

Learning Joint Demosaicing and Denoising Based on Sequential Energy Minimization

Teresa Klatzer¹

klatzer@icg.tugraz.at

Kerstin Hammernik¹

hammernik@icg.tugraz.at

Patrick Knöbelreiter¹

knoebelreiter@icg.tugraz.at

Thomas Pock^{1,2}

pock@icg.tugraz.at

¹Institute for Computer Graphics and Vision
Graz University of Technology

²Digital Safety and Security Department
AIT Austrian Institute of Technology

Abstract

Demoslicing is an important first step for color image acquisition. For practical reasons, demosaicing algorithms have to be both efficient and yield high quality results in the presence of noise. The demosaicing problem poses several challenges, e.g. zippering and false color artifacts as well as edge blur. In this work, we introduce a novel learning based method that can overcome these challenges. We formulate demosaicing as an image restoration problem and propose to learn efficient regularization inspired by a variational energy minimization framework that can be trained for different sensor layouts. Our algorithm performs joint demosaicing and denoising in close relation to the real physical mosaicing process on a camera sensor. This is achieved by learning a sequence of energy minimization problems composed of a set of RGB filters and corresponding activation functions. We evaluate our algorithm on the Microsoft Demosaicing data set in terms of peak signal to noise ratio (PSNR) and structured similarity index (SSIM). Our algorithm is highly efficient both in image quality and run time. We achieve an improvement of up to 2.6 dB over recent state-of-the-art algorithms.

1. Introduction

High quality and fast joint demosaicing and denoising algorithms are essential for practical application on modern camera hardware. To capture a photograph, modern digital cameras record light intensities from the scene on a *single* charge-coupled-device (CCD) or complementary metal-oxide-semiconductor (CMOS) sensor. Independent of the sensor type, each sensor element is able to record the intensities of either R, G or B color channel according to the pattern of a color filter array (CFA). The filter layout

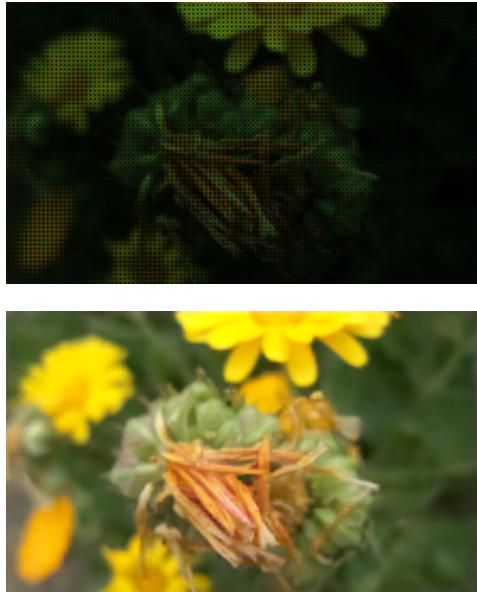


Figure 1. Top row: Noisy mosaiced input image in linRGB space. Bottom row: Output of our algorithm on the noisy input data, demosaiced in linRGB space and then transformed to sRGB space. The input image is taken from the Microsoft Demosaicing data set by Khashabi *et al.* [2].

varies according to the camera type. The most frequently used CFA is the Bayer CFA [1] which contains twice as much green pixels as red and blue pixels. This arrangement mimics the human visual system which is most responsive to green color. Using the single channel RAW image data and the specific CFA layout, a couple of operations have to be performed to generate a full color image since part of the pixel information is missing. This process to fill in the missing pixel values is called demosaicing.

Several problems have to be overcome during the demosaicing process. Simple interpolation techniques work

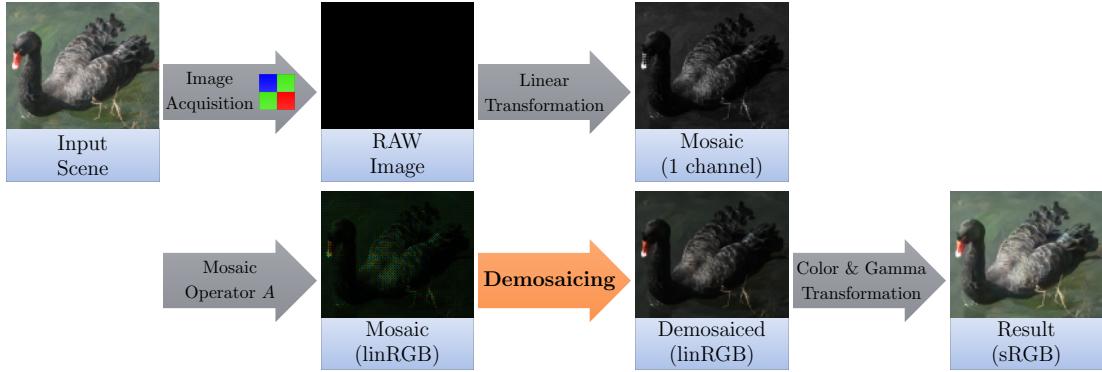


Figure 2. Illustration of the camera pipeline. The upper row shows the pre-processing steps applied on the RAW image (linear color scaling and black level correction) until the single channel mosaic image is acquired by the camera. The lower row shows the color mosaic image in linear RGB (linRGB) space, obtained through application of the mosaic operator A . In this space, demosaicing is performed. After demosaicing, the image is color transformed and gamma corrected and finally an image in standard RGB (sRGB) space is produced. The evaluation of our algorithm is done in both, linRGB and sRGB space.



Figure 3. Left: 2×2 Bayer CFA pattern, right: 6×6 Fujifilm Xtrans CFA pattern.

well in homogeneous regions of an image, but interpolating corners and edges is very challenging especially when inter-channel correlations are ignored. False color and zipper artifacts are common effects that occur due to the spatial offset of R, G and B pixels. Another problem is edge blur arising from commonly applied low-pass interpolation filters. Demosaicing the color channels separately or sequentially leads to severe error propagation, therefore alternating or iterative algorithms are preferred. Existing algorithms exploit correlations between the color channels to obtain better results. Another challenge for demosaicing is the noise arising from the image acquisition process on the camera sensor which is in fact not only Gaussian [3]. Therefore, an important property of a demosaicing algorithm is not only accurate reconstruction of missing pixels, but also removal of present noise.

We identify a general problem in the design of demosaicing algorithms: In many approaches, the algorithms are evaluated on already processed reference images that are artificially mosaiced again. Recent work by Khashabi *et al.* [2] proposed a fundamentally different approach to tackle the demosaicing problem by providing a novel method to produce realistic training and ground truth images for demosaicing research. Their work is based on the imaging pipeline depicted in Fig. 2: The RAW image that has to be demosaiced is present in linear RGB (linRGB) space, and only after demosaicing the images are fully developed into standard RGB (sRGB) space via color transformation and gamma correction. The specific CFA pattern

(see Fig. 3) is encoded in the mosaic operator A that maps the intensity values to color values according to the CFA. Khashabi *et al.* provide this data in the publicly available Microsoft Demosaicing data set [2] which is the basis of our work. This data set also contains noisy training data with camera noise according to the model proposed in [3].

Demosaicing is an essential step in processing images in cameras and thus well studied. Many approaches exist that attempt to solve it, still, no method has been capable of solving the problem completely. Most algorithms are designed specifically for a single CFA pattern (*i.e.* Bayer CFA), trying to interpolate the missing pixels. For extensive reviews on demosaicing methods see [4, 5]. Many algorithms exploit correlation between the color channels. Common assumptions are that color differences or color ratios are constant between the channels [6–8]. If the assumptions do not hold, zipper artifacts as well as false color artifacts appear at object boundaries. Other approaches consider additionally the edge directions for interpolation along an edge rather than across [8–15]. Additionally to approaches that use local edge information, methods in the spirit of the non-local means algorithm exploit self-similarity and redundancy of natural images [16–19]. Successful approaches do not only consider sequential interpolation, but some sort of alternating refinement strategies or additional post-processing to get rid of artifacts. Some of the previously mentioned works have been extended to handle demosaicing and denoising jointly, but mostly in the pure Gaussian setting [20–23].

There exist also learning based approaches to tackle the demosaicing problem. A very early work used artificial neural networks [24], other works are based on Support Vector Regression [25] or Markov Random Fields [26]. Also, dictionary learning approaches exist [27] with extensions to exploit self-similarity [28]. Khashabi *et al.* propose a machine learning method for joint demosaicing and de-

noising based on Regression Tree Fields [2].

Another important group of demosaicing algorithms form reconstruction approaches that view demosaicing as an inverse problem [5]. An inverse problem tries to find an estimate u^* of the original image g given observed data m corrupted by noise n :

$$m = Ag + n \rightarrow u^* = A^{-1}(m - n). \quad (1)$$

The linear operator A models the physical relationship between the observed data and the original image. This problem is inherently ill-posed, therefore regularization plays a crucial role [29] for the solution, *i.e.*

$$u^* = \arg \min_u \mathcal{R}(u) + \frac{\lambda}{2} \|Au - m\|_2^2. \quad (2)$$

Via a regularization term \mathcal{R} , prior information about natural images is encoded to compensate for the missing information, similar to statistical models. Different types of regularization have been explored in the literature, including Total Variation (TV) regularization on color differences [30] and inter- and intra-channel smoothness of color differences [31]. A combination of TV and BM3D [32] regularizers was exploited by [33] for joint demosaicing and denoising. We argue that handcrafted regularization is not able to capture natural image statistics well enough, therefore we propose to learn a suitable regularization term from training data.

Our approach combines ideas from learning based approaches and reconstruction approaches. Given training data, we learn how to optimally transform a noisy single channel mosaic image captured by the camera to a full color image by filling the missing color information accordingly. We train a powerful regularizer based on a sequential energy minimization procedure. With our approach we address several previously discussed challenges. Contrary to many works in demosaicing research, we perform demosaicing in a linRGB space, and propose a method that can be trained with different types of CFA patterns and camera types. Our method performs joint denoising and demosaicing under non-Gaussian camera noise that is present in every camera. It does not rely on handcrafted correlation assumptions, but learns the image statistics to produce natural images. An example of a noisy, mosaiced image can be seen in Fig. 1 as well as its demosaiced and denoised version using our proposed algorithm below.

Our main contributions are as follows:

- We extend the learning method presented in Chen *et al.* [34] for restoring color images in the presence of noise. We establish this method in a new application domain and show that it is superior to current state-of-the-art methods.
- We approach the demosaicing problem in a realistic setting by performing demosaicing in linRGB space.

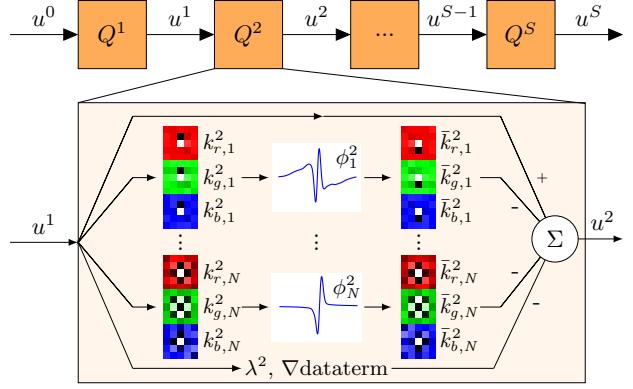


Figure 4. Illustration of the training scheme. The upper part shows the sequence of gradient steps Q^s to obtain a demosaiced image u^S from the initial input image u^0 . The internals of each step are shown below.

- Once trained for a specific camera model and CFA pattern, the algorithm can be efficiently implemented on hardware and directly applied on the camera chip. The model consists only of linear convolutions and the application of non-linear point-wise activation functions similar to a neural network.
- We provide insights to the learned model parameters that are specific for demosaicing.

2. Proposed Method

We view the demosaicing problem as an image restoration problem where missing color information has to be inferred from existing data points respecting natural image statistics. We approach this task from a variational perspective, where we define a dynamic energy function to solve the inverse problem. We formulate the demosaicing process as a sequence of energy minimization problems where the model parameters are optimized in an offline discriminative training scheme. The form of the energy function builds on the approach of Chen *et al.* [34] for learning an optimized diffusion process that was designed for denoising. We provide an alternative motivation, draw connections to multi-level learning and propose several extensions to the basic model. First, the model is extended for color image restoration, and second, we add the flexibility to adapt the data term for each step. In Fig. 4, an illustration of the method is shown. The upper part depicts the sequence Q^1, \dots, Q^S of optimized quadratic energy functions. The input to the algorithm is a linRGB mosaic image u^0 with missing color information that is computed from the single channel mosaic image m , and the output is a demosaiced image u^s , $s = 1, \dots, S$, after the application of S energy minimization steps. The internals of each step are shown in the lower part of the figure. In the following, the method is explained in more detail.

2.1. A multilevel view on the demosaicing problem

In a bi-level optimization approach [35,36], we typically have a higher level loss function and a lower level energy minimization problem. In our approach, we consider not a single lower level problem, but a sequence of energy minimization problems, therefore we name it multilevel optimization. For the demosaicing problem, we define the loss function for demosaiced color images $u_l^S \in \mathbb{R}^{3HW}$ and the corresponding ground truth images $g_l \in \mathbb{R}^{3HW}$, with l the sample index of the image, as follows:

$$\mathcal{L}(u_l^S, g_l)_{l=1}^L = \frac{1}{2} \sum_{l=1}^L \|u_l^S - g_l\|_2^2 \quad (3)$$

where H and W are the height and width of the image, respectively. The image u_l^S is the output of the sequence of lower level energy minimization problems for one training example. Therefore, the image u_l^S is a demosaiced image in linRGB space, and can be compared with g_l , the corresponding ground truth image in linRGB space.

Alternatively, the loss function can be evaluated in sRGB space. For this reason, we must first apply the color transformation and a gamma correction function to the images in linear space to transform them to sRGB space. We denote the transformation function $c(\cdot)$, and the loss function reads as follows:

$$\mathcal{L}_{sRGB}(u_l^S, g_l)_{l=1}^L = \frac{1}{2} \sum_{l=1}^L \|c(u_l^T) - c(g_l)\|_2^2. \quad (4)$$

The transformation function $c(\cdot)$ applied to a linRGB image u_{lin} follows the description in [2] and is defined as

$$c(u_{lin}) = \gamma(M u_{lin}) - b \quad (5)$$

with $\gamma(\cdot)$ the mean gamma correction curve as given in the Microsoft Demosaicing data set [2], M a color transformation operator that maps the color values from linRGB to sRGB space $M : \mathbb{R}^{3HW} \rightarrow \mathbb{R}^{3HW}$, and b the black correction constant. All variables are given according to the camera model used for producing the data set and can be customized if new training data with different camera models are created using the approach in [2]. In what follows, we drop the sample indices of u_l^S for simplicity.

The given loss function is our higher level objective. The aim of training is to find the parameters θ that minimize the loss function

$$\min_{\theta} \mathcal{L}(u^S(\theta), g) \quad (6)$$

which is subject to a sequence of $s = 1, \dots, S$ energy func-

tions Q being minimized on the training samples u

$$\begin{aligned} u^1 &= \arg \min_u Q^1(u, u^0, m, \theta^1) \\ u^2 &= \arg \min_u Q^2(u, u^1, m, \theta^2) \\ &\vdots \\ u^{S-1} &= \arg \min_u Q^{S-1}(u, u^{S-2}, m, \theta^{S-1}) \\ u^S &= \arg \min_u Q^S(u, u^{S-1}, m, \theta^S). \end{aligned} \quad (7)$$

The variables u^s , $s = 1, \dots, S$ represent the intermediate solutions of the sequential energy minimization procedure. Each intermediate solution u^s is dependent on the previous solution u^{s-1} , the initial single channel mosaic image m and the model parameters θ^s . The variable θ^s is a placeholder for all parameters of the respective energy function.

So far, the approach resembles a bi-level optimization framework, where the aim is to obtain the optimal parameters θ of the lower level energy minimization problem that result in a minimal higher level (overall) loss of the training problem. In our case, we have a multilevel problem, because the lower level problem consists of a sequence of energy minimization problems. Next, we define the concrete form of the energy minimization problems.

2.2. The sequential energy minimization model

Let us define a sequence of quadratic lower level energy functions Q^s . We define the lower level problem Q^s as

$$\begin{aligned} Q^s(u, u^{s-1}) &= f(u^{s-1}) + \\ &\langle u - u^{s-1}, \nabla f(u^{s-1}) \rangle + \frac{1}{2} \|u - u^{s-1}\|_2^2 \end{aligned} \quad (8)$$

which is the linearization of a differentiable function f at u^{s-1} plus a proximal regularization term [37]. For simplicity we dropped the dependency of the function Q^s on the parameters θ^s and m . We can minimize Q^s in closed form:

$$\begin{aligned} u^s &= \arg \min_u Q^s(u, u^{s-1}) \\ u^s &= u^{s-1} - \nabla f(u^{s-1}). \end{aligned} \quad (9)$$

The result in Eq. 9 is a simple gradient descent, hence the sequence of quadratic optimization problems in Eq. 7 reduces to a fixed number S of gradient descent steps on the parametrized quadratic energies $f(u, u^{s-1}, m, \theta^s)$ that adapt to the current progress of the algorithm. Through learning, we optimize the parameters θ^s of the gradient steps ∇f for each update of the input image u^s .

The optimal model parameters θ^s for each gradient step of the sequence $s = 1, \dots, S$ are obtained via standard back-propagation [38]. The loss function is evaluated after S gradient steps. The gradient of the loss function to the parameters θ^s for each u^s are then computed as follows:

$$\frac{\partial \mathcal{L}(u^S, g)}{\partial \theta^s} = \frac{\partial \mathcal{L}(u^S, g)}{\partial u^S} \cdot \frac{\partial u^S}{\partial u^{S-1}} \cdots \frac{\partial u^{s+1}}{\partial u^s} \cdot \frac{\partial u^s}{\partial \theta^s}. \quad (10)$$

In the next subsection, we will elaborate on the specific form of the energy function for the demosaicing problem.

2.3. Energy function for the demosaicing problem

We view the demosaicing problem as an inverse image restoration problem where missing pixels have to be determined based on the underlying image statistics. Inspired by a variational approach, we choose the function f as follows:

$$f(u, m, \theta) = \mathcal{R}(u, \theta) + \mathcal{D}(u, m, \theta) \quad (11)$$

which is the standard approach for solving inverse problems in image processing. This function consists of a regularization term \mathcal{R} , applied on the image $u \in \mathbb{R}^{3HW}$, and a data fidelity term \mathcal{D} that measures the similarity of the initial single channel mosaic image $m \in \mathbb{R}^{HW}$ to u , both parametrized with parameters θ . The regularization part is essential for performance of our algorithm and encodes the low-level image statistics.

The form of the regularizer we use is inspired by the Field of Experts prior by Roth and Black [39] and is defined as

$$\mathcal{R}(u, \theta) = \sum_{i=1}^N \sum_{p=1}^{HW} \rho_i((K_i u)_p) \quad (12)$$

with

$$(K_i u)_p = \sum_{c \in \{r,g,b\}} (k_{c,i} * u)_p \quad (13)$$

where on the right hand side $k_{c,i} * u$ denotes the 2D convolution. The penalty functions ρ_i , $i = 1, \dots, N$ are fully trainable functions and are applied point-wise on the filtered image. Each channel u_c of the linRGB image $u \in \mathbb{R}^{3HW}$ is convolved with one channel $k_{c,i}$ of the RGB kernels $k_i \in \mathbb{R}^{3K^2}$ and the result is summed over all channels to exploit inter-channel dependencies. The data term models the data fidelity and incorporates the physical process of demosaicing via the operator A :

$$\mathcal{D}(u, u_0, \theta) = \frac{\lambda}{2} \|Au - m\|_2^2 \quad (14)$$

with $A : \mathbb{R}^{3HW} \rightarrow \mathbb{R}^{HW}$ that maps the linRGB image u to the mosaic space according to the CFA sensor layout and a weighting parameter λ . Note that this term corresponds to the right hand side in Eq. 2.

Revisiting the update rule from Eq. 9, we define the final model and compute the gradient as

$$\nabla f^s(u^{s-1}, m, \theta^s) = \nabla \mathcal{R}^s(u^{s-1}, \theta^s) + \nabla \mathcal{D}^s(u^{s-1}, m, \theta^s). \quad (15)$$

Setting $\rho'_i(\cdot) = \phi_i(\cdot)$ which we call activation functions the derivative of the regularization term \mathcal{R} writes for each color channel c

$$\nabla \mathcal{R}_c^s(u^{s-1}, \theta^s) = \sum_{i=1}^N K_{c,i}^{sT} \text{vec} (\phi_i^s(K_i^s u^{s-1})_p)_{p=1}^{HW} \quad (16)$$

with $K_{c,i}^{sT}$ the convolution operator for one channel of a RGB kernel rotated by 180° (equivalent to $\bar{k}_{c,i}^s$ in Fig. 5). The operator vec vectorizes the argument pixel-wise by stacking the elements in a column vector.

The derivative of the data term is

$$\nabla \mathcal{D}^s(u^{s-1}, m, \theta^s) = \lambda^s A^T (Au^{s-1} - m). \quad (17)$$

The gradient scheme from Eq. 9 is therefore

$$u^s = u^{s-1} - \nabla \mathcal{R}^s(u^{s-1}, \theta^s) - \nabla \mathcal{D}^s(u^{s-1}, m, \theta^s). \quad (18)$$

as illustrated in the lower part of Fig. 4. The activation functions $\phi_i^s(\cdot)$ are modeled using radial basis functions (RBFs) and have the following form:

$$\phi_i^s(z, w) = \sum_{j=1}^M w_{ij}^s \exp \left(-\frac{(z - \mu_j)^2}{2\sigma^2} \right) \quad (19)$$

with mean values μ_j and standard deviation σ . To sum up, the parameter vector θ^s holds all parameters that define the model for gradient step s , that are the kernels k_i^s , the activation functions ϕ_i^s with weights w_{ij}^s and the weighting factor for the data term λ^s . All these parameters are optimized during training.

2.4. Adapting the data term

The data term as we described it in Eq. 14 implicitly assumes Gaussian distributed noise. Following the analysis of Foi *et al.* [3], this is not entirely true for the demosaicing problem. According to them, the noise in RAW images is mostly due to shot and read noise occurring on a camera sensor. Foi *et al.* categorized these types of noise into a mixture of Poisson and Gaussian distributed noise and propose a method for generating synthetic noise of this type. Their algorithm is also used for generating the noisy images of the Microsoft Demosaicing data set [2]. We conclude that the Gauss assumption does not hold for the data term when trained on the noisy data, and therefore we add the possibility to learn the exact function by our training algorithm. We name the alternative data term \mathcal{F}

$$\mathcal{F}(u, u_0, \theta) = \lambda \Psi(Au - m). \quad (20)$$

Setting $\Psi'(\cdot) = \psi(\cdot)$, the derivative of the data term is given by

$$\nabla \mathcal{F}^s(u^{s-1}, m, \theta^s) = \lambda^s A^T \psi^s(Au^{s-1} - m) \quad (21)$$

with $\psi^s(\cdot, v)$ analogous to Eq. 19. For the extended model, the parameter vector θ^s holds additionally parameters for the data term, the functions ψ^s with weights v_j^s .

Method	Noise-free Panasonic		Noisy Panasonic				Noisy Canon			
	PSNR (linRGB)	PSNR (sRGB)	PSNR (linRGB)	PSNR (sRGB)	SSIM (linRGB)	SSIM (sRGB)	PSNR (linRGB)	PSNR (sRGB)	SSIM (linRGB)	SSIM (sRGB)
Matlab [40]	35.22	29.92	34.16	27.56	0.966	0.917	36.38	29.10	0.977	0.919
OSAP [15]	38.29	31.07	36.25	29.93	0.966	0.928	39.00	31.95	0.976	0.939
WECD [11]	38.62	31.50	36.51	30.29	0.966	0.932	-	-	-	-
NLM [16]	38.42	32.09	36.55	30.56	0.970	0.939	38.82	32.28	0.980	0.948
DMMSE [10]	38.82	31.71	36.67	30.24	0.967	0.930	39.48	32.39	0.979	0.943
LPA [12]	39.24	32.40	37.00	30.86	0.969	0.938	39.66	32.84	0.980	0.947
CS [14]	39.41	32.89	37.20	31.41	0.972	0.941	39.82	33.24	0.980	0.946
JMCDM [19]	38.28	32.14	37.44	31.35	0.971	0.942	39.49	32.41	0.976	0.932
RTF [2]	39.39	32.40	37.77	31.77	0.976	0.951	40.35	33.82	0.983	0.955
FlexISP [33]	40.00	33.77	38.28	31.76	0.974	0.941	40.71	33.44	0.984	0.949
SEM (Ours)	40.92	34.56	38.93	32.93	0.980	0.960	41.09	34.15	0.985	0.953
SEM+D (Ours)	39.36	33.16	38.17	32.35	0.978	0.956	40.35	33.86	0.983	0.953

Table 1. Demosaicing and denoising results for the Bayer CFA. We tested our algorithm on noise-free and noisy Panasonic and noisy Canon images from the Microsoft Demosaicing data set. The first two columns show results on the test set for the noise-free Panasonic images, the next four columns show results for the noisy Panasonic images, and the last four columns show results on the noisy Canon images (our algorithm is trained on the Panasonic images). For the noise-free images we use 8 and for the noisy images 16 sequential energy minimization steps (SEM). The SEM+D model is trained on the noisy data only and includes learning the data term on 8 steps. All models are trained with filter size $5 \times 5 \times 3$.

3. Experiments and Results

To evaluate our joint demosaicing and denoising method, we report results in terms of peak signal to noise ratio (PSNR) and structured similarity index (SSIM) [41] on the Microsoft Demosaicing data set [2]. We trained our algorithm on 200 training images given in the data set until the PSNR values on the validation set (100 images) increased again. The model was optimized in linRGB space, and evaluated in both linRGB and sRGB space. For optimization we used the popular LBFGS-B algorithm [42]. The reported PSNR values in Tab. 1 and Tab. 2 are the mean over the individual color channel PSNR values, where the log is applied after taking the mean. The final result is computed as the mean over 200 test images. We abbreviate our sequential energy minimization model SEM for easier reference, and the model with additional data term learning from Section 2.4 SEM+D. The number of gradient steps is specified when reporting the concrete results.

In Tab. 1, we present demosaicing and denoising results for the Bayer CFA. We tested our algorithm on Panasonic and Canon images from the Microsoft Demosaicing data set. The result images from the data set were used to reproduce the evaluation results. There are some missing values marked with '-' because the data has not been available in the data set. We provide results for our SEM model trained on noise-free images with 8 steps (column 1-2), and a SEM model trained on noisy images with 16 steps (column 3-10). The SEM+D model was trained with 8 steps and includes learning the data term. This model is trained on

noisy data because adapting the data term only makes sense in the presence of noise. These facts explain the inferior performance on the noise-free data, and on the noisy data due to the lower number of steps. We also report results on noisy Canon images with our model trained on Panasonic images to show the generalization to unseen data. All models trained for the Bayer CFA use a filter size of $5 \times 5 \times 3$. In Tab. 2, we present demosaicing results on the noise-free images with Fujifilm Xtrans CFA. Here we only compare with results from Khashabi *et al.* [2] because most other methods are specifically designed for the Bayer CFA.

Method	PSNR (linRGB)	PSNR (sRGB)
RTF [2]	36.94	30.56
SEM8 (Ours)	38.45	31.96
SEM16 (Ours)	39.60	33.09

Table 2. Demosaicing results on the Fujifilm Xtrans CFA. We compare with the RTF model in terms of PSNR, and show results for trained models with 8 (SEM8) and 16 (SEM16) energy minimization steps with filter size $7 \times 7 \times 3$.

Overall, we outperform all competing methods listed in Tab. 1 and Tab. 2 by a significant margin. The results in sRGB space can still be improved if we optimized the model using the sRGB loss function. The algorithm does not only yield high quality results, but is also very efficient: We demosaic a 132×220 test image using the 16 step SEM

model with filter size $5 \times 5 \times 3$ in 0.11s with the current Python based GPU implementation using the Theano framework [43], and a 5MP image in 8s. The code runs on an Intel Core i7 CPU using a Nvidia GeForce GTX 980TI graphics card. However, this time can be beaten easily by an optimized hardware implementation as our model only consists of convolutions and the point-wise application of activation functions.

To show that our model learns meaningful model parameters, we visualize the learned activation and penalty functions as well as the RGB kernels in Fig. 5. The learned functions show similar forms as reported in [34]. Additionally, some functions show asymmetries (row 2, 5 and 6) due to the up- or down-weighting of color channels. The learned RGB kernels show clearly that relationships between the different color channels are encoded during training. In Fig. 6 we show learned derivatives and corresponding data terms for the first 6 steps of the trained SEM+D model. For this model, the data term was initialized to the derivative of a Huber ℓ_1 norm. If the noise characteristic was Gaussian, the optimal form of the data term would be quadratic. In the first step, the data term stays a ℓ_1 norm, and in the following steps the data term is quadratic. For a higher number of steps the data term gets non-quadratic. This result shows that the Gauss assumption works quite well in practice, but the assumption is not entirely accurate.

We provide also some qualitative results in Fig. 7. The images are taken from both test and validation images of the Microsoft Demosaicing data set [2] and are shown in linRGB space. We compare result images of the top 5 methods (CS, JMCDM, RTF, FlexISP, SEM (Ours), from top to bottom). When inspecting the images, we observe no false colors or zippering artifacts introduced by our method, as well as a superior denoising performance also in very dark areas (results are best viewed in color on screen).

3.1. Implementation details

In the following, we give some implementation details of our algorithm. To obtain the initial images for our training algorithm, we perform single channel interpolation, *i.e.* treating each channel separately and interpolate bilinearly (for the Bayer CFA) or linearly (for the Fujifilm Xtrans CFA). The pixel values at the CFA pixel grid positions are faithfully kept in place, and empty pixels are interpolated to start the optimization closer to the solution. We use 50 to 75 RGB filters per SEM step and initialize them with RGB Discrete Cosine Transform (DCT) basis filters for the Bayer CFA, or randomly for the Fujifilm Xtrans CFA. The results show that our model is not sensitive to kernel initialization, because both models achieve similar results. The activation functions are initialized to the derivative of a Student-t function which has been proven to be a successful regularizer [44,45].

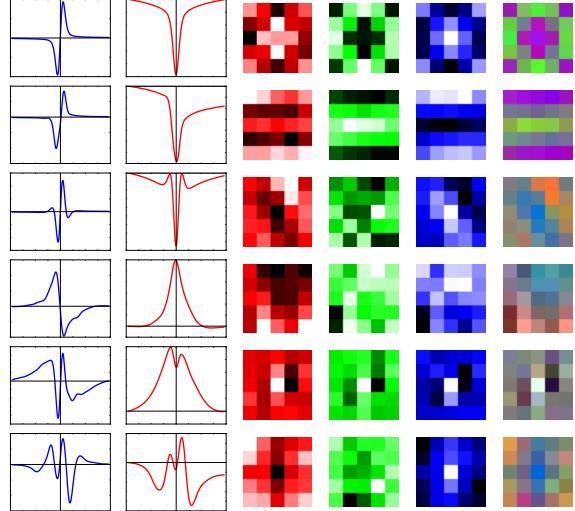


Figure 5. Examples for learned corresponding activation functions (blue), penalty functions (red) and RGB kernels (from left to right) by our sequential energy minimization model. Each line shows the network internals for one step and one specific filter. The filters are shown layer-wise for each color channel and in RGB.

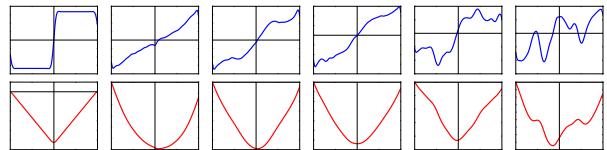


Figure 6. Learned derivative (blue) and corresponding data term (red) for the first 6 steps of the SEM+D model, from left to right.

Through the convolutions of the input images, some artifacts appear at the image boundaries if we do not take care of boundary conditions. We assume in our model symmetric boundary conditions. The border handling was carefully implemented in our model by first padding the image, convolving it in valid mode, zero-padding the result, convolving it with the transpose kernel in full mode, and cropping the resulting image by adding pixels from outside the original image boundary back to the image according to the assumed boundary conditions. This treatment is a key ingredient that only minimal artifacts at the image boundaries are introduced. For further details on implementation and computing gradients we refer the interested reader to the supplemental material of [34].

3.2. Experiments with real RAW images

To prove the practical applicability of our algorithm, we took photographs with a standard consumer camera and tested it on the obtained RAW images. The image shown in Fig. 8 is taken with a Sony A6000, the pre-processing in lin-RGB space was done using the `drawing`¹ software to obtain a single channel mosaic image with black level correction

¹<http://www.cybercom.net/~dc coffin/drawing/>

and color scaling. The shown test image is 2000×2964 pixels large. We compare the output of the `drawing` software with the output of our algorithm in sRGB space to show the denoising capabilities of our algorithm explicitly, and observe that the result is remarkable. This result verifies that our algorithm works also in a real world setting, and shows that the algorithm generalizes well to images taken with cameras having different hardware characteristics than the images used for training.

4. Conclusion and Outlook

In this paper, we presented a novel method for effective joint demosaicing and denoising that yields both quantitative and qualitative superior results compared to the current state-of-the-art. We believe that the strengths of our algorithm are the following: First, we do not make model assumptions, but let the algorithm learn the underlying image statistics to produce natural results, second, the algorithm can be trained for different CFA patterns without any modifications other than the mosaic operator A , and third, the application of the trained model is very efficient because it consists only of convolutions and point-wise application of the activation function. Due to this structure our model lends itself very well to a hardware implementation. However, an open question remains how we can reduce the model complexity further in order to achieve even better run time with the same demosaicing performance.

Acknowledgements

The authors acknowledge support from the Austrian Science Fund (FWF) under the START project BIVISION, No. Y729.

References

- [1] B. Bayer, "Color Imaging Array," July 20 1976. US Patent 3,971,065.
- [2] D. Khashabi, S. Nowozin, J. Jancsary, and A. W. Fitzgibbon, "Joint Demosaicing and Denoising via Learned Non-parametric Random Fields," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 4968–81, 2014.
- [3] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 1737–1754, 2008.
- [4] X. Li, B. Gunturk, and L. Zhang, "Image Demosaicing: A Systematic Survey," *Proceedings of SPIE*, vol. 6822, 2008.
- [5] D. Menon and G. Calvagno, "Color Image Demosaicing: An Overview," *Signal Processing: Image Communication*, vol. 26, no. 8-9, pp. 518–533, 2011.
- [6] D. Cok, "Signal Processing Method and Apparatus for Producing Interpolated Chrominance Values in a Sampled Color Image Signal," Feb. 10 1987. US Patent 4,642,678.
- [7] C. Laroche and M. Prescott, "Apparatus and Method for Adaptively Interpolating a Full Color Image Utilizing Chrominance Gradients," Dec. 13 1994. US Patent 5,373,322.
- [8] J. Hamilton and J. Adams, "Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera," May 13 1997. US Patent 5,629,734.
- [9] K. Hirakawa and T. W. Parks, "Adaptive Homogeneity-Directed Demosaicing Algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.
- [10] L. Zhang and X. Wu, "Color Demosaicking via Directional Linear Minimum Mean Square-Error Estimation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2167–2178, 2005.
- [11] C. Su, "Highly Effective Iterative Demosaicing Using Weighted-Edge and Color-Difference Interpolations," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, pp. 639–645, 2006.
- [12] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, "Spatially Adaptive Color Filter Array Interpolation for Noiseless and Noisy Data," *International Journal of Imaging Systems and Technology*, vol. 17, no. 3, pp. 105–122, 2007.
- [13] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing With Directional Filtering and a Posteriori Decision," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 132–141, 2007.
- [14] P. Getreuer, "Contour Stencils for Edge-Adaptive Image Interpolation," *Proceedings of SPIE*, vol. 7257, 2009.
- [15] Y. M. Lu, M. Karzand, and M. Vetterli, "Demosaicing by Alternating Projections: Theory and Fast One-Step Implementation," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2085–2098, 2010.
- [16] A. Buades, B. Coll, J. M. Morel, and C. Sbert, "Self-Similarity Driven Color Demosaicking," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1192–1202, 2009.
- [17] L. Zhang, X. Wu, A. Buades, and X. Li, "Color Demosaicking by Local Directional Interpolation and Nonlocal Adaptive Thresholding," *Journal of Electronic Imaging*, vol. 20, no. 2, 2011.
- [18] J. Duran and A. Buades, "Self-similarity and Spectral Correlation Adaptive Algorithm for Color Demosaicking," *IEEE transactions on image processing*, vol. 23, no. 9, pp. 4031–4040, 2014.
- [19] K. Chang, P. L. K. Ding, and B. Li, "Color Image Demosaicking Using Inter-Channel Correlation and Nonlocal Self-Similarity," *Signal Processing: Image Communication*, vol. 39, pp. 264–279, 2015.
- [20] K. Hirakawa and X.-L. Meng, "An Empirical Bayes EM-Wavelet Unification for Simultaneous Denoising, Interpolation, and/or Demosaicing," *IEEE Transactions on Image Processing*, 2006.
- [21] L. Zhang, X. Wu, and D. Zhang, "Color Reproduction from Noisy CFA Data of Single Sensor Digital Cameras," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2184–2197, 2007.

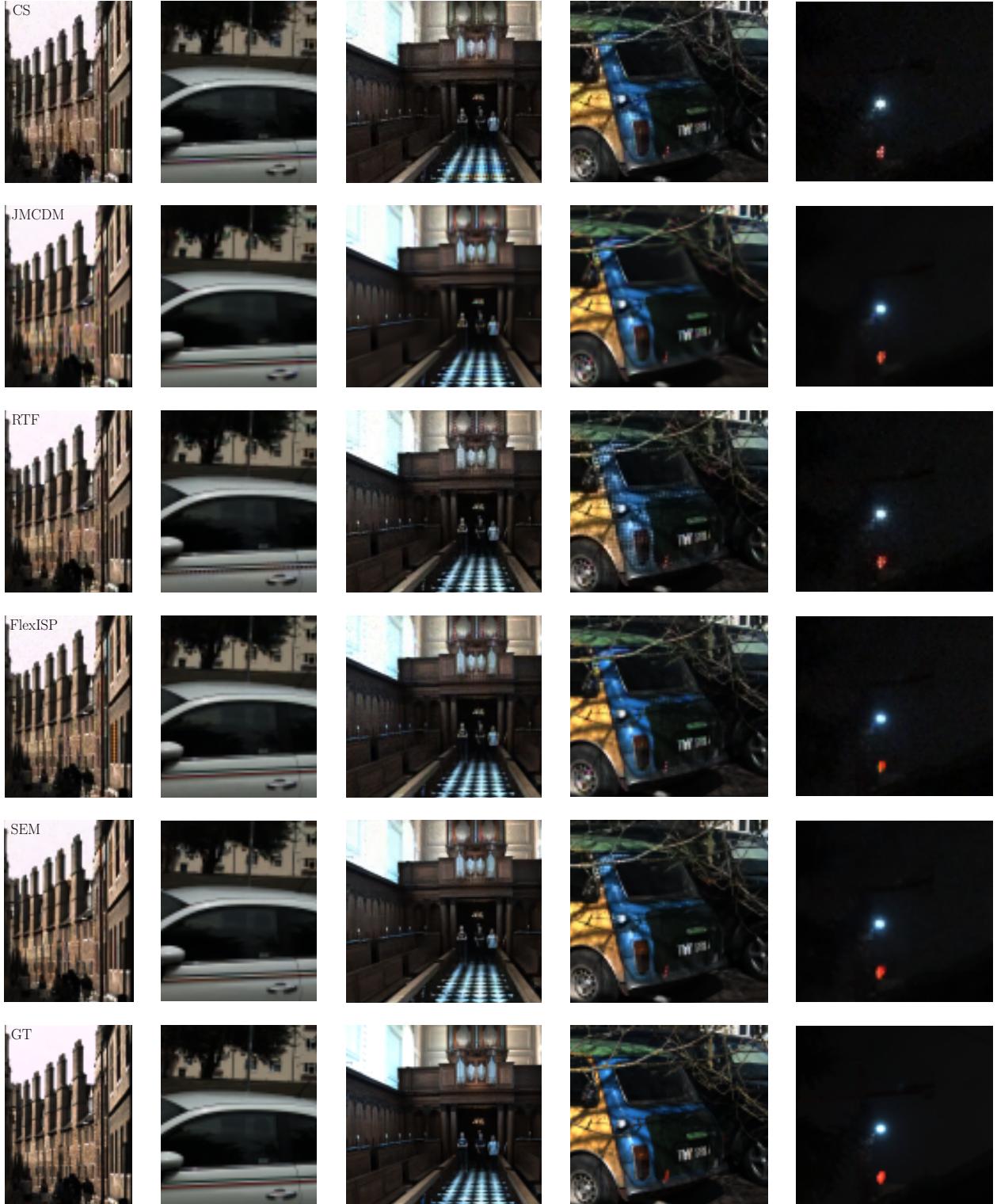


Figure 7. Qualitative results for the top 5 competing methods. Results for CS, RTF, JMCDM, FlexISP and SEM (Ours) methods and ground truth (GT) images from top to bottom. Left to right we show representative crops of images from the test and validation set of the Microsoft Demosaicing data set [2]. Note especially the zipper artifacts and false colors on the left building, on the stripes on the car, on the car wheels, and on the curtains in the church scene. Our superior denoising performance is best seen in the church image or in the night scene. Best viewed in color on screen.



Figure 8. Left: Demosaiced image by the dcraw software, right: Demosaiced image by our algorithm, both in sRGB space. The RAW image was taken with a Sony A6000, pre-processing before demosaicing was done using the dcraw software. Some important areas are zoomed in. Note especially the effective denoising performance of our algorithm on natural camera noise. Best viewed in color on screen.

- [22] D. Paliy, A. Foi, R. Bilcu, and V. Katkovnik, “Denoising and Interpolation of Noisy Bayer Data with Adaptive Cross-Color Filters,” *Proceedings of SPIE*, vol. 6822, 2008.
- [23] L. Condat, “A Simple, Fast and Efficient Approach to Demosaicking: Joint Demosaicking and Denoising,” *IEEE International Conference on Image Processing*, 2010.
- [24] O. Kapah and H. Z. Hel-Or, “Demosaicking Using Artificial Neural Networks,” *Applications of Artificial Neural Networks in Image Processing*, pp. 112–120, 2000.
- [25] F.-L. He, Y.-C. F. Wang, and K.-L. Hua, “A Self-Learning Approach to Color Demosaicking via Support Vector Regression,” in *IEEE International Conference on Image Processing*, 2012.
- [26] J. Sun and M. F. Tappen, “Separable Markov Random Field Model and its Applications in Low Level Vision,” *IEEE transactions on image processing*, vol. 22, no. 1, pp. 402–8, 2013.
- [27] J. Mairal, M. Elad, and G. Sapiro, “Sparse Representation for Color Image Restoration,” *IEEE transactions on image processing*, vol. 17, no. 1, pp. 53–69, 2008.
- [28] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local Sparse Models for Image Restoration,” *IEEE International Conference on Computer Vision*, vol. 2, pp. 2272–2279, 2009.
- [29] W. C. Karl, “Regularization in Image Restoration and Reconstruction,” *Handbook of Image and Video Processing*, pp. 141–160, 2000.
- [30] T. Saito and T. Komatsu, “Demosaicing Approach Based on Extended Color Total-Variation Regularization,” *IEEE International Conference on Image Processing*, pp. 885–888, 2008.
- [31] D. Menon and G. Calvagno, “Regularization Approaches to Demosaicking,” *IEEE Transactions on Image Processing*, vol. 18, no. 10, pp. 2209–2220, 2009.
- [32] K. Dabov, A. Foi, and K. Egiazarian, “Image Denoising by Sparse 3D Transform-domain Collaborative Filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, 2007.
- [33] F. Heide, M. Steinberger, Y.-T. Tsai, R. Mushfigur, D. Pajk, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, J. Kautz, and K. Pulli, “FlexISP : A Flexible Camera Image Processing Framework,” *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia)*, vol. 33, no. 6, 2014.
- [34] Y. Chen, W. Yu, and T. Pock, “On Learning Optimized Reaction Diffusion Processes for Effective Image Restoration,” in *International Conference on Computer Vision and Pattern Recognition*, pp. 5261–5269, 2015.
- [35] B. Colson, P. Marcotte, and G. Savard, “An Overview of Bilevel Optimization,” *Annals of Operations Research*, vol. 153, no. 1, pp. 235–256, 2007.
- [36] K. Kunisch and T. Pock, “A Bilevel Optimization Approach for Parameter Learning in Variational Models,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 938–983, 2013.
- [37] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.
- [38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

- [39] S. Roth and M. J. Black, “Fields of experts,” *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.
- [40] H. S. Malvar, L.-W. He, and R. Cutler, “High-quality Linear Interpolation for Demosaicing of Bayer-Patterned Color Images,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 5–8, 2004.
- [41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [42] R. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A Limited Memory Algorithm for Bound Constrained Optimization,” *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.
- [43] J. Bergstra, O. Breuleux, F. F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: a CPU and GPU Math Compiler in Python,” *Proceedings of the Python for Scientific Computing Conference*, 2010.
- [44] A. van den Oord and B. Schrauwen, “The Student-t Mixture as a Natural Image Patch Prior with Application to Image Compression,” *Journal of Machine Learning Research*, vol. 15, pp. 2061–2086, 2014.
- [45] Y. Chen, R. Ranftl, and T. Pock, “Insights Into Analysis Operator Learning: From Patch-Based Sparse Models to Higher Order MRFs,” *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1060–1072, 2014.