for the contribution of a single pixel. Note that we designed w^i_{dec} to converge to κ_c instead of 0 in order to avoid artifacts where the guidance image does not fit the degenerate areas perfectly. We rely on the natural image prior property of the network itself to prevent overfitting to noise for these pixels. We are using $\kappa_w = 225$, $\kappa_r = 0.15$ and $\kappa_c = 0.02$ for all our experiments. Corresponding plots are depicted in Fig. 4.

3.3. Guided Self2Self

The second exemplary algorithm we are taking a closer look at here is S2S [4]. Similarly to DIP, this algorithm relies on the inherent regularization capabilities offered by neural networks. In contrast to DIP though, as depicted in Fig. 5, S2S uses an U-Net like network to map the noisy image to the restored image. To prevent overfitting, the authors add two additional regularization mechanisms: masking of the network input image and loss as well as dropout in the decoder layers.

The masking is performed by applying Bernoulli sampling to the input image such that we get a mask containing a 1 with probability p_m and a 0 otherwise. Thus, this mask can be used to divide the pixels of the input image in two subsets. Before the image is given to the network, it is multiplied by the mask. Therefore, only the first subset of pixels is contributing to the network output. Additionally, the L2 loss is modified to only use pixels which were not visible to the network in the respective iteration. Intuitively, the network is optimized to predict unseen pixels as close to the noisy input image as possible.

The dropout in decoder layers is another means of regularization. In contrast to most approaches relying on dropout, it is not deactivated in test mode. Instead, the network is evaluated multiple times with random dropout to simulate the training of multiple separate networks and averaging of the respective results. The authors have shown that this improves the quality of the resulting images.

Similarly to our modified DIP approach we are using the original authors' architecture and hyperparameters where not explicitly stated otherwise.

Analogously to DIP, S2S results in unwanted blurriness in clean regions which is a problem for our setting of spatially concentrated noise. Additionally, we have no control over the denoising intensity, which can be problematic if the signal-to-noise ratio is low in the input image. To alleviate these issues, we propose a generalization of S2S to allow the utilization of our guidance information as well as an additional denoising intensity parameter.

As in the original algorithm, the goal is to generate two binary image masks — $\mathcal{M}_i$ for masking the network input image and $\mathcal{M}_t$ for masking the difference image in the loss function. The underlying key idea here is to make the network overfit pixels with high-importance but denoise the image where importance is low. We achieve this by sampling the binary masks based on per-pixel probabilities $p_{\{i,t\}}$ defined as

$$p_i = p_{\text{imp}} \cdot 1 + (1 - p_{\text{imp}}) \cdot p_m \cdot p_d \tag{17}$$

$$p_t = p_{\text{imp}} \cdot 1 + (1 - p_{\text{imp}}) \cdot (1 - p_m) \cdot p_d. \tag{18}$$

with

$$p_{\text{imp}}(x, y) = m(x, y)^{\kappa_m}, \tag{19}$$

controlling the overfitting to high-importance image parts based on the guidance image m. Furthermore, κ_m controls the denoising strength for pixels with medium importance values (we set $\kappa_m = 2$ in all experiments) and p_d is a probability to discard a pixel completely from both masks to further increase the denoising effect. Where not stated explicitly, we use $p_d = 0.01$ for our experiments.

Intuitively, resulting masks can be thought of as linear interpolation between no masking happening at all, i.e. $\mathcal{M}_{\{i,t\}} = \mathcal{I}$, and standard S2S input masking according to p_{imp} with an additional probability p_d of pixels being considered neither in network input nor in the loss calculation.

The original S2S approach samples disjoint input and target masks. To replicate this behavior in our generalization, for each pixel j we have to handle four separate cases during sampling with their respective probabilities:

$$\Pr(j \in \mathcal{M}_i \wedge j \in \mathcal{M}_t) = p_b = p_{\text{imp}} \tag{20}$$

$$\Pr(j \in \mathcal{M}_i \wedge j \notin \mathcal{M}_t) = p_i = p_m \cdot p_d \cdot (1 - p_{\text{imp}}) \tag{21}$$

$$\Pr(j \notin \mathcal{M}_i \wedge j \in \mathcal{M}_t) = p_t = (1 - p_m) \cdot p_d \cdot (1 - p_{\text{imp}}) \tag{22}$$

$$\Pr(j \notin \mathcal{M}_i \wedge j \notin \mathcal{M}_t) = p_n = 1 - (p_b + p_i + p_t). \tag{23}$$

Note that setting $p_{\text{imp}} = 0$ and $p_d = 1$ yields the original S2S algorithm.

To ensure that the network is able to overfit high importance pixels, we also modify the dropout used in the decoder layers to use a modified dropout weight

$$\hat{p}_{\text{dropout}} = p_{\text{imp}} \cdot 0 + (1 - p_{\text{imp}}) \cdot p_{\text{dropout}} \tag{24}$$

per neuron. For inner decoder layers with lower resolution, downsampled importance images are used accordingly.

4. Results

4.1. Test data

We evaluate the potential of our guided denoising approach using the diffuse textures of 14 different SVBRDFs produced by the fitting network of Merzbach et al. [34]. The measurements

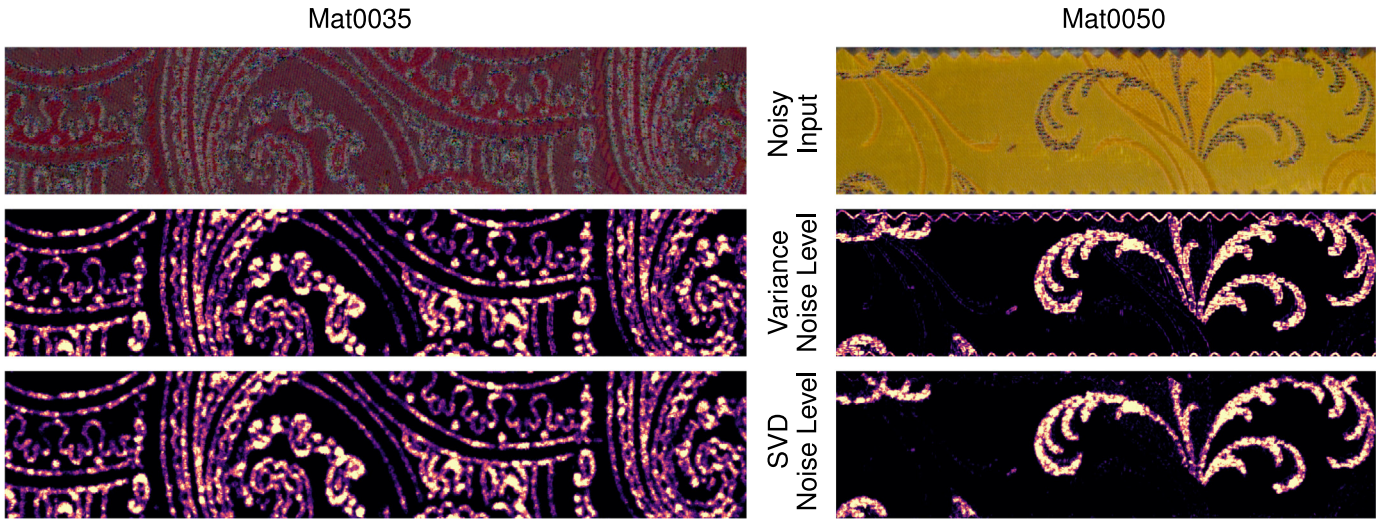

Fig. 6. Estimated noise level for two different images (top row) containing spatially concentrated noise. Using only the patchwise variance (middle row) erroneously yields high values for discontinuities in the image. Conducting the SVD-based analysis of the patches (bottom row) helps to filter out these errors.

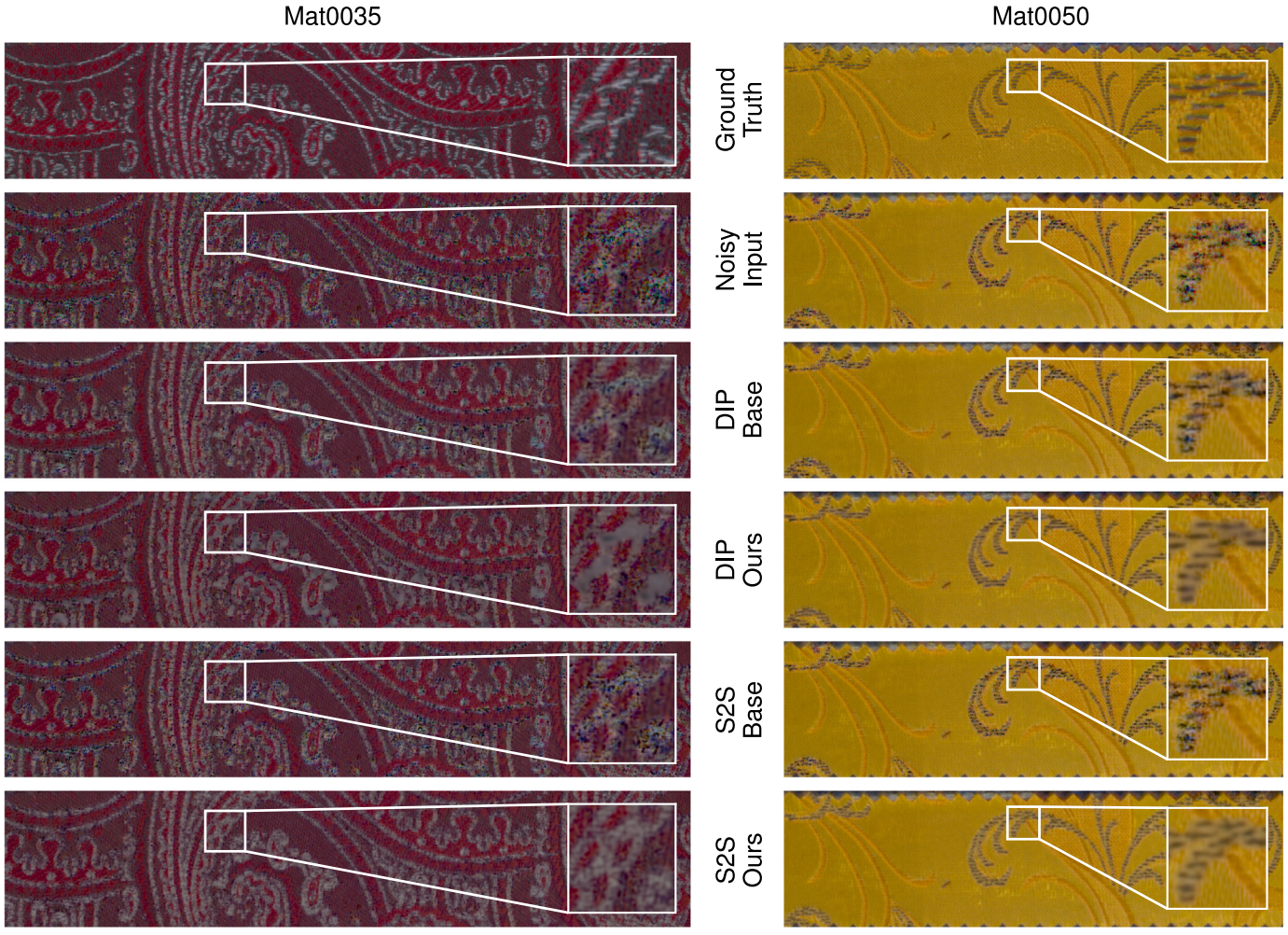

Fig. 7. Comparison of the results of the original DIP and S2S approaches with our modified variants. First row: Diffuse texture of the Pantora SVBRDF fit. Second row: Network-fitted textures used as input for all tested algorithms. Third and fifth row: Original denoising algorithms DIP and S2S. Fourth and sixth row: Out modified versions of the DIP and S2S algorithms using SVD-based guidance images.

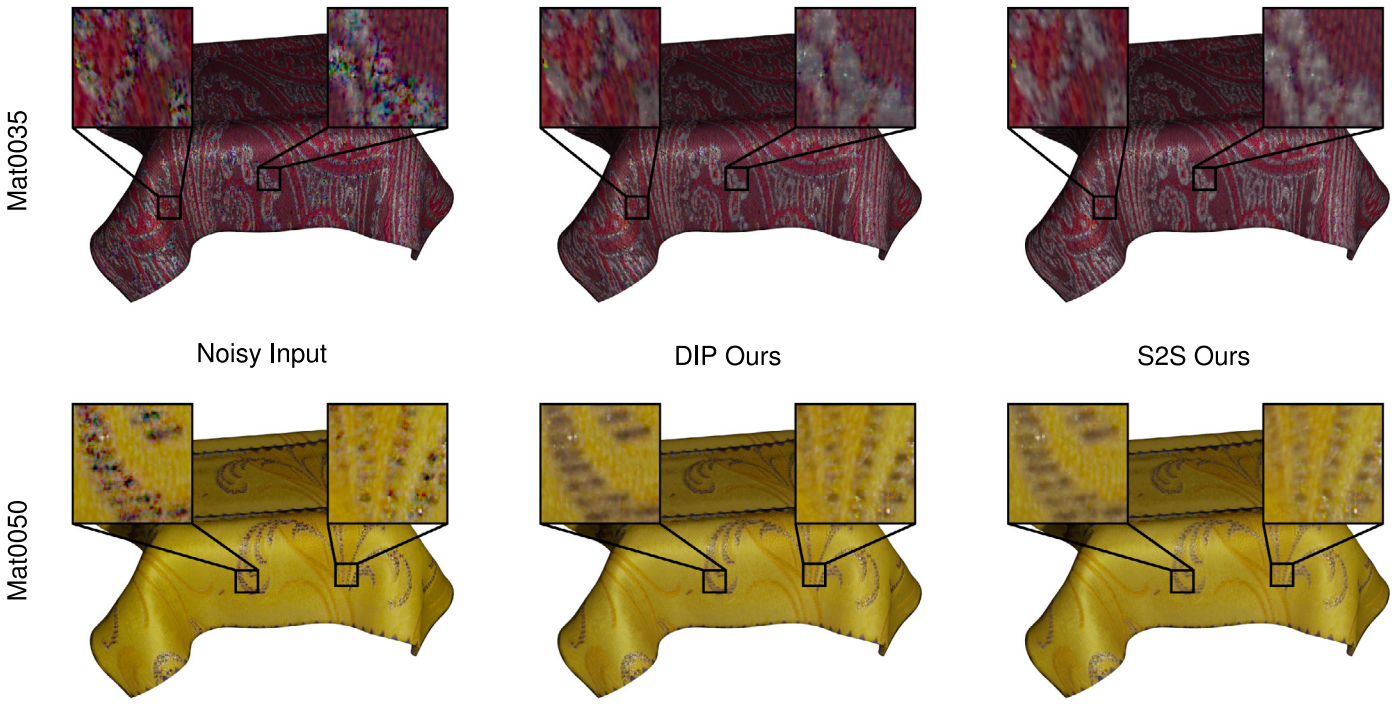

Fig. 8. Renderings of two different SVBRDFs fitted by the fitting network of Merzbach et al. [34] without denoising (left), denoised diffuse texture using our guided DIP variant (middle) and denoised diffuse texture using our guided S2S variant (right).

for all of these materials are publically available in the UBOFAB19 database [34].

The UBOFAB19 database uses the Geisler-Moroder variant [35] of the Ward BRDF [36] with Schlick's Fresnel approximation term as this model is expressive enough for a large variety of real-world materials. The model is parameterized based on the shading normal $\mathbf{n}_s \in \mathbb{R}^3$, the diffuse albedo $\mathbf{a}_d \in \mathbb{R}^3$, the specular albedo $\mathbf{a}_s \in \mathbb{R}^3$, the lobe roughness parameters $\sigma_x \in \mathbb{R}$ and $\sigma_y \in \mathbb{R}$, the anisotropy angle $\alpha \in \mathbb{R}$ and the 0-inclination reflection coefficient $F_0 \in \mathbb{R}$. We only apply our restoration process to the diffuse textures, since these are responsible for most of the artifacts in the final renderings. The diffuse textures of Pantora [37] SVBRDF fits as depicted in the first row of Fig. 7 are considered to be the ground truth as they are also used as labels for training the fitting network of Merzbach et al. Note, however, that the SVBRDFs fitted by the network do not only contain a high amount of noise but are also very likely to be biased. Therefore, we cannot expect to achieve perfect results indistinguishable from ground truth using only image restoration methods.

4.2. Noise level estimation

Resulting estimated noise levels of two different textures using the naive patchwise variance and the more sophisticated SVD-based approach are shown in Fig. 6. Both methods successfully assign high noise levels to actually noisy regions in the image. While the naive approach already performs well, the SVD-based algorithm reduces unwanted high values at discontinuities significantly. This is clearly visible in the upper regions of the yellow image, where the abrupt transition of yellow to gray pixels yields high patchwise variance values but low SVD-based noise levels as the individual subsets $\mathcal{N}_{\text{upper}}$ and $\mathcal{N}_{\text{lower}}$ can be approximated well by a plane. As the SVD-based noise level estimation performs better than the naive method without meaningful disadvantages, we are using the former for all denoising experiments.

Table 1
Quantitative comparison of several denoising approaches on a set of 14 diffuse textures of network fitted SVBRDFs using the diffuse textures of the respective Pantora fits as ground truth.

Algorithm	PSNR (↑)	SSIM (↑)
Input	25.9294	0.5437
CVF-SID [5]	27.8025	0.6085
DIP-Base [1]	26.8042	0.5846
DIP-Ours	27.5713	**0.6265**
S2S-Base [4]	27.4815	0.6149
S2S-Ours	**27.8747**	0.6197

4.3. Denoising

Fig. 7 depicts denoising results on two different textures. The network fitted textures contain strong noise artifacts especially in shiny areas of the captured fabric. Both, DIP and S2S, fail to remove these artifacts in a satisfactory manner, while at the same time blurring out the clean structure of the fabric. The modified algorithms are able to produce images which are mostly clean of colorful artifacts, but for the red fabric, our DIP variant seems to remove more details than necessary. More fine details are preserved using the guided S2S approach. On the yellow fabric, both of our algorithms produce similar results clearly outperforming their original counterpart respectively. The strong denoising effect of our guided approaches can also be seen in the rendered SVBRDFs in Fig. 8. The results rendered with denoised diffuse albedo textures are containing much less disturbing colorful artifacts.

A quantitative comparison can be found in Table 1 comparing our guided denoising methods with CVF-SID [5], DIP [1] and S2S [4] on a dataset of 14 different images. We are calculating PSNR and mean SSIM for all images and average the respective values. For both images, higher values are better. Our guided denoising algorithms not only outperform their original complement, but also perform slightly better than another state-of-the-art learning-based denoising algorithm.

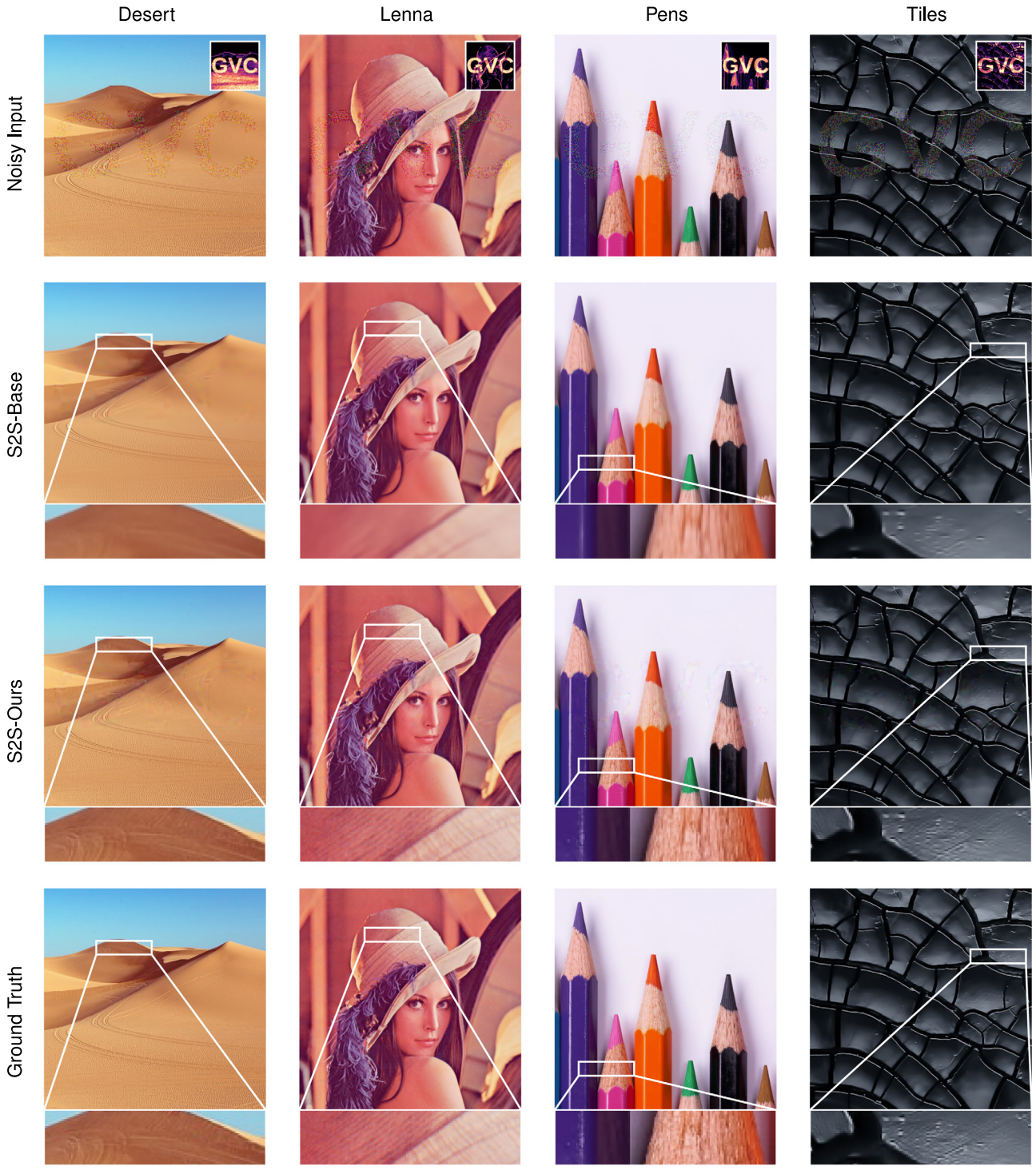

Fig. 9. Comparison of S2S-Base (second row) and S2S-Ours (third row) on three natural images and one texture corrupted with spatially concentrated white noise (first row, inset shows guidance map used for S2S-Ours). We used $p_d = 1$ for these experiments.

Additional results on natural images and a completely different texture are depicted in Fig. 9. We used $p_d = 1$ for these experiments. In contrast to the original S2S method, our guided approach is able to preserve fine details in clean pixels.

4.4. Limitations

While being able to distinguish between clean and noisy image parts well in our examples, it is easy to construct artificial scenarios in which our SVD-based noise level estimation fails. However, our results suggest, that it should work well in practice as we can fallback to the natural regularization capabilities of the denoising methods.

We are able to adaptively denoise a partially noisy image using our guided denoising algorithms and therefore are able to overcome some of the original approach's limitations, but other problems with the respective approaches remain untouched. As the original DIP, our guided version is still relying on hyperparameter tuning. Despite working out for our examples, the optimal number of training iterations and the choice of w^i_{dec}

control parameters might not be the same for every noisy input image, which might also be the reason for oversmoothing of noisy regions for the red fabric in Fig. 7. Similarly, independent of being guided or not, S2S has to be tuned to the variance of the expected noise since it was not able to remove strong noise out of the box.

Finally, Fig. 9 suggests, that the guided algorithms might produce slightly stronger artifacts in noisy pixels in comparison to the original approaches for some images. However, depending on the use-case, this is preferable over loss of details in clean pixels.

5. Conclusion and future work

In this work, we have shown the limitations of off-the-shelf denoising algorithms regarding their capability of handling images which contain location-dependent noise-like artifacts. We proposed a novel method for detecting such noisy pixels and utilizing this additional information to guide state-of-the-art learning-based denoising approaches with only minor modifications. Depending on the nature of the underlying denoising approach, the generated guidance images can be used to either stop the training process early for parts of the image while continuing the training process in others, or it can be used to guide stochastic regularization approaches. By incorporating this additional guidance information, the resulting denoising algorithms were able to beat their original counterparts as well as outperform another state-of-the-art denoising algorithm.

Since our results suggest that other denoising algorithms could benefit from our guidance information in a similar manner, this should be tested as part of future research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Funding

This work was supported by the DFG, Germany-Project KL 1142/9-2.

References

[1] Ulyanov D, Vedaldi A, Lempitsky V. Deep image prior. In: CVPR. 2018, p. 9446–54.
[2] Mataev G, Milanfar P, Elad M. DeepRED: Deep image prior powered by RED. In: ICCV workshops. 2019.
[3] Liu J, Sun Y, Xu X, Kamilov US. Image restoration using total variation regularized deep image prior. In: International conference on acoustics, speech and signal processing. IEEE; 2019, p. 7715–9.
[4] Quan Y, Chen M, Pang T, Ji H. Self2Self with dropout: Learning self-supervised denoising from single image. In: CVPR. 2020, p. 1890–8.
[5] Neshatavar R, Yavartanoo M, Son S, Lee KM. CVF-sid: Cyclic multi-variate function for self-supervised image denoising by disentangling noise from image. In: CVPR. 2022, p. 17583–91.
[6] Chambolle A. An algorithm for total variation minimization and applications. J Math Imaging Vision 2004;20(1):89–97.
[7] Osher S, Burger M, Goldfarb D, Xu J, Yin W. An iterative regularization method for total variation-based image restoration. Multiscale Model Simul 2005;4(2):460–89.
[8] Zoran D, Weiss Y. From learning models of natural image patches to whole image restoration. In: ICCV. IEEE; 2011, p. 479–86.
[9] Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans Image Process 2006;15(12):3736–45.
[10] Buades A, Coll B, Morel JM. A non-local algorithm for image denoising. In: CVPR, Vol. 2. IEEE; 2005, p. 60–5.
[11] Dabov K, Foi A, Katkovnik V, Egiazarian K. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans Image Process 2007;16(8):2080–95.
[12] Foi A, Katkovnik V, Egiazarian K. Pointwise shape-adaptive DCT denoising with structure preservation in luminance-chrominance space. In: Int. workshop on video processing and quality metrics for consumer electronics. 2006.
[13] Dabov K, Foi A, Katkovnik V, Egiazarian K. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In: Int. conference on image processing, Vol. 1. IEEE; 2007.
[14] Mairal J, Elad M, Sapiro G. Sparse representation for color image restoration. IEEE Trans Image Process 2008;17(1):53–69.
[15] Rajwade A, Rangarajan A, Banerjee A. Image denoising using the higher order singular value decomposition. TPAMI 2012;35(4):849–62.
[16] Miyata T. Inter-channel relation based vectorial total variation for color image recovery. In: International conference on image processing. IEEE; 2015, p. 2251–5.
[17] Zhang K, Zuo W, Gu S, Zhang L. Learning deep CNN denoiser prior for image restoration. In: CVPR. 2017, p. 3929–38.
[18] Yang X, Xu Y, Quan Y, Ji H. Image denoising via sequential ensemble learning. IEEE Trans Image Process 2020;29:5038–49.
[19] Quan Y, Chen Y, Shao Y, Teng H, Xu Y, Ji H. Image denoising using complex-valued deep CNN. Pattern Recognit 2021;111:107639.
[20] Pang T, Zheng H, Quan Y, Ji H. Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising. In: CVPR. 2021, p. 2043–52.
[21] Kattamis A, Adel T, Weller A. Exploring properties of the deep image prior. In: NeurIPS 2019. 2019, Posters.
[22] Cheng Z, Gadelha M, Maji S, Sheldon D. A bayesian perspective on the deep image prior. In: CVPR. 2019, p. 5443–51.
[23] Tölle. M, Laves MH, Schlaefer A. A mean-field variational inference approach to deep image prior for inverse problems in medical imaging. In: Proceedings of machine learning research. 2021.
[24] Heckel R, Hand P. Deep decoder: Concise image representations from untrained non-convolutional networks. In: ICLR. 2019.
[25] Chen YC, Gao C, Robb E, Huang JB. Nas-dip: Learning deep image prior with neural architecture search. arXiv preprint arXiv:2008117132020.
[26] Ho K, Gilbert A, Jin H, Collomosse J. Neural architecture search for deep image prior. arXiv preprint arXiv:2001047762020.
[27] Sidorov O, Yngve Hardeberg J. Deep hyperspectral prior: Single-image denoising, inpainting, super-resolution. In: ICCV workshops. 2019.
[28] Wang L, Sun C, Fu Y, Kim MH, Huang H. Hyperspectral image reconstruction using a deep spatial–spectral prior. In: CVPR. 2019, p. 8032–41.
[29] Gadelha M, Wang R, Maji S. Shape reconstruction using differentiable projections and deep priors. In: ICCV. 2019, p. 22–30.
[30] Williams F, Schneider T, Silva C, Zorin D, Bruna J, Panozzo D. Deep geometric prior for surface reconstruction. In: CVPR. 2019, p. 10130–9.
[31] Hanocka R, Metzer G, Giryes R, Cohen-Or D. Point2Mesh: A self-prior for deformable meshes. arXiv preprint arXiv:2005110842020.
[32] Chen M, Quan Y, Pang T, Ji H. Nonblind image deconvolution via leveraging model uncertainty in an untrained deep neural network. Int J Comput Vis 2022;1–20.
[33] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Int. conference on medical image computing and computer-assisted intervention. Springer; 2015, p. 234–41.
[34] Merzbach S, Hermann M, Rump M, Klein R. Learned fitting of spatially varying BRDFs. In: Comp. graph. forum. Wiley Online Library; 2019, p. 193–205.
[35] Geisler-Moroder D, Dür A. A new ward BRDF model with bounded albedo. In: Comp. graph. forum,Vol. 29. Wiley Online Library; 2010, p. 1391–8.
[36] Ward GJ. Measuring and modeling anisotropic reflection. In: Proceedings of the 19th annual conference on computer graphics and interactive techniques. 1992, p. 265–72.
[37] X-RITE. Pantora material hub. 2019, https://web.archive.org/web/20190424232441/https://www.xrite.com/categories/appearance/pantora-software.