



**HAL**  
open science

## How to best combine demosaicing and denoising?

Yu Guo, Qiyu Jin, Jean-Michel Morel, Gabriele Facciolo

► **To cite this version:**

Yu Guo, Qiyu Jin, Jean-Michel Morel, Gabriele Facciolo. How to best combine demosaicing and denoising?. *Inverse Problems and Imaging*, 2023, pp.0 - 0. 10.3934/ipi.2023044 . hal-04291023

**HAL Id: hal-04291023**

**<https://hal.science/hal-04291023>**

Submitted on 17 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## HOW TO BEST COMBINE DEMOSAICING AND DENOISING?

YU GUO<sup>✉1</sup>, QIYU JIN<sup>✉\*1</sup>, JEAN-MICHEL MOREL<sup>✉2</sup> AND GABRIELE FACCIOLLO<sup>✉3</sup>

<sup>1</sup>School of Mathematical Science, Inner Mongolia University, Hohhot 010020, China

<sup>2</sup>Department of Mathematics, City University of Hong Kong, Kowloon Tong, Hong Kong

<sup>3</sup>Centre Borelli, ENS Paris-Saclay, CNRS, 4, avenue des Sciences 91190 Gif-sur-Yvette, France

**ABSTRACT.** Image demosaicing and denoising play a critical role in the raw imaging pipeline. These processes have often been treated as independent, without considering their interactions. Indeed, most classic denoising methods handle noisy RGB images, not raw images. Conversely, most demosaicing methods address the demosaicing of noise free images. The real problem is to jointly denoise and demosaic noisy raw images. But the question of how to proceed is still not yet clarified. In this paper, we carry-out extensive experiments and a mathematical analysis to tackle this problem by low complexity algorithms. Indeed, both problems have been only addressed jointly by end-to-end heavy weight convolutional neural networks (CNNs), which are currently incompatible with low power portable imaging devices and remain by nature domain (or device) dependent. Our study leads us to conclude that, with moderate noise, demosaicing should be applied first, followed by denoising. This requires a simple adaptation of classic denoising algorithms to demosaiced noise, which we justify and specify. Although our main conclusion is “demosaic first, then denoise”, we also discover that for high noise, there is a moderate PSNR gain by a more complex strategy: partial CFA denoising followed by demosaicing, and by a second denoising on the RGB image. These surprising results are obtained by a black-box optimization of the pipeline, which could be applied to any other pipeline. We validate our results on simulated and real noisy CFA images obtained from several benchmarks.

**1. Introduction.** Most portable digital imaging devices acquire images as mosaics, with a color filter array (CFA), sampling only one color value for each pixel. The most popular CFA is the Bayer color array [5] where two out of four pixels measure the green (G) value, one measures the red (R) and one the blue (B). The two missing color values at each pixel need to be estimated for reconstructing a complete image from a CFA image. The process is commonly referred to as CFA interpolation or demosaicing. CFA images have noise, especially in low light conditions, so denoising is also a key step in the imaging pipeline.

Denoising and demosaicing are often handled as two independent operations [61] for processing noisy raw sensor data. Most of the literature addresses one or the other operation without discussing its combination with the other one.

All classic demosaicing methods have been proposed for noise free CFA images, while denoising algorithms have been designed for color or gray level images only considering additive white noise. Yet the input data is in reality different: it is either

2020 *Mathematics Subject Classification.* Primary: 68U10; Secondary: 62H35.

*Key words and phrases.* Demosaicing, denoising, pipeline, image restoration.

\*Corresponding author: Qiyu Jin.



a CFA image with noise, or a demosaiced image with structured noise. Therefore, we can distinguish three main pipeline strategies: denoising first followed by demosaicing ( $DN&DM$ ), demosaicing first followed by denoising ( $DM&DN$ ), and joint demosaicing-denoising. It might be argued that with the advent of deep learning, the joint operation will become standard and the first two solutions obsolete. But there are three good reasons to address them. The first one is that, contrary to classic image processing chains, processing chains based on deep learning remain domain and device dependent. In other terms, even if they can give the best results on a given test set or device, there is not guarantee that they will deliver good results on out of domain images, or on new devices. Hence, even with slightly apparent lower performance, classic algorithms still retain their value. Secondly, as has been verified many times, insight obtained by combining classic algorithms leads to conceive better deep learning structures. Last but not least, classical algorithms are characterized by computational efficiency and suitability for acceleration. This is exemplified by the successful implementation of classical algorithms, such as the BM3D algorithm, on select mobile devices, made possible through the adoption of advanced process chips, along with continued efforts in algorithmic enhancement and optimization. This accomplishment underscores the promising potential for classical algorithms to extend their reach to a broader spectrum of edge computing devices in the foreseeable future. In contrast, the computational demands of neural networks present challenges when it comes to deployment on low-performance hardware. For these reasons, we shall focus here on a comparison of denoising first followed by demosaicing ( $DN&DM$ ) with demosaicing first followed by denoising ( $DM&DN$ ), and to generalizations of both approaches.

Currently, the most popular classic pipeline is the  $DN&DM$  scheme. This is determined by two basic assumptions. First, after demosaicing, the noise becomes correlated and no longer retains its independent identically distributed (i.i.d) white Gaussian properties. This has a negative impact on traditional denoising algorithms that rely on additive white Gaussian noise (AWGN). Second, state-of-the-art demosaicing algorithms are often designed on a noise-free basis. As a result, many state-of-the-art works [61, 62, 45, 76] operate under the assumption that  $DN&DM$  outperforms  $DM&DN$ .

The advantage of  $DN&DM$  pipelines is that many excellent denoisers can be applied directly, such as model-based TV [67, 37, 11, 39], non-local [6, 52, 44, 42, 41], BM3D [16, 15], low rank [27, 29] and deep learning-based methods [74, 75, 24, 32], because the statistical nature of the noise is maintained. However, these methods are designed and optimized for grayscale or color images and need to be adapted for application to CFA images [62, 17]. Meanwhile, demosaicing algorithms designed on noise-free images can be applied directly after the noise is removed, e.g., [34, 71, 56, 64, 7, 78, 47, 54, 48, 49, 70, 69, 43].

For example Park *et al.* [62] consider the classic Hamilton-Adams (HA) [34] and a frequency-domain algorithm [20] for demosaicing, combined with two denoising methods, BLS-GSM [66] and CBM3D [15]. This combination raises the question of adapting BM3D to a CFA. To do so, the authors first transform the noisy CFA image into the half-size 4-channel image formed by joining the four observed raw values (R,G,G,B) in each four pixel block, then remove noise channel by channel via BM3D [16], finally get the denoised CFA image by the inverse color transform. However, this leads to a checkerboard effect that becomes more noticeable for higher noise levels. Similarly, BM3D-CFA [17] removes noise directly from the

CFA color array, which builds 3D blocks from the same CFA configuration. BM3D-CFA was considered to be a systematic improvement method over [76], in which the method [77] was used as demosaicing method for their comparison of the result after demosaicing. Analogously, [8] adjusted NL-means [6] to the CFA image. Zhang *et al.* [79] uses a filter [3] to extract the luminance of the CFA image. The authors of [76] proposed a PCA-based CFA denoising method that makes full use of spatial and spectral correlation. In [63], Patil and Rajwade remove Poisson noise from CFA images using dictionary learning.

In general, the classical denoising algorithms (such as BM3D, NL-means) can all be adapted to accommodate CFA image denoising in the *DN&DM* strategy. Several of them [61, 62, 45, 76] address this realistic case by processing the noisy CFA images as a half-size 4-channel color image (with one red, two green and one blue channels) and then apply a multichannel denoising algorithm to it. Albeit the *DN&DM* pipeline maintains the independent and identically distributed property of the white Gaussian noise (Poisson noise can be transformed to Gaussian noise by the classical Anscombe transform [4]), the disadvantage is the reduced resolution of the image (half size), which leads to loss of image detail after denoising. Another issue is that it does not take advantage of the relative spatial position of the R, G, and B pixels due to the separation of the image into four independent channels (R,G,G,B) during denoising, resulting in the color distortion problem. Meanwhile, since G is separated into two independent G channels, the difference between the two G channels after denoising causes checkerboard artifacts.

The *DM&DN* pipeline was considered for better image detail preservation and to avoid checkerboard artifacts. Unfortunately, there is not many literatures on such pipelines. This is due to the strong spatial and chromatic correlation of the image noise after demosaicing. These correlations are generated by the demosaicing algorithm and are difficult to be modeled, which is detrimental to model-based denoising algorithms. Condat made an attempt in [12], where he first performed demosaicing and then projected the noise into the luminance channel of the reconstructed image and then denoised it based on the grayscale image. The idea was then further refined in [14, 13]. This approach is similar to ours, but we will give a more elaborate theoretical explanation.

To avoid the accumulation of errors caused by the pipeline order, many researchers have proposed to perform a joint demosaicing and denoising [38, 46, 26]. With the popularity of deep learning, joint demosaicing denoising has gained great resolution and excellent performance. By constructing a large number of pairs of simulated data, joint demosaicing and denoising models can be readily trained. Algorithms based on convolutional neural networks (CNNs), such as [68], exhibit performance far exceeding those of handcrafted algorithms [58]. After [46] introduced the first machine learning-based joint demosaicing and denoising method, Gharbi *et al.* [26] proposed the first deep learning model. Subsequently, a number of algorithms based on deep learning (such as [19, 50, 23, 55, 33]) have been proposed. An unsupervised “mosaic-to-mosaic” training strategy for joint demosaicing and denoising was introduced by Ehret *et al.* [21]. In [30], Guo *et al.* focused on joint demosaicing and denoising of real-world burst images. Further, Xing *et al.* [72] discussed end-to-end joint demosaicing, denoising and super-resolution. In the face of increasing network size and memory consumption, [28] proposed memory efficient joint demosaicing denoising for Ultra High Definition images.

TABLE 1. Advantages and drawbacks of the three types of pipelines.

	<i>DN&amp;DM</i>	<i>DM&amp;DN</i>	Joint <i>DMDN</i>
Advantages	The noise is maintained AWGN	Richer details	Better imaging quality
Drawbacks	Detail loss and checkerboard artifacts	Spatial and chromaticity-related structural noise	High computational complexity and generalization concerns

The deep learning-based algorithms mentioned above achieve state-of-the-art performance, but suffer from a common problem of increasingly large network size and high computational complexity. This problem makes deploying these algorithms to devices, especially in low-power or portable devices, difficult to implement. Also, since deep learning algorithms rely on training, generalization issues might arise. For instance, if the noise range used during training is exceeded, or if the image is out of domain, the results might be significantly inferior to those obtained on a testing set. We have briefly summarized the advantages and drawbacks of the three pipelines in Table 1.

In this paper, we address the problem of combining optimally and adapting state-of-the-art demosaicing and denoising algorithms. A preliminary version of this study appeared in [40]. There, we presented evidence showing that by demosaicing first and then denoising with a higher noise level (denoted *DM&1.5DN* schemes) yields substantially improved result compared with the classic configurations. This paper extends considerably that preliminary work. In particular, we conduct thorough experiments and develop the arguments to confirm and to extend our conclusions. We first establish a model to optimize the denoising and demosaicing pipeline and use the black box optimizer CMA-ES [35] to solve the optimization problem. The optimal results indicate that the *DM&1.5DN* scheme can get almost the same result as the CMA-ES optimum with a CPSNR value difference  $\leq 0.08$  dB when  $\sigma \leq 20$  and performs much better than *DN&DM* and *DM&DN* schemes. Then, we theoretically analyze the statistical properties of demosaiced noise and explain the reason why the *DM&1.5DN* scheme works well. A series of experiments leads us to conclude that the *DM&1.5DN* scheme is always superior to the *DN&DM* and *DM&DN* ones. For large noise, the best scheme is more complex and has three stages, but we shall show that the *DM&1.5DN* scheme still is competitive. Our conclusions are different and actually opposite to those of [61, 62, 45, 76]. The advantages of *DM&1.5DN* scheme seem to be linked to the fact that this scheme does not handle half size 4-channels color image; it therefore uses the classic denoising methods directly on a full resolution color image; this results in more details being preserved and avoids checkerboard artifacts or loss of details. These conclusions also provide theoretical support for real sRGB image denoising [31] which removes noise from full color images after demosaicing. The fact that *DM&1.5DN* schemes improve on the results of raw image denoising will be verified by experiments carried out on two benchmarks, the Smartphone Image Denoising Dataset (SIDD) [1] and the Darmstadt Noise Dataset (DND) [65].

The rest of this paper is structured as follows. In Section 2 we discuss how to apply demosaicing followed by denoising to CFA images. In Section 3, the black

box optimizer CMA-ES is used to find the most general 3-step strategy. The results confirm the preference for *DM&DN* schemes in presence of moderate noise, and lead to a refinement for high noise levels with an *DN&DM&DN* scheme. In Section 4, we are led to define and analyze the statistical properties of the demosaicing residual noise in RGB and in a transformed space that decorrelates the color channels. Then, using these statistical properties, we find experimentally the appropriate noise level that must be used for the denoising method after demosaicing in a *DM&DN* scheme. Section 5 compares our strategy with other state-of-the-art ones on simulated and real image datasets. Section 6 concludes.

**2. The demosaicing and denoising pipeline.** The denoising and demosaicing pipeline consists in solving the ill-posed problem

$$\mathbf{v} = \text{Bayer}(\mathbf{u}) + \epsilon, \quad (1)$$

where  $\mathbf{v} \in \mathbb{R}^{m \times n \times 3}$  is the observed noisy mosaicked image, Bayer is the Bayer color filter,  $\mathbf{u} = (\mathbf{R}, \mathbf{G}, \mathbf{B}) \in \mathbb{R}^{m \times n \times 3}$  is the latent ground truth color image and  $\epsilon$  is Gaussian noise with zero mean and standard deviation  $\sigma$ . As stated in the introduction, we will consider the problem of combining demosaicing and denoising, i.e. which one should be executed first? This brings us to two main pipelines: *DM&DN* (demosaicing then denoising), *DN&DM* (denoising then demosaicing). In [40], we reached the preliminary conclusion: demosaicing should be executed with higher priority and subsequent denoising needs to be adjusted. In the next section we will propose to consolidate (and partly modify) this conclusions by optimizing freely a 3-step procedure. Let  $\sigma_1$  and  $\sigma_2$  be the noise level hyperparameters of *DN&DM* and *DM&DN* respectively.

The restored image can be evaluated by subjective criteria such as visual quality and by objective criteria such as the color signal-to-noise ratio (CPSNR) [3], defined by

$$\text{CPSNR}(\hat{\mathbf{u}}) = 10 \log_{10} \frac{255^2}{\text{MSE}(\hat{\mathbf{u}})}, \quad \text{with} \quad (2)$$

$$\text{MSE}(\hat{\mathbf{u}}) = \frac{1}{m \times n \times 3} \|\hat{\mathbf{u}} - \mathbf{u}\|_F^2,$$

where  $\|\cdot\|_F$  is the Frobenius norm,  $\mathbf{u}$  denotes the ground truth image and  $\hat{\mathbf{u}}$  is the estimated color image.

Park et al. [62] argued that demosaicing introduces chromatic and spatial correlations to the noise, which is no longer i.i.d. white Gaussian and therefore harder to model and eliminate. In [45] the authors argue that *DN&DM* schemes with a proper parameter are more efficient than *DM&DN* schemes. Figure 1 (d) shows an example where a noisy CFA image with noise of standard deviation  $\sigma$  was first demosaiced by RCNN [69] and then restored by CBM3D [15] assuming a noise parameter  $\sigma_2 = \sigma$ . The output of CBM3D with  $\sigma_2 = \sigma$  has a strong residual noise. A similar behavior is also observed with other image denoising algorithms such as nlBayes [40]. Based on this argument several papers [62, 76, 2, 53] propose raw CFA denoising methods applicable before demosaicing.

Other denoising methods that are not explicitly designed to handle raw CFA images (such as CBM3D and nlBayes) can also be adapted to noisy CFA images by rearranging the CFA image into a half-size four-channels image [62], or two half-size three-channel images as shown in Figure 2. In our comparative experiments, CBM3D will be used to process CFA images, which is the scheme in Figure 2, we will denote this method as cfaBM3D.

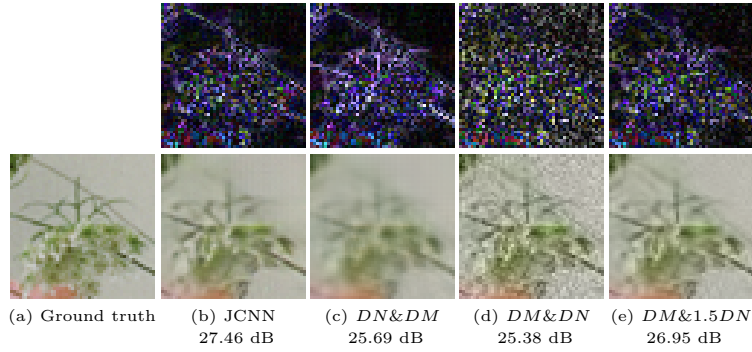


FIGURE 1. Image details at  $\sigma = 20$ . The lower row is the reconstructed image, and the upper row is the difference between the reconstructed image and ground truth. *DN*: cfaBM3D or CBM3D denoising; *DM*: RCNN demosaicing.  $1.5DN$  means that if the noise level is  $\sigma$ , the input noise level parameter of denoising method *DN* is  $\sigma_2 = 1.5\sigma$ .

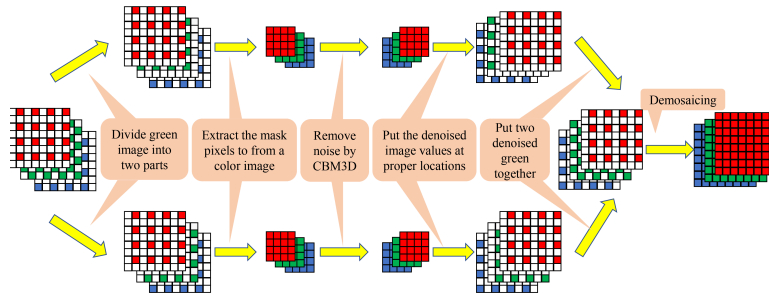


FIGURE 2. The framework used for denoising before demosaicing using an RGB denoiser. The Bayer CFA image is split in two half resolution RGB images, each one with a different green. Both RGB images are denoised independently. Then the pixels of both results are recombined into a denoised Bayer CFA image. The last step consists in applying a demosaicing algorithm.

In the case of splitting the raw image into two half-size 3-channel images (see Figure 2), both images are denoised independently and the denoised pixels are recombined. Each half-size image contributes one green pixel to the denoised CFA image, while the red and blue pixels are averaged. Despite the *DN&DM* pipeline effectively eliminates noise, it is not good at preserving details and produces artifacts such as checkerboard effect. Indeed, due to the rearrangement of the CFA pixels, much image detail is lost in the image after applying an *DN&DM* scheme. In addition, this procedure introduces visible checkerboard artifacts for noise levels  $\sigma > 10$ . These artifacts can be observed in Figure 1 (c). To address this last issue, Danielyan *et al.* [17] proposed BM3D-CFA, which amounts to denoise four different mosaics of the same image before aggregating the four values obtained for each pixel. In practice, we observed that BM3D-CFA and the cfaBM3D method described above attain very similar results. The main difference between the two



FIGURE 3. Generic raw image processing pipeline. This pipeline structure allows for an arbitrary order between  $DN$  and  $DM$  and sets free their parameters. We use the CMA-ES algorithm to optimize the parameter  $\alpha$ ,  $\beta$ ,  $\sigma_1$ ,  $\sigma_2$  in the pipeline.

comes with the execution time, as for cfaBM3D a fast GPU implementation is available [18]. Depending on the experiment we will use one or the other.

Jin *et al.* [40] revised the  $DM&DN$  pipeline and observed that a very simple modification of the noise parameter of the denoiser  $DN$  coped with the structure of demosaiced noise, and led to efficient denoising *after* demosaicing, i.e. a  $DM&1.5DN$  pipeline. This allows for a better preservation of fine structure often smoothed by the  $DN&DM$  schemes, and checkerboard artifacts are no longer observed (see Figure 1 (e)). In terms of quality and speed, demosaicing  $DM$  can be done by a fast algorithm RCNN [69] followed by CBM3D denoising  $1.5DN$ , namely CBM3D applied with a noise parameter equal to  $\sigma_2 = 1.5\sigma$ .

Figure 1 also illustrates that  $DN&DM$  has better CPSNR than  $DM&DN$ . However, the performance of  $DM&1.5DN$  pipeline is much superior to both  $DM&DN$  and  $DN&DM$ . Is  $DM&1.5DN$  pipeline the optimal one? In Section 3, we will explore a more generic optimal pipeline of denoising and demosaicing to confirm this optimality for moderate noise, and a near optimality for large noise. In Section 4, based on the analysis of demosaiced noise we shall seek an explanation of the efficiency of  $DM&1.5DN$ .

**3. Pipeline optimization and analysis.** In order to arrive at a rigorous decision in a more general framework, we designed a generic  $DN_1&DM&DN_2$  pipeline. The structure of the pipeline is illustrated in Figure 3. This pipeline allows for an arbitrary order between  $DN$  and  $DM$  and sets free their parameters. It has two denoisers and four hyperparameters. The two denoisers are a CFA denoiser  $DN_1$  (see Figure 2) and a full color image denoiser  $DN_2$ , which respectively remove noise before and after demosaicing. The four hyperparameters are  $\alpha$  (that controls the weight of CFA denoising),  $\beta$  (that controls the weight of color denoising),  $\sigma_1$  (the noise standard deviation of the CFA denoiser),  $\sigma_2$  (the noise standard deviation of the color denoiser). The results of the pipeline are visualised in Figure 4. The final result of the pipeline is given by

$$\hat{\mathbf{u}} = \beta DN_2(DM(\tilde{\mathbf{v}}), \sigma_2) + (1 - \beta)DM(\tilde{\mathbf{v}}), \quad (3)$$

where

$$\tilde{\mathbf{v}} = \alpha DN_1(\mathbf{v}, \sigma_1) + (1 - \alpha)\mathbf{v}.$$

It follows that if  $\alpha = 1$ ,  $\beta = 0$ ,  $\sigma_1 = \sigma$  and  $\sigma_2 = 0$ , then  $\tilde{\mathbf{v}} = DN(\mathbf{v})$  and  $\hat{\mathbf{u}} = DM(DN(\mathbf{v}))$ , i.e. the pipeline is  $DN&DM$ ; if  $\alpha = 0$ ,  $\beta = 1$ ,  $\sigma_1 = 0$  and  $\sigma_2 = \sigma$ , then  $\tilde{\mathbf{v}} = \mathbf{v}$  and  $\hat{\mathbf{u}} = DN(DM(\mathbf{v}))$ , i.e. the pipeline is  $DM&DN$ ; if  $\alpha = 0$ ,  $\beta = 1$ ,  $\sigma_1 = 0$  and  $\sigma_2 = 1.5\sigma$ , then  $\tilde{\mathbf{v}} = \mathbf{v}$  and  $\hat{\mathbf{u}} = DN(DM(\mathbf{v}))$ , i.e. the pipeline is  $DM&1.5DN$  [40].



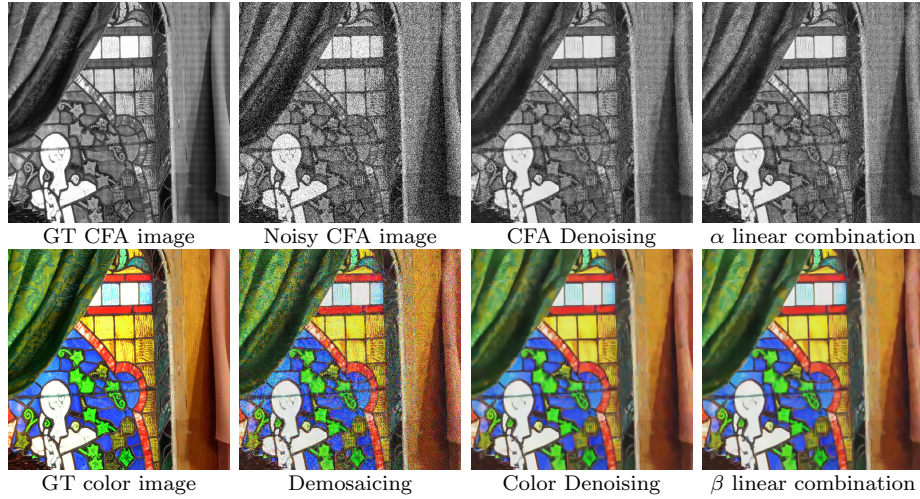


FIGURE 4. A visual representation of the process in Figure 3, where the noise level is  $\sigma = 60$ . The parameters are  $\alpha = 0.90$ ,  $\beta = 0.99$ ,  $\sigma_1 = 34.50$ ,  $\sigma_2 = 54.42$ . Since  $\beta$  is always close to 1 in the pipeline, the visual difference between Color Denoising and the  $\beta$  linear combination is not significant.

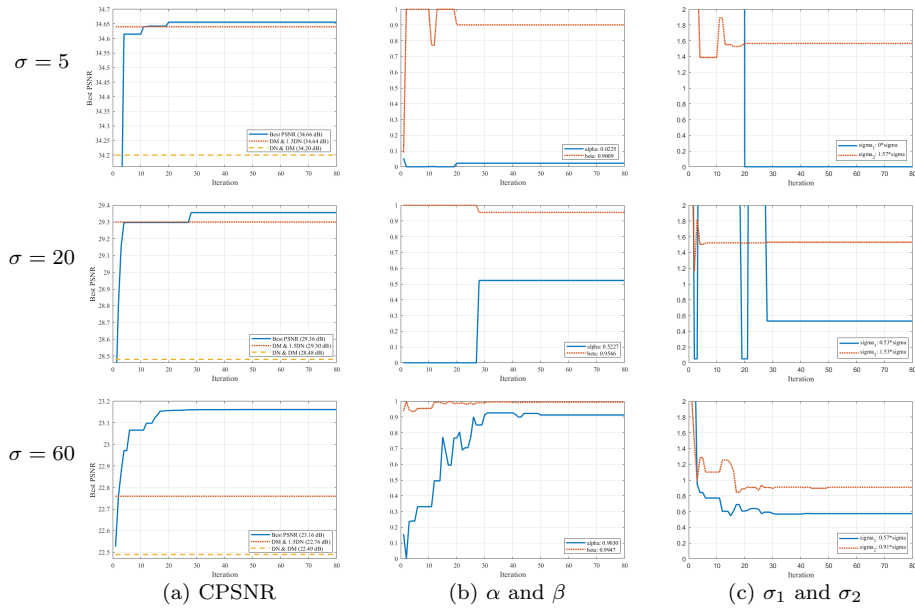


FIGURE 5. Evolution of the result of iterating CMA-ES when optimizing the parameters  $\alpha, \beta, \sigma_1, \sigma_2$  of the processing pipeline.

Our purpose is, for every noise level  $\sigma$ , to find the optimal values  $\{\alpha^*, \beta^*, \sigma_1^*, \sigma_2^*\}$  satisfying

$$\{\alpha^*, \beta^*, \sigma_1^*, \sigma_2^*\} = \arg \max_{\{\alpha, \beta, \sigma_1, \sigma_2\}} \text{CPSNR}(\hat{\mathbf{u}}), \quad (4)$$

TABLE 2. The optimization result of CMA-ES for the pipeline  $DN_1&DM&DN_2$  (see Eq. (3)), where  $\sigma, \sigma_1, \sigma_2 \in [0, 255]$  and  $\alpha, \beta \in [0, 1]$ . In this experiment  $DM$  is always MLRI and  $DN$  is CBM3D or cfaBM3D depending on the input data.

$\sigma$	Method	$\alpha$	$\beta$	$\sigma_1$	$\sigma_2$	CPSNR Imax	CPSNR Kodak
5	$DN&DM$	1.00	0.00	5.00	0	34.20	35.08
	$DM&DN$	0.00	1.00	0	5.00	34.18	35.03
	$DM&1.5DN$	0.00	1.00	0	7.50	34.64	35.77
	CMA-ES	0.02	0.90	0	7.83	<b>34.66</b>	<b>35.78</b>
10	$DN&DM$	1.00	0.00	10.00	0	31.68	32.15
	$DM&DN$	0.00	1.00	0	10.00	31.55	31.62
	$DM&1.5DN$	0.00	1.00	0	15.00	32.35	32.99
	CMA-ES	0.51	0.92	6.81	12.98	<b>32.43</b>	<b>33.02</b>
20	$DN&DM$	1.00	0.00	20.00	0	28.48	28.91
	$DM&DN$	0.00	1.00	0	20.00	28.07	27.75
	$DM&1.5DN$	0.00	1.00	0	30.00	29.30	29.85
	CMA-ES	0.52	0.95	10.58	30.63	<b>29.36</b>	<b>29.91</b>
40	$DN&DM$	1.00	0.00	40.00	0	24.90	25.84
	$DM&DN$	0.00	1.00	0	40.00	24.16	24.05
	$DM&1.5DN$	0.00	1.00	0	60.00	25.46	26.53
	CMA-ES	0.82	0.98	23.46	41.79	<b>25.74</b>	<b>26.72</b>
50	$DN&DM$	1.00	0.00	50.00	0	23.62	24.83
	$DM&DN$	0.00	1.00	0	50.00	22.87	23.00
	$DM&1.5DN$	0.00	1.00	0	75.00	24.01	25.33
	CMA-ES	0.72	1.00	30.55	49.75	<b>24.36</b>	<b>25.61</b>
60	$DN&DM$	1.00	0.00	60.00	0	22.49	23.90
	$DM&DN$	0.00	1.00	0	60.00	21.83	22.24
	$DM&1.5DN$	0.00	1.00	0	90.00	22.76	24.26
	CMA-ES	0.90	0.99	34.50	54.42	<b>23.16</b>	<b>24.60</b>

where  $\hat{\mathbf{u}}$  is defined by (3) and CPSNR is defined in (2).

Obviously, problem (4) is non-linear, non-convex and the gradients are not readily available. In order to obtain the optimal solution of (4) (and inspired by [59]), we used the black box optimizer CMA-ES [35], which is a random search optimizer that is based on evolutionary strategies. Unlike common gradient optimization, CMA-ES does not compute the gradient of the objective function. Only the ranking between candidate solutions is exploited for learning the sample distribution; neither derivatives nor even the function values themselves are required by the method [36].

We carried out experiments with different noise levels ( $\sigma = 5, 10, 20, 40, 50, 60$ ) on the images from the Imax [78] and Kodak [25] datasets. In this experiment we used the denoiser with the framework Figure 2 for  $DN_1$ , MLRI for  $DM$  and CBM3D [15] for  $DN_2$ . For each experiment,  $\{\alpha, \beta, \sigma_1, \sigma_2\}$  were initialized randomly. Figure 5 illustrates the evolution of the CPSNR during the optimization with respect to  $\{\alpha, \beta, \sigma_1, \sigma_2\}$ . In all cases, the parameters and the CPSNR stabilize after about 60-iterations. The final results are shown in Table 2 along with results of the  $DN&DM$  method (cfaBM3D+MLRI)<sup>1</sup>, the  $DM&DN$  method (MLRI+CBM3D)

<sup>1</sup>Here the CFA image is divided into two half-size RGB images then the noise is removed by CBM3D (see Figure 2).



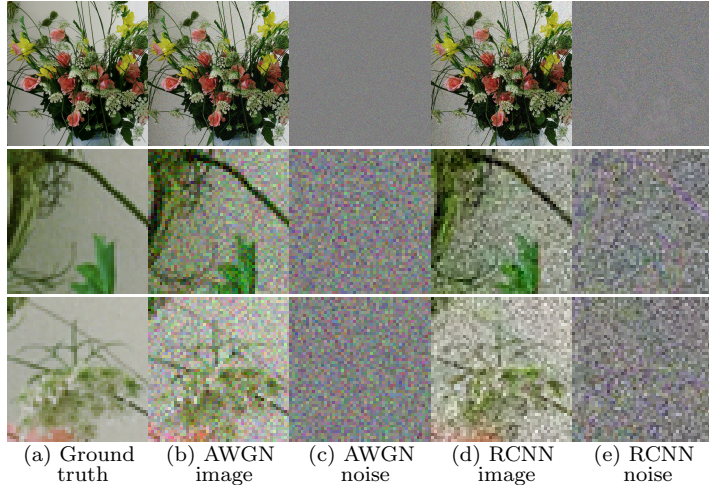


FIGURE 6. First row: (a) Ground truth Imax 3, (b) its noisy version, (c) added white noise ( $\sigma = 20$ ), (d) demosaiced version of (b) by RCNN, (e) the *demosaiced noise*, namely the difference (d)-(a). Second and third rows:  $50 \times 50$  extracts from the first row.

and  $DM\&1.5DN$  (MLRI+1.5CBM3D as in [40]). When  $\sigma \leq 20$  the optimal CMA-ES result is almost identical to the one of  $DM\&1.5DN$ , and much better than  $DN\&DM$  and  $DM\&DN$ . When  $\sigma \geq 40$  the optimal CMA-ES result is much better than the ones obtained by  $DN\&DM$ ,  $DM\&DN$  and  $DM\&1.5DN$ . When  $\sigma = 5$ , we observe that  $\sigma_1 = 0$ , which means that the pipeline is exactly  $DM\&DN$  with parameter  $\sigma_2/\sigma = 1.566$ , i.e.  $DM\&1.566DN$ . When  $\sigma \geq 10$ ,  $\sigma_1$  is almost equal to  $0.5\sigma$ , however the CPSNR gain is only marginal. The value  $\sigma_2/\sigma$  decreases as  $\sigma$  increases. Furthermore, from  $\sigma = 5$  to  $60$ ,  $\alpha$  increases from  $0.0225$  to  $0.9030$  while  $\beta$  remains always larger than  $0.9$ . This means that applying denoising before demosaicing is not important for low noise levels, but becomes necessary when  $\sigma$  increases, while applying denoising after demosaicing is always favorable, but with a little smaller denoising parameters.

When the noise level is high, the CPSNR of  $DM\&1.5DN$  is  $0.3$  to  $0.4$  dB below the optimal value obtained by the  $DN_1\&DM\&DN_2$  pipeline. However, this requires almost doubling the computational complexity due to denoising. Therefore, by trading-off image quality and computational cost, the simplified  $DM\&1.5DN$  pipeline remains a good option and it is almost optimal for moderate noise. For this reason, we shall explore in detail this pipeline and the reasons of its near optimality in the next section.

**4. Analysis of  $DM\&1.5DN$ .** As we saw in Section 3, The result of the  $DM\&1.5DN$  pipeline is almost equal to the result of the optimal  $DN_1\&DM\&DN_2$  pipeline and much better than the  $DN\&DM$  pipeline for all noise levels. The fact that a  $DM\&1.5DN$  pipeline surpasses than a  $DN\&DM$  scheme is surprising, considering that after demosaicing the noise is no longer white. Indeed, chromatic and spatial correlations have been introduced by the demosaicing, while the applied denoiser was conceived for white noise. This apparent paradox leads us to analyze the behavior of demosaiced noise.

TABLE 3. RMSE between ground truth and demosaicked image for different demosaicking algorithms in presence of noise of standard deviation  $\sigma$ .

$\sigma$	HA	GBTf	RI	MLRI	RCNN
1	5.04	5.10	4.17	4.06	3.21
3	5.70	5.79	4.97	4.88	4.17
5	6.78	6.87	6.12	6.10	5.59
10	10.18	10.27	9.53	9.74	9.65
15	13.93	14.01	13.15	13.64	13.87
20	17.75	17.83	16.77	17.56	18.04
30	25.36	25.42	23.94	25.30	26.21
40	32.67	32.76	30.77	32.64	33.98
50	39.58	39.71	37.25	39.55	41.21
60	46.14	46.35	43.43	46.11	47.95

**Definition 4.1.** Consider a ground truth color image  $(\mathbf{R}, \mathbf{G}, \mathbf{B})$  and its mosaic obtained by keeping only one value of either  $\mathbf{R}, \mathbf{G}, \mathbf{B}$  at each pixel, on a fixed Bayer pattern. Assume that white noise with standard deviation  $\sigma$  has been added to the mosaicked image, and that the resulting noisy mosaic has been demosaiced by  $DM$ , hence giving a noisy image  $(\tilde{\mathbf{R}}, \tilde{\mathbf{G}}, \tilde{\mathbf{B}})$ . We then call demosaiced noise the difference  $(\tilde{\mathbf{R}} - \mathbf{R}, \tilde{\mathbf{G}} - \mathbf{G}, \tilde{\mathbf{B}} - \mathbf{B})$ .

Figure 6 illustrates the above definition. The *demosaiced noise* is nothing but the difference between the demosaiced version of a noisy image and its underlying ground truth. The demosaiced noise of column (d) is (visually) not significantly higher than the white noise of column (b), but it is clearly no longer white, due to the introduction of chromaticity and spatial correlations. The properties of the *demosaiced noise* depend on the demosaicing algorithm, as developed in [40]. This paper compares  $DM \& 1.5DN$  pipelines composed of seven different state-of-the-art demosaicing algorithms (such as HA [34], GBTF [64], RI [47] and so on). To understand empirically the right noise model to adopt after demosaicing, and following the conclusions of [40], we applied CBM3D after demosaicing with a noise parameter  $\sigma_2$  corresponding to  $\sigma$  multiplied by  $(1.0, 1.1, \dots, 1.9)$ . These experiments show that the optimal parameter interval is  $[1.4, 1.7]$  and that the optimal factor is 1.5.

This surprising result would seem to imply that demosaicing increases noise. But this is not the case, as illustrated in Table 3, which gives the noise standard deviation estimated as the mean RMSE of the demosaiced images from the Imax [78] dataset with different noise levels. For low noise ( $\sigma = 1$ ) the large demosaicing error of about 4 clearly is caused by the demosaicing itself. However, for  $\sigma > 10$  the RMSE of the demosaiced image tends to be roughly equal to 3/4 of the initial noise standard deviation. In short, as expected from an interpolation algorithm, demosaicing (slightly) decreases the noise standard deviation. This is also consistent with the visual results observed in Figure 6.

At first sight, this 3/4 factor contradicts the observation that denoising with a parameter  $\sigma_2 = 1.5\sigma$  yields better results. This leads us to further analyze the structure of the demosaiced residual noise. To that aim, we applied an orthonormal

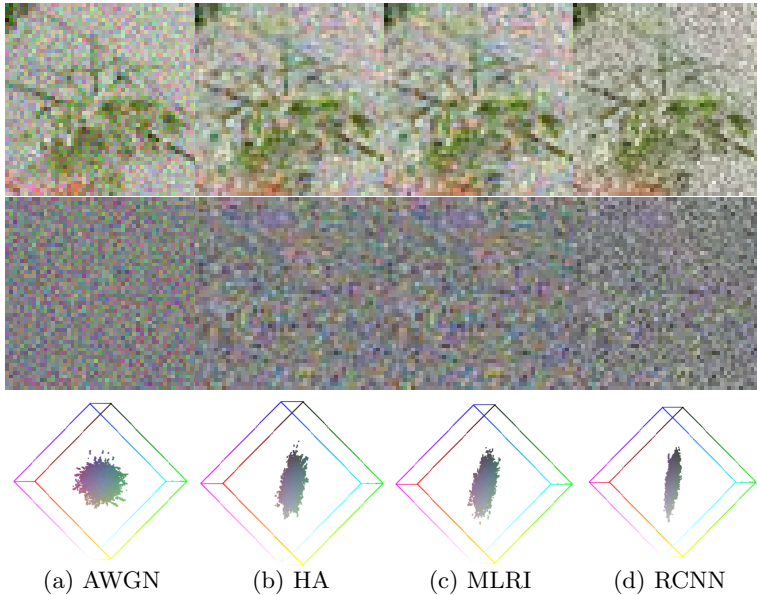


FIGURE 7. AWGN image and demosaicing noise with standard deviation  $\sigma = 20$  for respectively HA, MLRI, RCNN. Last row: the color spaces (in standard (R,G,B) Cartesian coordinates) of each noise, presented in their projection with maximal area. As expected, the AWG color space is isotropic, while the color space after demosaicing is elongated in the luminance direction  $\mathbf{Y}$  and squeezed in the others. This amounts to an increased noise standard deviation for  $\mathbf{Y}$  after demosaicing, and less noise in the chromatic directions. See table Table 4 for quantitative results.

Karhunen-Loeve transform to the residual noise to maximally decorrelate the color channels [57, 60]. This transform is commonly used in denoising algorithms [51] such as CBM3D [15]. Here, we used a transform  $(\mathbf{R}, \mathbf{G}, \mathbf{B}) \rightarrow (\mathbf{Y}, \mathbf{C}_1, \mathbf{C}_2)$ , in which the luminance direction is  $\mathbf{Y} = \frac{\mathbf{R} + \mathbf{G} + \mathbf{B}}{\sqrt{3}}$  and the orthogonal vectors  $\mathbf{C}_1$  and  $\mathbf{C}_2$  are arbitrarily chosen as in [45], which is defined as

$$\begin{pmatrix} \mathbf{Y} \\ \mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{6} & -2/\sqrt{6} & 1/\sqrt{6} \end{pmatrix} \begin{pmatrix} \mathbf{R} \\ \mathbf{G} \\ \mathbf{B} \end{pmatrix}. \quad (5)$$

The color distortion caused by denoising in the  $\mathbf{Y}\mathbf{C}_1\mathbf{C}_2$  space is much less than that in the RGB space, and this transformation does not change the properties of independent identically distributed noise. This explains why it is generally used for color image denoising. We further analyze the properties of residual noise in the  $\mathbf{Y}\mathbf{C}_1\mathbf{C}_2$  color space.

From Figure 7 one can see that the AWG noise is isotropic whereas the demosaiced noise is not isotropic anymore in the RGB space. The noise is elongated in the brightness direction  $\mathbf{Y} = \frac{\mathbf{R} + \mathbf{G} + \mathbf{B}}{\sqrt{3}}$ , and compressed in other directions. Furthermore, the noise becomes blurred after demosaicking. This indicates that the

TABLE 4. Noise intensity. Variance and covariance of  $(\mathbf{R}, \mathbf{G}, \mathbf{B})$  and  $(\mathbf{Y}, \mathbf{C}_1, \mathbf{C}_2)$  between pixels  $(i, j)$  and  $(i + s, j + t)$ ,  $s, t = 0, 1, 2$  first for AWGN (a) with standard deviation  $\sigma = 20$ , then for its demosaiced versions by HA (b), RI (c), MLRI (d) and RCNN (e).

	$(i,j)$	$(i,j+1)$	$(i,j+2)$	$(i+1,j)$	$(i+1,j+1)$	$(i+1,j+2)$	$(i+2,j)$	$(i+2,j+1)$	$(i+2,j+2)$
R	400.6	0.6	0.4	0.7	0.1	0.7	0.3	0.2	0.8
G	401.7	0.5	1.1	0.1	0.3	0.9	1.0	0.6	0.4
B	400.2	1.2	0.1	0.5	0.6	0.0	1.9	0.3	1.9
Y	399.6	1.1	0.1	0.3	0.1	0.9	0.2	0.5	1.2
C <sub>1</sub>	401.5	0.1	0.8	0.6	0.3	0.3	0.9	0.5	1.3
C <sub>2</sub>	401.4	0.2	1.8	0.9	0.2	1.0	0.6	0.2	0.2
(a) AWGN									
	$(i,j)$	$(i,j+1)$	$(i,j+2)$	$(i+1,j)$	$(i+1,j+1)$	$(i+1,j+2)$	$(i+2,j)$	$(i+2,j+1)$	$(i+2,j+2)$
R	359.6	152.1	15.1	154.8	92.5	18.9	18.6	17.6	8.5
G	359.3	91.4	1.0	100.3	23.9	1.8	0.8	0.4	5.1
B	377.4	150.7	15.2	155.5	89.3	18.5	20.6	17.5	8.1
Y	654.4	185.4	50.8	196.1	60.0	2.9	49.7	9.1	19.0
C <sub>1</sub>	274.6	143.2	42.5	144.9	99.3	22.1	48.3	24.5	6.4
C <sub>2</sub>	167.2	65.5	37.6	69.7	46.4	20.0	41.4	20.0	9.1
(b) HA									
	$(i,j)$	$(i,j+1)$	$(i,j+2)$	$(i+1,j)$	$(i+1,j+1)$	$(i+1,j+2)$	$(i+2,j)$	$(i+2,j+1)$	$(i+2,j+2)$
R	336.4	126.8	19.4	129.9	52.9	21.6	20.7	22.4	18.7
G	295.5	92.5	0.5	95.6	20.6	1.8	0.7	1.5	4.3
B	350.5	125.9	18.1	130.4	50.7	20.8	20.0	20.9	17.5
Y	715.6	170.9	32.3	178.6	2.6	5.4	34.0	7.1	20.5
C <sub>1</sub>	168.4	108.3	41.3	110.1	73.4	28.2	44.1	29.4	9.7
C <sub>2</sub>	98.3	66.0	27.9	67.3	48.1	21.4	29.9	22.4	10.4
(c) RI									
	$(i,j)$	$(i,j+1)$	$(i,j+2)$	$(i+1,j)$	$(i+1,j+1)$	$(i+1,j+2)$	$(i+2,j)$	$(i+2,j+1)$	$(i+2,j+2)$
R	361.4	128.4	18.9	130.5	46.4	20.6	21.6	21.5	19.8
G	298.9	93.0	0.5	95.1	19.1	0.9	1.0	0.5	3.8
B	370.9	127.8	19.3	130.4	46.0	20.6	21.2	20.3	19.0
Y	772.2	177.7	33.0	181.3	9.6	9.2	32.6	10.9	21.4
C <sub>1</sub>	164.8	107.1	43.7	108.8	72.8	29.3	46.1	30.2	10.1
C <sub>2</sub>	94.3	64.4	28.1	65.8	48.2	21.9	30.3	23.1	11.1
(d) MLRI									
	$(i,j)$	$(i,j+1)$	$(i,j+2)$	$(i+1,j)$	$(i+1,j+1)$	$(i+1,j+2)$	$(i+2,j)$	$(i+2,j+1)$	$(i+2,j+2)$
R	359.9	47.8	5.0	51.9	21.8	17.8	5.1	19.4	9.2
G	354.8	32.6	4.4	36.3	5.8	8.4	6.4	8.8	0.6
B	356.0	49.6	6.3	53.7	23.6	18.8	7.3	19.4	9.2
Y	972.3	69.0	20.8	76.4	3.6	18.6	28.9	17.3	2.2
C <sub>1</sub>	55.1	33.8	15.3	36.0	26.1	14.6	19.0	16.6	11.8
C <sub>2</sub>	43.3	27.3	12.3	29.4	21.5	11.7	16.0	13.7	9.4
(e) RCNN									

demosaiced noise is correlated between adjacent pixels. This is also verified in Table 4 which illustrates the variances and covariances of AWGN and demosaicked noise with  $\sigma = 20$  both in RGB and  $YC_1C_2$  spaces. One can observe that the statistical properties of AWG noise remains unchanged while that of demosaicked noise changes obviously after  $(\mathbf{R}, \mathbf{G}, \mathbf{B}) \rightarrow (\mathbf{Y}, \mathbf{C}_1, \mathbf{C}_2)$  transformation. The variance of  $\mathbf{Y}$  is a growing sequence for the demosaicked noise obtained by increasingly sophisticated demosaicing: 654 for HA, 715 for RI, 772 for MLRI, 972 for RCNN.

TABLE 5. Correlation between pixels. The corresponding correlations of  $(\mathbf{R}, \mathbf{G}, \mathbf{B})$  and  $(\mathbf{Y}, \mathbf{C}_1, \mathbf{C}_2)$  between pixels  $(i, j)$  and  $(i + s, j + t)$ ,  $s, t = 0, 1, 2$  first for AWGN (a) with standard deviation  $\sigma = 20$ , then for its demosaiced versions by HA (b), RI (c), MLRI (d) and RCNN (e).

	$(i, j)$	$(i, j+1)$	$(i, j+2)$	$(i+1, j)$	$(i+1, j+1)$	$(i+1, j+2)$	$(i+2, j)$	$(i+2, j+1)$	$(i+2, j+2)$
R	1.0000	0.0015	0.0010	0.0017	0.0002	0.0018	0.0007	0.0005	0.0021
G	1.0000	0.0012	0.0028	0.0004	0.0007	0.0023	0.0025	0.0016	0.0010
B	1.0000	0.0029	0.0002	0.0013	0.0015	0.0001	0.0047	0.0008	0.0047
Y	1.0000	0.0028	0.0004	0.0007	0.0002	0.0023	0.0005	0.0012	0.0030
C <sub>1</sub>	1.0000	0.0003	0.0021	0.0016	0.0007	0.0008	0.0024	0.0011	0.0033
C <sub>2</sub>	1.0000	0.0005	0.0045	0.0023	0.0005	0.0025	0.0014	0.0005	0.0005
(a) AWGN									
	$(i, j)$	$(i, j+1)$	$(i, j+2)$	$(i+1, j)$	$(i+1, j+1)$	$(i+1, j+2)$	$(i+2, j)$	$(i+2, j+1)$	$(i+2, j+2)$
R	1.0000	0.4229	0.0420	0.4307	0.2574	0.0525	0.0518	0.0489	0.0236
G	1.0000	0.2543	0.0029	0.2791	0.0666	0.0050	0.0022	0.0010	0.0142
B	1.0000	0.3994	0.0403	0.4122	0.2368	0.0490	0.0545	0.0464	0.0215
Y	1.0000	0.2834	0.0777	0.2997	0.0918	0.0044	0.0760	0.0138	0.0290
C <sub>1</sub>	1.0000	0.5215	0.1548	0.5278	0.3619	0.0804	0.1759	0.0892	0.0234
C <sub>2</sub>	1.0000	0.3919	0.2248	0.4166	0.2776	0.1194	0.2477	0.1198	0.0547
(b) HA									
	$(i, j)$	$(i, j+1)$	$(i, j+2)$	$(i+1, j)$	$(i+1, j+1)$	$(i+1, j+2)$	$(i+2, j)$	$(i+2, j+1)$	$(i+2, j+2)$
R	1.0000	0.3744	0.0588	0.3893	0.1536	0.0633	0.0671	0.0626	0.0542
G	1.0000	0.3099	0.0044	0.3265	0.0681	0.0063	0.0038	0.0040	0.0163
B	1.0000	0.3631	0.0579	0.3715	0.1431	0.0631	0.0612	0.0585	0.0523
Y	1.0000	0.2382	0.0419	0.2510	0.0003	0.0058	0.0407	0.0129	0.0298
C <sub>1</sub>	1.0000	0.6422	0.2442	0.6548	0.4345	0.1655	0.2639	0.1746	0.0568
C <sub>2</sub>	1.0000	0.6690	0.2795	0.6846	0.4904	0.2188	0.3012	0.2291	0.1075
(c) RI									
	$(i, j)$	$(i, j+1)$	$(i, j+2)$	$(i+1, j)$	$(i+1, j+1)$	$(i+1, j+2)$	$(i+2, j)$	$(i+2, j+1)$	$(i+2, j+2)$
R	1.0000	0.3496	0.0516	0.3624	0.1213	0.0544	0.0632	0.0543	0.0546
G	1.0000	0.3077	0.0001	0.3221	0.0623	0.0039	0.0099	0.0019	0.0145
B	1.0000	0.3449	0.0567	0.3525	0.1225	0.0589	0.0624	0.0561	0.0567
Y	1.0000	0.2271	0.0404	0.2371	0.0164	0.0103	0.0366	0.0165	0.0305
C <sub>1</sub>	1.0000	0.6479	0.2625	0.6632	0.4400	0.1748	0.2868	0.1863	0.0632
C <sub>2</sub>	1.0000	0.6806	0.2959	0.6965	0.5121	0.2343	0.3200	0.2472	0.1208
(d) MLRI									
	$(i, j)$	$(i, j+1)$	$(i, j+2)$	$(i+1, j)$	$(i+1, j+1)$	$(i+1, j+2)$	$(i+2, j)$	$(i+2, j+1)$	$(i+2, j+2)$
R	1.0000	0.1328	0.0138	0.1441	0.0605	0.0493	0.0141	0.0538	0.0256
G	1.0000	0.0919	0.0125	0.1022	0.0164	0.0237	0.0181	0.0246	0.0016
B	1.0000	0.1393	0.0176	0.1508	0.0662	0.0527	0.0206	0.0546	0.0260
Y	1.0000	0.0709	0.0214	0.0786	0.0037	0.0192	0.0298	0.0178	0.0022
C <sub>1</sub>	1.0000	0.6129	0.2773	0.6539	0.4730	0.2649	0.3443	0.3003	0.2143
C <sub>2</sub>	1.0000	0.6302	0.2851	0.6789	0.4963	0.2697	0.3688	0.3171	0.2161
(e) RCNN									

Hence, the noise standard deviation on  $\mathbf{Y}$  has been multiplied by a factor between 1.27 and 1.56. In contrast, the demosaiced noise is reduced in the  $\mathbf{C}_1$  and  $\mathbf{C}_2$  axes, with its variance passing from 400 for AWGN to 168 and 94 for RI, and even down to 43 and 55 for RCNN. Table 4 also shows that the covariances between adjacent pixels are no longer close to 0 and that the covariances of demosaicked noise is

TABLE 6. Correlation between channels. Covariance (each first row) and corresponding correlation (each second row) of the three color channels (R, G, and B) of the demosaicing noise when the initial CFA white noise satisfies  $\sigma = 20$ . See Figure 7 for an illustration.

	R	G	B		R	G	B
R	359.56 1.0000	172.02 0.4786	93.85 0.2548	R	336.44 1.0000	206.29 0.6542	175.01 0.5097
G	172.02 0.4786	359.30 1.0000	167.60 0.4551	G	206.29 0.6542	295.54 1.0000	200.96 0.6244
B	93.85 0.2548	167.60 0.4551	377.44 1.0000	B	175.01 0.5097	200.96 0.6244	350.46 1.0000
	Y	C <sub>1</sub>	C <sub>2</sub>		Y	C <sub>1</sub>	C <sub>2</sub>
Y	654.41 1.0000	5.50 0.0130	31.47 0.0951	Y	715.65 1.0000	3.55 0.0102	9.10 0.0343
C <sub>1</sub>	5.50 0.0130	274.65 1.0000	7.71 0.0360	C <sub>1</sub>	3.55 0.0102	168.44 1.0000	7.12 0.0554
C <sub>2</sub>	31.47 0.0951	7.71 0.0360	167.23 1.0000	C <sub>2</sub>	9.10 0.0343	7.12 0.0554	98.35 1.0000
	(a) HA				(b) RI		
	R	G	B		R	G	B
R	361.42 1.0000	224.39 0.6826	201.41 0.5501	R	359.90 1.0000	320.44 0.8967	302.85 0.8461
G	224.39 0.6826	298.94 1.0000	216.86 0.6512	G	320.44 0.8967	354.83 1.0000	299.85 0.8437
B	201.41 0.5501	216.86 0.6512	370.92 1.0000	B	302.85 0.8461	299.85 0.8437	355.99 1.0000
	Y	C <sub>1</sub>	C <sub>2</sub>		Y	C <sub>1</sub>	C <sub>2</sub>
Y	772.20 1.0000	0.80 0.0023	22.64 0.0839	Y	972.34 1.0000	10.00 0.0432	1.97 0.0096
C <sub>1</sub>	0.80 0.0023	164.76 1.0000	7.09 0.0569	C <sub>1</sub>	10.00 0.0432	55.09 1.0000	10.75 0.2202
C <sub>2</sub>	22.64 0.0839	7.09 0.0569	94.33 1.0000	C <sub>2</sub>	1.97 0.0096	10.75 0.2202	43.29 1.0000
	(c) MLRI				(d) RCNN		

an almost descending sequence by increasingly sophisticated demosaicing. In order to further analyze the correlation between adjacent pixel noises, the correlation coefficients of adjacent pixel noises are calculated and listed in Table 5. The correlation of AWGN is (almost) 0 due to the independent properties (see Table 5 (a)). However, the demosaiced noise have a strong correlation in  $(\mathbf{R}, \mathbf{G}, \mathbf{B})$  color space. After transformation, the channel correlation of  $\mathbf{Y}$  decreases significantly and the correlation of  $\mathbf{C}_1$  and  $\mathbf{C}_2$  increases.

These observations lead to a simple conclusion: As the computational complexity increases, the  $\mathbf{Y}$  component of the demosaiced noise gets closer to white. However,

the residual noise on  $\mathbf{C}_1$  and  $\mathbf{C}_2$  is strongly spatially correlated, it is therefore a low frequency noise, that will require stronger filtering than white noise to be removed. Since image denoising algorithms are guided by the  $\mathbf{Y}$  component [15, 52], we can denoise with methods designed for white noise, but with a noise parameter adapted to the increased variance of  $\mathbf{Y}$ .

To understand why the variance of  $\mathbf{Y}$  is far larger than the AWGN it comes from, let us study in Table 6 the correlation between the three channels ( $\mathbf{R}, \mathbf{G}, \mathbf{B}$ ) in the demosaiced noise of HA, RI, MLRI and RCNN. We observe a strong ( $\mathbf{R}, \mathbf{G}, \mathbf{B}$ ) correlation ranging from 0.4 for HA to 0.89 for RCNN, which is caused by the "tendency to grey" of all demosaicing algorithms (see Figures 6 and 7). Assuming that the demosaiced noisy pixel components (denoted  $\tilde{\epsilon}_{\mathbf{R}}, \tilde{\epsilon}_{\mathbf{G}}, \tilde{\epsilon}_{\mathbf{B}}$ ) have a correlation coefficient close to 1 then we have

$$\mathbf{Y} = \frac{\tilde{\epsilon}_{\mathbf{R}} + \tilde{\epsilon}_{\mathbf{G}} + \tilde{\epsilon}_{\mathbf{B}}}{\sqrt{3}} \sim \sqrt{3} N(0, \sigma).$$

This factor of about 1.7 corresponds to the case with maximum correlation. The empirical observation of an optimal factor near 1.5 responds to a lower correlation between the colors.

All in all, the analysis of the statistical properties of demosaicked noise explains why the  $DM\&DN$  scheme with an appropriate parameter  $\sigma_2 = 1.5\sigma$  performs similarly to the optimal CMA-ES, and is much better than  $DN\&DM$ .

**5. Experimental evaluation.** To evaluate the proposed framework for denoising and demosaicing, we conducted experiments on simulated images and real images separately. The most classic Imax [78] and Kodak [25] datasets were selected for the simulated images. To verify the effect on real raw images, we also evaluated it on the SIDD dataset [1] and on the DND [65] benchmark. The former comes with ground truth acquisitions, while the latter allows to evaluate the results by submitting them to the benchmark website.

**5.1. Evaluation of  $DN\&DM$  and  $DM\&1.5DN$  strategies on simulated images.** All Imax and Kodak images were corrupted by AWGN with standard deviations  $\sigma = 5, 10, 20, 40, 50, 60$ .

We compared nine different pipelines, namely:

- Best performing  $DN\&DM$  and  $DM\&1.5DN$  pipelines built by RCNN [69] and cfaBM3D or CBM3D [15].
- Low cost  $DN\&DM$ ,  $DM\&1.5DN$  and CMA-ES pipelines built by MLRI [48] and cfaBM3D or CBM3D [15].
- The CFA denoising framework proposed by Park *et al.* in [62], which effectively compresses the signal energy by using a color representation obtained by principal component analysis of the Kodak dataset, and then removes the noise in each channel by BM3D. We combined this framework with RCNN [69].
- The PCA-CFA filter proposed in [76] uses principal component analysis (PCA) and spatial and spectral correlation of images to preserve color edges and details. We combined it with DLMM demosaicing [77] and RCNN demosaicing [69].
- Since 2016, solving joint demosaicing denoising has typically used deep learning. As a reference, we included JCNN [26, 22], which is one of the classical deep learning algorithms for this problem, for comparison. It is important to emphasize that it was trained on noise standard deviations  $\sigma \leq 20$  only.



TABLE 7. The results of different combinations of denoising and demosaicing methods for the **Imax** image dataset. The best result for each row is **red**, the second best result is **brown**, and the third best result is **blue**.

$\sigma$	<i>DN&amp;DM</i>					<i>DM&amp;1.5DN</i>		CMA-ES	JCNN
	cfaBM3D+ MLRI	cfaBM3D+ RCNN	Park+ RCNN	PCA+ DLMM	PCA+ RCNN	RCNN+ CBM3D	MLRI+ CBM3D	cfaBM3D+ MLRI+ CBM3D	
5	34.20	<b>35.21</b>	32.86	32.69	<b>34.87</b>	<b>35.44</b>	34.64	34.66	33.48
10	31.68	32.26	30.06	30.73	31.89	<b>32.77</b>	32.35	<b>32.43</b>	<b>33.09</b>
20	28.48	28.73	26.86	27.57	27.99	<b>29.54</b>	29.30	<b>29.36</b>	<b>29.79</b>
40	24.90	24.92	23.86	23.50	23.57	<b>25.69</b>	<b>25.46</b>	<b>25.74</b>	–
50	23.62	23.59	22.67	22.08	22.10	<b>24.27</b>	<b>24.01</b>	<b>24.36</b>	–
60	22.49	22.43	21.75	20.89	20.89	<b>23.02</b>	<b>22.76</b>	<b>23.16</b>	–
Av	27.56	27.86	26.34	26.24	26.89	<b>28.46</b>	<b>28.09</b>	<b>28.29</b>	–

TABLE 8. The results of different combinations of denoising and demosaicing methods for the **Kodak** image dataset. The best result for each row is **red**, the second best result is **brown**, and the third best result is **blue**.

$\sigma$	<i>DN&amp;DM</i>					<i>DM&amp;1.5DN</i>		CMA-ES	JCNN
	cfaBM3D+ MLRI	cfaBM3D+ RCNN	Park+ RCNN	PCA+ DLMM	PCA+ RCNN	RCNN+ CBM3D	MLRI+ CBM3D	cfaBM3D+ MLRI+ CBM3D	
5	35.08	<b>36.10</b>	34.87	34.99	35.42	<b>36.58</b>	35.77	<b>35.78</b>	34.13
10	32.15	32.56	30.85	31.83	32.01	<b>33.36</b>	32.99	<b>33.02</b>	<b>33.27</b>
20	28.91	29.03	27.42	28.11	28.14	<b>30.12</b>	29.85	<b>29.91</b>	<b>29.95</b>
40	25.84	25.85	24.88	24.15	24.08	<b>26.82</b>	<b>26.53</b>	<b>26.72</b>	–
50	24.83	24.83	23.91	22.85	22.77	<b>25.67</b>	<b>25.33</b>	<b>25.61</b>	–
60	23.90	23.89	23.19	21.77	21.70	<b>24.62</b>	<b>24.26</b>	<b>24.60</b>	–
Av	28.45	28.71	27.52	27.28	27.35	<b>29.53</b>	<b>29.12</b>	<b>29.27</b>	–

Table 7 shows that RCNN+1.5CBM3D obtains the optimum on average. It comes to no surprise that JCNN [26, 22] performs slightly better than the other methods on the Imax dataset. Table 8 shows that the *DM&1.5DN* method RCNN + 1.5CBM3D yields the best results on the Kodak dataset. And when the noise increases, the ‘low-cost’ MLRI+1.5CBM3D also achieves impressive results. However, it is restricted to a limited range of noise levels and cannot handle the noise levels outside the training range. Furthermore, it requires much more memory and computation. In summary, *DM&1.5DN* methods are more robust and have a better performance than cfaBM3D+RCNN. All *DM&1.5DN* methods outperform the *DN&DM* methods Park+RCNN [62], PCA+DLMM [76] and PCA+RCNN [76].

We now examine the visual quality of restored images. Figures 8-10 compare the visual quality obtained by the main discussed methods. Key parts of images were zoomed-in for a better view. From the upper-left extract of Figure 8, we can see that textures are well restored by RCNN+1.5CBM3D and MLRI+1.5CBM3D, while they are blurred the cfaBM3D+RCNN and destroyed by JCNN. In the lower-left extract, the girl’s hairs are oversmoothed by cfaBM3D+RCNN and JCNN but are well preserved by our proposed method. In the upper-left and lower-left corner



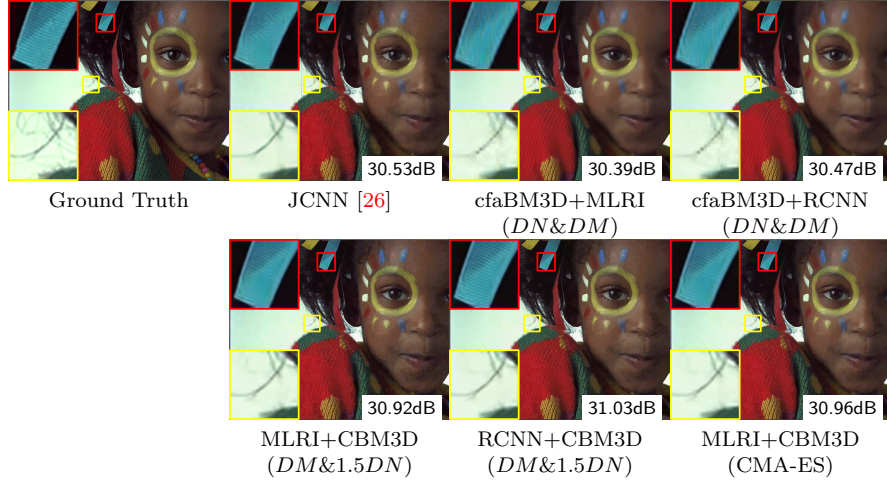


FIGURE 8. Demosaicing and denoising results on an image from the Kodak dataset with  $\sigma = 20$ . We compare the two schemes of  $DN&DM$ ,  $cfaBM3D+MLRI$  and  $cfaBM3D+RCNN$ , the two schemes of  $DM&1.5DN$ ,  $MLRI+CBM3D$  and  $RCNN+CBM3D$ , and the  $MLRI+CBM3D$  schemes optimized by the CMA-ES algorithm. As a reference we also include the result of JCNN, a joint CNN method.

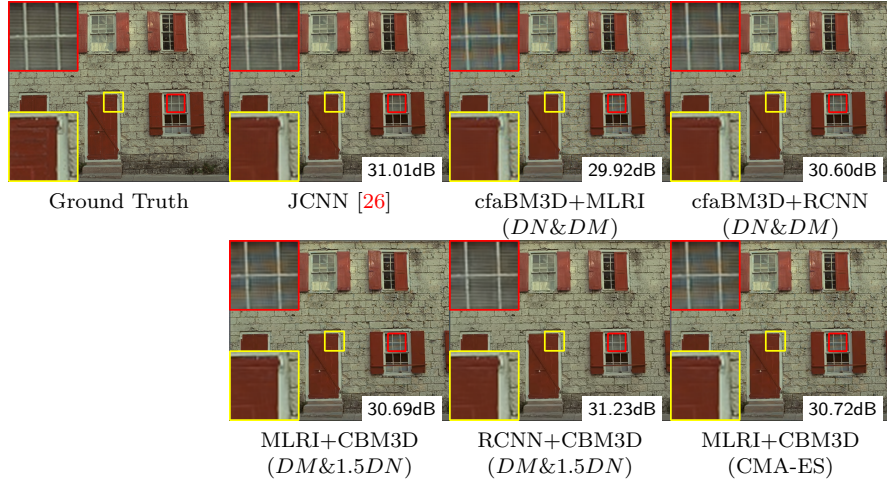


FIGURE 9. Demosaicing and denoising results on an image from the Kodak dataset with  $\sigma = 10$ . We compare the two schemes of  $DN&DM$ ,  $cfaBM3D+MLRI$  and  $cfaBM3D+RCNN$ , the two schemes of  $DM&1.5DN$ ,  $MLRI+CBM3D$  and  $RCNN+CBM3D$ , and the  $MLRI+CBM3D$  schemes optimized by the CMA-ES algorithm. As a reference we also include the result of JCNN, a joint CNN method.

of Figure 9,  $cfaBM3D+RCNN$  oversmooths the details and JCNN introduces some artifacts at the window and oversmooths the door. Instead,  $RCNN+1.5CBM3D$

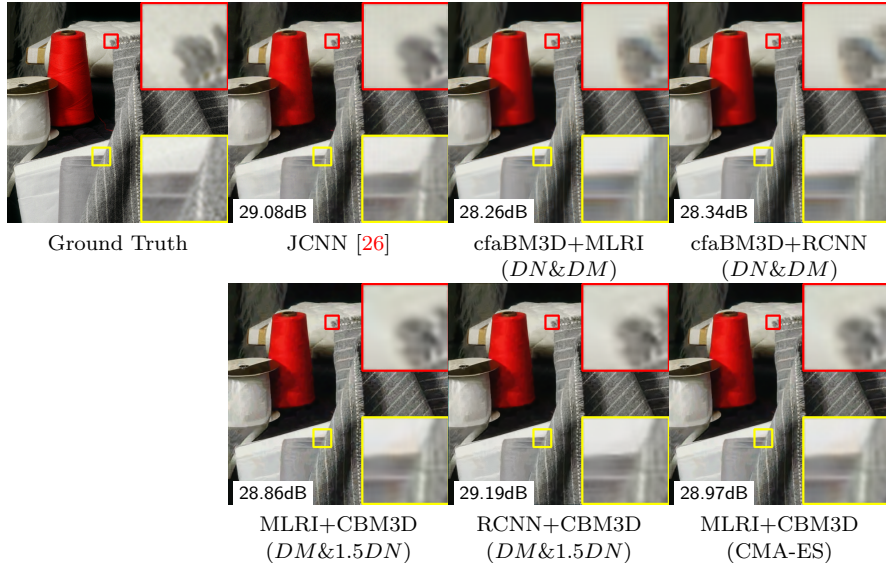


FIGURE 10. Demosaicing and denoising results on an image from the Imax dataset with  $\sigma = 20$ . We compare the two schemes of  $DN\&DM$ ,  $cfaBM3D+MLRI$  and  $cfaBM3D+RCNN$ , the two schemes of  $DM\&1.5DN$ ,  $MLRI+CBM3D$  and  $RCNN+CBM3D$ , and the  $MLRI+CBM3D$  schemes optimized by the CMA-ES algorithm. As a reference we also include the result of JCNN, a joint CNN method.

preserves the details and does not introduce artifacts. The zoomed-in parts of Figure 10 show that JCNN and  $cfaBM3D+RCNN$  introduce checkerboard artifacts while methods based on the  $DM\&1.5DN$  scheme do not. The advantage of our proposed approach becomes more obvious when dealing with high noise. There are severe checkerboard artifacts in the images restored by  $cfaBM3D+MLRI$  and  $cfaBM3D+RCNN$  (see in the bottom left-hand corner of the image of Figure 11), and the details are oversmoothed (see in the upper left corner of the image of Figure 11), while our proposed approach not only avoids checkerboard artifacts, but also retains the details. The image restored with JCNN is very noisy because JCNN was not trained beyond  $\sigma = 20$ .

As a rule of thumb, the  $DM\&DN$  scheme with an appropriate parameter (namely  $DM\&1.5DN$ ) outperforms the competition in terms of visual quality. This is due to the fact that it efficiently uses spatial and spectral image characteristics to remove noise, preserve edges and fine detail. Indeed, contrary to the  $DN\&DM$  schemes,  $DM\&1.5DN$  does not reduce the resolution of the noisy image. Using a  $DN\&DM$  scheme ends up over-smoothing the result. A comparison of CPSNRs and visual quality on these simulated examples leads to conclude that the  $DM\&1.5DN$  scheme is indeed much more robust and better performing than the  $DN\&DM$  scheme.

**5.2. Evaluation of  $DN\&DM$  and  $DM\&1.5DN$  strategies on real image datasets.** In order to prove the advantage of a  $DM\&1.5DN$  strategy on real images, we evaluated its application to the real sRGB images taken from the SIDD dataset [1]. In this dataset, the noisy sRGB images and their corresponding ground

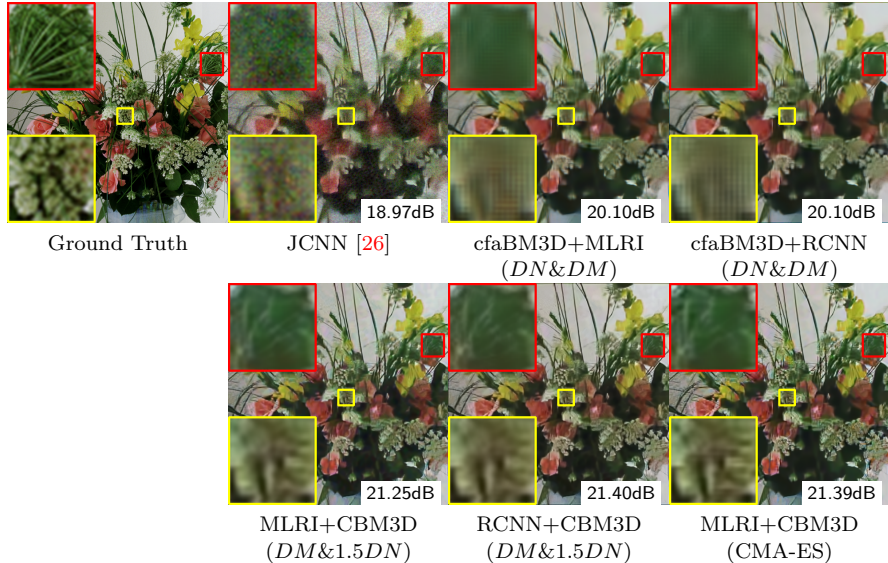


FIGURE 11. Demosaicing and denoising results on an image from the Imax dataset with  $\sigma = 60$ . We compare the two schemes of  $DN\&DM$ ,  $cfaBM3D+MLRI$  and  $cfaBM3D+RCNN$ , the two schemes of  $DM\&1.5DN$ ,  $MLRI+CBM3D$  and  $RCNN+CBM3D$ , and the  $MLRI+CBM3D$  schemes optimized by the CMA-ES algorithm. As a reference we also include the result of JCNN, a joint CNN method.

TABLE 9. Average CPSNR results on the SIDD dataset. Note that for each camera, images with different noise levels are being considered. The noise range is  $\sigma \in [3.28, 38.12]$ . The proposed  $DM\&1.5DN$  schemes outperforms the  $DN\&DM$  ones. The best result is in red, the second best one is in brown.

Camera	$\sigma$ range	JCNN	cfaBM3D+ MLRI	cfaBM3D+ RCNN	MLRI+ CBM3D	RCNN+ CBM3D
IP7	[5.29, 10.65]	36.79	37.30	37.43	37.72	38.37
S6	[3.71, 38.12]	32.89	33.15	33.31	33.96	33.97
GP	[3.28, 35.90]	36.42	36.78	37.15	37.52	37.58
N6	[4.03, 31.15]	33.38	33.96	34.16	34.36	34.21
G4	[4.66, 13.85]	37.03	37.00	37.20	37.94	37.97
Av.	[3.28, 38.12]	35.41	35.80	36.00	36.41	36.63

truth images were acquired by five different mobile phone models. We considered the five most effective demosaicing and denoising schemes among those considered above, namely  $cfaBM3D+MLRI$ ,  $cfaBM3D+RCNN$ ,  $MLRI+1.5CBM3D$ ,  $RCNN+1.5CBM3D$  and JCNN. The noise level was estimated by using the method [9] and provided to the denoising algorithms and JCNN. Since the sRGB images used in this experiment are already tone-mapped we assumed that the resulting noise is approximately homoscedastic. This allowed us to estimate a single noise level per image instead of a noise curve. Thus, a different noise level was computed for each image in the **SIDD sRGB image dataset**. The noise estimated for all

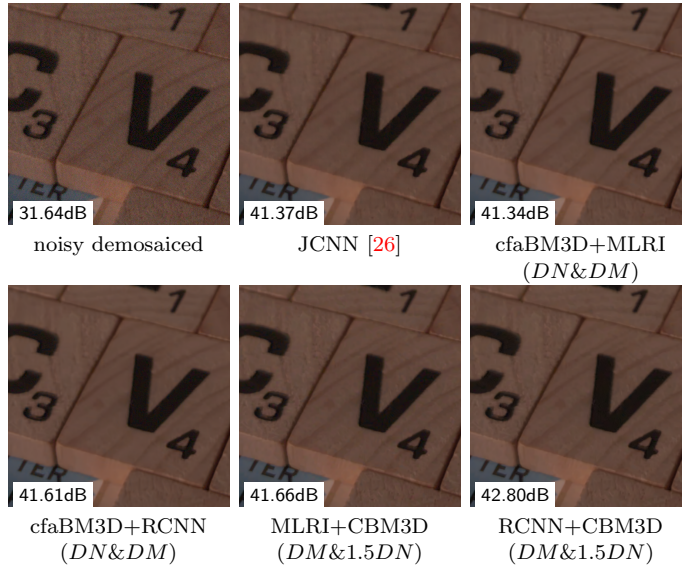


FIGURE 12. Demosaicing and denoising results on an image from the SIDD dataset. We compare the two schemes of  $DN&DM$ , cfaBM3D+MLRI and cfaBM3D+RCNN, the two schemes of  $DM&1.5DN$ , MLRI+CBM3D and RCNN+CBM3D. As a reference we also include the result of JCNN, a joint CNN method.

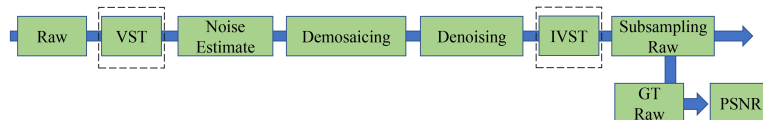


FIGURE 13. The flowchart of raw image denoising under  $DM&1.5DN$  scheme. The dashed VST/IVST blocks are active in just one of the pipeline variants.

the images is in the range  $\sigma \in [3.28, 38.12]$ , and the noise level of most of the images ( $\geq 93.75\%$ ) is no higher than 20. This justifies the choice of  $DM&1.5DN$ .

Table 9 shows the CPSNR and estimated noise levels of images generated by different schemes on the SIDD dataset. We list them separately by phone model. It can be seen from Table 9 that the  $DM&1.5DN$  solution is more competitive than the  $DN&DM$  solution in terms of CPSNR, with an average 0.60 dB gain. This is consistent with the previous results on the simulated data. Figure 12 shows the visual quality of both strategies. JCNN is not competitive on the SIDD dataset, because it was not trained on this dataset. This also shows that our proposed scheme has better robustness and adaptability than JCNN. The  $DM&1.5DN$  scheme keeps more image details than others.

In a word, the  $DM&1.5DN$  scheme clearly outperforms  $DN&DM$  in visual quality and numerical results for both simulated data and real data. Our results also provide theoretical support for real sRGB image denoising which removes noise from full color images after demosaicing. The next section addresses raw image denoising.



TABLE 10. Validation of the  $DM&1.5DN$  scheme on the SIDD dataset. Note that for each camera, images with different noise levels are being considered. The noise range is  $\sigma \in [0.48, 22.59]$  without VST and  $\sigma \in [0.38, 13.00]$  with VST. The best result is in **red**, the second best one is in **brown**.

		cfaBM3D	JCNN	HA+ CBM3D	RCNN+ FFDNet	RCNN+ CBM3D	MLRI+ CBM3D
Raw	VST	49.03	46.05	49.18	48.51	<b>49.30</b>	<b>50.55</b>
	non-VST	48.53	45.51	49.02	48.55	<b>49.22</b>	<b>50.45</b>

TABLE 11. Comparison results of the  $DM&1.5DN$  scheme on the SIDD and DND benchmarks (results as reported on the corresponding websites). \* indicates the use of the variance stabilizing transform (VST). The best result is in **red**, and the second best one is in **brown**.

Raw	TNRD	MLP	EPLL	WNNM	BM3D	RCNN+ CBM3D	MLRI+ CBM3D	CycleISP
	SIDD	42.77	43.17	40.73	44.85	45.52	<b>48.36</b>	
SIDD*	–	–	–	–	–	<b>48.56</b>	<b>49.48</b>	–
DND	44.97	42.70	46.31	46.30	46.64	<b>47.16</b>	<b>47.63</b>	<b>49.13</b>
DND*	45.70	45.71	46.86	47.05	47.15	<b>47.26</b>	<b>47.76</b>	–

**5.3. The  $DM&1.5DN$  strategy for raw image denoising.** We applied the  $DM&1.5DN$  scheme to raw image denoising. To that aim, we defined the pipeline shown in Figure 13. We considered two pipeline variants: with and without variance stabilizing transform. In the first case, a variance stabilizing transformation was used to transform the raw image noise into approximate Gaussian noise, and the noise level in each image was then estimated by the method [9]. In the second case, we applied the noise estimation method [9] directly on the original noise images. Table 10 shows the results of the  $DM&1.5DN$  scheme on the raw images of the SIDD dataset [1]. Note that applying the VST leads to slightly better results in almost all cases. RCNN underperforms when handling raw data, because its training data is sRGB data. MLRI is a traditional interpolation algorithm, which is not affected by different color spaces and achieves the best results. The estimated noise range for the original noisy images in the **SIDD raw image datasets** is  $\sigma \in [0.48, 22.59]$  and after VST is  $\sigma \in [0.38, 13.00]$ . According to Table 2, the results of the CMA-ES optimized scheme and the  $DM&1.5DN$  scheme are almost equal when the noise level  $\sigma \leq 20$ , which justifies the use of  $DM&1.5DN$  (more precisely, the noise level of all considered images is always less than 23). Considering the trade-off between reconstruction quality and computational consumption, the  $DM&1.5DN$  scheme is more valuable for the considered application.

To further validate the performance of the  $DM&1.5DN$  scheme, we compared MLRI+CBM3D and RCNN+CBM3D with TNRD [10], EPLL [80], WNNM [27], BM3D [16] and CycleISP [73] on the SIDD [1] and DND [65] benchmarks. As with the previous results, the noise ranges of the **raw images** in the SIDD and DND benchmarks are respectively  $\sigma \in [0.57, 21.39]$  and  $\sigma \in [0.59, 14.97]$ , and after VST the noise ranges are  $\sigma \in [0.46, 12.79]$  and  $\sigma \in [0.44, 9.17]$ , which still satisfy the best

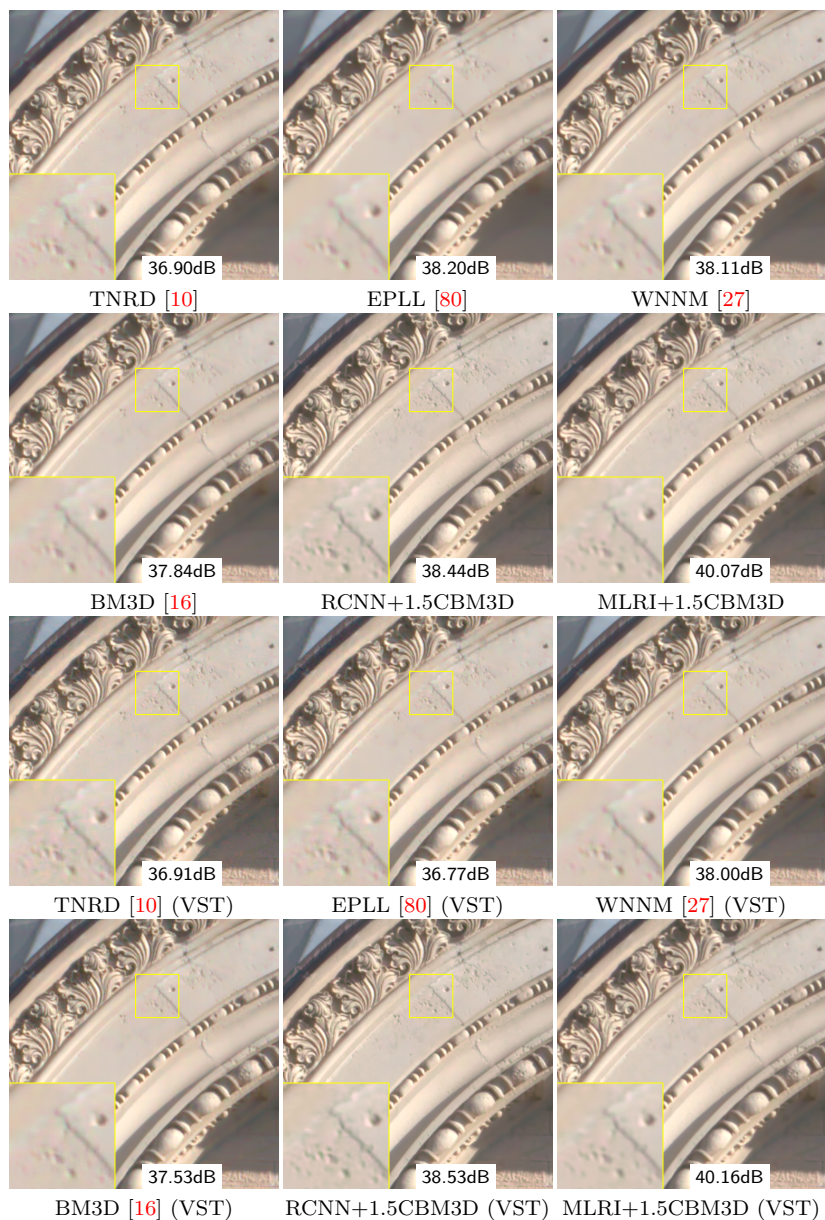


FIGURE 14. Denoising results on an image from the DND dataset. We compare the  $DM&1.5DN$  scheme (MLRI+CBM3D and RCNN+CBM3D), TNRD [10], EPLL [80], WNNM [27] and BM3D [16] (results as reported on the benchmark website).

use case for  $DM&1.5DN$ . The relevant results are shown in Table 11, and more detailed results can be found on the SIDD<sup>2</sup> and DND<sup>3</sup> websites. The CycleISP result is better on DND than our best proposed scheme MLRI+CBM3D, but not

<sup>2</sup><http://www.cs.yorku.ca/~kamel/sidd/benchmark.php>

<sup>3</sup>[https://noise.visinf.tu-darmstadt.de/benchmark/#results\\_raw](https://noise.visinf.tu-darmstadt.de/benchmark/#results_raw)

TABLE 12. Time consumption. The average running time (CPU) of the three strategies in processing 10 images on a PC with an Intel Core i7-9750H 2.60GHz CPU and 16GB memory. Note that we do not use the deep learning methods and only compared the traditional methods.

<i>DN&amp;DM</i>			<i>DM&amp;1.5DN</i>			CMA-ES
cfaBM3D+ HA	cfaBM3D+ RI	cfaBM3D+ MLRI	HA+ CBM3D	RI+ CBM3D	MLRI+ CBM3D	cfaBM3D+ MLRI+ CBM3D
7.41 s	7.64 s	7.85 s	16.16 s	16.66 s	16.72 s	23.93 s

on SIDD, this is likely due to the domain difference between DND and SIDD (as SIDD has darker images). Therefore, this deep learning based approach has several caveats: first MLRI and CBM3D offer guarantees of domain independence and were not trained on the specific image pipeline associated with DND. Second, a difference of 1.5 dB is anyway visually invisible for such high PSNRs as those involved in the table (see Figure 14). Third, MLRI and CBM3D can be accelerated without performance loss on dedicated architectures while the computational weight of a CNN is hardly reducible.

Although the *DM&1.5DN* scheme falls short of state-of-the-art deep learning raw image denoising methods such as CycleISP [73], our proposed lightweight scheme is still the best among traditional algorithms and it even outperforms some deep learning algorithms (see the DND benchmark website). Compared to the computational resources consumed by deep learning methods, our proposed scheme is computationally very competitive. Figure 14 shows the comparison of the visual quality of traditional algorithms on raw image denoising. Our scheme keeps more details, introduces fewer color artifacts than other traditional algorithms and avoids checkerboard artifacts. With a lightweight demosaicker, BM3D obviously improves on raw image denoising with an average gain of 3.91 dB for SIDD, 0.99 dB for DND and 0.61 dB for DND with VST. As a result, we can conclude that the *DM&1.5DN* scheme is very effective for raw image denoising.

**5.4. Time consumption and generalizability.** We examined the runtimes of three strategies and evaluated the generalizability of the CMA-ES scheme, aiming to achieve a balance between good performance and reasonable runtimes. We limited our comparison to traditional algorithms, as deep learning algorithms require long computing times on CPUs. Table 12 shows the running times of the three strategies on a PC with an Intel Core i7-9750H 2.60GHz CPU and 16GB memory. As the table demonstrates, the demosaicing algorithm has a negligible runtime, while the majority of the computational time is spent on denoising. The computation time of *DN&DM* is half that of *DM&1.5DN*, because *DN&DM* processes two half-size images, which is exactly half the size of the full-color images processed by *DM&1.5DN*. In terms of the trade-off between time consumption and performance, *DM&1.5DN* is the optimal choice, particularly for moderate levels of noise ( $\sigma \leq 20$ , as described in Section 5.3). However, for high noise scenes, the *DN<sub>1</sub>&DM&DN<sub>2</sub>* pipeline may be the best option for achieving optimal performance.

We now turn our attention to the generalization of the CMA-ES optimization parameters, which requires a large number of calculations, making the optimization process time-consuming. One critical aspect is the independence of the parameters

TABLE 13. Generalizability of CMA-ES optimal parameters to different noise levels. Evaluation of noise levels with  $\sigma = 50$  proximity (selected as 46 to 54) using two generalization schemes.

$\sigma$	<i>DN&amp;DM</i>	<i>DM&amp;1.5DN</i>	CMA-ES	
			image transformation	$\sigma$ transformation
46	24.10	24.60	24.83	<b>24.90</b>
47	23.98	24.46	24.74	<b>24.78</b>
48	23.85	24.32	24.63	<b>24.64</b>
49	23.74	24.19	<b>24.52</b>	<b>24.52</b>
51	23.50	23.91	<b>24.26</b>	<b>24.26</b>
52	23.35	23.77	<b>24.13</b>	24.12
53	23.24	23.64	<b>24.00</b>	<b>24.00</b>
54	23.14	23.52	<b>23.90</b>	23.89

from the dataset. This issue arises implicitly in the previous discussion. In Section 3, we employed the Imax dataset for the CMA-ES optimization, whereas the parameters were applied directly to the Kodak dataset in the comparison (see Tables 2 and 8). As demonstrated in these tables, the CMA-ES optimal parameters remain consistent when applied to the Kodak dataset, which leads to the conclusion that the CMA-ES optimization parameters exhibit good generalization across datasets.

Another crucial aspect is the generalization to different noise levels. Given that it is impractical to train optimal parameters each time for real-world applications, it is essential to discuss what to do when the noise level does not match the level of optimal parameters. We propose two schemes:

- Image transformation, where the image is transformed to the nearest noise level using the corresponding optimal parameters  $\alpha, \beta, \sigma_1, \sigma_2$ , namely  $\frac{x}{\sigma^*}\sigma$  and its inverse  $\frac{y}{\sigma}\sigma^*$ , where  $x$  is the noisy image,  $y$  is the reconstructed image,  $\sigma^*$  is the actual noise level, and  $\sigma$  is the nearest noise level with known optimal parameters;
- $\sigma$  transformation, where the optimal parameter  $\alpha, \beta$  for the nearest noise level is directly used, and the parameters  $\sigma_1$  and  $\sigma_2$  are transformed by  $\sigma_1^* = \frac{\sigma_1}{\sigma}\sigma^*$  and  $\sigma_2^* = \frac{\sigma_2}{\sigma}\sigma^*$ , where  $\sigma^*$  is the actual noise level, and  $\sigma$  is the nearest noise level with known optimal parameters.

We evaluated the how both schemes generalize around  $\sigma = 50$  (selected as 46 to 54). The corresponding results are presented in Table 13. As shown in the table, both schemes outperform the *DN&DM* and *DM&1.5DN* strategies, indicating the generality of the CMA-ES optimization parameters over a range without the need for repeated optimization.

From Table 12, it is apparent that the denoising stage is responsible for the majority of the time consumption. Therefore, it is advisable to use a fast algorithm, such as the BM3D algorithm implemented on the GPU [18] when using the CMA-ES algorithm to obtain optimal parameters.

**6. Conclusion.** This paper established a model to optimize the denoising and demosaicing pipeline. The optimal pipeline (obtained by CMA-ES) is a *DN<sub>1</sub>&DM* & *DN<sub>2</sub>* scheme with appropriate parameters and *DM&1.5DN* is almost equal to the optimal one when  $\sigma \leq 20$ . Our best performing combination in terms of quality and speed is a *DM&1.5DN* scheme for two reasons: the *DN<sub>1</sub>&DM&DN<sub>2</sub>* scheme gets the best result, but it takes twice as many calculations as *DM&1.5DN*; as



discussed in Section 5.3, in most cases, the noise level for raw images is less than 20. Experiments show a considerable gain. The results of the  $DM&1.5DN$  scheme show a 0.5 to 1 dB gain, when compared with the best  $DN&DM$  strategy. These conclusions apply for moderate noise ( $\sigma \leq 20$ ) but remain valid for high noise, where we nevertheless found a slight improvement of about 0.3 dB for a twice more complex pipeline  $DN_1&DM&DN_2$  with two denoising steps. We also gave a detailed theoretical explanation of why the  $DM&1.5DN$  scheme is superior to the  $DN&DM$  scheme.

We also saw that, unsurprisingly, heavy weight learning-based joint demosaicing and denoising achieves the best performance. However, the above conclusions are still crucial for practical light weight and domain independent application scenarios. They might also inspire the design and training of deep learning algorithms.

**Acknowledgment.** This work was supported by National Natural Science Foundation of China (No. 12061052), Natural Science Fund of Inner Mongolia Autonomous Region (No. 2020MS01002), Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (No. NJYT22090), Innovative Research Team in Universities of Inner Mongolia Autonomous Region (No. NMGIRT2207), Prof. Guoqing Chen’s “111 project” of higher education talent training in Inner Mongolia Autonomous Region, Inner Mongolia University Postgraduate Research and Innovation Programmes (No. 11200-5223737), the network information center of Inner Mongolia University, Office of Naval research grant N00014-17-1-2552, DGA Astrid project n° ANR-17-ASTR-0013-01. Y. Guo and Q. Jin are very grateful to Professor Guoqing Chen for helpful comments and suggestions. The authors are also grateful to the reviewers for their valuable comments and remarks.

## REFERENCES

- [1] A. Abdelhamed, S. Lin and M. S. Brown, A high-quality denoising dataset for smartphone cameras, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, 1692–1700.
- [2] H. Akiyama, M. Tanaka and M. Okutomi, Pseudo four-channel image denoising for noisy cfa raw data, in *Proc. IEEE Int. Conf. Image Process.*, 2015, 4778–4782.
- [3] D. Alleysson, S. Susstrunk and J. Hérault, Linear demosaicing inspired by the human visual system, *IEEE Trans. Image Process.*, **14** (2005), 439–449.
- [4] F. J. Anscombe, The transformation of poisson, binomial and negative-binomial data, *Biometrika*, **35** (1948), 246–254.
- [5] B. E. Bayer, Color imaging array, 1976, US Patent 3,971,065.
- [6] A. Buades, B. Coll and J.-M. Morel, A review of image denoising algorithms, with a new one, *Multiscale Model. Simul.*, **4** (2005), 490–530.
- [7] A. Buades, B. Coll, J.-M. Morel and C. Sbert, Self-similarity driven demosaicking, *Image Processing On Line*, **1** (2011), 51–56.
- [8] P. Chatterjee, N. Joshi, S. B. Kang and Y. Matsushita, Noise suppression in low-light images through joint denoising and demosaicing, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, 321–328.
- [9] G. Chen, F. Zhu and P. A. Heng, An efficient statistical method for image noise level estimation, in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, 477–485.
- [10] Y. Chen, W. Yu and T. Pock, On learning optimized reaction diffusion processes for effective image restoration, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, 5261–5269.
- [11] M. R. Chowdhury, J. Zhang, J. Qin and Y. Lou, Poisson image denoising based on fractional-order total variation, *Inverse Probl. Imaging*, **14** (2020), 77–96, URL [/article/id/16d8110f-d96e-4bf8-bbb1-d4838b09427a](https://doi.org/10.1088/1751-8058/ab9427).
- [12] L. Condat, A simple, fast and efficient approach to demosaicking: Joint demosaicking and denoising, in *Proc. IEEE Int. Conf. Image Process.*, 2010, 905–908.
- [13] L. Condat, A generic proximal algorithm for convex optimization—application to total variation minimization, *IEEE Signal Process. Lett.*, **21** (2014), 985–989.

- [14] L. Condat and S. Mosaddegh, Joint demosaicking and denoising by total variation minimization, in *Proc. IEEE Int. Conf. Image Process.*, 2012, 2781–2784.
- [15] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space, in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, 2007, I – 313–I – 316.
- [16] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, *IEEE Trans. Image Process.*, **16** (2007), 2080–2095.
- [17] A. Danielyan, M. Vehvilainen, A. Foi, V. Katkovnik and K. Egiazarian, Cross-color bm3d filtering of noisy raw data, in *Proc. Int. Workshop Local Non-Local Approx. Image Process.*, 2009, 125–129.
- [18] A. Davy and T. Ehret, Gpu acceleration of nl-means, bm3d and vbm3d, *J. Real-Time Image Process.*, **18** (2021), 57–74.
- [19] W. Dong, M. Yuan, X. Li and G. Shi, Joint demosaicing and denoising with perceptual optimization on a generative adversarial network, *arXiv:1802.04723*.
- [20] E. Dubois, Frequency-domain methods for demosaicking of bayer-sampled color images, *IEEE Signal Process. Lett.*, **12** (2005), 847–850.
- [21] T. Ehret, A. Davy, P. Arias and G. Facciolo, Joint demosaicking and denoising by fine-tuning of bursts of raw images, in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, 8867–8876.
- [22] T. Ehret and G. Facciolo, A study of two CNN demosaicking algorithms, *Image Processing On Line*, **9** (2019), 220–230.
- [23] O. A. Elgendy, A. Gnanasambandam, S. H. Chan and J. Ma, Low-light demosaicking and denoising for small pixels using learned frequency selection, *IEEE Trans. Comput. Imaging*, **7** (2021), 137–150.
- [24] F. Fang, J. Li, Y. Yuan, T. Zeng and G. Zhang, Multilevel edge features guided network for image denoising, *IEEE Trans. Neural Netw. Learn. Syst.*, **32** (2021), 3956–3970.
- [25] R. Franzen, Kodak lossless true color image suite, source: <http://r0k.us/graphics/kodak/>, **4**.
- [26] M. Gharbi, G. Chaurasia, S. Paris and F. Durand, Deep joint demosaicking and denoising, *ACM Trans. Graph.*, **35** (2016), 191:1–12.
- [27] S. Gu, L. Zhang, W. Zuo and X. Feng, Weighted nuclear norm minimization with application to image denoising, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, 2862–2869.
- [28] J. Guan, R. Lai, Y. Lu, Y. Li, H. Li, L. Feng, Y. Yang and L. Gu, Memory-efficient deformable convolution based joint denoising and demosaicing for uhd images, *IEEE Trans. Circuits Syst. Video Technol.*, 1–1.
- [29] J. Guo, Y. Guo, Q. Jin, M. Kwok-Po Ng and S. Wang, Gaussian patch mixture model guided low-rank covariance matrix minimization for image denoising, *SIAM J. Imaging Sci.*, **15** (2022), 1601–1622.
- [30] S. Guo, Z. Liang and L. Zhang, Joint denoising and demosaicking with green channel prior for real-world burst images, *IEEE Trans. Image Process.*, **30** (2021), 6930–6942.
- [31] S. Guo, Z. Yan, K. Zhang, W. Zuo and L. Zhang, Toward convolutional blind denoising of real photographs, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, 1712–1722.
- [32] Y. Guo, A. Davy, G. Facciolo, J.-M. Morel and Q. Jin, Fast, nonlocal and neural: A lightweight high quality solution to image denoising, *IEEE Signal Process. Lett.*, **28** (2021), 1515–1519.
- [33] Y. Guo, Q. Jin, J.-M. Morel, T. Zeng and G. Facciolo, Joint demosaicking and denoising benefits from a two-stage training strategy, *J. Comput. Appl. Math.*, 115330.
- [34] J. F. Hamilton Jr and J. E. Adams Jr, Adaptive color plan interpolation in single sensor color electronic camera, 1997, US Patent 5,629,734.
- [35] N. Hansen and A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation, in *Proc. IEEE Int. Conf. Evol. Comput.*, 1996, 312–317.
- [36] M. K. Heris, Implementation of covariance matrix adaptation evolution strategy (cma-es) in matlab, <https://yarpiz.com/235/yypea108-cma-es>, 2015.
- [37] M. Hintermüller and M. Rincon-Camacho, An adaptive finite element method in  $l^2$ -tv-based image denoising, *Inverse Probl. Imaging*, **8** (2014), 685–711, URL [/article/id/aa8c96bc-8026-4f22-a13f-238dbbfaed8d](https://doi.org/10.1088/1751-8006/8/4/045501).
- [38] K. Hirakawa and T. Parks, Joint demosaicing and denoising, *IEEE Trans. Image Process.*, **15** (2006), 2146–2157.
- [39] H. Hu, J. Froment, B. Wang and X. Fan, Spatial-frequency domain nonlocal total variation for image denoising, *Inverse Probl. Imaging*, **14** (2020), 1157–1184, URL [/article/id/0c194ea5-31fb-4cb6-b665-377c85db6263](https://doi.org/10.1088/1751-8006/14/0c/0c194ea5-31fb-4cb6-b665-377c85db6263).

- [40] Q. Jin, G. Facciolo and J. Morel, A review of an old dilemma: Demosaicking first, or denoising first?, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, 2169–2179.
- [41] Q. Jin, I. Grama, C. Kervrann and Q. Liu, Nonlocal means and optimal weights for noise removal, *SIAM J. Imaging Sci.*, **10** (2017), 1878–1920.
- [42] Q. Jin, I. Grama and Q. Liu, Convergence theorems for the non-local means filter, *Inverse Probl. Imaging*, **12** (2018), 853–881.
- [43] Q. Jin, Y. Guo, J.-M. Morel and G. Facciolo, A Mathematical Analysis and Implementation of Residual Interpolation Demosaicking Algorithms, *Image Processing On Line*, **11** (2021), 234–283.
- [44] Y. Jin, J. Jost and G. Wang, A new nonlocal variational setting for image processing, *Inverse Probl. Imaging*, **9** (2015), 415–430, URL [/article/id/a53e48fd-e7c0-48c2-8e2c-6d4668ca4774](#).
- [45] O. Kalevo and H. Rantanen, Noise reduction techniques for bayer-matrix images, in *Proc. Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III*, vol. 4669, 2002, 348–359.
- [46] D. Khashabi, S. Nowozin, J. Jancsary and A. W. Fitzgibbon, Joint demosaicing and denoising via learned nonparametric random fields, *IEEE Trans. Image Process.*, **23** (2014), 4968–4981.
- [47] D. Kiku, Y. Monno, M. Tanaka and M. Okutomi, Residual interpolation for color image demosaicking, in *Proc. IEEE Int. Conf. Image Process.*, 2013, 2304–2308.
- [48] D. Kiku, Y. Monno, M. Tanaka and M. Okutomi, Minimized-laplacian residual interpolation for color image demosaicking, in *Proc. Digital Photography X*, vol. 9023, 2014, 90230L.
- [49] D. Kiku, Y. Monno, M. Tanaka and M. Okutomi, Beyond color difference: Residual interpolation for color image demosaicking, *IEEE Trans. Image Process.*, **25** (2016), 1288–1300.
- [50] F. Kokkinos and S. Lefkimmiatis, Iterative joint image demosaicking and denoising using a residual denoising network, *IEEE Trans. Image Process.*, **28** (2019), 4177–4188.
- [51] M. Lebrun, M. Colom, A. Buades and J. M. Morel, Secrets of image denoising cuisine, *Acta Numer.*, **21** (2012), 475–576.
- [52] M. Lebrun, A. Buades and J.-M. Morel, A nonlocal bayesian image denoising algorithm, *SIAM J. Imaging Sci.*, **6** (2013), 1665–1688.
- [53] M. Lee, S. Park and M. Kang, Denoising algorithm for cfa image sensors considering inter-channel correlation, *Sensors*, **17** (2017), 1236.
- [54] J. Liang, J. Li, Z. Shen and X. Zhang, Wavelet frame based color image demosaicing, *Inverse Probl. Imaging*, **7** (2013), 777–794, URL [/article/id/d7c7ce92-a146-44ba-8606-6abcaa24ba25](#).
- [55] L. Liu, X. Jia, J. Liu and Q. Tian, Joint demosaicing and denoising with self guidance, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, 2237–2246.
- [56] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, Non-local sparse models for image restoration, in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, 2272–2279.
- [57] H. Malvar, L. wei He and R. Cutler, High-quality linear interpolation for demosaicing of bayer-patterned color images, in *Proc. IEEE Int. Conf. Acoust. Speech. Signal. Process.*, vol. 3, 2004, iii–485.
- [58] Y. Monno, D. Kiku, M. Tanaka and M. Okutomi, Adaptive residual interpolation for color and multispectral image demosaicking, *Sensors*, **17** (2017), 2787.
- [59] A. Mosleh, A. Sharma, E. Onzon, F. Mannan, N. Robidoux and F. Heide, Hardware-in-the-loop end-to-end optimization of camera image processing pipelines, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, 7529–7538.
- [60] Y. I. Ohta, T. Kanade and T. Sakai, Color information for region segmentation, *Computer Graphics & Image Processing*, **13** (1980), 222–241.
- [61] D. Paliy, M. Trimeche, V. Katkovnik and S. Alenius, Demosaicing of noisy data: spatially adaptive approach, in *Proc. Image Processing: Algorithms and Systems V*, vol. 6497, 2007, 179–190.
- [62] S. H. Park, H. S. Kim, S. Linsel, M. Parmar and B. A. Wandell, A case for denoising before demosaicking color filter array data, in *Proc. Conf. Rec. Asilomar Conf. Signals Syst. Comput.*, 2009, 860–864.
- [63] S. Patil and A. Rajwade, Poisson noise removal for image demosaicing., in *Proc. Br. Mach. Vis. Conf.*, 2016, 33.1–33.10.
- [64] I. Pekkucuksen and Y. Altunbasak, Gradient based threshold free color filter array interpolation, in *Proc. IEEE Int. Conf. Image Process.*, 2010, 137–140.

- [65] T. Plötz and S. Roth, Benchmarking denoising algorithms with real photographs, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, 2750–2759.
- [66] J. Portilla, V. Strela, M. Wainwright and E. Simoncelli, Image denoising using scale mixtures of gaussians in the wavelet domain, *IEEE Trans. Image Process.*, **12** (2003), 1338–1351.
- [67] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D: Nonlinear Phenomena*, **60** (1992), 259–268.
- [68] N.-S. Syu, Y.-S. Chen and Y.-Y. Chuang, Learning deep convolutional networks for demosaicing, *arXiv:1802.03769*.
- [69] R. Tan, K. Zhang, W. Zuo and L. Zhang, Color image demosaicking via deep residual learning, in *Proc. IEEE Int. Conf. Multimedia Expo*, 2017, 793–798.
- [70] J. Wu, R. Timofte and L. Van Gool, Demosaicing based on directional difference regression and efficient regression priors, *IEEE Trans. Image Process.*, **25** (2016), 3862–3874.
- [71] X. Wu and L. Zhang, Temporal color video demosaicking via motion estimation and data fusion, *IEEE Trans. Circuits Syst. Video Technol.*, **16** (2006), 231–240.
- [72] W. Xing and K. Egiazarian, End-to-end learning for joint image demosaicing, denoising and super-resolution, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, 3507–3516.
- [73] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang and L. Shao, CycleISP: Real image restoration via improved data synthesis, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, 2693–2702.
- [74] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, *IEEE Trans. Image Process.*, **26** (2017), 3142–3155.
- [75] K. Zhang, W. Zuo and L. Zhang, FFDNet: Toward a fast and flexible solution for cnn-based image denoising, *IEEE Trans. Image Process.*, **27** (2018), 4608–4622.
- [76] L. Zhang, R. Lukac, X. Wu and D. Zhang, PCA-based spatially adaptive denoising of cfa images for single-sensor digital cameras, *IEEE Trans. Image Process.*, **18** (2009), 797–812.
- [77] L. Zhang and X. Wu, Color demosaicking via directional linear minimum mean square-error estimation, *IEEE Trans. Image Process.*, **14** (2005), 2167–2178.
- [78] L. Zhang, X. Wu, A. Buades and X. Li, Color demosaicking by local directional interpolation and nonlocal adaptive thresholding, *J. Electron. Imaging*, **20** (2011), 023016.
- [79] X. Zhang, M.-T. Sun, L. Fang and O. C. Au, Joint denoising and demosaicking of noisy cfa images based on inter-color correlation, in *Proc. IEEE Int. Conf. Acoust. Speech. Signal. Process.*, 2014, 5784–5788.
- [80] D. Zoran and Y. Weiss, From learning models of natural image patches to whole image restoration, in *Proc. Int. Conf. Comput. Vis.*, 2011, 479–486.

Received xxxx 2022; revised xxxx 2023; early access xxxx 20xx.