

# Several approaches to improve noise removal in photographic images

Viorica Gabriela Ghimpeteanu

---

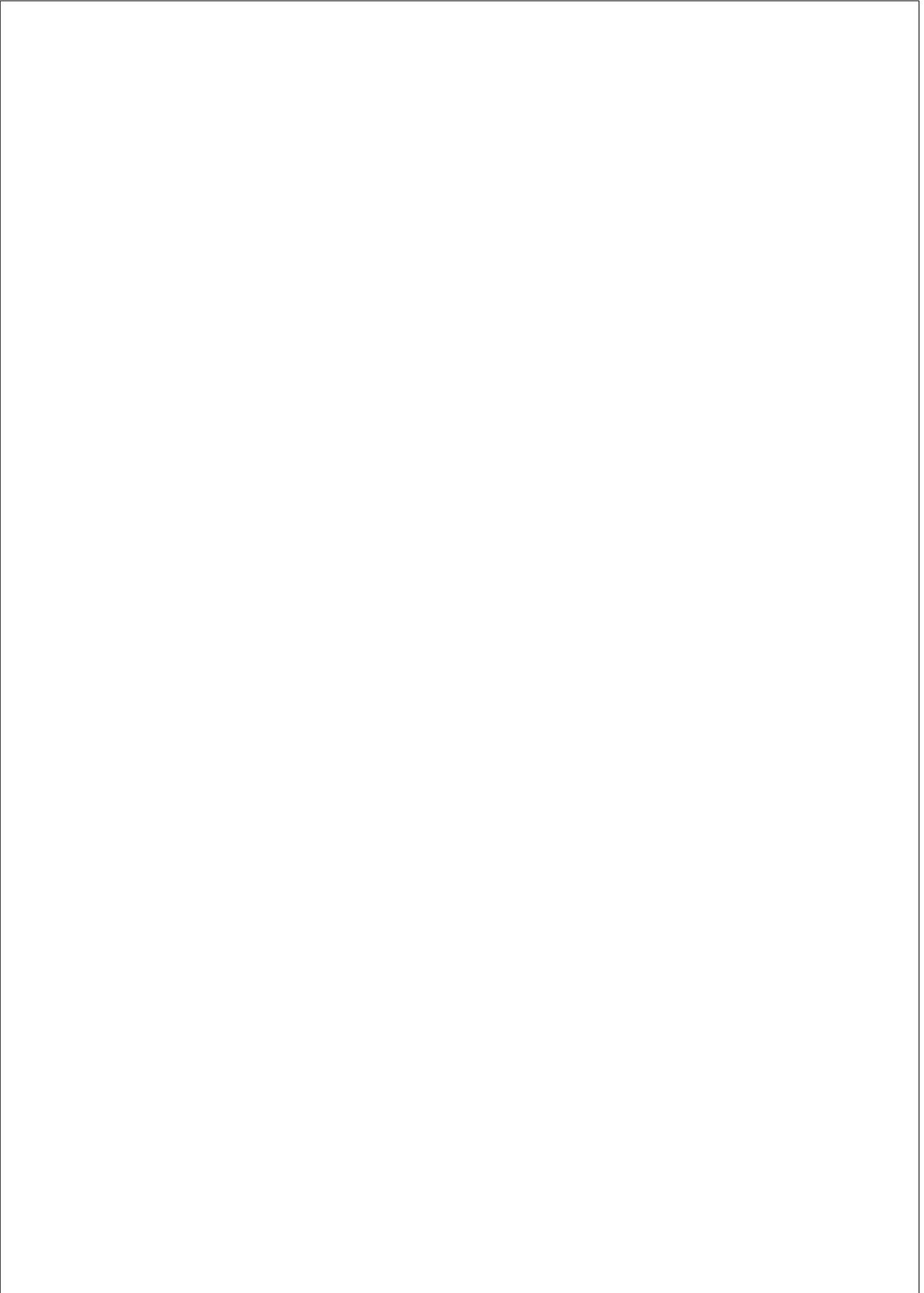
DOCTORAL THESIS UPF / YEAR 2017

**Thesis Advisors:** Marcelo Bertalmío and Thomas Batard  
Department of Information and Communication Technologies





To my family and Juan Pablo



## Acknowledgments

First of all, I would like to thank my supervisors Marcelo Bertalmío and Thomas Batard for their generous guidance, help and encouragement, that made possible the research leading to this thesis.

Many thanks to the co-authors of my publications for all their great support: Stacey Levine, Tamara Seybold and David Kane.

I would like to thank to all the participants of our psychophysical experiments.

Very special thank you for my colleagues that became friends, always helpful, supportive and fun: Raquel, Maria, Waqas, Praveen, Javier, Jiji, Antoine.

A warmly thank you to my family and to Juan Pablo.



## Abstract

Noise acquisition is an unavoidable component when capturing photographs, even in the case of current state of the art cameras. This problem is even accentuated when the lighting conditions are not ideal. Therefore, removing the noise present in the captured image is still an essential task in the camera image processing pipeline.

In this thesis, we analyze several approaches to improve current image denoising methods. First, we propose a general framework that can improve a denoising method, motivated by a simple principle: for any algorithm, the smaller the noise level, the higher the quality of the denoised image. Therefore, by carefully choosing an image decomposition of the noisy image into less noisy one(s) and applying the algorithm on the latter, the performance of any denoising method can increase.

Second, we accentuate the importance of using a realistic noise model for testing any denoising methods, as in the usual AWG scenario the results can be extremely different. The noise model can be estimated on RAW images, as the camera processing pipeline alters the noise, and denoising becomes a challenge when applied on camera output. We show how a local method applied on RAW can outperform a non-local one applied on camera output, in the realistic noise scenario.

Finally, in this thesis we propose a fast, local denoising method where the Euclidean curvature of the noisy image is approximated in a regularizing manner and a clean image is reconstructed from this smoothed curvature. User preference tests show that when denoising real photographs with actual noise our method produces results with the same visual quality as the more sophisticated, non-local algorithms, but at a fraction of their computational cost. These tests also highlight the limitations of objective image quality metrics like PSNR and SSIM, which correlate poorly with user preference.

## Resum

L'adquisició de soroll és una component ineludible quan capturem una fotografia, fins i tot en el cas de les càmeres d'última generació. Aquest problema s'accentua encara més quan les condicions d'illuminació no són ideals. Per tant, l'extracció del soroll que està present a la imatge capturada continua sent una tasca essencial dintre del processament d'imatges de la càmera.

En aquesta tesi, analitzem diversos enfocaments per millorar els mètodes actuals d'extracció de soroll. En primer lloc, proposem un marc general que permet millorar un mètode d'extracció. Aquest marc està motivat per un principi senzill: per a qualsevol algoritme, com més petit sigui el nivell de soroll a l'imatge original, més alta serà la qualitat de la imatge de sortida. Per tant, escollint acuradament una descomposició de la imatge sorollosa en una altra amb menys soroll i aplicant l'algoritme en aquesta última, podem augmentar el rendiment de qualsevol mètode d'extracció de soroll.

En segon lloc, remarquem la importància d'utilitzar un model de soroll realista per a evaluar qualsevol mètode d'extracció de soroll, ja que els resultats en imatges realistes poden divergir enormement en comparació amb l'escenari habitual de suposar AWG. Amb aquest fi, estimem un model de soroll en imatges RAW, ja que el processament de l'imatge dintre de la càmera altera el soroll, i l'extracció de soroll es converteix en un desafiament al no seguir el model AWG. Mostrem que, quan suposem un model de soroll realista, un mètode local aplicat a RAW pot superar un de no-local aplicat a la sortida de la càmera.

Finalment, en aquesta tesi proposem un mètode ràpid i local d'extracció de soroll on la curvatura euclidiana de la imatge sorollosa s'aproxima de manera regularitzadora i es reconstrueix una imatge neta d'aquesta curvatura suavitzada. Les proves de preferència dels usuaris mostren que el nostre mètode produeix resultats amb la mateixa qualitat visual que els algorismes més sofisticats i no-locales, però amb una fracció del seu cost computacional. Aquestes proves també posen de relleu les limitacions de mètriques de qualitat d'imatge objectives com PSNR i SSIM, que es correlacionen malament amb la preferència dels usuaris.

---

---

## Contents

---

<b>Nomenclature</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xx</b>
<b>List of Tables</b>	<b>xxii</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Contributions . . . . .	5
1.2 Publications . . . . .	6
1.3 Thesis Outline . . . . .	7
<b>2 DIGITAL IMAGE CREATION</b>	<b>9</b>
2.1 Capturing scene light with the camera’s image sensor . . . . .	9
2.1.1 Acquisition process . . . . .	9
2.1.2 Image sensors . . . . .	11
2.2 Digital image processing pipeline . . . . .	12
<b>3 NOISE IN A DIGITAL IMAGE</b>	<b>17</b>
3.1 Noise at the image sensors level . . . . .	18
3.1.1 CCD vs CMOS . . . . .	23
3.2 Optimal choice of the exposure triangle: aperture, ISO and shutter speed . . . . .	25

3.3	Noise and the dynamic range . . . . .	28
3.4	Noise perception . . . . .	30
<b>4</b>	<b>RELATED WORK IN IMAGE DENOISING</b>	<b>45</b>
4.1	Classic denoising methods . . . . .	45
4.2	Non-local methods . . . . .	47
4.3	Current trends: A global approach and deep neural networks . .	52
4.4	On denoising techniques . . . . .	54
4.4.1	Principles . . . . .	54
4.4.2	Tools . . . . .	55
4.4.3	Comparison . . . . .	57
4.4.4	Conclusion . . . . .	59
<b>5</b>	<b>DENOISING: APPLY IT ON A LESS NOISY EQUIVALENT IM- AGE</b>	<b>61</b>
5.1	Line of similar research . . . . .	62
5.2	Image decomposition in a moving frame . . . . .	64
5.2.1	The gray-level case . . . . .	64
5.2.2	The multi-channel case . . . . .	67
5.2.3	Application to image denoising . . . . .	68
	Gray-level images . . . . .	68
	Color images . . . . .	69
5.3	The noise level is higher on the intensity values of a gray-level image than on its components in a well-chosen moving frame . .	70
5.3.1	Edges . . . . .	70
5.3.2	Homogeneous regions . . . . .	74
5.3.3	Numerical results . . . . .	75
5.4	Experiments . . . . .	76
5.4.1	The moving frame approach applied to the Vectorial Total Variation-based denoising method . . . . .	77
	Gray-level images . . . . .	77

Color images . . . . .	78
5.4.2 The moving frame approach applied to the Non-Local Means algorithm. . . . .	82
The case of gray-level images . . . . .	83
The case of color images . . . . .	84
5.4.3 The moving frame approach applied to the Block Matching and 3D Filtering algorithm. . . . .	87
5.5 Conclusion . . . . .	88
<b>6 DENOISING: APPLY IT ON RAW IMAGES WITH ACTUAL REAL NOISE</b>	<b>91</b>
6.1 Motivation . . . . .	92
6.2 Estimate a realistic noise model on RAW images and create a test set . . . . .	95
6.2.1 Simulate the camera processing pipeline . . . . .	95
6.2.2 The noise model . . . . .	97
6.2.3 Validation of the noise model . . . . .	99
6.3 Testing denoising methods under different noise models: AWG vs. realistic . . . . .	102
6.4 A local denoising method for RAW images with realistic noise .	105
6.4.1 The denoising method . . . . .	105
6.4.2 Adapting a TV-based denoising method to the signal-dependent noise model . . . . .	106
6.4.3 Refine the denoising output by recovering lost details . .	108
6.5 Compare local denoising on RAW to non-local denoising on camera output . . . . .	109
6.6 Analysis of denoising experiments . . . . .	114
6.6.1 Apply the local denoising method at different locations in the camera processing pipeline . . . . .	114
6.6.2 Compare local with low-complexity non-local denoising applied at the same stage of the image processing chain .	115

6.7	Experiments on cinema camera images . . . . .	119
6.7.1	Camera processing pipeline . . . . .	119
6.7.2	Denoising experiments . . . . .	120
	Compare local denoising on RAW to non-local denoising on camera output . . . . .	120
	Compare local and low-complexity non-local denoising applied at the same stage of the image process- ing chain . . . . .	122
6.8	Conclusion . . . . .	124
<b>7</b>	<b>DENOISING: TUNE PARAMETERS FOR SUBJECT PREFER- ENCE NOT FOR OBJECTIVE MEASURES</b>	<b>127</b>
7.1	Motivation . . . . .	128
7.2	A new local denoising method . . . . .	129
	7.2.1 Foundations of the model . . . . .	129
	7.2.2 The proposed approach . . . . .	130
7.3	Experiments . . . . .	131
	7.3.1 Kodak database images . . . . .	133
	7.3.2 Real noise images . . . . .	135
7.4	Conclusion . . . . .	137
<b>8</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>141</b>

---

## Nomenclature

---

**2D** Two Dimensions

**3D** Three Dimensions

**AVTVE** Anscombe Vectorial Total Variation Edge

**AWG** Additive White Gaussian

**BLS** Bayesian least squares

**BM3D** Block-Matching and 3D Filtering

**CCD** Charge Coupled Device

**CDS** Correlated Double Sampling

**CFA** Color Filter Array

**CMOS** Complementary Metal Oxide Semiconductor

**CNN** Convolutional Neural Network

**CRT** Cathode Ray Tube

**CS** curvature smoothing

**DCT** Discrete Cosine Transform

**DnCNN** Denoising Convolutional Neural Network

**FoE** Field of Experts

**FPN** Fixed-Pattern Noise

**GMM** Gaussian Mixture Model

**GSM** Gaussian Scale Mixture

**HDR** High Dynamic Range

**HVS** Human Visual System

**K-SVD** K-Means Singular Value Decomposition

**LCD** Liquid Crystal Display

**LSSC** Learned simultaneous sparse coding

**MLP** Multi-Layer Perceptron

**MRF** Markov Random Fields

**MSE** mean square error

**NLM** Non-Local Means

**PSNR** Peak Signal to Noise Ratio

**RGB** Red Green Blue

**SSIM** Structural Similarity Index Measure

**SURE** Stein’s Unbiased Risk Estimator

**TNRD** Trainable Nonlinear Reaction Diffusion

**VOD** Vertical Overflow Drain

**VST** Variance Stabilizing Transformation

**VTV** Vectorial Total Variation

---

---

## List of Figures

---

1.1	The first digital camera. . . . .	1
2.1	Visible light as a part of the electromagnetic spectrum. . . . .	10
2.2	A scene image example and an approximation of how the camera represents it, using the Bayer pattern. . . . .	11
2.3	Image example to illustrate the camera processing pipeline. . . . .	12
2.4	The gamma correction curve on a CRT. . . . .	15
2.5	An image example of lossy compression . . . . .	16
3.1	Noise at the sensor level. . . . .	19
3.2	Figure illustrating temporal noise and fixed pattern noise. . . . .	19
3.3	Figure illustrating the smear image problem and its solution. . . . .	22
3.4	Figure illustrating the blooming image problem and its solution. . . . .	22
3.5	An illustration of the rolling shutter effect. . . . .	23
3.6	A simulation of the noise at the CMOS sensor level. . . . .	23
3.7	An illustration of the microlenses used on the CMOS chip, such that all light rays are focused onto the photo sensor. . . . .	25
3.8	Figure illustrating the aperture blades. . . . .	25
3.9	Figure illustrating the effect of different camera parameters. . . . .	26
3.10	Figure illustrating images taken with a wide aperture and a narrow one. . . . .	27

3.11	Figure illustrating images taken with a slow shutter speed and a fast one. . . . .	28
3.12	Figure illustrating images taken with a low ISO value and a high one. . . . .	28
3.13	Figure comparing a pixel cell from an ideal image sensor to a real one with dynamic range limitations. . . . .	29
3.14	Figure illustrating a low dynamic range scene and a high dynamic range one. . . . .	30
3.15	An example of a photograph with grain. . . . .	31
3.16	Figure illustrating how darker regions appear more noisy than brighter ones in a digital image. . . . .	31
3.17	Figure illustrating how a certain amount of noise of a certain type can improve the appearance of a digital photograph, giving a sharper look to it. . . . .	32
3.18	Figure illustrating the different appearance of luminance noise and chroma noise in a digital image. . . . .	33
3.19	Figure illustrating how for edges, the perceived sharpness decreases with increasing added noise, while for texture it increases up to a certain point. . . . .	34
3.20	Figure illustrating the concept of adaptation to image noise. . . . .	35
3.21	Figure illustrating three types of noise used for the psychophysical experiments regarding adaptation to image noise. . . . .	36
3.22	Figure illustrating the stimulus including the original image on the left and the a version of it on the right, while the grey background also has added noise. . . . .	37
3.23	Figure illustrating three types of noise used for the psychophysical experiments regarding adaptation to image noise. . . . .	38
3.24	Figure illustrating how the amount of analog black and white print noise changes with pixel values and one example of the amount of noise added by a professional photographer to a digital image to improve its appearance. . . . .	38
3.25	Figure illustrating the visual different appearance of noise created with a Gaussian distribution and chi-squared distribution. . . . .	39

3.26	Figure illustrating a piece of art created by Thomas Ruff, in his search for beauty in JPEGs. . . . .	40
3.27	Figure illustrating a glitch piece of art created in 2002 by Ant Scott, with the title “Digital screenshot”. Figure is from [Menkman, 2011]. . . . .	41
3.28	Figure illustrating the Monglot Glitch Software Interface created in 2011 by Menkman and Larsby. . . . .	41
3.29	Figure of a computer generated image example, processed to mimic the photographic film development, to which noise is added for a more realistic appearance. . . . .	42
3.30	Figure illustrating Perlin noise and its appearance when added to a rendered image for giving visually complex details. . . . .	43
3.31	Figure illustrating the appearance of different popular procedural noise functions used in computer graphics. . . . .	43
4.1	Figure illustrating the main idea of NLM denoising: in order to denoise a certain noisy patch, similar patches are contributing with different weights directly proportional to their similarity. . .	48
4.2	Figure illustrating the effect of several general steps that can improve any denoising approach. . . . .	56
5.1	Moving frame encoding the local geometry of a gray-level image.	65
5.2	From left to right: gray-level image “Lena”, component $J^1$ , component $J^3$ . . . . .	66
5.3	Example of our moving frame approach with VTV [Bresson and Chan, 2008] as denoising method. . . . .	82
5.4	Example of our moving frame approach with NLM [Buades et al., 2005a] as denoising method. . . . .	87
5.5	Example of our moving frame approach with BM3D [Dabov et al., 2007] as denoising method. . . . .	90
6.1	Figure illustrating our image test set. . . . .	96
6.2	Figure illustrating the Colorchecker set-up that we used for estimating the noise model. . . . .	97

6.3	Plot of variance as function of the mean, for a RAW image scaled between 0 and 1. . . . .	99
6.4	Crops from a noisy image example from our test set, with noise level corresponding to different ISO settings . . . . .	100
6.5	Comparison between an actual real noise photograph and synthesized noisy images obtained by: adding Gaussian noise with variance given by our realistic noise model to the RAW image, adding Gaussian noise of constant variance to the RAW image and adding Gaussian noise of constant variance to the camera output. . . . .	101
6.6	Comparison of BM3D, NLM and VTV applied on the camera output, under two noise models. . . . .	102
6.7	Comparison of VTV, NLM and BM3D denoising methods under AWG on camera output and a realistic noise model. . . . .	104
6.8	Evolution of our TV-based local denoising experiments, under the proposed realistic noise model, with average PSNR values computed over our image test set. . . . .	107
6.9	Comparison of our local TV-based denoising methods applied on the demosaicked RAW, under the proposed realistic noise model. . . . .	109
6.10	Comparison between the local denoising method AVTVE applied on the demosaicked RAW to the NLM and BM3D denoising algorithms applied on the camera output, under the proposed realistic noise model. . . . .	112
6.11	Comparison of the local denoising method AVTVE applied on the demosaicked RAW to the BM3D and NLM denoising methods applied on the camera output, under the proposed realistic noise model. . . . .	113
6.12	Comparison between our local denoising approach AVTVE applied on the demosaicked RAW and on the RAW image. . . . .	115
6.13	Comparison of the non-local low-complexity BM3D-based denoising methods applied on the demosaicked RAW, under the proposed realistic noise model. . . . .	116

6.14	Comparison between the local denoising method AVTVE to the low-complexity BM3D, ABM3D and ABM3DE applied on the demosaicked RAW, under the proposed realistic noise model. . .	117
6.15	Comparison of our local denoising method AVTVE to ABM3D and ABM3DE denoising algorithms applied on the demosaicked RAW, under the proposed realistic noise model. . . . .	118
6.16	Comparison of our local denoising method AVTVE to BM3D and NLM denoising algorithms applied on the cinema camera output.	122
6.17	Comparison of our local denoising method AVTVE to ABM3D and ABM3DE denoising approaches applied on the demosaicked RAW. . . . .	123
6.18	Comparison of our local denoising method AVTVE to the BM3D and NLM denoising methods applied on the camera output and the low-complexity denoising strategies ABM3D and ABM3DE applied on the demosaicked RAW. . . . .	125
7.1	Figure illustrating our test images for the AWG noise denoising experiment, that are crops from Kodak data set. . . . .	131
7.2	Figure illustrating real actual noise photographs that we used as test images for our denoising experiment. . . . .	132
7.3	Figure comparing the PSNR and SSIM results computed on the Kodak data base to the results of the psychophysical experiment for comparing our proposed local denoising method CS to BM3D and NLM. . . . .	133
7.4	Visual comparison for one test crop from an image from Kodak data base, PSNR and SSIM results, and user preferences. . . . .	134
7.5	Average visual comparison results for 30 real-noise images, for comparing our denoising method CS to BM3D and NLM. . . . .	136
7.6	Real-noise image crops and user preferences. . . . .	136
7.7	An example of real-noise image denoising: user preference and visual comparison. . . . .	138
7.8	An example of real-noise image denoising: user preference and visual comparison. . . . .	139

7.9 An example of real-noise image denoising: user preference and  
visual comparison. . . . . 140

---

## List of Tables

---

5.1	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 $\mu$ . . . . .	76
5.2	Comparison of the standard and our moving frame approach with $\mu = 0.001$ for the $VTV$ -based denoising method, at different noise levels. Average PSNR and SSIM Index (x100) over the kodak database: the gray-level case. . . . .	79
5.3	Our moving frame approach with optimal value of $\mu$ for the $VTV$ -based denoising method, at different noise levels. Average PSNR and SSIM Index (x100) over the kodak database: the gray-level case. . . . .	79
5.4	Comparison of the standard and our moving frame approach with $\mu = 0.001$ for the $VTV$ -based denoising method, at different noise levels. Average PSNR and SSIM Index (x100) over the kodak database: the color case. . . . .	81
5.5	Our moving frame approach with optimal value of $\mu$ for the $VTV$ -based denoising method, at different noise levels. Average PSNR and SSIM Index (x100) over the kodak database: the color case. . . . .	81
5.6	Comparison of the standard approach and our moving frame approach with $\mu = 0.001$ for $NLM$ , at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter $\sigma_3$ over the kodak database: the gray-level case. . . . .	85

5.7	Comparison of the standard approach and our moving frame approach with $\mu = 0.001$ for <i>NLM</i> , at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter $\sigma_3$ over the kodak database: the color case. . . . .	88
5.8	Comparison of the standard approach and our moving frame approach with $\mu = 0.001$ for <i>BM3D</i> , at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter $\sigma_3$ over the kodak database: the gray-level case. . . . .	89
5.9	Comparison of the standard approach and our moving frame approach with $\mu = 0.001$ for <i>BM3D</i> , at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter $\sigma_3$ over the kodak database: the color case. . . . .	89
6.1	Average noise levels given by different ISO noise curves on our test set. . . . .	103
6.2	Average running time (sec) for one 1000x2000 color test image of <i>AVTVE</i> , <i>BM3D</i> and <i>NLM</i> , for different noise levels given by different ISO settings. . . . .	112
6.3	Average running time (sec) for one 1000x2000 color test image of <i>AVTVE</i> , <i>ABM3D</i> and <i>ABM3DE</i> , for different noise levels given by different ISO settings. . . . .	119
6.4	Average running time (sec), for one 100x200 color test image, of <i>AVTVE</i> , <i>ABM3D</i> and <i>ABM3D</i> , for different noise levels. . . . .	124
7.1	Optimized parameter value $\epsilon_2$ as a function of $\sigma$ . . . . .	134

# CHAPTER 1

---

## Introduction

---

The history of digital still cameras began in 1975, when Steven Sasson (an engineer at Eastman Kodak) created the first digital camera [Prakel, 2009]. Fig.1.1



Figure 1.1: The first digital camera. Figure is from [<https://petapixel.com/>, 2017]

illustrates the invention that weighed 8 pounds, used a CCD sensor and had a resolution of 0.01 MP. The black-and-white images were read from a digital cassette tape and a television set was used as display. The invention was not received with great enthusiasm, as the Kodak company was monopolizing the

photography market at every step of the photographic process with their film cameras and products (film rolls and printing). Also, they doubt anyone would be interested to see their photographs on a television set. These days, digital cameras are strongly integrated in our everyday life.

Noise represents to digital photography what grain represents to film. However, while grain can have a positive connotation when used with an artistic intention, digital noise is mostly unwanted. Although the image quality greatly improved since the first black and white digital image of 10000 pixels, noise in digital cameras is still an important issue to solve.

Image denoising is a fundamental task in the field of image processing, due to the fact that acquiring an image that represents faithfully the real scene is a very difficult task. With image acquisition, we get also noise acquisition. In one of the most popular models for an image representation,  $\hat{I}$  is the result of the sum between the ideal noiseless image  $I$  and the noise image  $n$

$$\hat{I} = I + n, \tag{1.1}$$

where  $n$  is assumed to be additive white Gaussian (AWG) noise. The origin of the term white noise comes from the physical properties of white light, which contains equally almost all the frequencies in the visible spectrum, as seen in [Gonzalez and Woods, 2001], and it appears when the Fourier spectrum of the noise is constant. Gaussian noise models are frequently used in practice due to the tractability in spacial and frequency domain. For simplification reasons, usually the noise is considered to be independent of the spacial coordinates and uncorrelated to the image.

The main source of noise in digital images is the photon shot noise that appears due to photon counting, see [Lebrun et al., 2012] [Buades et al., 2005b]. Each captor of the digital sensor is a square for which the incoming photons are being counted for a certain time equal to the exposure time. The value  $u(i)$  observed by the sensor at pixel  $i$  is a Poisson random variable with standard deviation  $\sqrt{u(i)}$ . For large values of  $u(i)$  (larger than 1000) the Poisson distribution is well approximated by the Gaussian distribution with the same mean and standard variation. For  $u(i) > 10$  the approximation can still be used, but with a small correction. Besides the noise distribution, another problem appears that challenges the approximation of noise model as white additive Gaussian: the pixel value is

signal dependent, as its mean and standard deviation depend on  $u(\tilde{i})$ . The solution is to use a variance-stabilizing transformation, such as the Anscombe transform: instead of denoising the original data, denoise the transformed data. The shot noise is stronger for low lighting conditions. For low count Poisson noise, [Mäkitalo and Foi, 2013] argue that if a proper variance-stabilizing transformation is applied, algorithms designed for homoscedastic (homogeneity of variance) Gaussian noise work just as well as algorithms for signal dependent noise models. This justifies the use of the popular white Gaussian noise model [Lebrun et al., 2012]. Having one general noise model is very important in order to make a comparison between different denoising algorithms.

In [Schoeberl et al., 2012], the authors argue that the shot noise is more and more problematic due to the manufacturing of smaller structures and image sensors with shrinking sizes. To show a comparison, the devices have around  $1\mu m$  while the human eye has photo-receptors larger than  $3\mu m$ . Due to the small pixels, only few photons will reach the sensor for a certain exposure time, so the shot noise will be stronger.

Besides the shot noise, there are other noise sources which can be approximated as additive and white: thermal noise and electronic noise. Thermal noise appears due to the thermal energy of the chip. As it appears also when there is no light, it is called dark-current noise. It is reduced by cooling the sensor of the camera.

Noise does not appear only at the time of capture and image acquisition, but also with image processing. If an image is not in RAW format (the format in which the image is produced directly from the imaging system), it is more difficult to analyze its noise model, see [Lebrun et al., 2012]. The noise model is altered by changing the image format, as the change could include resampling, denoising, contrast changes or compression. All these transformations make the noise model more signal and scale dependent.

For example, one of the most popular image formats is JPEG. This is a lossy image format, which means that the noise is partially removed from the image by compression (it performs a Discrete Cosine Transform and sets to zero the small transform coefficients).

Moreover, image processing operations like demosaicking, white balance and tone curve change the noise - signal dependency, such that the alteration depends on the image itself.

The task of image denoising is to recover the denoised image  $I$  by eliminating the noise in the image  $\hat{I}$ , while keeping intact its geometrical structure. This is an ill-posed problem. With the noise, also small details in the image are removed, therefore the challenge is to remove the noise while keeping the sharpness of an image and not introducing artifacts. There is a trade-off between noise and denoising artifacts like ringing, blur, staircase effect, checkerboard effect, wavelet outliers and others, see [Buades et al., 2005b] [Buades et al., 2006].

In practice, Gaussian noise of different variance values is added to clean natural images, for testing purposes, such that we get a noisy image but also its ground truth, which is necessary for estimating the quality of the denoised output with an objective measure.

The simplest way of measuring the performance of a denoising algorithm is by computing the peak signal to noise ratio (PSNR):

$$PSNR = 10 \log_{10} \frac{(\max_{x,y}(I_c(x,y)))^2}{MSE} \quad (1.2)$$

where  $I_c$  is the original clean image and  $I_d$  is the denoised image.

$$MSE = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (I_c(x,y) - I_d(x,y))^2$$

The PSNR is expressed in decibels (dB). It is assumed that an image with a large PSNR value is an image of a high quality. Due to its simple form, it is very popular in practice, although the performance of an algorithm might be better estimated by a perceptual metric.

Perceptual image quality metrics try to measure how visible are the errors or differences between a reference image and another one that one wants to evaluate, involving features of the human visual system (HVS). These methods used a measure similar to MSE to penalize errors proportional to how visible they are. Under the assumption that the HVS is well trained in detecting structural information, Wang et al. [Wang and Bovik, 2002] developed a Structural Similarity Index Measure (SSIM) that does image quality assessment by estimating the degradation of the structural information. Their idea is that the observed luminance of the surface of a certain object in a scene is the result of the illumination multiplied by the reflectance. While the structure of the objects is related to the reflectance, it is independent of illumination. Therefore, structural information is defined by Wang

et al. as the feature that represents the geometry of a scene, independent of average luminance and contrast. The average luminance is modeled as a function of  $\mu_x$  and  $\mu_y$ , that represent average intensities of the images  $x$  and  $y$  to be compared, and the contrast is estimated with the standard deviation. The expression is given by the following formula:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y \pm C_1)(2\sigma_{xy} \pm C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Here the variables  $C_1$  and  $C_2$  are introduced for computation stability reasons, to avoid division by zero. The comparison is done between one image that we know it is of great quality, and another one that we want to evaluate. By measuring the similarity between the two, we evaluate also the quality of our image. This equation is also consistent with Weber’s law, that expresses that HVS is sensitive to relative luminance change, and not to absolute luminance change.

## 1.1 Contributions

Our first main contribution of this thesis is introducing a general denoising strategy that can improve the performance of any denoising method, by taking more carefully into account the local geometry of the noisy image, by processing the moving frame that describes the graph of a scaled version of the image. For this, we presented both a theoretical and a practical justification. The theoretical motivation for using a moving frame for image denoising is showing that the components of the noisy image in this frame are less affected by noise (AWG) than the image itself. Therefore, given any denoising method, better results are obtained by denoising the image components than the image itself. In practice, we demonstrated the consistency of the moving frame approach by showing that it systematically improved the performance of three types of denoising techniques: a local variational method (VTV), a patch-based method (NLM), and a method combining a patch-based approach with a filtering in spectral domain approach (BM3D).

Another contribution is to highlight the limitations of the popular AWG assumption in the field of image denoising. While denoising seems to be practically solved for the AWG noise, for the camera output photographs it remains an open problem. To obtain a more realistic denoising scenario, we created a realistic noise

model estimated on RAW images and we make publicly available our database of clean and noisy images, both the 12-bit RAW format and the 8-bit uncompressed camera output format. We show that not only the performance of denoising methods decays when switching from AWG to a realistic noise model, but also the ranking of denoising methods can change. Another main contribution is to show how a local TV-based denoising method applied on RAW can outperform (visually and in terms of PSNR) non-local patch-based methods applied to camera outputs, at a fraction of the running time. Moreover, when both are applied on RAW images, the local approach outputs better or comparable results (visually and in terms of PSNR) to those of a low-complexity implementation of a non-local patch-based denoising method, but for a smaller running time. As opposed to complex patch-based denoising methods that are computationally too intensive for camera implementation without simplification, a local TV-based approach is a good candidate for in-camera denoising.

Our last main contribution is to highlight the limitations of two popular objective measures, PSNR and SSIM, used in general for optimizing the parameters of denoising methods and evaluate their output quality. With this purpose, we have compared a local denoising method (that reconstructs a clean image from a smoothed version of the Euclidean curvature of the original noisy input) to the non-local algorithms NLM and BM3D (that are computationally much more intensive). Our experiments show that for moderate noise levels that commonly appear in properly-exposed photographs, our proposed method, NLM and BM3D all have the same average performance on real-noise images, in terms of user preference. Moreover, the PSNR and SSIM measures do not correlate with the user preference on an image-by-image basis. Also, we show that the experimental results obtained with the AWG noise model cannot be extrapolated to the real actual noise scenario.

## 1.2 Publications

This thesis is based on our work described in the following papers.

### **Journals:**

- G. Ghimpeteanu, T. Batard, M. Bertalmío, and S. Levine. A decomposition

framework for image denoising algorithms. *IEEE Transactions on Image Processing*, 25(1):388399, 2016.

- G. Ghimpeteanu, T. Batard, T. Seybold, and M. Bertalmío. On denoising photographs having actual noise. Submitted to *IEEE Transactions on Image Processing*.

#### **Conferences:**

- G. Ghimpeteanu, T. Batard, M. Bertalmío, S. Levine. Denoising an image by denoising its components in a moving frame. *International Conference on Image and Signal Processing (ICISP)*, 2014 (best paper award).
- G. Ghimpeteanu, T. Batard, T. Seybold, and M. Bertalmío. Local denoising applied to RAW images may outperform non-local patch-based methods applied to the camera output. In *proc. of IS&T Electronic Imaging Conference*, 2016.
- G. Ghimpeteanu, D. Kane, T. Batard, S. Levine, and M. Bertalmío. Local denoising based on curvature smoothing can visually outperform non-local methods on photographs with actual noise. In *proc. of the IEEE International Conference on Image Processing (ICIP)*, 2016.

### **1.3 Thesis Outline**

Hereby, we introduce the structure of the thesis, following the Introduction. In Chapter 2, we describe the process of capturing and processing a digital image. Chapter 3 illustrates how image acquisition comes, inevitably, with noise acquisition; this justifies the importance of the denoising task in image processing. A review of some of the most important denoising methods and their basic principles is covered in the Chapter 4. Having a denoising method introduces the need of output quality evaluation; we briefly analyze how the evaluation of a denoising method can be performed. Chapter 5 presents our simple moving frame approach that can improve any denoising method by using the local geometry of the noisy image. Chapter 6 underlines the need of a realistic noise model instead of the popular AWG noise for image denoising. We also show how a local approach can outperform non-local patch-based denoising methods, when

the former is applied on the RAW image and the latter on the camera output. Chapter 7 introduces an experiment indicating that objective measures such as PSNR and SSIM are not well correlated to human preference. We show how for real actual noise of moderate intensity, a curvature-based denoising method performs visually the same, on average, as non-local patch-based methods.

## CHAPTER 2

---

### Digital image creation

---

#### 2.1 Capturing scene light with the camera’s image sensor

Light is electromagnetic radiation from a particular range of the electromagnetic spectrum. The electromagnetic spectrum represents all known frequencies and the associated wavelengths of the known photons. Visible light (visible to the human eye) has wavelengths in the range of 380-740 nm, as illustrated in Fig.2.1. Generally, and in this thesis, light refers to visible light, although in physics, this concept is associated to any wavelength of the electromagnetic spectrum: from the shortest, measuring only a fraction of the diameter of the atom (the gamma rays) to the largest, in the range from 1mm to several meters (the radio waves). Light is emitted and absorbed in small packets (photons) and has a dual nature of both particle and wave.

##### 2.1.1 Acquisition process

In order to capture a photography with a digital camera, when we press the camera’s shutter button, the aperture (an opening controlling the amount of light) opens such that light enters and it is focussed by the lens onto a light detector. The lens can vary from a single piece to a sophisticated and precise arrangement of more than 20 lens pieces. The light detector is an image sensor that converts the light into electronic signals by using an array of millions of light cavities called

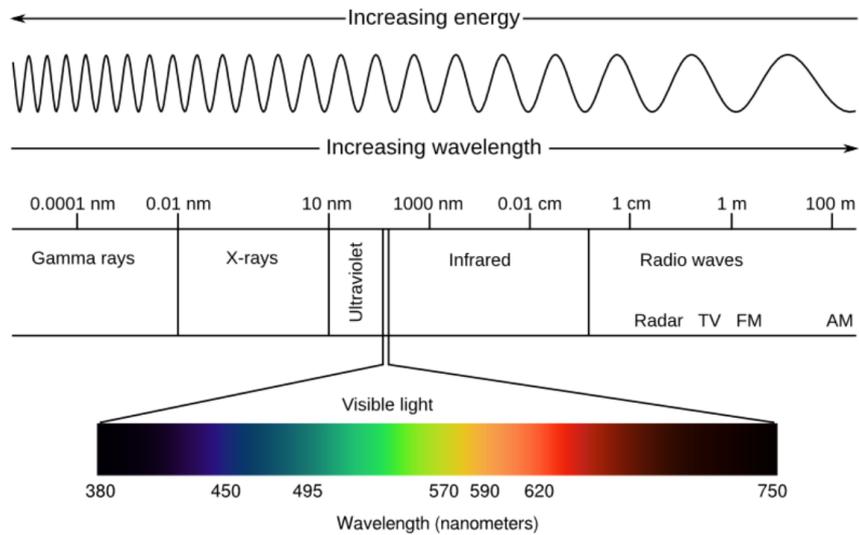


Figure 2.1: Visible light as a part of the electromagnetic spectrum. Figure is from [Bertalmío, 2014].

“photosites”. Once the image is captured, it is stored into an internal memory chip. When the shutter button is released, the exposure finishes and each photosite is closed. The number of photons that entered each light cavity gives a certain intensity level given by the bit depth (for example 0-4095 for a 12-bit image). This is how a greyscale image can be created. In order to obtain a color image, a color filter array (CFA) is placed over the photosites, such that each one captures only one particular color (red, green or blue) and the other two are interpolated later on. The Bayer pattern is one of the most popular types of color filter arrays. Fig.2.2b shows how the camera reproduces the scene from Fig.2.2a using the steps described above. The image obtained at this point is called a RAW image and some cameras allow recording it. As we can see, it is not ready to be displayed yet, but it allows complex off-camera processing.

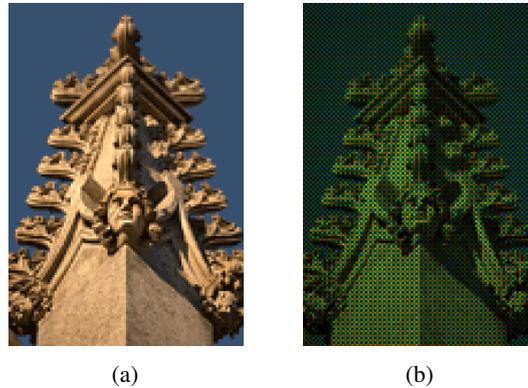


Figure 2.2: A scene image example (a) and an approximation of how the camera represents it (b), using the Bayer pattern. Figure is from [<http://www.cambridgeincolour.com>, 2017]

### 2.1.2 Image sensors

An image sensor is a semiconductor device that transforms an optical image formed by the camera lens into an electronic signal. In the context of photography, the sensor is designed to capture light visible to the human eye and convert it into signal charge. This is possible due to the so-called “photoelectric effect”, which is the emission of electrons (photo electrons) when some material is exposed to light, caused by the energy transferred from light to the particle. The sensor is an array of cells called pixels (picture elements). The number of electrons stored in each pixel is proportional to the number of photons that struck that pixel. When an exposure has been done, by scanning the image array, the electrons corresponding to each cell are converted to a number that estimates how dark or light that pixel should be, and stored in the image file. Until recently, the most popular image sensor was the charged-couple device (CCD). These days, complementary metal-oxide semiconductors (CMOS) are replacing the CCD sensors. While they both have the same functionality of creating electrical current proportional to the amount of light captured, they have a different implementation of the scanning of the image array [Bertalmío, 2014]. The CCD transfers charges from the first row of the sensor array to the last one, pixel by pixel, and then horizontally moves them towards one amplifier; each charge waits its turn to be transformed into voltage. Contrarily, the CMOS sensor has an amplifier for each

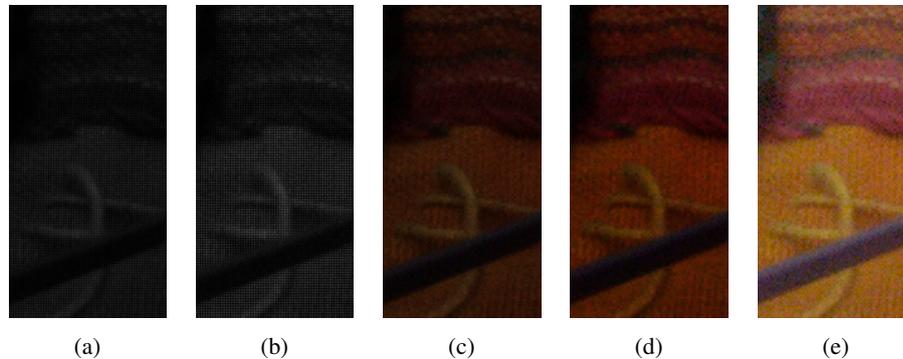


Figure 2.3: Image example to illustrate the camera processing pipeline. From left to right: (a) RAW original sensor image, (b) result after applying white-balance, (c) demosaicking, (d) color correction and (e) gamma correction .

pixel, therefore the conversion from charge to voltage is done at the same time for all pixels.

## 2.2 Digital image processing pipeline

The pipeline used by each camera maker is not made publicly available, in general. However, on the RAW sensor image there are a several main steps that camera makers apply to produce an image ready for display:

### 1. White-balance

This step assures that the image has no color cast. White balance is performed by scaling all intensity values with parameters read from the RAW file, such that neutral colors keep a correct appearance.

Humans are able to perceive a white object, for example, as white even under different illumination conditions that give a cast to that object (for example a bluish light source). This is called color constancy [Brainard et al., 2004] and we would like cameras to be able to mimic this human property. This can be done by approximating the illuminant and undo its influence on the sensor values [Ebner, 2007]. The image Fig.2.3b shows the result of applying white balance on the image Fig.2.3a.

## 2. Demosaicking

The camera sensor produces an image in which for each pixel we only get one of the image channel intensity values (either red, or green or blue), and we need to find an estimate of the other two missing values. This is done by an interpolation process called demosaicking, which produces an image with 3 channels.

One of the simplest demosaicking methods is based on bilinear interpolation: the missing value of a certain color channel is calculated as the average of the four neighbour pixels of the same color channel. This produces a modest image result, with many possible artifacts (false colors or zipper effects). In addition to the pixel neighbourhood information, most methods put emphasis on estimating a correct edge and preserve fine details by using a reconstruction that is done in a directional way. Many algorithms have been proposed for this challenging task [Menon and Calvagno, 2011]. The image Fig.2.3c illustrates the output of applying demosaicking on the image Fig.2.3b.

## 3. Denoising

Noise is unavoidable and, ideally, it should be removed as early as possible in the camera pipeline, to be closer to the noise source. One option is to apply noise removal on the RAW image before demosaicking, another one is to apply it afterwards. It can also be applied in a joint demosaicking and denoising step as in [Hirakawa and Parks, 2006, Buades et al., 2009, Khashabi et al., 2014].

Although the image processing chain that each camera uses is unknown, there are some in-camera denoising techniques widely used, such as correlated double sampling or coring, as described in [Bertalmío, 2014]. The first method consists in sampling a couple of images, one with the shutter closed and the other after exposure, followed by subtracting the latter from the former, thus reducing *dark current* noise. *Coring* is another common in-camera denoising method, which thresholds the DCT coefficients corresponding to high spatial frequency information that is usually associated with noise. All in-camera image processing algorithms must satisfy an essential criteria: a low computational cost, that highly limits the methods. Therefore the compromise between the algorithm complexity and the output image quality is crucial.

In-camera denoising is a challenging task that requires the design of a fast method. On the other hand, current state-of-the-art denoising methods are too complex to be directly implemented on digital camera hardware, without considerable simplifications, as they are based on image patch comparisons. In this thesis, we show that a fast and local denoising method inserted at the right location of the camera image processing pipeline can be a good candidate for in-camera denoising.

#### 4. Color Correction

After demosaicking and denoising, a color correction step is applied, which converts the image from the camera color space to sRGB (standard RGB color space) allowing the image to be displayed on the computer screen, a TV or a projector. In the image Fig.2.3d it is shown the color corrected image Fig.2.3c.

#### 5. Gamma Correction

Historically, this step is related to the cathode ray tube (CRT) displays technology. It exhibited a non linear connection between the luminance observed on screen to the voltage of the display:  $L = \alpha V^\gamma$ , where  $L$  represents the luminance,  $V$  the voltage,  $\gamma$  represents the exponent of the power function with a value around 2.5, and  $\alpha$  is the proportionality coefficient. However, in case of the camera, there is a linear connection between the luminance of the output and the light intensity. In order to display on the CRT the correct signal captured by the camera, one must apply on the camera luminance a power function with the inverse exponent:  $V = \beta L_c^{1/\gamma}$ , where  $L_c$  represents the luminance of the camera and  $\beta$  is the proportionality coefficient [Poynton, 2012]. Fig.2.4 shows the gamma correction curve associated to a CRT display. Also, it was a known fact that human perception is non-linear with respect to the physical luminance, with an exponent around 0.42. Humans have a more acute perception of the differences in the dark, compared to differences in the bright part of an image. As the analog TV signal was noisy, a linear transmission of camera luminance would accentuate noise in the dark part of an image. The solution to make noise less apparent was applying a non-linearity with an exponent around 0.42 on the luminance of the camera.

Coincidentally, the CRT non-linear connection between luminance and

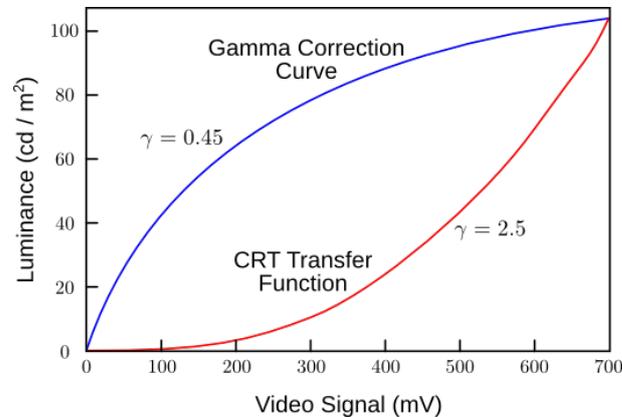


Figure 2.4: The gamma correction curve on a CRT. Figure is from [Bertalmío, 2014].

voltage, is extremely similar to the inverse of the non-linear connection between perceived luminance and physical luminance.

These were the two reasons for performing the gamma correction.

These days, gamma correction is still applied for digital signals, mainly for the so-called perceptual coding. The number of bits used to code the value of the digital signal in each pixel is limited. Therefore, using gamma correction, one can code more bits in the dark and less in the bright, according to the human sensitivity (higher in the dark than in the bright) of perceiving differences in luminance [Bertalmío, 2014]. In the image Fig.2.3e it is illustrated the result of performing gamma correction on the image Fig.2.3d.

## 6. Edge enhancement

Most camera makers apply an edge enhancement step in the camera pipeline for making the final image more pleasant. This is due to the fact that humans are extremely sensitive to sharp edges [Ramanath et al., 2005]: a crisp image with sharp edges is more appealing than a blurred image. Moreover, we are the most sensitive to horizontal and vertical edges, and less for other orientations. One side effect of edge enhancement is that it amplifies also the noise present in the image.

## 7. Compression

Compression is the step in the camera pipeline that makes the signal more compact by reducing the number of bits. It can be lossless (the recovery of the original signal is possible without lost information) or lossy (with the amount of lost information proportional to the compression rate). Ideally, only redundant information should be removed, and the final image quality should not suffer much. An extreme example of lossy compression is shown in Fig.2.5 , where the smooth gradient of the 16-bit depth image is lost in the corresponding 8-bit format.

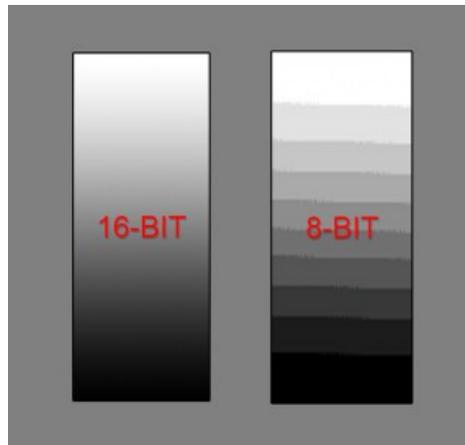


Figure 2.5: Comparison between a 16-bit depth black and white digital image (with up to 65,536 shades of gray) and the lossy compressed 8-bit associated image (with 256 possible values). Figure is from [<https://www.focusphotoschool.com>, 2017].

While some digital cameras are able to record the image in the RAW format (the sensor information) or the TIFF format, which are not lossy, almost all digital cameras produce as final image a JPEG file of 8-bit depth, for which lossy compression has been applied. The way of performing JPEG compression is extremely similar to coring denoising strategy : put to 0 the coefficients of certain high frequencies (that are associated to noise) of the discrete cosine transform (DCT) [Bertalmío, 2014]. More details about the camera processing chain can be found in [Bertalmío, 2014].

## CHAPTER 3

---

### Noise in a digital image

---

Digitizing color images is a process that has the tendency to introduce random variations in the pixel values, caused by imperfections in the optics or electronics. Improving the process of digitizing images is the obvious solution to diminish the effects of noise. Although during the last decades digital camera makers have made huge improvements in this direction, noise removal is still an important task in image processing. Noise in a digital image is caused by many factors, some of which are discussed in this chapter:

1. Pixel and sensor size.

A sensor with a larger pixel captures more photons than a smaller one. The electron capacity is also directly proportional to the sensor area. As a consequence, a larger pixel gives an image with a higher signal-to-noise ratio (SNR) for certain exposure parameters, compared to a smaller one. This also has an impact on the dynamic range.

These days, especially in the case of mobile camera technology, smaller thumbnail-sized sensors are being produced, with either fewer pixels, or smaller pixels, which produce more noise, especially under low illumination conditions.

As we will show in Section 3.1, not the whole sensor area captures light, due to the electronic circuitry. Therefore, another factor that influences noise is the fill factor that represents the storage capacity of well inside a sensor pixel.

## 2. Sensor technology.

The two main types of sensors are CMOS and CCD. Until the late 1990s, the CCD sensor was the preferred one, as it produced images with higher quality compared to the CMOS sensor that tended to introduce more noise. However, the technology of the CMOS sensor highly improved, and these days CMOS sensors are more popular than CCD sensors, due to similar image quality for a lower price and the possibility of added functionality to the sensor chip.

## 3. ISO speed.

The ISO parameter of a digital camera has a huge impact on the noise level: increasing the ISO value means amplifying at each pixel the signal along with noise.

## 4. Digital image processing pipeline.

Usually, sensors have analog-to-digital converters that have a 12-bit depth size. So, losing information and altering the observed signal already begins at the sensor level. This also has an impact on the noise. Moreover, the image processing pipeline also alters the noise, making denoising a challenging task, especially if applied on the output JPEG image.

### **3.1 Noise at the image sensors level**

The scheme presented in Fig.3.1 shows the main noise components at image sensors level, both in dark and illuminated conditions. Dark noise is not present only in the absence of light, as the name might suggest, but also under illumination.

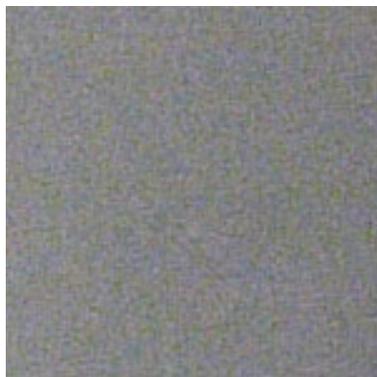
There are two main classes of noise: fixed-pattern noise (FPN), illustrated in Fig.3.2a and temporal noise, exemplified in Fig.3.2b.

As the name suggests, the fixed pattern noise is fixed to certain coordinates of the image array. Temporal noise is random noise that varies in time: it would be different for two consecutive images taken in identical conditions.

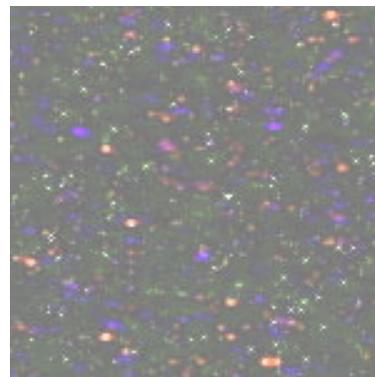
Dark current under no illumination is unwanted current accumulated in the storage well inside a pixel. Its amount is connected to the temperature and the integration time. It causes two main problems. As the size of the well in each pixel is limited, it reduces the amount of wanted current and therefore the dynamic

		Dark	Illuminated	
			Below saturation	Above saturation
Fixed Pattern Noise (FPN)		Dark signal nonuniformity Pixel random Shading	Photo-response nonuniformity Pixel random Shading	
		Dark current nonuniformity (Pixel-wise FPN) (Row-wise FPN) (Column-wise FPN)		
		Defects		
Temporal Noise		Dark current shot noise	Photon shot noise	
		Read noise (Noise floor) Amplifier noise, etc. (Reset noise)		
				Smear, Blooming
Image Lag				

Figure 3.1: Noise at the sensor level. Figure is from [Nakamura, 2005].



(a)



(b)

Figure 3.2: Figure illustrating (a) temporal noise and (b) fixed pattern noise. Figure is from [http://www.cambridgeincolour.com, 2017].

range of the sensor. The second problem is that it alters the value associated to “dark”, under no illumination. In order to estimate a correct black value, the borders of the image array are hidden from light such that the dark current amount can be measured there [Bertalmío, 2014].

One useful way to reduce dark current noise is to perform Correlated Double Sampling (CDS). This means that one can take two photos: one with shutter closed and the other with the sensor exposed to light, and subtract the first one from the second.

Dark current nonuniformity is the main source of FPN in a CCD sensor and is not visible in normal conditions. It appears in extreme cases, when an image is taken with a long exposure or at a high temperature. This fixed-pattern noise component cannot be eliminated by correlated double sampling (CDS). The primary sources of FPN in a CMOS sensor are dark current nonuniformity and slight variations in the output of each amplification transistor located at each pixel [Nakamura, 2005].

Shading is a slightly increasing variation visible in an image. It can be caused by the presence of a local source heat provoking dark currents gradient. Another source is a defective light collection towards the borders of the image array when light enters at an inclined angle. In the CCD image sensor, shading is also caused by changes in amplitude of driving pulses and, therefore, a defective charge transfer mechanism. In the CMOS sensor, a source of shading is also nonuniform biasing and grounding [Nakamura, 2005].

A “defect” pixel represents a pixel for which the incoming light does not cause a response (for example, a white spot defect) or causes an abnormal response compared to normal pixels.

Temporal noise includes three main sources of noise present in both the CCD and CMOS sensors: thermal noise, shot noise, and flicker noise.

Thermal noise is caused by thermal agitation of the electron particles inside a resistance. The sensor transistor acts as a resistance when it is switched ON (during an exposure), causing thermal noise. Moreover, when it is switched OFF, the capacitance node is reset, producing “reset” noise (or kTC noise). Read noise (or noise floor) is thermal noise generated by the readout electronics. Both the output amplifier in a CCD sensor and the amplifier inside each pixel in a CMOS sensor produce read noise.

Shot noise is related to the random creation of electron charge (for dark current

shot noise) and arrival of photons (for photon shot noise) in each pixel in the CCD and CMOS sensors. Both photon incidence and electron emission obey a Poisson distribution; for a low number of particles, shot noise is more visible as it produces a small variation in the image brightness values. Photon shot noise can be reduced by capturing more light. As a solution one could combine multiple frames or have a longer exposure, for example.

While thermal noise and shot noise have power spectral densities that are constant for all frequencies (called “white” noise inspired by the concept of white light), the flicker noise (or  $1/f$  noise, or pink noise) has power spectral density inversely proportional to frequency, due to the material that makes an amplifier. At low frequencies, the output amplifier of the CCD sensors and the amplifier in each pixel of the CMOS sensor produce  $1/f$  noise. With correlated double sampling (CDS) this noise is mostly eliminated [Nakamura, 2005].

Under extreme light illumination, some CCD sensors can produce smear and blooming.

A CCD sensor activity is given by two main steps: the exposure (collection of incoming photons and storage of electrons) and the readout (reading one line at a time). After the exposure, while the pixel values are shifted down the image array, they continue to collect light. If the shifting is slow, it might happen that more light is captured in a particular pixel that shifts a previous charge, thus increasing its charge value. This appears as a vertical line of bright values that can be positioned up and down a light source, as in the left image illustrated in Fig.3.3a, and the effect is called vertical smear.

An image example that exhibits blooming is shown in Fig.3.4a, for a photograph taken with a CCD sensor. Blooming happens for a longer exposure where there is a light source in the scene and the pixel charge makes the bin overflow. Due to the CCD flow down on the image array, the error appears vertically. Smear and blooming can be reduced by introducing a vertical overflow drain (VOD), which has the disadvantage that it decreases the sensor light sensitivity.

In comparison to a CCD image sensor, a CMOS sensor usually does not exhibit smear and blooming as there is no charge shift in its image array.

Some CMOS image sensors can produce the “rolling shutter” effect illustrated in Fig 3.5, under fast motion. This is due to capturing one row after another with a time lag in the range of  $1/50$ th of a second in between; therefore a fast movement can skew the image. This is solved by a CMOS sensor with a “global shutter” that

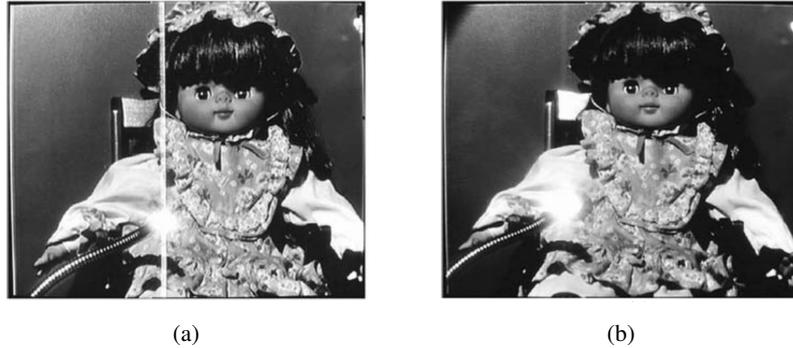


Figure 3.3: Figure illustrating (a) the smear image problem and (b) the improved version using a vertical overflow drain (VOD). Figure is from [Nakamura, 2005].

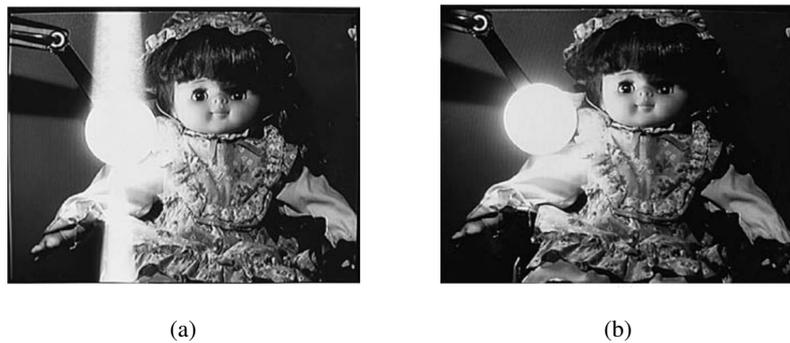


Figure 3.4: Figure illustrating (a) the blooming image problem and (b) the improved version using a vertical overflow drain (VOD). Figure is from [Nakamura, 2005].

makes the exposure at once [Bertalmío, 2014].

Image lag is a charge transfer problem, that can appear in consecutive frames or movies, when a residual charge in a pixel is kept from one photography to another.

Whenever we capture a photography of a scene, some amount of noise is inevitably present in the formed image, even for the highest quality image sensors. An example of a scene with simulated CMOS noise is given by Fig.3.6.



Figure 3.5: An illustration of the rolling shutter effect. Figure is from [Wikipedia, 2017].



Figure 3.6: Simulation of the noise at the CMOS sensor level, with an exaggerated noise level for visualisation purposes. Figure is from [Gow et al., 2007].

### 3.1.1 CCD vs CMOS

Until the late 1990s, CCD sensors were capturing images with significant higher quality than CMOS sensors. This was due to the fact that compared to CCDs, CMOS sensors had a lower light sensitivity requiring greater signal amplification and was more prone to noise. These days, the CMOS sensors are widely used, from high-end digital photography down to mobile phone cameras.

The main difference between a CCD and a CMOS sensor is in the way they read the accumulated charge of each image cell. In a CCD sensor, the charge is

shifted across the image array and read at a corner of the chip, where an analog-to-digital converter transforms each pixel value into a digital one. Most CMOS sensors have several transistors in each pixel that amplify and transport the charge with traditional wires. This offers flexibility to the CMOS devices, as each pixel can be read individually.

The CCDs employ a special manufacturing process for shifting the charge across the array without distortion, leading to high quality sensors in terms of fidelity, uniformity and light sensitivity. Comparatively, CMOS sensors use a traditional manufacturing processes to create the chip, same as for most microprocessors. The different manufacturing techniques introduce noticeable differences between the two sensors.

For two equivalent sensors, the CCD one uses a process that consumes up to 100 times more power than the CMOS one [Nilsson, 2017].

One main problem related to the CMOS sensor is that it has an amplifier for each pixel, which means having up to millions of amplifiers on the chip, each leading to an inevitable small variation in the output. This produces a noise pattern in the image that can be eliminated with correlated double sampling (CDS). When taking a digital photography, two exposures are being performed: one of the scene and the other, with the shutter closed, of the noise pattern. Subtracting the latter from the former outputs an image with less noise.

Another main problem of the CMOS unit is also related to its amplifiers: while the CCD array is almost fully occupied by photo sensor, the CMOS one has less space for photo sensors due to the amplifier circuitry. Therefore, part of the light captured during exposure is lost if it reaches the area between the photo sensors. To solve this problem, CMOSs use millions of microlenses that cover each photo sensor and its amplifier circuitry, such that all light is focused on the photo sensors (including light rays that would have fallen in between). Fig.3.7 shows a schematic representation of a microlens. This leads to an increased light sensitivity.

A huge advantage of the CMOS chip over the CCD one is given by the use of analog-to-digital converters that have been integrated onto the same chip, making it possible for a CMOS sensor to do digital signal and image processing on chip.

These days, it is rare to find a current-generation camera with a CCD sensor, most of them using the CMOS technology. If it happens to have a CCD chip, it usually belongs at the very high end of the premium market.

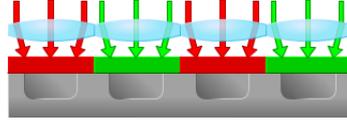


Figure 3.7: An illustration of the microlenses used on the CMOS chip, such that all light rays are focused onto the photo sensor. Figure is from [<http://cpn.canon.europe.com>, 2017].

### 3.2 Optimal choice of the exposure triangle: aperture, ISO and shutter speed

When the shutter button is not pressed, the shutter is closed such that no light reaches the sensor. For taking a photograph, we press the button and the shutter opens to allow the light to pass in: the longer the shutter is open, more light falls on the image sensor. This is controlled by the shutter speed. The notion of “exposure time” is related to the shutter speed: the fastest the shutter speed, the shorter the exposure time. Another control on the amount of light that reaches the sensor is given by the f-number (the aperture value). The lens has an aperture (an opening) which is a diaphragm formed by overlapping blades, as illustrated in Fig.3.8. A small f-number (for example  $f/2.8$ ) gives a wide aperture. The ISO



Figure 3.8: Figure illustrating the aperture blades. Figure is from [[www.flickr.com](http://www.flickr.com), 2017].

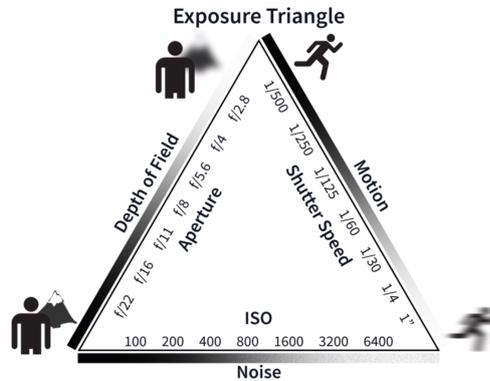


Figure 3.9: Figure illustrating the effect of different camera parameters. Figure is from [http://www.imatest.com, 2017].

speed (or ISO sensitivity) estimates the camera sensitivity to light. The usual values are given by numbers like ISO 100, ISO 200 and much higher; the higher the value the higher the sensitivity. The camera transforms the light captured by the image sensor into 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 (or from ISO 200 to ISO 400) the electrical signal is doubled and the camera sensor needs half the amount of light. However, amplifying the electrical signal means a better preservation of details in the dark area but it amplifies also noise. Therefore, higher ISO speed brings also more noise. Before taking a photograph of a scene, it is essential to choose these three camera parameters wisely: shutter speed, aperture and ISO. Fig.3.9 illustrates the "exposure triangle" and the main challenges for getting the correct exposure. Each camera parameter produces different effects on the image result: aperture is related to the depth of field, shutter speed to motion blur and ISO to image noise. The depth of field is the range of distance for which objects have a sharp and well-focused appearance. In a digital image, for a shallow depth of field (wide aperture/low f-number) the transition from sharp to blur is gradual, as exemplified by Fig.3.10a. An example of a photograph taken with a narrow aperture is illustrated in Fig.3.10b. By changing the shutter speed one can manipulate motion appearance. Fig.3.11a shows how a slow shutter speed



Figure 3.10: Figure illustrating images taken with a wide aperture (a) and a narrow one (b). Figure is from [<http://www.cambridgeincolour.com>, 2017].

can give a silky effect to the waterfall. Oppositely, with a high shutter speed one can freeze the movement of a close subject as in Fig.3.11b. While these motion blur examples are chosen for the artistic effect, in general photographers opt for the shutter speed that gives a sharp image result. In the case of the ISO speed, a lower value is preferred, since higher ISO increases the amount of image noise. In practice, the ISO sensitivity is increased when one cannot obtain the wanted image result by changing the aperture and shutter speed. Increasing the ISO speed means less light is necessary for taking a photograph and allows choosing a higher shutter speed and/or a smaller aperture. Fig.3.12a shows an example of the noise appearance for a small ISO value, while Fig.3.12b a high one.

A long exposure can produce fixed pattern noise, as exemplified by 3.2b.

In order to capture a photograph that minimizes the amount of noise introduced, these are the main steps that Marc Levoy recommends to choose as optimal exposure parameters in his lectures on Digital Photography [<https://sites.google.com/site/marclevoylectures/>, 2016]:

1. Chose the aperture associated to the wanted depth of field.
2. Increase the exposure time (decrease shutter speed) as long as there is no camera or motion blur.
3. Increase ISO such that the brightest object in the scene just reaches white

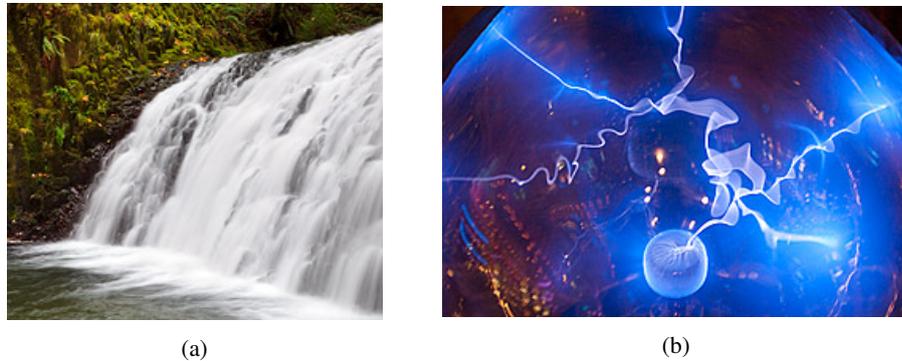


Figure 3.11: Figure illustrating images taken with a slow shutter speed (a) and a fast one (b). Figure is from [<http://www.cambridgeincolour.com>, 2017].

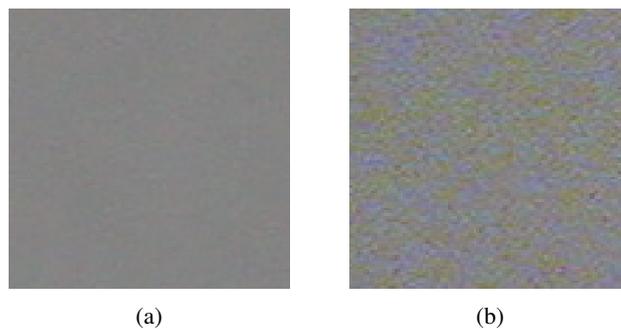


Figure 3.12: Figure illustrating images taken with a low ISO value (a) and a high one (b). Figure is from [<http://www.cambridgeincolour.com>, 2017].

on the histogram, as one does not want saturated bright objects.

### 3.3 Noise and the dynamic range

In photography, dynamic range measures the ratio between the highest and lowest measurable light intensity.

The notion of the dynamic range of the camera image sensor is connected

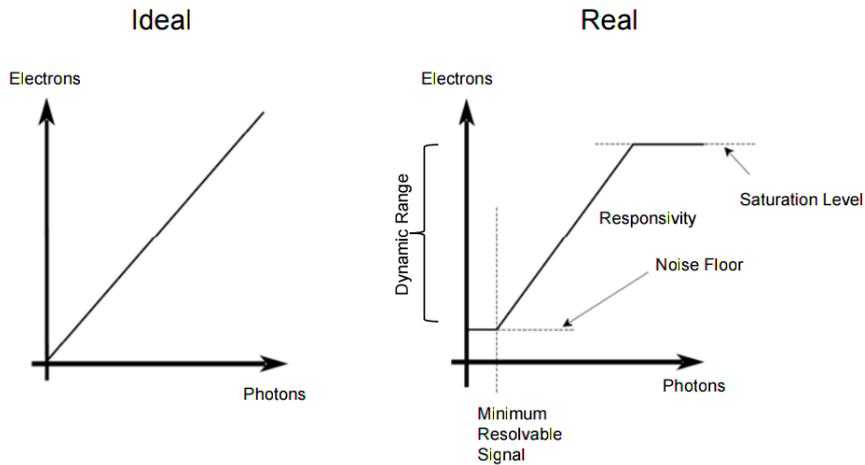


Figure 3.13: Figure comparing a pixel cell from an ideal image sensor to a real one with dynamic range limitations. Figure made by Alexander Hilgarth.

to the peak signal to noise ratio (PSNR), and is given by the ratio between the maximum and minimum measurable luminance:

$$PSNR = 20 \log_{10} \left( \frac{L_{peak}}{L_{noise}} \right). \quad (3.1)$$

The peak value is directly proportional to the full well capacity (the maximum number of electrons per pixel). It is desirable for the full well capacity to be as large as possible. Also, the photon shot noise limits the dynamic range, and one of the solutions is, again, a large well capacity. The 12-bit or 10-bit conversion from analog to digital signal employed by most consumer cameras also influences the peak value, and therefore the dynamic range. The minimum measurable value is related to the noise floor associated to low signal (black). The noise floor contains sensor readout noise, noise related to the analog-to-digital conversion and amplifier noise. However, for long exposures, the noise floor is given by thermal noise. More details regarding sensor noise components are introduced in Section 3.1.

For increasing the dynamic range of image sensor one should decrease the noise floor or/and increase the full well capacity.

A high dynamic range scene, that shows a wide range of details from bright to dark, can surpass 100dB. This is a challenge for conventional cameras that have around 70dB (approximately 3.5 orders of magnitude). Therefore, in a high dynamic scene as the example from Fig.3.14b, the bright regions or the dark ones, or both, must be clipped (according to the exposure parameters). In general, photographers prefer to under-expose a scene that to over-expose it, as recovering highlights is a much more challenging task than recovering shadows.

High Dynamic Range (HDR) imaging can solve this challenge by combining in one photograph multiple exposures taken with common cameras or using a special HDR image sensor.



Figure 3.14: Figure with images of a goose taken in low dynamic range (a) and high dynamic range (b). Figure is from [<https://digital-photography school.com>, 2017].

### 3.4 Noise perception

All the decades of film photography created the expectation that some grain noise is always present in photographs, and, when subtle, people actually like it [<https://photography.tutsplus.com>, 2017]. One uses the term grain due to the fact that the film actually contains small grains of silver. Grain noise has a uniform appearance with a surprising positive consequence due to the way our brain handles vision: small amount of it gives the appearance of higher resolution and sharpness, thus leading to a more likeable photograph. Fig.3.15 is an example



Figure 3.15: An example of a photograph with grain. Image by Tyler Allen on [www.flickr.com, 2017].

of a photograph containing grain. However, too much noise will make the image unpleasant. Also, noise that is not uniform, but highly localised, leads to an unlikeable image. And this is the case of digital image noise, that is not uniform but image-dependent, and it appears more pronounced in the dark areas and shadows, as seen in Fig.3.16. Fig.3.17 illustrates an image example in which



Figure 3.16: Figure illustrating how darker regions appear more noisy than brighter ones in a digital image. Figure is from [http://www.cambridgeincolour.com, 2017].



Figure 3.17: Figure illustrating how a certain amount of noise of a certain type can improve the appearance of a digital photograph, giving a sharper look to it. Figure is from [<https://photography.tutsplus.com>, 2017].

noise has a positive connotation. The left image is the original photograph, while the one on the right contains two layers of Gaussian noise: one to hide shadow noise (actual digital noise) and increase the sharpness and the other to give the appearance of grain noise.

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 [<https://photography.tutsplus.com>, 2017]. 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 monochromatic. 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. Figure 3.18 illustrates how different luminance and chroma noise

appear.

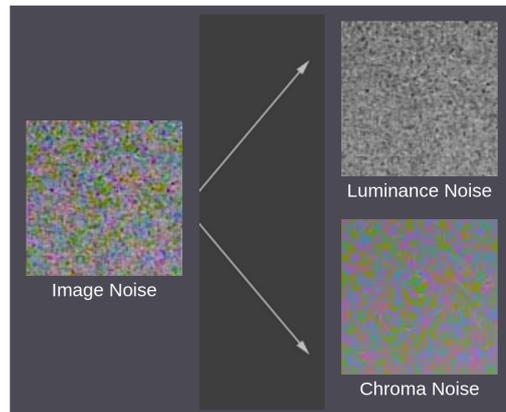


Figure 3.18: Figure illustrating the different appearance of luminance noise and chroma noise in a digital image. Figure is from [<http://www.cambridgeincolour.com>, 2017].

In the work with the title “Sharpness rules” [Johnson and Fairchild, 2000], Johnson and Fairchild examined some of the variables that influence image sharpness perception: resolution, noise (additive uniform noise applied independently in R, G and B color channels), contrast (increased by sigmoidal exponential shaping functions applied on R, G and B channels independently) and sharpening (using the Photoshop sharpen filter). They used original images of adequate quality for color printing with a 300 pixel-per-inch (ppi) resolution, and they decreased the resolution obtaining test image of 150 ppi and 100 ppi. Experiments of paired-comparison with observers concluded that the most important factor dictating perceived sharpness is the image resolution. However, the highest 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 R, G and B color channels) does increase the perceived sharpness, but only up to a point, after which it decreases. Interestingly, adding noise can also mask a reduction in image resolution: the 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.

Another work on estimating the increase of sharpness by image noise was done by Kurihara et al. [Kurihara et al., 2009]. Their result was that when noise is added to edges, sharpness decreases, while when added to texture, it increases up to a noise level with standard deviation of 15, as illustrated in Fig.3.19.

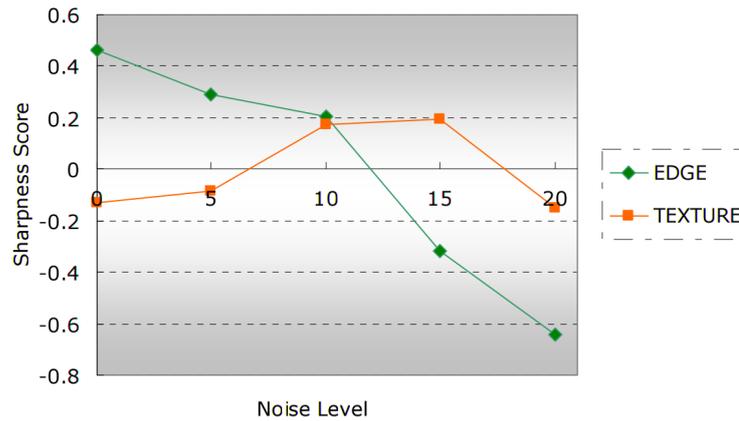


Figure 3.19: Figure illustrating how for edges, the perceived sharpness decreases with increasing added noise, while for texture it increases up to a certain point. Figure is from [Kurihara et al., 2009].

Kayargadde and Martens investigated in [Kayargadde and Martens, 1996] the connection between the noise level and blur. Their conclusion was that adding noise to a sharp image makes it to be perceived as more blurred, while adding noise to a blurred one makes it appear a little less blurred.

Fairchild and Johnson made another study involving noise perception in [Fairchild and Johnson, 2005]. The origin of this work is the claim of Webster [Webster, 2003]: “adaptation increases the salience of novel stimuli by partially discounting the ambient background”. Fairchild and Johnson used this idea to make psychophysical experiments on noise adaptation in color image perception. Fig.3.20 illustrates the idea of adaptation to image noise. Try gazing for 15 to 30 seconds at the bar between the top images; afterwards, for the bottom images, the left one looks noisier than the right one, although they are identical. The authors claim that the human visual system (HVS) might be able to adapt to noise present in photographs by “enhancing the perception of image content while minimizing the perception of artifacts introduced by imaging systems”. Essentially, their

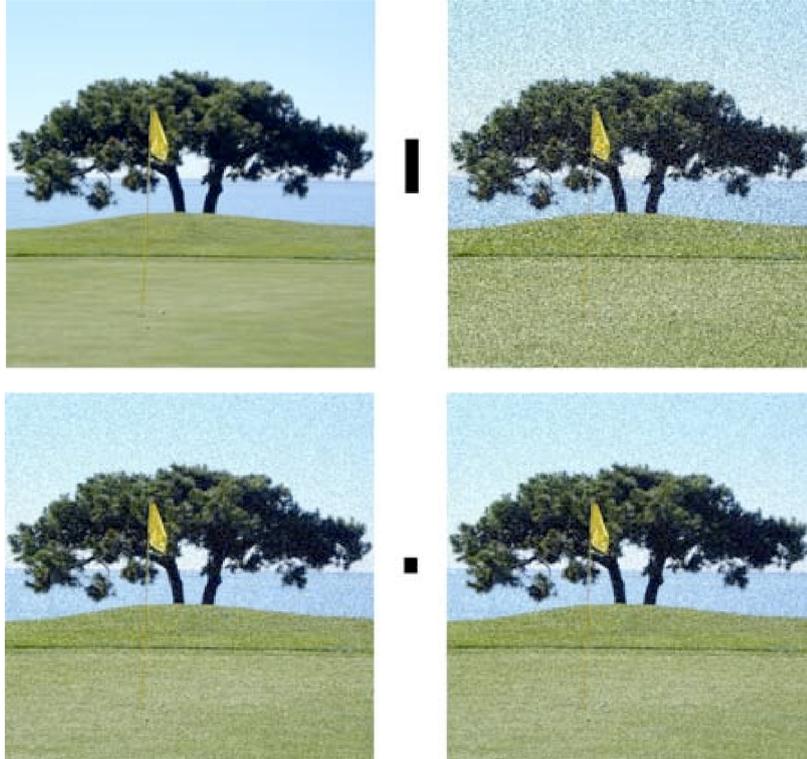


Figure 3.20: Figure illustrating the concept of adaptation to image noise. Figure is from [Fairchild and Johnson, 2005].

work was to investigate how the image content is perceived after adaptation to noise. The assumption was that noise adaptation causes a decrease in sensitivity (increases the threshold) to similar noise present in an image. As noise they use three spatially-structured types of different contrast intensities, illustrated in Fig.3.21, and they added it on several clean images and on the background. An example of the test window shown to observers is shown in Fig.3.22. The images were shown for 1 second and afterwards only the background was shown for 4 seconds. This actions repeated and subjects were asked to tune the noise contrast of the image in the right such that the noise level is large enough just to differentiate which of type it is (out of the three). Finally, it was measured the

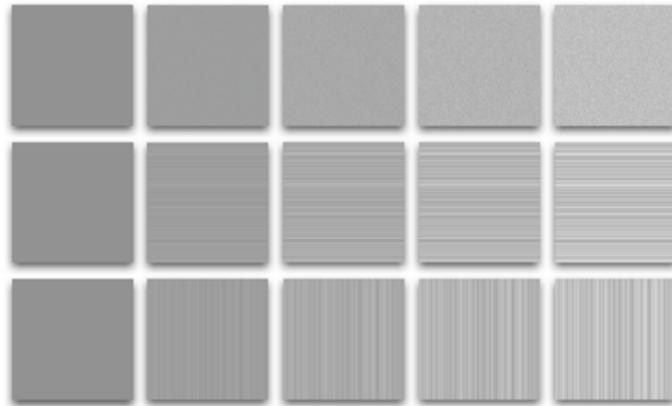


Figure 3.21: Figure illustrating three types of noise used for the psychophysical experiments regarding adaptation to image noise. For each type, from left to right, the contrast increases up to 37.5%. Figure is from [Fairchild and Johnson, 2005].

visibility of each noise type as a function of the adapting background contrast averaged for each adapting condition and over all images. The noise of each one of the three types has no (or little) effect upon how sensitive the HVS is to another type of noise. A general conclusion is that noise visibility is a function of the image content. The authors justify this with the phenomenon of noise “masking and adaptation to the spatial frequency content of the image itself”. The random noise is noticeable the most on images with homogeneous areas, and the least on images with many details and texture (high frequency information). However, in any case, it was confirmed that noise adaptation decreases the sensitivity to similar noise contained by the image, and this phenomenon is more accentuated for random noise than for vertical or horizontal one. As a consequence, the authors affirm that imaging systems can afford to exhibit slightly more artifacts like random noise or compression artifacts, for example, as the HVS masks some of it.

Shohara and Kotani investigate in [Shohara and Kazunori, 2010] the perception of color noise, and how it is influenced by the background, the colour itself and luminance. They find that color noise is more perceived when the average colour is blue, or when the average luminance has a value around 35; on the contrary, it is harder to perceive noise in green.

Interestingly, in [Kurihara et al., 2008] the scientists meet a professional



Figure 3.22: Figure illustrating the stimulus including the original image on the left and the a version of it on the right, while the grey background also has added noise. Figure is from [Fairchild and Johnson, 2005].

photographer, Kunitoshi Yabe, to investigate how a digital image can be improved by adding noise, from an aesthetic point of view. The authors affirm that the way professional photographers model the noise should have an impact on how to handle real noise in digital images. Fig.3.23 shows an example of how the noise looks like. By adding the noise, the banding and real actual noise of the image on the left are masked by the noise. The photographer also argues that noise can improve the appearance of blown highlights and the perceived resolution. Also, small images (especially), that show JPEG artifacts or blur, can benefit from this technique.

The researchers reproduced the way the photographer added the noise. First of all, the images were divided into 3 main categories: landscapes, portraits and still life. Each category was divided into 3 more, according to the average grey value: low-key (mean value less than 85), middle-key (mean value between 85 and 170) and high-key (mean value larger than 170). The photographer introduced several different noise layers for each image example, each layer with a different treatment for each category and for each type of detail (shadow, highlight or

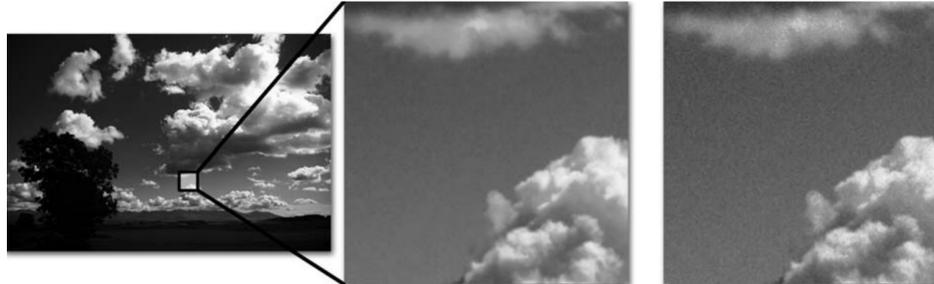


Figure 3.23: Figure illustrating three types of noise used for the psychophysical experiments regarding adaptation to image noise. For each type, from left to right, the contrast increases up to 37.5%. Figure is from [Kurihara et al., 2008].

midtone). For each of the final nine categories is created a plot of the amount noise depending on the pixel value. There is no flat plot; the amount of added noise depends on each pixel intensity value (the noise is image-dependent). A plot example is shown in the right image from Fig.3.24. All plots exhibit a higher noise level in midtones and a lower one in shadows and highlights. Actually, there is almost no noise towards 0 and 255. The photographer admitted to have chosen noise whose appearance emulates somehow the grain in black and white film photographs. In order to estimate the amount of noise as a function of mean pixel

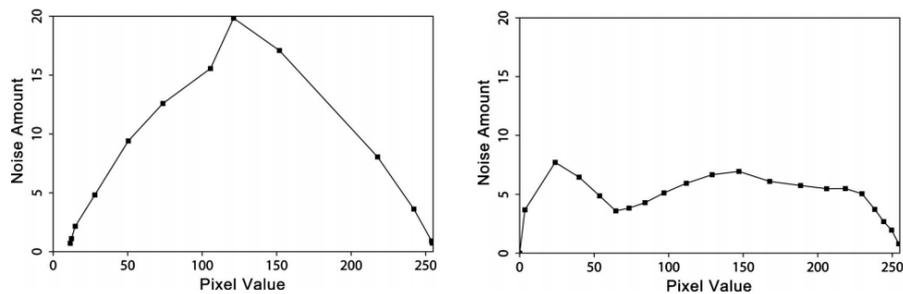


Figure 3.24: Figure illustrating how the amount of analog black and white print noise changes with pixel values (left) and one example of the amount of noise added by a professional photographer to a digital image to improve its appearance. Figure is from [Kurihara et al., 2008].

value for an analog print, the authors took 12 film photographs of a grey cardboard

with different exposure parameters, printed them on paper and, finally, scanned them. Fig.3.24 shows on the left a plot of the amount of noise as a function of the mean pixel value area, extracted from the 12 black and white analog photographs, while on the right plots the amount of noise as a function on pixel value, for the digital noise modeled by the photographer (aesthetically preferred noise). An important finding, obtained when analyzing the image histograms (for both analog prints and digital images), was that, 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). The visual difference between the two noise distribution can be seen in the example from Fig.3.25.

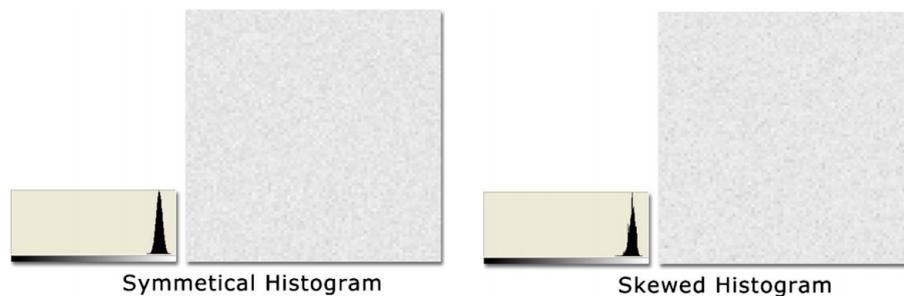


Figure 3.25: Figure illustrating the visual different appearance of noise created with a Gaussian distribution (left) and chi-squared distribution (right), and their associated histograms, for the same mean value of 230 and standard deviation of 5.7. The left image seems to have more dark pixels. Figure is from [Kurihara et al., 2008].

The photographer Thomas Ruff elevated JPEG artifacts at the art level. In his coffee table book called JPEGs [Ruff and Simpson, 2009], he shows a collection of images that contain JPEG artifacts, arguing that “a pixellated square is ugly, but if you present it in the right context it can become beautiful”. An example of the search of this artist for beauty within image artifacts is illustrated by Fig.3.26. Regarding his JPEG experiments he affirms: “I found that when you blow them up to about 2.5 metres by 1.8 metres, it creates a nice effect: when you see it from about 10 or 15 metres away, you think you are looking at a precise photograph, but if you go closer, to within about five metres, you recognise the image for what it is. Then if you go really close, you can’t recognise anything at all: you’re just



Figure 3.26: Figure illustrating a piece of art created by Thomas Ruff, in his search for beauty in JPEGs. [Ruff and Simpson, 2009].

standing in front of thousands of pixels.”

Image noise and artifacts end up in another coffee table book with the title “Glitch”, where Menkman [Menkman, 2011] describes the contemporary art current that explore the aesthetics of failure. The term glitch is connected to a problem, malfunction or a bug; it is something that went wrong. The glitch art has its roots in noise art. The authors affirms: “While not being new, noise art arises unpredictably in new forms across different technologies and cultural scenes. Over time, noise artists have migrated from exploring the grain, the scratching and burning of celluloid (for example, a colour box by Len Lye, 1937) to the magnetic distortion and scanning lines of the cathode ray tube (a significant work being Nam June Paik in magnetTV in 1965).” “In the end, the glitch is a subjective phenomenon. There is no unequivocal cultural definition of glitch, as there is none for noise, because in the end, what glitch is and what glitch is not is a subjective matter”. An example of glitch art is given in Fig.3.27.

Menkman and Larsby even created a glitch software, called Monglot, with the purpose of studying the aesthetics of different digital image artifacts. The

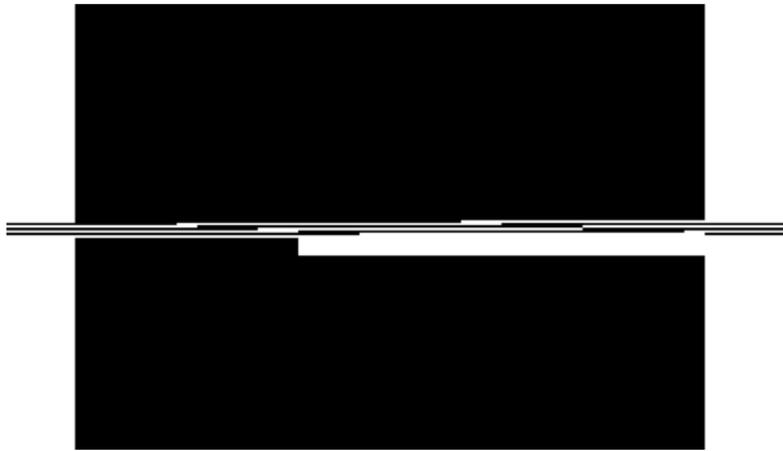


Figure 3.27: Figure illustrating a glitch piece of art created in 2002 by Ant Scott, with the title “Digital screenshot”. Figure is from [Menkman, 2011].



Figure 3.28: Figure illustrating the Monglot Glitch Software Interface created in 2011 by Menkman and Larsby. Figure is from [Menkman, 2011].

interface of this software is illustrated in Fig.3.28.

In the field of computer graphics, photorealism is an important goal when dealing with computer generated images (for example, rendering or introducing computer generated object into real scenes). Sophisticated techniques are employed in order to mimic reality. Geigel and Musgrave proposed in [Geigel and Musgrave, 1997] an algorithm that imitates the process of creating black and white print, and use it for rendered images for a more realistic look. Their algorithm involves also adding some amount of grain-like noise, and this step is quite essential for their purpose. Fig.3.29 shows an image example when their method is used on a rendered image.

In computer graphics, noise is widely used from rendering in movie production to video games, due to the fact that with a low cost method one gets the appearance of having more details and realism. An interesting study of the state of the art in procedural noise functions is created by Lagae et al. [Lagae et al., 2010]. The first and one of the most popular model is Perlin noise, used to create the marble vase image Fig.3.30. A usual technique for texture synthesis is to process noise patterns to produce useful textures for rendered objects; and usually one starts with Gaussian noise. When applying texture or noise to objects, one needs to model it to the topology of the object: one maps the noise or texture to the surface.

Fig.3.31 illustrates the appearance of different noise models used in computer graphics.

Another context in which noise has a positive connotation, is for stereoscopic

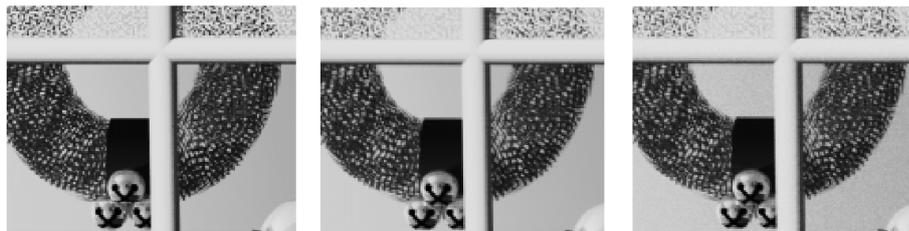


Figure 3.29: Figure of a computer generated image example (left), processed to mimic the photographic film development (middle) to which noise is added (right) for a more realistic appearance. Figure is from [Geigel and Musgrave, 1997].



Figure 3.30: Figure illustrating Perlin noise (left) and its appearance when added to a rendered image for giving visually complex details. Figure is from [Perlin, 1985].

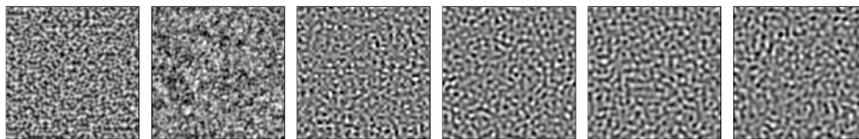


Figure 3.31: Figure illustrating the appearance of different popular procedural noise functions used in computer graphics. From left to right: Perlin noise, sparse convolution noise, wavelet noise, anisotropic noise, better gradient noise and Gabor noise. Figure is from [Lagae et al., 2010].

video creation, where the illusion of depth (3D) is given by using two slightly different images, one for each eye. Templin et al. [Templin et al., 2014] propose a technique to add grain noise to stereoscopic film such that it has a natural look in the 2D to 3D conversion. While the simplest way for doing this is to add the same grain for both the left eye image and the right eye one, this might cause problems such as the double vision effect. If uncorrelated noise is added for the left and right images, simulating the use of two different cameras in reality, this might cause visual discomfort (as it can lead to unmatched details); also, only a small

amount of uncorrelated noise can be accepted by the HVS. Another solution is to project the grain noise onto the objects in the scene; it has the disadvantages that noise might be mistaken for the texture of the objects and depth imperfections might be accentuated. To overcome all these problems, the authors propose a new perceptually-motivated technique that scatters the grain into the scene, by working with multiple layers.

## CHAPTER 4

---

### Related work in image denoising

---

Digital image denoising has the objective to detect and remove the unwanted noise component from a digital image. The challenge is to distinguish between genuine information and noise sources. It is assumed that in an image there are smooth transitions in its intensity values and colors, therefore, values that vary significantly in the neighbourhood are associated to noise. The problem is that some of those values are fine details or texture that one should preserve in the denoised image, otherwise the output can have a blurred appearance. Most denoising techniques have parameters adjusted for the trade off between blurring and the noise level.

#### 4.1 Classic denoising methods

The local denoising approach is the first important category of proposed denoising methods, that estimate each pixel value based on the values of their neighbors.

Early denoising methods separate noise from wanted image information based on the idea that natural images are band-limited while noise is not, see [Lebrun et al., 2012]. From this point diverged two main directions: the transform domain filtering methods that treat the high-frequency components of the transformed image as noise, and the adaptive averaging methods which replace a pixel value with an average of its neighbors.

The transform domain filtering methods originate with the study of orthogonal

function basis by Haar [Haar, 1910] in 1910. The Wiener filter [Wiener, 1949] is the first filter with a statistical approach, and it requires information on the spectral properties of the noise and the image. In general, the transform domain methods transform the image into the frequency domain, where different filters are applied to every transform coefficient. Weaver et al. [Weaver et al., 1991](for medical imaging) and Donoho and Johnstone [Donoho, 1993], [Donoho and Johnstone, 1998] proposed to set the transform coefficients to 0 or diminish them, for high frequency, in the attempt to remove noise. The denoised image is created as the reconstructed image with modified coefficients in the frequency domain, by applying the inverse transform. As examples of frequency domain methods, the most important are the Fourier transform, the Discrete Cosine Transform and the Wavelet Transform.

The Fourier Transform is global and it has the disadvantage that local features might not be preserved. On the other hand, using the Discrete Cosine transform (Yaroslavsky [Yaroslavsky, 1996]) and the Wavelet Transform (Donoho [Donoho, 1993], [Donoho and Johnstone, 1998] ) give a local approach, computed in a moving window frame, and the details are better preserved. While the Fourier Transform contains information only in the frequency domain, the wavelet method offers information in the frequency domain but also in the spatial domain. An advantage of the wavelet methods is a high PSNR results, while a disadvantage is that there might be certain artifacts, like ringing, that are being introduced.

Spatial domain filtering methods include traditional methods in which denoising is done with spacial filters. The median filter uses a window centered in each pixel of the image, and replaces a pixel value with the median value chosen from a sorted sequence of the neighbor pixels inside the window.

An important subcategory of image denoising methods is one that uses diffusion as main ingredient, expressed in a variational approach with PDEs. The most simple one is the isotropic diffusion , which gives a blurred denoised image, as it does not differentiate between noise and edge. Perona and Malik [Perona and Malik, 1990] proposed an anisotropic diffusion method that inhibits the diffusion across the edges. Thus, edges are kept, but also edge noise is kept. Weickert [Weickert, 1996] solves this problem by creating a model that applies a diffusion along the edges and inhibits the diffusion across the edges. Another important diffusion-based method is TV (Rudin, Osher, Fatemi) [Rudin et al., 1992] that outputs sharp edges denoised image. In the direction of TV there is a continuation, the Bregman iterations approach of Osher et al. [Osher et al., 2005]. Lysaker et

al. [Lysaker et al., 2004a] proposed to remove the noise by smoothing the normals of the level curves of the image.

All these diffusion based methods, actually use information from the neighborhood of a pixel. Similar methods that use this also, are average based methods like the bilateral filter of Tomasi and Manduchi [Tomasi and Manduchi, 1998] and Yaroslavsky Neighborhood Filter [Yaroslavsky, 1985]. Here, for each pixel, one takes as grey value an weighted average of the pixels close to it spatially and with a similar intensity value. For defining the weights, the bilateral filter employs Gaussian functions with the Euclidian distance.

Geman and Geman propose in [Geman and Geman, 1984] a statistical denoising method modeled with Markov random fields (MRF). The idea that the probability of pixel to have a certain intensity value depends on its neighbors intensities (spacial dependency or predictability of image intensities) justify the use of the Markov property for images. The algorithm changes a pixel intensity value according to the Gibbs sampling: based on the known conditional distribution of the pixel given the neighbors intensities, it generates a Markov chain of pixel intensity values which converges to denoised image. The conditional probabilities have the same functionality as the image energy for the variational approach. For example, in the MRF image model, one can model a better preservation of the edge intensities by using a hidden parameter.

## 4.2 Non-local methods

While local methods are able to keep the main geometry of the image after denoising, a more difficult task is to preserve small details; it is a challenge to distinguish between noise and fine details by just looking at the neighbor pixel values. However, if the neighborhood increases in size, for example by considering a window centred around each pixel, and looking for similar windows in the image, the chances of preserving fine details after removing the noise increase drastically. This search is justified by the self-similarity principle obeyed by natural images: an image contains image patches similar between each other. For image processing, this observation was implemented for the first time by Efros and Leung for texture synthesis [Efros and Leung, 1999], while for image denoising by Buades et al. [Buades et al., 2005a] that proposed a method called Non-local means (NLM). The NLM algorithm, together with the approach of Awate and



Figure 4.1: Figure illustrating the main idea of NLM denoising: in order to denoise the noisy patch  $p$ , similar patches are contributing with different weights directly proportional to their similarity to  $p$  (higher for patches  $q_1$  and  $q_2$ , and lower for  $q_3$ ). Figure is from [Buades et al., 2005a].

Whitaker [Awate and Whitaker, 2005] that was proposed simultaneously in 2005, gave much better results than previous methods, and opened a new trend in image denoising: the use of patch-based methods. These methods have the advantage of introducing fewer artifacts than the wavelet methods.

The NLM algorithm [Buades et al., 2005a] was designed such that the grey value for each pixel in the denoised image is computed as a normalized weighted sum of similar pixel values, chosen as central pixel for which the gaussian neighborhood is similar, searched for in a quadratic window. The dimension of the searching window is directly proportional to the standard deviation of the noise ( $\sigma$ ). The authors use the euclidean distance as a similarity measure for finding similar patches.

Fig.4.1 illustrates several choices of similar patches and gives a justification of the name (non-local).

The patch-based method of Awate and Whitaker [Awate and Whitaker, 2005], called unsupervised information-theoretic adaptive filter (UINTA), denoises pixels

by using the information of other pixels whose neighborhood is similar, similarly to the NLM method. It minimizes the non-linear entropy function of patterns of intensities in image patches, by using a gradient descent. The authors justify the choice of entropy with the observation that adding any type of noise to an image increases the joint entropy. This is due to the fact that the probability density function (PDF) of the sum of two random variables is given by the convolution of their PDF, which increases the entropy. Clean images have a low entropy compared to their equivalent noisy versions. Consequently, the denoising task implies a decrease of entropy. The UINTA method denoises pixels by decreasing the randomness of the computed conditional PDFs in the original image.

When NLM was proposed, the method of Portilla et al. [Portilla et al., 2003] was considered to be the state of the art in image denoising. Although it outperformed NLM in terms of PSNR, as mentioned in [Lebrun et al., 2012], visually, it produced more artifacts. Portilla et al. propose to decompose a noisy image into subbands at multiple scales by using a pyramid approach, apply wavelet-based denoising on each and reconstruct the image from the filtered subbands. To apply the wavelet-based denoising, they develop a model for the image patches of oriented pyramid coefficients by using a Gaussian scale mixture (GSM) defined as the product of a Gaussian random vector and an independent positive random scalar. Finally, they apply denoising as a Bayesian least squares (BLS) estimator, considering the noise as additive white Gaussian with known variance.

Dabov et al. [Dabov et al., 2007] extend the idea of NLM with the method BM3D(Block Matching 3D), which is still today one of the top performing denoising methods. BM3D uses both the spatial domain and the frequency domain: for each pixel in the image the authors take a patch around it and create a 3D block of patches similar to this patch. As the block has layers of similar information, the transform of the block has sparse coefficients so the noise can be well eliminated by shrinkage. The finest details that all patches share are well preserved. This produces a basic estimate of the denoised image. Using the noisy image and the estimated denoised image, in a second step a 3D Wiener filter is applied, which produces a denoised image of great quality. Levin and Nadler [Levin and Nadler, 2011a] show that BM3D is close to optimality.

In the context of the study of sparse representation of signals, Elad and Aharon [Elad and Aharon, 2006] [Aharon et al., 2006] propose a denoising algorithm that uses a trained dictionary that contains signal-atoms, and describe signals as sparse linear combinations of these atoms. For a certain training set of

signals, the authors seek the dictionary (containing natural images but also patches of the noisy image itself) that gives the best representation for each member of the training set with the sparsity constraint. A generalization of the K-means algorithm is used, called K-SVD, that alternates between sparse coding of the current dictionary atoms and an update of the dictionary to fit better the data.

However, dictionary learning is computationally extremely complex. With the advantages of sparsity representation and avoiding the high complexity, Yu et al. use Gaussian Mixture Models (GMM) for image restoration [Yu et al., 2012]. Their idea is that natural images have a global complex structure, but a locally simple one. Therefore, one can simply decompose locally an image into overlapping local patches, where the patches are seen as mixtures of Gaussian distributions.

Mosseri et al. continued the line of patch-based methods that exploit the concept of self-similarity, and classify denoising methods into two types: internal denoising (denoising an image patch with patches that belong to the noisy image) and external denoising (denoising an image patch using patches from clean natural images different than the input image), see [Mosseri et al., 2013]. Regarding internal denoising, Zontak and Irani [Zontak and Irani, 2011] argue that internal image-specific statistics might be more powerful than external image statistics (from a large database of natural images), as each patch will almost for sure reappear in a given noisy image, while it might not appear in another image. The authors show in see [Mosseri et al., 2013] that combining the power of internal and external denoising, for a certain noisy image, gives even better results: patches with a higher PSNR (edges, details) benefit from external denoising while a low PSNR patch (uniform noisy region) benefits from internal denoising. Also, the higher the noise level in an image the bigger the benefits from internal denoising. Another improvement is done in [Zontak et al., 2013] using the observation that each noisy patch has a cleaner version of itself present at the same relative coordinates but in a coarser scale of the image. The higher the noise level, the stronger this observation holds. The patch recurrence across scales is exploited with the help of directional pyramids (using blur and subsample in one direction).

Levin and Nadler [Levin and Nadler, 2011a] and Chatterjee and Milanfar [Chatterjee and Milanfar, 2010] investigate how much more space for improvement there is left for image denoising. One of the articles [Chatterjee and Milanfar, 2010] has the expressive title: Is denoising dead?. The answer is that for natural image, patch-based denoising methods are close to the optimum bound-

ary. However, for low and high frequency noise levels, there is still place for improvement.

Roth and Black argue in [Roth and Black, 2009] that when one wants to model noise or uncertainty in image processing, prior models of images are always a good solution. They propose to learn Markov random field (MRF) image prior, inspired by sparse image coding. The proposed Field of Experts (FoE) model is applied for image denoising and inpainting. The image results of the FOE denoising method are comparable to those of Portilla et al. [Portilla et al., 2003].

Mairal et al. propose in [Mairal et al., 2010] joint denoising and demosaicking in a framework, by combining two trends that proved to perform well: exploiting the self-similarity of natural images and learning a dictionary (represented by a basis set) created for sparse signal descriptors (sparse coding). The authors propose to combine the two trends, using the idea that similar patches can be expressed as a linear combination of the same few dictionary elements, in the sparse decomposition. Best results are obtained when the dictionary includes a large amount of patches ( $2 \times 10^7$ ) from natural images. They called the methods LSSC (Learned simultaneous sparse coding). They show that this approach, applied on RAW, outperforms the state of the art for both denoising and demosaicking.

Zoran and Weiss investigate in [Zoran and Weiss, 2009] and [Zoran and Weiss, 2011] what makes an image prior a good choice for image restoration (denoising, deblurring and inpainting). They continue the line of research of Roth and Black that proposed the FoE method. A popular task in image denoising is to learn priors over dictionaries of small patches. The authors show that priors that give a higher likelihood for patches of natural images, output denoised patches with a higher PSNR value. For the extension from patches to the whole image, using the trivial idea of dividing the image into patches and denoise each one, has the disadvantage of introducing patch border artifacts after reconstructing the image. They propose the following solution: to select a reconstructed image patch if it is likely under the considered prior. The idea is that the final image should be a collection of the patches likely under the prior, maximizing the Expected Patch Likelihood (EPLL) of the reconstructed image and imposing the similarity to the noisy image constraint. The authors claim that learning priors is an easier task than learning a Markov Random Field (MRF), as in FoE, for example. In FoE, the filters are trained by maximizing the likelihood of training images. In the task of learning better priors, Zoran and Weiss propose to learn a finite Gaussian Mixture Model (GMM) over natural images patches. The approach involving

EPLL and GMM prior outperforms the FoE and KSVD methods, and is better or comparable (depending on the noise level) to the BM3D and LLSC methods, in terms of PSNR.

### **4.3 Current trends: A global approach and deep neural networks**

An extension from non-local patch-based denoising methods to global ones is proposed by Talebi and Milanfar in [Talebi and Milanfar, 2014]. The performance of patch-based methods is tightly connected to the patch-matching step: only having enough similar patches guarantees a good denoising performance. To overcome this limitation, the authors propose a denoising strategy called global image denoising (GLIDE), in which for each pixel one uses information from all pixels in the image. While this sounds like an extremely costly step, a sampling approach based on the Nyström extension solved the complexity issue. Firstly, they pre-filter and sample the image such that the Nyström method is applied on a fraction of the image pixels. Afterwards, using the eigenvalues and the eigenvectors, one estimates the global filter. Finally, the output image is created by shrinking the eigenvalues. This strategy brings an improvement upon the best patch-based methods, both visually and in terms of PSNR. For the GLIDE method, the authors analyze the upper bound on the rate of convergence of the mean-squared error (MSE), and show that it decreases as the image size increasing. Therefore, global denoising is asymptotically optimal. To this purpose, they constructed the oracle GLIDE that includes all the global weights computed from the clean image and they compare it to the oracle for the patch-based methods NLM and BM3D. While the global method was capable of perfectly reconstructing the original clean image, the patch-based methods are not, even though the values of all the parameters are taken from the clean image. This shows that patch-based methods have a limitation that a global method does not.

Recently there is a line of research that increases in popularity: denoising with deep neural networks. Jain and Seung introduced in [Jain and Seung, 2009] the use of convolutional neural networks (CNNs) in the field of digital image denoising. The CNNs are some one of the most popular neural networks, chosen for different applications like traffic sign recognition or hand-written digit recognition. A convolutional network is represented by a concatenation of linear and nonlinear

filtering steps. While the algorithm input and output is represented by one or several images, the intermediate layers are given by the so called hidden feature maps that sum all internal computation of the method involving an extremely large number of parameters. In [Jain and Seung, 2009], they formulate the image denoising task as a learning problem for training the convolutional network. Applying unsupervised learning on specific noise models, the results obtained were better or comparable to those of the state of the art of their time.

An unified denoising and inpainting scheme is proposed in [Xie et al., 2012], combining sparse coding and deep neural networks. Its performance was comparable to that of the K-SVD sparse coding method.

The algorithms involving a multi-layer perceptron (MLP) [Burger et al., 2012] and the Trainable Nonlinear Reaction Diffusion TNRD [Chen and Pock, 2017] are two deep neural methods whose results are comparable the those of the BM3D method. A multi-layer perceptron is a non-linear function that receives as an input a vector and transforms it via hidden layers into an output vector. Its architecture is defined by the size of the layer and the number of hidden layers. While CNNs are created for dealing with images, MLPs are more general and more powerful when trained on large dataset. The MLP-based algorithm in [Burger et al., 2012] is a patch-based denoising method that is learned on an extremely large dataset using a plain neural network. Their idea is to learn an MLP that maps a noisy patch to a clean clean one, where the noisy one is created with additive white Gaussian noise. The best achieved results, that are comparable to the BM3D results, are obtained with a MLP with four hidden layers of size 2047 and a patch size of 17x17, trained on 362 million samples. Time-wise, this method is slower than BM3D but faster than K-SVD.

The TNRD approach in [Chen and Pock, 2017] proposes a framework for several image restoration applications: Gaussian denoising, single super resolution and deblocking of JPEG images. Related to the work of [Liu et al., 2010] that performed learning of PDEs by training data, the TNRD algorithm uses a nonlinear reaction diffusion model (capable of diffusion and enhancing) with trainable filters and influence functions, and parameters learn by using a convolutional network.

Both the MLP and the TNRD algorithms are trained for a particular noise level, and they do not perform well on a different level of noise. In [Zhang et al., 2017] the authors construct a type of denoising convolutional neural network

(DnCNN) able to perform Gaussian denoising with an unknown noise level (blind denoising). Instead of having as an output the denoised image, they designed the DnCNN to predict the residual image (the difference between the noisy and the latent clean image). This method outperforms BM3D and TNRD in terms of PSNR. As in the case of TNRD method, the proposed DnCNN method can handle not only denoising, but also single super resolution and JPEG de blocking.

## **4.4 On denoising techniques**

### **4.4.1 Principles**

There are many denoising method proposed in the literature, however, in the following we show four basic principles that they are based on, as classified in [Lebrun et al., 2012].

#### **1. Bayesian patch-based methods**

The task is computing the probability (likelihood) of the clean image given the noisy one, by using a prior probability distribution on the clean image patch. The Bayesian principle is complemented, in general, by the mixture of Gaussians model for the wanted clean patches.

#### **2. Transform thresholding**

This principle is based on the empirical observation that the majority of the coefficients of a natural image have large values and are sparse. Contrarily, the noise is modelled as white, for Gaussian white noise.

#### **3. Sparse coding**

This principle employs a dictionary of patches, such that most patches can be decomposed with only a few coefficients.

#### **4. Pixel/block averaging**

Pixel averaging and patch averaging are justified by the self-similarity principle of natural images.

#### 4.4.2 Tools

Lebrun et al. [Lebrun et al., 2012] suggest a recipe including several steps, introduced in this section, for improving any denoising approach.

##### 1. Aggregation of estimates

This technique associates to each pixel a set of estimates that are good candidates. This idea was introduced by Coifman and Donoho in [Coifman and Donoho, 1995]. The authors proposed to denoise with an wavelet thresholding algorithm several translations of an image and, after computing the inverse translation, average the estimates.

Common techniques are to weight each estimator inverse proportionally to its variance or with equal weights. While homogeneous patches have a low variance, the opposite happens for patches that contain an edge. Also, while for flat patches there are many similar ones repeated in a natural image, for edge patches there are much fewer and averaging over less candidates gives a noisier results. Therefore, using the aggregation approach can eliminate possible artifacts (halos) introduced by the denoising near edges. Consequently, most of the patch-based denoising methods, including NLM or BM3D, use this technique.

The improvement of using aggregation with a denoising method is caused by averaging more similar patch candidates and by allowing to choose candidates with lower variance (less noisy). Another criteria used for aggregation is to give higher weight to estimators that have a lower risk, for example, with the Stein’s Unbiased Risk Estimator (SURE) as in [Duval et al., 2011] and [Van De Ville and Kocher, 2009].

Deledalle et al. [Deledalle et al., 2012] use different estimates of NLM, obtained by using different size and shape for the patches, such that the local geometry of the image is better preserved.

##### 2. Iteration and oracle filters

After denoising, there can still be some amount of residual noise kept in the image. One can apply a second denoising step to remove it, by using the first denoised image as an oracle for the next step. Making more than one iteration of the oracle filter does not improve much, as mentioned in [Milanfar, 2011].



Figure 4.2: Top (left) clean image and (right) noisy with additive white Gaussian noise,  $\sigma = 25$ . Rows 2 and 3 from left to right: noisy crop from top image followed by its denoising with a DCT thresholding filter (PSNR = 26.85) and successively adding the steps described in this section: use the YUV color space (PSNR = 27.33), uniform aggregation (PSNR = 30.65), variance based aggregation (PSNR = 30.73), and using the first step of denoising as an oracle (PSNR = 31.25). Figure is from [Lebrun et al., 2012].

Another similar idea is to filter the residual image, given by the difference between the noisy and denoised image, and recover some wanted signal that was removed by the first denoising step. Tukey denotes this strategy as

twicing [Tukey, 1976].

### 3. Dealing with colour images

Given a method designed for grey images, performing the trivial extension to color, of applying the denoising independently on each channel, is not ideal, as it can show colour artifacts.

One solution with good results in practice is to change the RGB color space to an YUV one, where the first component models the geometry of the image and the other two its chromatic information. This can be done, for example, by multiplying the following matrix with RGB vector:

$$\begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & -1/2 \\ 1/4 & -1/2 & 1/4 \end{pmatrix} \quad (4.1)$$

As seen from the first line of the YUV matrix, the first channel has a noise level that is lower than each of the RGB channels, as it is an average of the three. Denoising an equivalent less noisy image gives a better result.

Fig.4.2 includes an image example illustrating the usefulness of these proposed techniques, showing an improvement both visually and in terms of PSNR.

#### 4.4.3 Comparison

There are several popular ways to compare the quality of the denoising methods. However, there is no universally accepted measure that gives an estimate for the quality of a denoised image, therefore, a combination of different measures should be employed.

##### 1. Objective measures

One of the most popular one is the PSNR value, described in the Introduction, which measures the Euclidean distance between the original clean image and the denoised one.

In a realistic context, taking a possibly noisy picture with a digital camera, one does not have the clean image. Therefore, the PSNR measure can be used mostly for the scenario when noise is added to a clean image. The same holds for another popular objective measure, the SSIM index, which is supposed to be connected to a perceptual model.

Another limitation of the objective measures is that they are not good in detecting artifacts.

## 2. Visual quality

Contrary to objective measures, humans are excellent in detecting artifacts. However, different subjects might choose different parameters for a certain denoising method (some prefer more blurred images while others would vote for sharper images with some amount of noise kept) or a different ranking of the compared denoising methods.

Also, estimating the visual quality of a large set of images and denoising method is a very difficult and time consuming task for humans.

## 3. The method noise

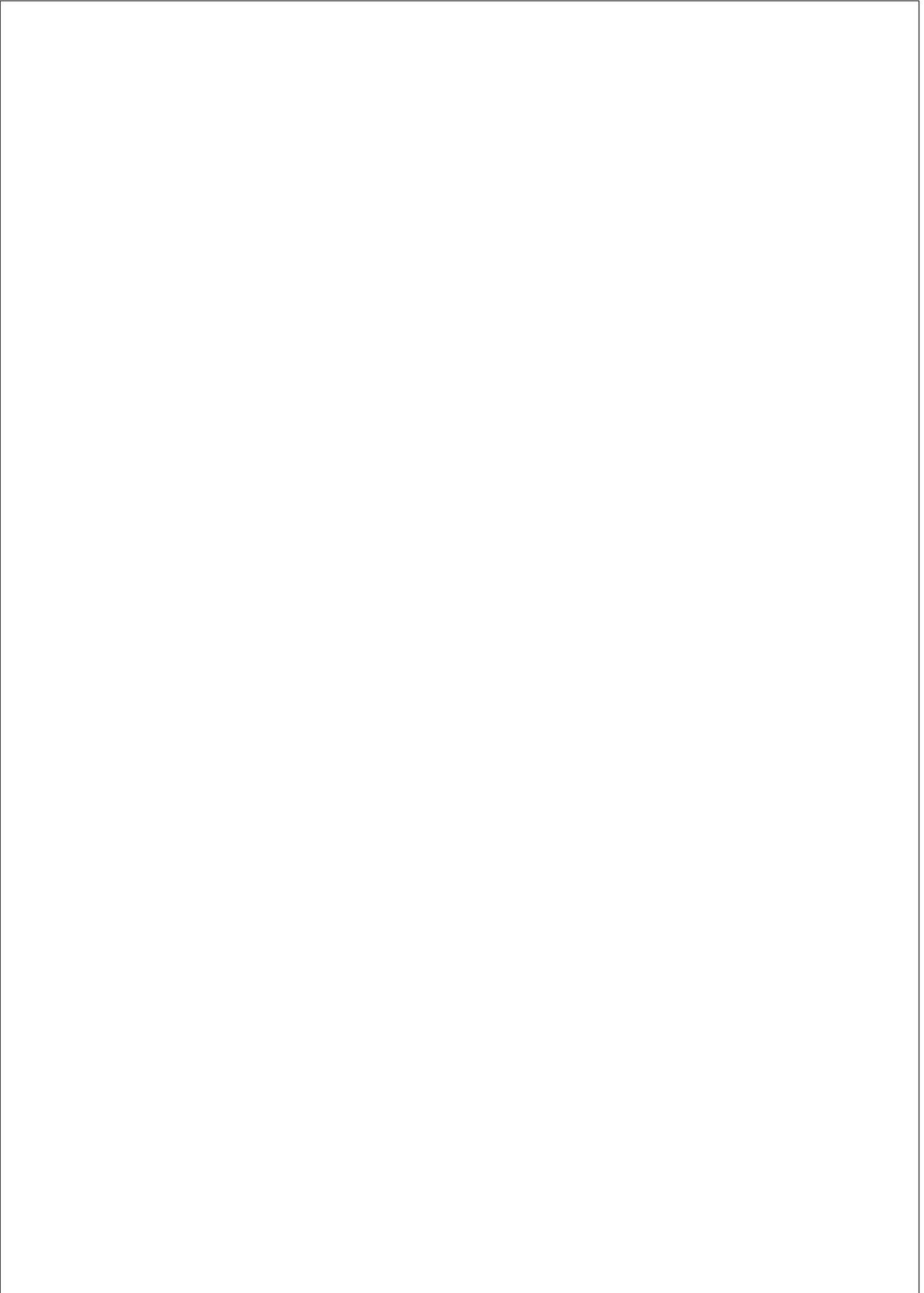
The authors in [Buades et al., 2008] propose the so called method noise: computing the difference between the original noisy image and the denoised one, one should obtain an image that should have an appearance of noise with no recognizable details of the original image. However, this is true for additive white Gaussian noise. As a mathematical analysis of the method noise is a challenging task, a visual inspection is necessary. And even so, the method noise is not reliable for larger noise levels (when standard deviation of white Gaussian noise is larger than 5 or 10) as mentioned in [Lebrun et al., 2012].

## 4. The noise-to-noise principle

The authors in [Buades et al., 2008] affirm that a good denoising method should transform white noise into white noise. For this claim, a homogeneous grey test image is created, and white noise is added to each channel. On this noisy image, a denoising method is applied and the output image is analyzed. This method has the ability of detecting possible artifacts introduced by a denoising method; the fewer the artifacts visible in the denoised grey image, the better the method.

#### 4.4.4 Conclusion

The constant push for increasing resolution, dynamic range and frame-rate carries with it the challenging need to adequately deal with noise. Image denoising has been a topic extensively investigated over the last three decades and many sophisticated and efficient methods have been proposed, such as local iterative approaches based on the TV-minimization model [Rudin et al., 1992] and non-local algorithms like Non-Local Means (NLM) [Buades et al., 2005a] and Block-Matching and 3D Filtering (BM3D) [Dabov et al., 2007]. Lebrun et al. [Lebrun et al., 2012] study the so-called shotgun patch-based methods, which instead of using patches extracted from one image, it simulates using all the possible existing patches. Surprisingly, it does not improve much. One of the possible conclusions is that an image includes sufficient information for a good denoising result. In fact, denoising algorithms have become so good that many researchers have started questioning the need to pursue further this line of research: papers on error bounds for denoising suggest that state-of-the-art algorithms are just a small fraction of a dB below the theoretical optimum performance [Levin and Nadler, 2011b, Chatterjee and Milanfar, 2010], the recent CNN-based denoising methods [Zhang et al., 2017] seem to corroborate this by providing only slight improvements over BM3D, and several recent works implicitly or explicitly assume that denoising algorithms can not be bettered in any significant way so instead they concentrate on boosting their performance by applying them to transforms of image data instead of applying them directly on the image [Bertalmío and Levine, 2014, Ghimpeteanu et al., 2016a], or simply use denoising algorithms to improve solutions to other problems different from denoising [Romano et al., 2017].



## CHAPTER 5

---

### **Denoising: apply it on a less noisy equivalent image**

---

In this chapter we consider a general strategy to improve any image denoising technique by more carefully taking into account the local geometry (direction of gradients and level-lines) of the image to process. It is an image decomposition model that computes the components of the image to be processed in a moving frame that encodes its local geometry (directions of gradients and level-lines). Then, the strategy we develop is to denoise the components of the image in the moving frame in order to preserve its local geometry, which would have been more affected if processing the image directly.

The text of this chapter is based on our journal article with the title “A decomposition framework for image denoising algorithms” [Ghimpeteanu et al., 2016a], and our conference article “Denoising an image by denoising its components in a moving frame” [Ghimpeteanu et al., 2014].

We start this chapter by introducing the line of research that lead to the moving frame approach. We continue with a description of the image decomposition model for gray-level and multi-channel images. We prove that our approach is suitable for image denoising by showing, both theoretically and numerically, that the PSNRs of the components of the noisy image in a well-chosen frame are higher than the PSNR of the noisy image itself on image contours. We conclude by presenting the results obtained by applying the moving frame approach to three denoising methods: VTV minimization, NLM and BM3D. The results confirm the suitability of our approach since they show that it improves each aforementioned denoising method with respect to the PSNR and SSIM metrics on

a whole database of gray-level and color images tested with different noise levels.

## 5.1 Line of similar research

Motivated by the construction of a Fourier theory for  $n$ -channel images that would involve the local geometry of the image, Batard and Berthier [Batard and Berthier, 2013] proposed the following approach: construct an orthonormal moving frame of  $\mathbb{R}^{n+2}$  over the image domain where the first two vector fields are tangent to the image graph and the  $n$  remaining vector fields are normal to that surface. Then, representing the image in this  $n+2$  dimensional moving frame, compute the standard 2D Fourier transform of each of the components, apply Gaussian kernels of different sizes in the Fourier domain, and then project back. This essentially applies a Euclidean heat diffusion on each of the components, obtaining a heat diffusion on the original image that preserves its local geometry throughout the diffusion process. Later on, Batard and Bertalmío [Batard and Bertalmío, 2013], [Batard and Bertalmío, 2014] followed this strategy for the purpose of image denoising. More precisely, instead of applying a heat equation on the components of the image in the moving frame, they applied a denoising method and compared the results with applying that method directly on the image. In [Batard and Bertalmío, 2013], they dealt with the vectorial extension of the total variation-based denoising method of Rudin et al. [Rudin et al., 1992] proposed by Blomgren and Chan [Blomgren and Chan, 1998], and in [Batard and Bertalmío, 2014] they dealt with the so-called vectorial total variation (VTV) of Bresson and Chan [Bresson and Chan, 2008]. In both cases, they showed that this methodology can improve the standard approach in terms of PSNR and Q-Index [Wang and Bovik, 2002] metrics, the latter being more related to perception than the PSNR.

A comparable framework was adopted by Bertalmío and Levine [Bertalmío and Levine, 2014]. Based on the observation that the curvature of the level-lines of a gray-level image is less affected by the noise than the intensity values of the image (assuming that the image has been corrupted by additive Gaussian noise), they conjectured that it can be easier to recover the curvature of the clean image than the clean image itself. Experiments involving 4 different denoising methods: TV denoising performed through gradient descent [Rudin et al., 1992] and Bregman iterative algorithm [Osher et al., 2005], orientation matching using smooth unit tangents [Hahn et al., 2011], NLM [Buades et al., 2005a] and BM3D [Dabov et al., 2007], confirmed that applying a denoising method on the

curvature of the noisy image and then reconstructing an image from the denoised curvature provides a better approximation to the clean image (in terms of PSNR and Q-Index metrics) than applying this denoising method directly on the noisy image.

A similar approach for image denoising is the one of Lysaker et al. [Lysaker et al., 2004a] who first smooth the unit normal field to the image level-lines, then construct a denoised image whose unit normal field would match the smooth field. Some similar approaches for denoising that have been inspired by the work of Lysaker et al. can be found in [Hahn et al., 2011], [Osher et al., 2005], [Rahman et al., 2007].

In this chapter, we show that given a denoising method we can obtain better, cleaner results by denoising the components of an image in a moving frame (as in [Batard and Berthier, 2013], [Batard and Bertalmío, 2013], [Batard and Bertalmío, 2014]), compared to what we would get by denoising the image directly. We formally prove that, along image contours, the PSNR of the components is higher than that of the image, which would explain the ability of our framework to better preserve image details regardless of the particular image denoising technique that is applied. We have been able to improve three denoising methods of different types: a local variational method (VTV, [Bresson and Chan, 2008]), a patch-based method (NLM, [Buades et al., 2005a]), and a method combining a patch-based approach with a filtering in spectral domain approach (BM3D, [Dabov et al., 2007]); the improvement is both in terms of PSNR and SSIM [Wang et al., 2004] metrics, and for grayscale and color images over a standard image database, demonstrating the consistency of our strategy. Compared to the curvature-based approach in [Bertalmío and Levine, 2014] or the above mentioned methods based on denoising fields ( [Lysaker et al., 2004a], [Hahn et al., 2011], [Osher et al., 2005], [Rahman et al., 2007]), the asset of our framework holds in the simplicity of the reconstruction step, consisting simply in applying a matrix transform to the denoised components to obtain the denoised image, instead of a second or third order PDE evolution equation as in the previous techniques.

## 5.2 Image decomposition in a moving frame

### 5.2.1 The gray-level case

Let  $I: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  be a gray-level image, and  $(x, y)$  be the standard coordinate system of  $\mathbb{R}^2$ . We denote by  $I_x$  resp.  $I_y$  the derivative of  $I$  with respect to  $x$  resp.  $y$ , and by  $\nabla I$  the gradient of  $I$ .

Our image decomposition model for  $I$  is a two-stages approach: first, we construct an orthonormal moving frame  $(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{N})$  of  $(\mathbb{R}^3, \|\cdot\|_2)$  over  $\Omega$  that encodes the local geometry of  $I$ . Then, we compute the components  $(J^1, J^2, J^3)$  of the  $\mathbb{R}^3$ -valued function  $(0, 0, I)$  in that moving frame.

More precisely, we consider a scaled version  $\mu I$  of  $I$ , for  $\mu \in ]0, 1]$ , and its graph, which is the surface  $S$  in  $\mathbb{R}^3$  parametrized by

$$\psi: (x, y) \mapsto (x, y, \mu I(x, y)) \quad (5.1)$$

The orthonormal moving frame  $(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{N})$  we consider is the following: the vector field  $\mathbf{Z}_1$  is tangent to the surface  $S$  and indicates the direction of the steepest slope at each point of  $S$ ; the vector field  $\mathbf{Z}_2$  is tangent to  $S$  and indicates the direction of the lowest slope at each point of  $S$ . It follows that  $\mathbf{N}$  is normal to the surface since we require  $(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{N})$  to be orthonormal.

The moving frame  $(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{N})$  can be constructed as follows. Let  $\mathbf{z}_1 = (\mu I_x, \mu I_y)^T$  be the gradient of  $\mu I$  and  $\mathbf{z}_2 = (-\mu I_y, \mu I_x)^T$  indicating the direction of the level-lines of  $\mu I$ . On homogeneous regions of  $I$ , i.e. at pixel locations  $(x, y)$  where  $I_x(x, y) = I_y(x, y) = 0$ , we define  $\mathbf{z}_1 = (1, 0)^T$  and  $\mathbf{z}_2 = (0, 1)^T$ . Then,  $\mathbf{Z}_1$  and  $\mathbf{Z}_2$  are given by the following expressions

$$\mathbf{Z}_i = \frac{d\psi(\mathbf{z}_i)}{\|d\psi(\mathbf{z}_i)\|_2}, \quad i = 1, 2 \quad (5.2)$$

where  $d\psi$  stands for the differential of  $\psi$ , which maps vector fields on  $\Omega$  to tangent vector fields of  $S$ . The expression of the unit normal  $\mathbf{N}$  is then obtained as the vectorial product between  $\mathbf{Z}_1$  and  $\mathbf{Z}_2$ .

The explicit expressions of the vector fields  $\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{N}$  are given by the matrix

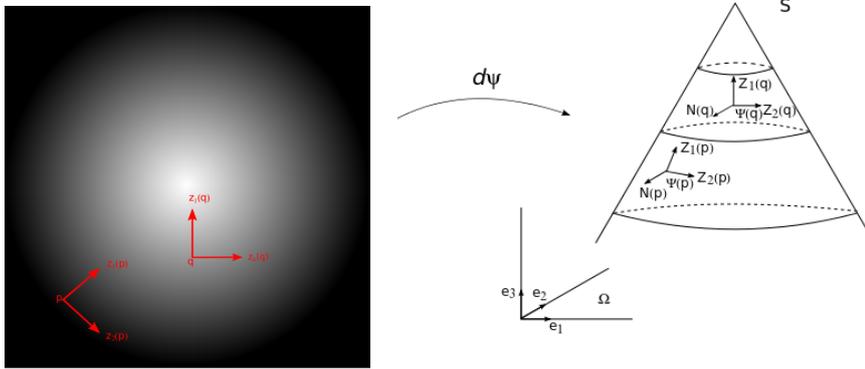


Figure 5.1: Moving frame encoding the local geometry of a gray-level image. Left: original gray-level image and a moving frame  $(z_1, z_2)$  indicating the direction of the gradient and the level-line of the image at two points  $p$  and  $q$  of the image domain  $\Omega$ . Right: the orthonormal moving 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  $\psi(q)$ .

field

$$P = \begin{pmatrix} \frac{I_x}{\sqrt{|\nabla I|^2(1 + \mu^2|\nabla I|^2)}} & \frac{-I_y}{|\nabla I|} & \frac{-\mu I_x}{\sqrt{1 + \mu^2|\nabla I|^2}} \\ \frac{I_y}{\sqrt{|\nabla I|^2(1 + \mu^2|\nabla I|^2)}} & \frac{I_x}{|\nabla I|} & \frac{-\mu I_y}{\sqrt{1 + \mu^2|\nabla I|^2}} \\ \frac{\mu|\nabla I|^2}{\sqrt{|\nabla I|^2(1 + \mu^2|\nabla I|^2)}} & 0 & \frac{1}{\sqrt{1 + \mu^2|\nabla I|^2}} \end{pmatrix}, \quad (5.3)$$

where the coordinates of the vector field  $Z_1$  are given in the first column, the coordinates of  $Z_2$  in the second column, and the coordinates of  $N$  in the third column.

Fig.5.1 illustrates the moving frames  $(z_1, z_2)$  and  $(Z_1, Z_2, N)$  aforementioned for a simple image. The left image shows the moving frame  $(z_1, z_2)$  at two points  $p$  and  $q$  of the domain  $\Omega$ , and the right image shows the induced moving frame  $(Z_1, Z_2, N)$  attached to the surface  $S$  at the points  $\psi(p)$  and  $\psi(q)$ .

Denoting by  $(e_1, e_2, e_3)$  the orthonormal frame of  $(\mathbb{R}^3, \|\cdot\|_2)$ , where  $e_1 =$



Figure 5.2: From left to right: gray-level image "Lena", component  $J^1$ , component  $J^3$ .

$(1, 0, 0)$ ,  $e_2 = (0, 1, 0)$ ,  $e_3 = (0, 0, 1)$ , the matrix  $P$  in (5.3) is nothing but the frame change field from  $(e_1, e_2, e_3)$  to  $(Z_1, Z_2, N)$ , meaning that the components of the  $\mathbb{R}^3$ -valued function  $(0, 0, I)$  in the new frame, denoted by  $(J^1, J^2, J^3)$ , are given by

$$\begin{pmatrix} J^1 \\ J^2 \\ J^3 \end{pmatrix} = P^{-1} \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}. \quad (5.4)$$

Computing formula (5.4) gives that the component  $J^2$  is identically 0.

Fig. 2 shows the gray-level image "Lena" and its components  $J^1$  and  $J^3$  for  $\mu = 0.05$ . We observe that the component  $J^1$  contains the edges and textures of the image, which was expected according to formula (5.4) since the third component of the vector field  $Z_1$  is determined by the image gradient's norm, which is high on edges and textures. The component  $J^3$  is similar to the original image from which the gradient's norm has been subtracted.

Finally, the parameter  $\mu$  can be viewed as a smoothing parameter for the moving frame encoding the local geometry of the original image  $I$ . In fact, it turns out that  $\mu$  plays a key role in our strategy for image denoising, as shown in the next sections.

### 5.2.2 The multi-channel case

We aim at extending the image decomposition model of Sect. II.A from gray-level to  $n$ -channel images  $I = (I^1, \dots, I^n): \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n$ ,  $n > 1$ , by following a similar approach: first, we construct an orthonormal moving frame  $(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{N}_1, \dots, \mathbf{N}_n)$  of  $(\mathbb{R}^{n+2}, \|\cdot\|_2)$  over  $\Omega$  that encodes the local geometry of  $I$ . Then, we compute the components  $(J^1, J^2, \dots, J^{n+2})$  of the  $\mathbb{R}^{n+2}$ -valued function  $(0, 0, I^1, \dots, I^n)$  in that moving frame.

As in the gray-level case, the first step consists in considering a scaled version  $\mu I$  of  $I$ , for  $\mu \in ]0, 1]$ , and its graph, which is the surface  $S$  in  $\mathbb{R}^{n+2}$  parametrized by

$$\psi: (x, y) \longrightarrow (x, y, \mu I^1(x, y), \dots, \mu I^n(x, y)) \quad (5.5)$$

The moving frame  $(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{N}_1, \dots, \mathbf{N}_n)$  we consider is then the following: the vector field  $\mathbf{Z}_1$  is tangent to the surface  $S$  and indicates the direction of the steepest slope at each point of  $S$ ; the vector field  $\mathbf{Z}_2$  is tangent to  $S$  and indicates the direction of the lowest slope at each point of  $S$ , and  $\mathbf{N}_1, \dots, \mathbf{N}_n$  are normals to the surface. Note that, unlike the gray-level case, there is an infinite number of unit normals to the surface.

The moving frame  $(\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{N}_1, \dots, \mathbf{N}_n)$  can be constructed as follows. As in the gray-level case,  $\mathbf{Z}_1$  and  $\mathbf{Z}_2$  can be recovered from the directions  $\mathbf{z}_1$  and  $\mathbf{z}_2$  of highest and lowest variations of the scaled image  $\mu I$  under the map (5.2), these latter being the eigenvectors of the structure tensor associated to  $\mu I$ , which is given by

$$\begin{pmatrix} \sum_{k=1}^n (\mu I_x^k)^2 & \sum_{k=1}^n \mu^2 I_x^k I_y^k \\ \sum_{k=1}^n \mu^2 I_x^k I_y^k & \sum_{k=1}^n (\mu I_y^k)^2 \end{pmatrix} \quad (5.6)$$

On homogeneous regions, i.e. at pixel locations  $(x, y)$  where  $I_x^k(x, y) = I_y^k(x, y) = 0 \ \forall k \in \{1, \dots, n\}$ , we set  $\mathbf{z}_1 = (1, 0)^T$  and  $\mathbf{z}_2 = (0, 1)^T$ . It is worth noting that, unlike gray-level images, multi-channel images do not have necessarily level-lines, meaning that the smallest eigenvalues of the structure tensor are not necessarily 0.

We then need to select a set of  $n$  vector fields  $\mathbf{N}_1, \dots, \mathbf{N}_n$  normal to the surface and orthogonal to each other in order to complete the orthonormal moving frame. A natural approach for constructing them is to consider the canonical vectors  $\mathbf{e}_3 = (0, 0, 1, 0, \dots, 0), \dots, \mathbf{e}_{n+2} = (0, \dots, 0, 1)$  from which we apply the Gram-Schmidt orthonormalization process to the frame

field  $(Z_1, Z_2, e_3, \dots, e_{n+2})$ .

Finally, denoting by  $P$  the matrix field encoding the moving frame  $(Z_1, Z_2, N_1, \dots, N_n)$ , i.e. the first column of  $P$  contains the coordinates of  $Z_1$ , the second column the coordinates of  $Z_2$ , and the  $i$ -th column the coordinates of  $N_{i-2}$  for  $i \in \{3, \dots, n+2\}$ , the components  $(J^1, \dots, J^{n+2})$  of the  $\mathbb{R}^{n+2}$ -valued function  $(0, 0, I^1, \dots, I^n)$  in the frame  $(Z_1, Z_2, N_1, \dots, N_n)$  are given by

$$\begin{pmatrix} J^1 \\ J^2 \\ J^3 \\ \vdots \\ J^{n+2} \end{pmatrix} = P^{-1} \begin{pmatrix} 0 \\ 0 \\ I^1 \\ \vdots \\ I^n \end{pmatrix} \quad (5.7)$$

Unlike the gray-level case, it is not possible to plot the moving frame in a trivial way since the surface  $S$  lives in a space of dimension greater than or equal to 4.

### 5.2.3 Application to image denoising

The framework we propose for denoising an image while systematically taking into account its local geometry is based on applying image denoising techniques to the components of the image in the moving frame constructed above instead of applying the technique to the image itself. This methodology has already been used in [Batard and Berthier, 2013], [Batard and Bertalmío, 2013], [Batard and Bertalmío, 2014] with local regularization/denoising methods, but it can actually be extended to any denoising technique. In this section, we give more details about our approach dealing with gray-level and color images.

#### Gray-level images

In the experiments performed throughout this article, the strategy on gray-level images  $I: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  is the following:

1. Process  $I$  with some denoising technique  $F$  and call the output image  $I_{den}$ .

2. Compute the components  $(J^1, J^2, J^3)$  of  $I$  in the moving frame (5.3), for some scalar  $\mu$ , with formula (5.4). Apply the same denoising technique  $F$  to the components to obtain the processed components  $(J_{den}^1, J_{den}^2, J_{den}^3)$ . Then, apply the inverse frame change matrix field to the processed components, i.e.

$$\begin{pmatrix} I_{denMF}^1 \\ I_{denMF}^2 \\ I_{denMF}^3 \end{pmatrix} := P \begin{pmatrix} J_{den}^1 \\ J_{den}^2 \\ J_{den}^3 \end{pmatrix} \quad (5.8)$$

and denote by  $I_{denMF}$  the third component  $I_{denMF}^3$ .

3. Compare  $I_{den}$  and  $I_{denMF}$  with the metrics PSNR and SSIM.

We have  $\lim_{\mu \rightarrow 0} I_{denMF} = I_{den}$  since  $\lim_{\mu \rightarrow 0} (J^1, J^2, J^3)^T = \lim_{\mu \rightarrow 0} P^{-1}(0, 0, I)^T = (0, 0, I)^T$ , in which case  $(I_{denMF}^1, I_{denMF}^2, I_{denMF}^3)^T = P(J_{den}^1, J_{den}^2, J_{den}^3)^T = P(0, 0, I_{den})^T = (0, 0, I_{den})^T$ .

### Color images

The extension to color images is not straightforward because of the flexibility of the choice of color space and the way in which the moving frame approach can be applied (channel-wise, only to selected channels, or vectorially). We will see in the next two sections that the color space and manner in which the approach is applied both depend on the image denoising technique. However, in all of the experiments performed throughout this article, our approach for color images  $I: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$  is of the form:

1. Process  $I$  with an image denoising technique  $F$  and call the output image  $I_{den}$ .
2. Apply the same image denoising technique  $F$  to the components in some moving frame related to the channels of the image or the full image itself. Then apply the inverse frame change matrix field to the processed components, from which a color image  $I_{denMV}$  is reconstructed.

3. Compare  $I_{den}$  and  $I_{denMV}$  with the metrics PSNR and SSIM. Note that SSIM has been originally designed for gray-level images, and we define the SSIM Index for color images as the mean of the SSIM Index of each color channel.

Finally, we would like to point out that the strategy described above can actually be applied using any moving frame. The problem regarding the choice of the moving frame has already been treated in [Batard and Bertalmío, 2014], where we demonstrated numerically that when applying our approach to the Vectorial Total Variation-based denoising method, the best results are obtained with the frame constructed above, but close results are obtained when  $Z_1, Z_2$  are randomly chosen (but still orthonormal) in the tangent planes of the surface parametrized by (5.1) dealing with gray-level images and (5.5) for color images. We also observed that very poor results are obtained when  $Z_1, Z_2$  are not in the tangent spaces anymore.

### 5.3 The noise level is higher on the intensity values of a gray-level image than on its components in a well-chosen moving frame

The aim of this section is to demonstrate that for properly chosen  $\mu$ , the components  $J^1(I)$  and  $J^3(I)$  of a gray-level image  $I$  in the moving frame (5.3), determined by formula (5.4), are less affected under additive Gaussian noise than the image is. We then deduce that, for a given denoising method, it can be easier to recover the clean components than the clean image directly. In our analysis, we distinguish image edges from the homogeneous regions.

Let  $I = a + n$  be a gray-level image resulting from the corruption of an image  $a$  with additive Gaussian noise  $n$  of zero mean and standard deviation  $\sigma$ .

#### 5.3.1 Edges

Inspired by the approach of Bertalmío and Levine [Bertalmío and Levine, 2014], we obtain the following result:

Proposition: At the locations of the image domain where  $|\nabla a| \gg |\nabla n|$  (likely the case at contours of  $I$ ), and assuming that central differences are used to approximate  $\nabla I$ , we have that for  $\mu > 0$ ,

$$PSNR(J^1(I)) \geq PSNR(I)$$

$$PSNR(J^3(I)) > PSNR(I)$$

The computation of the local PSNR of the components  $J^1(I)$  and  $J^3(I)$  requires the knowledge of their amplitude  $ampl$ . Assuming that  $I$  is in the range  $[0, 255]$  and central differences are used to compute  $\nabla I$ , the amplitudes are

$$ampl(J^3(I)) = 255 \tag{5.9}$$

$$ampl(J^1(I)) = 255 \times \frac{\sqrt{2} \times 127.5 \mu}{\sqrt{1 + 2(127.5 \mu)^2}} \tag{5.10}$$

Indeed, from (5.3) and (5.4) we know that

$$J^3(I) = \frac{I}{\sqrt{1 + \mu^2 |\nabla I|^2}} \tag{5.11}$$

and

$$J^1(I) = \frac{\mu I |\nabla I|}{\sqrt{1 + \mu^2 |\nabla I|^2}} \tag{5.12}$$

From (5.11), it is clear that  $J^3(I) \in [0, 255]$ , meaning that  $ampl(J^3(I)) = 255$ . The upper bound of  $J^1(I)$  can easily be computed once we notice that  $J^1(I)$  is of the form

$$f: A \mapsto I \frac{A}{\sqrt{1 + A^2}}$$

for  $A = \mu |\nabla I|$ . Then, as the function  $f$  is increasing, we deduce that the maximum of  $J^1(I)$  is reached when  $I = 255$  and  $|\nabla I|$  is maximum, this latter quantity depending on the discrete differentiation used. Assuming that central differences are used, we obtain the following upper bound for  $J^1(I)$ :

$$255 \times \frac{\sqrt{2} \times 127.5 \mu}{\sqrt{1 + 2(127.5 \mu)^2}}$$

which corresponds to its amplitude since its lower bound is clearly 0.

At the locations of the image domain where  $|\nabla a| \gg |\nabla n|$  (likely the case at contours of  $I$ ), we have

$$\begin{aligned}
 J^1(I) &= \frac{\mu(a+n)|\nabla I|}{\sqrt{1+\mu^2|\nabla I|^2}} \\
 &= \frac{\mu a|\nabla a|}{\sqrt{1+\mu^2|\nabla a|^2}} \frac{|\nabla I|\sqrt{1+\mu^2|\nabla a|^2}}{|\nabla a|\sqrt{1+\mu^2|\nabla I|^2}} + \frac{\mu n|\nabla I|}{\sqrt{1+\mu^2|\nabla I|^2}} \\
 &\approx \frac{\mu a|\nabla a|}{\sqrt{1+\mu^2|\nabla a|^2}} + \frac{\mu n|\nabla I|}{\sqrt{1+\mu^2|\nabla I|^2}} = J^1(a) + n_1. \quad (5.13)
 \end{aligned}$$

Then

$$n_1 = \frac{\mu n|\nabla I|}{\sqrt{1+\mu^2|\nabla I|^2}}$$

can be considered as additive noise for the first component  $J^1(I)$  with local variance

$$Var(n_1) = Var\left(\frac{\mu n|\nabla I|}{\sqrt{1+\mu^2|\nabla I|^2}}\right) = \frac{\mu^2|\nabla I|^2}{1+\mu^2|\nabla I|^2}\sigma^2.$$

Indeed, we can assume  $|\nabla I|$  is constant at edge locations and  $Var(n) = \sigma^2$  since  $n$  is an independent and identically distributed random variable.

Thus at likely edges,

$$PSNR(J^1(I)) = 20 \log_{10}\left(255 \frac{127.5\sqrt{2}\mu}{\sqrt{1+2\mu^2(127.5)^2}} \times \right. \quad (5.14)$$

$$\left. \frac{\sqrt{1+\mu^2|\nabla I|^2}}{\mu|\nabla I|\sigma} \right) \quad (5.15)$$

$$\geq 20 \log_{10}\left(\frac{255}{\sigma}\right) = PSNR_{loc}(I).$$

As in the computation for  $J^1(I)$ , at likely edges,  $|\nabla I| \approx |\nabla a|$  and thus

$$\begin{aligned} J^3(I) &= \frac{a+n}{\sqrt{1+\mu^2|\nabla I|^2}} \\ &= \frac{a}{\sqrt{1+\mu^2|\nabla a|^2}} \frac{\sqrt{1+\mu^2|\nabla a|^2}}{\sqrt{1+\mu^2|\nabla I|^2}} + \frac{n}{\sqrt{1+\mu^2|\nabla I|^2}} \\ &\approx \frac{a}{\sqrt{1+\mu^2|\nabla a|^2}} + \frac{n}{\sqrt{1+\mu^2|\nabla I|^2}} = J^3(a) + n_3 \end{aligned} \quad (5.16)$$

where

$$n_3 = \frac{n}{\sqrt{1+\mu^2|\nabla I|^2}}$$

can be considered as additive noise for the third component  $J^3(I)$ . Then its local variance at likely edges is approximately

$$\text{Var}(n_3) = \text{Var}\left(\frac{n}{\sqrt{1+\mu^2|\nabla I|^2}}\right) = \frac{\sigma^2}{1+\mu^2|\nabla I|^2}.$$

Thus at likely edges, we have that

$$\begin{aligned} PSNR(J^3(I)) &= 20 \log_{10} \left( 255 \frac{\sqrt{1+\mu^2|\nabla I|^2}}{\sigma} \right) \\ &> 20 \log_{10} \left( \frac{255}{\sigma} \right) = PSNR(I). \end{aligned} \quad (5.17)$$

According to formula (5.17),  $PSNR(J^3(I))$  is a strictly increasing function of  $\mu$  that tends to  $+\infty$  when  $\mu \rightarrow +\infty$  and to  $PSNR(I)$  when  $\mu \rightarrow 0$ . According to formula (5.14),  $PSNR(J^1(I))$  is a decreasing function of  $\mu$  that tends to  $PSNR(I)$  when  $\mu \rightarrow +\infty$  and whose limit at  $\mu = 0$  is

$$20 \log_{10} \left( \frac{255 \times 127.5 \sqrt{2}}{|\nabla I| \sigma} \right)$$

Hence, we deduce that, at the locations of likely edges, the higher  $\mu$  is, the better the recovery of the clean component  $J^3(a)$ , whereas the smaller  $\mu$  is, the better

the recovery of the clean component  $J^1(a)$ .

Finally, from the fact that  $|\nabla I| \approx |\nabla a|$ , we have

$$P_{31}(I) \approx P_{31}(a) \quad P_{33}(I) \approx P_{33}(a) \quad (5.18)$$

(see (5.3)) and it follows from the reconstruction equation

$$I_{denMF} := P_{13}(I)J^1(I)_{den} + P_{33}(I)J^3(I)_{den} \quad (5.19)$$

(see (5.8)) that

$$I_{denMF} \approx P_{13}(a)J^1(I)_{den} + P_{33}(a)J^3(I)_{den} \quad (5.20)$$

From Prop. 1, we intuit that  $J^1(I)_{den}$  and  $J^3(I)_{den}$  are respectively a better approximation of  $J^1(a)$  and  $J^3(a)$  than  $I_{den}$  is an approximation of  $a$ . Hence, from (5.20) and

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

we claim that (5.19) is a better reconstruction of  $a$  than  $I_{den}$ , from which it follows that the value for the parameter  $\mu$  that provides the better reconstruction of the edges of the clean image is strictly positive since for  $\mu = 0$ , we have  $I_{denMF} = I_{den}$ .

### 5.3.2 Homogeneous regions

We now treat the case where  $|\nabla a| \ll |\nabla n|$  which occurs on homogeneous or slowly varying regions. At such locations in the image domain, we have

$$J^1(I) \approx \frac{\mu I |\nabla n|}{\sqrt{1 + \mu^2 |\nabla n|^2}} \quad (5.21)$$

$$J^3(I) \approx \frac{I}{\sqrt{1 + \mu^2 |\nabla n|^2}} \quad (5.22)$$

Note that for  $\mu > 0$ , the range, and thus the fluctuations, of  $J^1(I)$  and  $J^3(I)$  are attenuated with respect to that of  $I$ . In addition, while it is challenging to give a formal proof, the experiments in Table 5.1 (explained in the section C. Numerical results) indicate that for small values of  $\mu > 0$ ,

$$PSNR(J^1(I)) > PSNR(I). \quad (5.23)$$

In the same case, for small values of  $\mu > 0$ , we obtain the approximation:

$$PNSR(J^3(I)) \approx PNSR(I). \quad (5.24)$$

Therefore, in homogeneous regions, it also appears that rather than denoising  $I$  directly it should be better to denoise  $J^1(I)$  and  $J^3(I)$ .

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

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

so  $J^1(I)_{den}$  is given increasingly more weight in the reconstruction as  $|\nabla n|$  gets larger. Given the observation in (5.23) for appropriately chosen  $\mu$ , this should also benefit the result of the proposed approach. Therefore, in homogeneous regions, we can expect that  $I_{denMF}$  should be at least as good as  $I_{den}$ , and more likely, better. Experimental results in Section IV corroborate this conjecture.

Finally, from the analysis performed in this section, we conclude that the best reconstruction of the clean image  $a$  should be obtained by taking  $\mu$  very small.

### 5.3.3 Numerical results

From the two previous sections, we know that the parameter  $\mu$  plays a key role in the quality of the reconstruction of the image. In order to get more insight on the optimal values that  $\mu$  should take, we run an experiment where we compute the PSNR of the components  $J^1(I)$  and  $J^3(I)$  for several (constant) values of  $\mu$  and noise levels  $\sigma$  on the whole Kodak database [<http://r0k.us/graphics/kodak/>, 2017]. Table 5.1 reports the results for  $\sigma = 5, 10, 15, 20, 25$  and  $\mu = 1.0, 0.1, 0.01, 0.005, 0.001, 0.0001$ . We observe that the PSNR of the components are higher than the PSNR of the image for  $\mu \in ]0, 0.005]$  at each of the noise level aforementioned. Note also that the upper bound 0.005 can be raised to 0.01 for the noise levels  $\sigma = 5, 10$ .

Table 5.1: 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

## 5.4 Experiments

Preliminary experiments showed that the value of the parameter  $\mu$  that provides the best denoising result depends on the image content, the noise level, the denoising method involved, and the measure we use for evaluating the denoised image. As a consequence, it is rather difficult to automatize the value of  $\mu$ .

Nonetheless, we have found in all our experiments that the value  $\mu = 0.001$  systematically provides better results than the value  $\mu = 0$ , this latter corresponding to the standard denoising method according to Remark 1 in sect. II.C. Experiments also showed that the optimal value is systematically very close to 0.001 when testing non local methods (NLM and BM3D) whatever the image content, the noise level, and the measure we use for evaluating the denoised image are. On the other hand, the optimal value for  $\mu$  greatly depends on the noise level when testing the local method VTV. In what follows, we report our results for  $\mu = 0$ ,  $\mu = 0.001$  with the three denoising methods VTV, NLM, BM3D, as well as for the optimal values of  $\mu$  in the VTV case.

### 5.4.1 The moving frame approach applied to the Vectorial Total Variation-based denoising method

In this section, we apply our moving frame approach to the Vectorial Total Variation-based denoising method of Bresson and Chan [Bresson and Chan, 2008], defined as follows on a  $n$ -channel image  $I_0: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n$

$$I_{VTV} := \arg \min_I \int_{\Omega} \frac{1}{2\lambda} \|I - I_0\|_2^2 + VTV(I) \, d\Omega \quad (5.25)$$

where the Vectorial Total Variation  $VTV(I)$  of  $I$  is defined by

$$\sup \left( \int_{\Omega} \langle I, \nabla^* \eta \rangle_2 \, d\Omega; \eta \in C_c^\infty(\Omega; \mathbb{R}^n), \|\eta(x)\|_2 \leq 1 \, \forall x \in \Omega \right) \quad (5.26)$$

for  $\nabla^*$  being the adjoint of the Jacobian operator.

Notice that on gray-level images, this approach reduces to the Rudin-Osher-Fatemi (ROF) model [Rudin et al., 1992].

In what follows, we describe our methodology for both gray-level and color images and report the PSNR and SSIM index values.

#### Gray-level images

As shown by Chambolle [Chambolle, 2004], the ROF denoising model can be solved through a projection algorithm. In [Bresson and Chan, 2008], Bresson and Chan showed that their vectorial extension of the ROF denoising model can be solved through a vectorial extension of Chambolle’s projection algorithm. Our moving frame approach associated to the ROF denoising model can then be summarized as follows:

1. Take a clean gray-level image  $a$  and add Gaussian noise of variance  $\sigma$  to it to create a noisy image  $I$ .
2. Apply the ROF denoising model (5.25) to  $I$  using Chambolle’s projection algorithm, obtaining a denoised image  $I_{ROF}$ .

3. Consider the moving frame associated to  $I$  (see (5.3)), for some parameter value  $\mu$ , and compute the components  $(J^1, J^2, J^3)^T$  of  $I$  in this moving frame (see (5.4)).  
Then, apply the VTV-based denoising model (5.25) to these components using the vectorial extension of Chambolle’s projection algorithm, obtaining the denoised components  $(J_{VTV}^1, J_{VTV}^2, J_{VTV}^3)$ , from which a denoised image  $I_{ROFMF} := I_{ROFMF}^3$  is reconstructed using Eq. (5.8).
4. Compute the PSNR and SSIM index values of  $I_{ROF}$  and  $I_{ROFMF}$  with respect to the ground truth  $a$ .

The ROF model that we compute in step 2 does not have any parameter to tune (assuming that  $\lambda$  in (5.25) is a Lagrange multiplier associated to the noise level), whereas our moving frame approach associated to that model in step 3 is parametrized by the scalar  $\mu$  involved in the moving frame.

Table 5.2 reports the average PSNR and SSIM index over the Kodak database of both  $I_{ROF}$  and  $I_{ROFMF}$  images for the parameter value  $\mu = 0.001$ . We see that our moving frame approach slightly outperforms the standard approach at each noise level tested with respect to both PSNR and SSIM metrics.

Table 5.3 reports the average PSNR and SSIM index over the Kodak database for the values of  $\mu$  that provides the best average PSNR values on the database for our approach. Comparing the results with the ones reported in Table 5.2, we see that the improvement is much more important when optimizing  $\mu$  on the whole database. Note that optimizing  $\mu$  for each image would have improved the results reported even more.

### Color images

We proceed as follows.

1. Take a clean color image  $a$  and add Gaussian noise of variance  $\sigma$  to it to create the noisy image  $I$ .
2. Apply the VTV-based denoising model (5.25) to  $I$  using the vectorial extension of the Chambolle’s projection algorithm, obtaining a denoised image  $I_{VTV}$ .

Table 5.2: Comparison of the standard and our moving frame approach with  $\mu = 0.001$  for the *VTV*-based denoising method, at different noise levels. Average PSNR and SSIM Index (x100) over the kodak database: the gray-level case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	35.44	31.45	29.30	27.83	26.79
PSNR Moving frame	35.54	31.56	29.39	28.02	26.89

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	93.73	87.00	81.22	76.05	72.39
SSIM Index Moving frame	93.88	87.22	81.57	77.33	73.24

Table 5.3: Our moving frame approach with optimal value of  $\mu$  for the *VTV*-based denoising method, at different noise levels. Average PSNR and SSIM Index (x100) over the kodak database: the gray-level case.

Approach \ Noise variance	5	10	15	20	25
PSNR Moving frame	36.36	32.23	30.04	28.60	27.49
SSIM Index Moving frame	94.61	88.37	83.22	78.71	74.78

Parameter \ Noise variance	5	10	15	20	25
$\mu$	0.008	0.005	0.005	0.004	0.004

3. Consider the moving frame associated to  $I$  following the method described in sect. 5.2.2, and compute the components  $(J^1, J^2, J^3, J^4, J^5)^T$  of  $I$  in this moving frame using (5.7).

Then apply the VTV-based denoising model to these components using again the vectorial extension of Chambolle’s projection algorithm, obtaining the denoised components  $(J_{VTV}^1, J_{VTV}^2, J_{VTV}^3, J_{VTV}^4, J_{VTV}^5)^T$ , to which the inverse transform of (5.7) is applied, i.e.

$$\begin{pmatrix} I_{VTVMF}^1 \\ I_{VTVMF}^2 \\ I_{VTVMF}^3 \\ I_{VTVMF}^4 \\ I_{VTVMF}^5 \end{pmatrix} : = P \begin{pmatrix} J_{VTV}^1 \\ J_{VTV}^2 \\ J_{VTV}^3 \\ J_{VTV}^4 \\ J_{VTV}^5 \end{pmatrix}.$$

The output denoised color image is

$$I_{VTVMF} : = (I_{VTVMF}^3, I_{VTVMF}^4, I_{VTVMF}^5).$$

4. Compute the PSNR and SSIM index values of  $I_{VTV}$  and  $I_{VTVMF}$  with respect to the ground truth  $a$ .

As in the gray-level case, the VTV-based denoising model that we compute in step 2 does not have parameters to tune (assuming that  $\lambda$  in (5.25) is a Lagrange multiplier associated to the noise level), whereas our moving frame approach associated to that model in step 3 is parametrized by the scalar  $\mu$  determining the moving frame.

Table 5.4 reports the average PSNR and SSIM index over the Kodak database of both  $I_{VTV}$  and  $I_{VTVMF}$  images for the parameter value  $\mu = 0.001$ , and Table 5.5 reports the average PSNR and SSIM index over the Kodak database for the values of  $\mu$  that provides the best average PSNR values on the database for our approach. As in the gray-level case, we observe that the improvement is low for  $\mu = 0.001$  and much more important when optimizing  $\mu$  on the whole database. Again, optimizing  $\mu$  for each image would have improved our results.

In Fig. 5.3, we show an example comparing our approach with the standard VTV-based denoising model. Our method better preserves fine details such as, for example, the texture of the tree leaves and grass.

Table 5.4: Comparison of the standard and our moving frame approach with  $\mu = 0.001$  for the *VTV*-based denoising method, at different noise levels. Average PSNR and SSIM Index (x100) over the kodak database: the color case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	36.31	32.46	30.35	28.79	27.57
PSNR Moving frame	36.45	32.60	30.51	28.96	27.75

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	93.82	87.79	82.77	78.24	74.34
SSIM Index Moving frame	93.97	88.10	83.21	78.65	74.98

Table 5.5: Our moving frame approach with optimal value of  $\mu$  for the *VTV*-based denoising method, at different noise levels. Average PSNR and SSIM Index (x100) over the kodak database: the color case.

Approach \ Noise variance	5	10	15	20	25
PSNR Moving frame	37.08	33.19	31.05	29.61	28.45
SSIM Index Moving frame	94.66	89.43	84.78	80.89	77.11

Parameter \ Noise variance	5	10	15	20	25
$\mu$	0.0075	0.005	0.0045	0.004	0.004

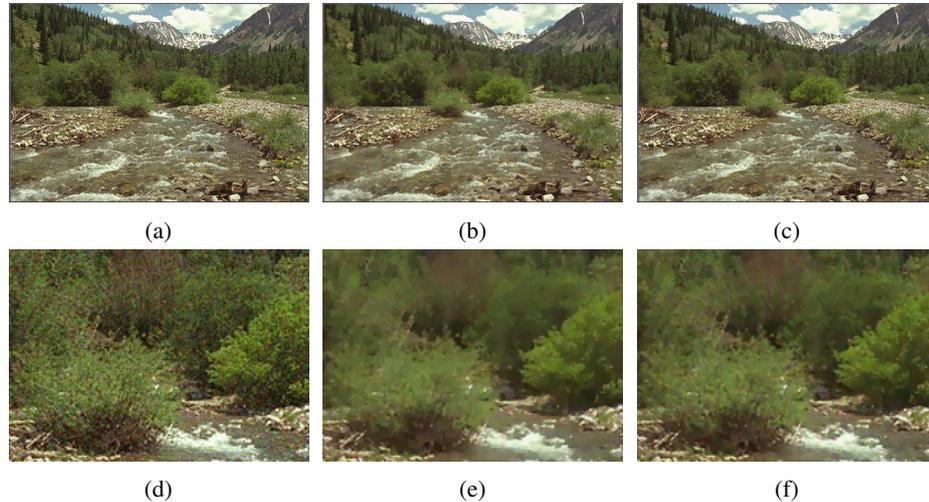


Figure 5.3: Example of our framework with VTV [Bresson and Chan, 2008] as denoising method. Row 1: (a) noisy image “kodim13” with  $\sigma = 15$ . (b) result of applying VTV to the image, PSNR=27.00. (c) our result, applying VTV to the components, PSNR=27.56. Row 2: zoomed-in details from the first row.

#### 5.4.2 The moving frame approach applied to the Non-Local Means algorithm.

In this section, we apply our framework to the Non-Local Means (NLM) algorithm of Buades et al. [Buades et al., 2005a], [Buades et al., 2011], defined for a multi-channel image  $I = (I^1, \dots, I^n)$ , by

$$I_{NLM}^k(p) := \frac{1}{C(p)} \sum_{q \in \mathcal{B}(p,r)} w(p,q) I^k(q) \quad (5.27)$$

where

$$C(p) = \sum_{q \in \mathcal{B}(p,r)} w(p,q)$$

and  $\mathcal{B}(p, r)$  denotes a neighborhood (patch) of the pixel  $p$  of size  $(2r + 1) \times (2r + 1)$ ; The weight function  $w(p, q)$  is of the form

$$w(p, q) = e^{\frac{-\max(d^2 - 2\sigma^2, 0)}{h^2}} \quad (5.28)$$

where  $h$  is a function of the noise level decreasing with the size of the patches, and  $d$  is a distance between patches, given by  $d^2(\mathcal{B}(p, f), \mathcal{B}(q, f)) =$

$$\frac{1}{n(2f + 1)^2} \sum_{i=1}^n \sum_{j \in \mathcal{B}(0, f)} (I^i(p + j) - I^i(q + j))^2 \quad (5.29)$$

In what follows, we describe our approach for both gray-level and color images and report the PSNR and SSIM index values.

### The case of gray-level images

The way we apply the moving frame approach in the context of the NLM algorithm for gray-level images is described as follows.

1. Take a clean image  $a$  and add Gaussian noise of variance  $\sigma$  to it to create the noisy image  $I$ .
2. Apply NLM to  $I$  using the code available online [Buades et al., 2011], obtaining a denoised image  $I_{NLM}$ .
3. Consider the moving frame associated to  $I$  (see (5.3)), for the parameter value  $\mu = 0.001$ , and compute the components  $(J^1, J^2, J^3)^T$  of  $I$  in this moving frame (see (5.4)).  
Then, apply the NLM algorithm in [Buades et al., 2011] **channel-wise** to these components, obtaining the denoised components  $(J_{NLM}^1, J_{NLM}^2, J_{NLM}^3)$ , from which a denoised gray-level image  $I_{NLM MF} := I_{NLM MF}^3$  is constructed (see formula (5.8)).
4. Compute the PSNR and SSIM index values of  $I_{NLM}$  and  $I_{NLM MF}$  with respect to the ground truth  $a$ .

The NLM algorithm is parametrized by the size of the window search (the parameter  $r$  in (5.27)), the size of the patches (the parameter  $f$  in (5.29)), and the function  $h$  in (5.28). In step 2, we test NLM with the default parameters that are given in [Buades et al., 2011] and which depend on the noise variance  $\sigma$ .

Regarding the parameters used in step 3, let us first point out that, in practice, the component  $J^2$  does not vanish identically, which comes from the fact that the numerical computation of the matrix  $P^{-1}$  in (5.4) does not correspond exactly to the transpose  $P^T$  of  $P$ . However, since the values of  $J^2$  are negligible, there is no need of denoising that component. Moreover, taking  $\mu$  small makes the component  $J^1$  have small values too. Hence,  $J^1$  has little weight in the denoising process, meaning that there is no need of optimizing the corresponding parameters, and we can then use the default parameters induced by the noise level  $\sigma$  for denoising the components  $J^1$ . Finally, the only parameter of our method is the noise level  $\sigma_3$  that will determine the parameters  $r, f, h$  used for denoising the component  $J^3$ .

Table 5.6 reports the average PSNR and SSIM index values over the Kodak database of both  $I_{NLM}$  and  $I_{NLMMF}$ , as well as the values of the parameter  $\sigma_3$  that have been used, and that were chosen to give the best PSNR results on average over the entire database for our approach. It is worth noting that, the optimal noise parameter  $\sigma_3$  is slightly greater than  $\sigma$  in every cases.

We can see that our approach is consistently better, for all noise levels, both in terms of PSNR and SSIM metrics. Note that the biggest improvement occurs when  $\sigma = 15$ .

### The case of color images

Unlike the VTV-based denoising method in sect. IV.A, the extension to color images is not trivial when dealing with NLM. Indeed, preliminary experiments showed that denoising a color image by applying our previous approach for gray-level images to each of its channels does not improve the results of directly applying NLM to the color image given by [Buades et al., 2011], regardless of the color space. We tested  $RGB, Lab, YUV, YC_rC_b$ , as well as a new color space

Table 5.6: Comparison of the standard approach and our moving frame approach with  $\mu = 0.001$  for *NLM*, at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter  $\sigma_3$  over the kodak database: the gray-level case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	37.41	33.38	31.05	30.04	28.91
PSNR Moving frame	37.52	33.59	31.57	30.12	29.00

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	94.96	88.71	82.17	80.34	75.94
SSIM Index Moving frame	95.11	89.54	85.37	81.03	76.95

Parameter \ Noise variance	5	10	15	20	25
$\sigma_3$	5.6	11	16	21	26

$A_{opp}$  introduced in Dabov et al. [Dabov et al., 2007]

$$\begin{pmatrix} A_{opp}^1 \\ A_{opp}^2 \\ A_{opp}^3 \end{pmatrix} := \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & -1/2 \\ 1/4 & -1/2 & 1/4 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (5.30)$$

where  $A_{opp}^1$  encodes the luminance information, and both  $A_{opp}^2$  and  $A_{opp}^3$  encode the chrominance information. Treating the color image in the *RGB* color space in a vectorial way by applying a 5D extension of *NLM* to the components  $(J^1, J^2, J^3, J^4, J^5)^T$  following the construction in sect. II.B, as we did for the *VTV*-based denoising method in sect. IV.A, fails when dealing with *NLM*. On the other hand, the space proposed in [Dabov et al., 2007] gave the most promising results.

To this end, we were able to improve the standard *NLM* algorithm by applying our moving frame approach only on the component  $A_{opp}^1$  of the color image and taking the components  $A_{opp}^2$  and  $A_{opp}^3$  from the image denoised directly with *NLM*. More precisely, our approach is the following one:

1. Take a clean color image  $a$  and add Gaussian noise of variance  $\sigma$  to it to

create a noisy image  $I$ . Compute its component  $A_{opp}^1(I)$  in the color space (5.30).

2. Apply NLM to  $I$  using the code available online [Buades et al., 2011], obtaining a denoised image  $I_{NLM}$ , and compute its chrominance components  $A_{opp}^2(I_{NLM})$  and  $A_{opp}^3(I_{NLM})$  in the color space (5.30).
3. Apply step 3 of the previous section to the gray-level image  $A_{opp}^1(I)$ , which produces a denoised image  $(A_{opp}^1(I))_{NLMMF}$ . Then, consider the image  $I_{NLMMF}$  whose components in the color space (5.30) are  $((A_{opp}^1(I))_{NLMMF}, A_{opp}^2(I_{NLM}), A_{opp}^3(I_{NLM}))$ .
4. Compute the PSNR and SSIM index values of  $I_{NLM}$  and  $I_{NLMMF}$  with respect to the ground truth  $a$  in the RGB color space.

The NLM algorithm in step 2 was applied with the default parameters for color images, that depend on the noise level  $\sigma$  and which can be found in [Buades et al., 2011], and the first component  $J^1$  of  $A_{opp}^1(I)$  in the moving frame in step 3 was denoised using the default parameters (for gray-level images) induced by the noise level  $\sigma$ . Finally, the only parameter we tune is the noise level  $\sigma_3$  for denoising the third component  $J^3$ .

Table 5.7 reports the average PSNR and SSIM index values over the Kodak database of both  $I_{NLM}$  and  $I_{NLMMF}$ , as well as the values of the parameter  $\sigma_3$  that have been used, and that were chosen in order to give the best PSNR results on average over the entire database for our approach. As in the gray-level case, our method outperforms the standard approach at each noise level, for both PSNR and SSIM metrics. It is also worth noting that unlike the gray-level case, the optimal value of the noise level  $\sigma_3$  is systematically lower than  $\sigma$ .

In Fig. 5.4, we illustrate our approach on the image in the database where the improvement of PSNR of our method with respect to the standard approach is one of the highest (0.41dB), and we compare this with the result of the standard NLM algorithm. We can see that our method better preserves fine details like the texture of the ground, for example, which is consistent with the fact that our method is aiming at preserving the local geometry of the processed image.

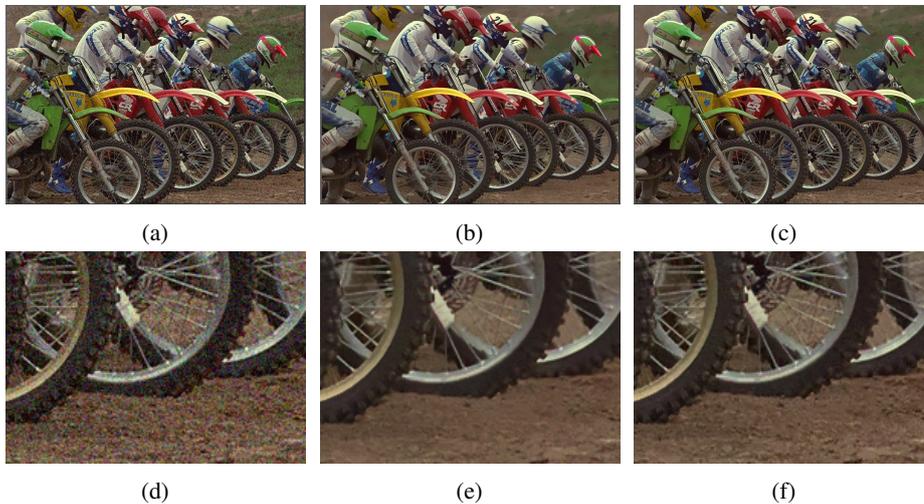


Figure 5.4: Example of our framework with NLM [Buades et al., 2005a] as denoising method. Row 1: (a) noisy image “kodim05” with  $\sigma = 20$ . (b) result of applying NLM to the image, PSNR=29.29. (c) our result, applying NLM to the components, PSNR=29.70. Row 2: zoomed-in details from the first row.

### 5.4.3 The moving frame approach applied to the Block Matching and 3D Filtering algorithm.

In order to emphasize the consistency of our framework for image denoising, we test it on one of the best algorithms available in the literature: the Block Matching and 3D filtering algorithm BM3D introduced by Dabov et al. [Dabov et al., 2007].

For both gray-level and color images, our approach follows exactly the one of the NLM case, only replacing the NLM algorithm in steps 2 and 3 by the BM3D algorithm whose code is available online [Lebrun, 2012]. More precisely, in step 2, we test BM3D with the default parameters determined by the noise level  $\sigma$ . In step 3, we apply BM3D on the component  $J^1$  with the default parameters induced by the noise levels  $\sigma_1 = \sigma$ , and we test several values of the noise levels  $\sigma_3$  when applying BM3D on the component  $J^3$ , in order to find the ones that provide the best average results on the whole database in terms of PSNR.

Results on gray-level images are reported in table 5.8 and results on color images are reported in table 5.9. In both cases, we can see that our approach is

Table 5.7: Comparison of the standard approach and our moving frame approach with  $\mu = 0.001$  for *NLM*, at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter  $\sigma_3$  over the kodak database: the color case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	38.35	34.84	32.70	31.18	30.00
PSNR Moving frame	38.76	35.18	33.08	31.61	30.42

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	96.01	91.92	87.74	83.74	79.93
SSIM Index Moving frame	96.32	92.69	88.85	85.04	81.08

Parameter \ Noise variance	5	10	15	20	25
$\sigma_3$	2.75	6.2	9.6	12.3	16

consistently better, though very slightly, for all noise levels, and for both PSNR and SSIM metrics. The increase in PSNR that we obtain, while modest, is in agreement with the optimality bounds estimated by Levin and Nadler [Levin and Nadler, 2011a], and Chatterjee and Milanfar [Chatterjee and Milanfar, 2010], and are comparable with the ones obtained through recent boosting techniques [Romano and Elad, 2015]. Finally, we notice that the optimal values of  $\sigma_3$  are systematically lower than the true noise level  $\sigma$ .

In Fig. 5.5, we show an example of our approach where we compare the result with the one of the standard BM3D denoising model. We can see that our method preserves better fine details like the wooden decoration engraved in the balcony, for example.

## 5.5 Conclusion

In this chapter, we have developed a framework that enables any denoising method to take more into account the local geometry of the image to be denoised by preserving the moving frame describing the graph of a scaled version of the image. Experiments with the VTV-based denoising method, NLM and BM3D algorithms on both gray-level and color images tested over the Kodak database

Table 5.8: Comparison of the standard approach and our moving frame approach with  $\mu = 0.001$  for *BM3D*, at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter  $\sigma_3$  over the kodak database: the gray-level case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	38.23	34.34	32.26	30.89	29.88
PSNR Moving frame	38.25	34.38	32.31	30.93	29.92

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	95.71	91.38	87.52	84.19	81.32
SSIM Index Moving frame	95.74	91.49	87.71	84.38	81.44

Parameter \ Noise variance	5	10	15	20	25
$\sigma_3$	4.9	9.7	14.4	19.1	23.9

Table 5.9: Comparison of the standard approach and our moving frame approach with  $\mu = 0.001$  for *BM3D*, at different noise levels. Average PSNR and SSIM Index (x100), and optimal parameter  $\sigma_3$  over the kodak database: the color case.

Approach \ Noise variance	5	10	15	20	25
PSNR Standard	40.35	36.50	34.32	32.83	31.72
PSNR Moving frame	40.38	36.53	34.36	32.88	31.77

Approach \ Noise variance	5	10	15	20	25
SSIM Index Standard	97.05	94.22	91.49	88.89	86.50
SSIM Index Moving frame	97.08	94.26	91.62	89.06	86.71

Parameter \ Noise variance	5	10	15	20	25
$\sigma_3$	2.75	5.6	8.2	11.1	13.8

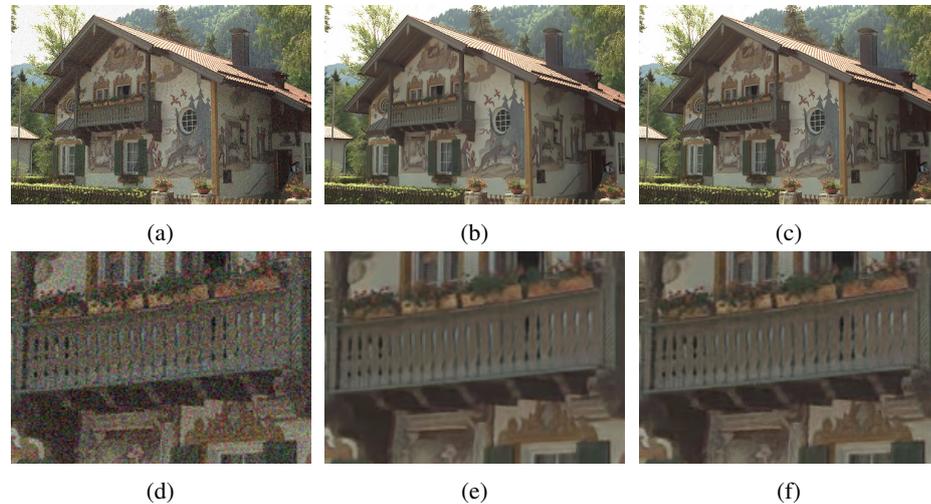


Figure 5.5: Example of our framework with BM3D [Dabov et al., 2007] as denoising method. Row 1: (a) noisy image “kodim24” with  $\sigma = 20$ . (b) result of applying BM3D to the image, PSNR=31.26. (c) applying BM3D to the components with optimized parameters, PSNR=31.40. Row 2: zoomed-in details from the first row.

showed that our strategy systematically improves the denoising method it is applied to, in terms of PSNR and SSIM metrics. The fact that we have been able to improve the performance of three denoising methods of different types: a local variational method, a patch-based method, and a method combining a patch-based approach with a filtering in spectral domain approach, demonstrates the consistency of our methodology.

In the proposed strategy for denoising, we either combine the components into a single vector-valued function to which we apply a denoising method (VTV) or treat them separately applying the same denoising method but with different parameters (NLM and BM3D).

## CHAPTER 6

---

### **Denoising: apply it on RAW images with actual real noise**

---

Most denoising methods in the literature are based on modelling noise as being additive and independent from the image data. While it is known that in reality this is not the case, its impact is minimized in the context of image denoising. In this chapter we show that not only the performance of algorithms decay under realistic models, but also that the ranking of denoising algorithms is different in the real noise scenario than when working under the AWG assumption. In order to evaluate the denoising methods under realistic conditions, we have created a dataset of images taken with a standard digital camera and corrupted with a realistic noise model estimated on RAW images. For realistic noise, we show how a local denoising method, applied to the RAW image, can outperform non-local methods applied to the camera output. Moreover, when low-complexity implementations of these non-local methods are applied to the RAW image, the results are comparable to those of the local method but the computational cost is higher. This suggests the possibility for a local denoising algorithm to replace more elaborate denoising methods for in-camera implementations.

This chapter is based on the work presented in the paper “Local denoising applied to RAW images may outperform non-local patch-based methods applied to the camera output” [Ghimpeteanu et al., 2016b] and its journal extension, recently submitted, with the title “On denoising photographs having actual noise”. We start by introducing a realistic signal-dependent noise model on RAW images and how we create the proposed test set of images. For this purpose, we also simulate the camera processing pipeline. We accentuate how important the noise

model is for image denoising with an experiment showing how several denoising methods perform differently under AWG and a realistic noise model. We continue by introducing a local denoising approach and several strategies that give good results when denoising RAW images. We underline how essential is to apply denoising as early as possible in the image processing pipeline, by showing how a simple local approach applied on the RAW can outperform two non-local patch-based denoising methods applied at the end of the image processing chain. We further include an experiment comparing local with low-complexity non-local denoising applied at the same stage in the image processing pipeline, with comparable results for a smaller running time.

## 6.1 Motivation

In the context of this chapter we will only be dealing with photographic images captured with digital cameras, and so we will equate “noise” with the actual noise that is always present in photos, as digital cameras have unavoidable physical and technological limitations.

As concluded in Chapter 4, current denoising methods produce such excellent results that actually raise the question: “Is denoising dead?” [Chatterjee and Milanfar, 2010].

But a key aspect that is overlooked when posing that denoising is an almost closed problem is the following: *most denoising methods in the literature are based on modelling noise as being additive and independent from the image data.* In fact, validations and comparisons are performed by taking clean photographs as ground truth, creating noisy versions with Additive White Gaussian (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 the Peak Signal-to-Noise Ratio (PSNR). *Everything is dependent on this AWG assumption:* the design of the denoising algorithms, their ranking according to quality of the outputs, even the computation of the optimality bounds that suggest that state-of-the-art algorithms are close to optimal. Furthermore, while it’s well-known that camera noise is not AWG [Faraji and MacLean, 2006, Foi et al., 2008, Healey and R., 1994], it’s implicitly supposed that this difference should not have an impact in how we address the denoising problem, nor in how results are validated or

methods compared.

Contradicting this notion, a recent work [Seybold et al., 2014] shows how the performance of denoising methods decays drastically when using realistic noise models instead of AWG noise. The first contribution of this chapter will be to highlight that not only the performance, but also the ranking of denoising algorithms is different in the real noise scenario than when working under the AWG assumption.

Regarding camera noise in RAW images, several works propose noise models that follow a distribution combining Poisson and Gaussian models of signal-dependent variance, and if the photon count is sufficiently large then the Poisson process can also be approximated as a signal-dependent Gaussian distribution [Trussell and Zhang, 2012]. Taking then that RAW image noise can be modelled by a Gaussian distribution with variance proportional to image intensity, a suitable transform can be applied to the image so that the noise becomes AWG [Donoho, 1993, Mäkitalo and Foi, 2011], and therefore any regular denoising method can be used in the RAW case (afterwards, the inverse transform must be applied to the denoised result). The second contribution of this chapter will be to show how a local denoising method, applied to the RAW image, can outperform non-local methods applied to the camera output. Furthermore, when low-complexity implementations of these non-local methods are applied to the RAW image, results are very much comparable visually to those of the local method but require a considerably higher computational cost, suggesting the possibility for a local denoising algorithm to replace more elaborate denoising methods for in-camera implementations.

Finally, our third contribution is to provide in [<http://ip4ec.upf.edu/realisticnoise>, 2017] a publicly available database of clean and noisy versions of photos, in formats corresponding to the sensor image (12 bit RAW) and the non-linear and uncompressed camera output (8 bit PNG), so that researchers can develop and test denoising algorithms for real-case scenarios. With examples from this database we will show in this paper how the noise for real photographs is visually very different from AWG.

While on RAW images each channel can be treated independently for estimating the noise model, this is not the case for the camera output image, due to several steps of the camera image processing pipeline. Nam et al. [Nam et al., 2016] propose a cross-channel noise model for JPEG images, that takes into account

the correlation between the color channels. They show that the linear connection between the noise variance and image intensity on RAW is mostly kept by white balancing and demosaicking, but it is lost after camera image processing steps like color correction or tone mapping. Moreover, the JPEG compression has a huge impact on the noise characteristics. The compression is done typically by dividing the image into 8x8 patches that are processed individually, thus compression influences the noise according to the characteristics of each patch. While for the RAW image the noise simply depends on the pixel intensity value, for JPEG image it depends both on the pixel value and the patch that surrounds it. The JPEG noise model is extremely complex. The authors propose an approach involving a neural network to estimate the noise model and show that using it for denoising leads to an improvement (both visually and in terms of PSNR and SSIM index values) in the output quality.

Plötz and Roth [Plötz and Roth, 2017] propose a database of real noise photographs and their corresponding ground truth. They capture a serie of images of the same scene, with different ISO values and exposure times. The reference is the photograph taken with low ISO, that shows almost no noise. The ground truth is created by a post-processing step that corrects for differences in illuminations and minor displacements (of objects in the scene or camera shake) between several exposures. They compare the BM3D denoising method to several state-of-the-art algorithms for AWG noise denoising. They found that when applied on real noise (on 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. Compared to the database proposed in [Plötz and Roth, 2017], ours includes an estimated RAW noise model that can be added to a RAW ground truth image that does not necessary belong to our database and to analyze different noise level appearances on a variety of images. Another possible application could be to add subtle noise given by our model, instead of the common AWG, to extremely clean studio photographs to avoid a fake-appearance and increase perceived sharpness, and to image images rendered or processed with computer graphics techniques, for a more realistic look (common techniques used in photography and computer graphics, described in Section 3.4).

Although most denoising algorithms in the literature are applied on the camera output images, there is a line of research of denoising applied to RAW images, for which we give several examples in the following. The BM3D denoising method has been adapted to denoise RAW images in [Danielyan et al., 2009] providing

great results. Another non-local patch-based denoising method, NLM, has been applied to RAW images in [Lee et al., 2017]. Though the non-local methods output denoised images of excellent quality, they share the drawback of being computationally too intensive for camera implementations (without significant modifications). With a machine learning approach, Khashabi et al. proposed in [Khashabi et al., 2014] a simultaneous demosaicking and denoising method that offers image results with a good performance on several cameras. However, this method also is computationally intensive. Due to its low computational cost, we propose to use a local TV-based approach adapted to a realistic signal-dependent noise model.

## **6.2 Estimate a realistic noise model on RAW images and create a test set**

To reproduce a realistic scenario for the task of image denoising, we create a signal-dependent noise model estimated at the RAW level, described in this section. For processing the image from the RAW format to the camera output, we perform a simulation of the camera processing pipeline, as camera makers do not describe the steps involved. We continue with introducing the noise model and its visual validation. Finally, we show in this section how AWG noise is visually different than real actual noise.

In order to evaluate and compare denoising methods, we create a test image set containing clean and noisy images. For this, we capture a clean image and add noise to it in order to create a noisy image. In this way, we have a noisy image and the corresponding clean reference image, which allows us to compute an objective image quality metrics like PSNR.

Our test set consists of 20 clean images presented in Fig.6.1, taken with a Nikon D3100 in the RAW format. We captured each image scene trying to obtain the best exposed photograph and minimize the unavoidable noise acquisition (by fixing the ISO sensitivity to the value to 100).

### **6.2.1 Simulate the camera processing pipeline**

The basic steps common to standard cameras are detailed in Section 2.2. In the following, we show a concise enumeration of the steps that we use:

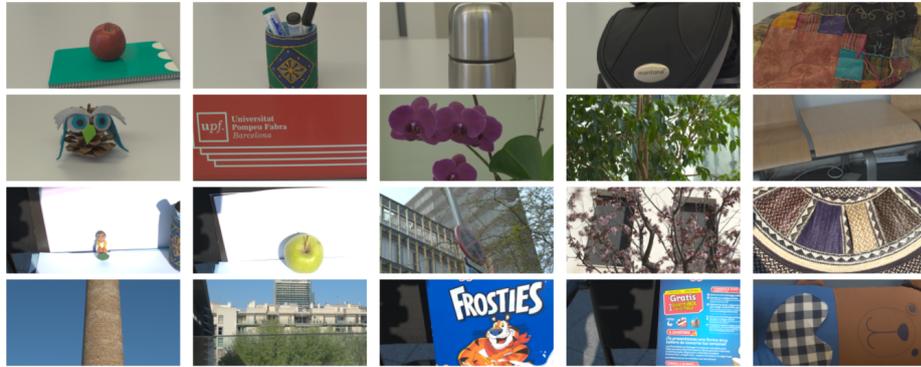


Figure 6.1: Our image test set.

1. **Capture.** We conduct our experiments on images captured with a Nikon D3100 camera. A photo taken with this camera can be saved in the RAW format NEF of Nikon cameras as a 12-bit depth image, obtaining the CFA (color filter array) RAW data with a Bayer mosaic pattern. An image example is illustrated in Fig.2.3a.
2. **White-balance,** which 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. Fig.2.3b depicts the white-balanced image example.
3. **Demosaicking** is an interpolation process which outputs a 3-channel image, as exemplified in Fig.2.3c. 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); this step estimates the other two missing values. In our experiments, we include in the camera processing pipeline the local demosaicking algorithm proposed by Malvar et al. [Malvar et al., 2004], 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,** which makes the conversion from the camera color space to sRGB (standard RGB) color space, as illustrated in Fig.2.3d.

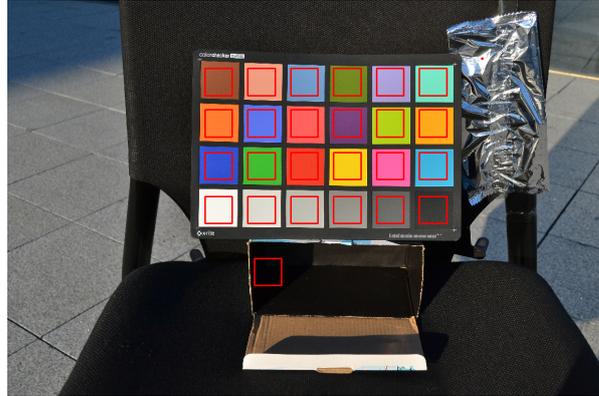


Figure 6.2: The Colorchecker set-up: an image of our noise estimation set-up captured with ISO 800, with marked regions used for estimating the noise model.

5. **Gamma Correction**, consisting of raising the (normalized) image values to the power of  $1/2.2$ , the standard value. An image example is shown in Fig.2.4. This step assures an optimized encoding that models the non-linear human perception of luminance: more sensitive to details in darker areas.
6. **Quantization**. The final step of the pipeline (for our purposes) represents quantizing the image from 12-bit depth to 8-bit depth, getting as an output an RGB image ready for display, as in Fig.2.4.

### 6.2.2 The noise model

Hereby we give a detailed description of how we construct the realistic signal-dependent noise model, estimated on the RAW image. As a model for RAW image noise, we use the Gaussian distribution with variance proportional to image intensity. Therefore, we need to estimate the noise variance corresponding to each intensity value. For this task, the RAW Colorchecker image is ideal, as it offers homogeneous color patches from which one can compute variance as a function of the mean intensity value of each patch. We start by capturing a RAW photograph of our noise estimation set-up, as exemplified by Fig.6.2. Thus, for each clean image from the test set we can add noise with properties obtained from this image, as described as follows.

Following the line of experiments of [Healey and R., 1994], [Seybold et al., 2014], [Trussell and Zhang, 2012], we analyze the RAW Colorchecker photograph by segmenting and extracting all its 24 color patches, and computing the noise variance in each color square and for each RAW color channel. The result for each channel is a set of 24 pairs of mean and variance, computed for each of its color patch. Tests on several RAW images of the Colorchecker, taken with different camera settings, allow us to conclude, as in [Seybold et al., 2014] and [Trussell and Zhang, 2012], that the variance as function of the mean can be fitted by an increasing linear function, showing that the noise is signal-dependent.

To add noise to a clean RAW image, we add white Gaussian noise with the variance given by the noise curve for each pixel intensity value of the clean RAW image. Afterwards, we apply the rest of the camera pipeline: white balance, demosaicking, color correction, gamma correction and quantization to 8-bit depth.

In practice, applying Gaussian noise with variance given by this linear model gives visually realistic noise, except for the extreme cases: very dark regions in the shadow and very bright regions associated to specular highlights (bright spots of light produced on shiny illuminated surfaces). Notice that the black patch in the Colorchecker image shown in Fig.6.2 does not offer the lowest intensity value in the scene. The same holds for the white patch, which does not give the highest intensity value in the scene. To have a noise model that better takes into account the extreme values, we introduce two objects in the Colorchecker scene. The first one is a paper box with the interior present in the shadow and painted with black matte paint. The second one is an aluminium foil that receives direct light and creates specular highlights. We compute the noise variance for a crop from the black box and also a small crop from the area with specular highlights. In Fig.6.2 we mark in red all the patches for which we compute the variance.

Finally, we estimate a noise model from the 26 pairs of mean and variance. Interestingly, the variance in the small specular highlight region is zero for each color channel. To take this into account, we consider a linear noise model that decreases after the white patch mean value, and becomes 0 after the specular highlight mean value. An example of the channel-wise variance plot as a function of mean pixel value is shown in Fig.6.3.

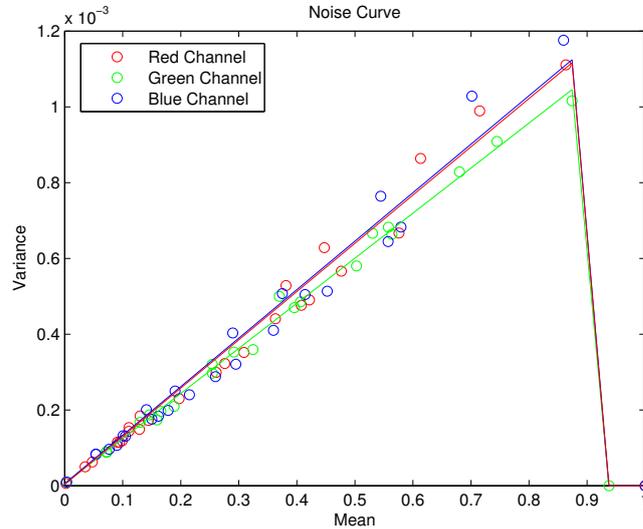


Figure 6.3: Plot (example for one set of fixed camera parameters) of variance as function of the mean, for a RAW image scaled between 0 and 1. Values extracted from a RAW Colorchecker image taken with ISO 3200. Dots show the real values obtained for each color square and each channel, while the continuous lines show the fitted linear functions.

### 6.2.3 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 offers a better preservation of the 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 (ISO 100, 200, 400, 800, 1600 and 3200) produced by our camera. Table. 6.1 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.

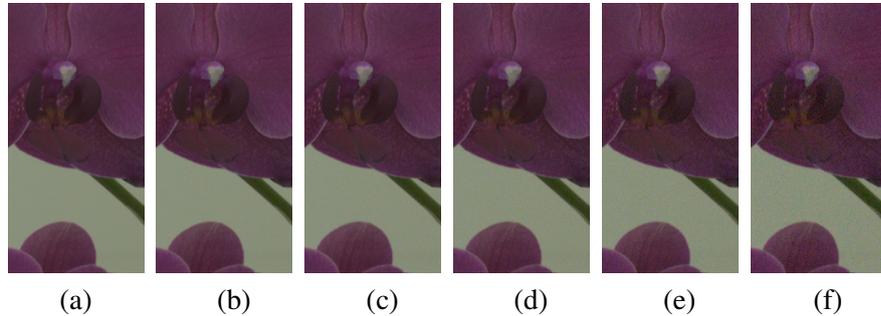
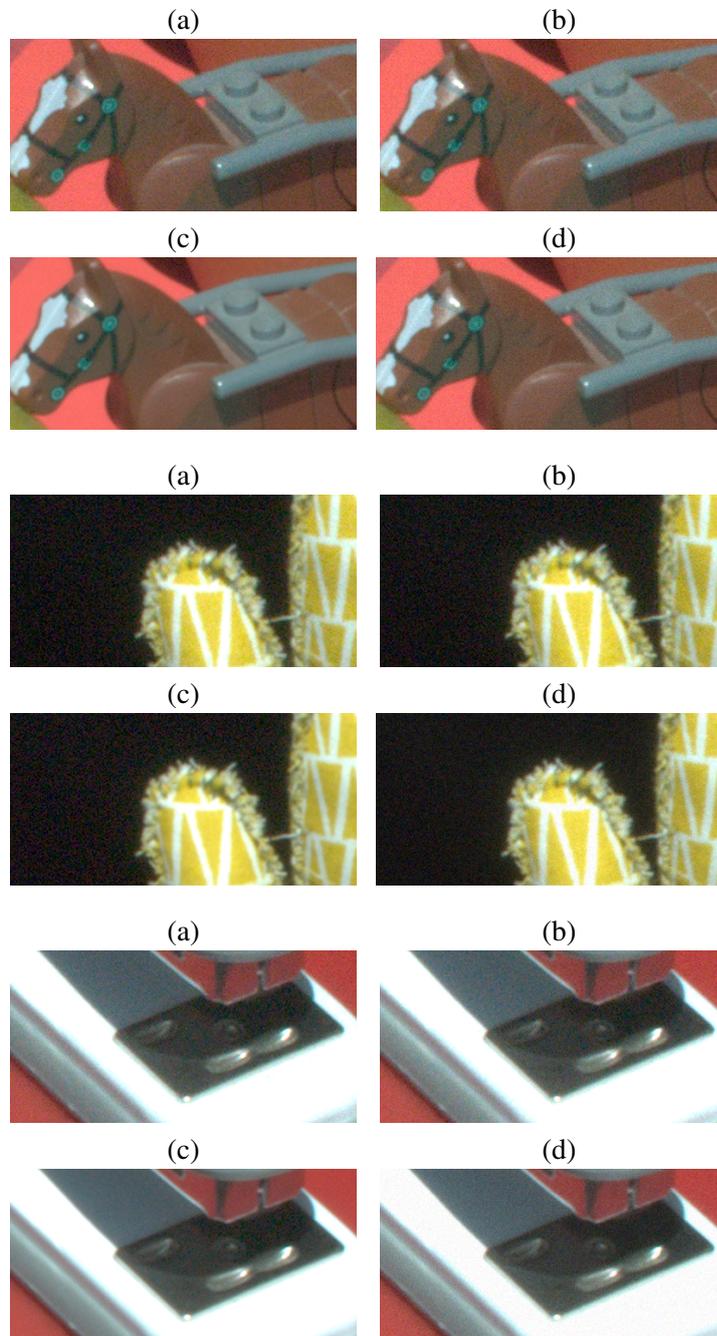


Figure 6.4: Crops from a noisy image example from our test set. From left to right: (a) original image, (b) to (g) synthesized noisy images obtained with the noise curve associated to: (b) ISO 100 ( $\sigma=2.42$ ), (c) ISO 400 ( $\sigma=3.17$ ), (d) ISO 800 ( $\sigma=3.95$ ), (e) ISO 1600 ( $\sigma=5.43$ ), (f) ISO 3200 ( $\sigma=7.98$ ).

Fig.6.4 illustrates a crop from an image example from our database, the original image and noisy images created with the realistic noise model aforementioned and followed by applying the camera processing pipeline.

Fig.6.5 represents a validation of our noise model on an image example taken with ISO 3200, which produces the highest noise level. We start by capturing a serie of 30 photographs taken in identical conditions, and average them to obtain a clean image. We compare one of these 30 real noise images to a noisy synthesized output obtained by adding to the clean RAW photograph Gaussian noise with variance given by our realistic noise model illustrated in Fig.6.3, and show that they are visually similar. We also compare against two other methods for creating noisy images: adding Gaussian noise of constant variance to the RAW clean image (which has the problem that darker regions appear very noisy, while brighter regions appear too clean) and adding Gaussian noise of constant variance to the camera output (where the noise has a uniform but unrealistic appearance); for both cases, we choose variance values such that visually they produce results as close as possible to the noise level of the real noise image.



101

Figure 6.5: Examples of comparison between an actual real noise photograph (a) and synthesized noisy images obtained by: (b) adding Gaussian noise with variance given by our realistic noise model to the RAW image, (c) adding Gaussian noise of constant variance to the RAW image and (d) adding Gaussian noise of constant variance to the camera output.

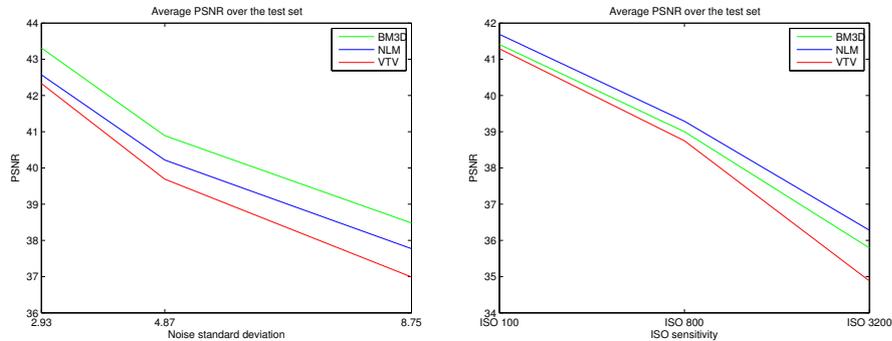


Figure 6.6: Comparison of BM3D, NLM and VTV applied on the camera output, under two noise models. Average PSNR value plots of denoising applied on noisy images created with additive white Gaussian noise (left) and our realistic noise model (right).

### 6.3 Testing denoising methods under different noise models: AWG vs. realistic

In this section we present an experiment that shows how essential is the noise model for evaluating denoising methods. For this, we compare three denoising methods applied on camera output images created with two different noise models. We use the patch-based NLM and BM3D denoising methods, with the code available online [Buades et al., 2011], respectively [Lebrun, 2012], and the local VTV-based denoising method.

This latter method, proposed by Blomgren and Chan [Blomgren and Chan, 1998], is a vectorial extension of the channel-wise TV-based denoising and consists in replacing the gradient operator acting on each channel by the Jacobian operator acting on the whole image:

$$I_{t+dt} = I_t - dt \mathcal{J}^* \left( \frac{\mathcal{J}(I)}{\sqrt{\|\mathcal{J}(I)\|^2 + \epsilon}} \right), I_{|t=0} = I_0 \quad (6.1)$$

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

The parameters of these algorithms are the standard deviation of the noise,

Table 6.1: 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

in the case of NLM and BM3D, and the number of iterations for the VTV-based denoising. We 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. We start with a clean RAW image  $I_{cleanRAW}$  and add Gaussian noise, with variance given by the associated noise curve as detailed in Section 6.2, 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  $I_{VTV}$ .
2. On the clean camera output image  $I_{clean}$  we add white Gaussian noise (AWG) to obtain a noisy image  $I_{awgnoisy}$ ; the noise level used is explain in the following paragraph. Apply NLM, BM3D and VTV-based denoising methods on  $I_{awgnoisy}$  to obtain the denoised images  $I_{awgNLM}$ ,  $I_{awgBM3D}$  and  $I_{awgVTV}$ .

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.6.6. The left plot shows on the horizontal axis the average noise standard deviation computed in the 8-bit depth noisy images over the database. We have chosen comparable noise levels for both experiments, as seen from the average standard deviation values shown in Table. 6.1. 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. Also, for each denoising method, there is a large drop in the PSNR value from denoising the AWG noise images to realistic noise images. Applied on the camera output, the local denoising method gives worse results in terms of PSNR than the non-local patch-based methods, for both noise models. Fig.6.7 illustrates an image example

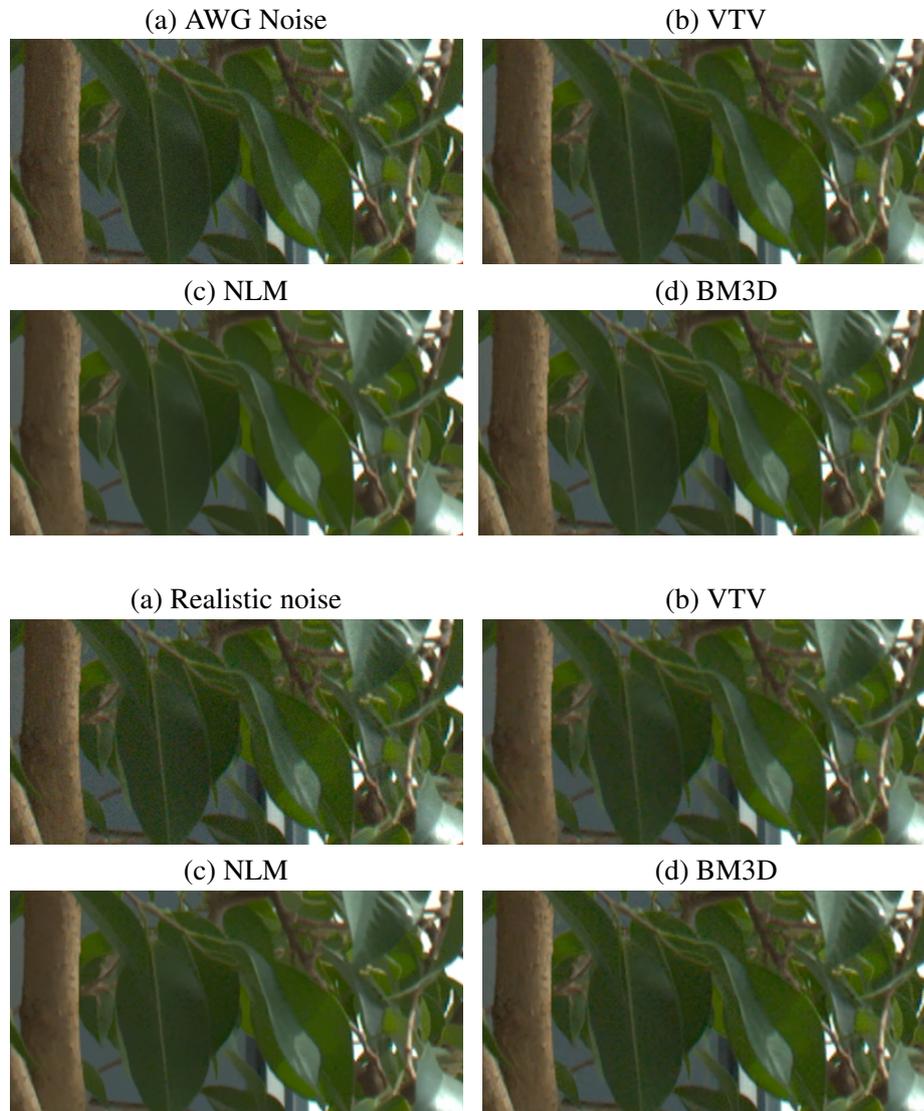


Figure 6.7: Comparison of VTV, NLM and BM3D denoising methods under AWG on camera output and a realistic noise model. Rows 1-2. (a) crop from AWG noise image “image9” with  $\sigma = 4.96$ . (b) VTV, PSNR=37.66. (c) NLM, PSNR=38.37. (d) BM3D, PSNR=38.92. Rows 3-4. (a) crop from realistic noise image “image9” with  $\sigma = 5.67$  and ISO 800. (b) VTV, PSNR=35.39. (c) NLM, PSNR=35.89. (d) BM3D, PSNR=35.72.

of this experiment. Denoising with BM3D an image created with AWG noise gives an excellent output, while for a realistic noise image it produces blocking artifacts on the leaf in the shadow.

## 6.4 A local denoising method for RAW images with realistic noise

### 6.4.1 The denoising method

As seen in Section 6.3, a simple local method like VTV-based denoising cannot outperform a non-local patch-based method applying it on the camera output. We propose applying it at some stage in the camera processing pipeline, more precisely on the demosaicked RAW.

In [Ghimpeteanu et al., 2016b], the denoising method we applied on noisy demosaicked RAW images was a simple iterative procedure reducing the TV of the images. More precisely, given an *RGB* image  $I_0 = (I_0^1, I_0^2, I_0^3): \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ , we considered the evolution equation

$$I_{t+dt}^k = I_t^k - dt \nabla^* \left( \frac{\nabla I^k}{\sqrt{\|\nabla I^k\|^2 + \epsilon}} \right), I_{|t=0}^k = I_0^k \quad (6.2)$$

for  $k = 1, 2, 3$ , where  $\nabla$  and  $\nabla^*$  are respectively the gradient and the divergence operator, and  $\epsilon$  is a small positive constant used to avoid division by 0.

Letting the Eq. (6.2) run until convergence will produce a flat image, as the algorithm is the gradient descent associated to the minimization of the convex energy

$$E(I^k) = \int_{\Omega} \sqrt{\|\nabla I^k\|^2 + \epsilon} d\Omega.$$

Hence, we stopped the iterative procedure after a certain number of iterations, chosen empirically for each noise level in order to optimize the mean PSNR of the camera output images over the database we considered. The average value of the PSNR index computed over the images from our test set denoised by the TV-based denoising algorithm (6.2) are included in Fig.6.8.

Hereby, we extend our approach in [Ghimpeteanu et al., 2016b] by replacing the channel-wise iterative procedure (6.2) with its vectorial extension from Eq.

(6.1), following the model proposed by Blomgren and Chan [Blomgren and Chan, 1998].

#### 6.4.2 Adapting a TV-based denoising method to the signal-dependent noise model

The TV-based denoising methods were proposed in the context of images corrupted by additive white Gaussian noise assumed to be signal-independent. As shown previously, our database images exhibit a realistic signal-dependent noise.

In the context of a signal-dependent noise model, there are two possible denoising techniques [Mäkitalo and Foi, 2013]. The first one is to adapt a certain denoising method to treat the particular noise model properties. For example, Luisier et al. propose in [Luisier et al., 2011] a methodology to adapt transform-domain thresholding algorithms for the mixed Poisson-Gaussian noise model. The second technique is, instead of adapting the denoising method to a certain noise model, to apply a variance stabilizing transformation (VST) created for the particular noise model, such that signal-dependency is removed and the noise variance becomes constant over the image. After applying a VST, 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 Gaussian noise is an extremely popular topic, that produced many algorithms over the last decades. Mäkitalo and Foi suggest in [Mäkitalo and Foi, 2013] that instead of designing different denoising methods for each particular noise distribution, one better uses a well designed VST and a suitable inverse.

For denoising images corrupted by signal-dependent noise, Donoho was the first to propose the Anscombe transform [Anscombe, 1948] as VST for image denoising in [Donoho, 1993]. Mäkitalo and Foi [Mäkitalo and Foi, 2011] also used the Anscombe transform to remove the signal-dependency, and accentuated the importance of applying a suitable inverse. Following this line, we apply the Anscombe transform  $f_{Anscombe}$  on the demosaicked noisy RAW image  $I_{noisyDRAW}$ :

$$f_{Anscombe}(I_{noisyDRAW}) = 2\sqrt{I_{noisyDRAW} + \frac{3}{8}} \quad (6.3)$$

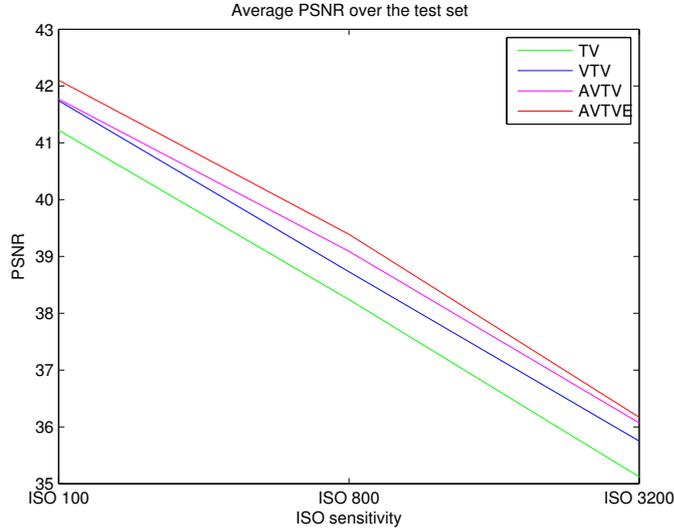


Figure 6.8: Evolution of our TV-based local denoising experiments, under the proposed realistic noise model: average PSNR values computed over our image test set.

and denoise the image  $f_{Anscombe}(I_{noisyDRAW})$  with VTV, instead of denoising  $I_{noisyDRAW}$ ; 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 [Mäkitalo and Foi, 2011] to  $D$ :

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

We denote the resulting image by  $I_{denDRAW}$  and we call AVTV this method combining VTV with the Anscombe transform. Fig.6.8 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 Eq.(6.1).

### 6.4.3 Refine the denoising output by recovering lost details

Even the best denoising algorithms can benefit from the so-called “boosting” techniques [Romano and Elad, 2015]. One of these techniques adds back to the image some content from the residual (difference between noisy and denoised image). This is justified by the fact that denoising is an imperfect process that eliminates not only noise but small details as well. While signal leftovers can be kept in the residual, the opposite is also true: noise is kept in the denoised image. Another type of boosting method tries to eliminate the noise kept in the denoised image, producing 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. With the removed noise, our local algorithm removes also important details. As a boosting technique, we propose to add back to the denoised image some of the useful information that was removed:

$$I_{AVTVE} = (1 - a)I_{noisyDRAW} + aI_{denDRAW}, \quad (6.4)$$

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

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 intact the pixel intensity values of the denoised image, so the value of  $a$  should be large; on image contours, on the other hand, 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 to use for estimating  $a$  a local edge indicator, like the Charbonnier diffusivity function [Charbonnier et al., 1994]:

$$f_{Charbonnier} = \frac{1}{\sqrt{1 + \frac{\|\nabla L(\widehat{I_{denDRAW}})\|^2}{\lambda}}} \quad (6.5)$$

where  $L$  denotes the luminance component, and  $\lambda > 0$  is a contrast parameter related to edge localization. The image  $\widehat{I_{denDRAW}}$  is obtained by finding the number of iterations in the iterative scheme introduced in Eq.(1) 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:

$$a = \frac{1 + f_{Charbonnier}}{2}. \quad (6.6)$$

Experiments show that both the step of applying the Anscombe transform before denoising and its inverse after, and the step of refinement described by Eq.(3), bring an improvement (both in terms of PSNR and visually) compared to only denoising with the iterative scheme introduced in Eq.(1), as seen in Fig.6.8 and Fig.6.9. Also, we can confirm with these figures that the vectorial VTV-based denoising approach improves the channel-wise TV-based denoising strategy that was used in [Ghimpeteanu et al., 2016b].

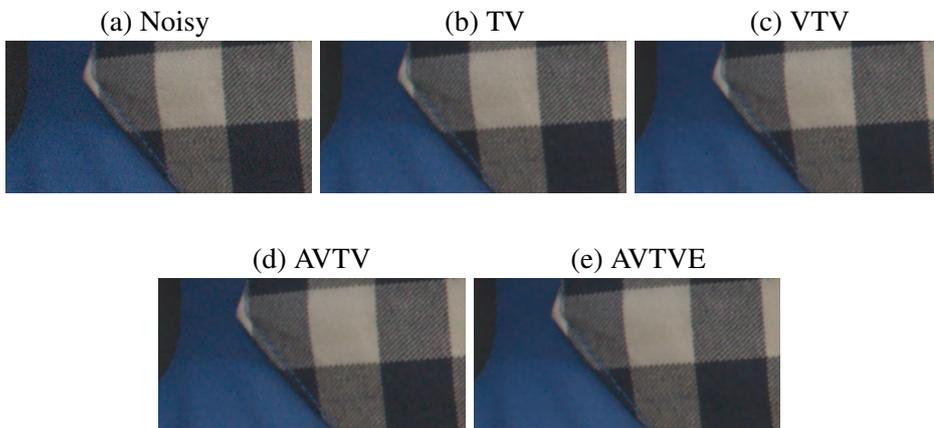


Figure 6.9: Comparison of our local TV-based denoising methods applied on the demosaicked RAW, under the proposed realistic noise model. (a) crop from noisy image “image20” with  $\sigma = 4.84$  and ISO 800. (b) TV, PSNR=37.14. (c) VTV, PSNR=37.38. (d) AVTV, PSNR=37.45. (e) AVTVE, PSNR=37.73.

## 6.5 Compare local denoising on RAW to non-local denoising on camera output

In this section we present an experiment showing that a local denoising method applied early in the camera pipeline may outperform non-local denoising of the

camera output. We compare our local denoising approach AVTVE applied on the demosaicked RAW noisy image, with two non-local 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  $I_{cleanRAW}$  and add Gaussian noise, with variance given by the associated noise curve as detailed in Section 6.2, to obtain a noisy image  $I_{noisyRAW}$ .
2. 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  $I_{noisy}$ .
3. Apply non-local patch-based denoising methods (NLM and BM3D) on  $I_{noisy}$  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 on  $I_{noisyRAW}$  and obtain  $I_{noisyDRAW}$ . Afterwards denoise with our local method AVTVE, detailed in Section 6.4, followed by applying color correction, gamma correction and quantization to 8-bit, to output our denoised image  $I_{OUR}$ . The denoising parameters, described in the following, are optimized so as to maximize the PSNR of  $I_{OUR}$ .
5. Evaluate the images  $I_{BM3D}$ ,  $I_{NLM}$  and  $I_{OUR}$ , visually and with respect to objective metrics.

We tested the denoising methods on our test set. The fixed parameters of our method are the small positive constant  $\epsilon = 10^{-6}$  and the time step  $dt = 0.001$ . The parameters that we optimize are the number of iterations and  $\lambda$  from Eq.(4). The parameter  $\lambda$  is optimized for the whole database (for each noise level). Finally, given a noise level and an image, we stop our algorithm after a certain number of iterations that maximizes the PNSR value of  $I_{OUR}$ . For each noise level of our test set, the number of iterations is similar for all the images. This suggests the possibility for camera makers to develop an automated method for setting the optimal number of iterations, given the camera parameters. The non-local patch-based methods that we apply at Step 3 on the image  $I_{noisy}$  are the NLM

and BM3D algorithms, whose implementations are available online [Buades et al., 2011], respectively [Lebrun, 2012]. The default parameters of these models are determined by the standard deviation of the noise. As we did for the local method, we optimize the parameters for each image and noise level of the database, choosing the ones that maximize the PSNR values of the denoised results  $I_{NLM}$  and  $I_{BM3D}$ .

The plot of Fig.6.10 shows the average PSNR values over our proposed image dataset, for each noise level and each denoising strategy aforementioned. The plots show on the horizontal axis the average noise standard deviation computed in the 8-bit depth noisy images over the database. 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 noise 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.6.11, where the images are denoised with the optimal parameters described above. For a better comparison we include images that show the difference between the clean and denoised image with each method, on the Row 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. In the crop images from the first two rows, we can see that for a small noise level the denoising methods BM3D, NLM and AVTVE produce comparable results. On the third and fourth row, all images preserve the apple texture; while our result has a cleaner appearance than the NLM and BM3D denoised images that exhibit small blocking artifacts. On the bottom two rows, for the highest noise level investigated, the BM3D output reveals strong blocking artifacts in the homogeneous area, while our image result and NLM have a cleaner appearance.

Table 6.2 shows the average running time for the *AVTVE*, *NLM* and *BM3D* methods for a 1000x2000 color image from our test set, on a i7-4770 CPU with 3.4 GHz and 8 cores. At a fraction of the running time of *NLM* and *BM3D*, our method, although not optimized for speed, produces results that are better or comparable both visually and in terms of PSNR.

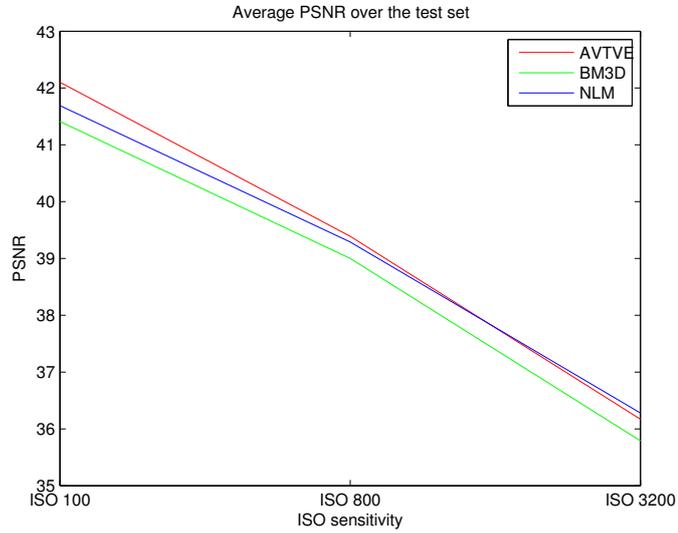


Figure 6.10: Comparison between the local denoising method AVTVE applied on the demosaicked RAW to the NLM and BM3D denoising algorithms applied on the camera output, under the proposed realistic noise model. Average PSNR values computed over our image test set.

Table 6.2: Average running time (sec) for one 1000x2000 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

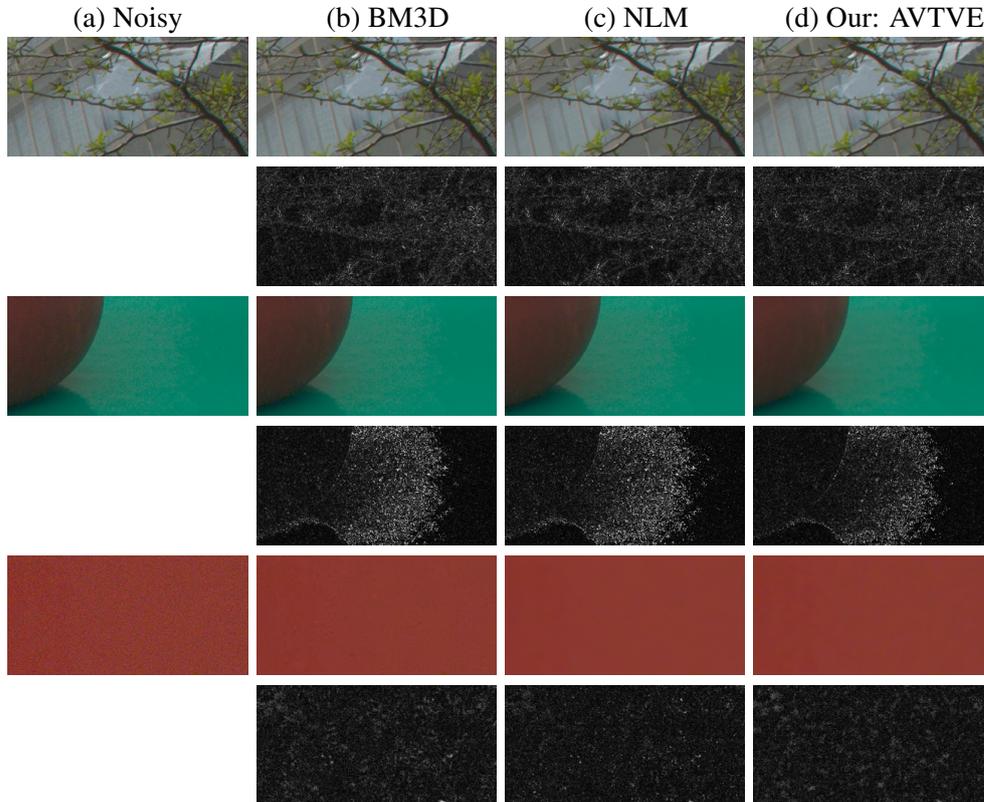


Figure 6.11: Comparison of the local denoising method AVTVE applied on the demosaicked RAW to the BM3D and NLM denoising methods applied on the camera output, under the proposed realistic noise model. Row 1: crop from noisy image “image13” with  $\sigma = 4.11$  and ISO 800, BM3D result with PSNR=36.79, 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.

## 6.6 Analysis of denoising experiments

We conducted a serie of experiments that show the importance of the precise location in the camera processing pipeline at which we insert a denoising method.

### 6.6.1 Apply the local denoising method at different locations in the camera processing pipeline

We showed in [Ghimpeteanu et al., 2016b] that for Gaussian noise with constant variance, the advantage of working directly on a demosaicked RAW image, is due to the fact that we apply the local denoising algorithm before the gamma correction and the quantization steps. Each step in the camera processing pipeline alters the noise and makes it more complex than the predicted piecewise linear model at the RAW level. Therefore, ideally, one should apply denoising directly on the RAW image. We investigate this strategy in the following.

The challenge for a local method is that the neighbouring information comes from different color channels. A solution is proposed by Park et al. [Park et al., 2009]. The authors suggest to apply on the 4-channel image  $RG_1G_2B$  (constructed from the CFA image) an orthonormal transformation that changes the color space. The orthonormal transformation from  $RG_1G_2B$  to the proposed color space is given by:

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ \Delta G \end{pmatrix} = \begin{pmatrix} 0.541 & 0.436 & 0.436 & 0.572 \\ -0.794 & 0.107 & 0.107 & 0.588 \\ -0.274 & 0.546 & 0.546 & -0.572 \\ 0 & 0.707 & -0.707 & 0 \end{pmatrix} \begin{pmatrix} R \\ G_1 \\ G_2 \\ B \end{pmatrix} \quad (6.7)$$

where the axes are based on a principal component analysis of pixel RGB values estimated on the Kodak dataset [<http://r0k.us/graphics/kodak/>, 2017]. The orthonormal property guaranties that the sensor noise keeps the same distribution in both color spaces. Afterwards, a grey-image denoising method can be applied on each channel of the 4-channel image  $C_0C_1C_2\Delta G$ , followed by the inverse of the transform. An image example is shown in Fig.6.12. The result of applying directly AVTVE on the demosaicked image is better in terms of PSNR, than denoising with AVTVE each of the channels of  $C_0C_1C_2\Delta G$ . Also, visually, the method AVTVE applied on the demosaicked image produces a sharper result that better preserves the fine details in the apple texture. Therefore, we choose



Figure 6.12: Left: noisy image. Middle: AVTVE applied on the demosaicked RAW, PSNR=43.80. Right: AVTVE applied on 4-channels RAW image, PSNR=42.33.

to apply our local denoising after demosaicking, on an RGB image, instead of applying denoising on the RAW image transformed into another color space. We think that for a simple method like VTV-based denoising, whose performance relies heavily on an accurate estimation of the local geometry, the disadvantage of having neighbours of different color channels, if applied on RAW, has a larger weight than the disadvantage of applying it after demosaicking, when noise characteristics might have changed but the color channels now have a very similar geometry [Caselles et al., 2002].

### 6.6.2 Compare local with low-complexity non-local denoising applied at the same stage of the image processing chain

As non-local 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  $10 \times 10$ . This choice of parameters reduces the cost of finding similar patches, which is the most complex and time-consuming task of a non-local method.

As the BM3D method is designed to treat Gaussian noise, we apply the Anscombe transform introduced in Eq.(2) before denoising and its inverse after, like we did for our denoising method. We denote this approach by ABM3D.

As in the case of our local approach, we also consider the refinement step described by Eq.(3), producing a result denoted ABM3DE. This step increases the computation time, but shows the best PSNR index value.

Both the steps of applying the Anscombe transform and the refinement step produce image results that are better compared to the BM3D output, both objec-

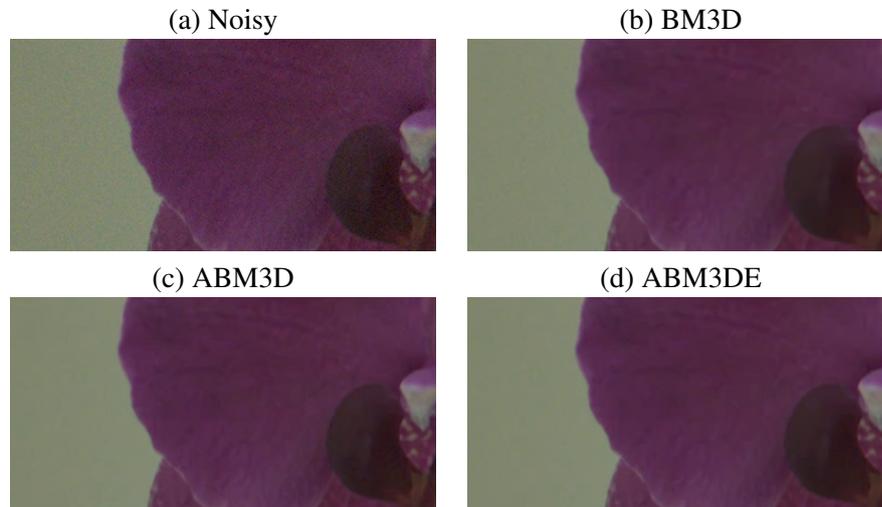


Figure 6.13: Comparison of the non-local low-complexity BM3D-based denoising methods applied on the demosaicked RAW, under the proposed realistic noise model. (a) crop from noisy image “image8” with  $\sigma = 3.96$  and ISO 800. (b) BM3D, PSNR=40.91. (c) ABM3D, PSNR=41.20. (d) ABM3DE, PSNR=41.44.

tively, as seen in the plot of Fig.6.14, and subjectively, as exemplified in Fig.6.13.

The default parameter of the BM3D algorithm is determined by the standard deviation of the noise. As we did for the local method, we optimize this parameter for each image and noise level of the test set, choosing the one that maximizes the PSNR value of the denoised result  $I_{BM3D}$ . For the implementation of ABM3DE that produced the optimal result, we combine information from two denoised versions of the same image, therefore the running time is higher than in the case of the ABM3D method. While a naive implementation considers applying denoising twice, a faster one can reuse patch information and compute two denoised versions corresponding to two  $\sigma$  values. Nevertheless, the ABM3DE method is slower than the ABM3D method. Table 6.3 shows the average running time for the AVTVE, ABM3D and ABM3DE methods for one 1000x2000 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 [Lebrun, 2012], while we

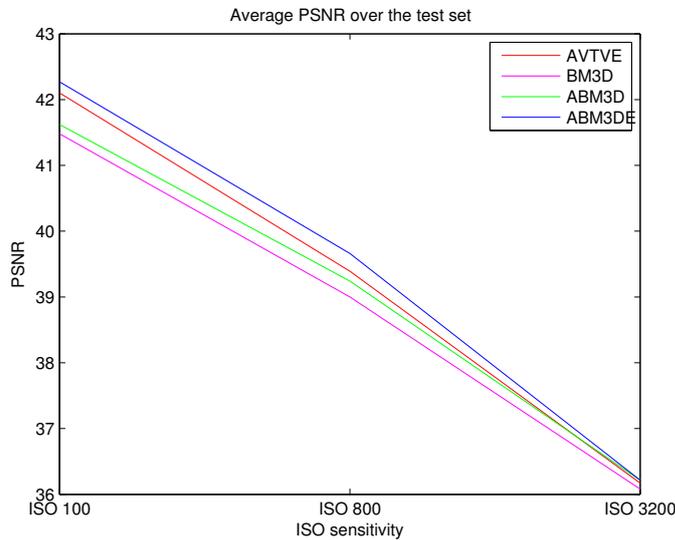


Figure 6.14: Comparison between the local denoising method AVTVE to the low-complexity BM3D, ABM3D and ABM3DE applied on the demosaicked RAW, under the proposed realistic noise model. Average PSNR values computed over our image test set.

point out again that our current implementation of the AVTVE algorithm is not optimized for speed.

The plot in Fig.6.14 shows the average PSNR values over our proposed image dataset, for each noise level 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 values. Notice that for the highest noise level, all methods produce denoised images with a very similar PSNR value.

Fig.6.15 includes examples of images denoised with AVTVE, and the ABM3D and ABM3DE approaches described above, with images denoised with the optimal parameters aforementioned.

For these noise levels, that arise in common situations, the results are visually comparable. This visual similarity among denoising results is inconsistent with the numeric differences in PSNR, which points out the limitation of this metric to evaluate denoising algorithms, as it has been shown recently in [Ghimpeteanu

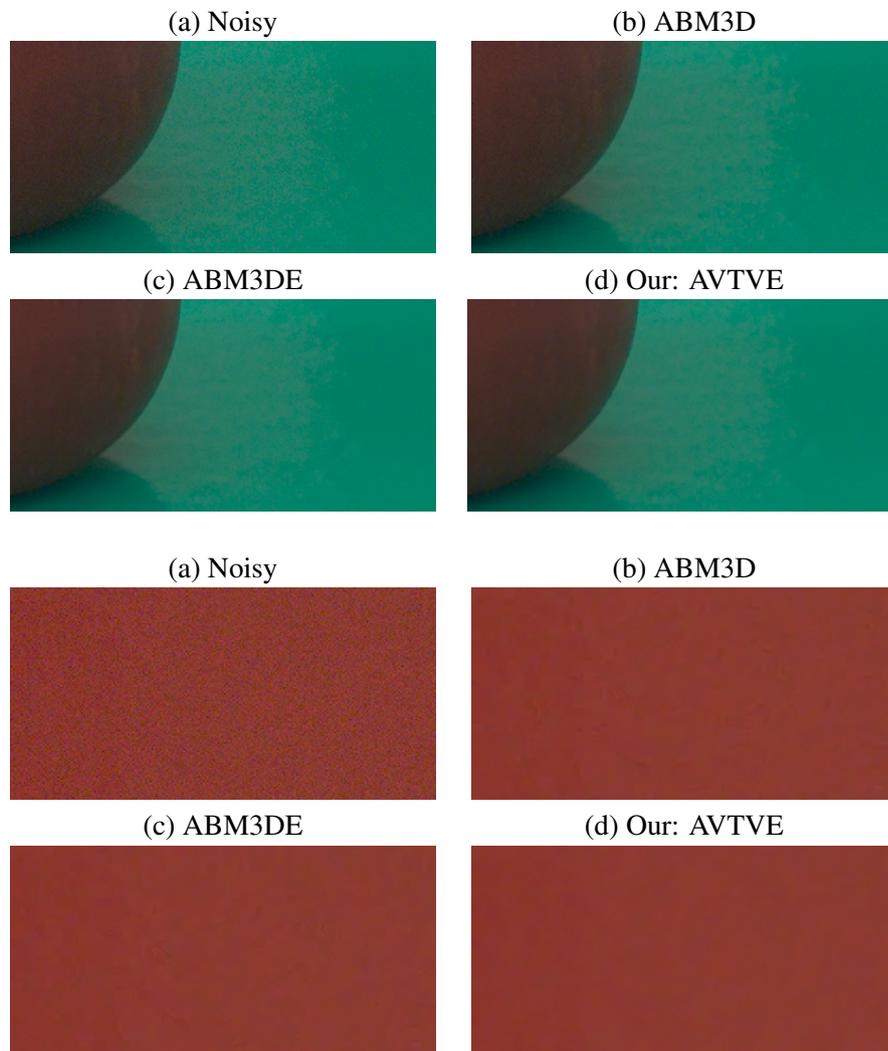


Figure 6.15: Comparison of our local denoising method AVTVE to ABM3D and ABM3DE denoising algorithms applied on the demosaicked RAW, under the proposed realistic noise model. Rows 1-2. (a) crop from noisy image “image1” with  $\sigma = 4.55$  and ISO 100. (b) ABM3D result, PSNR=37.32. (c) ABM3DE result, PSNR=37.93. (d) Our result AVTVE, PSNR=37.95. Rows 3-4. (a) crop from noisy image “image7” with  $\sigma = 9.14$  and ISO 3200. (b) ABM3D result, PSNR=38.98. (c) ABM3DE result, PSNR=38.95. (d) Our result AVTVE, PSNR=39.70.

Table 6.3: Average running time (sec) for one 1000x2000 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

et al., 2016c]. On the two top rows, all images keep the apple texture after denoising; however, we can see that our result and ABM3DE have a cleaner appearance in the green surface. On the two bottom rows, ABM3D and ABM3DE show blocking artifacts in the homogeneous area.

## 6.7 Experiments on cinema camera images

### 6.7.1 Camera processing pipeline

In order to highlight the applicability of our method, we test it on another type of camera: a cinema camera, that employs logarithmic encoding and uses a higher bit-depth. In this section, we present experiments done on images taken with an ARRI ALEXA cinema camera [Andriani et al., 2013]. We simulate its pipeline and design our experiment following these steps:

1. **Recording** The pipeline begins with a linear RAW image of 16-bit depth.
2. **White-balance**
3. **Demosaicking**
4. **Normalization** The data is normalized for the exposure, using a channel-wise formula. For the red channel, the normalization reads as:

$$R = \frac{18EI}{400} \cdot \frac{R_{RAW} - 256}{65535 - 256}, \quad (6.8)$$

and identically for the channels  $G$  and  $B$ . The parameter  $EI$  is the exposure index.

5. **Color Correction** The conversion from the camera color space to the sRGB color space Rec709 is done by the following matrix multiplication:

$$\begin{pmatrix} R_s \\ G_s \\ B_s \end{pmatrix} = \begin{pmatrix} 1.9128 & -0.7779 & -0.1348 \\ -0.1054 & 1.4353 & -0.3299 \\ 0.0492 & -0.5537 & 1.5046 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (6.9)$$

6. **Logarithmic encoding** After color correction, we apply a Log C function [Manual, 2012] on the image data and then quantize from 16-bit to 10-bit, followed by applying the inverse Log C function. Encoding the data to 10-bit depth in the Log domain minimizes the quantization errors. We represent the 10-bit image as a 16-bit image.

7. **Gamma Correction** Finally, the correction power function is applied channel-wise:

$$C' = \begin{cases} 12.92C & \text{if } C \leq 0.0031308 \\ 1.055C^{1/2.4} - 0.055 & \text{if } C > 0.0031308 \end{cases} \quad (6.10)$$

where  $C$  is  $R_s$ ,  $G_s$  or  $B_s$ . The final image is a 16-bit depth image ready for display.

## 6.7.2 Denoising experiments

### Compare local denoising on RAW to non-local denoising on camera output

We create the following denoising experiment: apply a local denoising method to the demosaicked RAW image before quantizing to 10-bits followed by the rest of the pipeline, and compare this result to applying a non-local patch-based method to the final 16-bit RGB image (obtained after a 10-bit quantization step).

In our test set, as a noisy image we have an actual real noise image captured by a cinema camera, and the corresponding clean image is obtained as the average of a serie of noisy images taken in identical conditions. After applying white balance and demosaicking on the two 16-bit noisy and clean RAW images, we proceed with the following steps:

1. Start with the demosaicked and white-balanced noisy RAW image and apply the previously described basic color processing pipeline: normalization,

color correction, Log C quantization from 16-bit to 10-bit, applying the inverse of the Log C function and applying the correction power function. We denote the output of the color processing pipeline for the noisy image by  $I_{noisy}$ . The same pipeline applied to the clean RAW image outputs the image  $I_{clean}$ .

2. Apply the non-local denoising methods NLM and BM3D on the 16-bit depth noisy image  $I_{noisy}$ , getting the denoised images  $I_{NLM}$  and  $I_{BM3D}$ .
3. Start with the demosaicked and white-balanced noisy RAW image and do the normalization step. Afterwards apply our local denoising algorithm AVTVE, followed by the rest of the basic cinema camera pipeline: conversion to sRGB color space, Log C, quantization from 16-bit to 10-bit, applying the inverse of the Log C function and applying the correction power function. Denote this denoised image by  $I_{OUR}$ .
4. Compute the PSNR and SSIM index values of  $I_{OUR}$ ,  $I_{NLM}$  and  $I_{BM3D}$  with respect to the ground truth  $I_{clean}$ .

As test set images we use 10 crops from an actual real noise photograph taken with a cinema camera, with 3 different noise levels.

Fig.6.18 shows on the first row an actual real noise image with marked regions that we select as input for our test set. As in the previous case of the standard camera, we optimized image-wise the denoising parameters according to the PSNR value, for each denoising method, and each of the 3 noise levels. A visual comparison is illustrated on the next four rows from Fig.6.18, where the images are denoised with the optimal parameters. In the crop images from the second and third row, we can see an example where for a small noise level the denoising methods BM3D, NLM and AVTVE produce comparable results, with our method having a higher PSNR. The plot in Fig.6.16 shows the average PSNR values over our test set. The plots have on the horizontal axis the average noise standard deviation corresponding to 8-bit depth noisy images. The BM3D method produces images with higher PSNR values compared to our method.

The visual similarity among denoising results is inconsistent with the numeric differences in PSNR, which points out the limitation of this metric to evaluate denoising algorithms, as it has been shown recently in [Ghimpeteanu et al., 2016c].

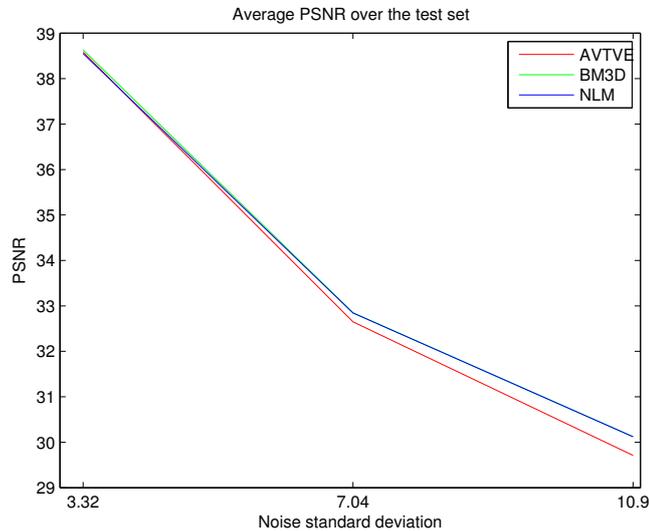


Figure 6.16: Comparison of our local denoising method AVTVE to BM3D and NLM denoising algorithms applied on the camera output. Average PSNR value plot computed over our image test set.

### Compare local and low-complexity non-local denoising applied at the same stage of the image processing chain

We performed another denoising experiment, as in the case of the tests with a standard camera: apply a local denoising method to the demosaicked RAW image before quantizing to 10-bits followed by the rest of the pipeline, and compare this result to applying a low-complexity non-local patch-based method at the same stage of the image processing pipeline and followed by the rest of the steps.

We use the BM3D method with the patch size 8x8 and search for similar patches in a window of size 10x10. We perform the denoising experiments on the same test set shown in the first row of Fig.6.18.

For each denoising method, and each of the 3 noise levels the denoising parameters are optimized according to the PSNR value. In the crop images from the second and third row from Fig.6.18, we can see an example where for a small noise level the denoising methods AVTVE, ABM3D and ABM3DE produce comparable results, with our method having a higher PSNR than ABM3D and

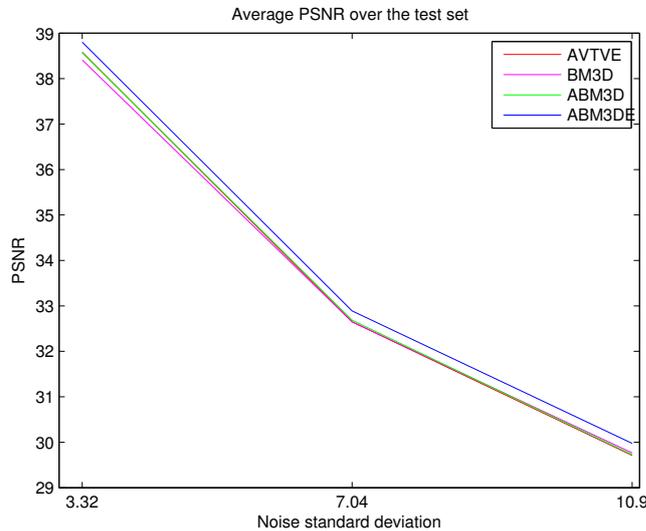


Figure 6.17: Comparison of our local denoising method AVTVE to ABM3D and ABM3DE denoising approaches applied on the demosaicked RAW. Average PSNR value plot computed over our image test set.

lower than ABM3DE. In the bottom rows, we can see an example where, for a higher noise level, our method has a higher PSNR than the other two methods. Our method produces a cleaner denoised image than ABM3D and ABM3DE, which reveal blocking artifacts. Table 6.4 shows the average running time for the AVTVE and ABM3D methods for a 100x200 color image from our test set, on the same machine.

The plot in Fig.6.17 shows the average PSNR values over our test set. The plots show on the horizontal axis the average noise standard deviation corresponding to 8-bit depth noisy images. The ABM3DE approach produces images with higher PSNR, but also higher running-time compared to our method and ABM3D, which produce very similar results. We can notice that AVTVE produces a slightly better PSNR result than the ABM3D method, for a small noise level, and comparable results for higher noise level.

As in the previous experiment, for these noise levels appearing in common situations, the results are visually comparable, while our method is faster.

Table 6.4: Average running time (sec), for one 100x200 color test image, of *AVTVE*, *ABM3D* and *ABM3DE*, for different noise levels.

Noise variance	3.32	7.04	10.9
AVTVE	0.036	0.038	0.068
ABM3D	0.07	0.07	0.07
ABM3DE	0.14	0.14	0.14

## 6.8 Conclusion

Denoising seems to be a practically solved problem for the AWG noise case. This includes also the category of RAW pictures, as the Anscombe transform removes the noise signal-dependency, allowing RAW noise to be treated with methods designed for AWG noise.

Given real-time computational cost constraints, local TV-based methods are a good option for in-camera denoising. In this chapter, we showed that denoising a RAW image in the camera hardware with a fast local method gives better or comparable results, visually and in terms of PSNR, to denoising the camera output with a non-local patch-based method, at a fraction of the running time. We also showed the same advantage, better or comparable results for a smaller running time, of our local denoising approach over applying a low-complexity non-local patch-based denoising method at the same stage of the image processing pipeline.

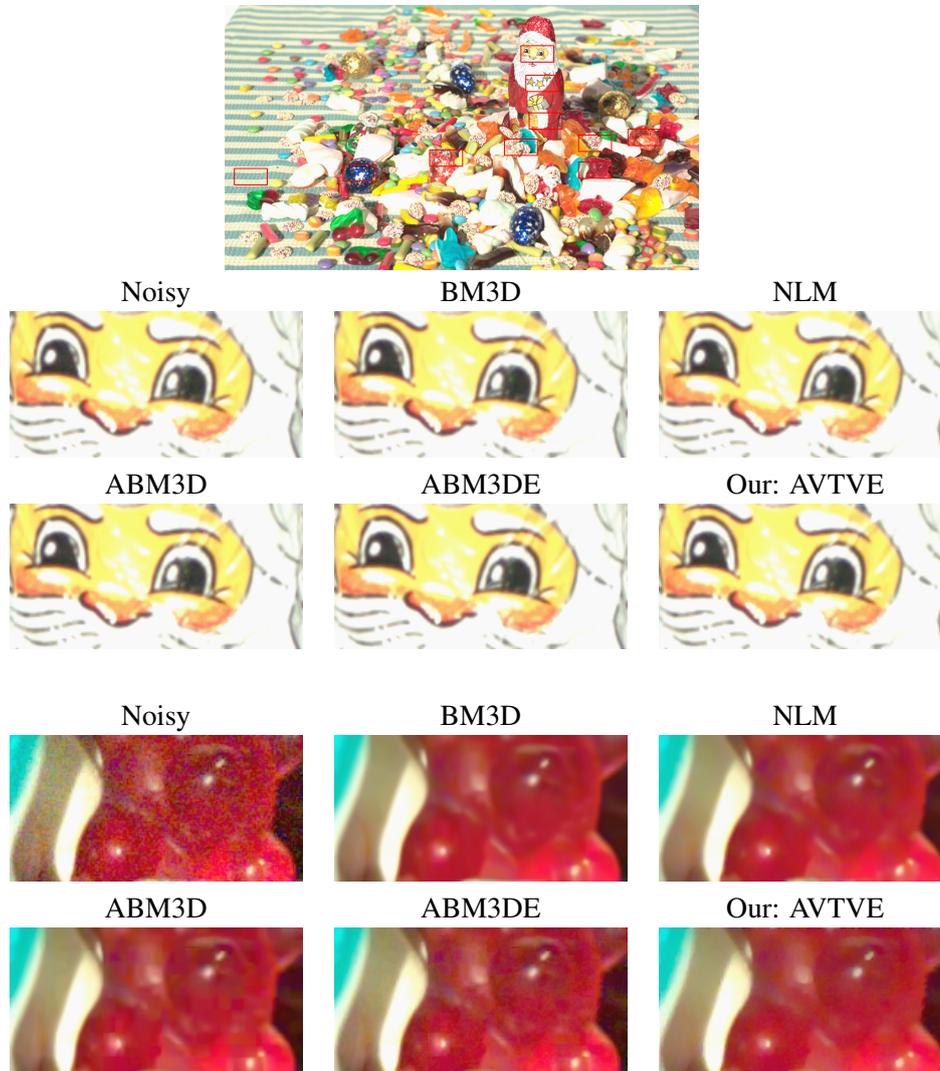
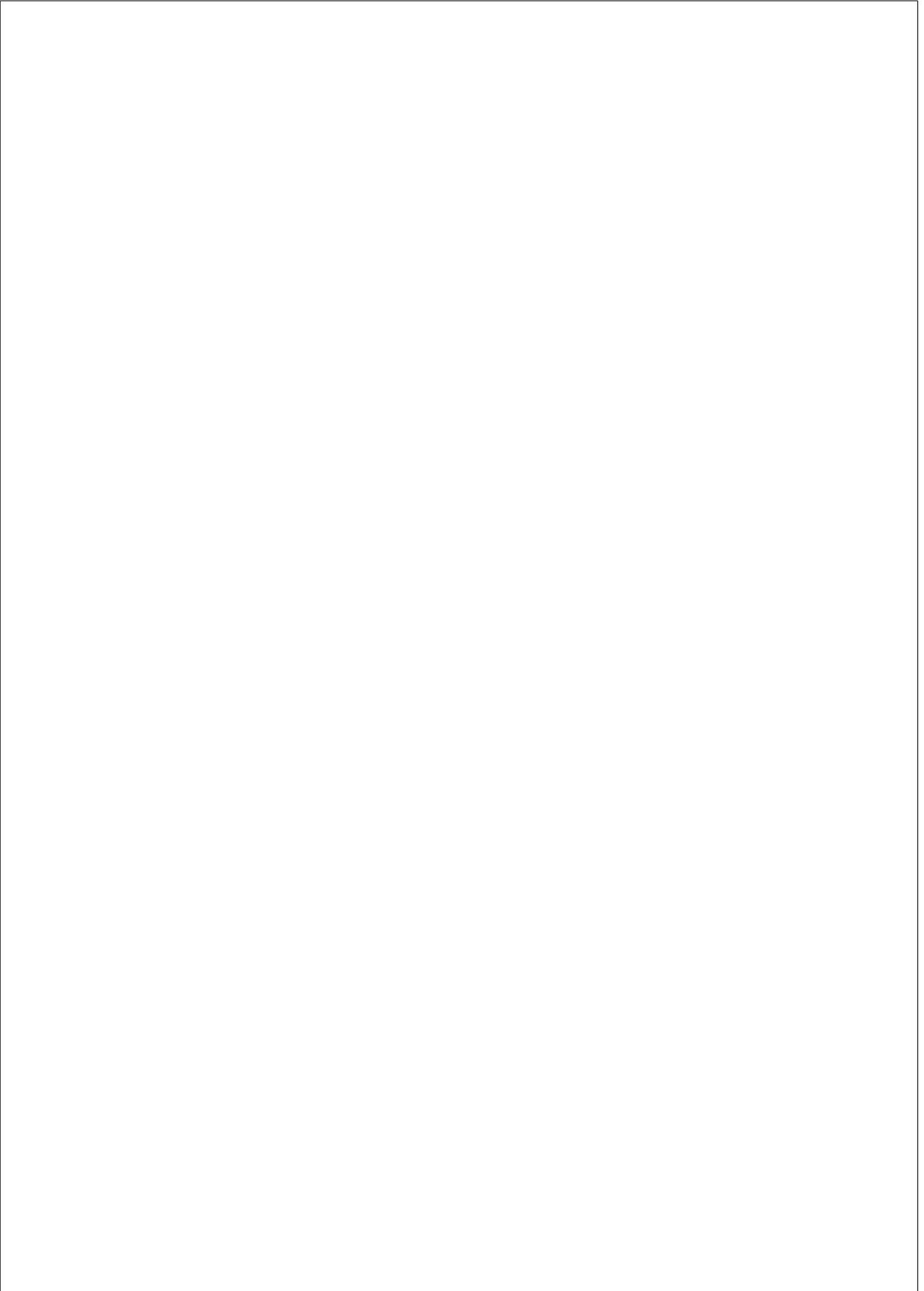


Figure 6.18: Comparison of our local denoising method AVTVE to the BM3D and NLM denoising methods applied on the camera output and the low-complexity denoising strategies ABM3D and ABM3DE applied on the demosaicked RAW. Row 1: test image taken with a cinema camera with marked crops selected for our “candy” test set. Row 2 and 3: noisy image “candy2” with  $\sigma=2.68$ , BM3D result with PSNR=39.74, NLM result with PSNR=39.75, ABM3D result with PSNR=39.73, ABM3DE result with PSNR=39.94, AVTVE result with PSNR=39.86. Row 4 and 5: noisy image “candy4” with  $\sigma=13.82$ , BM3D result with PSNR=30.49, NLM result with PSNR=30.13, ABM3D result with PSNR=29.58, ABM3DE result with PSNR=29.40, AVTVE result with PSNR=29.66.



## CHAPTER 7

---

### **Denoising: tune parameters for subject preference not for objective measures**

---

This chapter is based on the work described in the paper “Local denoising based on curvature smoothing can visually outperform non-local methods on photographs with actual noise” [Ghimpeteanu et al., 2016c], and the text is mostly reproduced from it.

In this chapter we propose a local, low complexity denoising method. From [Bertalmío and Levine, 2014] we take the notion of reconstructing the denoised image from the denoised curvature, but instead of applying a denoising method  $\mathcal{F}$  to the original (noisy) curvature we directly compute a smooth approximation to the curvature for each color channel. A subjective evaluation was performed by asking subjects to choose their preferred image from three on real photographs with visible noise. We find that our method yields solutions that visually match (on average) or surpass (for some images) results from two more sophisticated algorithms (Non-local Means [Buades et al., 2005b] and BM3D [Dabov et al., 2007]), but at a fraction of their computational cost. This suggests the possibility of incorporating the proposed method into the camera processing pipeline. We also conduct a subjective evaluation using clean images with three levels of added synthetic (Gaussian) noise. We test whether the metrics PSNR and SSIM can predict the subjective data. We find that SSIM can reasonably approximate the averaged preference of subjects as a function of the noise level, but that neither metric can reliably predict the preferred denoising algorithm on an image-by-image basis. This observation highlights the limitations of the quantitative metrics

PSNR and SSIM when evaluating denoising results.

## 7.1 Motivation

As seen in Chapter 3, noise in photographs is unavoidable and causes a loss of visual quality and dynamic range. All cameras perform some form of denoising, but this is limited by the computational power of the camera. Techniques such as spectral analysis, variational methods and PDEs have all been successfully used for denoising [Aubert and Kornprobst, 2002] and while some are fast to compute, they typically do not perform as well as newer, non-local, but computationally intensive, patch based methods [Awate and Whitaker, 2006, Buades et al., 2005b, Dabov et al., 2007, Elad, 2010, Lebrun et al., 2013]. These latter algorithms can produce high quality results, but their computational complexity makes in-camera implementation impractical.

Another line of denoising algorithms have suggested denoising some geometric feature of an image, instead of denoising the image directly [Bertalmío and Levine, 2014, Ghimpeteanu et al., 2016a, Lysaker et al., 2004b, Osher et al., 2005]. The authors in [Bertalmío and Levine, 2014] proposed the following approach for removing noise from (grayscale) image  $I_0 = a + n$ , where  $a$  is a ‘clean’ image and  $n$  is Gaussian noise of mean zero: given a denoising method  $\mathcal{F}$ , instead of applying it directly on  $I_0$  it’s better to use it to denoise the curvature of level lines of  $I_0$ ,  $\kappa(I_0) = \text{div}(\nabla I_0 / |\nabla I_0|)$ , resulting in  $\kappa_d(I_0)$ , and then reconstruct a ‘clean’ image whose curvature matches  $\kappa_d(I_0)$ . In [Bertalmío and Levine, 2014] it is shown that this framework improves on a variety of denoising methods  $\mathcal{F}$  including local and patch-based techniques (e.g. [Rudin et al., 1992, Buades et al., 2005b, Dabov et al., 2007]), but the evaluation is only performed in terms of objective measures such as PSNR. Several works have demonstrated that while state of the art patch-based methods are nearing optimality with respect to mean squared error, and thus PSNR, there is some room for improvement [Chatterjee and Milanfar, 2010, Levin and Nadler, 2011b, Levin et al., 2012]. Still, while PSNR accounts for the total pixel error across the image, it is widely accepted that it is not a good indicator of perceived image quality. And while SSIM [Wang and Bovik, 2002] aims to estimate perceived errors, it still does not consistently match human preference [Pambrun and Noumeir, 2015]. Thus the study of image quality metrics is an active research area, with the main challenge being the lack

of accurate models of visual perception and subjective quality.

## 7.2 A new local denoising method

### 7.2.1 Foundations of the model

The origin of the denoising method that we propose in this chapter is related to the following energy model, introduced in [Ballester et al., 2001]:

$$E(I) := \int_{\Omega} \|\nabla I\| - \nabla I \cdot \theta(I_0) \, d\Omega \quad (7.1)$$

where  $\theta(I_0)$  is a unit vector field indicating the normals to the level lines of a grey-level image  $I_0: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ . In [Bertalmío and Levine, 2013] the authors propose a variational method to fuse a short-exposure image, dark and noisy but sharp, with a long-exposure image, bright and clean but blurry. Their technique involves minimizing an energy functional with two terms, where one term locally matches the color histogram of the result to that of the long-exposure image; the other term is in fact the energy (7.1), with the purpose of matching the level-lines of the solution with the level lines of the short-exposure image  $I_0$ . The numerical solution is achieved through a gradient descent algorithm associated to a differentiable approximation of the whole energy. In particular, the gradient descent corresponding to the term (7.1) is

$$I_t = \kappa_{\epsilon_1}(I) - \kappa_{\epsilon_2}(I_0) \quad (7.2)$$

$$= \nabla \cdot \left( \frac{\nabla I}{\sqrt{\|\nabla I\|^2 + \epsilon_1}} \right) - \nabla \cdot \left( \frac{\nabla I_0}{\sqrt{\|\nabla I_0\|^2 + \epsilon_2}} \right) \quad (7.3)$$

where  $\kappa_{\epsilon_1}(I)$  is the Euclidean curvature of the level lines of  $I$ ,  $\kappa_{\epsilon_2}(I_0)$  is the Euclidean curvature of the level lines of  $I_0$ , and  $\epsilon_1, \epsilon_2 > 0$  are introduced to avoid division by zero, a common practice in this sort of scheme [Rudin et al., 1992]. The authors in [Bertalmío and Levine, 2013] observed that the term  $\kappa_{\epsilon_1}(I) - \kappa_{\epsilon_2}(I_0)$  may have a noticeable and good-quality denoising effect if  $\epsilon_1$  is kept very small but  $\epsilon_2$  is large: the final fusion result has substantially less noise than the short-exposure image, and the edges are still kept sharp. As the value of  $\epsilon_2$  increases, the denoising is more pronounced.

In [Bertalmío and Levine, 2014] it is shown that, given any denoising method  $\mathcal{F}$ , rather than applying  $\mathcal{F}$  directly to a noisy image  $I_0$  it’s better (in terms of objective metrics PSNR and SSIM [Wang and Bovik, 2002]) to apply  $\mathcal{F}$  to denoise the curvature  $\kappa(I_0)$ , and then reconstruct a clean image by running to steady state the following evolution equation

$$I_t = \kappa_\epsilon(I) - \mathcal{F}(\kappa(I_0)) + \lambda(I - I_0), \quad (7.4)$$

where  $0 < \epsilon \ll 1$  such that  $\kappa_\epsilon$  approximates  $\kappa(I)$ , and  $\lambda > 0$  is a Lagrange multiplier, based on the noise variance in a Gaussian noise scenario, that automatically stops the evolution.

### 7.2.2 The proposed approach

Our aim is to design a computationally fast denoising method that produces good visual quality results on photos with real noise that is generated during the acquisition process of the camera. From [Bertalmío and Levine, 2013] we take the idea of using a large value  $\epsilon_2$  to compute the curvature of the noisy data  $I_0$ , and use  $\kappa_{\epsilon_2}(I_0)$  as a quick and easy way to obtain a regularized version of the curvature. We follow [Bertalmío and Levine, 2014] in reconstructing a denoised image from its denoised curvature, but we replace  $\mathcal{F}(\kappa(I_0))$  with  $\kappa_{\epsilon_2}(I_0)$  and remove from Eq. (7.4) the term  $\lambda(I - I_0)$  since it privileges noise with a Gaussian distribution, whereas actual noise in photographs isn’t well described by an additive Gaussian model. Our proposed method is then the following. For each color channel (R,G,B) we take the original noisy data  $I_0$ , compute its regularized curvature  $\kappa_{\epsilon_2}(I_0)$ , and starting from  $I^0 = I_0$  iterate for some number  $N$  steps of this equation:

$$I^{n+1} = I^n + \Delta t \left[ \nabla^- \cdot \left( \frac{\nabla^+ I^n}{\sqrt{\|\nabla^+ I^n\|^2 + \epsilon_1}} \right) - \kappa_{\epsilon_2}(I_0) \right], \quad (7.5)$$

where  $\nabla^+$  (resp.  $\nabla^-$ ) denotes the forward (resp. backward) spatial difference operator. This is clearly a local method since the curvature at each pixel location is estimated using just a  $3 \times 3$  stencil around it. The parameter  $\epsilon_1$  is fixed and is chosen very small in order to approximate  $\kappa(I)$ , and the time step  $\Delta t$  and the number of iterations  $N$  are also fixed. The values we used are:  $\epsilon_1 = 10^{-6}$ ,  $\Delta t = 0.002$  and  $N = 30$ . Therefore, our denoising approach has only one



Figure 7.1: Figure illustrating our test images for the AWG noise denoising experiment, that are crops from Kodak data set.

parameter: the regularizing value  $\epsilon_2$ . We will specify the way it is determined in the next Section.

### 7.3 Experiments

We compare the proposed curvature smoothing (CS) algorithm against two established, non-local, patch based algorithms: NLM [Buades et al., 2005b] and BM3D [Dabov et al., 2007]. The comparison is performed on two image databases: images from the Kodak database [<http://r0k.us/graphics/kodak/>, 2017] with added Gaussian noise, illustrated in Fig.7.1, and photographs taken by us with real noise, shown in Fig.7.2. We evaluate the three denoising methods using subjective testing, and in the case of the Kodak images where we have an established ground truth we also perform an evaluation using the metrics PSNR and SSIM.

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 observer was asked to look at the original image and then indicate which of the three denoised images provided they preferred.



Figure 7.2: Figure illustrating real actual noise photographs that we used as test images for our denoising experiment.

In terms of computational cost, our method, being local, has linear complexity (as we mentioned, curvature is computed on  $3 \times 3$  stencils) whereas NLM and BM3D involve comparing patches over an area so their complexity is quite higher; in practice, NLM takes 7 times longer and BM3D 10 times longer than CS, even though our implementation is not optimized while for NLM and BM3D we use the optimized code from [Buades et al., 2011] and [Lebrun, 2012], respectively.

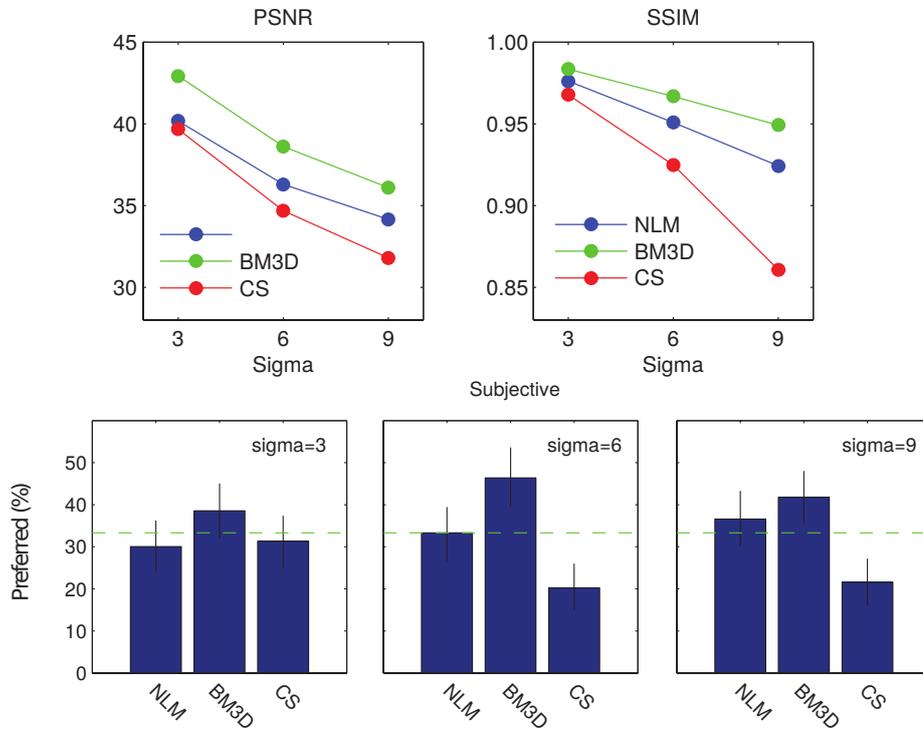


Figure 7.3: Top row: Average PSNR and SSIM computed for 3 images from the Kodak data base. Bottom row: the results of psychophysical experiment for comparing our proposed local denoising method to BM3D and NLM.

### 7.3.1 Kodak database images

We randomly picked three images from the Kodak database (“kodim1”, “kodim3” and “kodim13”) and added three levels of Gaussian noise  $\sigma = 3, 6, 9$ ; these values may seem low, but they correspond to normal noise present in properly exposed pictures, as the real examples in the next subsection will attest. The implementations of denoising algorithms NLM [Buades et al., 2011] and BM3D [Lebrun, 2012] take the value of  $\sigma$  as input and require no further configuration. For our method we find the value of  $\epsilon_2$  via a subjective methodology; subjects are asked to adjust  $\epsilon_2$  via key presses until they find the most pleasing result. For

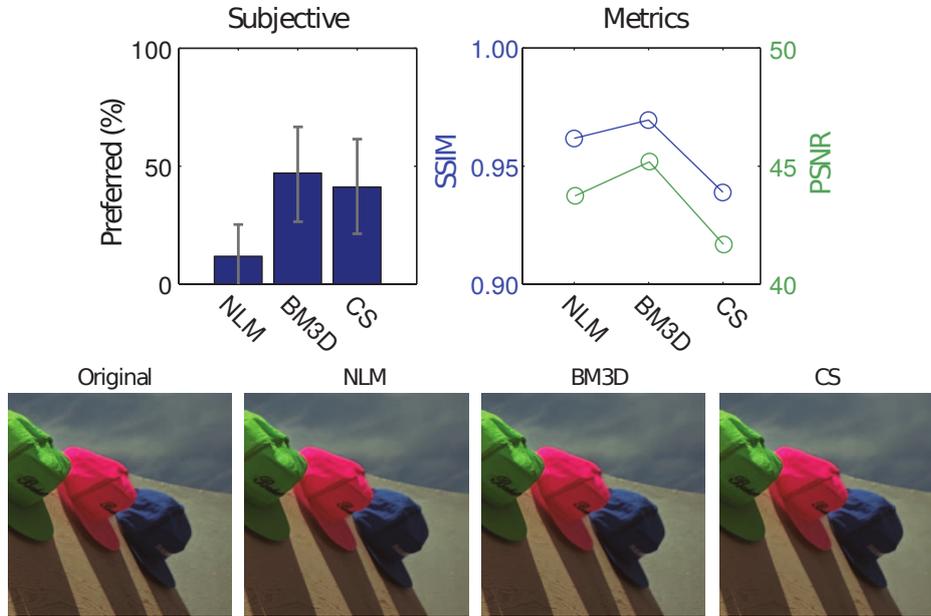


Figure 7.4: Visual comparison for one test crop from image “kodim3” and user preferences.

each noise level, we average across subjects and images to obtain a single value of  $\epsilon_2$  as shown in Table 7.1.

Having the ground truth we can compare the values of PSNR and SSIM for each denoising method. We also perform a user preference test (with the procedure described above). We take crops from the full size images to allow the simultaneous presentation of the 4 images at their native resolution (thus avoiding resizing). The results are summarised in Fig.7.3. We add 95% error bars (estimated via bootstrapping), thus significance can be inferred from visual

Table 7.1: Optimized parameter value  $\epsilon_2$  as a function of  $\sigma$ .

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

inspection. We find that the SSIM metric provides a reasonable approximation of the subjective results, correctly predicting that the differences between the algorithms should be small at low noise levels and that the proposed CS method performs poorly at high noise levels. Despite this, both the PSNR and SSIM metrics are poor at predicting which algorithm is preferred on an image-by-image basis. To assess the metric performance we first compute an upper bound by randomly splitting the subjective data into two subject groups (A and B). For each image we then compute a percentage correct score, 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. We find that on average group A predicts the data from group B 64% of the time. In contrast, both the SSIM and PSNR achieve a score of less than 46%. Note the baseline score is 33%.

In Fig.7.4 we show an image example where the PSNR and SSIM index values predict that the image quality of the NLM output is superior to that of CS, however, subjects prefer the latter to the former.

### 7.3.2 Real noise images

Now we compare user preference on the results from CS, NLM and BM3D over 30 images cropped from 5 real noise photos. Since the noise standard deviation is not known, we find the values for  $\sigma$  (the parameter for NLM and BM3D) and  $\epsilon_2$  (parameter for CS) through user tests in the same manner as above. The results of the psychophysical experiment are shown in Fig.7.5. We plot the user preference average across all thirty images. The results show that the performance of the model is statistically identical for the three algorithms. The Fig.7.6 show the results for five individual images as denoted above each graph. Notice how some of these images are quite noisy. For the same image, the optimal value  $\sigma$  chosen by user tests differs for NLM (values up to 5.10) and BM3D (values up to 8.49). If the noise were Gaussian, according to PSNR and SSIM there should be a clear difference in performance of algorithms. Despite the image-dependent variability in the preferred algorithms, the overall results show no one algorithm wins overall. Again, this contradicts the result that *for additive Gaussian noise*, PSNR and SSIM predict that the visual quality of BM3D is always clearly superior to that of NLM, which is always supposed to outperform CS. Fig.7.7-7.9 show individual examples where CS does well in terms of visual quality.

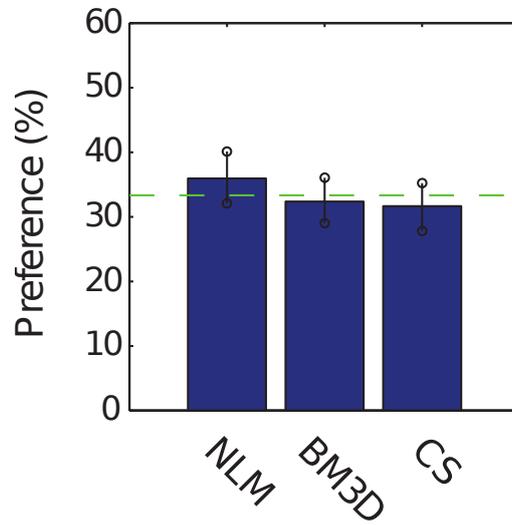


Figure 7.5: Average visual comparison results for 30 real-noise images, for comparing our denoising method to BM3D and NLM.

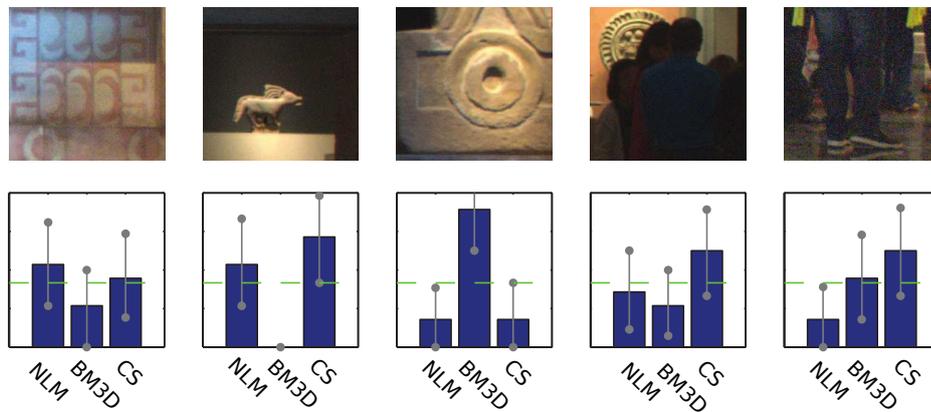


Figure 7.6: Image example crops and user preferences. Notice the case of the second image example, where the noise level is high and users may prefer CS: user estimation of  $\sigma = 4.73$  for NLM and  $\sigma = 7.15$  for BM3D.

## 7.4 Conclusion

We have introduced a local denoising method based on reconstructing a clean image from a smoothed version of the Euclidean curvature of the original noisy input. The method was compared with the non-local, computationally much more intensive algorithms of NLM and BM3D. Our results indicate that in terms of user preference, our proposed method, NLM and BM3D all have the same average performance on real-noise images, that PSNR and SSIM do not correlate with user preference at low, but still quite noticeable, noise levels common in many properly-exposed photographs, and finally, that the outcomes of user preference tests on the results of denoising algorithms applied to images with added Gaussian noise can not be extrapolated to the real case of noisy images where the noise is due to the acquisition process.

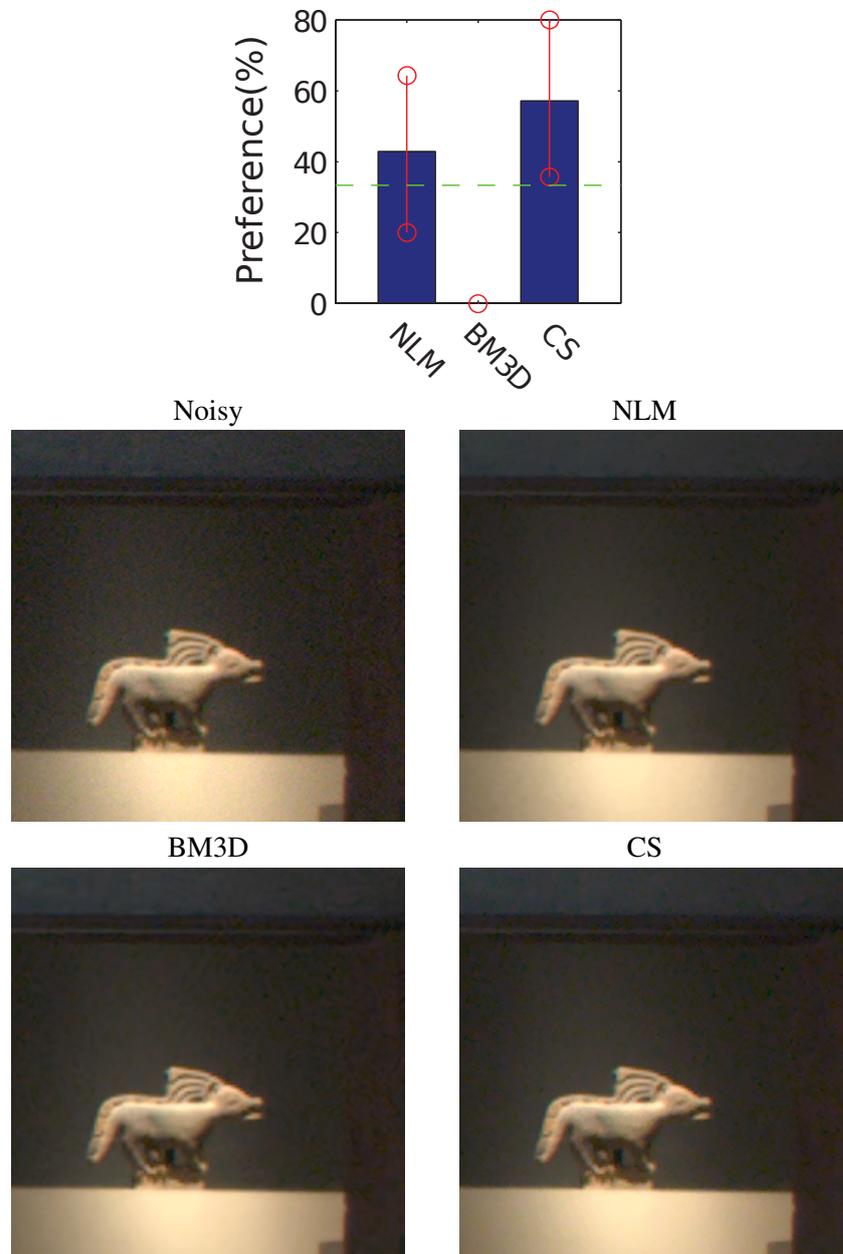


Figure 7.7: An example of real-noise image denoising: user preference and visual comparison.

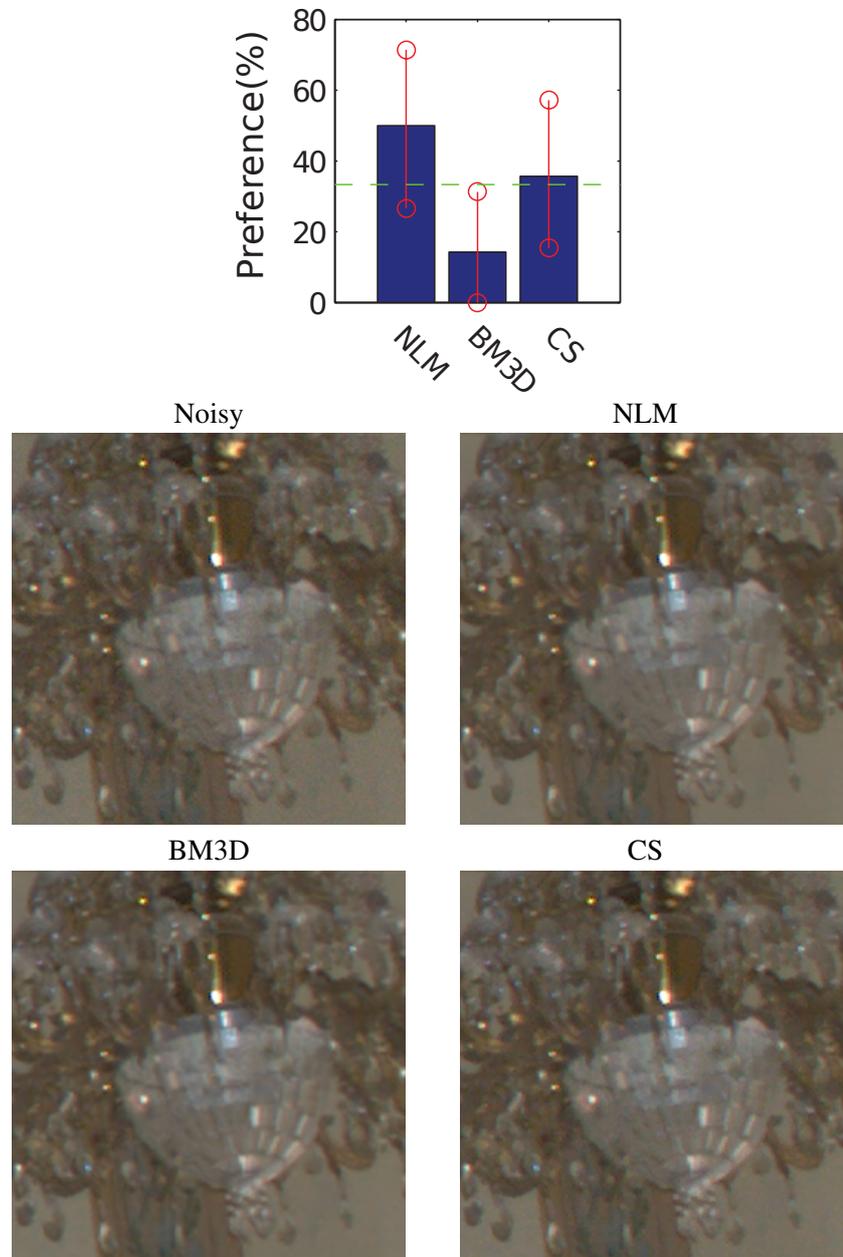


Figure 7.8: An example of real-noise image denoising: user preference and visual comparison.

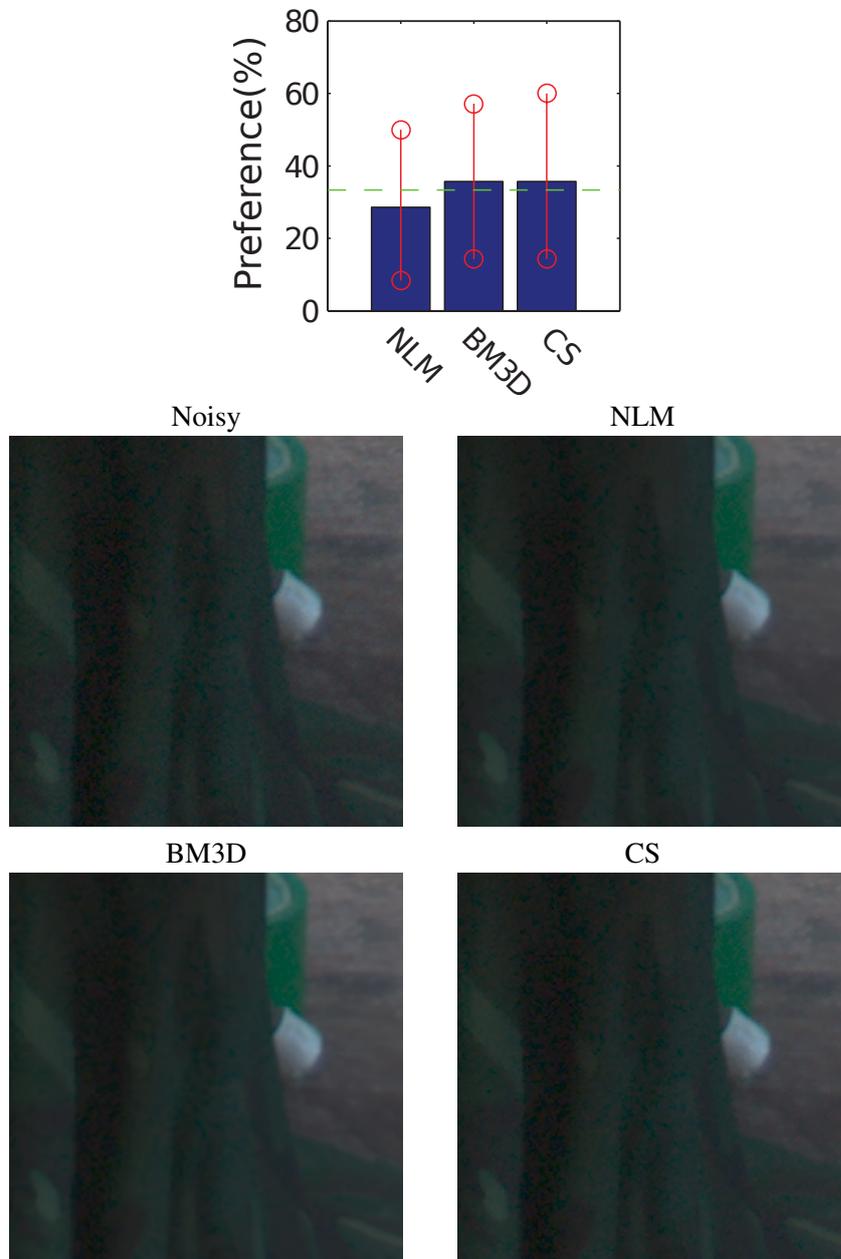


Figure 7.9: An example of real-noise image denoising: user preference and visual comparison.

## CHAPTER 8

---

### Conclusions and Future Work

---

In this thesis, we have shown several approaches to improve noise removal in photographic images. In Chapter 5 we have shown the advantages of decomposing an image into less noisier ones and denoise these instead of the original one. This approach was validated by experiments done with the VTV-based denoising method (a local variational technique), NLM (a non-local patch-based technique) and BM3D (a non-local patch-based technique combined with filtering in spectral domain approach) algorithms on both gray and color images. As noise model we used the AWG noise model.

We approached a more realistic scenario for image denoising in the work described in Chapter 6. While in the context of AWG noise denoising seems to be a practically solved task, for real noise it remains an open problem. Methods designed for AWG noise can be applied also on noisy RAW images, as by using a variance stabilizing transform (such as the Anscombe transform) the noise signal-dependency at the RAW level is eliminated.

Using a noise model that we estimate on RAW images, we showed how a local TV-based approach can be a good choice for in-camera denoising, as it gives better or comparable results (visually and in terms of PSNR) to those of two non-local patch based methods (NLM and BM3D) applied on camera outputs, but at a fraction of their running time. When comparing the local method to a low-complexity non-local patch based method applied also on RAW, the results are comparable, with our approach being faster.

We conclude the thesis with an experiment described in Chapter 7 that under-

lines the limitations of two popular image quality metrics: PSNR and SSIM. We compared a local method based on curvature smoothing to two more sophisticated non-local patch based method (NLM and BM3D). We showed how for real actual noise of moderate intensity, human observers give, on average, equal votes to the three denoising methods. Our experiment indicates that researchers should not extrapolate the conclusions obtained for AWG noise to real actual noise. Moreover, on an image-by-image basis, the PSNR and SSIM metrics are not able to predict subjective preference.

Motivated by these findings, current work involves to extend the experiment described in Chapter 7 to the data set created with the realistic noise model introduced in Chapter 6.

Future work involves extending the moving frame approach used for denoising to other image processing tasks, such as contrast enhancement.

Another line of research would be to adapt the local denoising approach to be applied on RAW data before the demosaicking step; or, even better, to perform in a single step denoising and demosaicking.

As described in Section 3.4, noise is not always something to be removed. There are few cases where it can have a positive connotation. For example, studio photographs taken with a low ISO value can have a fake-look and a subtle amount of noise can improve the perceived quality and sharpness. Or in computer graphics applications, adding a small amount of noise to an image makes it more realistic. An interesting task would be to investigate (with psychophysical experiments) the aesthetic appearance of the noise given by our model and adapt it to user preferences.

A very important issue in the context of image denoising is the lack of a reliable image quality metric, well correlated to subjective preference. This affects not only the evaluation rank of different denoising methods, but also the way denoising methods tune their internal parameters.

Even though current cameras can achieve great image and video quality, there are still many denoising challenges for high ISO settings, low light conditions or in the case of mobile cameras with small image sensors. Also, generally, there is more noise in a video than in one photograph, caused by the high speed with which camera capture the data. Video denoising is still an open problem. As in the case of image denoising, or even more, there is a strong need for a reliable video quality metric.

---

## Bibliography

---

- [Aharon et al., 2006] Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322.
- [Andriani et al., 2013] Andriani, S., Brendel, H., Seybold, T., and Goldstone, J. (2013). Beyond kodak image set: A proposal for a new reference set of motion color images. *IEEE Int. Conf. Im. Process.*
- [Anscombe, 1948] Anscombe, F. (1948). The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3):246–254.
- [Aubert and Kornprobst, 2002] Aubert, G. and Kornprobst, P. (2002). *Mathematical problems in image processing*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, New York. Partial differential equations and the calculus of variations, With a foreword by Olivier Faugeras.
- [Awate and Whitaker, 2006] Awate, S. and Whitaker, R. (2006). Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):364–376.
- [Awate and Whitaker, 2005] Awate, S. P. and Whitaker, R. T. (2005). Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering. 2:44–51.
- [Ballester et al., 2001] Ballester, C., Bertalmío, M., Caselles, V., Sapiro, G., and Verdera, J. (2001). Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Process.*, 10(8):1200–1211.

- [Batard and Bertalmío, 2013] Batard, T. and Bertalmío, M. (2013). Generalized gradient on vector bundle- application to image denoising. *Lecture Notes Comput. Sci.*, 7893:12–23.
- [Batard and Bertalmío, 2014] Batard, T. and Bertalmío, M. (2014). On covariant derivatives and their applications to image regularization. *SIAM J. Imag. Sci.*, 7(4):2393–2422.
- [Batard and Berthier, 2013] Batard, T. and Berthier, M. (2013). Spinor fourier transform for image processing. 7(4):605–613.
- [Bertalmío, 2014] Bertalmío, M. (2014). *Image Processing for Cinema*, volume 4. CRC Press, Taylor & Francis.
- [Bertalmío and Levine, 2013] Bertalmío, M. and Levine, S. (2013). Variational approach for the fusion of exposure bracketed pairs. *IEEE Trans. Image Process.*, 22(2):712–723.
- [Bertalmío and Levine, 2014] Bertalmío, M. and Levine, S. (2014). Denoising an image by denoising its curvature image. *SIAM J. Imag. Sci.*, 7(2):187–211.
- [Blomgren and Chan, 1998] Blomgren, P. and Chan, T. F. (1998). Color tv: total variation methods for restoration of vector-valued images. *IEEE Transactions on image processing*, 7(3):304–309.
- [Brainard et al., 2004] Brainard, D., Chalupa, L., and Werner, J. (2004). *Color Constancy*. MIT Press, Cambridge, MA.
- [Bresson and Chan, 2008] Bresson, X. and Chan, T. F. (2008). Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Probl. Imaging*, 2(4):455–484.
- [Buades et al., 2005a] Buades, A., Coll, B., and Morel, J. (2005a). A non local algorithm for image denoising. *IEEE Computer Vision and Pattern Recognition*, 2:60–65.
- [Buades et al., 2006] Buades, A., Coll, B., and Morel, J. (2006). The staircasing effect in neighborhood filters and its solution. *IEEE Trans. Image Process.*, 15:1499–1505.

- [Buades et al., 2008] Buades, A., Coll, B., and Morel, J. (2008). Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139.
- [Buades et al., 2011] Buades, A., Coll, B., and Morel, J. (2011). Non-local means denoising. *Image Processing On Line*, 1.
- [Buades et al., 2005b] Buades, A., Coll, B., and Morel, J. M. (2005b). A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.*, 4(2):490–530 (electronic).
- [Buades et al., 2009] Buades, A., Coll, B., Morel, J.-M., and Sbert, C. (2009). Self-similarity driven color demosaicking. *IEEE Transactions on image processing*, 18(6):1192–1202.
- [Burger et al., 2012] Burger, H. C., Schuler, C. J., and Harmeling, S. (2012). Image denoising: Can plain neural networks compete with bm3d? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399.
- [Caselles et al., 2002] Caselles, V., Coll, B., and Morel, J. (2002). Geometry and color in natural images. *Journal of Mathematical Imaging and Vision*, 16(2):89–105.
- [Chambolle, 2004] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20:89–97.
- [Charbonnier et al., 1994] Charbonnier, P., Blanc-Féraud, L., Aubert, G., and Barlaud, M. (1994). Two deterministic half-quadratic regularization algorithms for computed imaging. *Proc. IEEE Int. Conf. Image Processing*, 2:168–172.
- [Chatterjee and Milanfar, 2010] Chatterjee, P. and Milanfar, P. (2010). Is denoising dead? *IEEE Transactions on Image Processing*, 19(4):895–911.
- [Chen and Pock, 2017] Chen, Y. and Pock, T. (2017). Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272.
- [Coifman and Donoho, 1995] Coifman, R. and Donoho, D. (1995). Translation-invariant denoising. In *Lecture Notes In Statistics*, pages 125–125.

- [Dabov et al., 2007] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007). Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095.
- [Danielyan et al., 2009] Danielyan, A., Vehvilainen, M., Foi, A., Katkovnik, V., and Egiazarian, K. (2009). Cross-color bm3d filtering of noisy raw data. *Proceedings of the International Workshop Local and Non-Local Approximation in Image Processing*.
- [Deledalle et al., 2012] Deledalle, C., Duval, V., and Salmon, J. (2012). Non-local methods with shape-adaptive patches (nlm-sap). *Journal of Mathematical Imaging and Vision*, 43:103–120.
- [Donoho, 1993] Donoho, D. (1993). Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data. *Proc. Symp. Appl. Math.*, pages 173–205.
- [Donoho and Johnstone, 1998] Donoho, D. and Johnstone, I. (1998). Minimax estimation via wavelet shrinkage. *The Annals of Statistics*, 26(3):879–921.
- [Duval et al., 2011] Duval, V., Aujol, J., and Gousseau, Y. (2011). A bias-variance approach for the nonlocal means. *SIAM Journal on Imaging Sciences*, 4(2):760–788.
- [Ebner, 2007] Ebner, M. (2007). *Color Constancy*. John Wiley and Sons Ltd.
- [Efros and Leung, 1999] Efros, A. and Leung, T. (1999). Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision*, pages 1033–1068.
- [Elad, 2010] Elad, M. (2010). *Sparse and redundant representations*. Springer, New York. From theory to applications in signal and image processing, With a foreword by Alfred M. Bruckstein.
- [Elad and Aharon, 2006] Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745.
- [Fairchild and Johnson, 2005] Fairchild, M. D. and Johnson, G. M. (2005). On the salience of novel stimuli: Adaptation and image noise. *Proc. IS&T/SID 13th Color Imaging Conference*, pages 333–338.

- [Faraji and MacLean, 2006] Faraji, H. and MacLean, W. (2006). Ccd noise removal in digital images. *IEEE Transactions on Image Processing*, 15(9):2676–2685.
- [Foi et al., 2008] Foi, A., Trimeche, M., Katkovnik, V., and Egiazarian, K. (2008). Practical poissonian-gaussian noise modeling and fitting for single-image raw data. *IEEE Trans. Image Process.*, 17(10):1737–1754.
- [Geigel and Musgrave, 1997] Geigel, J. and Musgrave, F. K. (1997). A model for simulating the photographic development process on digital images. In *Proc. ACM SIGGRAPH*, pages 135–142.
- [Geman and Geman, 1984] Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Pat. Anal. Mach. Intell.*, 6:721–741.
- [Ghimpeteanu et al., 2014] Ghimpeteanu, G., Batard, T., Bertalmío, M., and Levine, S. (2014). Denoising an image by denoising its components in a moving frame. *Lecture Notes Comput. Sci.*, 8509:375–383.
- [Ghimpeteanu et al., 2016a] Ghimpeteanu, G., Batard, T., Bertalmío, M., and Levine, S. (2016a). A decomposition framework for image denoising algorithms. *Image Processing, IEEE Transactions on*, 25(1):388–399.
- [Ghimpeteanu et al., 2016b] Ghimpeteanu, G., Batard, T., Seybold, T., and Bertalmío, M. (2016b). Local denoising applied to raw images may outperform non-local patch-based methods applied to the camera output. *IS&T Electronic Imaging Conference*.
- [Ghimpeteanu et al., 2016c] Ghimpeteanu, G., Kane, D., Batard, T., Levine, S., and Bertalmío, M. (2016c). Local denoising based on curvature smoothing can visually outperform non-local methods on photographs with actual noise. *Proceedings of the IEEE International Conference on Image Processing ICIP*.
- [Gonzalez and Woods, 2001] Gonzalez, R. and Woods, R. (2001). *Digital Image Processing*. Addison-Wesley Longman Publishing Co.
- [Gow et al., 2007] Gow, R., Renshaw, D., Findlater, K., Grant, L., McLeod, S., Hart, J., and Nicol, R. (2007). A comprehensive tool for modeling cmos

image-sensor-noise performance. *IEEE Transactions on Electron Devices*, 54(6):1321–1329.

[Haar, 1910] Haar, A. (1910). Zur theorie der orthogonalen funktionen-systeme. *Mathematische Annalen*, 69(3):331–371.

[Hahn et al., 2011] Hahn, J., Tai, X., Borok, S., and Bruckstein, A. M. (2011). Orientation-matching minimization for image denoising and inpainting. *Int. J. Comput. Vis.*, 92(3):308–324.

[Healey and R., 1994] Healey, G. and R., K. (1994). Radiometric ccd camera calibration and noise estimation. *IEEE Trans. PAMI*, 16(3):267–276.

[Hirakawa and Parks, 2006] Hirakawa, K. and Parks, T. (2006). Joint demosaicing and denoising. *IEEE Transactions on image processing*, 15(8):2146–2157.

[<http://cpn.canon-europe.com>, 2017] <http://cpn.canon-europe.com> (2017). [http://cpn.canon-europe.com/content/education/infobank/capturing\\_the\\_image/ccd\\_and\\_cmos\\_sensors.do](http://cpn.canon-europe.com/content/education/infobank/capturing_the_image/ccd_and_cmos_sensors.do).

[<http://ip4ec.upf.edu/realisticnoise>, 2017] <http://ip4ec.upf.edu/realisticnoise> (2017). <http://ip4ec.upf.edu/realisticnoise>.

[<http://r0k.us/graphics/kodak/>, 2017] <http://r0k.us/graphics/kodak/> (2017). <http://r0k.us/graphics/kodak/>.

[<https://digital-photography-school.com>, 2017] <https://digital-photography-school.com> (2017). <https://digital-photography-school.com/understanding-dynamic-range-photography/>.

[<https://petapixel.com/>, 2017] <https://petapixel.com/> (2017). <https://petapixel.com/2010/08/05/the-worlds-first-digital-camera-by-kodak-and-steve-sasson/>.

[<https://photography.tutsplus.com>, 2017] <https://photography.tutsplus.com> (2017). <https://photography.tutsplus.com/tutorials/unlikely-friends-grain-noise-and-sharpening--cms-26634>.

- [<https://sites.google.com/site/marclevoylectures/>, 2016] <https://sites.google.com/site/marclevoylectures/> (2016). <https://sites.google.com/site/marclevoylectures/>.
- [<https://www.focusphotoschool.com>, 2017] <https://www.focusphotoschool.com> (2017). <https://www.focusphotoschool.com/digital-camera-file-formats-raw-jpeg/>.
- [<http://www.cambridgeincolour.com>, 2017] <http://www.cambridgeincolour.com> (2017). <http://www.cambridgeincolour.com/tutorials/camera-sensors.htm>.
- [<http://www.imatest.com>, 2017] <http://www.imatest.com> (2017). <http://www.imatest.com/solutions/noise/>.
- [Jain and Seung, 2009] Jain, V. and Seung, H. S. (2009). Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776.
- [Johnson and Fairchild, 2000] Johnson, G. M. and Fairchild, M. D. (2000). Sharpness rules. *Proc. IS&T/SID Eighth Color Imaging Conference*, pages 24–30.
- [Kayargadde and Martens, 1996] Kayargadde, V. and Martens, J. (1996). Perceptual characterization of images degraded by blur and noise: experiments. *Journal of Optical Society of America*, 13(6):1166–1177.
- [Khashabi et al., 2014] Khashabi, D., Nowozin, S., Jancsary, J., and Fitzgibbon, A. (2014). Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Transactions on Image Processing*, 23(12):4968–4981.
- [Kurihara et al., 2009] Kurihara, T., Aoki, N., and Kobayashi, H. (2009). Analysis of sharpness increase by image noise. In *Human Vision and electronic Imaging XIV*, 7240:1–9.
- [Kurihara et al., 2008] Kurihara, T., Y., M., Aoki, N., and Kobayashi, H. (2008). Digital image improvement by adding noise: An example by a professional photographer. In *Image Quality and System Performance V*, 6808:1–10.

- [Lagae et al., 2010] Lagae, A., Lefebvre, S., Cook, R., Derose, T., Drettakis, G., Ebert, D. S., Lewis, J., Perlin, K., and Zwicker, M. (2010). State of the art in procedural noise functions. *In EG 2010 - State of the Art Reports*.
- [Lebrun, 2012] Lebrun, M. (2012). An analysis and implementation of the bm3d image denoising method. *Image Processing On Line*, 2:175–213.
- [Lebrun et al., 2013] Lebrun, M., Buades, A., and Morel, J. M. (2013). A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688.
- [Lebrun et al., 2012] Lebrun, M., Colom, M., Buades, A., and Morel, J. (2012). Secrets of image denoising cuisine. *Acta Numerica*, 21(1):475–576.
- [Lee et al., 2017] Lee, M., Park, S., and Kang, M. (2017). Denoising algorithm for cfa image sensors considering inter-channel correlation. *Ed. Sensors*, 17(6).
- [Levin and Nadler, 2011a] Levin, A. and Nadler, B. (2011a). Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2:2833–2840.
- [Levin and Nadler, 2011b] Levin, A. and Nadler, B. (2011b). Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2833–2840. IEEE.
- [Levin et al., 2012] Levin, A., Nadler, B., Durand, F., and Freeman, W. (2012). Patch complexity, finite pixel correlations and optimal denoising. Technical report, MIT - Computer Science and Artificial Intelligence Laboratory.
- [Liu et al., 2010] Liu, R., Lin, Z., Zhang, W., and Su, Z. (2010). Learning pdes for image restoration via optimal control. In *Proc. 11th Eur. Conf. Comput. Vis.*, pages 115–128.
- [Luisier et al., 2011] Luisier, F., Blu, T., and Unser, M. (2011). Image denoising in mixed poisson-gaussian noise. *IEEE Transactions on Image Processing*, 20(3):696–708.
- [Lysaker et al., 2004a] Lysaker, M., Osher, S., and Tai, X. (2004a). Color constancy by category correlation. *IEEE Transactions on image processing*, 13(10):1345–1357.

- [Lysaker et al., 2004b] Lysaker, M., Osher, S., and Tai, X.-C. (2004b). Noise removal using smoothed normals and surface fitting. *IEEE Trans. Image Process.*, 13(10):1345–1357.
- [Mairal et al., 2010] Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2010). Non-local sparse models for image restoration. In *IEEE International Conference on Computer Vision*, pages 2272–2279.
- [Mäkitalo and Foi, 2011] Mäkitalo, M. and Foi, A. (2011). A closed-form approximation of the exact unbiased inverse of the anscombe variance-stabilizing transformation. *IEEE Transactions on Image Processing*, 20(9):2697–2698.
- [Mäkitalo and Foi, 2013] Mäkitalo, M. and Foi, A. (2013). Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. *IEEE Transactions on Image Processing*, 22(1):91–103.
- [Malvar et al., 2004] Malvar, H., He, L., and Cutler, R. (2004). High-quality linear interpolation for demosaicing of bayer-patterned color images. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 485–488.
- [Manual, 2012] Manual, A. R. (2012). Alexa log c curve - usage in vfx. *ARRI Reference Manual*.
- [Menkman, 2011] Menkman, R. (2011). *The Glitch Moment(um)*. Institute of Network Cultures.
- [Menon and Calvagno, 2011] Menon, D. and Calvagno, G. (2011). Color image demosaicking: an overview. *Signal Processing: Image Communication*, 26(8):518–533.
- [Milanfar, 2011] Milanfar, P. (2011). A tour of modern image filtering. *Invited feature article to IEEE Signal Processing Magazine*.
- [Mosseri et al., 2013] Mosseri, I., Zontak, M., and Irani, M. (2013). Combining the power of internal and external denoising. *ICCP*.
- [Nakamura, 2005] Nakamura, J. (2005). *Image sensors and signal processing for digital still cameras*. CRC.

- [Nam et al., 2016] Nam, S., Hwang, Y., Matsushita, Y., and Kim, S. (2016). A holistic approach to cross-channel image noise modeling and its application to image denoising. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1683–1991.
- [Nilsson, 2017] Nilsson, F. (2017). *Intelligent Network Video: Understanding Modern Video Surveillance Systems*. CRC Press, Taylor & Francis.
- [Osher et al., 2005] Osher, S., Burger, M., Goldfarb, D., Xu, J., and Yin, W. (2005). An iterative regularization method for total variation-based image restoration. *Multiscale Modeling and Simulation*, 4(2):460–489.
- [Pambrun and Noumeir, 2015] Pambrun, J.-F. and Noumeir, R. (2015). Limitations of the ssim quality metric in the context of diagnostic imaging. In *ICIP*, pages 2960–2963. IEEE.
- [Park et al., 2009] Park, S., Kim, H., Linsel, S., Parmar, M., and Wandell, B. (2009). A case for denoising before demosaicking color filter array data. *Conference Record of the 43rd Asilomar Conference on Signals, Systems and Computers*, pages 860–864.
- [Perlin, 1985] Perlin, K. (1985). An image synthesizer. In *Computer Graphics (Proceedings of ACM SIGGRAPH 85)*, 19:287–296.
- [Perona and Malik, 1990] Perona, P. and Malik, J. (1990). Scale - space and edge detection using anisotropic diffusion. *On PAMI*, 12(7):629–639.
- [Plötz and Roth, 2017] Plötz, T. and Roth, S. (2017). Benchmarking denoising algorithms with real photographs. *To appear in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Portilla et al., 2003] Portilla, J., Strela, V., Wainwright, M. J., and Simoncelli, E. P. (2003). Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on image processing*, 12(11):1338–1351.
- [Poynton, 2012] Poynton, C. (2012). *Digital Video and HDTV Algorithms and Interfaces*. Morgan Kaufmann Publishers Inc., 2 edition.
- [Prakel, 2009] Prakel, D. (2009). *The Visual Dictionary of Photography*. Bloomsbury Publishing PLC.

- [Rahman et al., 2007] Rahman, T., Tai, X., and Osher, S. (2007). A tv-stokes denoising algorithm. *Lecture Notes Comput. Sci.*, 4485:473–483.
- [Ramanath et al., 2005] Ramanath, R., Snyder, W., Yoo, Y., and Drew, M. (2005). Color image processing pipeline. *Signal Processing Magazine, IEEE*, 22(1):34–43.
- [Romano and Elad, 2015] Romano, Y. and Elad, M. (2015). Boosting of image denoising techniques. *SIAM J. Imag. Sci.*, 8(2):1187–1219.
- [Romano et al., 2017] Romano, Y., Elad, M., and Milanfar, P. (2017). The little engine that could: Regularization by denoising (red). *To appear in SIAM J. of Imaging Sciences*.
- [Roth and Black, 2009] Roth, S. and Black, M. (2009). Fields of experts. *International Journal of Computer Vision*, 82(2):205–229.
- [Rudin et al., 1992] Rudin, L., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268.
- [Ruff and Simpson, 2009] Ruff, T. and Simpson, B. (2009). *JPEGs*. Aperture.
- [Schoeberl et al., 2012] Schoeberl, M., Bruckner, A., Foessel, S., and Kaup, A. (2012). Photometric limits for digital camera systems. *IEEE Trans. Image Process.*, 21(2).
- [Seybold et al., 2014] Seybold, T., Cakmak, Ö., Keimel, C., and Stechele, W. (2014). Noise characteristics of a single sensor camera in digital color image processing. *Color and Imaging Conference*, pages 53–58.
- [Shohara and Kazunori, 2010] Shohara, M. and Kazunori, K. (2010). Measurement of color noise perception. *IEEE 17th International Conference on Image Processing*, pages 3225–3228.
- [Talebi and Milanfar, 2014] Talebi, H. and Milanfar, P. (2014). Global image denoising. *IEEE Transactions on image processing*, 23(2):755–768.
- [Templin et al., 2014] Templin, K., Didyk, P., Myszkowski, K., and Seidel, H.-P. (2014). Perceptually-motivated stereoscopic film grain. *Computer Graphics Forum (Proc. Pacific Graphics)*, 33(7).

- [Tomasi and Manduchi, 1998] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. *In ICCV 98: Proceedings of the Sixth International Conference on Computer Vision*, pages 839–846.
- [Trussell and Zhang, 2012] Trussell, H. and Zhang, R. (2012). The dominance of poisson noise in color digital cameras. *IEEE Int. Conf. Im. Process.*, pages 329–332.
- [Tukey, 1976] Tukey, J. (1976). *Exploratory data analysis*. Massachusetts: Addison-Wesley.
- [Van De Ville and Kocher, 2009] Van De Ville, D. and Kocher, M. (2009). Sure-based non-local means. *Signal Processing Letters, IEEE*, 16(11):973–976.
- [Wang and Bovik, 2002] Wang, Z. and Bovik, A. (2002). A universal image quality index. *IEEE Signal Process. Lett.*, 9:81–84.
- [Wang et al., 2004] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on image processing*, 13(4):600–612.
- [Weaver et al., 1991] Weaver, J., Xu, Y., Healy, D., and Cromwell, L. (1991). Filtering noise from images with wavelet transforms. *Magnetic Resonance in Medicine*, 21:288–295.
- [Webster, 2003] Webster, M. (2003). Light adaptation, contrast adaptation, and human colour vision. *In R. Mausfeld and D. Heyer, eds., Colour Perception: Mind and the Physical World, Oxford University Press*, pages 67–110.
- [Weickert, 1996] Weickert, J. (1996). Theoretical foundations of anisotropic diffusion in image processing. *Computing Suppl.*, 11:221–236.
- [Wiener, 1949] Wiener, N. (1949). Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications. *In Technology Press of the Massachusetts Institute of Technology*.
- [Wikipedia, 2017] Wikipedia (2017). [https://en.wikipedia.org/wiki/Active\\_pixel\\_sensor](https://en.wikipedia.org/wiki/Active_pixel_sensor).
- [www.flickr.com, 2017] www.flickr.com (2017). [ps://www.flickr.com/photos/tyleruk2000/6711060387](https://www.flickr.com/photos/tyleruk2000/6711060387).

- [Xie et al., 2012] Xie, J., Xu, L., and Chen, E. (2012). Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 341–349.
- [Yaroslavsky, 1985] Yaroslavsky, L. (1985). *Digital picture processing. An introduction.*, volume 9. Springer Series in Information Sciences.
- [Yaroslavsky, 1996] Yaroslavsky, L. (1996). Local adaptive image restoration and enhancement with the use of dft and dct in a running window. In *SPIEs 1996 International Symposium on Optical Science, Engineering, and Instrumentation*, pages 2–13.
- [Yu et al., 2012] Yu, G., Sapiro, G., and Mallat, S. (2012). Solving inverse problems with piecewise linear estimators: from gaussian mixture models to structured sparsity. *IEEE Trans. Image Process.*, 21(5):2481–2499.
- [Zhang et al., 2017] Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*.
- [Zontak and Irani, 2011] Zontak, M. and Irani, M. (2011). The internal statistics of a single natural image. In *CVPR*.
- [Zontak et al., 2013] Zontak, M., Mosseri, I., and Irani, M. (2013). Separating signal from noise using patch recurrence across scales. *CVPR*.
- [Zoran and Weiss, 2009] Zoran, D. and Weiss, Y. (2009). Scale invariance and noise in natural image. In *IEEE International Conference on Computer Vision*, pages 2209–2216.
- [Zoran and Weiss, 2011] Zoran, D. and Weiss, Y. (2011). From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision*, pages 479–486.

