

# Chapter 11

## Three Approaches to Improve Denoising Results that Do Not Involve Developing New Denoising Methods



Gabriela Ghimpeteanu, Thomas Batard, Stacey Levine  
and Marcelo Bertalmío

**Abstract** Image denoising has been a topic extensively investigated over the last three decades and, as repeatedly shown in this book, denoising algorithms have become incredibly good, so much so that many researchers have started questioning the need to further pursue this line of research. In this chapter, we argue that there is indeed room for improvement of denoising results, and we propose three different avenues to explore, none of which requires the development of new denoising methods. First, we describe how it can be better to denoise a transform of the noisy image rather than denoise the noisy image directly. We mention several possible transforms, and an open problem is to find a transform that is optimal for denoising, according to a proper image quality metric. Next, we point out the importance of having a proper noise model for JPEG pictures, so that a variance stabilization transform can be developed that transforms noise in JPEG images into additive white Gaussian noise, enabling existing denoising methods to be properly applied to the JPEG case. Finally, we highlight the fact that while virtually all denoising methods are optimized and validated in terms of the PSNR or SSIM measures, these metrics are not well correlated with perceived image quality, and therefore, it could be best

---

G. Ghimpeteanu (✉) · M. Bertalmío  
Universidad Pompeu Fabra, Barcelona, Spain  
e-mail: gabriela.ghimpeteanu@upf.edu

M. Bertalmío  
e-mail: marcelo.bertalmio@upf.edu

T. Batard  
TU Kaiserslautern, Kaiserslautern, Germany  
e-mail: tomas.batard@gmail.com

S. Levine  
Duquesne University, Pittsburgh, PA, USA  
e-mail: sel@mathcs.duq.edu

to optimize the parameter values of denoising methods according to subjective testing. A remaining challenge is to develop perceptually based image quality metrics that match observer preference.

## 11.1 Introduction

In this chapter we propose, in order to improve denoising results, to explore three different avenues that do not require the development of new denoising methods.

First, we review recent works that improve the performance of denoising algorithms by applying them to transforms of image data instead of applying them directly to the noisy image. An open challenge is then to find a transform that is optimal for the denoising problem, according to a proper image quality metric.

Second, we show how not only the performance but also the ranking of denoising algorithms is different in the real noise scenario than when working under the common assumption that noise is additive white Gaussian (AWG) of known variance which is fixed and independent from the image values. A second way to improve denoising results would then be to develop a noise model for JPEG pictures and a corresponding variance stabilization transformation, so that existing denoising methods that assume AWG noise can be properly applied to the JPEG case.

Finally, we note that although the PSNR and SSIM error measures are not correlated with perceived image quality, virtually all denoising methods are optimized and validated in terms of these metrics. This suggests a third approach to improve denoising results, that of developing a perceptually based image quality metric that matches observer preference, or using subjective experiments to select the optimal parameter values for denoising methods.

## 11.2 Denoise a Transform of an Image Instead of the Image Itself

There are often benefits to processing a linear or nonlinear transform of an image rather than processing the observed image data directly. In the context of image denoising, this has traditionally taken the form of thresholding Fourier or wavelet coefficients, which then has a denoising effect on the underlying image data. The line of research we follow here is different in that we specifically *apply an image denoising method* to a transform of an image which in turn has a denoising effect on the original image data. The latter differs from the former in several ways. First, the choice of processing applied to the transform is different, the latter being one that is borrowed from algorithms for denoising the image data directly while the former, such as thresholding, is not. Second, the latter may require a specially designed mechanism for reconstructing the denoised image, particularly for nonlinear transforms.

The first instance we know of applying an image denoising technique to an image transform is the work of Lysaker et al. [30] in which the authors propose using a constrained total variation (TV) minimization problem to denoise the unit normal vector field of a noisy image surface. The denoised image is then reconstructed using a variational approach whose solution has a unit normal field that matches the results of this constrained TV problem. Similar denoising strategies are presented in [41] and [20] where the unit tangent field to an image surface is denoised, allowing for a mathematically sound model. The approach in [30] is also directly related to the Bregman iterative algorithm of Osher et al. [38].

A similar approach was proposed by Bertalmío and Levine [8]. Justified by the fact that the curvature of the level lines of a gray-level image has a higher SNR along likely edges than the image itself (in the context of AWG noise), they demonstrate that reconstructing an image from its denoised level line curvature is consistently more effective than denoising the image directly. This is confirmed by experiments with four denoising methods: TV denoising performed through both gradient descent [43] and the Bregman iterative algorithm [38], orientation matching using smoothed unit tangents [20], nonlocal means (NLM) [11] and block-matching and 3D filtering (BM3D) [14]. A variational method was successful for the reconstruction step.

Batard and Berthier [5] introduced a moving frame approach in the process of describing a Fourier theory for  $n$ -channel images which takes into account the local geometry of an image. Their orthonormal moving frame in  $\mathbb{R}^{n+2}$ , defined over the image domain, consists of two vector fields that are tangent to the image graph and  $n$  components that are normal to the surface. Their idea was to construct the components of an image in this moving frame, compute the standard 2D Fourier transform of each of the  $n+2$  components, apply a different Gaussian kernel to each one, and finally project back. By applying Euclidean heat diffusion to each component, the output is a filtered image that retains its local geometry after the diffusion step.

Batard and Bertalmío [6, 7] used this moving frame approach for image denoising. Instead of directly applying a denoising method to an image, they proposed to apply it to its components in the moving frame described above. The authors used the vectorial extension of the total variation-based denoising method of Rudin et al. [43] proposed by Blomgren and Chan [9], and the vectorial total variation (VTV) denoising method of Bresson and Chan [10]. In both cases, this strategy produced better results in terms of PSNR than denoising the image directly with these approaches.

In this section, we revisit the work of [16] and detail how to improve the result of a denoising method by denoising the components of an image in a moving frame, as in [5–7], instead of denoising the original noisy image directly. We note a theoretical analysis of why with this approach we can expect cleaner results with better preserved details, regardless of denoising algorithm. We validate the consistency of this approach by showing how the moving frame strategy brings an improvement in terms of PSNR and SSIM for three different noise removal techniques: a local variational method (VTV, [10]), a patch-based method (NLM [11]), and a method combining patch based processing with filtering in the spectral domain (BM3D [14]). The approach in [16] has the advantage of simplicity in the reconstruction step which is expressed as a matrix transform, in comparison to the similar curvature-based strat-

egy in [8] or the vector fields smoothing techniques of [20, 30, 38, 41] which require solving a second- or third-order PDE evolution equation for the reconstruction step.

### 11.2.1 Image Decomposition in a Moving Frame

We denote a gray-level image by  $I: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ , and the standard coordinate system of  $\mathbb{R}^2$  by  $(x, y)$ .  $I_x$  and  $I_y$  are the derivatives of  $I$  with respect to  $x$  and  $y$ , respectively, and  $\nabla I$  is the gradient of  $I$ . We construct an image decomposition model for  $I$  comprised of two steps. First, construct an orthonormal moving frame  $(Z_1, Z_2, N)$  of  $(\mathbb{R}^3, \|\cdot\|_2)$  over  $\Omega$  that takes into account the local geometry of  $I$ . Next, calculate the components  $(J^1, J^2, J^3)$  of the  $\mathbb{R}^3$ -valued function  $(0, 0, I)$  in that moving frame.

We use  $\mu I$  (a scaled version of  $I$ , with  $\mu \in ]0, 1]$ ), and its graph, given by the surface  $S$  in  $\mathbb{R}^3$  parametrized by

$$\psi : (x, y) \mapsto (x, y, \mu I(x, y)). \tag{11.1}$$

We construct an orthonormal moving frame  $(Z_1, Z_2, N)$  by choosing  $Z_1$  to be the vector field tangent to the surface  $S$  that points in the direction of the steepest slope at each point of  $S$ , and  $Z_2$  to be the vector field tangent to  $S$  that points in the direction of the lowest slope at each point of  $S$ . To complete the orthonormal frame, the component  $N$  is the unit normal to the surface.

This can be realized by considering the gradient of  $\mu I$ ,  $z_1 = (\mu I_x, \mu I_y)^T$ , and the vector indicating the direction of the level lines of  $\mu I$ ,  $z_2 = (-\mu I_y, \mu I_x)^T$ . On smooth regions of  $I$  where  $I_x(x, y) = I_y(x, y) = 0$ , we fix  $z_1 = (1, 0)^T$  and  $z_2 = (0, 1)^T$ . Then  $Z_1$  and  $Z_2$  are defined as

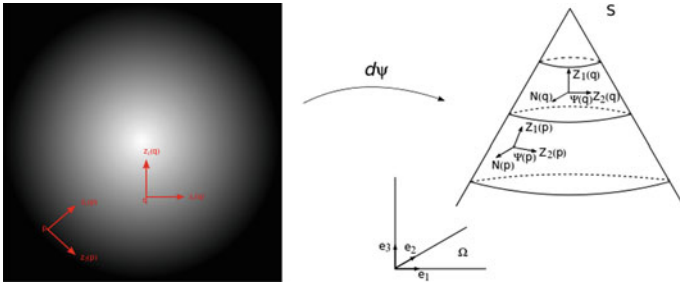
$$Z_i = \frac{d\psi(z_i)}{\|d\psi(z_i)\|_2}, \quad i = 1, 2, \tag{11.2}$$

where  $\psi$  maps vector fields from  $\Omega$  to tangent vector fields of  $S$ , and  $d\psi$  denotes its differential. The unit normal  $N$  is given by the vectorial product between  $Z_1$  and  $Z_2$ .

More explicitly, the coordinates of the vector fields  $Z_1, Z_2, N$  are given by the first, second, and third columns of the matrix field

$$P = \begin{pmatrix} \frac{I_x}{\sqrt{|\nabla I|^2(1 + \mu^2|\nabla I|^2)}} & \frac{-I_y}{|\nabla I|} & \frac{-\mu I_x}{\sqrt{1 + \mu^2|\nabla I|^2}} \\ \frac{I_y}{\sqrt{|\nabla I|^2(1 + \mu^2|\nabla I|^2)}} & \frac{I_x}{|\nabla I|} & \frac{-\mu I_y}{\sqrt{1 + \mu^2|\nabla I|^2}} \\ \frac{\mu|\nabla I|^2}{\sqrt{|\nabla I|^2(1 + \mu^2|\nabla I|^2)}} & 0 & \frac{1}{\sqrt{1 + \mu^2|\nabla I|^2}} \end{pmatrix} \tag{11.3}$$

Figure 11.1 shows the moving frames  $(z_1, z_2)$  and  $(Z_1, Z_2, N)$  associated to a simple image. On the left, the figure illustrates the moving frame  $(z_1, z_2)$  at the points  $p$  and



**Fig. 11.1** Moving frame encoding the local geometry of a gray-level image. Left: original gray-level image and a moving frame  $(z_1, z_2)$  indicating the direction of the gradient and the level line of the image at two points  $p$  and  $q$  of the image domain  $\Omega$ . Right: the orthonormal moving frame  $(Z_1, Z_2, N)$  of  $(\mathbb{R}^3, \|\cdot\|_2)$  over  $\Omega$  indicating the direction of the steepest and lowest slopes of the surface  $S$ , for some smoothing parameter  $\mu$ , at the points  $\psi(p)$  and  $\psi(q)$

$q$  of the domain  $\Omega$ , and on the right, it shows the induced moving frame  $(Z_1, Z_2, N)$  constructed on the surface  $S$  at the points  $\psi(p)$  and  $\psi(q)$ .

If  $(e_1, e_2, e_3)$  represents the orthonormal frame of  $(\mathbb{R}^3, \|\cdot\|_2)$ , where  $e_1 = (1, 0, 0)$ ,  $e_2 = (0, 1, 0)$ , and  $e_3 = (0, 0, 1)$ , then the matrix  $P$  in (11.3) can be seen as the frame change field from  $(e_1, e_2, e_3)$  to  $(Z_1, Z_2, N)$ . More precisely, the components of the  $\mathbb{R}^3$ -valued function  $(0, 0, I)$  in the new frame are given by  $(J^1, J^2, J^3)$ , where

$$(J^1, J^2, J^3)^T = P^{-1}(0, 0, I)^T. \tag{11.4}$$

From (11.3) and (11.4), we see that these components can be explicitly expressed as

$$J^1(I) = \frac{\mu I |\nabla I|}{\sqrt{1 + \mu^2 |\nabla I|^2}}, \quad J^2(I) = 0, \quad \text{and} \quad J^3(I) = \frac{I}{\sqrt{1 + \mu^2 |\nabla I|^2}}. \tag{11.5}$$

Figure 11.2 illustrates the gray-level test image “castle” and its components  $J^1$  and  $J^3$  computed for  $\mu = 0.05$ . The component  $J^1$  contains edge and texture information due to its inclusion of the norm of the image gradient. The component  $J^3$  better reproduces the original image, from which the norm of the gradient has been diminished.

The multichannel case involves embedding an  $n$ -channel image into  $\mathbb{R}^{n+2}$ , then using similarly motivated choices for  $Z_1$  and  $Z_2$  with extra care taken in the construction of the remaining normal vector fields. Details can be found in [16].

An important role is given to the parameter  $\mu$ . It represents a smoothing parameter for the moving frame associated to the image  $I$ . In the next sections, we analyze just how crucial the value of this parameter is for image denoising.



**Fig. 11.2** From left to right: gray-level image “castle”, component  $J^1$ , and component  $J^3$

### 11.2.2 Application to Image Denoising

We propose denoising the components of an image in the moving frame described above instead of denoising the image directly. The effect on image denoising using these two approaches can be compared by performing the following experiment:

1. Denoise  $I$  with some method and denote the output image  $I_{den}$ .
2. Compute the components of  $I$  in the moving frame. Apply the same denoising method to the components to obtain the processed components. Then, apply the inverse frame change matrix field to the processed components, obtaining a reconstructed image denoted by  $I_{denMF}$ . In the grayscale case, this can be explicitly expressed as

$$I_{denMF} = P_{13}(I)J^1(I)_{den} + P_{33}(I)J^3(I)_{den}. \quad (11.6)$$

3. Compare  $I_{den}$  and  $I_{denMF}$  using PSNR, SSIM or any image quality metric.

The extension from single to multichannel denoising algorithms is not always straightforward. Depending on the denoising method, there are several ways to do this, given by the use of different color spaces and the manner in which to apply the algorithm (channel-wise, only to selected channels, or vectorially). Details can be found in [16].

It is interesting to note that the denoising approach introduced above can in fact be used with any moving frame. Several choices were previously analyzed in [7], showing a similar output quality is attained when  $Z_1, Z_2$  are any randomly chosen, but orthonormal, vector fields in the tangent planes of the surface parametrized by (11.1) for gray-level images. However, when  $Z_1, Z_2$  are not in the tangent space, the results are of low quality.

**Table 11.1** Average values of the PSNR for the components  $J^1$ ,  $J^3$  and the image  $I$  over the Kodak database for different noise levels and values of the parameter  $\mu$

Noise level	Function	$\mu = 1$	$\mu = 0.1$	$\mu = 0.01$	$\mu = 0.005$	$\mu = 0.001$	$\mu = 0.0001$
$\sigma = 5$	Component $J^1$	20.51	20.09	34.37	37.84	40.17	40.31
	Component $J^3$	18.56	26.02	34.24	34.22	34.19	34.19
	Image $I$	34.19	34.19	34.19	34.19	34.19	34.19
$\sigma = 10$	Component $J^1$	19.34	15.96	28.21	31.51	33.84	33.97
	Component $J^3$	16.94	19.84	28.27	28.24	28.21	28.21
	Image $I$	28.21	28.21	28.21	28.21	28.21	28.21
$\sigma = 15$	Component $J^1$	18.32	14.16	24.44	27.79	30.09	30.22
	Component $J^3$	16.32	16.93	24.80	24.77	24.73	24.73
	Image $I$	24.73	24.73	24.73	24.73	24.73	24.73
$\sigma = 20$	Component $J^1$	17.37	13.10	21.86	25.12	27.38	27.51
	Component $J^3$	15.98	15.22	22.38	22.33	22.28	22.27
	Image $I$	22.27	22.27	22.27	22.27	22.27	22.27
$\sigma = 25$	Component $J^1$	16.47	12.36	19.89	23.03	25.25	25.38
	Component $J^3$	15.77	14.10	20.50	20.44	20.37	20.37
	Image $I$	20.37	20.37	20.37	20.37	20.37	20.37

### 11.2.3 The Noise Level Is Higher on the Intensity Values of a Gray-Level Image Than on Its Components in a Well-Chosen Moving Frame

In this section, we study how for carefully selected  $\mu$  values, the components  $J^1(I)$  and  $J^3(I)$  of a gray-level image  $I$  in the moving frame (11.3), given by (11.4) and (11.5), are less degraded by AWG image noise than the image itself.

Assume  $I = a + n$  is a gray-level image obtained by adding to the image  $a$  Gaussian noise  $n$  of mean zero and standard deviation  $\sigma$ . Before delving into a more formal analysis, we consider an experiment in which we calculate the PSNR values of the components  $J^1(I)$  and  $J^3(I)$  of the images from the Kodak database [2], for noise levels  $\sigma = 5, 10, 15, 20, 25$  and  $\mu = 1.0, 0.1, 0.01, 0.005, 0.001, 0.0001$ . The results of this experiment are reported in Table 11.1.

Notice that the PSNR values of the components are consistently larger than the PSNR of the image for sufficiently small  $\mu$ . Specifically, the components are generally “less noisy” when  $\mu \in ]0, 0.005]$  for all noise levels considered. For  $\sigma = 5, 10$ , the upper bound of 0.005 can be increased to 0.01.

While the values in Table 11.1 were computed across the entire image, we can more formally study this behavior by considering locations of likely image contours and homogeneous regions separately.

### 11.2.3.1 Edges

As in [8], we attain the following conclusion comparing the PSNR of a grayscale image and its moving frame components along likely image contours.

**Proposition 1** *Using central differences for the approximation of  $\nabla I$  and  $\mu > 0$ , at locations in the image domain where  $|\nabla a| \gg |\nabla n|$ , likely edges of  $I$ ,*

$$PSNR(J^1(I)) \geq PSNR(I) \text{ and } PSNR(J^3(I)) > PSNR(I).$$

The proof, detailed in [16], uses the fact that the explicit representations for the moving frame components  $J^1(I)$  and  $J^3(I)$  given in (11.5) make it possible to approximate their noise as additive along likely contours, leading to the estimates

$$PSNR(J^1(I)) = 20 \log_{10} \left( 255 \frac{127.5\sqrt{2}\mu}{\sqrt{1+2\mu^2}(127.5)^2} \times \frac{\sqrt{1+\mu^2|\nabla I|^2}}{\mu|\nabla I|\sigma} \right) \quad (11.7)$$

$$\geq 20 \log_{10} \left( \frac{255}{\sigma} \right) = PSNR(I) \quad (11.8)$$

and

$$PSNR(J^3(I)) = 20 \log_{10} \left( 255 \frac{\sqrt{1+\mu^2|\nabla I|^2}}{\sigma} \right) \quad (11.9)$$

$$> 20 \log_{10} \left( \frac{255}{\sigma} \right) = PSNR(I). \quad (11.10)$$

To better understand the role of  $\mu$  along likely contours of  $I$ , (11.9) indicates that  $PSNR(J^3(I))$  is a strictly increasing function of  $\mu$ , tending to  $+\infty$  as  $\mu \rightarrow +\infty$ , and to  $PSNR(I)$  as  $\mu \rightarrow 0$ . On the other hand, (11.7) indicates that  $PSNR(J^1(I))$  is a decreasing function of  $\mu$ , tending to  $PSNR(I)$  as  $\mu \rightarrow +\infty$ , and tending to  $20 \log_{10} \left( \frac{255 \times 127.5 \sqrt{2}}{|\nabla I| \sigma} \right)$  when  $\mu \rightarrow 0$ . Thus, we infer that along image contours, the larger the value of  $\mu$ , the better the estimation of the clean component  $J^3(a)$ , while the smaller the value of  $\mu$ , the better the estimation of the clean component  $J^1(a)$ .

Furthermore, since  $|\nabla I| \approx |\nabla a|$  at contours, we obtain (see (11.3))

$$P_{31}(I) \approx P_{31}(a) \text{ and } P_{33}(I) \approx P_{33}(a). \quad (11.11)$$

Thus (11.6) and (11.11) imply that, along likely contours,

$$\begin{aligned} I_{denMF} &= P_{13}(I)J^1(I)_{den} + P_{33}(I)J^3(I)_{den} \\ &\approx P_{13}(a)J^1(I)_{den} + P_{33}(a)J^3(I)_{den}. \end{aligned} \quad (11.12)$$



From Proposition 1, we deduce that  $J^1(I)_{den}$  and  $J^3(I)_{den}$  are better estimates of  $J^1(a)$  and  $J^3(a)$  than  $I_{den}$  is of  $a$  at these locations. Therefore, from (11.12) and the fact that

$$a = P_{13}(a)J^1(a) + P_{33}(a)J^3(a)$$

we conclude that at likely image contours,  $I_{denMF}$  is a better approximation of  $a$  than  $I_{den}$ . Thus, the value for the parameter  $\mu$  that gives a better reconstruction of image contours in the clean image is strictly positive, since for  $\mu = 0$  we get  $I_{denMF} = I_{den}$ .

### 11.2.3.2 Homogeneous Regions

In this section, we consider the case of homogeneous or slowly varying regions, where  $|\nabla a| \ll |\nabla n|$ . At these locations, we obtain

$$J^1(I) \approx \frac{\mu I |\nabla n|}{\sqrt{1 + \mu^2 |\nabla n|^2}} \quad \text{and} \quad J^3(I) \approx \frac{I}{\sqrt{1 + \mu^2 |\nabla n|^2}}. \quad (11.13)$$

Note that for  $\mu > 0$ , the range, and therefore the variations, of  $J^1(I)$  and  $J^3(I)$  is diminished compared to those of  $I$ .

Moreover, if  $|\nabla a| \ll |\nabla n|$  then

$$\begin{aligned} I_{denMF} &= P_{13}(I)J^1(I)_{den} + P_{33}(I)J^3(I)_{den} \\ &\approx \frac{\mu |\nabla n|}{\sqrt{1 + \mu^2 |\nabla n|^2}} J^1(I)_{den} + \frac{1}{\sqrt{1 + \mu^2 |\nabla n|^2}} J^3(I)_{den}. \end{aligned}$$

Therefore, as  $|\nabla n|$  increases,  $J^1(I)_{den}$  has a larger weight in the reconstruction. This is advantageous, as the results from Table 11.1 suggest that for small values of  $\mu > 0$ , across the entire image  $I$ , including edges, textures, and homogeneous regions, we see a clear trend that

$$PSNR(J^1(I)) > PSNR(I) \quad \text{and} \quad PSNR(J^3(I)) \approx PSNR(I). \quad (11.14)$$

While a formal proof is not trivial, it is reasonable to infer that the proposed moving frame approach should be successful for a carefully chosen  $\mu$  value in homogeneous areas as well. Combining this argument with Proposition 1, one can expect that across the entire image,  $I_{denMF}$  should be at least as good as, if not better than,  $I_{den}$ .

## 11.2.4 Experiments

We report results comparing  $\mu = 0$  and  $\mu > 0$  for three denoising methods, VTV, NLM and BM3D. Although automating  $\mu$  could be a challenge, we found that for the nonlocal algorithms NLM and BM3D a fairly consistent value of  $\mu = 0.001$  gave

**Table 11.2** The gray-level case: variational approach

Approach\noise variance	5	10	15	20	25
PSNR of $I_{den} = ROF(I)$	35.39	31.51	29.48	28.15	27.17
PSNR of $I_{denMF} = P_{13}(I)ROF(J^1(I)) + P_{33}(I)ROF(J^3(I))$	<b>36.36</b>	<b>32.23</b>	<b>30.04</b>	<b>28.60</b>	<b>27.49</b>
SSIM index of $I_{den} = ROF(I)$	93.76	87.07	81.91	77.63	74.12
SSIM index of $I_{denMF} = P_{13}(I)ROF(J^1(I)) + P_{33}(I)ROF(J^3(I))$	<b>94.61</b>	<b>88.37</b>	<b>83.22</b>	<b>78.71</b>	<b>74.78</b>
$\mu$ value used to compute $I_{denMF}$	0.008	0.005	0.005	0.004	0.004

the best results, independent of image content, noise level, and the measure chosen for the denoising evaluation. This is not the case for the local method VTV, for which the optimal  $\mu$  value was highly related to the noise level. The results for VTV were obtained with noise-dependent, optimal values of  $\mu$  (found experimentally), while those for NLM, BM3D all used  $\mu = 0.001$ .

Tables 11.2 and 11.3 summarize the average PSNR and SSIM values of comparing the final image denoising results when the same denoising algorithm is applied to an image directly to obtain  $I_{den}$ , Eq. (11.6) with  $\mu = 0$ , as opposed to its moving frame components to obtain  $I_{denMF}$ , Eq. (11.6) with  $\mu > 0$ . The results are averaged across the grayscale versions of all images in the Kodak database; analogous PSNR and SSIM results for the Kodak database color images are reported in [16] with similarly chosen values of  $\mu$ . It is important to note that while the differences in these image quality metrics diminish as the more powerful denoising techniques are applied, e.g., BM3D, there is still a consistent increase across all noise levels, and this level of increase reflects previously reported mean squared error optimality bounds [28, 29]. Furthermore, while the differences in the image quality metrics may be leveling off, the difference in the resulting image details when comparing these approaches is notable, with more accurate details preserved in the result of denoising the geometrically motivated moving frame components. Visual examples comparing all three denoising algorithms, VTV, NLM and BM3D, can be found in Fig. 11.3.

### 11.2.5 Research Avenue to Explore

We have just seen that it often makes sense to denoise the transform of an image such as its moving frame components, level line curvature, or unit normal vector field. Still, it would be interesting to find a transform that is **optimal** for image denoising. The optimality should be evaluated according to some criterion (not necessarily higher PSNR), and the noise model should carefully reflect the image acquisition model as well; the analysis in this section has assumed AWG noise for simplicity, but the importance of using the correct noise model cannot be overstated, as we detail in the following section.

**Table 11.3** The gray-level case: patch-based approaches ( $\mu = 0.001$ )

Approach\noise variance	5	10	15	20	25
PSNR of $I_{den} = NLM(I)$	37.41	33.38	31.05	30.04	28.91
PSNR of $I_{denMF} = P_{13}(I)NLM(J^1(I)) + P_{33}(I)NLM(J^3(I))$	<b>37.52</b>	<b>33.59</b>	<b>31.57</b>	<b>30.12</b>	<b>29.00</b>
SSIM index of $I_{den} = NLM(I)$	94.96	88.71	82.17	80.34	75.94
SSIM index of $I_{denMF} = P_{13}(I)NLM(J^1(I)) + P_{33}(I)NLM(J^3(I))$	<b>95.11</b>	<b>89.54</b>	<b>85.37</b>	<b>81.03</b>	<b>76.95</b>
PSNR of $I_{den} = BM3D(I)$	38.23	34.34	32.26	30.89	29.88
PSNR of $I_{denMF} = P_{13}(I)BM3D(J^1(I)) + P_{33}(I)BM3D(J^3(I))$	<b>38.25</b>	<b>34.38</b>	<b>32.31</b>	<b>30.93</b>	<b>29.92</b>
SSIM index of $I_{den} = BM3D(I)$	95.71	91.38	87.52	84.19	81.32
SSIM index of $I_{denMF} = P_{13}(I)BM3D(J^1(I)) + P_{33}(I)BM3D(J^3(I))$	<b>95.74</b>	<b>91.49</b>	<b>87.71</b>	<b>84.38</b>	<b>81.44</b>

### 11.3 Have a Proper Noise Model

A key aspect that is overlooked when suggesting that denoising is an almost solved problem is the following: *most denoising methods in the literature are based on modeling noise as being additive and independent from the image data.* In fact, validations and comparisons are normally performed by taking clean photographs as ground truth, creating noisy versions with AWG noise of known variance (fixed and independent from the image values), applying denoising algorithms to them, and comparing each denoised result with the corresponding clean ground truth image using an objective metric such as PSNR. *Everything is dependent on this AWG assumption:* the design of the denoising algorithms, their ranking according to the quality of the outputs, even the computation of the optimality bounds that suggest that state-of-the-art algorithms are close to optimal.

It is well known that noise in regular output images, in JPEG format, is not AWG, but despite this fact, the vast majority of the denoising literature implicitly assumes that this difference should not have an impact on how we address the denoising problem, nor on how results are validated or methods compared. A few exceptions in the literature are, for instance:

- Nam et al. [37] propose a quite complex cross-channel noise model for JPEG images, and a neural network to estimate the noise model parameters per camera model and per ISO sensitivity value; this model can then be used to improve denoising results.



**Fig. 11.3** Row 1:  $VTV$  result for image 13 [2] with  $\sigma = 15$ . Left: noisy image,  $I$ . Middle: results of denoising  $I$  directly,  $I_{den} = VTV(I)$ , PSNR=27.00. Right: moving frame result (11.6),  $I_{denMF}$  with  $J^i(I)_{den} = VTV(J^i(I))$ , PSNR=27.56. Row 2:  $NLM$  results for image 15 [2] with  $\sigma = 20$ . Left:  $I$ . Middle:  $I_{den} = NLM(I)$ , PSNR=29.29. Right:  $I_{denMF}$  with  $J^i(I)_{den} = NLM(J^i(I))$ , PSNR=29.70. Row 3:  $BM3D$  result for image 24 [2] with  $\sigma = 20$ . Left:  $I$ . Middle:  $I_{den} = BM3D(I)$ , PSNR=31.26. Right:  $I_{denMF}$  with  $J^i(I)_{den} = BM3D(J^i(I))$ , PSNR=31.40. The PSNR is computed across the entire image, but the images included here are zoomed-in to better visualize details. Full resolution results can be found in [16]

- The Noise Clinic of Lebrun et al. [25] adapts the nonlocal Bayes approach [26] (that assumes AWG noise) to signal-, scale-, and frequency-dependent noise, which requires an estimate of the covariance matrix of the noise; the authors state that inaccuracies in the estimate of this covariance matrix can introduce artifacts.
- Seybold et al. [44] show how the performance of denoising methods decays drastically when using realistic noise models instead of AWG noise.
- Plötz and Roth [40] compared the BM3D denoising method to several state-of-the-art algorithms for AWG noise images. They concluded that when applied to real noise (for both RAW images and camera outputs), BM3D outperforms (visually and in terms of PSNR) several methods that under the AWG assumption were supposed to perform better.

In this section, we will stress the importance of having a proper noise model for the images one intends to denoise, extending the work presented in [17]. First, we introduce an image database consisting of clean and noisy photo pairs, in formats corresponding to the sensor image (12-bit RAW) and the nonlinear and

uncompressed camera output (8-bit PNG). Next, we use the database to show how both the performance and the ranking of denoising methods are considerably different in the realistic noise scenario as compared with the AWG noise case. Finally, we show how a simple local denoising method, applied to the RAW input, can outperform nonlocal methods applied to the camera output. The local method can also perform as well as low-complexity versions of the nonlocal algorithms applied to the RAW image, but with a much lower computational cost, suggesting the possibility for local denoising to replace more elaborate denoising methods for in-camera implementations.

### 11.3.1 *An Image Database with Clean and Noisy Versions of Photos in RAW and Nonlinear Camera Output Versions*

In order to evaluate and compare denoising methods, we created a test image set containing clean and noisy image pairs with both RAW and nonlinear output versions of each image. Using a Nikon D3100 camera, optimizing exposure, and setting ISO to 100 so as to minimize noise, we capture the twenty “clean” reference RAW images shown in Fig. 11.4. We add noise to each image and then both the clean original and the noisy version are passed through the basic image processing pipeline of a digital camera producing the nonlinear camera outputs. In this way, we have a noisy picture and the corresponding clean reference, which allows us to evaluate denoising results using objective image quality metrics like PSNR.

Recently, Plötz and Roth [40] also proposed a database of real noise photographs and their corresponding ground truth. Each pair in their database has a reference image and a ground truth which are generated from a series of images of the same scene, with different ISO values and exposure times. The reference is chosen to be the photograph taken with low ISO that shows almost no noise, and the ground truth

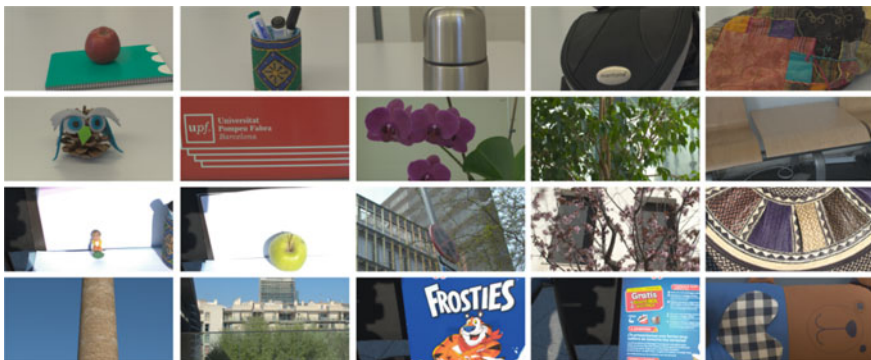


Fig. 11.4 Our image test set

is generated by a post-processing step that corrects for differences in illumination and minor displacements (of objects in the scene or camera shake) between several exposures. Our database has a different approach than the one proposed in [40], in that it includes an estimated RAW noise model so that new noisy images can be generated, with arbitrary noise levels. It also allows the user to introduce new clean images from which additional clean/noisy pairs (in both RAW and nonlinear output versions) can be added to the database. This database is publicly available [1] so that researchers can develop and test denoising algorithms for real-world scenarios.

The creation of our database involves the following components:

- *Simulating the camera processing pipeline.* This pipeline is necessary for processing the image from its RAW format to the camera output (as camera makers do not make public the specific steps each model performs).
- *Creating a signal-dependent noise model estimated at the RAW level.*
- *Validating the results.* A noisy RAW image should have the same appearance as a clean RAW image to which noise has been added according to our estimated model.

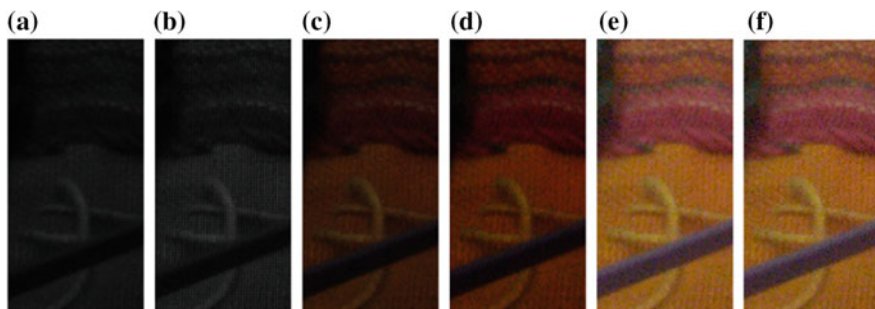
We detail each of these stages in the following subsections.

### 11.3.1.1 Simulate the Camera Processing Pipeline

The following is a concise enumeration of the basic steps of the image processing pipeline, common to digital cameras.

1. **Capture.** A photo is saved in the RAW format as a 12-bit depth image, obtaining the CFA (color filter array) RAW data with a Bayern mosaic pattern. An image example is illustrated in Fig. 11.5a.
2. **White balance.** This process guarantees that the image has no color cast. For neutral colors to keep the correct appearance, a scaling of all intensity values from the RAW file is performed. Figure 11.5b depicts the white balanced image example.
3. **Demosaicking.** The camera sensors produce an image in which for each pixel we only get one of the image channel intensity values (either red or green or blue); demosaicking is an interpolation process that estimates the other two missing values, as exemplified in Fig. 11.5c. For our image processing pipeline, we chose the local demosaicking algorithm proposed by Malvar et al. [34], which is based on bilinear interpolation and further refined by using the correlation among the RGB channels, with Laplacian cross-channel corrections.
4. **Color correction.** This process makes the conversion from the camera color space to sRGB (standard RGB) color space, as illustrated in Fig. 11.5d.
5. **Gamma correction.** In this step, the (normalized) image values are raised to the standard power of  $1/2.2$ . An image example is shown in Fig. 11.5e. This step assures an optimized encoding that models the nonlinear human perception of luminance: more sensitive to details in darker areas.





**Fig. 11.5** Image example to illustrate the camera processing pipeline. From left to right: RAW original image (a), result after applying white balance (b), demosaicking (c), color correction (d), gamma correction (e), and quantizing (f)

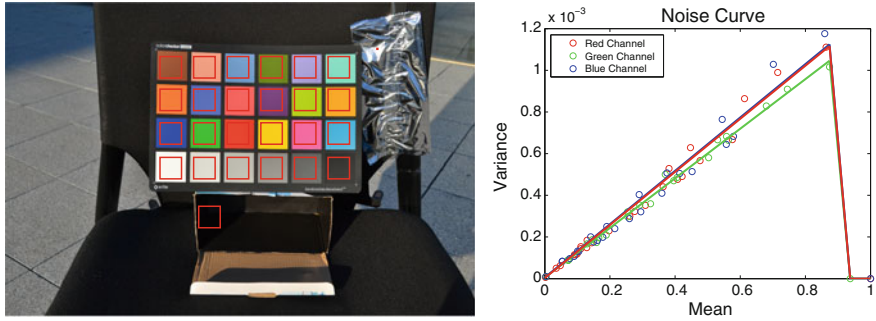
- Quantization.** This final step (for our purposes) of the pipeline quantizes the image from 12-bit depth to 8-bit depth, outputting an RGB image ready for display, as in Fig. 11.5f.

### 11.3.1.2 The Noise Model

For constructing a realistic signal-dependent noise model, estimated on the RAW image, we follow the line of experiments of [19, 44, 46]. We analyze a RAW Colorchecker photograph by segmenting and extracting all of its 24 homogeneous color patches, and computing the noise variance in each color square and for each RAW color channel. Tests on several RAW images of the Colorchecker, taken with different camera settings, allow us to conclude, as in [44] and [46], that the variance as a function of the mean can be fitted by an increasing linear function, indicating that the noise is signal-dependent.

Our noise estimation setup, exemplified by the left side of Fig. 11.6, also includes two objects associated to two extreme cases that the Colorchecker does not take into account: a cardboard box with the interior in shadow and painted with black matte paint, and an aluminum foil that receives direct light and creates specular highlights. The noise variance is then computed from crops from the black box as well as the area with specular highlights. All of the patches from which we compute the variance are marked in red in the left image in Fig. 11.6. Finally, we estimate a noise model from the 26 mean and variance pairs. An example of the channel-wise variance plot as a function of mean pixel value is shown in right side of Fig. 11.6.

To add noise to a clean RAW image, we add to each pixel in the clean RAW image white Gaussian noise with the local variance given by the variance plot value corresponding to the pixel's intensity. Afterward, we apply the rest of the camera pipeline: white balance, demosaicking, color correction, gamma correction and quantization to 8-bit depth.



**Fig. 11.6** The Colorchecker setup: an image of our noise estimation setup captured with ISO 800, with marked regions used for estimating the noise model (left). Plot (example for one set of fixed camera parameters) of variance as function of the mean, for a RAW image scaled between 0 and 1 (right). Values extracted from a RAW Colorchecker image taken with ISO 3200. Dots show the real values obtained for each color square and each channel, while the continuous lines show the fitted linear functions

### 11.3.1.3 Validation of the Noise Model

The ISO speed (or ISO sensitivity) estimates the camera sensitivity to light: the higher the value, the higher the sensitivity. The camera transforms the light captured by the sensors into an electrical signal, and increasing the ISO means amplifying the electrical signal before the signal conversion from analog to digital. For example, when increasing the ISO value from 100 to 200, the original electrical signal is doubled. Amplifying the electrical signal better preserves image details. However, this comes with the cost of amplified noise: the higher the ISO speed, the higher the noise level. This justifies our choice of noise levels: we associate one to each possible ISO value. Table 11.4 shows the average standard deviation computed over our test set on the output images for the ISO levels given by our camera. Notice that the highest noise level associated to ISO 3200 produces a relatively small standard deviation.

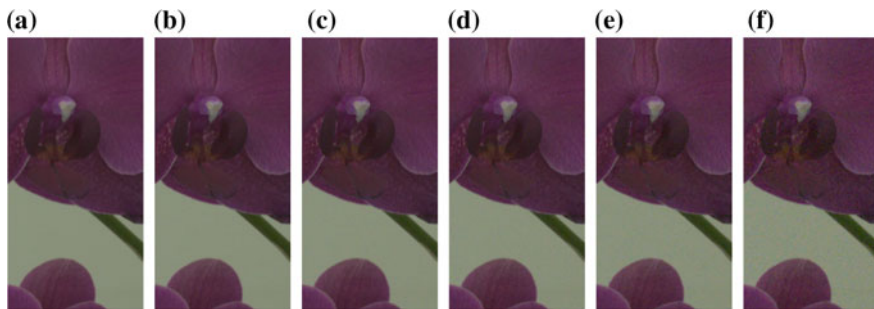
Figure 11.7 illustrates a crop from an image example from our database, for which the original image and noisy images were created with the realistic noise model in Sect. 11.3.1.2 and then followed the camera processing pipeline in Sect. 11.3.1.1.

Figure 11.8 contains a validation of our noise model, comparing several image examples taken with ISO 3200, which produces the highest noise level.

**Table 11.4** Average noise levels given by different ISO noise curves on our test set

ISO sensitivity	100	400	800	1600	3200
$\sigma$	2.57	3.4	4.24	5.82	8.39





**Fig. 11.7** Crops from a noisy image example from our test set. From left to right: **a** original image, **b–f** synthesized noisy images obtained with the noise curve associated to: **b** ISO 100 ( $\sigma = 2.42$ ), **c** ISO 400 ( $\sigma = 3.17$ ), **d** ISO 800 ( $\sigma = 3.95$ ), **e** ISO 1600 ( $\sigma = 5.43$ ), and **f** ISO 3200 ( $\sigma = 7.98$ )

### 11.3.2 Ranking Denoising Algorithms: AWG Noise Versus Realistic Noise Model

In this section, we present an experiment that shows how essential the noise model is for evaluating denoising methods. For this, we compare three denoising methods applied to camera output images created with two different noise models. We use the patch-based NLM and BM3D denoising methods, implemented with their publicly available IPOL code [12] and [27], and the local VTV-based denoising method.

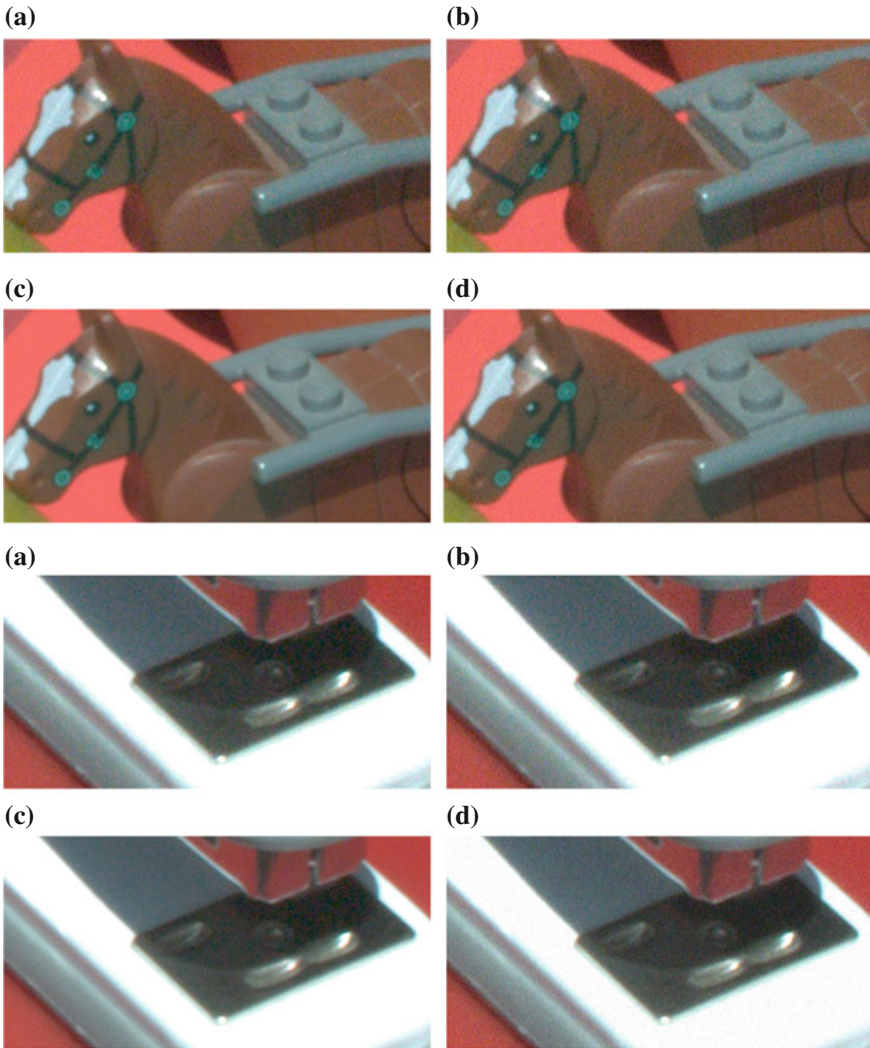
This latter method, proposed by Blomgren and Chan [9], is a vectorial extension of the channel-wise TV-based denoising and consists of replacing the gradient operator acting on each channel by the Jacobian operator acting on the whole image:

$$I_{t+dt} = I_t - dt \mathcal{J}^* \left( \frac{\mathcal{J}(I)}{\sqrt{\|\mathcal{J}(I)\|^2 + \varepsilon}} \right), I_{t=0} = I_0, \quad (11.15)$$

where  $\mathcal{J}$  and  $\mathcal{J}^*$  are, respectively, the Jacobian operator and its adjoint, and  $\varepsilon$  is a small positive constant used to avoid division by 0. We stop the iterative procedure after a fixed number of iterations.

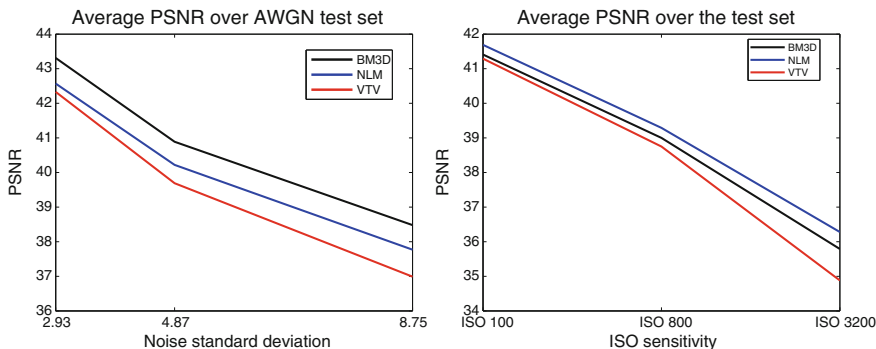
The parameters of these algorithms are the standard deviation of the noise, in the case of NLM and BM3D, and the number of iterations for the VTV-based denoising. We experimentally optimize these parameters for each image and noise level of the database, choosing the ones that maximize the PSNR values of the denoised results. We compute the denoising results under the following two noise models:

1. Starting with a clean RAW image  $I_{cleanRAW}$ , add Gaussian noise, with variance given by the associated noise variance plot as detailed in Sect. 11.3.1.2, to obtain a noisy image  $I_{noisyRAW}$ . For  $I_{cleanRAW}$  and  $I_{noisyRAW}$ , apply white balance, demosaicking, color correction, gamma correction, and quantize to 8 bit to obtain the camera outputs  $I_{clean}$  and  $I_{noisy}$ . Apply NLM, BM3D and VTV-based denoising on  $I_{noisy}$  to obtain the denoised images  $I_{NLM}$ ,  $I_{BM3D}$  and  $I_{VTV}$ .



**Fig. 11.8** Comparing a real noise photograph (a) and synthesized noisy images obtained by adding: **b** Gaussian noise with variance given by our realistic noise model to the RAW image, **c** Gaussian noise of constant variance to the RAW image, and **d** Gaussian noise of constant variance to the camera output

2. Starting with the clean camera output image  $I_{clean}$ , add white Gaussian noise (AWG), with fixed variance described in the next paragraph, to obtain a noisy image  $I_{awgnoisy}$ . Apply NLM, BM3D and VTV-based denoising methods on  $I_{awgnoisy}$  to obtain the denoised images  $I_{awgNLM}$ ,  $I_{awgBM3D}$ , and  $I_{awgVTV}$ .



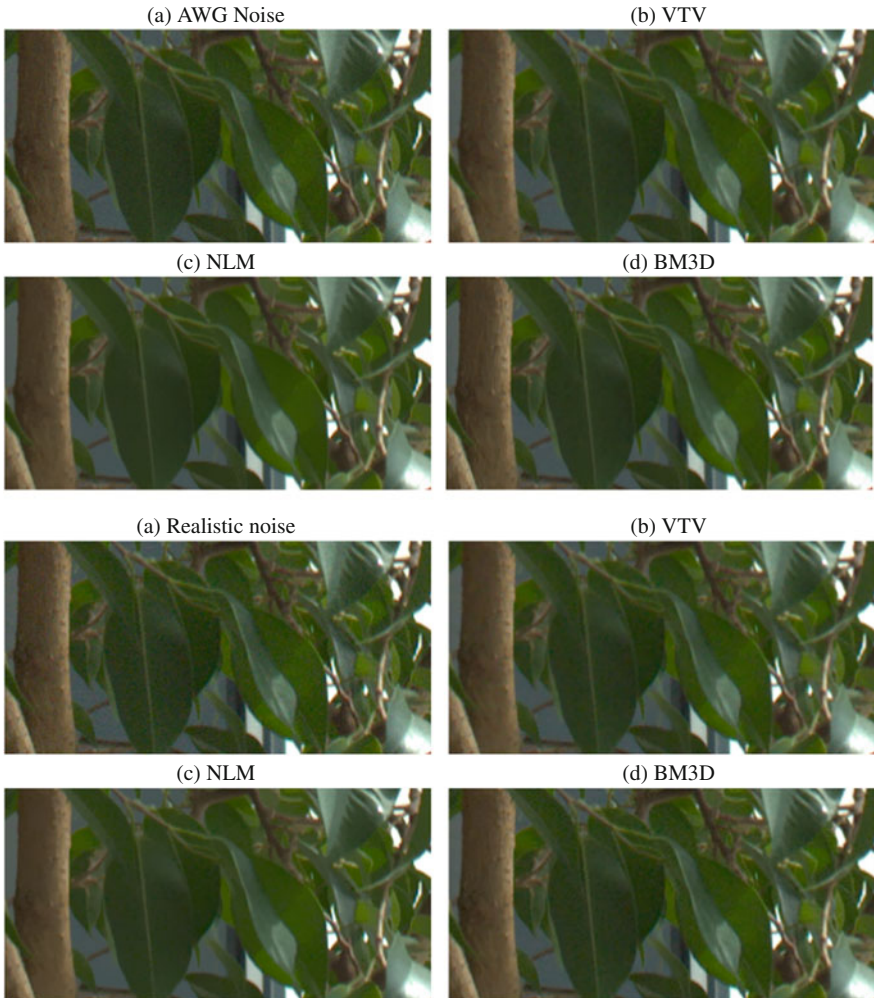
**Fig. 11.9** Comparison of BM3D, NLM, and VTV applied to the camera output, under two noise models. Average PSNR value plots of denoising applied to noisy images created with additive white Gaussian noise (left) and our realistic noise model (right)

The reference clean image  $I_{clean}$  serves as a ground truth for both experiments. We tested the denoising methods on our test set, and the PSNR results are shown in Fig. 11.9. The left shows a plot of PSNR as a function of the average noise standard deviation computed in the 8-bit depth noisy images with AWG noise over the database. The right shows PSNR as a function of ISO sensitivity for images degraded using our realistic noise model. Comparable noise levels are used for both experiments, as seen from the average standard deviation values shown in Table 11.4.

Notice how the ranking of the denoising methods is different with realistic noise than with AWG noise. This justifies the use of a realistic noise model for image denoising. There is also a large drop in the PSNR value for each denoising method from denoising the AWG noise images to realistic noise images, consistent with what was reported by Seybold et al. [44]. The local denoising method applied to the camera output gives worse results in terms of PSNR than the nonlocal patch-based methods, for both noise models. Figure 11.10 illustrates this behavior. Denoising an AWG noise image with BM3D gives an excellent output, while for the same original image but with realistic noise, BM3D produces blocking artifacts on the leaf in the shadow.

### 11.3.3 Comparing Local Denoising on DRAW Versus Nonlocal Denoising on Camera Output

In this section, we illustrate the power of processing the RAW data as opposed to the camera output by showing how a local denoising method, applied to the demosaicked RAW (DRAW) image, can outperform nonlocal methods applied to the camera output.



**Fig. 11.10** Comparison of VTV, NLM, and BM3D denoising methods under AWG on camera output and a realistic noise model. Rows 1–2. **a** Crop from AWG noise image “image9” with  $\sigma = 4.96$ . **b** VTV, PSNR=37.66. **c** NLM, PSNR=38.37. **d** BM3D, PSNR=38.92. Rows 3–4. **a** Crop from realistic noise image “image9” with  $\sigma = 5.67$  and ISO 800. **b** VTV, PSNR=35.39. **c** NLM, PSNR=35.89. **d** BM3D, PSNR=35.72

### 11.3.3.1 Adapting a TV-based Denoising Method to the Signal-Dependent Noise Model

TV-based denoising methods were proposed in the context of images corrupted by signal-independent AWG noise. However, our database images suggest realistic signal-dependent noise is a more accurate model.

Two reasonable approaches are discussed in [33] for removing signal-dependent noise. The first approach is to *adapt an existing denoising method* to treat specific noise model properties. For example, Luisier et al. propose a methodology to adapt transform-domain thresholding algorithms for the mixed Poisson–Gaussian noise model [31]. The second approach is to *create a variance stabilizing transformation (VST) for the particular noise model*. Applying the VST to an image removes signal-dependency and the noise variance becomes constant over the entire image. Then one can use a denoising algorithm created to eliminate Gaussian noise with constant variance. After denoising, one needs to apply the inverse VST. The advantage of the second technique is that denoising images corrupted by AWG noise is an extremely popular topic that has produced many algorithms over the last decades.

Donoho [15] was the first to propose applying the Anscombe transform [4] as a VST. As described above, a denoising algorithm created to eliminate AWG noise with constant variance can then be applied, followed by the inverse VST. Mäkitalo and Foi [32] also used the Anscombe transform to remove the signal-dependency, but emphasized the importance of applying a suitable inverse. Following this approach, we apply the Anscombe transform  $f_{\text{Anscombe}}$  to the demosaicked noisy RAW image  $I_{\text{noisyDRAW}}$ :

$$f_{\text{Anscombe}}(I_{\text{noisyDRAW}}) = 2\sqrt{I_{\text{noisyDRAW}} + \frac{3}{8}} \quad (11.16)$$

and denoise the image  $f_{\text{Anscombe}}(I_{\text{noisyDRAW}})$  with VTV, instead of denoising  $I_{\text{noisyDRAW}}$ ; this intermediate result is denoted  $D$ . Then, we apply the closed-form approximation of the exact unbiased inverse Anscombe transform proposed by Mäkitalo and Foi [32] to  $D$ :

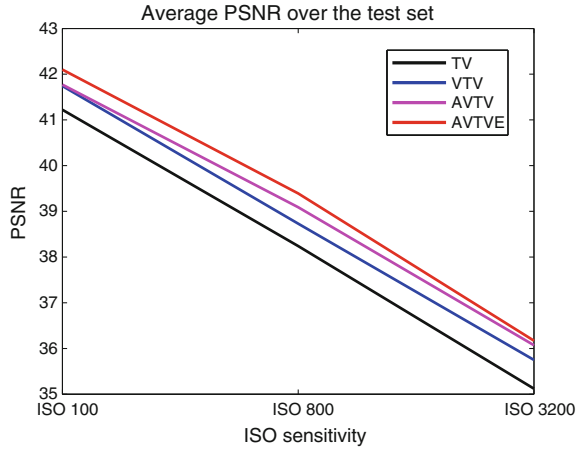
$$\widetilde{f_{\text{Anscombe}}^{-1}}(D) = \frac{1}{4}D^2 + \frac{1}{4}\sqrt{\frac{3}{2}}D^{-1} - \frac{11}{8}D^{-2} + \frac{5}{8}\sqrt{\frac{3}{2}}D^{-3} - \frac{1}{8}.$$

We denote the resulting image by  $I_{\text{denDRAW}}$  and refer to this method combining VTV with the Anscombe transform as AVTV. Figure 11.11 illustrates an improvement in the average PSNR value of image results obtained by applying the Anscombe transform before denoising with the VTV-based procedure described by (11.15).

### 11.3.3.2 Refine the Denoising Output by Recovering Lost Details

Even the best denoising algorithms can benefit from the so-called “boosting” techniques [42]. One boosting mechanism involves adding content from the residual (difference between the noisy and denoised image) back to the denoised image. This is justified by the fact that denoising is an imperfect process that eliminates not only noise but small details as well. But while signal leftovers can be retained in the residual, the opposite is also true: noise is retained in the denoised image. An alternate

**Fig. 11.11** Evolution of our TV-based local denoising experiments, under the proposed realistic noise model: average PSNR values computed over our image test set



boosting mechanism aims to eliminate the noise retained in the denoised image, but this produces oversmoothed images as a result.

The biggest challenge for any denoising method, especially for a local one, is to make the distinction between noise and details. As a boosting technique, we propose adding back selected useful information to the denoised image, determined by

$$I_{AVTVE} = (1 - a)I_{noisyDRAW} + aI_{denDRAW}, \tag{11.17}$$

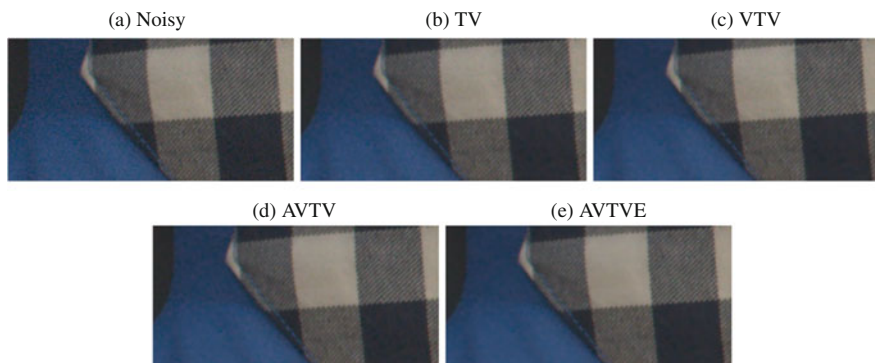
where the weight function  $a$  should ideally include information related to the local image content. The essential part here is the criteria for differentiating between noise and details we want to recover from the residual.

The function  $a$  we consider is an indicator of the local information in the luminance channel of the denoised image. In smooth areas we want to keep the pixel intensity values of the denoised image intact, so the value of  $a$  should be large; on the other hand, along image contours we want to partially recover some of the details of the original noisy image, so the value of  $a$  should be small (positive and close to 0). Therefore, we propose estimating  $a$  using a local edge indicator, like the Charbonnier diffusivity function [13]

$$f_{Charbonnier} = \frac{1}{\sqrt{1 + \frac{\|\nabla L(I_{denDRAW})\|^2}{\lambda}}}, \tag{11.18}$$

where  $L$  denotes the luminance component, and  $\lambda > 0$  is a contrast parameter related to edge localization. The image  $I_{denDRAW}$  is obtained by finding the number of iterations in the iterative scheme introduced in (11.15) that maximizes the PSNR index computed after color correction, gamma correction, and the quantization step. We give a higher weight to the denoised image than to the noisy one, by choosing:





**Fig. 11.12** Comparison of our local TV-based denoising methods applied to the demosaicked RAW, under the proposed realistic noise model. **a** Crop from noisy image “image20” with  $\sigma = 4.84$  and ISO 800. **b** TV, PSNR=37.14. **c** VTV, PSNR=37.38. **d** AVTV, PSNR=37.45. **e** AVTVE, PSNR=37.73

$$a = \frac{1 + f_{\text{Charbonnier}}}{2}. \quad (11.19)$$

Experiments show that both the step of applying the Anscombe transform before denoising and its inverse after, and the step of refinement described by (11.17), bring an improvement (both in terms of PSNR and visually) compared to only denoising with the iterative scheme introduced in (11.15), as seen in Figs. 11.11 and 11.12. These experiments also illustrate that the vectorial VTV-based denoising approach improves the channel-wise TV-based denoising strategy that was used in [17].

### 11.3.3.3 The Comparison

The experiment in this section is intended to mimic a realistic scenario. Nonlocal patch-based denoising methods are too complex to be implemented in-camera without essential simplifications. Therefore, we compare our local denoising approach AVTVE applied to the demosaicked RAW noisy image, with two nonlocal patch-based methods (NLM and BM3D) applied at the end of the noisy image processing chain, following these steps:

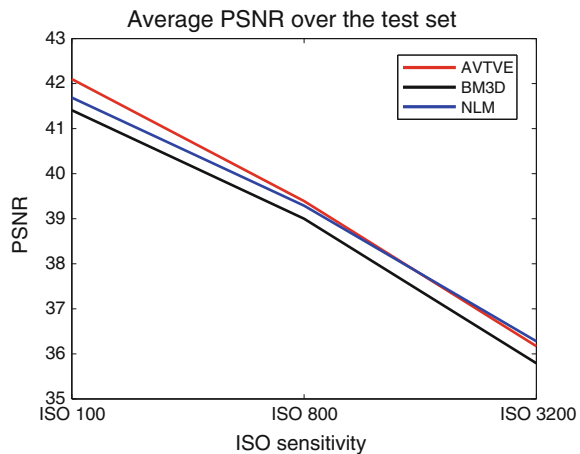
1. We take a clean RAW image  $I_{\text{cleanRAW}}$  and add Gaussian noise, with variance given by the associated noise variance plot as detailed in Sect. 11.3.1.2, to obtain a noisy image  $I_{\text{noisyRAW}}$ .
2. For  $I_{\text{cleanRAW}}$  and  $I_{\text{noisyRAW}}$ , apply white balance, demosaicking, color correction, gamma correction, and quantize to 8 bit to obtain the camera outputs  $I_{\text{clean}}$  and  $I_{\text{noisy}}$ .

3. Apply nonlocal patch-based denoising methods (NLM and BM3D) to  $I_{noisy}$  to obtain the denoised image ( $I_{NLM}$  and  $I_{BM3D}$ ), optimizing the denoising parameters so as to maximize the PSNR. The reference clean image  $I_{clean}$  serves as a ground truth.
4. Apply white balance and demosaicking to  $I_{noisyRAW}$  to obtain  $I_{noisyDRAW}$ . Then denoise with our local method AVTVE, followed by color correction, gamma correction, and quantization to 8 bit, to output our denoised image  $I_{AVTVE}$ . The denoising parameters, described in the following, are optimized so as to maximize the PSNR of  $I_{AVTVE}$ .
5. Evaluate the images  $I_{BM3D}$ ,  $I_{NLM}$ , and  $I_{AVTVE}$ , visually and with respect to PSNR.

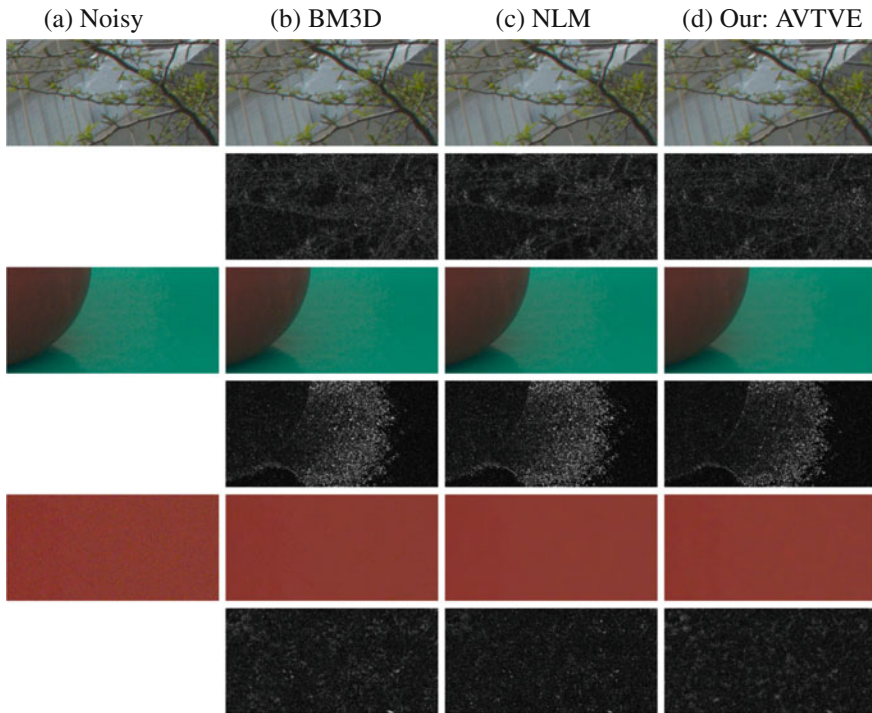
The plot of Fig. 11.13 shows the average PSNR values over our proposed image dataset, for each noise level given by the considered ISO sensitivity and each denoising strategy aforementioned. We can see that our denoising method AVTVE produces better results in terms of PSNR than the BM3D and NLM denoising methods, for almost all ISO levels. However, for the highest noise level associated to ISO 3200, NLM is the best in terms of PSNR, while our method is second.

A visual comparison is illustrated in Fig. 11.14, where the images are denoised with the optimal parameters described above. For a better comparison, the difference between the clean and denoised images for each method is included in Rows 2, 4, and 6. Ideally, the difference image should be completely black, as the difference is given by lost details or artifacts introduced by the denoising method. The close-up images in the first two rows demonstrate that for small noise levels, the denoising methods BM3D, NLM and AVTVE produce comparable results. All three algorithms preserve the apple in the example in the third and fourth rows, although the AVTVE result has a cleaner appearance while the NLM and BM3D denoised images exhibit small blocking artifacts. An image with the highest noise level was investigated in the bottom two rows, where the BM3D output reveals strong blocking artifacts in the homogeneous area, while the AVTVE and NLM results have a cleaner appearance.

**Fig. 11.13** Comparison between the local denoising method AVTVE applied to the demosaicked RAW to the NLM and BM3D denoising algorithms applied to the camera output, under the proposed realistic noise model. Average PSNR values computed over our image test set







**Fig. 11.14** Comparison of the local denoising method AVTVE applied to the demosaicked RAW to the BM3D and NLM denoising methods applied to the camera output, under the proposed realistic noise model. Row 1: crop from noisy image “image13” with  $\sigma = 4.11$  and ISO 800, BM3D result with PSNR=36.79, NLM result with PSNR=36.97, AVTVE result with PSNR=37.11. Row 2: difference images for crops of Row 1, scaled for visualization with the scaling factor 7. Row 3: crop from noisy image “image1” with  $\sigma = 4.55$  and ISO 100, BM3D result with PSNR=36.35, NLM result with PSNR=35.94, AVTVE result with PSNR=37.95. Row 4: difference images for crops of Row 3, scaled for visualization with the scaling factor 7. Row 5: crop from noisy image “image7” with  $\sigma = 9.14$  and ISO 3200, BM3D result with PSNR=38.93, NLM result with PSNR=40.11, AVTVE result with PSNR=39.70. Row 6: difference images for crops of Row 5, scaled for visualization with the scaling factor 7

Table 11.5 shows the average running time for the AVTVE, NLM, and BM3D methods for a  $1000 \times 2000$  color image from our test set, on a i7-4770 CPU with 3.4GHz and 8 cores. At a fraction of the running time of NLM and BM3D, the AVTVE method, although not optimized for speed, produces results that are comparable or better both visually and in terms of PSNR.

**Table 11.5** Average running time (s) for one  $1000 \times 2000$  color test image of *AVTVE*, *BM3D*, and *NLM*, for different noise levels given by different ISO settings

ISO sensitivity	100	400	800	1600	3200
AVTVE	1.5	2.0	2.5	3.5	5.0
BM3D	31	31	31	31	31
NLM	19	19	19	19	19

#### 11.3.3.4 Compare Local with Low-Complexity Nonlocal Denoising Applied at the Same Stage of the Image Processing Chain

As nonlocal methods have a high complexity and cannot be implemented in-camera unless some simplifications are done, we consider a reduced-complexity version of the BM3D method, obtained by tuning several parameters such that the method reaches the lowest complexity while producing a reliable image output. We fix the patch size to  $8 \times 8$  and search for similar patches in a small window of size  $10 \times 10$ . As we did for the local method, we optimize the default parameter for each image and noise level of the test set, choosing the one that maximizes the PSNR value of the denoised result.

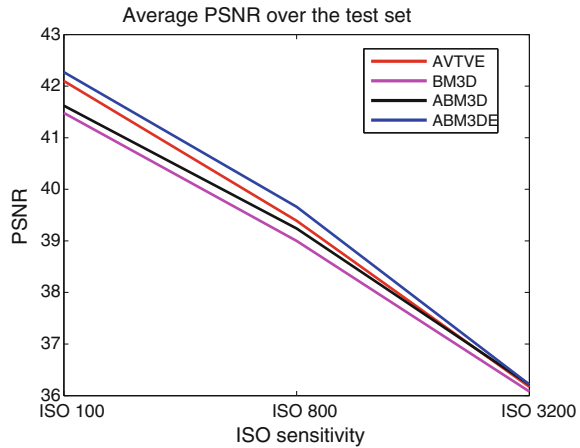
As the BM3D method is designed to treat Gaussian noise, we apply the Anscombe transform introduced in (11.16) before denoising and its inverse after, like we did for our denoising method. We denote this approach by ABM3D. We also consider the refinement step described by (11.17), producing a result denoted ABM3DE. Both the steps of applying the Anscombe transform and the refinement step produce image results that have a higher PSNR value compared to the BM3D output, as seen in the plot of Fig. 11.15. Table 11.6 shows the average running time for the AVTVE, ABM3D and ABM3DE methods for one  $1000 \times 2000$  color image from our test set, on the same machine. For computing the running time of the BM3D algorithm, we use the fast C++ implementation available online [27], while we point out again that our current implementation of the AVTVE algorithm is not optimized for speed.

The plot in Fig. 11.15 shows the average PSNR value over our proposed image dataset, for each noise level given by the considered ISO sensitivity and each denoising strategy aforementioned. For almost all noise levels, our method produces images that are better than BM3D and ABM3D in terms of PSNR. For a higher running time, ABM3DE gives the best PSNR value. Notice that for the highest noise level, all methods produce denoised images with a very similar PSNR value.

### 11.3.4 Research Avenue to Explore

It is clear that many powerful algorithms exist for removing AWG noise from images, and they can be used also to handle RAW pictures after they have been processed with a VST like the Anscombe transform. However, denoising is still a challenging

**Fig. 11.15** Comparison between the local denoising method AVTVE and the low-complexity BM3D, ABM3D, and ABM3DE applied on the demosaicked RAW, under the proposed realistic noise model. Average PSNR values computed over our image test set



**Table 11.6** Average running time (s) for one  $1000 \times 2000$  color test image of AVTVE, ABM3D, and ABM3DE, for different noise levels given by different ISO settings

ISO sensitivity	100	400	800	1600	3200
AVTVE	1.5	2.0	2.5	3.5	5.0
ABM3D	5.2	5.2	5.2	5.2	5.2
ABM3DE	10.4	10.4	10.4	10.4	10.4

task for regular camera output images, where the noise model is extremely complex. There is a need to develop a noise model for JPEG images from which a VST for JPEG noise can be derived, allowing then regular denoising methods that assume AWG noise to be applied to the JPEG case as well. An alternative would be to develop a noise model for JPEG images whose parameters can be estimated from the image itself, and develop new denoising methods adapted to this model.

### 11.4 Optimize Denoising Methods According to Perceived Quality of Results

In film photography, noise is called “film grain” as it is due to the presence of minuscule grains of silver. When subtle, people actually prefer its presence [3] as it improves image appearance. This is a phenomenon due to visual perception: a small amount of noise makes the image look sharper and appear to have higher resolution.

However, too much noise, or noise that is not uniform but highly localized, will make the image unpleasant. This is the case of digital image noise, which is not uniform but image-dependent, appearing more pronounced in dark areas and shadows.

The noise level introduced by common consumer cameras is surprisingly small compared to the level of AWG noise added to clean images in academic works

in order to create synthetic noisy images, as is traditionally used in testing image denoising algorithms. This fact can be concluded from Table 11.4 that includes the average standard deviation computed over realistic noise images, for the ISO levels given by our camera, as in our experiments described in Sect. 11.3.1.2. It can be seen that even the highest noise level corresponding to ISO 3200 gives a small standard deviation, on average. This fact is confirmed in [40], where the authors claim that using noise standard deviations of at least  $\sigma = 10$  for synthetic AWG noisy images is “mostly a historical artefact”.

Photographers often add a small amount of noise to studio images taken at low ISO, due to the fact that a photo that is “too perfect” can be perceived as fake [3]. To emulate grain noise, several layers of Gaussian noise with different variance are added. Photographers always add the noise after the sharpening step and the added noise is achromatic. While the luminance noise in a digital sensor has some sort of similarity to grain, the chroma noise (given by variations in colors) is not something we like to see in a photograph.

In [23] researchers worked with a professional photographer to learn specific ways in which a digital image can be aesthetically improved by adding noise: masking actual noise and banding artifacts in the original, improving the appearance of blown highlights, increasing the perceived resolution. The photographer introduced several different noise layers for each image example resulting in a higher noise level in midtones and a lower one in shadows and highlights, with almost no noise toward 0 and 255. Regarding the noise distribution, while for midtones a Gaussian distribution is a good choice, for highlights the histograms are skewed and the best candidate is the chi-squared distribution characterized by an asymmetrical shape.

Johnson and Fairchild [21] examined some of the ingredients that influence image sharpness perception. The highest image score was achieved by the highest resolution image to which noise with  $\sigma = 10$  was added, and processed with both contrast enhancement and increased sharpness. Regarding the noise factor, the conclusion was that additive uniform noise applied independently in each color channel increases the perceived sharpness only up to a point, after which it decreases. Interestingly, adding noise can also mask a reduction in image resolution; images with 300 ppi and 150 ppi were evaluated as having similar perceived sharpness when the lower resolution photo had added noise and increased contrast. Kurihara et al. [24] concluded that when noise is added to edges, sharpness decreases, while when added to texture, it increases up to a point, decreasing afterward. Kayargadde and Martens [22] investigated the connection between the noise level and blur, finding that adding noise to a sharp image causes it to be perceived as more blurred, while adding noise to a blurred one makes it appear a little less blurred.

None of these aspects of perceived image quality are considered when evaluating denoising results using existing image quality metrics. The PSNR measure is known to have problems in indicating the perceived image quality, and although SSIM [47] is designed to take into account perceived errors, it is not well correlated to human preference [39]. Even though their limitations are known, PSNR and SSIM are still the most popular measures for image denoising. There are many more image quality metrics, as, for example, the visual information fidelity (VIF) measure [45],

the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [35], or the naturalness image quality evaluator (NIQE) [36]. However, the vast majority of metrics are based on measuring differences between the denoised result and the clean ground truth, which does not necessarily correlate with the perceived image quality of the denoised result, as we will demonstrate in the next subsections. This section is extending the work described in [18].

### 11.4.1 Experiments

We compare three denoising methods: we introduce a local curvature smoothing (CS) algorithm and compare it with the nonlocal patch-based methods NLM [11] and BM3D [14].

For the CS algorithm, we start with the original noisy image  $I^0 = I_0$ , compute channel-wise its regularized level line curvature  $\kappa_{\varepsilon_2}(I_0)$  (which is just the usual level line curvature computed with an added value  $\varepsilon_2$  in the denominator: as this value increases, the curvature becomes smoother), and iterate the following equation  $N$  times:

$$I^{n+1} = I^n + \Delta t \left[ \nabla^- \cdot \left( \frac{\nabla^+ I^n}{\sqrt{\|\nabla^+ I^n\|^2 + \varepsilon_1}} \right) - \kappa_{\varepsilon_2}(I_0) \right], \quad (11.20)$$

where  $\nabla^+$  and  $\nabla^-$  are the forward and backward spatial difference operator. Due to the fact that the curvature can be estimated for each pixel with a  $3 \times 3$  stencil around it, the proposed method is local. We fix the parameters:  $\varepsilon_1 = 10^{-6}$  (very small for a good approximation of  $\kappa(I)$ ),  $\Delta t = 0.002$  and  $N = 30$ . The CS method has only one parameter, the regularizing value  $\varepsilon_2$ , and how it is chosen will be described later.

We perform this comparison on two image databases: images from the Kodak database [2] with AWG noise and photographs taken by us with the noise model proposed in Sect. 11.3.1.2. The evaluation is done using subjective testing as well as the objective PSNR and SSIM metrics.

#### 11.4.1.1 AWG Noise Case

The subjective evaluation involved 17 participants (all with normal or corrected to normal vision). Subjects sat in a well-lit office environment at approximately 64 cm from the display and were presented with four versions of an image: the original at the top, and the three denoising results (CS, NLM, and BM3D) in some random order at the bottom. The observers were asked to look at the original image and then indicate which of the three provided denoised images they preferred (Figs. 11.16 and 11.17).

We picked randomly three images from the Kodak database (“kodim1”, “kodim3”, and “kodim13”) and created three noise levels by adding Gaussian noise with



Fig. 11.16 Test images: crops from Kodak images

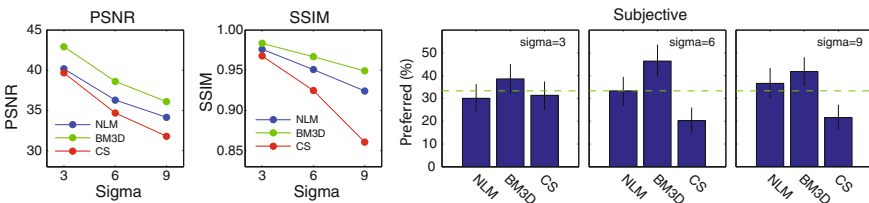


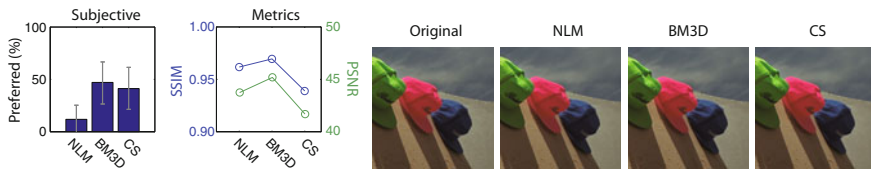
Fig. 11.17 From left to right: average PSNR and SSIM computed for three images from the Kodak database, and results of psychophysical experiment for comparing our proposed local denoising method to BM3D and NLM

Table 11.7 Optimized parameter value  $\epsilon_2$  as a function of  $\sigma$

$\sigma$	3	6	9
$\epsilon_2$	0.00032	0.003	0.00608

$\sigma = 3, 6, 9$ . As commented above, although these  $\sigma$  values seem low, they are common noise levels in photography. The denoising algorithms NLM [12] and BM3D [27] take as input the value of  $\sigma$ . We find the value of  $\epsilon_2$  of our method with a subjective methodology; we ask participants to adjust  $\epsilon_2$  via key presses for finding their preferred image result. We average over subjects and images to get one value of  $\epsilon_2$ , as shown in Table 11.7, for each noise level.

We compute the values of PSNR and SSIM for each denoising method, as we have the clean ground truth. We also conduct a user preference test (using the procedure described above). We crop the images to be able to simultaneously show all of them at the native resolution and avoid resizing. The results in Fig. 11.17 indicate that the SSIM metric gives a reasonable approximation of the subjective scores, predicting that the differences between the algorithms are small for a low noise level and that for high noise levels, the local CS method gives a poor performance. However, both PSNR and SSIM are poor at predicting the preferred algorithm on an image-by-image basis, as Fig. 11.18 shows. To evaluate the metric performance we compute an upper



**Fig. 11.18** Visual comparison for one test crop from image “kodim3” and user preferences

bound by randomly dividing the subjective data into two subject groups (A and B). We then compute a percentage correct score, for each image. The score is 100% if the order is entirely correct, 33% for only getting the order of one correct or 0% for a complete failure. The result is that, on average, group A is able to predict the data from group B 64% of the time. However, both the SSIM and PSNR achieve a score of less than 46%, with a baseline score of 33%.

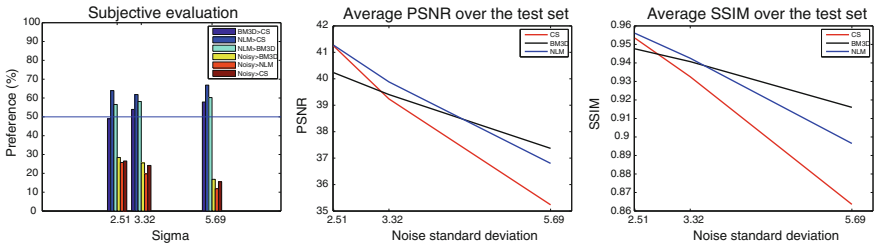
### 11.4.1.2 Realistic Noise Case

We conduct a user preference test on realistic noise removal, with results given by denoising with CS, NLM and BM3D, as well as no denoising at all. The subjective evaluation involved 19 participants (all with normal or corrected to normal vision) that sat in a well-lit office environment at approximately 64 cm from the display. We used the 20 images and three noise levels given by ISO 100, 400 and 1600 from our proposed test set following the realistic noise model proposed in Sect. 11.3.1.2, cropped to allow a simultaneous display of two images at their native resolution. At the first stage, we find the values for  $\sigma$  (the parameter for NLM and BM3D) and  $\varepsilon_2$  (parameter for CS) through user tests. Subjects are presented with a sequence of 51 versions of the same image denoised with different values of the pertinent parameter, from a minimum (no denoising) to a maximum (the image is fully denoised but blurry), and the observer chooses the one he/she prefers. At the second stage, subjects are asked to select their preferred image between two versions of it displayed simultaneously, which can be either the original noisy or the result of the preferred output of the denoising methods NLM, BM3D and CS obtained at the previous stage. The results of the psychophysical experiment are shown in Fig. 11.19.

On the left-hand side of the figure, we plot the user preference averaged across all images, for each noise level considered. People preferred NLM for all noise levels. They slightly preferred CS over BM3D, for a small noise level, while the results were comparable for a medium noise level. As the noise increased, for ISO 1600, people preferred NLM even more compared to the BM3D and CS algorithms.

Regarding the comparison between the original noisy and the denoised images, observers voted for applying a denoising method over not applying any method. However, the original noisy image was preferred surprisingly often (around 20%), especially when compared to the BM3D output. Averaging over all subjects, there is no image for which no denoising was preferred over all denoising methods. Therefore, choosing the noisy image penalized the denoised image, it was not a vote of





**Fig. 11.19** A two by two comparison between the three denoising algorithms and the original noisy image

appreciation for the noisy image. There is only one image (Image 18, cartoon-like) for which the noisy original did not get any vote from any observer, for all noise levels considered. This might indicate that we do not like noise in cartoon-like images.

At the first stage, observers could choose between the noisy original and 50 denoised images with increasing parameter values: only one observer chose the noisy image in several cases. At the second stage, of comparing two by two each of the denoising methods and no denoising, all subjects gave at least one vote to one original noisy image. Therefore, when the noisy stimulus was shown next to the denoised one, the preference for the noisy image increased.

For each denoising method, we computed the average of PSNR and SSIM for all images that people chose as their preferred one. We compared these objective metrics result to the subjective one, included in Fig. 11.19. For a small noise level, in the case of the BM3D denoising method, people preferred images denoised with parameters giving a surprisingly low PSNR value. The PSNR value also estimated a large difference in quality between the outputs of CS and BM3D, while people perceived it as small. While the PSNR and SSIM index ranked the NLM and CS methods similarly, people preferred the former one. In the case of the second noise level considered, the objective measures were better correlated to human preference. However, the SSIM index estimated that the image quality of the BM3D output is much higher than that of the CS algorithm, while subjects only perceived a small difference between the two methods. In the case of the highest noise level considered, the objective measures and the subjective preference ranked differently the denoising methods. While the BM3D method gave images with the highest PSNR and SSIM values, people chose NLM results as their favorite. Both the PSNR and SSIM metrics estimated a large difference in quality between the outputs of BM3D and CS, while people perceived it as small.

### 11.4.2 Research Avenue to Explore

We have seen examples that once more show how the use of popular metrics like PSNR or SSIM is problematic in the evaluation of different denoising outputs, since they are not well correlated with personal preference. Therefore, there is a need



to create an image metric for image denoising that is based on human perception, according to which noise removal algorithms can be optimized. Alternatively, until such a metric is developed, it would improve results if denoising methods are tuned and ranked via user tests based on perceived appearance.

## 11.5 Conclusion

In this chapter, we have pointed out several avenues to pursue in order to improve denoising results that do not entail developing new denoising algorithms. First, we described how it can be better to denoise a transform of the noisy image rather than denoise the noisy image directly. We mention several possible transforms, and an open problem is to find one that is optimal for denoising, according to a proper image quality metric. Next, we pointed out the importance of having a proper noise model for JPEG pictures, so that a VST can be developed that transforms noise in JPEG images into AWG noise, enabling existing denoising methods to be properly applied to the JPEG case. Finally, we highlighted the fact that while virtually all denoising methods are optimized and validated in terms of the PSNR or SSIM measures, these metrics are not well correlated with perceived image quality and therefore it could be better to optimize the parameter values of denoising methods according to subjective testing. A remaining challenge is to develop perceptually based image quality metrics that match observer preference.

**Acknowledgements** This work has received funding from NSF-DMS # 1320829, from the European Research Council (ERC) under Starting Grant ref. 306337, from the European Union's Horizon 2020 research and innovation program under grant agreement number 761544 (project HDR4EU) and under grant agreement number 780470 (project SAUCE), by the Spanish government and FEDER Fund, grant ref. TIN2015-71537-P (MINECO/FEDER,UE), and by the Icrea Academia Award.

## References

1. <http://ip4ec.upf.edu/realisticnoise>
2. <http://r0k.us/graphics/kodak/>
3. <https://photography.tutsplus.com>
4. Anscombe FJ (1948) The transformation of Poisson, binomial and negative-binomial data. *Biometrika* 35(3):246–254
5. Batard T, Berthier M, Spinor Fourier transform for image processing. *IEEE J Sel Topics Signal Process* 7(4):605–613 (2013)
6. Batard T, Bertalmío M (2013) Generalized gradient on vector bundle-application to image denoising. In: *Lecture Notes Computer Science*, vol 7893, pp 12–23
7. Batard T, Bertalmío M (2014) On covariant derivatives and their applications to image regularization. *SIAM J Imag Sci* 7(4):2393–2422
8. Bertalmío M, Levine S (2014) Denoising an image by denoising its curvature image. *SIAM J Imag Sci* 7(1):187–201

9. Blomgren P, Chan TF (1998) Color TV: total variation methods for restoration of vector-valued images. *IEEE Trans Image Process* 7(3):304–309
10. Bresson X, Chan TF (2008) Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Probl Imag* 2(4):455–484
11. Buades A, Coll B, Morel J-M (2005) A non-local algorithm for image denoising. In: *Proceedings IEEE international conference on computer vision and pattern recognition*, vol 2, pp 60–65
12. Buades A, Coll B, Morel J-M (2011) Non-local means denoising. *Image Process On Line* 1
13. Charbonnier P, Blanc-Féraud L, Aubert G, Barlaud M (1994) Two deterministic half-quadratic regularization algorithms for computed imaging. In: *Proceedings of IEEE international conference on image processing*, vol 2. IEEE Computer Society Press, Los Alamitos, pp 168–172
14. Dabov K, Foi A, Katkovnik V, Egiazarian K (2007) Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans Image Process* 16(8):2080–2095
15. Donoho DL (1993) Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data. In: *Proceedings of symposia in applied mathematics*, pp 173–205
16. Ghimpeteanu G, Batard T, Bertalmío M, Levine S (2016) A decomposition framework for image denoising algorithms. *IEEE Trans Image Process* 25(1):388–399
17. Ghimpeteanu G, Batard T, Seybold T, Bertalmío M (2016) Local denoising applied to RAW images may outperform non-local patch-based methods applied to the camera output. In: *IS&T electronic imaging conference*
18. Ghimpeteanu G, Kane D, Batard T, Levine S, Bertalmío M (2016) Local denoising based on curvature smoothing can visually outperform non-local methods on photographs with actual noise. In: *Proceedings of the IEEE international conference on image processing ICIP*
19. Healey GE, Kondepudy R (1994) Radiometric CCD camera calibration and noise estimation. *IEEE Trans PAMI* 16(3):267–276
20. Hahn J, Tai X-C, Borok S, Bruckstein AM (2011) Orientation-matching minimization for image denoising and inpainting. *Int J Comput Vis* 92(3):308–324
21. Johnson GM, Fairchild MD (2000) Sharpness rules. In: *Proceedings IS&T/SID eighth color imaging conference*, pp 24–30
22. Kayargadde V, Martens JB (1996) Perceptual characterization of images degraded by blur and noise: experiments. *J Opt Soc Am* 13(6):1166–1177
23. Kurihara T, Manabe Y, Aoki N, Kobayashi H (2008) Digital image improvement by adding noise: an example by a professional photographer. *Image Qual Syst Perf V* 6808:1–10
24. Kurihara T, Aoki N, Kobayashi H (2009) Analysis of sharpness increase by image noise. *Human Vis Electron Imag XIV* 7240:1–9
25. Lebrun M, Colom M, Morel JM (2014) The noise clinic: a universal blind denoising algorithm. In: *IEEE international conference on image processing*, pp 2674–2678
26. Lebrun M, Buades A, Morel JM (2013) A nonlocal Bayesian image denoising algorithm. *SIAM J Imag Sci* 6(31):1665–1688
27. Lebrun M (2012) An analysis and implementation of the BM3D image denoising method. *Image Process On Line* 2:175–213
28. Levin A, Nadler B (2011) Natural image denoising: optimality and inherent bounds. In: *Proceedings IEEE international conference on computer vision and pattern recognition* 2:2833–2840
29. Levin A, Nadler B, Durand F, Freeman WT (2012) Patch complexity, finite pixel correlations and optimal denoising. MIT-Computer Science and Artificial Intelligence Laboratory
30. Lysaker M, Osher S, Tai XC (2004) Noise removal using smoothed normals and surface fitting. *IEEE Trans Image Process* 13(10):1345–1357
31. Luisier F, Blu T, Unser M (2011) Image denoising in mixed Poisson-Gaussian noise. *IEEE Trans Image Process* 20(3):696–708
32. Mäkitalo M, Foi A (2011) A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation. *IEEE Trans Image Process* 20(9):2697–2698
33. Mäkitalo M, Foi A (2013) Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. *IEEE Trans Image Process* 22(1):91–103

34. Malvar HS, He L, Cutler R (2004) High-quality linear interpolation for demosaicing of Bayer-patterned color images. In: IEEE international conference on acoustics, speech and signal processing, pp 485–488
35. Mittal A, Moorthy AK, Bovik AC (2011) No-reference image quality assessment in the spatial domain. *IEEE Trans Image Process* 21:4695–4708
36. Mittal A, Soundararajan R, Bovik AC (2013) Making a completely blind image quality analyzer. *IEEE Signal Process Lett* 22(3):209–212
37. Nam S, Hwang Y, Matsushita Y, Kim SJ (2016) A holistic approach to cross-channel image noise modeling and its application to image denoising. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1683–1991
38. Osher S, Burger M, Goldfarb D, Xu J, Yin W (2005) An iterative regularization method for total variation-based image restoration. *Multiscale Model Simul* 4(2):460–489
39. Pambrun JF, Noumeir R (2015) Limitations of the SSIM quality metric in the context of diagnostic imaging. In: Proceedings of the IEEE international conference on image processing ICIP
40. Plötz T, Roth S (2017) Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
41. Rahman T, Tai X-C, Osher S (2007) A tv-stokes denoising algorithm. In: Lecture notes on computer science, vol 4485, pp 473–483
42. Romano Y, Elad M (2015) Boosting of image denoising techniques. *SIAM J Imag Sci* 8(2):1187–1219
43. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. *Phys D Nonlinear Phen* 60(1–4):259–268
44. Seybold T, Cakmak Ö, Keimel C, Stechele W (2014) Noise characteristics of a single sensor camera in digital color image processing. In: Color and imaging conference, pp 53–58
45. Sheikh HR, Bovik AC (2006) Image information and visual quality. *IEEE Trans Image Process* 15(2):430–444
46. Trussell HJ, Zhang R (2012) The dominance of Poisson noise in color digital cameras. In: IEEE international conference in image processing, pp 329–332
47. Wang Z, Bovik AC (2002) A universal image quality index. *IEEE Signal Process Lett* 9:81–84