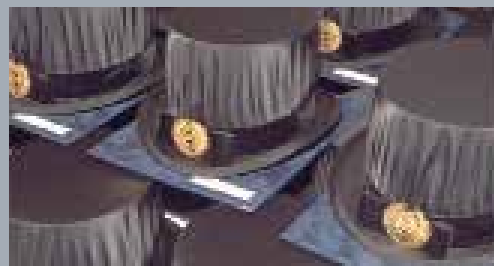




TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Lucio Azzari

**Models and Methods for Estimation and Filtering of
Signal-Dependent Noise in Imaging**



Julkaisu 1429 • Publication 1429

Tampere 2016

Tampereen teknillinen yliopisto. Julkaisu 1429
Tampere University of Technology. Publication 1429

Lucio Azzari

Models and Methods for Estimation and Filtering of Signal-Dependent Noise in Imaging

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB 109, at Tampere University of Technology, on the 11th of November 2016, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2016

Supervisor:**Associate Professor Alessandro Foi**

Department of Signal Processing
Tampere University of Technology
Tampere, Finland

Pre-examiners:**Assistant Professor Joseph Salmon**

Statistics and Applications group
TELECOM ParisTech
Paris, France

Charles Kervrann, PhD

IRISA / INRIA Rennes
Campus Universitaire de Beaulieu
Rennes, France

Opponent:**Associate Professor Yiqiu Dong**

Department of Applied Mathematics and Computer Science
Technical University of Denmark
Lyngby, Denmark

ISBN 978-952-15-3837-7 (printed)

ISBN 978-952-15-3887-2 (PDF)

ISSN 1459-2045

Abstract

The work presented in this thesis focuses on Image Processing, that is the branch of Signal Processing that centers its interest on images, sequences of images, and videos. It has various applications: imaging for traditional cameras, medical imaging, *e.g.*, X-ray and magnetic resonance imaging (MRI), infrared imaging (thermography), *e.g.*, for security purposes, astronomical imaging for space exploration, three-dimensional (video+depth) signal processing, and many more.

This thesis covers a small but relevant slice that is transversal to this vast pool of applications: noise estimation and denoising. To appreciate the relevance of this thesis it is essential to understand why noise is such an important part of Image Processing. Every acquisition device, and every measurement is subject to interferences that causes random fluctuations in the acquired signals. If not taken into consideration with a suitable mathematical approach, these fluctuations might invalidate any use of the acquired signal. Consider, for example, an MRI used to detect a possible condition; if not suitably processed and filtered, the image could lead to a wrong diagnosis. Therefore, before any acquired image is sent to an end-user (machine or human), it undergoes several processing steps. Noise estimation and denoising are usually parts of these fundamental steps.

Some sources of noise can be removed by suitably modeling the acquisition process of the camera, and developing hardware based on that model. Other sources of noise are instead inevitable: high/low light conditions of the acquired scene, hardware imperfections, temperature of the device, *etc.* To remove noise from an image, the noise characteristics have to be first

estimated. The branch of image processing that fulfills this role is called noise estimation. Then, it is possible to remove the noise artifacts from the acquired image. This process is referred to as denoising.

For practical reasons, it is convenient to model noise as random variables. In this way, we assume that the noise fluctuations take values whose probabilities follow specific distributions characterized only by few parameters. These are the parameters that we estimate. We focus our attention on noise modeled by Gaussian distributions, Poisson distributions, or a combination of these. These distributions are adopted for modeling noise affecting images from digital cameras, microscopes, telescopes, radiography systems, thermal cameras, depth-sensing cameras, *etc.* The parameters that define a Gaussian distribution are its mean and its variance, while a Poisson distribution depends only on its mean, since its variance is equal to the mean (signal-dependent variance). Consequently, the parameters of a Poisson-Gaussian distribution describe the relation between the intensity of the noise-free signal and the variance of the noise affecting it. Degradation models of this kind are referred to as *signal-dependent* noise.

Estimation of signal-dependent noise is commonly performed by processing, individually, groups of pixels with equal intensity in order to sample the aforementioned relation between signal mean and noise variance. Such sampling is often subject to outliers; we propose a robust estimation model where the noise parameters are estimated optimizing a likelihood function that models the local variance estimates from each group of pixels as mixtures of Gaussian and Cauchy distributions. The proposed model is general and applicable to a variety of signal-dependent noise models, including also possible clipping of the data. We also show that, under certain hypotheses, the relation between signal mean and noise variance can also be effectively sampled from groups of pixels of possibly different intensities.

Then, we propose a spatially adaptive transform to improve the denoising performance of a specific class of filters, namely nonlocal transform-domain collaborative filters. In particular, the proposed transform exploits the spatial coordinates of nonlocal similar features from an image to better decorrelate the data, and consequently to improve the filtering. Unlike non-adaptive transforms, the proposed spatially adaptive transform is ca-

pable of representing spatially smooth coarse-scale variations in the similar features of the image. Further, based on the same paradigm, we propose a method that adaptively enhances the local image features depending on their orientation with respect to the relative coordinates of other similar features at other locations in the image.

An established approach for removing Poisson noise utilizes so-called variance-stabilizing transformations (VST) to make the noise variance independent of the mean of the signal, hence enabling denoising by a standard denoiser for additive Gaussian noise. Within this framework, we propose an iterative method where at each iteration the previous estimate is summed back to the noisy image in order to improve the stabilizing performance of the transformation, and consequently to improve the denoising results. The proposed iterative procedure allows to circumvent the typical drawbacks that VSTs experience at very low intensities, and thus allows us to apply the standard denoiser effectively even at extremely low counts.

The developed methods achieve state-of-the-art results in their respective field of application.

Foreword

A lot of the things I learnt during the past four and an half years are summarized in this thesis, and most of them come from my supervisor Alessandro. It goes without saying, but I say it anyway, that it has all been possible thanks to him. He is the first person I wish to thank, because he has been the first to explain me the reasons why one should pursue a Ph.D., and he is the patient one who spent a lot of time explaining me the basics of Image and Signal Processing when I started my doctoral studies. I will always remember our meeting sessions, most of them when outside was dark... because it was late, very late.

On the same note, I am also thankful to Prof. Marco Carli, because he has been (first) an excellent teacher and (second) the one offering me the opportunity to visit Finland for the first time, to Prof. Atanas Gotchev, because he offered me my first position in the university as a *researcher*, and he is the one that informed me when a position for doctoral studies has been opened, and to Prof. Moncef Gabbouj for giving me the opportunity of working in the Nokia Research Center of Hermia.

The Department of Signal Processing has been a warm and stimulating working environment, mostly thanks to all my colleagues. I want to express my gratitude especially to Prof. Vladimir Katkovnik, who has been the kindest of the office roommates, and to Prof. Karen Egiazarian, because he has been an exceptionally friendly and welcoming figure. I also thank my colleagues and friends Misha, Aram, Robert and Tom.

Since its first draft, this thesis has seen major improvement due mostly to the invaluable contributions of the examiners Dr. Joseph Salmon and

FOREWORD

Dr. Charles Kervrann. Their comments were stimulating, and helped me to improve the level of the dissertation. However, they have also been very cordial, available, and punctual, allowing me to meet the pre-examination deadline. I want to also express my gratitude to Prof. Yiqiu Dong for accepting to serve as opponent for my defense. It is an honor to have my thesis examined and discussed by such respected professionals.

A special thanks goes to the university International HR Specialist, Ulla Siltaloppi. She has been the first person with whom I had a whole conversation in English (at least she had), and she has been always helpful, doing most of the times stuff that I was supposed to take care of myself. A big thanks also to the university staff that I have met during these years: Virve Larmila, Noora Rotola-Pukkila, Susanna Anttila, Pirkko Ruotsalainen, Johanna Pernu, and Elina Orava.

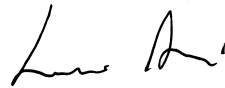
My research has been mostly financed by the Doctoral Programme of TUT's President, and has been entirely conducted at Tampere University of Technology. It has also been supported in part by the Academy of Finland (Project 252547). I am grateful for the financial support received during these years.

In these years I have noticed that the life of a Ph.D. student, although being very interesting and full of challenges, sometimes gets a bit lonely. For this reason I would like to thank the people that made this time pleasant, and with whom I have shared most of the time in University and outside. I am referring to Matteo, Roberto, Marco, Bruno, Waqar, Sergio, Stefano, Andrea, Davide, Ugur, and Dario. It has been extremely amusing spending time with you guys (we had the longest lunches). All of you are interesting and particular in your own way, and each one of you gave me something that I will always bring with me.

This is my sixth year in Finland, and the sixth year I am away from my homeland Italy. I really love living in Finland-Tampere-Hervanta, but sometimes I really miss my childhood friends with whom I still pleasantly enjoy my Christmas and summer holidays. It is wonderful to visit my hometown and find you there always greeting me with a smile on your faces and a ready joke. Thank you Roberto, Andrea, José, Piergiorgio & Francesca, Sirio & Manuela. You are and will always be special to me.

Speaking of *special* persons, I want to thank “minun pikku” girlfriend Johanna. You have been exceptionally helpful, especially during difficult times. I cannot describe how much I appreciate to go home and find a sunny face (in the cloudy Finnish winter) welcoming me, asking how my day was and telling me how hers was. It grows more and more the affection that I have for you, and without you this Ph.D. would have been a pretty dull journey.

Finally, I wish to thank my parents Beatrice and Valter. I am a very reserved person, and I take this opportunity to write what I have never told you. Thank you for everything you have done for me, and for everything you have given me. Looking back at when I was with you, I cannot think of anything I would change. Thank you, and I dedicate this thesis to you... and also to my silly brother Emanuele.



Tampere, September 2016

Contents

Abstract	i
Foreword	v
Contents	ix
List of publications	xiii
Notation and abbreviations	xv
1 Introduction	1
1.1 Scope of the thesis	2
1.2 Outline of the thesis	3
1.3 Author contribution	4
2 Preliminaries	5
2.1 Acquisition devices and noise sources	5
2.2 Adopted noise models	8
2.2.1 Additive white Gaussian noise (AWGN)	8
2.2.2 Poisson noise	9
2.2.3 Poisson-Gaussian noise	12
2.3 Image denoising methods	14
2.3.1 Denoising of additive Gaussian noise	14
2.3.2 Denoising of Poisson noise	18

2.4	Poisson-Gaussian noise estimation	20
2.4.1	Patch-based methods	21
2.4.2	Segmentation-based methods	22
2.4.3	Alternative approaches	23
2.5	Block matching and 3-D filtering (BM3D)	23
2.5.1	Grouping and transform	24
2.5.2	Collaborative filtering	24
2.5.3	Aggregation	25
2.5.4	Second iteration with Wiener filter	25
2.5.5	BM3D extensions	25
2.6	Quality metrics	26
3	Contribution to noise estimation	27
3.1	Affine noise variance estimation using heterogeneous data	28
3.1.1	Patch statistics and noise analysis	28
3.1.2	Outliers, robust estimators, and estimation errors	30
3.1.3	Procedure	31
3.1.4	Results	33
3.2	Robust noise estimation for any parametric variance function	35
3.2.1	Segmentation and local statistics	36
3.2.2	Clipping	37
3.2.3	Robust fitting	38
3.2.4	Results	39
4	Contribution to denoising	41
4.1	Collaborative filtering based on coordinates of similar features	42
4.1.1	Orthogonal polynomials transform	42
4.1.2	Directional enhancement	43
4.1.3	Results	45
4.2	Poisson image denoising	46
4.2.1	Inverse Anscombe transform	48
4.2.2	Noisy+estimate convex combination	49
4.2.3	Forward and inverse VST adopted	50
4.2.4	Binning	51

4.2.5	SNR as function of λ_i and h_i	52
4.2.6	Algorithm parameters	54
4.2.7	Results	54
5	Discussion and Conclusions	57
5.1	Automatic noise estimation and denoising	59
	Bibliography	61

List of publications

- I. Azzari, L. and Foi, A. (2014a). Gaussian-Cauchy mixture modeling for robust signal-dependent noise estimation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5357-5361.
- II. Azzari, L. and Foi, A. (2014b). Indirect Estimation of Signal-Dependent Noise With Nonadaptive Heterogeneous Samples. *IEEE Transactions on Image Processing*, 23(8):3459-3467.
- III. Azzari, L. and Foi, A. (2015). Collaborative filtering based on group coordinates for smoothing and directional sharpening. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1573-1577.
- IV. Azzari, L. and Foi, A. (2016c). Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising. *IEEE Signal Processing Letters*, 23(8):1086-1090.

Notation and abbreviations

In Table 1 and Table 2 we report the most common abbreviations and mathematical notation used in this thesis.

1-D,2-D,3-D	One-, Two-, Three- Dimensional
AWGN	Additive White Gaussian Noise
BM3D	Block Matching and 3-D filtering
CCD	Charge Coupled Semiconductor Devices
CMOS	Complementary Metal-Oxide Semiconductor
dB	Decibel
DCT	Discrete Cosine Transform
i.i.d.	independent and identical distributed
MAD	Median of Absolute Deviation
ML	Maximum Likelihood
MSE	Mean Squared Error
NLM	Nonlocal Means
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index
VST	Variance-Stabilizing Transformation

Table 1: List of Abbreviations.

y	Noise-free data
z	Noisy data
\hat{y}	Estimate of y
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$\mathcal{P}(\lambda)$	Poisson distribution with mean and variance λ
$E\{z\}$	Expected value of z
$E\{z y\}$	Expected value of z given y
$\text{var}\{z\}$	Variance of z
$\text{var}\{z y\}$	Conditional variance of z given y
$\text{std}\{z\}$	Standard deviation of z
$\text{std}\{z y\}$	Conditional standard deviation of z given y

Table 2: List of Mathematical Notation.

Chapter 1

Introduction

An acquisition device, such as a camera, is an instrument used for measuring and quantifying a specific physical attribute. When a device is acquiring a signal, its sensors are measuring the quantity of the physical attribute interacting with them. In case of pictures, for example, the device measures the energy of the light beams incident on the sensors. Like any other measurements, image measurements are subject to fluctuations due to both internal and external factors. The most influential among these factors are commonly referred to as noise, and produce visible artifacts that we model as realizations of random variables that follow a specific mathematical model, and that affect a *clean* deterministic signal. The knowledge of the parameters defining the model allows us to optimally remove the aforementioned artifacts, delivering a restored signal. The estimation of these parameters is referred to as noise estimation, and the removal of the artifacts is referred to as denoising.

In the particular case of imaging devices, an image is the result of the acquisition process of a rectangular array of sensing elements. A sensing element generates an electric charge proportional to the number of incident photons [Gonzales and Wintz, 1987]; this acquisition step is commonly called photon-counting process. Each resulting electric potential is then quantized and stored as an integer number that will represent a pixel

value in the final data array that constitutes a digital image. The two most common classes of imaging sensors for visible spectrum are *Charge Coupled Semiconductor Devices* CCD [Boyle and Smith, 1970] and *Complementary Metal-Oxide Semiconductor* CMOS [Noble, 1968]. Other important imaging technologies for the non-visible spectrum are the *Flat Panel Detectors* (FPD) [Jaffray et al., 2002] and *Thermal Kinetic Inductance Detectors* (TKIDs) [Cecil et al., 2015] for x-ray imaging, *Magnetic Resonance Imaging* (MRI) [Damadian, 1971], the *Long Wave Infrared* (LWIR) sensors [Hackwell et al., 1996] for thermography and hyperspectral imaging, *Continuous-Wave Light Detection and Ranging* (CW-LiDAR) [Bashkansky et al., 2004] and *Time-of-Flight* (ToF) [Lange and Seitz, 2001] sensors for depth cameras, etc.

A clean signal is usually corrupted by noise from multiple sources; for example, there is a non-zero probability that an incident photon does not generate an electric charge [Koczyk et al., 1996; Fowler et al., 1998]; the modeling of the photon counting process, therefore, has to take into consideration this type of uncertainty. Other sources of noise are commonly present in standard acquisition devices, and are usually independent from each other, e.g., dark current (also known as thermal noise) [Gonzales and Wintz, 1987; Johnson, 1928; Nyquist, 1928]. If the noise produced by these sources are not included in the denoising model, they will eventually affect the quality of the output image.

1.1 Scope of the thesis

Every acquired image has to undergo throughout a series of modular processing stages; noise estimation and denoising are the steps of this chain upon which we focus the interest of this thesis. In particular, we develop new and effective models and methods for estimation of signal-dependent noise and image denoising.

The common framework for signal-dependent noise estimation first 1) builds a scatterplot from the statistics of different parts of a noisy image, and 2) then estimates the noise parameters by fitting a curve over the scatterplot.

Our contributions to noise estimation address both steps: we propose a novel model for the estimation of the local statistics, and a robust model for the scatterplot fitting.

In image denoising, our contributions address the design of adaptive non-local transforms in collaborative filters, as well as the development of iterative algorithms for signal-dependent noise removal using variance-stabilizing transformation (VST).

Overall, we aim at improving different steps of the image denoising pipeline, and the presented methods can be ultimately combined to improve the removal of signal-dependent noise.

1.2 Outline of the thesis

In Chapter 2 we discuss the main sources of noise in imaging, we introduce the observation models that we use throughout the thesis, and we give a brief summary of the prior works most relevant to this thesis. In Chapter 3 we deal with the noise estimation problem. We introduce two methods for estimating the noise affecting an image. In particular, we investigate the case of signal-dependent noise, in which the noise variance depends on the signal intensity. We approach this problem from two distinct standpoints; we first study the possibility of estimating affine noise variance using heterogeneous samples [Publication II], then we propose a robust estimator for generic (also non-affine) signal-dependent noise variance (that uses instead homogeneous samples) [Publication I]. In Chapter 4 we tackle the problem of denoising. We first introduce a denoising algorithm that exploits the spatial coordinates of nonlocal similar features of an image to generate a spatially adaptive basis that is then used to transform and filter the data [Publication III]. Based on the same spatial information, we also incorporate in the algorithm an adaptive enhancement of the image. Then, we propose an iterative algorithm combined with variance stabilization techniques for denoising Poisson noise. At each iteration we combine the noisy observation with the previous estimate to increase the signal-to-noise-ratio (SNR) of the signal that we stabilize, increasing therefore the accuracy of the stabi-

lizing transformation [Publication IV], especially for low SNR. Finally, the conclusions on the thesis are given in Chapter 5.

The proposed ideas and algorithms contribute to advance the state of the art in both noise estimation and denoising.

1.3 Author contribution

The author of the thesis has actively contributed to the development of the ideas, models, and methods introduced in Publications I-IV, on which this thesis is based. Furthermore, he is also the first author of all these publications, taking care of the manuscript writing and of the implementation and experimental validation the proposed methods, under the supervision of Dr. Alessandro Foi.

Chapter 2

Preliminaries

This chapter provides general concepts that we use throughout this thesis. In Section 2.1 we describe the main sources of noise in an imaging acquisition device. Section 2.2 presents observation models for: additive white Gaussian noise (AWGN), Poisson noise, and Gaussian-Poisson noise. In Section 2.3 we analyze some relevant methods for Gaussian and Poisson noise removal, while in Section 2.4 we analyze relevant methods for Poisson-Gaussian noise estimation. In Section 2.5 we give a more detailed description of the Block Matching and 3-D Filtering (BM3D) algorithm, since it is often referred to in this thesis. Finally, in Section 2.6, we introduce the quality metrics adopted to validate the proposed ideas and quantify the performance of the proposed methods.

2.1 Acquisition devices and noise sources

Raw images are far from their final end-user form, and have to be processed to improve their quality before any further operation and/or visualization. There are several steps (modules) that a raw image has to go through: white balancing, demosaicing, denoising, optical correction, contrast enhancement, color transformation, and tone mapping. For the interest of this thesis we consider only the denoising block, that usually includes a

prior noise estimation step.

To model correctly the noise affecting a signal, it is necessary to comprehend the functioning of the acquisition device, and opportunely identify the sources of the noise itself. An image is generated by converting the light coming from a natural scene to digital data. In particular, a camera performs this conversion using a semiconducting array of sensing elements positioned after an aperture: when the shutter opens, the light from the scene goes through the lenses and the aperture, finally colliding with the sensors. This array converts the energy of the incident light beams to electric charges, that are successively accumulated in an electric potential. The electric potentials are then converted to digital values, and finally stored collectively as a raw image, whose pixel intensities correspond to the energy accumulated by each sensing element.

The most common digital camera sensors are Charge Coupled Semiconductor Devices (CCD) and Complementary Metal-Oxide Semiconductor (CMOS). While most of the modern devices have CCD sensors, CMOS sensor technology is becoming more and more popular due to its flexibility [Jähne, 2004]. The main difference between the two is that, while in CCD arrays the charge of a row of sensors is transported via the same circuit, sharing also the same amplifier, CMOS arrays are based on the Active Pixel Sensor (ASP) technology, for which every single sensor is treated independently, having a unique transport line. Although there are clear differences, the sensors share the same light acquisition procedure, since they are both silicon photosensitive diodes; therefore, we consider a single model to represent the noise affecting both sensors.

To understand the nature of the noise from the sensors, let us consider the acquisition of a still scene; although the average incident energy over a relatively long period of time might be constant (constant brightness), the amount of photons incident on the camera sensors during the exposure fluctuates with time. Furthermore, not all the incident photons are converted to electric charge. Therefore, taking several pictures of the same still scene does not give identical images. This whole phenomenon is known as shot noise, and is well modeled by the family of Poisson distributions [Mandel, 1959].

Another relevant source of noise is the so-called thermal noise. Thermal noise is generated by thermal agitation [Nyquist, 1928; Johnson, 1928], and is due to the fact that at any given temperature (except absolute zero), conductors have a probability to emit charges due to heat, even when there is no electric potential to stimulate them. This results in a background current, present also in absence of input signals (dark current) [Jähne, 2004], which alters the measurements of the sensors. The inevitable fluctuations of this current are thus modeled as noise. By definition, this type of noise is proportional to the working temperature of the device, and it could be reduced by decreasing the surrounding temperature of the electric circuits; however, in most applications this solution is unacceptable or unfeasible, and therefore we have to find alternative ways to suppress this sort of artifacts.

Flicker noise is a type of noise with power spectral density inversely proportional to the frequency, and it is present in all electronic devices. It is also known as $1/f$ noise [Weissman, 1988] or pink noise, from the corresponding power distribution in the light-wave color spectrum. A consequence of its power distribution is that the variance of the noise is large at low frequency, while being small at higher frequency.

The last types of noise sources that we investigate are those due to the imperfections of the electronic components. For example, the sensing elements of a sensor might have slightly different photoelectric responses, or the amplifiers of a sensor might have slightly different gains. These nonuniformities are referred to as readout noise and Fixed-Pattern Noise (FPN). Readout noise is due to the amplification and analog-to-digital conversion, and it is a particular type of FPN. In general, a fixed pattern noise is a type of noise that repeats itself over time with the same structure, and could be corrected by using dark frames, *i.e.* pictures taken without exposure to light.

2.2 Adopted noise models

2.2.1 Additive white Gaussian noise (AWGN)

From Section 2.1 it is clear that a signal acquired by a digital device is affected by noise from several sources. It is neither reasonable, nor convenient, to try to separate and treat each noise source individually; therefore, we conventionally group them together and we address them as a single noise. This procedure is theoretically supported by the Central Limit Theorem (CLT) [Tijms, 2007; Papoulis and Pillai, 2002]. Considering a set of N independent random variables X_1, \dots, X_N , with respective means μ_1, \dots, μ_N and standard deviations $\sigma_1, \dots, \sigma_N$, the sum

$$\frac{1}{s} \sum_{i=1}^N (X_i - \mu_i) \xrightarrow{d} \mathcal{N}(0, 1) \quad \text{with} \quad s = \sqrt{\sum_{i=1}^N \sigma_i^2}, \quad (2.1)$$

when $N \rightarrow \infty$, with \xrightarrow{d} denoting a convergence in distribution. Therefore, we can represent the sum of the noise from various sources as a Gaussian random variable.

The Gaussian noise is further commonly assumed additive, independent and identically distributed (i.i.d.); with this assumption we can model our observed image z as

$$z(x) = y(x) + \eta(x), \quad (2.2)$$

where y is the noise-free image that we want to recover, $x \in \Omega \subset \mathbb{Z}^d$ is the d -dimensional spatial coordinate of the signal ($d = 2$ for images), and $\eta \sim \mathcal{N}(0, \sigma^2)$ is the zero-mean random variable that models the additive white Gaussian noise (AWGN) affecting y . From (2.2) we find that

$$\mathbb{E}\{z(x) \mid y(x)\} = y(x) \quad \text{and} \quad \text{var}\{z(x) \mid y(x)\} = \sigma^2, \quad (2.3)$$

where $\mathbb{E}\{\cdot\}$ and $\text{var}\{\cdot\}$, are, respectively, the expectation and variance of the value in-between curly brackets. Clearly this type of noise is signal-independent, in the sense that its standard deviation does not depend on the noise-free signal y , and it has a flat power spectrum.

The above simplifications make the problem of denoising more convenient to solve, and it allows us to use simpler denoising tools. However, we have made strong assumptions, and one can object the validity of the model. In particular, signal-dependent noises, such as shot noise, is not taken into account by this model. Nonetheless, there are cases in which the noise sources can be assumed signal-independent [Nakamura, 2005], where this model holds and is very advantageous.

2.2.2 Poisson noise

The conventional way to model data corrupted by Poisson noise (signal-dependent noise) is to represent the observed image z as realizations of independent Poisson random variables. In particular, z is modeled as

$$z(x) \sim \mathcal{P}(y(x)),$$

$$P(z(x) | y(x)) = \begin{cases} \frac{y(x)^{z(x)} e^{-y(x)}}{z(x)!} & z(x) \in \mathbb{N} \cup \{0\} \\ 0 & \text{elsewhere,} \end{cases} \quad (2.4)$$

where y is the noise-free image to be estimated. Therefore, in (2.4) we consider each pixel as an independent realization of a Poisson random variable whose mean and variance are the noise-free signal y that we want to estimate

$$E\{z(x) | y(x)\} = \text{var}\{z(x) | y(x)\} = y(x). \quad (2.5)$$

From (2.5) we can observe an important property of Poisson noise. Since the variance is equal to the mean of the signal, there is a square root dependency between mean and standard deviation. This implies that the signal-to-noise ratio SNR, *i.e.* ratio between noise-free signal energy and noise energy, increases when the intensity of the noise-free signal y increases

$$\text{SNR} = 10 \log_{10} \frac{y^2(x)}{\text{var}\{\text{noise}\}} = 10 \log_{10} y(x). \quad (2.6)$$

Since denoising is easier for large SNR scenarios, denoising of Poisson images becomes quite challenging for low intensities, where the noise is relatively stronger compared to larger intensities.

Variance-stabilizing transform (VST)

As mentioned above, in a Poisson image each pixel is affected by noise with variance equal to the mean of the pixel intensity. This dependency has to be taken into consideration when developing a denoiser for such images. In literature, one of the most common way to denoise Poisson images is by variance-stabilizing transformation (VST). As the name suggests, a VST is a point-to-point transformation applied to the data, that makes the resulting signal affected by noise with constant variance (signal-independent). As a result from this transformation, we can use standard denoising filters for additive noise, like those listed in Section 2.2.1, to denoise the Poisson image. Applying then an inverse VST, we obtain a denoised image that is considered the final estimate. This framework has been the backbone of Poisson denoising for a long time, and many works based on the VST+(AWGN denoising) filters can be found in the literature.

Let us consider a Poisson random variable z whose mean and standard deviation are, respectively, $E\{z | y\} = y$ and $\text{std}\{z | y\} = \sigma(y)$. Denoting with f a generic VST function, ideally, we would like to obtain $\text{std}\{f(z) | y\} = c$, with $c > 0$ being a constant value. We obtain the first order of the Taylor polynomial approximation of $f(z)$ around the value y as

$$f(z) \approx f(y) + (z - y) f'(y). \quad (2.7)$$

Computing the standard deviation of both members of (2.7), we obtain

$$\text{std}\{f(z) | y\} \approx \sigma(y) f'(y), \quad (2.8)$$

since y is deterministic, *i.e.* $\text{std}\{y\} = 0$. By imposing the standard deviation equal to the desired constant c , *i.e.* $\text{std}\{f(z) | y\} = c$, we find an approximate analytical expression of the VST as primitive function:

$$f(z) = \int_0^z \frac{c}{\sigma(y)} dy. \quad (2.9)$$

Equation (2.9) can be used to find a VST for Poisson random variables.

Indeed, by substituting $c = 1$ and $\sigma(y) = \sqrt{y}$ we can solve the integral

$$f(z) = \int_0^z \frac{1}{\sqrt{y}} dy = 2\sqrt{z}. \quad (2.10)$$

This is a generic result that has not been used only in signal processing; in fact, the stabilizing performances of the VST in (2.10) were already studied by [Bartlett \[1936\]](#), while [Curtiss \[1943\]](#) proves that (2.10) is an asymptotic stabilizer of the Poisson distribution; in this scenario asymptotic means that the variance of the transformed signal converges to a constant as $y \rightarrow +\infty$. However, in the same work, the author shows that there are better expressions of VST for Poisson variance stabilization, with faster rate of convergence. More specifically, he proposes a VST of the form $f(z) = 2\sqrt{z + 1/2}$. Finally, [Anscombe \[1948\]](#) proves that the optimal constant to be summed to z is $3/8$:

$$f(z) = 2\sqrt{z + \frac{3}{8}}. \quad (2.11)$$

This function is known as the Anscombe transform, and the optimality is intended in terms of asymptotic rate of convergence for large values of intensity. In his work, [Anscombe \[1948\]](#) gives the asymptotic expression of the variance of the transformed variable $f(z) = 2\sqrt{z + \beta}$ for a generic constant $\beta \geq 0$,

$$\text{var}\{f(z)\} = \left\{ 1 + \frac{3 - 8\beta}{8y} + \frac{32\beta^2 - 52\beta + 17}{32y^2} \right\} + \mathcal{O}\left(\frac{1}{y^3}\right). \quad (2.12)$$

By setting $\beta = 3/8$, (2.12) becomes

$$\text{var}\{f(z)\} = \left\{ 1 + \frac{1}{16y^2} \right\} + \mathcal{O}\left(\frac{1}{y^3}\right), \quad (2.13)$$

proving that $\beta = 3/8$ is the optimal value for convergence to 1 in case of large values of mean y . For small value of intensity, *i.e.* $y \leq 2$, the Anscombe transform becomes inaccurate.

Although many other VST have been proposed [[Freeman and Tukey, 1950](#); [Bar-Lev and Enis, 1988, 1990](#); [Foi, 2009](#)], in our implementations we use as VST the Anscombe transform.

Inverse transformation

An important aspect of VST algorithms is the inverse transformation. In fact, once the signal has been transformed and denoised, it is contaminated by bias due to the non-linearity of the forward VST. The design and computation of an unbiased inverse transformation, which maps the data back to its original range, can be challenging, and exact solutions became available only recently [Mäkitalo and Foi, 2011]. Further details are given in Section 4.2.1.

2.2.3 Poisson-Gaussian noise

The Poisson-Gaussian noise model is given by the sum of two independent sources of noise, one whose variance is signal-dependent (Poisson), and one whose variance is signal-independent (Gaussian). Its formal model is

$$z(x) = \alpha p(x) + n(x), \quad (2.14)$$

where

$$p(x) \sim \mathcal{P}(y(x)) \quad \text{and} \quad n(x) \sim \mathcal{N}(0, b), \quad (2.15)$$

and α is a constant $\alpha \geq 0$. From (2.14) we define the Poisson-Gaussian noise η_{pg} as

$$\eta_{pg}(x) = z(x) - \alpha y(x). \quad (2.16)$$

Although (2.14) is the rigorous Poisson-Gaussian model, for our applications we consider an alternative approximation model that reformulates $z(x)$ as the sum of a deterministic signal $y(x)$ and the random noise $\sigma(y(x))\xi(x)$

$$z(x) = y(x) + \sigma(y(x))\xi(x), \quad (2.17)$$

where ξ is a zero-mean random variable with unitary variance, *i.e.* $E\{\xi\} = 0$ and $\text{var}\{\xi\} = 1$, and σ is a univariate function (so called standard deviation function), which gives the signal-dependent standard deviation of the

noise as a function of the deterministic noise-free signal $y(x)$. We also assume the noise model in (2.17) as heteroskedastic Gaussian, *i.e.* a Gaussian distribution whose variance depends on the noise-free signal $y(x)$:

$$\sigma(y(x))\xi(x) \sim \mathcal{N}(0, \sigma^2(y(x))). \quad (2.18)$$

This simplification is validated by the fact that, for large values of mean, the Poisson distribution is well approximated by the Gaussian distribution $\mathcal{P}(y) \approx \mathcal{N}(y, y)$. Since we estimate noise affecting images whose range of intensities (usually $y \in [0, 255]$) is mostly confined where the approximation is valid, we can express the noise distribution as in (2.18).

Trivially,

$$\mathbb{E}\{z(x)\} = y(x) \quad \text{and} \quad \text{std}\{z(x)\} = \sigma(y(x)). \quad (2.19)$$

We finally give the simplified variance expression

$$\sigma^2(y(x)) = ay(x) + b, \quad (2.20)$$

where a represents the signal-dependent part of the variance, while b the independent one. Although higher order of approximation could be used, in this thesis we model the noise variance (2.20) as an affine function of the signal mean, since it well represents noise affecting the digital acquisition devices most interesting for the thesis, such as digital cameras and microscopes. Nonetheless, as in [Sutour et al., 2015], the algorithm presented in [Publication I](#) supports any order of approximation of the noise variance, and can be also used in scenarios where a higher order of approximation is needed.

Variance stabilization via generalized Anscombe transformation

Similar to the case of Poisson images, the Gaussian-Poisson noise can be stabilized using the generalized Anscombe transform [Starck et al., 1998] f_g . In particular, the noise in (2.14) is stabilized via the following transformation

$$f_g(z) = \begin{cases} \frac{2}{\alpha} \sqrt{\alpha z + \frac{3}{8}\alpha^2 + b - \alpha\mu}, & z > -\frac{3}{8}\alpha - \frac{b}{\alpha} + \mu \\ 0, & z \leq -\frac{3}{8}\alpha - \frac{b}{\alpha} + \mu, \end{cases} \quad (2.21)$$

where μ is the expected value of the Gaussian distribution.

2.3 Image denoising methods

In this chapter we give a brief and non-exhaustive review of some of the most significant works on Gaussian and Poisson noise removal.

2.3.1 Denoising of additive Gaussian noise

Although there are many criteria that could be used to classify AWGN denoising methods, we decide to distinguish three, partly overlapping, classes of algorithms: local algorithms, nonlocal algorithms, and transform-domain algorithms.

Local algorithms

As the name suggests, local methods denoise each pixel of an image by exploiting the information from the pixels spatially close to it, *i.e.* its local neighbourhood. The first local methods relevant to the thesis are presented in [Nadaraya, 1964; Watson, 1964], where the authors estimate a reference pixel via a weighted average of its neighborhood: the weights are computed with a Gaussian kernel (window) whose arguments are the Euclidean distances between reference pixel and neighbors. Since the weights depend on the spatial coordinates of the neighbors but not on the pixel intensities, pixels of the image that belong to non-uniform areas are usually poorly estimated, because neighbors may have different intensities. Smooth intensity variations can be dealt with by the *local polynomial approximation* (LPA) methods, that estimate the reference pixel by finding a polynomial function that best fits its windowed neighborhood [Stone, 1977; Cleveland and Devlin, 1988; Fan and Gijbels, 1996; Katkovnik et al., 2006].

Even though LPA methods improve the estimate in non-uniform areas of the image, the size of the support window does not adapt to each specific neighbourhood or discontinuities. Katkovnik et al. [2004] propose a method to determine the pointwise adaptive size of directional windows in image

denoising. This method is based on the intersection of confidence intervals (ICI) method first proposed in [Goldenshluger and Nemirovski, 1997].

Another relevant adaptive local denoising algorithm is the bilateral filter [Tomasi and Manduchi, 1998]; this algorithm denoises a reference pixel with a weighted average of its neighborhood, in which the weights are computed from a bi-variate Gaussian kernel whose arguments are, respectively, the Euclidean distance and difference in intensity between reference and neighbor. In this way the neighbors that are closer spatially and in intensity have more weight compared to the others, improving the filter performance in edges and texture of an image.

Nonlocal algorithms

The idea behind nonlocal algorithms is that the information required to denoise a pixel is not necessarily in a close surrounding of the pixel under interest, but could be found anywhere in the noisy image. Although already formulated in 1997 [De Bonet, 1997], the first relevant work on non-locality is the *NonLocal Means* (NLM) algorithm [Buades et al., 2005]. There, the authors propose to replace the current reference pixel with a weighted average of the pixels whose neighborhood is similar to the neighborhood of the pixel to be estimated. Similar neighborhoods give larger weights, while dissimilar neighborhoods have smaller weights. The similarity between two neighborhoods is measured via ℓ_2 norm of their intensities. In principle, the whole image can be used to denoise a single pixel, but for practical reasons the search of similar neighborhoods is restricted to a smaller region. The difference with the local algorithms is clear: non-local algorithms use a collection of neighborhoods for denoising, instead of a single one. In the same year Awate and Whitaker [2005] propose a similar patch-oriented nonlocal method, called UINTA, that denoises an image by minimizing a penalty function based on the entropy of patterns of intensities from regions of the image with similar neighborhood.

Several variations of NLM can be found in literature. For example, in [Kervrann and Boulanger, 2006, 2008; Deledalle et al., 2012] the authors improve the filtering results by optimizing search areas, aggregation

weights, and shape of the patches. A particular example is the SAFIR algorithm [Boulanger et al., 2008], that optimizes the search window based on the estimation error and on the estimates at previous iteration. Buades et al. [2013], furthermore, present a variation of NLM in which the covariance of a group of similar blocks is used to increase the performance of the nonlocal weighted mean. Nonlocality is also exploited for video denoising, like Liu and Freeman [2010] do in their work; there, the redundancy of information is considered also in the temporal direction, extending the research of similar neighbours to different frames of a video.

Another category of nonlocal filters developed from NLM are the collaborative filters; compared to NLM, that exploits nonlocal information to denoise one pixel per time, collaborative filters process together nonlocal parts of an image, denoising them jointly at the same time [Dabov et al., 2006, 2007b,b]. We dedicate the entire Section 2.5 to the description of the collaborative Block Matching 3-D (BM3D) filter algorithm of Dabov et al. [2006], because it is the denoising filter for additive noise that we mostly refer to throughout the thesis.

Transform-domain algorithms

Transform-domain filters perform the so called denoising in sparse domain [Donoho and Johnstone, 1995; Donoho, 1995]. The idea is to represent the observed signal in a domain in which its noise-free version is sparse, *i.e.* most of the energy of the noise-free signal is confined in a small amount of transform coefficients. Since the observed noisy signal is affected by noise with a flat power spectrum, most of its transform coefficients are different from zero due to the effect of noise, even if a sparse transform is applied. To perform denoising in transform domain, a shrinkage operator is commonly applied to the transform coefficients in order to suppress the coefficients that are different from zero due to noise. Because sparse transform coefficients that well-represent the signal have usually much larger magnitude than those that only depend on the noise, hard-thresholding to zero the small magnitude coefficients is usually the simplest and sometimes most efficient shrinkage operator.

The first transforms used for transform-domain filtering have been the Discrete Fourier Transform (DFT) and the Discrete Cosine Transform (DCT). The main advantages of these transforms is that, especially for relatively small supports (*e.g.*, 8×8 or 16×16 blocks), the transform coefficients are very sparse, *i.e.* only few are different from zero. This is due to the fact that, using a small support, the majority of the observed blocks can be approximated smooth, and are well represented by a small amount of transform coefficients. Between DFT and DCT it is common practice to opt for the DCT, due to its periodicity [Gonzales and Wintz, 1987].

An example of DCT transform-domain filter is the sliding DCT algorithm [Oktem et al., 1998; Yaroslavsky et al., 2001]. A sliding window is shifted along the observed image; at each step the authors compute the DCT transform of the block, and then they perform shrinkage. The inverse transformed block is finally considered an estimate of the noise-free block. Since patches estimated at different steps usually overlap, the algorithm also introduces the aggregation of overlapping patches, that is performed via point-wise averaging (or weighted averaging) of overlapping pixels. An interesting improvement of sliding DCT is the shape adaptive DCT (SA-DCT) algorithm [Foi et al., 2007, 2006]. There, LPA is combined with the ICI rule in order to compute the optimal shape of the patch for each specific region of the image. The patch is successively transformed with a shape adaptive-DCT transform, and filtered via hard-thresholding. Furthermore, the authors introduce also a second stage filtering, in which a Wiener filter is applied using as pilot signal the previous estimate.

Multiscale transforms constitute a large subcategory among the transform-domain denoising algorithms. The general idea is to represent the observed signal in a transform domain at different scales. In this way it is possible to process (parts of) the image with filters at different scales. This allows to correctly smooth large smooth areas, while still preserving details and texture from the original image. The most convenient multiscale transform used in literature is the wavelet transform. The wavelet basis is usually generated from a single *mother wavelet* function that is scaled and shifted in order to obtain an orthogonal basis used for the transformation. The first wavelet was introduced in 1911 by Haar [1911], and since

then many others have been proposed, for example, by Daubechies [1988], Beylkin et al. [1991], and Coifman and Meyer [1991]. Recently, more sophisticated transforms have been proposed for multiresolution denoising: directional wavelets [Antoine et al., 1996], curvelets [Starck et al., 2002], ridgelets [Starck et al., 2002], shearlets [Labate et al., 2005], *etc.* The main difference with standard wavelets is that they use particular basis capable of preserving specific features of an image, like straight lines, curves, or edges.

Elad and Aharon [2006] develop a transform-domain algorithm that exploits singular value decomposition (SVD) and K-means: the K-SVD denoising algorithm. There, the authors propose a two-stage iterative algorithm; in the first stage, atoms for sparse representation of the image are computed via K-means, while in the second stage the dictionary is updated applying SVD decomposition of the atoms. Denoising is performed by representing the estimated signal using only few atoms, corresponding to the larger eigenvalues from the SVD.

Notably, BM3D [Dabov et al., 2007b] can be treated as a special *nonlocal transform-based* method, since it performs denoising via shrinking of the transform coefficients of groups of similar blocks.

2.3.2 Denoising of Poisson noise

In this section we present the most representative methods for Poisson image denoising. We divide these algorithms in two main categories: VST based, and direct denoisers. While the first category applies variance stabilizing transformation before denoising, the second directly filters the Poisson image.

VST-based methods for Poisson images denoising

Exploiting the variance stabilization concept introduced in Section 2.2.2, Poisson denoising algorithms based on VST have been proposed extensively during the course of the years. The idea is to stabilize the variance of the noise with a VST (*e.g.*, Anscombe transform), and to denoise the resulting

signal with a filter for additive noise. Then, an inverse transformation is applied.

Fryzlewicz and Nason [2004] stabilize Poisson images via Haar-Fisz transform, that is a wavelet transform (generated from the better known Haar transform) that stabilizes the Poisson distribution into a Gaussian one, especially effective for low intensity values. The authors exploit this property to stabilize the noisy signal, and to treat it as an additive noise. Zhang et al. [2008] introduce a multi-scale variance stabilizing transform (MS-VST) that outperforms [Fryzlewicz and Nason, 2004]. There, the authors propose to decompose a signal via a generic filter-bank of wavelets, ridgelets and curvelets. Each representation of the signal is then stabilized via an ad-hoc VST (computed from the Anscombe transform), and then filtered using an AWGN filter. Each transform sub-band is therefore interpreted as a signal affected by signal-dependent noise to stabilize with an adaptive transform.

Although many alternatives have been proposed, the Anscombe VST remains appealing for a wide audience within the scientific community, for its simplicity and practicality. Boulanger et al. [2008], for example, adopt the Anscombe transform to denoise Poisson data. Even though based on the simple Anscombe VST, these methods achieve state-of-the-art results in terms of image quality. For this reason, we use the same framework in our approach to Poisson image denoising, and we develop an iterative scheme to cope with the low-intensity case mentioned in [Fryzlewicz and Nason, 2004].

Direct Poisson images denoising

While VST algorithms adapt AWGN denoising algorithms to denoise Poisson images, in literature we also find specific algorithms for direct Poisson image denoising. In this section we analyze those that we will later on use to compare the performances of the proposed Poisson denoising scheme.

The first algorithm that we discuss is the Non-Local Sparse-Principal Component Analysis (Poisson-NLSPCA) [Salmon et al., 2014]. There, the authors combine the Poisson-PCA [Collins et al., 2001] with a sparse dictio-

nary learning technique for Poisson data, to iteratively converge to a final estimate. In particular, the image is divided in overlapping patches, that are successively grouped into clusters via K-means. Then, a dictionary is found for each cluster, and a sparse representation of the patches is computed. This allows to denoise the data directly, without any variance stabilization.

Giryès and Elad [2014] propose an iterative Sparse Poisson Denoising Algorithm (SPDA) similar to [Salmon et al., 2014], but they use a global dictionary for all the patches from the entire image, and introduce a novel greedy technique for the computation of the sparse representation of each patch. There, the authors also impose that similar enough patches share the same sparse coefficients.

Based on the work by Danielyan et al. [2010], another algorithm recently proposed is the Plug-&-Play Prior for Poisson Inverse Problem (P⁴IP) [Rond et al., 2015]. In their work the authors propose a Plug-&-Play framework [Venkatakrisnan et al., 2013] solved iteratively by the alternating direction method of multipliers (ADMM) algorithm [Boyd et al., 2011]. The denoising part of this method is performed using a Gaussian denoising filter plugged directly into the processing chain of P⁴IP. Indeed, in a loose sense, this algorithm could be even considered a VST algorithm in which the VST is iteratively refined to adapt to the data to be denoised.

The algorithm in [Pyatykh and Hesser, 2015] processes and denoises, individually, patches extracted from the image using a minimum mean squared error (MMSE) estimator performed by k-d tree search [Muja and Lowe, 2014] combined with a K-nearest neighbor graph search.

Oh et al. [2014] present instead an approach in which the estimation is performed by minimizing a cost function whose regularization term is an hybrid function of the image and its logarithm, depending on the local intensity of the image itself.

2.4 Poisson-Gaussian noise estimation

The main purpose of noise estimation algorithms is to estimate the parameters a and b of the function (2.20). The most common framework first

builds a scatterplot with mean values of the signal on the abscissa, and corresponding standard deviations (variances) on the ordinate; then, it fits a curve over these points to estimate the function parameters. It is common practice to compute each scatter-point from homogeneous samples, *i.e.* each element in a sample share a unique common expectation value (hence they also share a common unique variance value). This practice is based on the fact that the sample variance of homogeneous samples is an unbiased estimator of the noise variance for that particular expectation value. Consequently, each point in the scatterplot refers to a specific point of (2.20), and therefore has a direct relation with the curve whose parameters we want to estimate. We first analyze relevant algorithms that adopt a *patch-based* approach, where each scatter-point is estimated from the statistics of a single patch from the noisy image; then, we analyze relevant *segmentation-based* approaches, where each scatter-point is estimated from segmented (homogeneous) elements extracted from the whole noisy image. Finally, we introduce a category of algorithms that does not compute a scatterplot, but instead use alternative methods to the aforementioned framework to estimate the noise parameters.

2.4.1 Patch-based methods

A relevant *patch-based* algorithm is introduced by Lee and Hoppel [1989], that divide the image into small patches, and directly compute their mean and variance to build the scatterplot of mean-variance pairs. Since image blocks might contain heterogeneous elements that would mislead the estimation of the local variances, the authors estimate the noise parameters by finding the curve that intersect most of the scatter-points. In this way they reduce the effect of outliers that usually appear far from the majority of the scatter-points. In a similar work, Amer and Dubois [2005] evaluate, using directional derivative filters, the uniformity of each patch that generated a data point. Comparing the uniformity against a threshold, the algorithm decides whether to use the scatter-point (if the patch elements are homogeneous) or to discard it. Finally, since the outliers have been already excluded, a simple least square (LS) fitting is adopted. In [Sutour

et al., 2015] the authors divide the image in non-overlapping blocks; based on the Kendall's τ coefficients, adopted to find the correlation between elements from the same block, the blocks are then classified as homogeneous or heterogeneous. The heterogeneous blocks are discarded, while the homogeneous ones are used to compute the local statistics for the fitting of the mean-variance curve. An important aspect of the algorithm is that a robust fitting is performed by minimizing the ℓ_1 error of the residuals. Similar works can be found in [Meer et al., 1990; Lee, 1981; Mastin, 1985], however we decide not to go into further details.

An interesting variant has been proposed by Boulanger et al. [2010]. They divide the image into adaptive blocks whose size depend on the variance of their elements (homogeneity). If the variance of a block matches the variance model (Fisher test is used to compare the two), then the block is considered homogeneous, otherwise the block is further split into four parts and each *sub*-block is then analyzed as before. Finally, the authors perform noise parameters estimation via robust linear regression of the local estimates.

2.4.2 Segmentation-based methods

We now describe the most relevant *segmentation-based* approaches for noise estimation. Gravel et al. [2004] segment the observed noisy image into homogeneous samples, that are each used to compute a scatter-point. The segmentation is performed by first smoothing the observed image, and then by grouping pixels with similar intensity. This leverages the fact that a smoothing operator suppresses the noise, and facilitates the segmentation process. Pixels from edges and texture are excluded from the estimation, since the segmentation is inaccurate in those regions. The noise parameters are finally estimated using a weighted regression of the scatterplot points.

Another type of segmentation is proposed in [Liu et al., 2006], where the authors do not filter the image, but they bin the image elements using a K-means clustering method. A robust fitting algorithm is then adopted to cope with possible inaccuracies of the K-means clustering: the noise parameters are estimated by fitting a lower envelope of the scatterplot, computed

by maximizing a likelihood function that takes into consideration the possible overestimation of the scatter-points variances. Similarly, [Foi et al. \[2008\]](#) filter the observed image, segment it, and then maximize a likelihood function to estimate the noise parameters. A major novelty introduced by this work is that it takes under consideration the clipping of the data, *i.e.* values of the image that exceed certain boundaries are replaced by the values of the boundaries.

2.4.3 Alternative approaches

In our work [[Publication II](#)] we show that the restriction on the use of homogeneous samples is not necessary to correctly estimate the affine noise variance in (2.20), and like the methods [[Mäkitalo and Foi, 2014](#); [Pyatykh et al., 2013](#)] we estimate the noise parameters without any segmentation of the image. The algorithm in [[Mäkitalo and Foi, 2014](#)] estimates the noise by exploiting the stabilizing capability of the generalized Anscombe transformation. In particular, the authors try to find the parameters for the transformation that best stabilizes the standard deviation of the transformed image, and assume those as the final estimates. The algorithm in [[Pyatykh et al., 2013](#)], instead, estimates the noise parameters by analyzing the last eigenvalues of the singular value decomposition (SVD) of homogeneous patches from a noisy image. These eigenvalues depends only on the noise, and therefore are used to estimate the noise parameters.

2.5 Block matching and 3-D filtering (BM3D)

Since the denoising methods from [Publication I](#) and [Publication III](#) involve the use of the BM3D algorithm, in this section we discuss the key features of its original implementation [[Dabov et al., 2007b](#)], also useful for understanding Chapter 4. Although introduced a decade ago, BM3D still gives state-of-the-art results. It can be considered a hybrid algorithm that performs collaborative filtering by exploiting the sparsity of groups of similar patches extracted from different parts of the image (*i.e.* nonlocality).

2.5.1 Grouping and transform

The first step of BM3D is the *grouping* of similar blocks. In this step the observed image is divided into overlapping reference blocks; for each reference block the algorithm looks for similar blocks in its surroundings (search window). The similarity criterion is the Euclidean patch difference, intended as the sum of the squared pointwise differences between corresponding pixels in the two blocks (*i.e.* squared ℓ_2 -norm). Each group of similar blocks is then collected and stacked together in a 3-D volume, that is eventually transformed to decorrelate its pixels, and to collect most of the group energy into few transform coefficients. In BM3D, a separable 3-D transform is applied to the group: first, a 2-D transform (*e.g.* 2-D DCT or 2-D wavelet) is applied to each block individually, then, a 1-D transform (Haar) is applied to the obtained 2-D coefficients along the remaining orthogonal dimension.

Since multiple reference blocks overlap, and 3-D volumes usually share common pixels, eventually the algorithm produces multiple estimates of the same pixel. The overlapping will be taken into account when producing the final estimated image.

2.5.2 Collaborative filtering

Once the 3-D spectrum is computed, we can proceed to filter the group. Since the amount of redundant information of a group of similar patches is abundant, the 3-D transform should promote sparsity of the spectrum coefficients. A very sparse 3-D spectrum may result from spatial local regularity of the patch content, nonlocal regularity (*i.e.* repeated spatial content across the different patches in the group), or a combination of both. In order to perform denoising we apply a hard-thresholding operator to the 3-D spectrum to zero the small coefficients that only depend on the noise power (variance). The threshold used for shrinkage is set proportional to the standard deviation of the noise affecting the signal.

2.5.3 Aggregation

Once all the groups are filtered, we compute the inverse transform and we aggregate the estimates into the final image. In BM3D the aggregation is done via convex combination of all the overlapping estimates, where the weights are inversely proportional to the residual variance of the corresponding group. In this way, we give larger weights to pixels belonging to groups enjoying a sparser 3-D spectrum.

2.5.4 Second iteration with Wiener filter

A second iteration of the algorithm is usually performed to improve the denoising results. Since we have already an approximation of the noise-free data, in this second stage the block matching is performed on the previously denoised image. Then, the similar blocks are extracted from the noisy image at the corresponding coordinates. The group is transformed like in the previous stage, and shrunk by Wiener filtering. In particular, we extract also the corresponding group from the previously denoised image, and we use its spectrum as pilot signal for the Wiener filter. Note that the Wiener filter is applied only in the second stage because it requires a pilot signal for denoising, and at the first iteration we do not have at our disposal such signal.

As a side note, it has been tested that iterating BM3D more than once does not provide any benefit, and does not improve the estimation. On the contrary, in most cases the PSNR of the denoised image in successive iterations decreases compared to the output of the first Wiener stage.

2.5.5 BM3D extensions

Several BM3D extensions have been developed in the last ten years [Boracchi and Foi, 2008; Dabov et al., 2007a; Maggioni et al., 2014]. A direct improvement, for example, is presented in [Dabov et al., 2009], where the 2-D DCT transform is replaced by an adaptive PCA transform. This algorithm gives state-of-the-art performance due its outstanding capability of both, sparse representing piece-wise smooth areas from the image, while

preserving fine details due to the PCA adaptability. However, performing PCA for each group of similar patches is more time consuming than adopting a fixed 2-D transform.

2.6 Quality metrics

In order to quantitatively evaluate the performance of a denoising filter, one needs objective quality metrics. We mainly use the peak-signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) [Wang et al., 2004].

The PSNR is defined as

$$\text{PSNR} = 10 \log_{10} \frac{y_{\max}^2}{\text{MSE}}, \quad (2.22)$$

where y_{\max} is the maximum value of intensity in the image y , and MSE is the mean squared error

$$\text{MSE} = \frac{1}{N} \sum_{x \in \Omega} (y(x) - \hat{y}(x))^2, \quad (2.23)$$

with \hat{y} being the estimate of the noise-free signal y . In other words, the PSNR is the logarithm of the ratio between the maximum value of intensity in the image and the average estimation error, and it is higher for better estimates.

Although the PSNR is conventionally the main quality metric adopted in literature, it presents some shortcomings. For example, it does not take into account that some type of artifacts are more noticeable to the human visual system compared to others [Wang et al., 2004]. In fact, since it is essentially based on the MSE, the PSNR does not consider at all any structure in the image. In this thesis we also consider the SSIM metric, that takes also into consideration the changes in structural information.

Chapter 3

Contribution to noise estimation

In this chapter we first introduce a novel idea for affine noise variance estimation where we use heterogeneous samples to compute the mean-variance scatter-points. We prove that the sample mean and sample variance applied to heterogeneous samples still are unbiased estimators of the population mean and variance of a specific point of the affine relation that we want to estimate; consequently the only problem is handling outliers generated when separating noise from the observed signal. We then show, using a basic prototype algorithm, that the influence of these outliers can be effectively mitigated adopting robust estimators. In the second part of the chapter we instead introduce a more traditional noise estimation model that can be used to estimate the noise variance with any order of dependency with respect to the signal mean, and that is based on the framework presented in Section 2.4. In particular, we introduce a robust fitting of the scatterplot points using as regression tool a maximum-likelihood estimator that models each mean estimate with a Gaussian distribution, and each standard deviation estimate with a Gaussian-Cauchy mixture distribution. In this way we include in our model the possibility that some of the scatter-points standard deviations are outliers wrongly estimated.

3.1 Affine noise variance estimation using heterogeneous data

The general idea that we introduce is that, in case of affine noise variance, sample estimators of mean and variance applied to heterogeneous samples extracted from the noisy image are unbiased estimators of points of the variance curve whose parameters we want to estimate. This is in contrast with most of the methods found in literature. The hypothesis of affine variance, although being restrictive, is well suited to approximate noise variance of signal-dependent classic models such as Poisson and Poisson-Gaussian. A *pseudo*-algorithm based on our proposed idea is the following. Let us consider a noisy image z corrupted by Poisson-Gaussian noise. We extract random patches (*e.g.* squared blocks) from z ; we then compute the first two centered moments (mean and variance) for each patch individually. The computed pairs are then collected on a scatterplot, on which we fit a parametric model whose parameters we want to estimate.

3.1.1 Patch statistics and noise analysis

Let us suppose for the moment that, in an ideal scenario, we can perfectly extract the noise component from the observed image z . Denoting with W a random block from z , and with W^H the corresponding noise counterpart, in [Publication II](#) we prove that the expectation of standard estimators of mean and variance, applied respectively to W and W^H , are unbiased estimators of the patch noise-free signal mean and of the noise variance. In other words, the expectations of the mean-variance pairs lie on the mean-variance curve that we want to estimate.

Considering the noise heteroskedastic Gaussian (2.18), we can express the probability density functions (*p.d.f.*'s) f^W and f^{W^H} of, respectively, W and W^H as

$$f^W(x) = \sum_{k=1}^R \lambda_k p_k^W(x), \quad p_k^W(x) \sim \frac{1}{s_k} \phi\left(\frac{x-m_k}{s_k}\right), \quad (3.1)$$

3.1. Affine noise variance estimation using heterogeneous data

$$f^{W^H}(x) = \sum_{k=1}^R \lambda_k p_k^{W^H}(x), \quad p_k^{W^H}(x) \sim \frac{1}{s_k} \phi\left(\frac{x}{s_k}\right), \quad (3.2)$$

where R is the number of Gaussian functions that compose the mixture, ϕ is the standard normal distribution, p_k^W and $p_k^{W^H}$ are, respectively, the *p.d.f.*'s of the k -th Gaussian distributions of f^W and f^{W^H} , λ_k is the proportion of the elements of the k -th population respect to the total number of elements in W , *i.e.* $\sum_{k=1}^R \lambda_k = 1$, m_k is the mean of the k -th Gaussian function in f^W , and s_k^2 is the variance of both p_k^W and $p_k^{W^H}$. Trivially we have

$$y_W = \sum_{k=1}^R \lambda_k m_k, \quad (3.3)$$

with y_W being the mean of the block W . From the general expression of the variance of a mixture of Gaussian distributions [Johnson et al., 1994], we obtain

$$\sigma_W^2 = \sum_{k=1}^R \lambda_k s_k^2. \quad (3.4)$$

Considering the affine noise variance model (2.20),

$$s_k^2 = am_k + b.$$

Consequently,

$$\begin{aligned} \sigma_W^2 &= \sum_{k=1}^R \lambda_k am_k + \sum_{k=1}^R \lambda_k b = \\ &a \sum_{k=1}^R \lambda_k m_k + b = ay_W + b. \end{aligned} \quad (3.5)$$

This means that the point (y_W, σ_W^2) belongs to the variance curve (2.20). This result constitutes the core of our idea, because it proves that in this ideal scenario there is no difference in applying sample mean and sample variance to homogeneous or heterogeneous samples, since they both give mean-variance pairs that belong to the mean-variance curve to be estimated. However, the assumption of perfect separation of the noise from the observed

signal is very strong. In the next section we analyze the estimation errors, in terms of MSE, of the proposed framework in the practical case of natural images.

3.1.2 Outliers, robust estimators, and estimation errors

In our work we use a wavelet function to high-pass the observed image z , and to extract a signal that could be approximated as zero-mean, with variance equal to the noise variance at the corresponding location; we estimate the local variances using this high-pass filtered image. However, the zero-mean approximation is accurate only in smooth regions of the image, because the high-frequency of texture and edges generate, in the high-pass image, non-zero coefficients that sum to the noise coefficients. If not taken into consideration, the high-frequency coefficients from edges and texture cause overestimation of the local variances, and eventually an overall biased estimation of the noise parameters. Therefore, we treat those high-frequency coefficients as outliers, and we reduce their effect on the overestimation of the variance by using a robust variance estimator. In particular we use the median of absolute deviation (MAD) as estimator of the patch standard deviation. The expression of the MAD is

$$\text{MAD} = \text{med} \left\{ \left| W^H - \text{med} \{ W^H \} \right| \right\}, \quad (3.6)$$

where med is the median operator. Since we model the noise as a mixture of zero-mean Gaussian random variables, we can estimate the mean directly from the observed patch using a standard estimator of the mean. For coherence we estimate the mean using the median.

It is common practice to characterize the statistics of the adopted estimators applied to specific types of data; therefore, we give an expression of the MSE of our estimator for our particular scenario. We consider as error the minimum distance (point-line distance) between the estimated mean-variance point from a patch and the mean-variance line that describes the noise variance affecting the patch. Since it is quite challenging to find its exact analytical expression, we compute it via a Monte Carlo simulation

in which we artificially generate random noisy blocks containing piecewise-smooth regions, and we apply our estimator to them. We perform the estimation for a wide range of noise parameters and for several percentages of edges in the patch. Finally, in Figure 3.1 (reproduced from [Publication II](#)) we report the root mean normalized squared error (RMNSE) in function of the amount of edges $B\%$ in a patch, and for 4 different patch size n . We average normalized errors because it makes little sense to average together errors corresponding to different noise magnitudes. In particular, we normalize each estimation error dividing it by the MSE of the estimator applied to a flat patch and affected by noise with constant variance, *i.e.* ideal easiest case:

$$\text{RMNSE}_{(B\%,n)} = \sqrt{\frac{1}{N_p} \sum_{i=1}^{N_p} \frac{e_i^2}{\bar{e}^2 (\hat{\sigma}_{i\perp}^2)}}, \quad (3.7)$$

where N_p is the total number of patches used for the estimation of the point at $(B\%, n)$, e_i^2 is the squared estimation error, and $\bar{e}^2 (\hat{\sigma}_{i\perp}^2)$ is the squared estimation error of our estimator applied to a flat patch affected by noise with variance $\hat{\sigma}_{i\perp}^2$, *i.e.* the orthogonal projection of the estimated patch variance $\hat{\sigma}_i^2$ on the ground-truth variance curve $\sigma^2(y)$.

3.1.3 Procedure

We now give a brief description of the prototype algorithm that we have used to validate our assumptions.

- *High-pass filtering*: the observed image z is convolved with a 2-D high-pass kernel in order to extract its noisy component. In this particular implementation, we use a Daubechies kernel to generate the 2-D function.
- *Local estimation*: we take, at random, N blocks from z and its corresponding noise image. With these blocks we compute, respectively, mean and variance with the aforementioned estimators med and MAD.

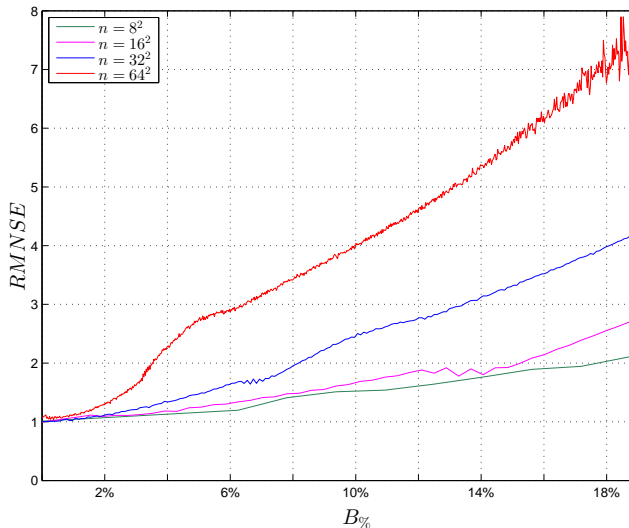


Figure 3.1: Root mean normalized squared error (RMNSE) of the estimators in function of the percentage of edges $B_{\%}$ in the patch and the number of samples n in a patch. This result has been computed with a Monte Carlo simulation.

The mean-variance pairs are the abscissa and ordinate of the scatterplot points that we are going to fit.

- *Fitting*: once the scatterplot has been computed, we use a least square (LS) method to fit a line on the cloud of points, to finally estimate the parameters a and b of our noise model.

The presented prototype algorithm has been implemented using elementary tools. This has been done on purpose, to prove that the proposed scheme gives valid results independently from the complexity of the adopted tools. In the next section we show the performances of this algorithm compared to a state-of-the-art method of [Foi et al. \[2008\]](#).

3.1. Affine noise variance estimation using heterogeneous data

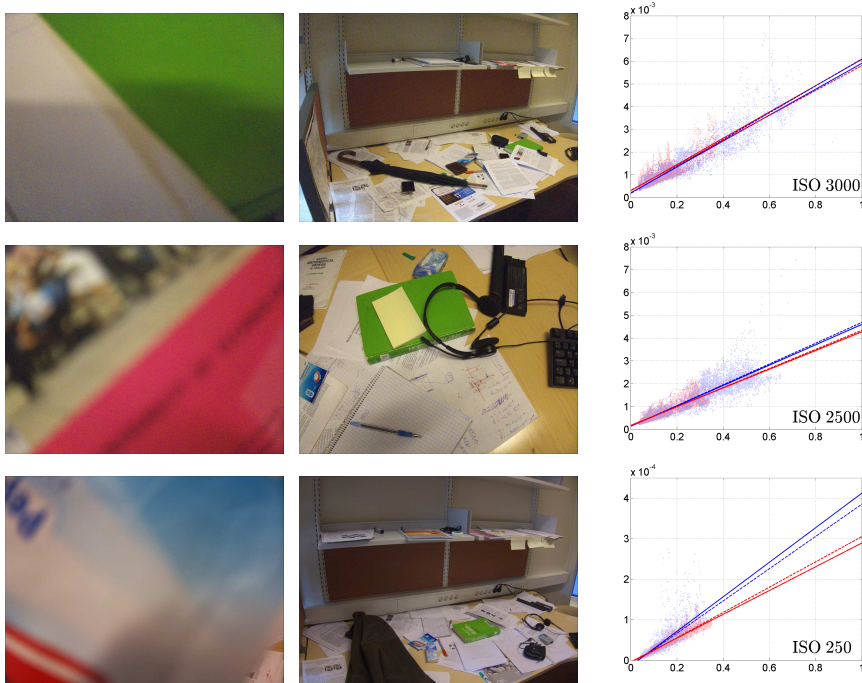


Figure 3.2: Scatterplots and estimated functions (continuous lines) for out-of-focus (leftmost column, red color) and complex natural (central columns, blue color) images. The images have been taken with a *Canon PowerShot S90*, at different ISO and with different exposure times. In the same scatterplots are shown the estimated functions (dashed lines) computed with the reference algorithm [Foi et al., 2008].

3.1.4 Results

In Figure 3.2, reproduced from [Publication II](#), we compare our estimation results (continuous lines) with the reference algorithm [Foi et al., 2008] (dashed lines). This comparison is meant to validate our assumptions on the use of heterogeneous data; therefore, we evaluate the performance of

the proposed algorithm under different acquisition conditions. We consider two type of images: out-of-focus images (OoF) (leftmost column, red color), and natural images (central column, blue color). We include the OoF case because it well approximates the ideal case described in Section 3.1.1, where the wavelet kernel well separates the noise from the noisy image. We thus consider the estimates from OoF images as ground-truth, and we evaluate the estimates from natural images by comparison with respect to the corresponding OoF cases.

The first thing one can notice is that the continuous lines are always close to the corresponding dashed lines. This means that the proposed prototype algorithm gives results comparable to the state-of-the-art method in all cases. This is an important result because it practically confirms our hypothesis. Furthermore, in the case where the proposed algorithm fails, *i.e.* where the blue continuous line is far from the red continuous one, also the algorithm [Foi et al., 2008] fails. In general, the performances of both, proposed and reference algorithms, drop when reducing the level of noise (large ISO). This proves that challenging cases for the proposed scheme are challenging also for algorithm based on the conventional framework. For this purpose, we show in Figure 3.3 (from [Publication II](#)) a complete failure in estimating the noise parameters. In fact, due to the underexposure and the large presence of texture in the image, the proposed algorithm (red), as well as the reference one (green), estimate a line completely different from the ground-truth (black).

The most relevant conclusion that we can draw from the presented experiments is that, in case of affine noise variance, processing homogeneous samples has no advantage compared to processing heterogeneous samples, and that estimation algorithms can thus be simplified by avoiding segmentation.

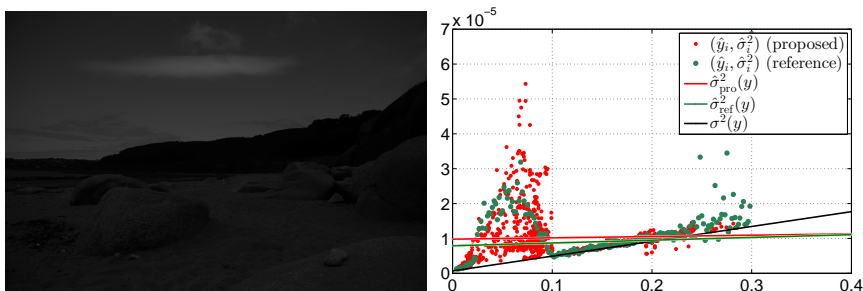


Figure 3.3: Left: challenging image. Right: scatterplot and estimated lines with the proposed prototype algorithm (red) and by [Foi et al., 2008] (green). The result is compared with the ground-truth (black).

3.2 Robust noise estimation for any parametric variance function

As mentioned above when discussing about Figure 3.3, edges and texture in an image can heavily interfere with noise estimation. In particular, they may create outliers that, if not considered in the model, eventually lead to overestimation of the noise. In this section we introduce a robust model for noise estimation that includes the potential presence of outliers among the scatter-points. The model can be used to estimate noise variance that depends on the signal mean according to any parametric function. This is a far more general setting than the affine model treated in Section 3.1, and thus we cannot apply the heterogeneous estimation method under this hypothesis.

Based on this model, we propose an algorithm that first segments the observed noisy image, grouping pixels with same expectation. Then, it computes the mean and the standard deviation of the groups (referred to as bins) using the segmented samples. Each group therefore represent a point (*i.e.* mean-standard deviation pair) in the scatterplot that is eventually fitted to estimate the noise parameters. Since, among the computed standard deviations, there could be outliers, we propose to use a fitting method

to take into account this possibility. We maximize a likelihood function in which each scatter-point standard deviation is modeled as realization of a mixture of a Gaussian and a Cauchy random variable: the heavy tails of the Cauchy distribution are capable of representing the possibility that a point in the scatterplot is an outlier, and lies far from its mean. In fact, compared with a Gaussian distribution, whose bell is mostly confined in the range $\pm 3\sigma$, the Cauchy distribution presents a much lower decay rate. We favour the Cauchy distribution over other heavy tailed distributions such as Laplacian because it is a symmetric bell-shaped function that resembles the Gaussian distribution; consequently, the resulting mixture is similar to a Gaussian distribution in the vicinity its mean value, hence being more suited for modeling estimation errors in the lack of outliers.

3.2.1 Segmentation and local statistics

The first step of the algorithm consists in computing the local mean-standard deviation pairs that are used for fitting. We filter the observed image using two filters to generate high- and low- pass versions of z . Since in smooth areas the standard deviation of the high-pass image is approximately equal to the standard deviation of the noisy image, the high-pass image is used for the estimation of the local standard deviations. We instead use the low-pass image to perform segmentation and to compute the mean of each bin. We adopt sample mean and sample standard deviation as estimators of the local statistics. Since both estimators are unbiased only in smooth regions, before computing the local estimates we discard samples that belong to edges and texture. To do so we compute, with a second derivative filter, the *variation* of the image, and we discard the pixels where the derivative is large.

Once the samples have been grouped in bins, we compute mean and standard deviation of each bin using the sample mean applied to the low-frequency samples and the sample standard deviation applied to the high-frequency samples.

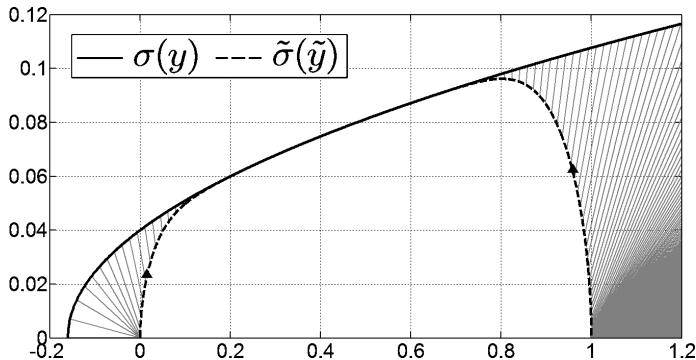


Figure 3.4: Effect of clipping on the standard deviation curve.

3.2.2 Clipping

All acquisition devices have a finite dynamic range that may not represent the large variation in luminosity in the scene. The device usually replaces values of intensities that exceed the range with the boundary values; in other words, an observed image \tilde{z} is usually generated as $\tilde{z} = \max(0, \min(1, z))$, where the range of the image is normalized in $[0, 1]$, and the acquired image is denoted with z . This procedure is commonly known as *clipping*.

It is clear that the noise statistics are not preserved by the clipping operator; in other words, if the acquired image z is affected by Poisson-Gaussian noise, the statistics of the noise affecting the clipped image \tilde{z} are not the same. Since we observe \tilde{z} , and we want to estimate the noise affecting the image z , we have to take into consideration the clipping effect in our estimation model. It is out of the scope of the thesis to describe accurately the mathematical model for clipping, however this has been discussed extensively by [Foi et al. \[2008\]](#), and we report here the main results, in order to give the reader an idea of how we deal with clipping.

In Figure 3.4, reproduced from [Publication I](#), we show the effect of clipping on the standard deviation curve. While the curve σ of the noise standard deviation affecting z is a square-root function of the noise-free signal y ,

the standard deviation $\tilde{\sigma}$ of the clipped data \tilde{z} is substantially different from σ . In [Foi et al., 2008], the authors find the analytical expression that gives the clipped standard deviation $\tilde{\sigma}(y)$ in function of the original curve $\sigma(y)$. Using this transformation, we are able to build a fitting model that, from the local statistics of the clipped data, estimates the noise parameters a and b of $\sigma(y)$. In particular, we find the curve that, after transformation (*i.e.* clipping), best fits the scatterplot of the local estimates that we computed on clipped samples. In this way we cope with the problem of clipping in the fitting (last) step. This is an important result, because the proposed fitting method for clipped data can be used in any algorithm, independently from the method used for computing the scatter-points.

3.2.3 Robust fitting

In [Foi et al., 2008], the authors model each local mean-standard deviation pair as a bi-variate Gaussian random variable centered on the standard deviation curve. Considering each estimate independent, they maximize a likelihood function in which the joint probability is simply given by the product of each distribution. However, in challenging cases, *e.g.* highly textured images, a large number of outliers might be present among the mean-standard deviation pairs. We cope with this possibility by modifying the distribution model that represent each local estimate. In particular, we model the standard deviation estimates as realizations of a mixture of zero-mean Gaussian and Cauchy distributions, where the mixture coefficient adapt to the particular image under interest [Publication I]. Since there are no outliers among the estimates of the local means, we model the mean estimates as realizations of Gaussian random variables.

The main advantage of using a Gaussian-Cauchy mixture is that the Cauchy distribution better adapts to the possibility to have samples far from the mean value. In particular, since the Cauchy distribution has heavier tails compared to the Gaussian distribution, it better models outliers that lie far from the curve that we want to fit. In normal conditions, *i.e.* using only a Gaussian distribution, the outliers would heavily influence the final estimate; the adoption of Cauchy distributions, instead, includes them as

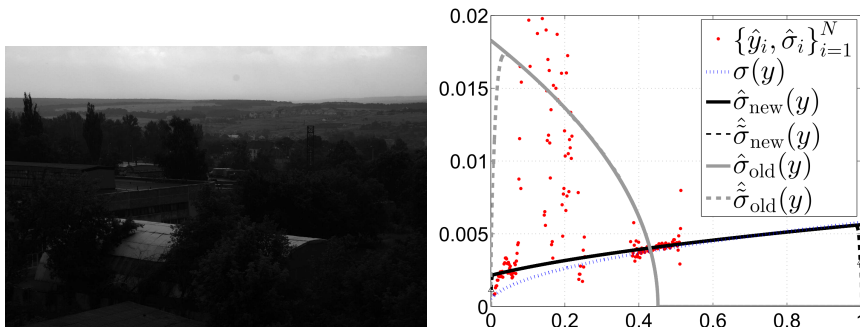


Figure 3.5: Underexposed raw image and the relative scatterplot of mean-standard deviation pairs, severely corrupted by outliers.

part of the model, and models them as improbable realizations far from the distribution mean.

We also include the mixture coefficient in the optimization process, *i.e.* the likelihood is optimized considering as variables the noise parameters and the mixture coefficient. In this way we adapt our model to the amount of textured areas in the observed image. In particular, when the final estimate of the mixture coefficient gives a distribution that is mostly Gaussian, we can assume that the amount of texture (or/and edges) in the observed image is relatively low. On the other hand, if the final mixture coefficient gives more weight to the Cauchy distribution, it is reasonable to assume that the image contains large amount of texture and edges.

3.2.4 Results

We now show the advantage of using a robust estimator compared to the non-robust version from [Foi et al., 2008]. Figure 3.5, reproduced from [Publication I](#), shows a typical challenging case, in which the estimation is severely affected by the amount of texture and by the lack of scatter-points. On the left side of the figure we show the image on which we perform noise estimation, and on the right we show the results of the proposed

3. CONTRIBUTION TO NOISE ESTIMATION

method compared to the original one. In the scatterplot we report the mean-standard deviation pairs (red dots), the ground-truth $\sigma(y)$, the curves estimated with the proposed algorithm $\hat{\sigma}_{\text{new}}(y)$ and $\hat{\hat{\sigma}}_{\text{new}}(y)$, and with the reference one $\hat{\sigma}_{\text{old}}(y)$ and $\hat{\hat{\sigma}}_{\text{old}}(y)$. We denote with the symbol $\hat{\hat{\cdot}}$ on top of a variable the curve estimated from the clipped data, and with the symbol $\hat{\cdot}$ the final estimate.

Since the observed picture is underexposed, the algorithm segments the image in a limited amount of bins. As a consequence, we have only few local estimates, each corresponding to a bin. Furthermore, several of the local estimates are outliers, that has been miscalculated due to the lack of samples per bin, or mostly due to the heavy presence of texture and edges in the image. Note how, in fact, the reference algorithm is influenced by these outliers, and fails to fit a correct curve over the scatter-points. On the other hand, the presented robust estimator, although partially misled by outliers, well estimates most of the standard deviation curve.

Chapter 4

Contribution to denoising

In the first part of this chapter we introduce a spatially adaptive 1-D group transform for BM3D based on the spatial coordinates of similar blocks. The adaptive transform increases the sparsity of the transform coefficients, and consequently improves the filtering results from shrinkage. We also use the same spatial information to design an algorithm based on alpha-rooting [Dabov et al., 2007c] that, adaptively, enhances local image features depending on their orientation with respect to the coordinates of similar features at other locations.

In the second part of this chapter we propose a model for iterative denoising of Poisson images. The proposed model is based on the VST denoising framework, and exploits convex combinations of noisy signal and its previous estimate to improve the signal-to-noise-ratio (SNR) of the signal to be stabilized. We prove that, especially for low signal intensities, the performance of the stabilization, and in general of the denoising filter, are vastly improved, outperforming the state of the art.

4.1 Collaborative filtering based on coordinates of similar features

As mentioned in Section 2.5, the BM3D algorithm performs a 3-D transform of groups of similar blocks via separable 2-D+1-D transform, in which a 2-D DCT (or wavelet) applied separately to each block is followed by a 1-D Haar transform in the orthogonal (nonlocal) direction. It is clear that in this scheme the spatial information (*i.e.* coordinates) of similar blocks is never used. However, the spatial coordinates of the grouped blocks is a feature that could be exploited to enhance the sparsity of the 3-D group spectrum. Let us consider, for example, the case in which similar blocks lie on a surface that smoothly changes its average intensity. Since, before applying the 3-D transform, the blocks are reordered according to their similarity with respect to the reference block, the Haar transform may not be effective at sparsifying the spectra of the blocks. By using, instead, polynomials that approximate the smooth variation of the surface from which the blocks are extracted, we can improve the sparsity of the 3-D spectrum coefficients. From this idea, we propose a novel 1-D transform based on the spatial coordinates of the similar blocks, that adapts to the particular group that we are processing [Publication III]. We then use the principal directions of the coordinates of similar blocks to derive an enhancement algorithm that adaptively chooses whether to soften or sharpen selective coefficients of the group spectrum.

4.1.1 Orthogonal polynomials transform

We now describe how to generate the set of orthogonal polynomial (OP) functions that we use as 1-D transform basis in the presented extension of BM3D. Denoting with n_B the number of similar blocks in a group, we consider a collection of n_B bi-variate polynomial functions defined over the image spatial domain, and linearly independent over the coordinates of similar blocks. We then sample the functions at the spatial coordinates of the blocks, and we rearrange them as columns of a matrix P . We compute the QR decomposition of P (Gram-Schmidt factorization) to generate an orthogonal basis Q whose elements (matrix columns) are bi-variate polyno-

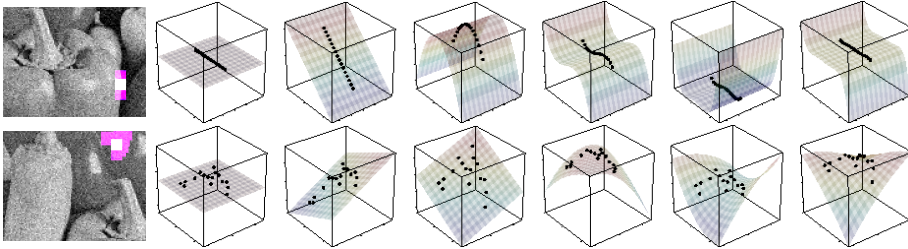


Figure 4.1: Example of adaptive basis functions generated by orthogonal polynomials for two different groups of similar blocks (purple areas). At each row we show the first 6 basis functions of the corresponding group. Observe how the basis functions adapt to the group coordinates.

mial functions sampled at the spatial coordinates of the similar blocks. We use the matrix Q as 1-D transform in BM3D.

In Figure 4.1, reproduced from [Publication III](#), we show two examples of basis functions (visualizing the first 6 basis functions for each basis) obtained from the block coordinates of two different groups. On the top row we show the basis functions of a group aligned along an edge, while on the bottom row we plot the basis functions of a group from a uniform region. Note how the basis functions strongly adapt to the particular blocks spatial coordinates, and how the proposed 1-D transform introduces a new level of adaptability to BM3D.

The scenario in which one can distinctly appreciate the advantages of the proposed adaptive transform compared to the standard Haar transform is when the search window is on an area that smoothly varies, like for example the shoulder of Lena (shown in Figure 4.2). In this scenario, the polynomial functions better represent such smooth changes compared to the Haar transform.

4.1.2 Directional enhancement

The spatial coordinates of similar blocks may indicate whether the blocks contain an edge or not. In fact, if the blocks are aligned, it is reasonable

to assume that they contain an edge, and that the direction of the edge is aligned with the blocks. An example can be seen in the first row of Figure 4.1, where we show the basis of a group of blocks (purple area) whose reference one (white block) lies on an edge. Here, it is clear that the blocks are aligned with the direction of the edge. On the bottom row, instead, we report similar blocks from a uniform region; note how the blocks are scattered, and a principal orientation cannot be discerned.

We propose a directional algorithm that exploits the above spatial information to enhance the content of similar blocks by increasing or decreasing, individually, each spectrum coefficients; this results in, respectively, sharpening or softening of the corresponding basis functions. Let us consider the set of blocks spatial coordinates S ; we compute its PCA decomposition as $S = U\Lambda V^T$. By definition, the columns of U indicate the two principal directions along which the cloud of points is oriented; at the same time, the diagonal elements of Λ are the length of these components: the larger the value, the more the cloud is aligned with the corresponding principal component. Based on the ratio between the first and second component, we can decide whether to sharpen or soften a block. In fact, if the ratio between the first and second component is large, then the set of coordinates is mostly oriented on a thin line. On the contrary, if the ratio is close to one, the blocks coordinates are spread uniformly on a broad area, suggesting that the blocks are extracted from a regular region. In this way we can sharpen the blocks from an edge, and we can soften the blocks from a uniform area. This concept is similar to the tensor methods developed by [Feng and Milanfar \[2002\]](#) and [Weickert \[1999\]](#), that estimate whether an image region is an edge or a smooth area based on the principal components of its gradient.

Since we use the alpha-rooting sharpening method [[Dabov et al., 2007c](#)], we also adapt the sharpening and softening to specific 2-D transform coefficients. In particular, for each 2-D spectrum coefficient, we compute the energy (in terms of ℓ_2 norm) of the derivative of its corresponding basis function in the directions of the two principal components of the coordinates of the group. Based on the ratio between them, we decide whether to sharpen the coefficient, if the ratio is larger than one, or to soften it, if the

ratio is smaller than one. In this way we perform at the same time softening along an edge, while sharpening the edge itself. In our implementation, we adjust the sharpening/softening strength with the ratio between first and second principal component of the set of coordinates S : if the ratio is large we sharpen more compared to a case in which the ratio is close to 1.

4.1.3 Results

In Figure 4.2, reproduced from [Publication III](#), we report a detail of *Lena* denoised with the proposed method compared to the standard BM3D (Haar 1-D transform); the image has been corrupted by additive Gaussian noise with standard deviation $\sigma = 35$. We specifically report the shoulder detail because it is where the proposed algorithm visibly outperforms its canonical counterpart.

In Figure 4.3, also reproduced from [Publication III](#), we denoise and sharpen *Peppers* (we show only a detail) corrupted by AWGN with standard deviation $\sigma = 20$. On the left-bottom we report the results from the proposed enhancement scheme, while on the right-bottom we show the conventional alpha-rooting result, with constant alpha coefficient. As a mean of comparison, on the top-right position of the same figure we also report the result of the proposed denoising algorithm with no sharpening applied. Note how the proposed enhancing algorithm sharpens the edges of the image while softening the piecewise smooth areas. In contrast, the standard alpha-rooting algorithm sharpens all the image details indiscriminately, enhancing in some cases artifacts introduced by the denoising algorithm itself.

Finally, to show the adaptive sharpening strength of the proposed alpha-rooting filter, we report in Figure 4.4 the ratio between the principal components of each coordinates set of similar blocks. Red areas indicate high ratio (edges), while blue areas approach the value 1 (smooth regions). We remind the reader that we do not perform directly sharpening on the red areas, but for the red areas we perform sharpening only on the transform coefficients whose corresponding basis functions have most variations orthogonal to the edge, while we soften the coefficients whose corresponding basis functions mostly vary along the edge.

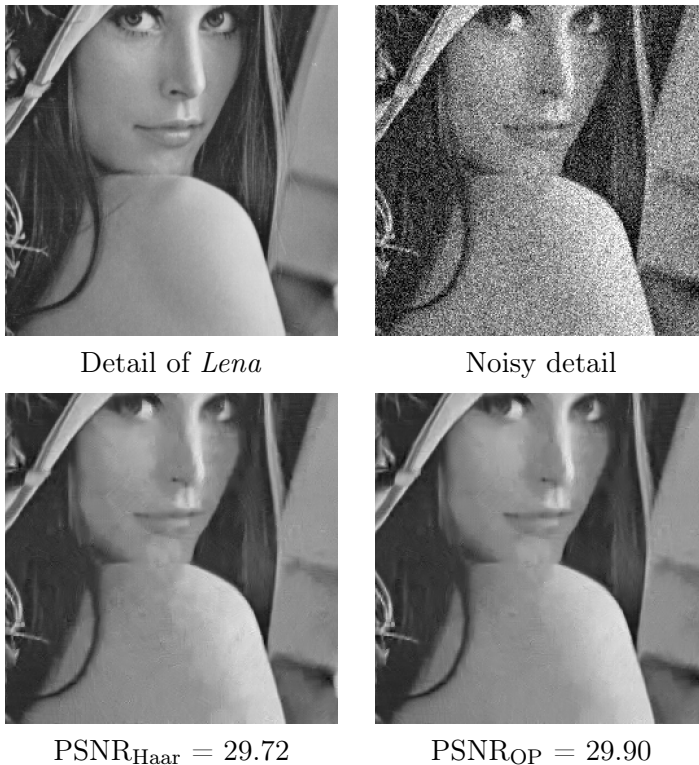


Figure 4.2: Denoising of *Lena* corrupted by i.i.d. Gaussian noise with $\sigma = 35$. From left to right, top to bottom: original image, noisy observation, image denoised by the standard BM3D algorithm, images denoised by the proposed algorithm based on adaptive orthonormal polynomials. Notice the improvement, particularly in smooth regions, such as the shoulder area.

4.2 Poisson image denoising

We now introduce a denoising model for Poisson images [Publication IV] based on the variance-stabilizing transform (VST) framework described in Section 2.3.2. We propose an iterative method in which, at each iteration,

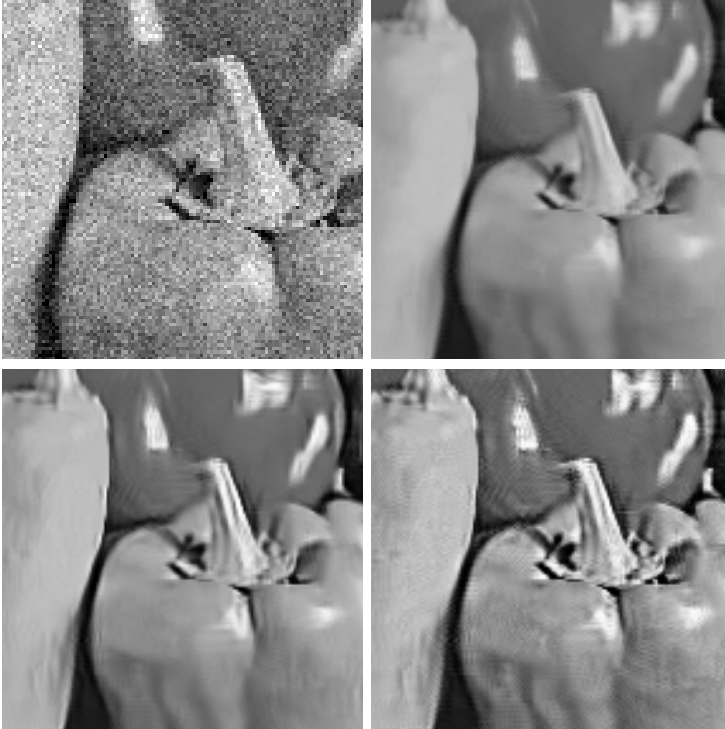


Figure 4.3: Enhancement of *Peppers* corrupted by Gaussian noise with $\sigma = 20$. From left to right, top to bottom: Details of *Peppers* noisy image; denoised image with no sharpening; image denoised and enhanced by the proposed algorithm; conventional alpha-rooting.

we compute a convex combination of the previous estimate and the noisy signal in order to increase the SNR of the signal to be deionised. This algorithm has proved to be very effective especially for low-count images, where a standard VST approach fails to stabilize the data, and consequently performs poor denoising. We show that, also for low-count images, a VST approach is still valuable, and gives state-of-the-art results within the proposed iterative framework.

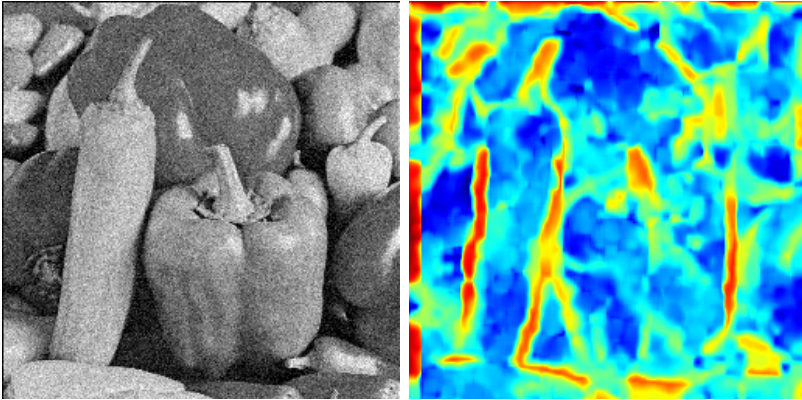


Figure 4.4: Left: *Peppers* noisy image with $\sigma = 20$. Right: ratio between the lengths of the principal axes of the group coordinates: higher values are shown in red, and lower values in dark blue.

4.2.1 Inverse Anscombe transform

The algebraic inverse f^{-1} of the Anscombe transform is a biased inverse transformation

$$f^{-1}(\mathbb{E}\{f(z) | y\}) \neq \mathbb{E}\{z | y\}, \quad (4.1)$$

and therefore cannot be used for practical applications. This is due to the fact that the Anscombe forward and algebraic inverse transforms are non-linear transformations, and therefore the algebraic inverse does not commute with the expectation operator, resulting in the disequality (4.1).

[Anscombe \[1948\]](#) proposes an asymptotically unbiased inverse transformation for large intensity values. However, the bias becomes significant for $y < 5$. This problem has been considered an intrinsic fault of the *forward* Anscombe transform, and for a period of time VST Poisson denoising algorithms were neglected in favour of algorithms designed to directly denoise Poisson data [[Salmon et al., 2014](#); [Giryas and Elad, 2014](#)]. However, the interest on the VST framework has been rekindled when [Mäkitalo and Foi \[2011\]](#) noted that the inaccuracy of the VST framework for low inten-

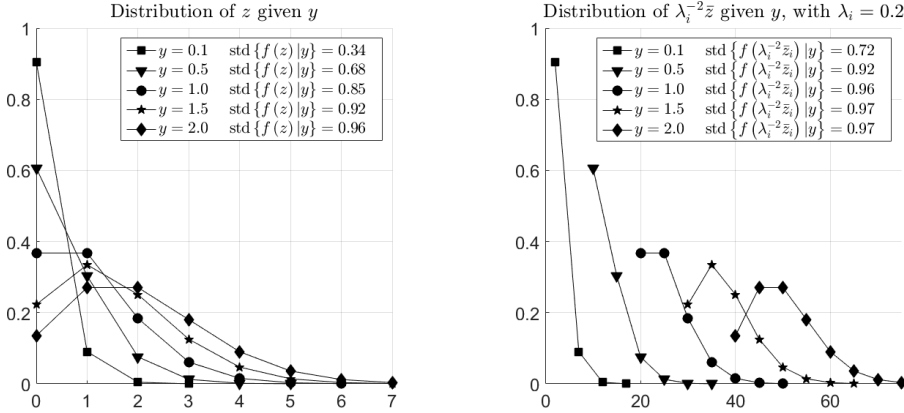


Figure 4.5: Effect of convex combination on the data distributions and on the standard deviations of the stabilized data.

sity signals is essentially due to the inverse transform; there, the authors propose an exact unbiased inverse transform that is unbiased in the range $[2\sqrt{3/8}, +\infty)$, achieving, in combination with the BM3D filter, state-of-the-art performances. Most importantly to our scope, we derive an exact unbiased inverse for our data like Mäkitalo and Foi [2011] do in their work. Details are given in Section 4.2.3.

4.2.2 Noisy+estimate convex combination

For low count signals the forward Anscombe transform, and forward variance stabilizing transformations in general, become inaccurate. To improve the stabilization we propose an iterative approach in which, at each iteration, we sum to the noisy image its previous estimate in order to increase the SNR of the image to be stabilized, and consequently to make the transformation more accurate.

We propose the following normalized convex combination

$$\frac{\bar{z}_i}{\lambda_i^2} = \frac{\lambda_i z + (1 - \lambda_i) \hat{y}_{i-1}}{\lambda_i^2}, \quad (4.2)$$

where $0 < \lambda_i \leq 1$ and i is iteration index. In our implementation λ_i changes at each iteration i and for each image z ; in Section 4.2.6 we give more details about the parameters selection.

The expectation and variance of $\lambda_i^{-2}\bar{z}_i$ are

$$\mathbb{E} \{ \lambda_i^{-2}\bar{z}_i | y \} = \text{var} \{ \lambda_i^{-2}\bar{z}_i | y \} = \lambda_i^{-2}y. \quad (4.3)$$

Compared to the SNR of the observed image z , the SNR of the new image $\lambda_i^{-2}\bar{z}_i$ increases, and the Anscombe transform better stabilizes it. In Section 4.2.5 we give the explicit relation between SNR and the combination coefficient λ_i .

Figure 4.5, reproduced from the [supplementary to Publication IV](#), shows on the left side the Poisson distributions for 5 different means (and variances). Note how most of these distributions overlap, making any VST inaccurate. The right side of Figure 4.5, instead, shows the corresponding normalized convex combinations using $\lambda_i = 0.2$. Due to the convex combination and the normalization, the distributions shift to higher values of mean; and most importantly, the distributions overlap less compared to their original counterparts. This is the main reason why the Anscombe transform is more accurate for the convex combinations. Furthermore, in the legends of Figure 4.5 we also report the standard deviations computed from the stabilized Poisson distributions and from the stabilized convex combinations. Note how the latter are close to 1 also at low intensities $y < 1.5$.

4.2.3 Forward and inverse VST adopted

Forward VST and filtering

Although the variable $\lambda_i^{-2}\bar{z}_i$ in (4.2) is not Poisson distributed, it has been proved that this family of combinations is still stabilized (asymptotically) by the Anscombe transform [Bar-Lev and Enis, 1990]. This result justifies using the forward Anscombe transform f to stabilize $\lambda_i^{-2}\bar{z}_i$.

Once the noise variance has been made signal-independent, we denoise the image $f(\lambda_i^{-2}\bar{z}_i)$ with a filter for additive noise

$$D_i = \Phi [f(\lambda_i^{-2}\bar{z}_i)], \quad (4.4)$$

where Φ is the filter and D_i is the filtered image. In our implementation we use BM3D, but any other filter for removal of additive noise can be used instead.

Inverse transform

Although we use the canonical forward Anscombe transform, the exact unbiased inverse introduced in [Mäkitalo and Foi, 2011] is biased for the convex combination of variables that we propose.

We derive the exact unbiased inverse transformation for D_i in (4.4) by computing the expectation $E\{f(\lambda_i^{-2}\bar{z}_i) | y\}$ for a finite grid of values y and λ ; in this way we obtain the inverse mapping

$$E\{f(\lambda_i^{-2}\bar{z}_i) | y\} \mapsto y \quad (4.5)$$

for the grid points. Then, we interpolate the results to compute the inverse mapping at the specific coordinates λ_i and D_i . For elements outside the grid we derive an analytical asymptotic form. For mathematical details we refer the reader to [Publication IV](#).

4.2.4 Binning

The main advantage of combining observed signal and previous estimates is that it improves the stabilizing performance of the Anscombe transform by increasing the SNR of the signal to be stabilized. In fact, the Anscombe forward transform is inaccurate for very low intensity (very low count) signals, and by increasing the SNR, we improve its stabilizing accuracy. Considering that initially we have poor estimates of the noise-free signal, at these first iterations the VST is not able to correctly stabilize the signal. To cope with this problem, at early stages of the algorithm we increase the SNR with *binning*.

Binning \mathcal{B}_{h_i} is an operator that replaces each $h_i \times h_i$ block from an image with the sum of the elements within the block. Since it commutes with (4.2), we express the binned convex combination as

$$\mathcal{B}_{h_i} [\bar{z}_i] = \lambda_i \mathcal{B}_{h_i} [z] + (1 - \lambda_i) \mathcal{B}_{h_i} [\hat{y}_{i-1}]. \quad (4.6)$$

Since $\mathcal{B}_{h_i} [z] \sim \mathcal{P}(\mathcal{B}_{h_i} [y]) = \mathcal{P}(\mathbb{E}\{\mathcal{B}_{h_i} [z]|y\})$, and modeling \hat{y}_{i-1} as y , we have that $\mathcal{B}_{h_i} [\bar{z}_i]$ (respectively $\lambda_i^{-2} \mathcal{B}_{h_i} [\bar{z}_i]$) is subject to the same conditional probability of \bar{z}_i (respectively $\lambda_i^{-2} \bar{z}_i$), which means that the adoption of binning does not interfere with the subsequent VST, denoising, and inverse VST. Thus, we use binning in combination with the forward Anscombe transform.

Debinning

Once the stabilized signal has been filtered and the inverse transform applied, it is necessary to perform debinning. We call debinning $\mathcal{B}_{h_i}^{-1}$ the inverse operator of binning, and its main purpose is to upscale the resulting image to its original size. During this process, clearly, the binned samples are also divided by the number of pixels in each bin, to make each binned value the mean of the block (and not the sum of its elements). In [Salmon et al., 2014; Giryes and Elad, 2014] debinning is performed via interpolation. However, in this way it is not guaranteed that $\mathcal{B}_{h_i} [\mathcal{B}_{h_i}^{-1} [x]] = x$, *i.e.* the local means over the bins do not coincide with the input of debinning. In our implementation of debinning, instead, we guarantee the aforementioned condition by alternating, iteratively, binning and interpolation.

4.2.5 SNR as function of λ_i and h_i

It is convenient to express the dependency of the SNR with respect to the combination coefficient λ_i and the bin size h_i .

Considering that binning is essentially the sum of the pixels inside a block, the SNR of a given binned block of pixels from z is

$$\text{SNR}(\mathcal{B}_{h_i}[z(\Omega_{h_i})]) = 10 \log_{10} \frac{\mathbb{E}^2\{\mathcal{B}_{h_i}[z(\Omega_{h_i})]\}}{\text{var}\{\mathcal{B}_{h_i}[z(\Omega_{h_i})]\}} = 10 \log_{10} \sum_{x \in \Omega_{h_i}} y(x), \quad (4.7)$$

where Ω_{h_i} are the coordinates of a bin of size $h_i \times h_i$, and $z(\Omega_{h_i})$ is the block from z at the coordinates Ω_{h_i} .

From the expectation and variance in (4.3), the SNR of a combined pixel is

$$\text{SNR}(\bar{z}_i(x)) = 10 \log_{10} \frac{\mathbb{E}^2\{\bar{z}_i(x)\}}{\text{var}\{\bar{z}_i(x)\}} = 10 \log_{10} \frac{1}{\lambda_i^2} y(x). \quad (4.8)$$

Finally, the SNR of binned and combined data is

$$\text{SNR}(\mathcal{B}_{h_i}[\bar{z}_i(\Omega_{h_i})]) = 10 \log_{10} \frac{\mathbb{E}^2\{\mathcal{B}_{h_i}[\bar{z}_i(\Omega_{h_i})]\}}{\text{var}\{\mathcal{B}_{h_i}[\bar{z}_i(\Omega_{h_i})]\}} = 10 \log_{10} \frac{1}{\lambda_i^2} \sum_{x \in \Omega_{h_i}} y(x), \quad (4.9)$$

where $\bar{z}_i(\Omega_{h_i})$ is approximated as $\lambda_i z_i(\Omega_{h_i}) + (1 - \lambda_i) y(\Omega_{h_i})$. Since $y(x) \geq 0$, $\forall x \in \Omega_{h_i}$, and $\lambda_i \leq 1$, then for any $x \in \Omega_{h_i}$

$$\text{SNR}(\mathcal{B}_{h_i}[\bar{z}_i(\Omega_{h_i})]) \geq \left| \begin{array}{c} \text{SNR}(\mathcal{B}_{h_i}[z(\Omega_{h_i})]) \\ \text{SNR}(\bar{z}_i(x)) \end{array} \right| \geq \text{SNR}(z(x)) = y(x), \quad (4.10)$$

proving that binning and convex combination indeed increase the SNR of the signal to be stabilized.

Finally, we give the dependence of the SNR in (4.9) upon the coefficients λ_i and h_i :

$$\text{SNR}(\mathcal{B}_{h_i}[\bar{z}_i(\Omega_{h_i})]) = 10 \log_{10} \frac{h_i^2}{\lambda_i^2} \bar{y}(\Omega_{h_i}), \quad (4.11)$$

where $\bar{y}(\Omega_{h_i})$ is the average intensity of the block $y(\Omega_{h_i})$.

4.2.6 Algorithm parameters

The set of parameters that the algorithm uses are

- the number of iterations K ;
- the initial h_1 and final h_K bin sizes;
- the last value of the combination coefficient λ_K (since $\lambda_1 = 1$).

The aforementioned parameters are determined adaptively for each observed image z based on the image quantiles; for a detailed description of the adaptive parameters calculation process we refer the reader to the [supplementary to Publication IV](#). Note that the available online code [[Azzari and Foi, 2016a](#)] can be used to reproduce the results from [Publication IV](#), and also returns the values of the parameters used to denoise, individually, each input image z .

An important requirement of the parameters is that the bin size and combination coefficient both decrease at each iteration, *i.e.* $h_i \leq h_{i-1}$ and $\lambda_i \leq \lambda_{i-1}$. In this way, initially, the proposed denoising algorithm increases the SNR mostly by binning with large bin size, to compensate the lack of an accurate previous estimate. Then, it gives increasingly more relevance to the convex combination by decreasing the bin size h_i and the combination coefficient λ_i .

4.2.7 Results

The proposed algorithm has been tested against the methods [[Salmon et al., 2014](#); [Giryès and Elad, 2014](#); [Mäkitalo and Foi, 2011](#); [Rond et al., 2015](#); [Pyatykh and Hesser, 2015](#)] on a dataset of 11 images. Table I and Table II in [Publication IV](#) show the comparison in terms of PSNR and execution time, and proves that the proposed algorithm gives best overall results. An example is reported in [Figure 4.6](#), where we compare the denoising of the image *Bridge* at peak 1. Note how the proposed iterative scheme preserves details that the other algorithms are unable to recover from such very noisy observations.

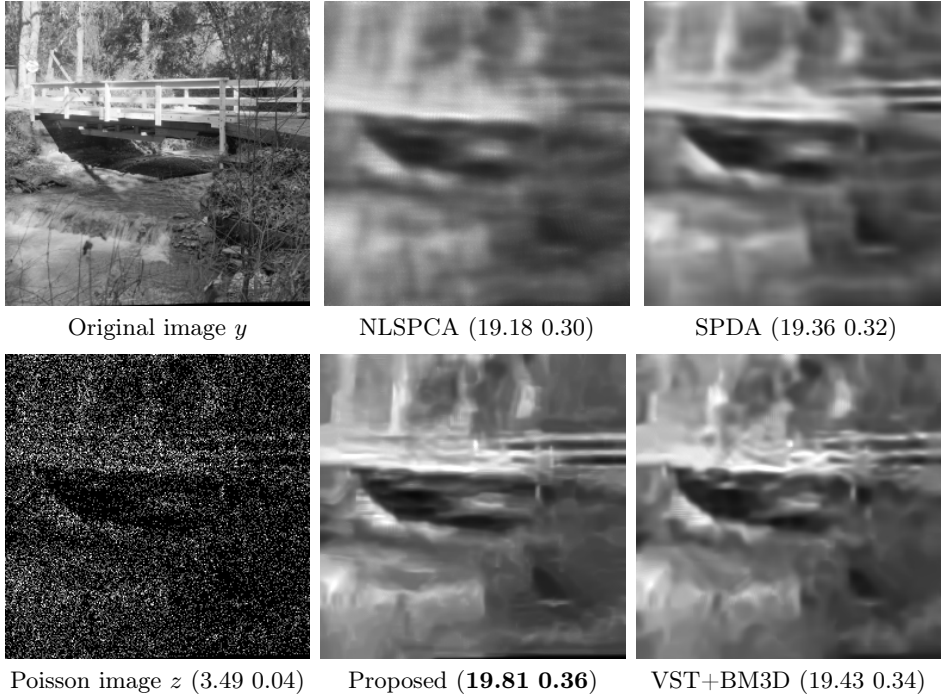


Figure 4.6: Denoising of *Bridge* at peak 1. PSNR (dB) and SSIM [Wang et al., 2004] are given in brackets. For clarity, z is visualized on a compressed range.

A relevant feature of the the proposed method is that its execution requires a tiny fraction of the time required by the other methods. The only algorithm as fast as the proposed one is the VST+BM3D method [Mäkitalo and Foi, 2011], that however performs poorly for very low counts. This marks an additional advantage compared to the other methods. The low execution time is mainly due to binning and to the fact that the algorithm performs at most 4 iterations.

In Figure 4.7, reproduced from [Publication IV](#), we show the robustness of the proposed framework with respect to the adopted AWGN denoising algo-

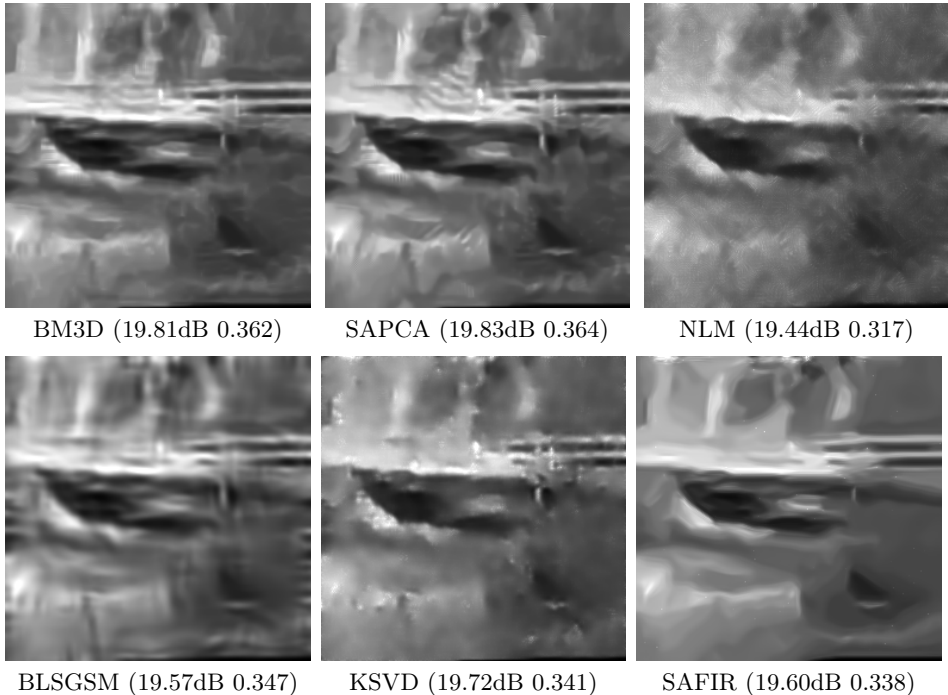


Figure 4.7: Denoising of *Bridge* at peak 1 adopting different AWGN filters. PSNR and SSIM of \hat{y} are given in brackets.

rithm; there, we give the denoising results (PSNR and SSIM) of our framework for 6 different filters: standard BM3D [Dabov et al., 2007b], BM3D with Shape-Adaptive Principal Components Analysis (SAPCA) [Dabov et al., 2009], Non-Local Means (NLM) [Buades et al., 2005], Structure-Adaptive Filtering for Image Restoration (SAFIR) [Boulanger et al., 2008], Bayesian Least Squares-Gaussian Scale Mixture (BLSGSM) [Portilla et al., 2003], K-SVD algorithm (KSVD) [Elad and Aharon, 2006]. Most of these results outperform those obtained by the algorithms used for comparison in Table I of Publication IV, proving the robustness of the proposed iterative framework.

Chapter 5

Discussion and Conclusions

The presented work provides novel and innovative solutions for two main problems in image processing: noise estimation and denoising.

The common framework for noise estimation found in literature is based on the assumption that the best way of estimating the noise parameters is to segment and isolate parts of the image with equal statistics, using then these statistics to fit the model. Based on this approach, in [Publication I](#) we propose a robust fitting that uses mixtures of random variables (Gaussian-Cauchy) to include in the model the probability of having outliers among the scatter-points. The mixture coefficient is also embedded in the optimization to incorporate the fact that different images might present different amount of outliers. Compared to its non-robust counterpart (where only Gaussian distributions are used) the presented algorithm gives better estimates, especially for challenging cases. A possible way to further improve the robust fitting would be to adopt different mixture coefficients for different local estimates, and to include them into the optimization step. This would give more adaptability to the algorithm, since it would model different scatter-points with different mixture distributions. However, the optimization of multiple mixture coefficients could produce unstable and/or over-fitted results, resulting in erroneous estimates. Moreover, it would also increase the number of parameters to be estimated in the optimization, hence increasing

significantly the execution time.

We then propose a novel approach to the problem of affine noise variance estimation for Poisson-Gaussian noise, in which we show that it is also possible to perform noise estimation by jointly processing heterogeneous elements [Publication II]. This idea has been validated theoretically and via experiments, based on an elementary prototype implementation. To improve the noise estimation accuracy, one could develop an estimation algorithm based on the same model, yet exploiting more sophisticated components 1) to perform signal decomposition and 2) to estimate the noise parameters from the scatterplot of the local estimates. The noise separation could be performed, for example, via PCA decomposition and shrinkage, while the fitting could be improved by formulating a statistical model of the local estimates, and by optimizing a likelihood function that incorporates those statistics (as proposed in Publication I).

Further, we develop a spatially adaptive transform for denoising AWGN based on the spatial information of similar blocks from an image. In particular, a group of similar blocks is transformed using an adaptive 1-D transform generated from orthogonal polynomials sampled at their spatial coordinates [Publication III]. The advantages of a spatially adaptive transform are especially appreciable in cases in which similar blocks lie on a smoothly varying area. We then exploit the blocks coordinates to develop an adaptive image enhancement algorithm based on alpha-rooting. Since usually aligned blocks contain an edge directed along the alignment, while scattered blocks are from an uniform region, we amplify via alpha-rooting only the coefficients whose basis functions have large variations orthogonal to the alignment direction (*i.e.* the principal direction of the blocks coordinates), while softening the basis functions that have large variations along the edge. In this way, we sharpen the features orthogonal to the edge, sharpening the edge itself, while we smooth along the direction of the edge. The current version of the algorithm only performs one iteration of denoising, stopping after hard-thresholding, therefore the next step would be the development of the Wiener counterpart.

Finally, in Publication IV, we consider the problem of denoising Poisson images. We propose an iterative algorithm based on the VST scheme that,

at each iteration, increases the SNR of the signal to be stabilized via convex combination with the previous estimate, and via binning. This increases the stabilization capabilities of the VST, that is inaccurate for low SNR. The presented algorithm outperforms the state of the art in terms of quality (PSNR and SSIM) and execution time. A natural evolution of the presented framework is the extension to spatial deconvolution of images corrupted by Poisson noise; we are currently working on the matter, and we are developing an iterative algorithm based on the same iterative binning+VST scheme for deblurring of images corrupted by Poisson noise.

5.1 Automatic noise estimation and denoising

As a fit conclusion to the thesis we present an experiment in which, automatically, we estimate the parameters of the noise affecting an image using the method from [Publication II](#), and we then denoise using the method proposed in [Publication IV](#). The image is the blue channel of a raw picture taken using a *Nokia N9* smartphone at ISO 3200.

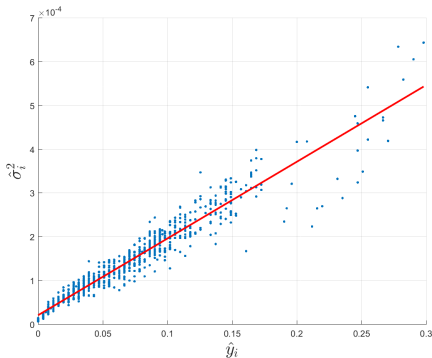


Figure 5.1: Automatic estimation of the noise variance affecting the blue channel of a raw image acquired with a *Nokia N9* smartphone at ISO 3200. We report the scatterplot of the local statistics (mean-variance pairs) and the estimated noise variance function.



Figure 5.2: Denoising of a raw image acquired with a *Nokia N9* smartphone at ISO 3200. Left: noisy image. Right: denoised image.

By modeling the noise as Poisson-Gaussian, we estimate the noise parameters a and b (2.20) using the algorithm from [Publication II](#). The estimated function is shown in red in Figure 5.1; in the same figure we also report the local statistics of the image. We then transform the noisy image z as

$$z' = \frac{1}{a}z + \frac{b}{a^2}, \quad (5.1)$$

to make expected value and variance of z' equal:

$$\mathbb{E} \{z'(x)\} = \frac{1}{a}y(x) + \frac{b}{a^2} = \text{var} \{z'(x)\}. \quad (5.2)$$

Thus, if we ignore the actual shape of the noise distribution, we can treat z' as a Poisson image and denoise it with the iterative algorithm from [Publication IV](#).

On the left side of Figure 5.2 we show the noisy observations z , while on the right side we show the denoising result.

The entire procedure was automatic and unsupervised.

Bibliography

- Amer, A. and Dubois, E. (2005). Fast and reliable structure-oriented video noise estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):113–118.
- Anscombe, F. J. (1948). The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254.
- Antoine, J. P., Vandergheynst, P., and Murenzi, R. (1996). Two-dimensional directional wavelets in image processing. *International journal of imaging systems and technology*, 7(3):152–165.
- Awate, S. P. and Whitaker, R. T. (2005). Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 44–51.
- Azzari, L. and Foi, A. (2014a). Gaussian-Cauchy mixture modeling for robust signal-dependent noise estimation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5357–5361.
- Azzari, L. and Foi, A. (2014b). Indirect Estimation of Signal-Dependent Noise With Nonadaptive Heterogeneous Samples. *IEEE Transactions on Image Processing*, 23(8):3459–3467.

BIBLIOGRAPHY

- Azzari, L. and Foi, A. (2015). Collaborative filtering based on group coordinates for smoothing and directional sharpening. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1573–1577.
- Azzari, L. and Foi, A. (2016a). Iterative Poisson image denoising software. [Online]. Available: www.cs.tut.fi/~foi/invansc/. Accessed: 07.09.2016.
- Azzari, L. and Foi, A. (2016b). Supplementary to the manuscript “Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising”. [Online]. Available: http://www.cs.tut.fi/~foi/papers/Azzari_Foi-Supplementary-Iterative_Poisson_denoising-2016.pdf. Accessed: 07.09.2016.
- Azzari, L. and Foi, A. (2016c). Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising. *IEEE Signal Processing Letters*, 23(8):1086–1090.
- Bar-Lev, S. K. and Enis, P. (1988). On the classical choice of variance stabilizing transformations and an application for a Poisson variate. *Biometrika*, 75(4):803–804.
- Bar-Lev, S. K. and Enis, P. (1990). On the construction of classes of variance stabilizing transformations. *Statistics & Probability Letters*, 10(2):95–100.
- Bartlett, M. S. (1936). The square root transformation in analysis of variance. *Supplement to the Journal of the Royal Statistical Society*, 3(1):68–78.
- Bashkansky, M., Burris, H. R., Funk, E. E., Mahon, R., and Moore, C. I. (2004). RF phase-coded random-modulation LIDAR. *Optics communications*, 231(1):93–98.
- Beylkin, G., Coifman, R., and Rokhlin, V. (1991). Fast wavelet transforms and numerical algorithms I. *Communications on pure and applied mathematics*, 44(2):141–183.

- Boracchi, G. and Foi, A. (2008). Multiframe raw-data denoising based on block-matching and 3-D filtering for low-light imaging and stabilization. In *Proceeding International Workshop on Local and Non-Local Approximation in Image Processing*, volume 1, pages 277–284.
- Boulanger, J., Kervrann, C., Bouthemy, P., Elbau, P., Sibarita, J.-B., and Salamero, J. (2010). Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. *IEEE transactions on medical imaging*, 29(2):442–454.
- Boulanger, J., Sibarita, J. B., Kervrann, C., and Bouthemy, P. (2008). Non-parametric regression for patch-based fluorescence microscopy image sequence denoising. In *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI 2008)*, pages 748–751.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.
- Boyle, W. S. and Smith, G. E. (1970). Charge coupled semiconductor devices. *The Bell System Technical Journal*, 49(4):587–593.
- Buades, A., Coll, B., and Morel, J. (2005). A Review of Image Denoising Algorithms, with a New One. *Multiscale Modeling and Simulation*, 4(2):490–530.
- Buades, A., Lebrun, M., and Morel, J. (2013). Implementation of the “Non-Local Bayes” (NL-Bayes) Image Denoising Algorithm. *Image Processing On Line*, 3:1–42.
- Cecil, T. W., Gades, L., Madden, T., Yan, D., and Miceli, A. (2015). Optimization of thermal kinetic inductance detectors for x-ray spectroscopy. *IEEE Transactions on Applied Superconductivity*, 25(3):2400805–2400809.

BIBLIOGRAPHY

- Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610.
- Coifman, R. and Meyer, Y. (1991). Ondelettes et opérateurs iii. *Opérateurs multilinéaires, Actualites Mathematiques, Hermann, Paris*.
- Collins, M., Dasgupta, S., and Schapire, R. E. (2001). A generalization of principal components analysis to the exponential family. *Advances in neural information processing systems*, pages 617–624.
- Curtiss, J. H. (1943). On transformations used in the analysis of variance. *The Annals of Mathematical Statistics*, 14(2):107–122.
- Dabov, K., Foi, A., and Egiazarian, K. (2007a). Video denoising by sparse 3D transform-domain collaborative filtering. In *Signal Processing Conference, 2007 15th European*, pages 145–149.
- Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2006). Image denoising with block-matching and 3D filtering. *Proceedings SPIE*, 6064:606414–606425.
- Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007b). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095.
- Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007c). Joint image sharpening and denoising by 3D transform-domain collaborative filtering. In *Proceedings 2007 International TICSP Workshop Spectral Mathematical Multirate Signal Processings (SMMSP)*. Citeseer.
- Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2009). BM3D Image Denoising with Shape-Adaptive Principal Component Analysis. In *Proceedings 2009 workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS’09)*.

- Damadian, R. (1971). Tumor Detection by Nuclear Magnetic Resonance. *Science*, 171(3976):1151–1153.
- Danielyan, A., Katkovnik, V., and Egiazarian, K. (2010). Image deblurring by augmented Lagrangian with BM3D frame prior. In *Workshop on Information Theoretic Methods in Science and Engineering (WITMSE), Tampere, Finland*, pages 16–18.
- Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7):909–996.
- De Bonet, J. S. (1997). Noise reduction through detection of signal redundancy. *Rethinking Artificial Intelligence, MIT AI Lab, Tech. Rep.*
- Deledalle, C. A., Duval, V., and Salmon, J. (2012). Non-local Methods with Shape-Adaptive Patches (NLM-SAP). *Journal of Mathematical Imaging and Vision*, 43(2):103–120.
- Donoho, D. L. (1995). De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627.
- Donoho, D. L. and Johnstone, I. M. (1995). Adapting to Unknown Smoothness via Wavelet Shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224.
- Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745.
- Fan, J. and Gijbels, I. (1996). *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, volume 66. CRC Press.
- Feng, X. and Milanfar, P. (2002). Multiscale principal components analysis for image local orientation estimation. In *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002*, volume 1, pages 478–482.

BIBLIOGRAPHY

- Foi, A. (2009). Optimization of variance-stabilizing transformations. *Preprint, 2009b*.
- Foi, A., Dabov, K., Katkovnik, V., and Egiazarian, K. (2006). Shape-adaptive DCT for denoising and image reconstruction. In *Electronic Imaging 2006*, pages 60640N–60640N. International Society for Optics and Photonics.
- Foi, A., Katkovnik, V., and Egiazarian, K. (2007). Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images. *IEEE Transactions on Image Processing*, 16(5):1395–1411.
- Foi, A., Trimeche, M., Katkovnik, V., and Egiazarian, K. (2008). Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754.
- Fowler, B., Gamal, A. E., Yang, D., and Tian, H. (1998). A method for estimating quantum efficiency for cmos image sensors. In *in Proceedings of SPIE*.
- Freeman, M. F. and Tukey, J. W. (1950). Transformations related to the angular and the square root. *The Annals of Mathematical Statistics*, pages 607–611.
- Fryzlewicz, P. and Nason, G. P. (2004). A Haar-Fisz algorithm for Poisson intensity estimation. *Journal of computational and graphical statistics*, 13(3):621–638.
- Giryès, R. and Elad, M. (2014). Sparsity-Based Poisson Denoising With Dictionary Learning. *IEEE Transactions on Image Processing*, 23(12):5057–5069.
- Goldenshluger, A. and Nemirovski, A. (1997). On spatially adaptive estimation of nonparametric regression. *Mathematical methods of Statistics*, 6(2):135–170.

- Gonzales, R. C. and Wintz, P. (1987). *Digital Image Processing, Second Edition*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Gravel, P., Beaudoin, G., and De Guise, J. A. (2004). A method for modeling noise in medical images. *IEEE Transactions on Medical Imaging*, 23(10):1221–1232.
- Haar, A. (1911). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 71(1):38–53.
- Hackwell, J. A., Warren, D. W., Bongiovi, R. P., Hansel, S. J., Hayhurst, T. L., Mabry, D. J., Sivjee, M. G., and Skinner, J. W. (1996). LWIR/MWIR imaging hyperspectral sensor for airborne and ground-based remote sensing. In *SPIE's 1996 International Symposium on Optical Science, Engineering, and Instrumentation*, pages 102–107. International Society for Optics and Photonics.
- Jaffray, D. A., Siewerdsen, J. H., Wong, J. W., and Martinez, A. A. (2002). Flat-panel cone-beam computed tomography for image-guided radiation therapy. *International Journal of Radiation Oncology, Biology, Physics*, 53(5):1337–1349.
- Jähne, B. (2004). *Practical Handbook on Image Processing for Scientific and Technical Applications, Second Edition*. CRC Press, Inc., Boca Raton, FL, USA.
- Johnson, J. B. (1928). Thermal agitation of electricity in conductors. *Physical Review*, 32:97–109.
- Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994). *Continuous multivariate distributions, models and applications*, volume 1. New York: John Wiley & Sons.
- Katkovnik, V., Egiazarian, K., and Astola, J. (2006). Local approximation techniques in signal and image processing. *SPIE Bellingham*.

BIBLIOGRAPHY

- Katkovnik, V., Foi, A., Egiazarian, K., and Astola, J. (2004). Directional varying scale approximations for anisotropic signal processing. In *12th European Signal Processing Conference, 2004*, pages 101–104.
- Kervrann, C. and Boulanger, J. (2006). Optimal spatial adaptation for patch-based image denoising. *IEEE Transactions on Image Processing*, 15(10):2866–2878.
- Kervrann, C. and Boulanger, J. (2008). Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *International Journal of Computer Vision*, 79(1):45–69.
- Koczyk, P., Wiewiór, P., and Radzewicz, C. (1996). Photon counting statistics - Undergraduate experiment. *American Journal of Physics*, 64(3):240–244.
- Labate, D., Lim, W. Q., Kutyniok, G., and Weiss, G. (2005). Sparse multi-dimensional representation using shearlets. In *Optics & Photonics 2005*, pages 59140U–59140U. International Society for Optics and Photonics.
- Lange, R. and Seitz, P. (2001). Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics*, 37(3):390–397.
- Lee, J. S. (1981). Refined filtering of image noise using local statistics. *Computer graphics and image processing*, 15(4):380–389.
- Lee, J. S. and Hoppel, K. (1989). Noise modeling and estimation of remotely-sensed images. *Geoscience and Remote Sensing Symposium, 1989. IGARSS'89. 12th International Canadian Symposium on Remote Sensing.*, 2:1005–1008.
- Liu, C. and Freeman, W. T. (2010). A high-quality video denoising algorithm based on reliable motion estimation. In *European conference on computer vision*, pages 706–719. Springer.
- Liu, C., Freeman, W. T., Szeliski, R., and Kang, S. B. (2006). Noise estimation from a single image. In *Computer Vision and Pattern Recognition*,

- 2006 *IEEE Computer Society Conference on*, volume 1, pages 901–908. IEEE.
- Maggioni, M., Sánchez-Monge, E., and Foi, A. (2014). Joint removal of random and fixed-pattern noise through spatiotemporal video filtering. *IEEE Transactions on Image Processing*, 23(10):4282–4296.
- Mäkitalo, M. and Foi, A. (2011). Optimal Inversion of the Anscombe Transformation in Low-Count Poisson Image Denoising. *IEEE Transactions on Image Processing*, 20(1):99–109.
- Mäkitalo, M. and Foi, A. (2014). Noise Parameter Mismatch in Variance Stabilization, With an Application to Poisson-Gaussian Noise Estimation. *IEEE Transactions on Image Processing*, 23(12):5348–5359.
- Mandel, L. (1959). Fluctuations of Photon Beams: The Distribution of the Photo-Electrons. *Proceedings of the Physical Society*, 74(3):233.
- Mastin, G. A. (1985). Adaptive filters for digital image noise smoothing: An evaluation. *Computer Vision, Graphics, and Image Processing*, 31(1):103–121.
- Meer, P., Jolion, J. M., and Rosenfeld, A. (1990). A fast parallel algorithm for blind estimation of noise variance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(2):216–223.
- Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240.
- Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142.
- Nakamura, J. (2005). *Image sensors and signal processing for digital still cameras*. CRC press.
- Noble, P. J. W. (1968). Self-scanned silicon image detector arrays. *IEEE Transactions on Electron Devices*, 15(4):202–209.

BIBLIOGRAPHY

- Nyquist, H. (1928). Thermal agitation of electric charge in conductors. *Physical Review*, 32:110–113.
- Oh, A. K., Harmany, Z. T., and Willett, R. M. (2014). To e or not to e in poisson image reconstruction. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2829–2833.
- Oktem, R., Yaroslavsky, L., and Egiazarian, K. (1998). Signal and image denoising in transform domain and wavelet shrinkage: A comparative study. In *9th European Signal Processing Conference (EUSIPCO 1998)*, pages 1–4. IEEE.
- Papoulis, A. and Pillai, S. U. (2002). *Probability, random variables, and stochastic processes, Third Edition*. Tata McGraw-Hill Education.
- 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.
- Pyatykh, S. and Hesser, J. (2015). MMSE Estimation for Poisson Noise Removal in Images. *arXiv e-print*.
- Pyatykh, S., Hesser, J., and Zheng, L. (2013). Image Noise Level Estimation by Principal Component Analysis. *IEEE Transactions on Image Processing*, 22(2):687–699.
- Rond, A., Giryes, R., and Elad, M. (2015). Poisson Inverse Problems by the Plug-and-Play scheme. *arXiv e-print*.
- Salmon, J., Harmany, Z., Deledalle, C. A., and Willett, R. (2014). Poisson noise reduction with non-local PCA. *Journal of Mathematical Imaging and Vision*, 48(2):279–294.
- Starck, J. L., Candès, E. J., and Donoho, D. L. (2002). The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684.

-
- Starck, J. L., Murtagh, F., and Bijaoui, A. (1998). *Image Processing and Data Analysis: The Multiscale Approach*, chapter Appendix A1, pages 263–265. Cambridge University Press.
- Stone, C. J. (1977). Consistent nonparametric regression. *The annals of statistics*, pages 595–620.
- Sutour, C., Deledalle, C. A., and Aujol, J. F. (2015). Estimation of the Noise Level Function Based on a Nonparametric Detection of Homogeneous Image Regions. *SIAM Journal on Imaging Sciences*, 8(4):2622–2661.
- Tijms, H. (2007). *Understanding Probability: Chance Rules in Everyday Life, Second Edition*. Cambridge University Press Cambridge.
- Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846.
- Venkatakrishnan, S., Bouman, C. A., and Wohlberg, B. (2013). Plug-and-play priors for model based reconstruction. *IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2013*, pages 945–948.
- Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.
- Watson, G. S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*:359–372.
- Weickert, J. (1999). Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2-3):111–127.
- Weissman, M. B. (1988). $\frac{1}{f}$ noise and other slow, nonexponential kinetics in condensed matter. *Review of Modern Physics*, 60:537–571.
- Yaroslavsky, L., Egiazarian, K., and Astola, J. (2001). Transform domain image restoration methods: review, comparison, and interpretation. In

BIBLIOGRAPHY

Photonics West 2001-Electronic Imaging, pages 155–169. International Society for Optics and Photonics.

Zhang, B., Fadili, J. M., and Starck, J.-L. (2008). Wavelets, ridgelets, and curvelets for Poisson noise removal. *IEEE Transaction on Image Processing*, 17(7):1093–1108.

Publication I

Azzari, L. and Foi, A. (2014a). Gaussian-Cauchy mixture modeling for robust signal-dependent noise estimation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5357-5361.

© 2014 Institute of Electrical and Electronics Engineers (IEEE). Reprinted, with permission, from IEEE International Conference on Acoustics, Speech and Signal Processing.

**GAUSSIAN-CAUCHY MIXTURE MODELING FOR
ROBUST SIGNAL-DEPENDENT NOISE ESTIMATION**

Lucio Azzari and Alessandro Foi

Department of Signal Processing, Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland

ABSTRACT

We introduce an adaptive Gaussian-Cauchy mixture modeling for the likelihood of pairwise mean/standard-deviation scatter points found when estimating signal-dependent noise. The maximization of the likelihood is used to identify the noise-model parameters, following an adaptive mixture parameter that controls the balance between the Gaussian and the heavy-tailed Cauchy. This renders the estimation robust with respect to outliers, typically present in large quantities among the scatter points from images dominated by texture. The modeling is directly suited to describing also observations subject to clipping, i.e. under- or over-exposure. Experiments on a dataset of badly exposed and highly textured images demonstrate the effectiveness of the adaptive Gaussian-Cauchy mixture likelihood for the accurate estimation of the noise standard-deviation curve.

Index Terms— Signal-dependent noise, robust estimation, mixture modeling, clipping

1. INTRODUCTION

With signal-dependent noise we refer to a model of noise with variable standard deviation that depends on the particular intensity value of the affected signal. This model has been widely adopted by the scientific and engineering community, because it well approximates the noise affecting data acquired by many sensing devices, including imaging sensors such as CCD and CMOS cameras.

The common procedure [1–8] for signal-dependent noise estimation consists in dividing the data or image of interest into uniform or homogeneous regions. Each group of samples is then used for the estimation of a mean-standard deviation pairs. The collection of such pairs yields a cloud of points scattered around the curve that describes the dependency of the noise standard deviation from the mean of the signal. Finally, in order to estimate the noise standard-deviation curve, a global parametric model is fitted to the points.

The global fitting can be hampered by outliers among the mean and standard deviation pairs. This problem becomes

particularly relevant when dealing with images with numerous highly textured regions.

The estimation of the standard-deviation curve is further complicated by the inevitably limited range of the sensing device, which causes clipping of data whenever it falls outside an admissible interval. In imaging, this phenomenon may occur when the scene is under- or over-exposed.

To provide a unique and reliable solution to these two distinct problems, we introduce a novel likelihood function based on an adaptive Gaussian-Cauchy mixture modeling of the mean and standard-deviation pairs. The presented work generalizes the method [1], significantly improving its robustness against outliers. The estimation algorithm based on the introduced Gaussian-Cauchy mixture likelihood provides accurate estimates of the noise standard-deviation curve from images that can be badly exposed as well as dominated by texture.

The remainder of the paper is organized as follows. In Section 2, we introduce the observation model for signal-dependent noise, also considering the case of clipped data. The important affine variance model is presented too. Section 3 briefly describes the first part of the algorithm, which deals with the estimation of the mean-standard deviation pairs that constitute the scatterplot. The core of our contribution is given in Section 4, with a complete description of the proposed robust Gaussian-Cauchy mixture likelihood. The maximization of the likelihood with an adaptive mixture parameter is also explained. Section 5 provides experimental validation of our approach over a dataset of real raw images affected by clipping and high-frequency texture, confirming the effectiveness of the proposed adaptive mixture modeling.

2. PROBLEM STATEMENT

Let us consider a noisy image z as the sum of a noise-free image y and noise with signal-dependent standard deviation $\sigma(y)$,

$$z(\mathbf{x}) = y(\mathbf{x}) + \sigma(y(\mathbf{x}))\xi(\mathbf{x}), \quad (1)$$

where $\mathbf{x} \in \mathbf{X} \subset \mathbb{N}^2$ denotes the spatial coordinate of a pixel, and $\xi(\mathbf{x})$ is a standardized random variable with zero-mean and unitary variance. We model the noise as spatially independent (i.e. diagonal covariance matrix). For the sake of simplicity and in agreement with the analysis in [1, 9], we

Contact info: firstname.lastname@tut.fi

© This work was supported by the Academy of Finland (project no. 252547).

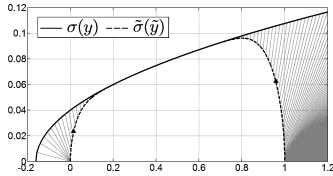


Fig. 1: Effect of clipping on the standard deviation function in case of affine variance (2) [9]. Due to clipping, the standard deviation $\tilde{\sigma}(\hat{y})$ goes to zero as it approaches the bounds (0, 1).

treat $\xi(\mathbf{x})$ as a standard normal, i.e. $\xi(\mathbf{x}) \sim \mathcal{N}(0, 1)$, so that only the standard deviation of $\sigma(y(\mathbf{x}))\xi(\mathbf{x})$ depends on the particular value of the noise-free signal $y(\mathbf{x})$.

Our goal is to robustly estimate the function $\sigma: \mathbb{R} \rightarrow \mathbb{R}^+$ that links the noise standard deviation $\sigma(y)$ to the signal y .

A peculiar and recurrent signal-dependent noise model is the affine variance model that well approximates the noise affecting digital image sensors:

$$\sigma^2(y(\mathbf{x})) = ay(\mathbf{x}) + b. \quad (2)$$

According to this model, the noise $\eta(y(\mathbf{x}))$ has variance that affinely depends on the signal intensity. In this case, the estimation of the curve $\sigma(y)$ can be reduced to the estimation of a and b .

The output of any imaging device has a limited range, which, without loss of generality, we assume rescaled to the $[0, 1]$ interval. Here, 0 and 1 are the lowest (i.e. darkest) and highest (i.e. brightest) values, respectively. Values of z that exceed these bounds are inevitably *clipped*, i.e. they are replaced by the bounds themselves:

$$\tilde{z}(\mathbf{x}) = \max(0, \min(z(\mathbf{x}), 1)). \quad (3)$$

Throughout the text, the *tilde* decoration $\tilde{\cdot}$ is used to indicate variables and quantities subject to clipping.

The noise model for the observed clipped image \tilde{z} can be expressed as

$$\tilde{z}(\mathbf{x}) = \tilde{y}(\mathbf{x}) + \tilde{\sigma}(\tilde{y}(\mathbf{x}))\tilde{\xi}(\mathbf{x}), \quad (4)$$

where \tilde{y} is the expectation of the clipped data \tilde{z} , and $\tilde{\sigma}(\tilde{y}(\mathbf{x}))$ is the standard deviation of the clipped noisy data. Clipping causes an apparent distortion of the standard-deviation curve that describes the signal-dependent noise model, as illustrated in Fig. 1. In particular, the noise standard deviation approaches zero whenever the intensity approaches the range bounds 0 and 1. A comprehensive analysis of the interplay between clipping and signal-dependent noise models can be found in [9].

The difference between $\tilde{\sigma}(\tilde{y}(\mathbf{x}))$ and $\sigma(y(\mathbf{x}))$ constitutes a dramatic deviation from the traditional affine-variance model (2). This makes the estimation of the noise parameters particularly difficult when portions of the image are not correctly exposed.

3. LOCAL ESTIMATION OF MEAN-STANDARD DEVIATION PAIRS

Overall, the proposed approach is characterized by two main stages: first, an estimation of a collection of expectation/standard deviation pairs (i.e. the construction of a scatterplot); and, second, the maximization of a likelihood model that explains these estimates (i.e. the fitting of a global parametric curve to the scatterplot). Our present contribution concerns the latter stage only, as described in detail in the next section. For the former stage, we leverage, without modification, the first stage of the original algorithm [1], which we briefly summarize here below.

The pairwise estimates of standard deviation and mean are computed from a high-pass and from low-pass version of the image, respectively denoted as \tilde{z}^{wdet} and \tilde{z}^{wapp} . These are obtained by convolving \tilde{z} with a 2-D wavelet function ψ and its scaling function φ , respectively:

$$\tilde{z}^{\text{wdet}} = \downarrow_2(\tilde{z} \otimes \psi) \quad \text{and} \quad \tilde{z}^{\text{wapp}} = \downarrow_2(\tilde{z} \otimes \varphi), \\ \sum \psi = 0, \quad \|\psi\|_2 = 1, \quad \sum \varphi = 1,$$

where \downarrow_2 is the decimation operator, here used in order to discard every second row and second column of an image.

It can be shown [1] that the standard deviation (std) of the detail coefficients is comparable to plugging the expectation of the approximation coefficients into the standard deviation function $\tilde{\sigma}$:

$$\text{std}\{\tilde{z}^{\text{wdet}}\} \simeq \tilde{\sigma}(\mathbb{E}\{\tilde{z}^{\text{wapp}}\}). \quad (5)$$

The approximation (5) becomes locally accurate in uniform regions of \tilde{z} . Therefore, in order to reliably compute the local mean-variance pairs excluding texture areas (non-uniform regions), it is necessary to segment \tilde{z} .

For the sake of brevity, we refer the reader to [1] for the technical details about the segmentation. The obtained segments S_i , $i = 1, \dots, N$, include pixels whose expected intensity values belong to a narrow intensity interval, and can thus be treated as level sets.

Finally, for each level set S_i , we estimate its mean \hat{y}_i and standard deviation $\hat{\sigma}_i$ as the sample mean and sample median of \tilde{z}^{wapp} and $|\tilde{z}^{\text{wdet}}|/0.6745$ restricted on S_i , respectively.

4. ROBUST ML FITTING OF A GLOBAL PARAMETRIC MODEL

The global optimization step requires a model of the distributions of \hat{y}_i and $\hat{\sigma}_i$. In the ideal case without outliers, the estimates follow normal distributions of the forms, respectively,

$$\hat{y}_i \sim \mathcal{N}(\hat{y}_i, \hat{\sigma}^2(\hat{y}_i)c_i), \quad \hat{\sigma}_i \sim \mathcal{N}(\hat{\sigma}(\hat{y}_i), \hat{\sigma}^2(\hat{y}_i)d_i) \quad (6)$$

where the coefficients c_i and d_i are scaling parameters that depend on the number of samples n_i in the level set S_i [1]. However, in case of images dominated by texture, the high-frequency components of y can introduce severe outliers in

the local estimates $\hat{\sigma}_i$, and consequently mislead the estimation of the noise model parameters, as illustrated in Fig. 3. The Cauchy PDF is characterized by heavy tails, which makes it suitable for estimation in presence of outliers.

The PDFs of the estimates are therefore expressed as

$$\begin{aligned} \varphi(\hat{y}_i | \tilde{y}_i = \tilde{y}) &= g_N(\hat{y}_i; \tilde{y}, \hat{\sigma}^2(\tilde{y}) c_i), \\ \varphi(\hat{\sigma}_i | \tilde{y}_i = \tilde{y}) &= (1 - \lambda) g_N(\hat{\sigma}_i; \hat{\sigma}(\tilde{y}), \hat{\sigma}^2(\tilde{y}) d_i) \\ &\quad + \lambda g_C(\hat{\sigma}_i; \hat{\sigma}(\tilde{y}), \hat{\sigma}^2(\tilde{y}) d_i), \end{aligned} \quad (7)$$

where g_N and g_C are the normal and Cauchy PDFs with location parameter μ and scaling parameter ν :

$$\begin{aligned} g_N(x; \mu, \nu^2) &= \frac{1}{\sqrt{2\pi\nu^2}} e^{-\frac{(x-\mu)^2}{2\nu^2}}, \\ g_C(x; \mu, \nu^2) &= \frac{\nu}{\pi \left(\nu^2 + (x-\mu)^2 \right)}, \end{aligned}$$

and $\lambda \in [0, 1]$ is the mixture coefficient between these distributions.

The joint probability density can be expressed as the product

$$\varphi((\hat{y}_i, \hat{\sigma}_i) | \tilde{y}_i = \tilde{y}) = \varphi(\hat{y}_i | \tilde{y}_i = \tilde{y}) \varphi(\hat{\sigma}_i | \tilde{y}_i = \tilde{y}).$$

Given the distributions of all the pairs $\{(\hat{y}_i, \hat{\sigma}_i)\}_{i=1}^N$, we finally obtain the posterior likelihood function \tilde{L} integrating all the densities $\varphi((\hat{y}_i, \hat{\sigma}_i) | \tilde{y}_i = \tilde{y})$ with respect to the prior density $\varphi_0(y)$ of y as

$$\tilde{L}(\theta, \lambda) = \prod_{i=1}^N \int_{-\infty}^{\infty} \varphi((\hat{y}_i, \hat{\sigma}_i) | \tilde{y}_i = \tilde{y}) \varphi_0(y) dy, \quad (8)$$

where θ is an m -dimensional vector composed by the model parameters to be estimated. The vector θ determines univocally both the clipped standard-deviation curve $\hat{\sigma}(\tilde{y})$ and the (unclipped) standard-deviation curve $\sigma(y)$. E.g., $\theta = [a, b]$, i.e. $m = 2$, in case of the affine mean-variance relation (2). The integration in (8) makes the distributions independent from the unknown value of y . Similar to [1], in our experiments we assume a uniform prior density φ_0 .

A principal feature of the proposed approach consists in treating the mixture coefficient λ as an unknown variable. Therefore, our optimization is performed with respect to $m + 1$ parameters, i.e. the vector θ and the mixture coefficient λ :

$$\hat{\theta} = \arg \max_{\theta \in \mathbb{R}^m, \lambda \in [0, 1]} \tilde{L}(\theta, \lambda). \quad (9)$$

Note that the maximization of \tilde{L} corresponds to optimizing the fit of the clipped standard-deviation curve $\hat{\sigma}(\tilde{y})$ to the scatterplot, because the probabilities of the points $(\hat{y}_i, \hat{\sigma}_i)$ are determined by this curve through the PDFs (7). The value of λ upon maximization of (9), denoted as $\hat{\lambda}$, provides an indication of the amount of outliers in the scatterplot. When estimating noise in an image dominated by texture, one can expect to obtain $\hat{\lambda} \approx 1$, i.e. the mixture reduces to a Cauchy distribution; conversely, if the image is mostly smooth or piecewise smooth, $\hat{\lambda} \approx 0$, i.e. the mixture reduces to a Gaussian.

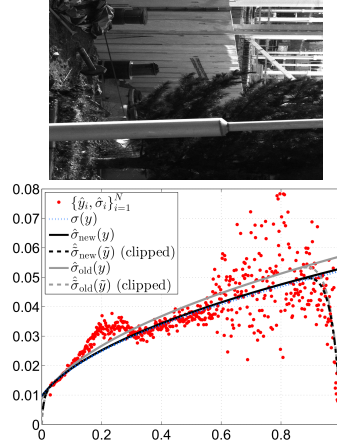


Fig. 2: Top: Raw image (blue channel) with extensive portions of high-frequency texture and overexposed areas. The image was captured with a Fujifilm FinePix S9600 camera at ISO 800. Bottom: Mean vs. standard deviation diagram. The red scatterplot points correspond to individual mean and standard-deviation estimates $(\hat{y}_i, \hat{\sigma}_i)$. Note the presence of outliers in the scatterplot. The standard-deviation curves $\hat{\sigma}_{old}(\tilde{y})$ and $\hat{\sigma}_{new}(\tilde{y})$ fitted through maximization of the original and of the proposed robust likelihood functions are shown as dashed lines. The respective *unclipped* curves $\hat{\sigma}_{old}(y)$ and $\hat{\sigma}_{new}(y)$ are drawn as solid lines and are compared against the ground truth $\sigma(y)$ (dotted line).

4.1. Numerical Solution

We solve (9) numerically, using the Nelder-Mead iterative downhill simplex method [10], under the constraint $\lambda \in [0, 1]$. Similar to [1], we initialize the optimization from a very rough least-square (LS) solution. For what concerns λ , in our experiments we use 1 as initial value, in order to account for the worst-case scenario of a scatterplot corrupted by several outliers.

We remark that the need for solving (9) numerically does not follow from introducing the Cauchy term in the mixture, but was already necessary with the Gaussian-only model [1] due to the heteroskedasticity.

5. EXPERIMENTS ON HIGHLY TEXTURED IMAGES

To validate the effectiveness of the robust likelihood with adaptive mixture of Gaussian-Cauchy distributions, we com-

pare the proposed approach against the original algorithm [1], which is based on a Gaussian likelihood (i.e. fixed $\lambda = 0$). The corresponding estimated standard-deviation curves are denoted as $\hat{\sigma}_{new}(\bar{y})$ and $\hat{\sigma}_{old}(\bar{y})$ for the clipped model (4), and as $\hat{\sigma}_{new}(y)$ and $\hat{\sigma}_{old}(y)$ for the unclipped model (1), respectively. A typical example of a clipped highly textured image is shown in Fig. 2. The image has been taken with a Fujifilm S9600 camera at ISO 800. The effect of clipping can be easily seen at the right in the scatterplot, where the red dots drop towards the limiting coordinate (0, 1).

The textures in the image cause overestimation of the standard deviations, producing outliers, which can be seen in the scatterplot particularly for $\bar{y} \in [0.1, 0.3] \cup [0.6, 0.9]$. The original algorithm [1] results in the *unclipped* standard-deviation curve $\hat{\sigma}_{old}(y)$ (solid gray line), which misestimates the ground-truth curve $\sigma(y)$ ¹ (dotted line). On the contrary, the proposed mixture model is by design robust to outliers in the scatterplot, and $\hat{\sigma}_{new}(y)$ is in good agreement with the ground-truth $\sigma(y)$. The estimated λ is 1; which indicates that the amount of outliers is significant and justifies the use of a heavy-tailed distribution for modeling the likelihood.

We have also verified the advantage of the proposed robust algorithm on the NED dataset [11]. This challenging dataset consists of 25 heavily textured raw images captured with a Nikon D80 camera, at ISO values from 100 to 320. In Table 1 we report the average normalized integral error (ANIE) over all images in the dataset, each channel separately. The normalized integral errors are computed as

$$NIE = \int_{\Omega} \frac{|\hat{\sigma}(\bar{y}) - \sigma(\bar{y})|}{\hat{\sigma}(\bar{y})} d\bar{y} \Big/ \int_{\Omega} 1 d\bar{y},$$

where $\hat{\sigma}(\bar{y})$ is the estimated clipped standard deviation curve, $\sigma(\bar{y})$ is the ground-truth curve, and $\Omega = \{\bar{y} | \hat{\sigma}(\bar{y}) > 0\}$.

The proposed robust algorithm systematically improves over the previous one. While the numerical values in the table may be difficult to appreciate, Fig. 3 gives a clear illustration of the kind of misestimation problems resolved through the proposed adaptive Gaussian-Cauchy mixture modeling. As can be seen in the figure, large regions of texture yield severe outliers in the estimation of mean-standard deviation pairs and a considerable amount of points in the scatterplot are far from the correct parametric trend.

Matlab codes implementing the proposed algorithm are available [12].

6. DISCUSSION AND CONCLUSIONS

The experiments and results described in Section 5 confirm the advantage of adopting a robust global fitting model when dealing with highly textured images. The presented algorithm, with its adaptive Gaussian-Cauchy mixture, is robust to outliers and ensures more reliable results than the previous

¹The ground-truth curve has been estimated from an ideal texture-free image obtained with out-of-focus optics, as described in [1].

ANIE	Red	Green	Blue
Proposed	0.3088	0.6036	0.3157
Old	0.4399	0.8387	0.3708

Table 1: Average normalized integral errors (ANIE) from the estimates on the NED dataset. The results for each channel is shown separately.

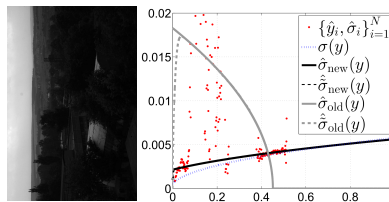


Fig. 3: Underexposed raw image (red channel) with intensity range normalized to $[0, 1]$ from the NED dataset [11], and the relative scatterplot of mean-standard deviation pairs, severely corrupted by outliers. The noise curve $\hat{\sigma}_{old}$ estimated by the method [1] is directly impacted by the outliers and results in a failure. The proposed method provides instead a robust fit to the scatterplot and the estimated curve $\hat{\sigma}_{new}$, although not perfect, achieves a reasonable approximation of the underlying noise model.

algorithm based on a Gaussian-only modeling. The benefit of the presented approach is particularly significant in challenging cases with severe outliers. Through the automatic optimization of the mixture parameter, we conveniently encompass also less problematic images attaining high precision.

As can be seen in the figures, the outliers $\hat{\sigma}_i$ do typically lie in the right heavy tail (i.e. overestimation) of the conditional PDF (7) given the true $\bar{\sigma}$. Therefore, one may question the role of the left heavy tail. However, when maximizing the likelihood (8) starting from an inaccurate guess $\hat{\theta}$, it can happen that the candidate function $\hat{\sigma}$ fits the overestimated $\hat{\sigma}_i$, while the non-overestimated values appear instead as outliers in the left tail of the PDF (7). Thus, the left heavy tail effectively prevents (9) from getting stuck at such local maxima.

The proposed likelihood model is relatively independent of the procedure utilized to produce the scatterplot pairs; thus, the proposed optimization is relevant to a wide class of algorithms based on similar fitting of a standard-deviation curve. A special feature of our model consists in the explicit treatment of the clipping due to under- or over-exposure (3)(4), commonly encountered with imagers having limited dynamic range. Moreover, while in this paper we have considered only the affine variance model (2), we note that the procedure is independent of the specific parametric model between θ and σ^2 (1). In particular, the algorithm [12] can handle any polynomial order.

7. REFERENCES

- [1] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data," *Image Processing, IEEE Transactions on*, vol. 17, no. 10, pp. 1737–1754, 2008.
- [2] P. Gravel, G. Beaudoin, and J.A. De Guise, "A method for modeling noise in medical images," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 10, pp. 1221–1232, 2004.
- [3] S.I. Olsen, "Estimation of Noise in Images: An Evaluation," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 4, pp. 319 – 323, 1993.
- [4] S.K. Abramov, B. Vozel, J.T. Astola, K. Chehdi, and V.V. Lukin, "Segmentation-based method for blind evaluation of noise variance in images," *Journal of Applied Remote Sensing*, vol. 2, no. 1, pp. 023533–023533–16, 2008.
- [5] M. Lebrun, M. Colom, A. Buades, and J. M. Morel, "Secrets of image denoising cuisine," *Acta Numerica*, vol. 21, pp. 475–576, 5 2012.
- [6] J. Immerkaer, "Fast Noise Variance Estimation," *Computer Vision and Image Understanding*, vol. 64, no. 2, pp. 300–302, 1996.
- [7] B. Aiazzi, L. Alparone, S. Baronti, M. Selva, and L. Stefani, "Unsupervised estimation of signal-dependent CCD camera noise," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 231, 2012.
- [8] N. Acito, M. Diani, and G. Corsini, "Signal-Dependent Noise Modeling and Model Parameter Estimation in Hyperspectral Images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 8, pp. 2957–2971, 2011.
- [9] A. Foi, "Clipped noisy images: Heteroskedastic modeling and practical denoising," *Signal Processing*, vol. 89, no. 12, pp. 2609 – 2629, 2009.
- [10] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.
- [11] M.L. Uss, B. Vozel, V.V. Lukin, and K. Chehdi, "NED2012 Database," <http://rsd.khai.edu/ned2012/ned2012.php>.
- [12] A. Foi and L. Azzari, "ClipPoisGaus_stdEst2D," <http://www.cs.tut.fi/~foi/sensornoise.html>, Matlab software.

Publication II

Azzari, L. and Foi, A. (2014b). Indirect Estimation of Signal-Dependent Noise With Nonadaptive Heterogeneous Samples. *IEEE Transactions on Image Processing*, 23(8):3459-3467.

© 2014 Institute of Electrical and Electronics Engineers (IEEE). Reprinted, with permission, from IEEE Transaction on Image Processing.

Indirect estimation of signal-dependent noise with non-adaptive heterogeneous samples

Lucio Azzari* and Alessandro Foi
 Department of Signal Processing, Tampere University of Technology
 P.O. Box 553, FIN-33101 Tampere, Finland

Abstract—We consider the estimation of signal-dependent noise from a single image. Unlike conventional algorithms that build a scatterplot of local mean-variance pairs from either small or adaptively selected homogeneous data samples, our proposed approach relies on arbitrarily large patches of heterogeneous data extracted at random from the image. We demonstrate the feasibility of our approach through an extensive theoretical analysis based on mixture of Gaussian distributions. A prototype algorithm is also developed in order to validate the approach on simulated data as well as on real camera raw images.

Index Terms—Noise estimation, signal-dependent noise, Poisson noise.

I. INTRODUCTION

The popularity of signal-dependent noise models, in which the variance of the noise affecting the signal depends on the mean of the signal, is based on the fact that they well approximate noise affecting data of several kinds of acquisition devices, e.g., raw data from a CCD camera. Figure 1 illustrates how the signal-dependent noise differently affects bright and dark regions of an image, and shows a curve that describes the typical mean-variance relation of imaging sensors. Conventional methods [1], [6]–[13] estimate points of such mean-variance curve isolating and separately processing segments or patches of the signal with common mean and noise variance, so that on each segment or patch simple sample estimators of mean and variance can be applied. In this way, a scatterplot in the mean-variance plane is produced. Then, a curve is fitted to the scatterplot, yielding an estimate of the relation for the whole range of the signal.

In this paper we show that, contrary to common belief, the estimation can be accurate even if each scatterplot point is estimated from a heterogeneous sample (e.g., a patch whose pixels can have very different mean values). We justify this result through a mathematical modeling based on mixtures of normal distributions. Thus, unlike conventional signal-dependent noise estimation techniques that preprocess the image in order to work with homogeneous samples, our approach applies robust estimators to arbitrarily large patches of heterogeneous data extracted at random from the image.

Our analysis is focused on the camera noise models such as the affine-variance model depicted in Figure 1. For the sake of

Contact email: first.name.last.name@tut.fi.

∗ This work was supported by the Academy of Finland (project no. 252547). Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

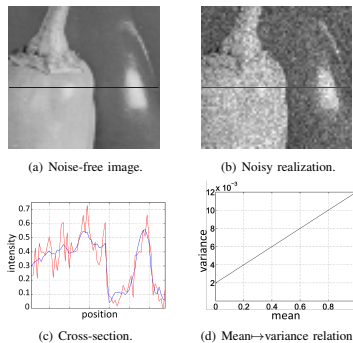


Figure 1. Detail of the "Peppers" image corrupted by signal-dependent noise with affine variance (2), with parameters $a = 0.01$ and $b = 0.002$.

clarity and due to length limitation, we restrict the presentation to the 2-D image case; nevertheless, the introduced concepts and the proposed approach apply universally to 1-D signals as well as to multidimensional data.

The paper is organized as follows. In Section II we introduce the considered signal-dependent noise model and we describe the conventional approach for its estimation. Next, we present our novel noise estimation technique and a prototype algorithm that exploits it, discussing its difference w.r.t. conventional methods. In Section III we study the main factors contributing to estimation errors, through a theoretical analysis and a Monte Carlo simulation. In Section IV we show the effectiveness of the method in real applications by estimating noise affecting raw data from a CCD camera, and a comparison with a state-of-the-art algorithm. Finally, in Section V and Section VI we provide discussions and conclusions.

II. METHOD

A. Problem statement

Let us consider a noisy observation z of a deterministic noise-free signal y , corrupted by additive spatially uncorrelated noise with signal-dependent variance:

$$z(\mathbf{x}) = y(\mathbf{x}) + \sigma(y(\mathbf{x}))\xi(\mathbf{x}), \quad (1)$$

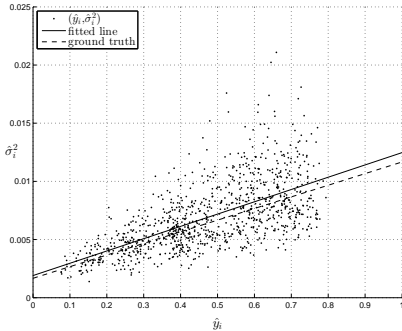


Figure 2. Scatterplot of the mean-variance pairs (y_i, σ_i^2) , fitted line $\hat{\sigma}(y) = \hat{a}y + \hat{b}$, and ground truth line $\sigma(y) = ay + b$ from the "Peppers" image, corrupted by noise with parameters $a = 0.01$ and $b = 0.0017$. We use 1000 blocks of size 16×16 , each yielding a point in the scatterplot.

where $\sigma : \mathbb{R} \rightarrow \mathbb{R}^+$ is a function giving the signal-dependent standard deviation of the noise, $\mathbf{x} \in \mathbf{X} \subset \mathbb{Z}^2$ is the pixel coordinate, and $\xi : \mathbf{X} \rightarrow \mathbb{R}$ is a zero-mean independent random noise with standard deviation equal to 1. Our goal is to estimate the function σ .

The expectation of $z(\mathbf{x})$, denoted as $E\{z(\mathbf{x})\}$, is the noise-free signal $y(\mathbf{x})$; at the same time, the variance $\text{var}\{z(\mathbf{x})\}$ and the standard deviation $\text{std}\{z(\mathbf{x})\}$ of $z(\mathbf{x})$ are, respectively, $\sigma^2(y(\mathbf{x}))$ and $\sigma(y(\mathbf{x}))$, because $\text{var}\{y(\mathbf{x})\} = 0$.

As discussed in [2], the term $\xi(\mathbf{x})$ can generally have a different probability distribution for each different coordinate \mathbf{x} , i.e. $\xi(\mathbf{x}_1) \approx \xi(\mathbf{x}_2)$ if $\mathbf{x}_1 \neq \mathbf{x}_2$; in order to simplify the mathematical model, we approximate $\xi(\mathbf{x})$ as a normal distribution $\mathcal{N}(0, 1)$. In this way the noise can be considered heteroskedastic Gaussian, with zero mean and signal-dependent variance $\sigma^2(y(\mathbf{x}))$, i.e. $\sigma(y(\mathbf{x}))\xi(\mathbf{x}) \sim \mathcal{N}(0, \sigma^2(y(\mathbf{x})))$.

To provide practical experimental results of our method, we shall refer to the affine noise variance model [5], which is one of the most suitable for modeling the noise in digital image sensors. According to this model, the noise variance is approximated as

$$\sigma^2(y(\mathbf{x})) = ay(\mathbf{x}) + b, \quad (2)$$

where $ay(\mathbf{x})$ and b are, respectively, the variances of the signal-dependent and signal-independent parts of the noise. The former part is due to a photon-counting process (Poisson distribution), while the latter is caused by a combination of dark noise (Poisson distribution) and thermal-electronic noise (normal distribution). Because of a central-limit theorem argument and because of the good approximation of the Poisson by a Gaussian, the normal approximation of $\xi(\mathbf{x})$ is valid. For (2), the problem of estimating σ^2 can be reduced to the estimation of the two constants a and b .

B. Conventional approach

The conventional approach for the estimation of signal-dependent noise is to segment the image into regions where pixels have constant intensity, and hence, because of (1), constant noise; then, the mean and noise variance are estimated for each region independently. In this way it is possible to create a scatterplot that relates the noise-free intensity values of y (abscissa) with the respective noise variances (ordinate), that, finally, is used to approximate the function $\sigma(y)$ in (1) (or equivalently the function $\sigma^2(y)$).

There are different methods for partitioning the image, with different complexity and accuracy. The partition can be made, e.g., by simply using pixels extracted from a sufficiently small window from the noisy image [9], with the constraint that the intensity does not change much within the window [7], [8], or by segmenting the image into level sets (bins) with individual intensity values [1], [6], [10], [11], [13]. More sophisticated techniques, such as DCT-based estimators [12], have been also proposed. However, the backbone idea is still to exploit homogeneous samples for the actual noise estimation.

The rationale of these techniques is that, being the segments homogeneous, also the noise variance is homogeneous, as can be trivially concluded from (1). Hence, standard estimators of the sample mean and sample variance can be directly applied to the segments, yielding unbiased estimates of the mean and noise variance. In other words, the resulting scatterplot points are distributed about the noise variance curve $\sigma^2(y)$.

C. Main idea

In contrast with the common procedure based on relatively small homogenous segments, we show that the estimation of each scatterplot point can be performed processing large heterogeneous samples. As we shall demonstrate, considering a heterogeneous group of elements taken from z , the expectations of the estimators of its mean and noise variance are still a coordinate of a point that belongs to the function $\sigma^2(y)$. Consequently, it is not necessary to partition the image into segments of constant intensity levels and noise variances, but it is possible to process together parts of the image corrupted by noise with various variance values, without compromising the estimation. In particular, adaptive segmentation is no longer required in order to estimate signal-dependent noise, but its only advantage consists in limiting the positive bias due to outliers that could occur when estimating the variance. In this way we can avoid the segmentation step and, consequently, simplify the entire process.

We define our approach *indirect* because the pair estimated from one block does not represent directly a single relation mean-noise variance, like for the conventional methods, but it represents the mean and the variance of an heterogeneous group of elements, i.e. a mixture of distributions.

An example of the scatterplot computed from the blocks taken at random positions from the whole noisy image in Figure 1(a) is shown in Figure 2 (black dots), with its estimation of $\sigma^2(y)$ and the ground truth.

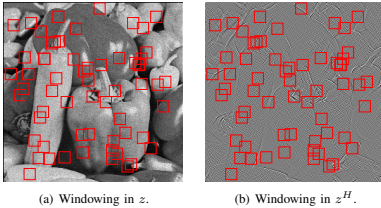


Figure 3. Example of 16×16 windows at random position in z and at corresponding positions in z^H .

D. Prototype algorithm

The simplest algorithm that can leverage the above idea can be divided in three basic steps:

- (a) *High-Pass Filtering*: most of the energy of the noise-free signal y is usually confined to the lower frequencies of z , thus, applying an high-pass filter to z permits to extract the zero mean noise from it [3]. We obtain the high-frequency part of z , referred to as z^H , by convolving z against a 2-D high-pass function ψ (e.g., a wavelet):

$$z^H = z \circledast \psi, \quad (3)$$

where ψ has zero mean, i.e. $\sum_i \psi(i) = 0$, and ℓ^2 -norm equal to one, i.e. $\sum_i \psi^2(i) = 1$.

- (b) *Local Estimation*: once the detail image z^H is computed, we randomly choose N coordinates within the image z , like in Figure 3; then, from these locations, N square blocks W_i^z , $i = 1, \dots, N$, of size $\sqrt{n} \times \sqrt{n}$ are extracted from z . Similarly, N blocks W_i^H , $i = 1, \dots, N$, of the same size and from the same positions of W_i^z , are extracted from z^H . We estimate the means y_i from the blocks W_i^z , while from W_i^H we estimate the corresponding noise variances σ_i^2 . In this way, for each block W_i^z , we obtain a pair $(\hat{y}_i, \hat{\sigma}_i^2)$ which can be represented by a point in the scatterplot. The pairs $(\hat{y}_i, \hat{\sigma}_i^2)$ are, therefore, the estimates of the blocks means and noise variances (y_i, σ_i^2) .

Because the blocks are taken from random positions within the image, each block may contain pixels having various expected intensity levels. Therefore, the distribution of noise in a single block W_i^z can be considered as a mixture of normal distributions with different variances. This marks a principal difference with the conventional methods that look for uniform blocks (or regions) for the estimation, and that model the noise within a single block as realization of a single normal distribution with given mean and variance.

In the next section we investigate the effects of exploiting elements taken from a mixture instead of from a single normal distribution.

- (c) *Fitting*: in order to estimate the parameters that describe the curve $\sigma^2(y)$, we fit the pairs $(\hat{y}_i, \hat{\sigma}_i^2)$, $i = 1, \dots, N$, using a least squares (LS) method, which is the simplest

fitting technique at our disposal.

III. ESTIMATION ERROR

A. Noise analysis

Let us model image blocks as composed by R_i regions (piecewise modeling), with $R_i \leq n$, and let W_i^y denote the noise-free block corresponding to W_i^z .

We shall refer as *ideal* the case in which, in W_i^z , the amount of energy due to y is negligible with respect to the noise energy. For example, this is the case when W_i^y can be treated as piecewise constant with edges having small excursions with respect to the noise standard deviation, or, equivalently, when the high-pass filter perfectly extract the noise component from z . In this case, the elements of W_i^z and W_i^H are, respectively, realization of two mixtures of R_i normal distributions with probability density functions (*p.d.f.*'s):

$$f_i^z(x) = \sum_{k=1}^{R_i} \lambda_k^{(i)} p_k^z(x), \quad p_k^z \sim \mathcal{N}(m_k, s_k^2), \quad (4)$$

$$f_i^H(x) = \sum_{k=1}^{R_i} \lambda_k^{(i)} p_k^H(x), \quad p_k^H \sim \mathcal{N}(0, s_k^2), \quad (5)$$

where p_k^z and p_k^H are, respectively, the *p.d.f.*'s of the k -th normal distributions of f_i^z and f_i^H , $\lambda_k^{(i)}$ is the proportion of the elements of the k -th population respect to the total number of elements n , m_k is the mean of the k -th normal function in f_i^z , i.e. the k -th intensity value in W_i^y , and s_k^2 is the variance of both p_k^z and p_k^H . It is important to notice that the ideality of this case relies mainly on the fact that the variances of the k -th distributions are equal.

Trivially we have

$$y_i = \sum_{k=1}^{R_i} \lambda_k^{(i)} m_k. \quad (6)$$

Exploiting the moments of a general mixture of normal distributions¹, and the fact that all the p_k^H have zero mean, we obtain

$$\sigma_i^2 = \sum_{k=1}^{R_i} \lambda_k^{(i)} s_k^2. \quad (7)$$

Considering now the particular Poisson-Gaussian noise, it follows that the elements of W_i^H can be individually modeled as realizations of independent normal random variables with variances defined by the affine transformation (2) of W_i^y :

$$s_k^2 = am_k + b.$$

¹The expectation m and the variance s^2 of a mixture of G normal distributions are

$$m = \sum_{k=1}^G \nu_k m_k, \\ s^2 = \sum_{k=1}^G \nu_k [(m_k - m)^2 + s_k^2],$$

where m_k , s_k^2 and ν_k are, respectively, the expectation, the variance and the proportion of the k -th normal distribution [4].

Consequently, noting that $\sum_{k=1}^{R_i} \lambda_k^{(i)} = 1$,

$$\begin{aligned} \sigma_i^2 &= \sum_{k=1}^{R_i} \lambda_k^{(i)} a m_k + \sum_{k=1}^{R_i} \lambda_k^{(i)} b = \\ &a \sum_{k=1}^{R_i} \lambda_k^{(i)} m_k + b = a y_i + b. \end{aligned} \quad (8)$$

This means that the point (y_i, σ_i^2) belongs to the line (2). Therefore, if \hat{y}_i and $\hat{\sigma}_i^2$ are computed, respectively, with unbiased estimators of the population mean and variance of a mixture of normal distributions, the points $(\hat{y}_i, \hat{\sigma}_i^2)$ will yield a cloud scattered about the line (2), and the only error occurring in the computation of the pair $(\hat{y}_i, \hat{\sigma}_i^2)$ is the one due to the variances of the estimators.

The above proof shows that, in ideal conditions, the presented algorithm ensures correct estimation even using blocks affected by different noise levels.

Let us now consider a more *practical* scenario where the presence of the noise-free signal W_i^H is still appreciable, influenced by strong edges and texture in W_i^y . In this case, the noise distribution in W_i^H can no longer be approximated as a mixture of zero-mean normal distributions. In practice, this means that W_i^H does not contain only noise, and that, among its detail coefficients, there could be elements that introduce a bias in the estimation of σ_i^2 . Consequently, the estimation error does not depend only on the variance of the estimator, but it is also influenced by the presence of edges in W_i^y .

To reduce the effect of these outliers, we use the median of absolute deviation (MAD) [15], [16] as robust estimator of σ_i^2 and, for coherence, the median (med) as estimator of the mean:

$$\hat{y}_i = \text{med} \{W_i^y\}, \quad (9)$$

$$\hat{\sigma}_i^2 = \left[\frac{\text{MAD} \{W_i^y\}}{\Phi^{-1}(\frac{3}{4})} \right]^2. \quad (10)$$

Here, $\text{MAD} \{W_i^y\} = \text{med} \{|W_i^y - \text{med} \{W_i^y\}|\}$, and Φ^{-1} denotes the inverse cumulative distribution function (*c.d.f.*) of the standard normal distribution, and the constant factor $1/\Phi^{-1}(\frac{3}{4}) = 1.4826$ makes the estimator asymptotically unbiased in case of *i.i.d.* normal samples.

When using MAD, it is important to consider that the relation (8) may fail, because of the potential discrepancy between the mean and the median of distributions that are not *i.i.d.* normal. Nevertheless, the use of the MAD estimator on W_i^H can be justified because of the *Gaussianization* of the coefficients resulting by a transformation of the type (3) [2]. We support this thesis providing, in the next section, an accurate study of the robust estimators errors in practical applications.

B. Error analysis

As described in the previous section, the estimation error is composed by two parts: one due to the variance of the estimators (the only one in the ideal case), and one due to the presence of outliers (e.g., edges). In this section we analyze quantitatively how these outliers affect the computation of the pairs $(\hat{y}_i, \hat{\sigma}_i^2)$.

For this purpose we performed a Monte Carlo simulation

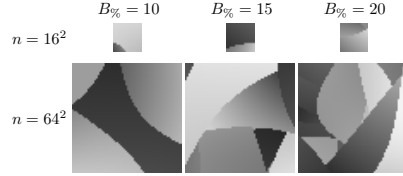


Figure 4. Examples of the patches W_i^y used in the Monte Carlo simulation, with different block sizes n and percentages of boundaries $B\%$.

where we compute the average estimation error on a pair (y_i, σ_i^2) from a block containing a certain amount of edges:

- for each task, a patch containing a random number of regions and corrupted by affine signal-dependent noise is created;
- the patches are then grouped depending on the amount of edges within them;
- the mean-noise variance pairs are then estimated;
- the estimation errors are computed for each block;
- finally, the errors are averaged, separately, for each group.

In this way, we compute the average estimation error in function of the amount of edges in the block.

We now describe more accurately the entire process.

1) *Patch generation and grouping*: we generate patches W_i^y containing a random number of regions; each region of each patch is piecewise smooth with piecewise smooth boundaries (examples are shown in Figure 4). The minimum and maximum intensity values of each region are realization of random variables uniformly distributed in $[0, 1]$. The patches are then grouped depending on the percentage of edges $B\%$ within them. Every patch is corrupted by the noise defined in (2), and filtered as described in (3)². In this way we create W_i^z and W_i^H , which are used for computing \hat{y}_i and $\hat{\sigma}_i^2$, respectively. The noise parameters a and b are chosen, for each patch, as realization of random variables uniformly distributed respectively in $[0, 0.002]$ and $[0, 0.0006]$, in order to operate on noise ranges comparable to those considered in, e.g., [1], [2], which are representative of typical consumer camera sensors.

2) *Error computation and normalization*: for every patch, the estimation error e_i is computed as the distance between the point $(\hat{y}_i, \hat{\sigma}_i^2)$ estimated with (9) and (10), and the ground-truth line $ay + b$, i.e. the distance between $(\hat{y}_i, \hat{\sigma}_i^2)$ and its orthogonal projection $(\hat{y}_{i\perp}, \hat{\sigma}_{i\perp}^2)$ on the line $ay + b$.

Intuitively, the estimation errors of the mean and variance are function of the noise variance that we are estimating, i.e. larger noise variance implies larger estimation error. Consequently, estimation errors on patches having the same amount of edges, but affected by different noise levels, can be significantly different. We normalize the square estimation error e_i^2 by dividing it by the mean square error (MSE) $\bar{e}_{i\perp}^2(\hat{\sigma}_{i\perp}^2)$ that we would have had if we were performing the estimation

²To eliminate the boundary artifacts in the computation of W_i^H , we create a bigger patch (padding) in order to discard the boundaries once the filtering is performed.

on a flat patch containing only one region, and affected by constant noise variance σ_{\perp}^2 . In this way, the normalized error becomes an index of the goodness of the estimation with respect to the simplest possible case, i.e. a single flat region. Let us now show how the MSE $\bar{e}^2(\cdot)$ depends on the noise variance $\bar{\sigma}^2$ of a generic flat patch \bar{W}^z , denoting \bar{W}^{H} its filtered version:

$$\text{MSE} \{ \text{med} \{ \bar{W}^z \} \} = \text{var} \{ \text{med} \{ \bar{W}^z \} \} = v^z(\bar{\sigma}^2) = \frac{\pi}{2n} \bar{\sigma}^2, \quad (11)$$

$$\text{MSE} \{ \text{MAD} \{ \bar{W}^{H} \} \} = \text{var} \{ \text{MAD} \{ \bar{W}^{H} \} \} = v^H(\bar{\sigma}^2) = \frac{\alpha}{n} \bar{\sigma}^4, \quad (12)$$

$$\bar{e}^2(\bar{\sigma}^2) = v^z(\bar{\sigma}^2) \sin^2(\theta) + v^H(\bar{\sigma}^2) \cos^2(\theta), \quad (13)$$

where $v^z(\bar{\sigma}^2)$ and $v^H(\bar{\sigma}^2)$ are, respectively, the variances of the median and MAD estimators applied to the patches \bar{W}^z and \bar{W}^{H} , and α is a constant that depends on the function³ ψ that we use to filter \bar{W}^z in order to obtain \bar{W}^{H} . The MSEs of the estimators coincide with their variances because the patches are flat and the estimation errors have zero mean, i.e. the samples are unbiased because there are no outliers.

In (13), the terms $\sin^2(\theta)$ and $\cos^2(\theta)$ are used to compute the orthogonal components of (11) and (12) to the line $ay+b$, the only components of the variances that mislead the estimation, with θ being the angle between the line $ay+b$ and the horizontal axes, i.e. $\theta = \arctan(a)$.

We can finally define the normalized square estimation error \bar{e}_i^2 as

$$\bar{e}_i^2 = \frac{e_i^2}{\bar{e}^2(\bar{\sigma}_{\perp}^2)}. \quad (14)$$

3) Averaging and error trend: Figure 5 shows the root mean square error (RMSE) and the root mean normalized square error (RMNSE) resulting from respectively averaging the estimation errors e_i^2 and \bar{e}_i^2 over groups of patches having the same percentage of edges $B_{\%}$. We separately consider four different window sizes n .

The RMNSE curves in Figure 5(b) are approximately monotonically increasing with common minimum 1 at $B_{\%} = 0$, where patches are composed of a single region and have no internal edges. Note that the patches W_i^z are piecewise smooth, and not perfectly flat as in the ideal case; nevertheless at $B_{\%} = 0$ the RMNSE is practically 1. This means that, when $B_{\%} = 0$, the RMSE essentially coincides with the standard deviation of the estimator and, when $B_{\%} > 0$, the estimation errors are almost entirely due to the presence of edges.

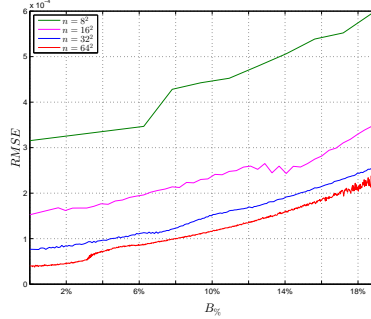
IV. EXPERIMENTS ON CAMERA RAW IMAGES

To validate the proposed algorithm in a practical context, we apply it to raw images from a digital camera. The images are shown in the left and center columns of Figure 6 and were taken using a *Canon PowerShot S90* 10-Megapixel camera. We

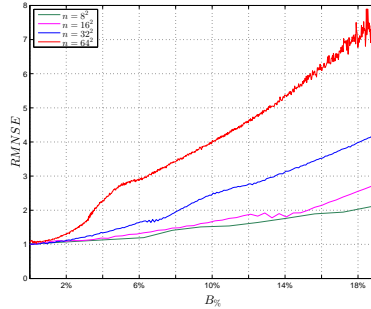
³In our experiments ψ is generated by separable convolution of one 1-D Daubechies wavelet kernel,

$$\psi = \psi_{1D} \otimes \psi_{1D}^T,$$

where $\psi_{1D} = [-0.333, 0.807, -0.460, -0.135, 0.085, 0.035]$. For this ψ , we empirically computed $\alpha = 9.9076$.



(a) Root mean square error (RMSE).



(b) Root mean normalized square error (RMNSE).

Figure 5. RMSE and RMNSE as function of the percentage of edges $B_{\%}$ within each block, for block size $n = 8^2, 16^2, 32^2, 64^2$. The estimations have been performed using the robust estimators in (9) and (10).

adjusted the exposure times in order to avoid clipping (e.g., overexposure). The pictures were acquired with various ISOs and exposure times, so to have realizations of different noise levels [14].

In the rightmost column of Figure 6, the lines estimated by the proposed prototype algorithm (continuous lines) are compared against those estimated by a state-of-the-art algorithm [1] (dashed lines), here used as reference method. This algorithm first preprocesses the image in order to detect and exclude edges and texture from the noise estimation; it then partitions the remaining image into segments of constant intensity level; a scatterplot is thus obtained by applying a robust unbiased estimator of the variance on each segment, with each point of the scatterplot being modeled according to a bivariate normal distribution; the noise model parameters a and b are finally estimated through a maximum *a posteriori* fitting. For these experiments, our prototype algorithms uses

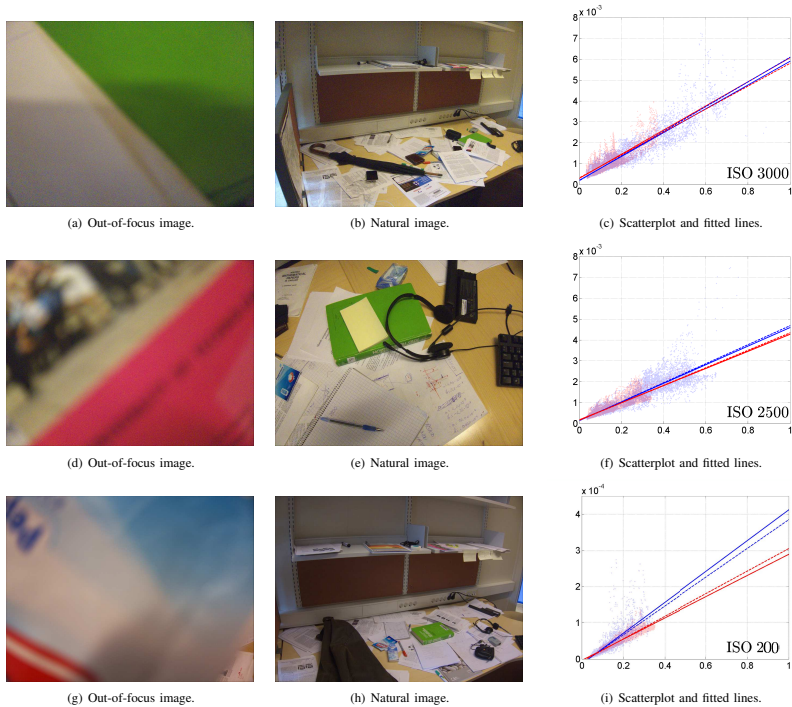


Figure 6. Scatterplots and estimated functions for out-of-focus (red clouds and red continuous lines) and complex natural (blue clouds and blue continuous lines) images. The images have been taken with a *Canon PowerShot S90*, ISO 3200 (first row), ISO 2500 (second row), and ISO 200 (third row) using exposure times respectively equal to $1/1000$, $1/600$, and $1/125$. The estimation is performed using 2000 patches for each channel ($\{R, B; G_1, G_2\}$) of size 64×64 . The dashed lines show the functions estimated by the ref. [1].

blocks of size 64×64 , and, in order to reduce the variability of the results on the particular random choice of the block positions, 2000 patches are extracted from each color channel of the images.

In Section III-A we discussed the theoretical behavior of our method in the ideal conditions where the extracted patches are free of edges ($B_{\%} = 0$ in Section III-B). In order to reproduce these assumptions, the raw images include 3 out-of-focus (OoF) pictures, shown in the leftmost column of Figure 6. The lines estimated by the two algorithms (red continuous and dashed lines) are always close to each other, confirming that, in the ideal case, the proposed algorithm gives results congruent to those of the reference algorithm.

The 3 pictures of a complex natural scene, shown in the center column of Figure 6 are used to investigate the practical case. The lines estimated with the proposed algorithm (blue

continuous lines) are again close to the reference ones (blue dashed lines), confirming that the proposed algorithm performs similar to the reference algorithm also on complex images.

In Figure 6, the OoF and natural pictures that are on the same row were acquired under the same operating conditions (ISO, exposure time, ambient temperature) and are hence corrupted by noise with the same parameters [14]. Therefore, the blue solid and dashed lines in each subplot may be expected to coincide with the respective red lines. Indeed, for large ISO (top and middle rows of Figure 6), the lines estimated from OoF and natural images are very close to each other, because the large noise variance makes easier for the algorithms to separate the noise from the noise-free signal. In case of small ISO (bottom row), instead, the estimation from the natural image diverges from the OoF ones, for both proposed and reference algorithms, since the variance of the

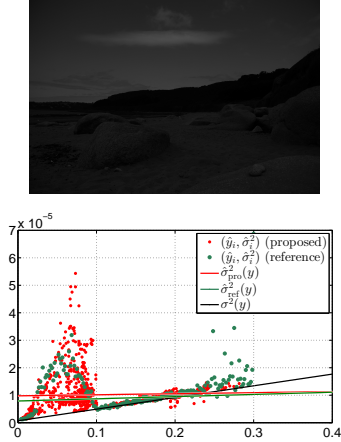


Figure 7. Example in which both proposed and reference algorithm fail the estimation due to the presence of several outliers. Top: image with large highly textured areas from the NED dataset [20]. Bottom: scatterplot of mean-variance pairs with corresponding noise line $\hat{\sigma}_{priv}^2(y)$ estimated by the proposed prototype algorithm (red). The result is compared with the line $\hat{\sigma}_{ref}^2(y)$ estimated using the reference algorithm (green) and the ground-truth $\sigma^2(y)$ (black). Due to the overwhelming presence of outliers in the scatterplot, both the proposed and the reference algorithm fail to correctly estimate the noise line.

noise is small with respect to the signal. The degradation of accuracy of the proposed algorithm is comparable to that of the reference one.

In Figure 7 we report the result $\hat{\sigma}_{priv}^2(y)$ of the proposed prototype algorithm applied to an image that contains large highly textured areas. The image belongs to the NED dataset [20] of raw images with large areas of high-frequency texture, which makes noise estimation particularly challenging. The image has been captured with a *Nikon D80* at ISO 125, and the response of the sensor has been linearized by a calibrated nonlinear correction function. In the same scatterplot we also present the mean-variance pairs and the line $\hat{\sigma}_{ref}^2(y)$ estimated with algorithm [1], and the ground-truth line $\sigma^2(y)$ too. Both scatterplots reveal the presence of several outliers in the intensity range $y \in [0, 0.1]$, mostly generated by textures present on the mountains. These outliers cause the misestimation of the lines fitted by either the proposed and the reference algorithm⁴. This result confirms that textures and edges are the main cause of misestimation, since they affect similarly proposed and reference algorithm, and that the scatterplot points can be estimated using heterogeneous samples.

To evaluate the impact of the block size, we repeat the

⁴A robust variant of [1] was recently published [19] while the present article was already in press. The variant models the scatterplot points as an adaptive mixture of Gaussian and Cauchy distributions, and thus yields more accurate results in cases with outliers such as that illustrated in Figure 7.

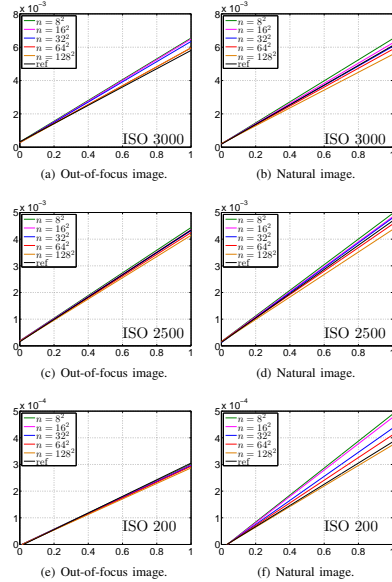


Figure 8. Lines estimated from the images in Figure 6 separately using 2000 patches for each channel (R, B; G1, G2) of size $n = 8^2, 16^2, 32^2, 64^2, 128^2$. The results are compared with the estimates of the reference algorithm.

experiment presented in Figure 6 separately using patches of size $n = 8^2, 16^2, 32^2, 64^2, 128^2$; the results are reported in Figure 8. The lines estimated from the OoF images, showed on the left column of the figure, are relatively close to each other independent of the block size. Observing the results from natural images, showed on the right column of the figure, we can notice, especially at low ISO, that estimates from larger blocks are less affected by overestimation bias.

Finally, to illustrate the essential role of the robust estimators in alleviating the bias effect due to outliers from edges and texture, we performed the same Monte Carlo simulation described in Section III-B, using the sample mean and sample variance to estimate the scatterplot points. In Figure 9 we show the RMNSE of these non-robust estimators in comparison to that of the med-MAD (9)-(10). Note how the robust estimators, and in particular the MAD, drastically reduce the error.

V. DISCUSSION

In Section III-B, as well as in Figure 8, we demonstrated that the average estimation error of the points (y_i, σ_i^2) may get smaller if larger blocks are used in conjunction with

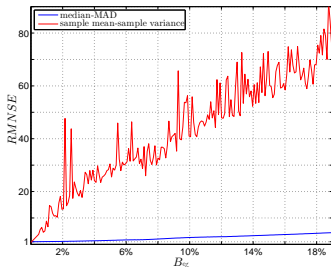


Figure 9. Root mean normalized square error (RMNSE) of the pairs *median-MAD* and *sample mean-sample variance* for blocks of size $n = 32^2$. Robust estimators lead to such a reduction of error also for the other block sizes.

robust estimators, in spite of the fact that the samples get more heterogeneous. However, there is also an inevitable trade-off in the choice of the block size: when using large patches it is unlikely that the mean (or median) y_i reaches the extremes of the distribution of the image intensity values y . As a consequence, the scatterplot may cluster about the point $(c, ac + b)$, c being the mean (or median) of y over the whole image, and, thus, the accuracy of the estimated line may be degraded. On the other hand, smaller patches allow the scatterplot points to distribute on a wider interval, at the expense of higher estimation variance for each point, and risk of larger bias on some of them. While the variance errors may cancel out through the curve fitting, the bias errors will eventually corrupt the final estimate unless a robust line fitting is utilized.

In Figure 9, the average error for robust and standard estimators are compared, demonstrating the complete failure caused by non robust.

Our analysis and algorithm are developed and validated on the specific affine-variance model (2), and may fail for a generic non-affine $\sigma^2(y)$. On the other hand, if σ^2 is well approximated by a locally (i.e. separately on each block) affine function of y , we can still use the proposed algorithm, ensuring accurate results. However, in many cases (e.g., in the case of clipping) it can be difficult to verify the local affinity of σ^2 without any strong assumptions on the image y .

Let us discuss also about ways how to possibly improve the estimation accuracy. In its prototype implementation, our algorithm is limited by the accuracy of the MAD estimator and thus cannot reach the accuracy of algorithms (e.g., [17]) that adopt more sophisticated estimators for the estimation of the variance. Likewise, the simplest LS fitting method is not robust to outliers in the scatterplot. Therefore, the use of a better variance estimator and a better (e.g., robust) fitting algorithm [19] could further improve the estimation, so to possibly deal with highly textured images such as the example in Figure 7.

Adaptive procedures such as segmentation may be crucial for alleviating the impact of high-frequency texture on the variance estimation, but we especially emphasize that this is

not a peculiarity of signal-dependent noise models, and it applies also to constant-variance (homoskedastic) noise models, including additive white Gaussian noise (AWGN). In fact, the advanced methods [17] and [18] are developed for AWGN estimation. As shown in our theoretical and experimental analysis, the fact that the variance of the noise is not constant (heteroskedasticity), and depends instead on the signal, does not *per se* imply an additional need for adaptive segmentation.

Finally, let us note that the proposed model deals with the estimation of signal-dependent noise that is spatially uncorrelated, i.e. noise with diagonal covariance matrix. It is nevertheless possible to extend the proposed approach also to the correlated-noise case. If the correlation model (i.e. the shape of the noise power spectral density (PSD)) is known, one can compute the noise energy in the high-pass image z^H from which the blocks W_i^H are extracted, and hence normalize the output of the variance estimator based on the product of the PSD with the spectrum of ψ . This product can be preconditioned by suitably downsampling the data prior to analyzing the noise; downsampling may be also desirable, as a means to reduce the amount of data to be processed.

VI. CONCLUSIONS

As opposed to conventional methods that require homogeneous samples for the estimation of mean-variance pairs, our approach to signal-dependent noise estimation utilizes arbitrarily large samples of possibly heterogeneous data. The approach is backed by a Gaussian-mixture modeling, which shows that the individual mean-variance estimates computed from the heterogeneous samples are still representative of the true mean-variance curve. An elementary prototype algorithm based on this modeling is presented for the estimation of signal-dependent noise from a single image. The algorithm extracts large heterogeneous samples from random locations in the image. This corresponds to a fundamental difference versus traditional algorithms, which often involve an adaptive segmentation of the image into narrow homogeneous segments, and it also results in a simplification of the estimation procedure. This approach can be therefore suitable in all applications where a simple noise estimation algorithm is required, and which has to operate on non-intelligent devices. Experiments on real data demonstrate the reliability of the algorithm applied to natural images, showing that its results are comparable with those from a state-of-the-art method.

REFERENCES

- [1] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, *Practical Poissonian-Gaussian noise modeling and fitting for single image raw data*, IEEE Trans. Image Process., vol. 17, no. 10, pp. 1737-1754, October 2008. doi:10.1109/TIP.2008.2001399.
- [2] A. Foi, *Clipped noisy images: heteroskedastic modeling and practical denoising*, Signal Processing, vol. 89, no. 12, pp. 2609-2629, December 2009. doi:10.1016/j.sigpro.2009.04.035.
- [3] D.L. Donoho and I.M. Johnstone, *Ideal spatial adaptation via wavelet shrinkage*, Biometrika (1994) 81(3): 425-455. doi:10.1093/biomet/81.3.425.
- [4] N. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, vol. 1, Wiley & Sons, New York, Second edition, 1994, Section 13.10.

- [5] G.K. Froehlich, J.F. Walkup, and R.B. Asher, *Optimal estimation in signal-dependent noise*, JOSA, Vol. 68, Issue 12, pp. 1665-1672, 1978. doi: 10.1364/JOSA.68.001665.
- [6] C. Liu, W. T. Freeman, R. Szeliski and S. B. Kang, *Noise Estimation from a Single Image*, Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '06, vol. 1, pp. 901-908, 17-22 June 2006. doi: 10.1109/CVPR.2006.207.
- [7] A. Amer and E. Dubois, *Fast and reliable structure-oriented video noise estimation*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 1, pp. 113-118, January 2005. doi: 10.1109/TCSVT.2004.837017(410) 1.
- [8] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, M. Martín-Fernández and C. Alberola-López, *Automatic noise estimation in images using local statistics. Additive and multiplicative cases*, Image and Vision Computing, vol. 27, Issue 6, pp. 756-770, ISSN 0262-8856, 4 May 2009. doi: 10.1016/j.imavis.2008.08.002.
- [9] J.S. Lee and K. Hoppel, *Noise Modeling and Estimation of Remotely-Sensed Images*, Geoscience and Remote Sensing Symposium, 1989. IGARSS'89. International 12th Canadian Symposium on Remote Sensing, vol.2, no., pp.1005-1008, 10-14 July 1989. doi: 10.1109/IGARSS.1989.579061.
- [10] P. Gravel, G. Beaudoin and J.A. De Guise, *A method for modeling noise in medical images*, Medical Imaging, on IEEE Transactions, vol.23, no.10, pp.1221-1232, October 2004. doi: 10.1109/TMI.2004.832656.
- [11] B. Aiazzi, L. Alparone, S. Baronti, M. Selva and L. Stefani, *Unsupervised estimation of signal-dependent CCD camera noise*, Springer International Publishing AG, EURASIP Journal on Advances in Signal Processing, No.1, pp. 1-11, 2012. doi: 10.1186/1687-6180-2012-231.
- [12] M. Uss, B. Vozel, V. Lukin, S. Abramov, I. Baryshev, K. Chehdi, *Image Informative Maps for Estimating Noise Standard Deviation and Texture Parameters*, EURASIP Journal on Advances in Signal Processing, No. 1, vol. 2011, p. 806516, 2011. doi: 10.1155/2011/806516.
- [13] T. Buades, Y. Lou, J.M. Morel, Zhongwei Tang, *A note on multi-image denoising*, International Workshop on Local and Non-Local Approximation in Image Processing, 2009. LNLA 2009, pp. 1-15, 2009. doi: 10.1109/LNLA.2009.5278408.
- [14] P. Ojala, *Dependence of the parameters of digital image noise model on ISO number, temperature and shutter time*, prepared for the 2008 TUT/Nokia Mobile Imaging course. http://www.cs.tut.fi/~foi/MobileImagingReport_PetteriOjala_Dec2008.pdf
- [15] F. R. Hampel, *The influence curve and its role in robust estimation*, Journal of the American Statistical Association, vol. 69 (346), pp. 383-393, 1974.
- [16] F. Mosteller and J.W. Tukey, *Data Analysis and Regression: A Second Course in Statistics*, Addison Wesley, 1997.
- [17] A. Danielyan and A. Foi, *Noise variance estimation in nonlocal transform domain*, Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2009, Tuusula, Finland, pp. 41-45, August 2009. doi:10.1109/LNLA.2009.5278404
- [18] N.N. Ponomarenko, V.V. Lukin, M.S. Zriakhov, A. Kaarna and J. Astola, *An automatic approach to lossy compression of AVIRIS images*, Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International. pp. 472-475, 2007. doi: 10.1109/IGARSS.2007.4422833.
- [19] L. Azzari and A. Foi, *Gaussian-Cauchy Mixture Modeling for Robust Signal-Dependent Noise Estimation*, in Proc. IEEE ICASSP2014, pp. 5394-5398, May 2014.
- [20] Image database for benchmarking signal-dependent noise estimation algorithms: NED2012, Online: <http://rsd.khai.edu/med2012/med2012.php>. Accessed date: December 2013.

Publication III

Azzari, L. and Foi, A. (2015). Collaborative filtering based on group coordinates for smoothing and directional sharpening. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1573-1577.

© 2015 Institute of Electrical and Electronics Engineers (IEEE). Reprinted, with permission, from IEEE International Conference on Acoustics, Speech and Signal Processing.

COLLABORATIVE FILTERING BASED ON GROUP COORDINATES FOR SMOOTHING AND DIRECTIONAL SHARPENING

Lucio Azzari and Alessandro Foi

Department of Signal Processing, Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland

ABSTRACT

Groups of mutually similar image blocks are the key element in nonlocal image processing. In this work, the spatial coordinates of grouped blocks are leveraged in two distinct parts of the transform-domain collaborative filtering within the BM3D algorithm. First, we introduce an adaptive 1-D transform for 3-D collaborative filtering based on sampling 2-D smooth functions at the positions of grouped blocks. This adaptive transform is applied for improved decorrelation of the 2-D spectra of the grouped blocks. Second, we propose a directional sharpening procedure whose strength varies adaptively according to the relative orientation of the transform basis functions with respect to the group coordinates. Experiments confirm the efficacy of the proposed adaptations, for denoising as well as for sharpening of noisy images.

Index Terms— BM3D, adaptive transforms, collaborative filtering, denoising, sharpening.

1. INTRODUCTION

BM3D [1] is one of the most effective nonlocal image denoising algorithm. Its state-of-the-art performance is based on the so-called grouping and collaborative filtering approach, which consists in 1) stacking mutually similar image blocks into 3-D *groups*, 2) a 3-D transformation of the groups into typically very sparse group spectra; 3) shrinkage of these spectra; 4) inversion of the 3-D transform and 5) aggregation of the resulting block estimates at their original local within the image.

A peculiarity of this procedure is that steps 2), 3), and 4) are carried out irrespective of the positions that the grouped blocks had in the image, with the 3-D transform realized as a separable composition of a 2-D block transform with a standard 1-D decorrelating transform along the stacking dimension of the group.

We here introduce a procedure for the adaptive design of a 3-D separable decorrelating transform based on the spatial coordinates of the similar blocks that enter the group, thus

This work was supported by the Academy of Finland (project no. 252547).

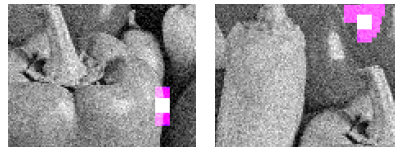


Fig. 1: Details of a noisy realization of the *Peppers* image. The purple area highlights patches similar to the white reference ones. Note how the patches that contain an edge are typically organized along the edge itself, in collinear relative positions (left); conversely, uniform regions yield groups of blocks scattered without a clear pattern (right).

leading to a spatially consistent collaborative filtering of the nonlocal features of the data.

The proposed procedure is designed to enhance BM3D performance whenever the extracted similar blocks spectra can be approximated as smooth functions of the blocks' spatial coordinates, while maintaining its usual performances in the generic case. We thus define, in Section 3, an adaptive orthonormal transform based on the spatial coordinates of the blocks entering the group.

Further, observing that the relative positions of the blocks in the group can be indicative of the block content (see Fig. 1), in Section 4 we also devise a sophisticated modification of collaborative sharpening [2] where both the position of the blocks and the dominant direction of each basis function modulate the strength of enhancement of the corresponding spectrum coefficient.

2. OBSERVATION MODEL AND BM3D

Let us consider a noisy observation z of a noise free signal y corrupted by additive i.i.d. Gaussian noise η ,

$$z(x) = y(x) + \eta(x), \tag{1}$$

where $x \in X \subset \mathbb{R}^2$ are the coordinates of the samples and $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$. The goal is to estimate y from z .

Image blocks are denoted by a capital letter and a subscript that indicates the top-left coordinate of the block; e.g.,

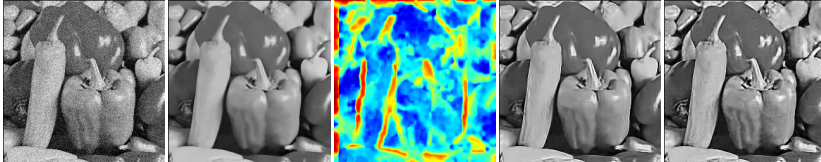


Fig. 2: Sharpening of *Peppers* corrupted by Gaussian noise with $\sigma = 20$. From left to right: *Peppers* noisy image; denoised image with no sharpening; ratio between the lengths of the principal axes of the group coordinates: higher values are shown in red, and lower values in dark blue; image denoised and sharpened by the proposed algorithm (adaptive $0.7 \leq \alpha \leq 1.8$); conventional alpha-rooting (constant $\alpha = 1.5$).

Z_x is a block extracted from z at position x .

In the basic BM3D formulation [1], the image z is raster scanned and for each position x_R and corresponding reference block Z_{x_R} , we find the *ordered* set S_{x_R} of the coordinates of the N blocks most similar to Z_{x_R} . The similarity is measured by computing the ℓ_2 -distances of the blocks content. The blocks are then stacked together in a 3-D volume $\mathbf{Z}_{S_{x_R}}$, that is subsequently transformed through the 3D separable linear transform \mathcal{T}_{3D} , filtered, and synthesized with the 3D inverse transform \mathcal{T}_{3D}^{-1} :

$$\hat{\mathbf{Y}}_{S_{x_R}} = \mathcal{T}_{3D}^{-1} \left(\Upsilon \left(\mathcal{T}_{3D} \left(\mathbf{Z}_{S_{x_R}} \right) \right) \right), \quad (2)$$

where $\hat{\mathbf{Y}}_{S_{x_R}}$ is the filtered 3-D volume, and Υ is a shrinkage operator, such as hard-thresholding or a Wiener filter. For the success of the procedure it is important that the underlying unknown spectrum $\mathcal{T}_{3D} \left(\mathbf{Y}_{S_{x_R}} \right)$ is sparse, as this permits to effectively attenuate the noise without introducing severe bias [1]. The coordinates S_{x_R} do not play any role in the collaborative filtering (2) applied to $\mathbf{Z}_{S_{x_R}}$; S_{x_R} regain their significance only when the block estimates $\hat{Y}_x, x \in S_{x_R}$, are extracted from $\hat{\mathbf{Y}}_{S_{x_R}}$ and aggregated at their original position into the resulting image estimate \hat{y} .

3. ADAPTIVE GROUP TRANSFORM BASED ON GROUP COORDINATES

Although a number of works [3, 4] have discussed the performance limits of denoising, indicating that BM3D is essentially attaining a performance bound on complex natural images, it is otherwise established that nonlocal methods often yields suboptimal results when filtering simpler images, such those composed by large regular surfaces. Thus, in this work, we aim at improving the effectiveness of collaborative filtering on such large regular content by embedding a smooth local model within the 3-D transform \mathcal{T}_{3D} , while maintaining comparable performance on complex heterogeneous images.

For simplicity, we consider the usual separable decomposition of \mathcal{T}_{3D} into a spatial 2-D transform \mathcal{T}_{2D} (e.g., DCT, DWT [1], PCA [5]) followed by orthonormal 1-D transform

\mathcal{T}_{1D} along the stacking dimension, and focus our attention on the latter. In particular, a curious feature of \mathcal{T}_{1D} , is that its particular choice bears negligible influence on the denoising performance, as long as \mathcal{T}_{1D} is chosen from fixed non-adaptive transforms having a DC term [1]. However, here we consider an adaptive design for \mathcal{T}_{1D} .

Let $\Phi = \{\phi_j\}_{j=1}^{|S_{x_R}|}$ be a collection of bivariate functions $\phi_j : X \rightarrow \mathbb{R}$, such that the one-dimensional vectors

$$\{\phi_j(S_{x_R})\}_{j=1}^{|S_{x_R}|} \quad (3)$$

form a set of linearly independent generators for $\mathbb{R}^{|S_{x_R}|}$. We can then build the $|S_{x_R}| \times |S_{x_R}|$ matrix P whose columns are the aforementioned vectors. Since the functions ϕ_j are linearly independent, *i.e.* $\text{rank}(P) = |S_{x_R}|$, we can apply the QR decomposition (Gram-Schmidt orthonormalization) to P . In this way we obtain

$$QR = P, \quad (4)$$

where Q is an orthonormal matrix and R is an upper-triangular matrix. We can interpret Q as an orthonormal transform for inputs of length $|S_{x_R}|$. Most importantly, the columns of Q (*i.e.* the basis functions of the transform) inherit from Φ the regularity w.r.t. S_{x_R} ; such spatially adaptive orthonormalization of P can be seen as a particular case of shape-adapted transforms [6]. We use such Q as a direct replacement of \mathcal{T}_{1D} in BM3D.

By design, the proposed transform yields sparser spectra when the individual \mathcal{T}_{2D} -spectrum coefficients of the blocks in $\mathbf{Y}_{S_{x_R}}$ agree with the regularity of Φ over $x \in S_{x_R}$. The validity of this assumption can be different for different \mathcal{T}_{2D} -spectrum components and different S_{x_R} ; it is reasonable to expect that high-frequency components of the \mathcal{T}_{2D} spectrum meet this assumption to a lesser degree than the low-frequency ones. In practice, for each group, Φ is expressed with respect to the principal axes of S_{x_R} , and in the orthonormalization (4) the basis functions ϕ_j are sorted so to maximize their independence over S_{x_R} . The principal axes play an essential role also in the following adaptive sharpening strategy.

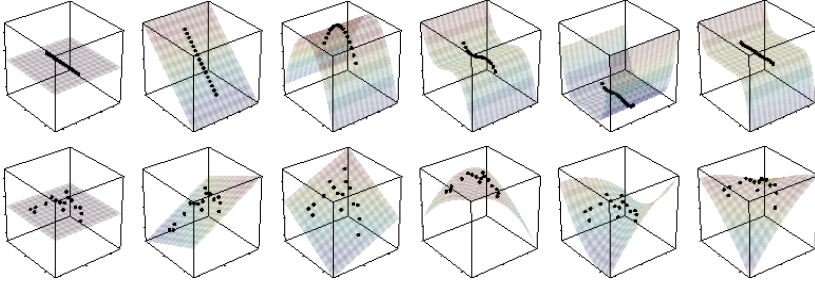


Fig. 3: Example of adaptive bases for the \mathcal{T}_{1D} transform generated by orthogonal polynomials. On the top row, we show the first 6 basis functions for the group in Fig. 1(left), whose blocks are aligned along an edge. On the bottom row, we show the first 6 basis functions for the group in Fig. 1(right), located within a uniform area. Observe how the bases change depending on the group coordinates.

4. DIRECTIONAL SHARPENING WITH ADAPTIVE TRANSFORM-DOMAIN ALPHA-ROOTING

The collaborative-filtering framework allows to conveniently sharpen the noisy image z without amplification of the noise through *alpha-rooting* of the thresholded \mathcal{T}_{3D} spectrum [2]. Here, instead of adopting an alpha-rooting of the image with constant α [2], we propose to amplify the \mathcal{T}_{3D} -spectrum coefficients $t(i)$ with adaptive $\alpha(i)$ that can be different for each i , i being the index within the \mathcal{T}_{3D} spectrum:

$$t_{\alpha(i)}(i) = \begin{cases} \text{sign}[t(i)] |t(0)| \left| \frac{t(i)}{t(0)} \right|^{\frac{1}{\alpha(i)}} & \text{if } t(0) \neq 0 \\ t(i) & \text{otherwise,} \end{cases} \quad (5)$$

where $t_{\alpha(i)}(i)$ is the amplified \mathcal{T}_{3D} coefficient and $t(0)$ is the \mathcal{T}_{3D} -spectrum DC.

Specifically, we adapt $\alpha(i)$ based on 1) the ratio between the major and minor axes of the group coordinates, as well as on 2) how the orientation of the major axis agrees with the orientation of the \mathcal{T}_{2D} basis functions that contribute to the individual coefficients $t(i)$ via \mathcal{T}_{1D} . If the major axis is sensibly longer than the minor one, the blocks are mostly distributed along the major axis: this occurs when the group includes pixels from a straight edge in the image. Conversely, when the axes' lengths are similar, the blocks are scattered, indicating that the group is likely from a wide regular region. This feature is shown in Fig. 2(center), where we show the ratio between major and minor axis for *Peppers*. Thus, when the ratio between the axes' lengths is large, we assume that the image features an intensity change in the direction orthogonal to the major axis, while when the ratio is close to 1, we assume that the group contains a regular region (see Fig. 1). Therefore, when sharpening the blocks content, it is reasonable 1) to use large α values only in the presence of an intensity change, i.e. when the ratio is large, and 2) to amplify only those coeffi-

cients whose corresponding basis functions are characterized by variations collinear with those of the image, i.e. orthogonal to the major axis. We thus compute individually the energy (ℓ_2 norm) of the derivative of each \mathcal{T}_{2D} -transform basis function in the direction of the major and minor axes, namely E_{\parallel} and E_{\perp} . If $E_{\perp} > E_{\parallel}$, we perform sharpening ($\alpha(i) > 1$), otherwise we soften the coefficient ($\alpha(i) < 1$). Hence, we modulate $\alpha(i)$ monotonically with $E_{\perp} - E_{\parallel}$ with rate proportional to the axes ratio.

5. EXPERIMENTS

In our experiments we use bivariate orthogonal polynomials for Φ . In Fig. 3 one can appreciate how the bases given by Q adapt to the particular relative positions of the grouped blocks. On the top row, we show the first six basis elements of the \mathcal{T}_{1D} transform of the group of blocks from the edge in Fig. 1(left). The first six basis functions shown in the bottom row are for the group from the flat area in Fig. 1(right), and are visibly different from the former ones. In the figure, we also show the continuous surfaces obtained by applying R^{-1} to Φ .

In order to compare the performances of the proposed algorithm with respect to the standard BM3D, we denoise a standard set of eleven natural images (*Lena*, *Cameraman*, *Peppers*, etc.) corrupted by i.i.d. Gaussian noise with $\sigma = 15, 35$ (1). We further consider a piecewise smooth test image (Fig. 4, bottom) to validate the advantage of the proposed scheme in its ideal conditions.

To clearly discriminate the performances of the two implementations, and also because collaborative sharpening had not been designed within a two-stage algorithm [2], we restrict our analysis to the hard-thresholding only. Furthermore, in order to have a fair comparison, we use the same parameters for both implementations except for the parameter λ of

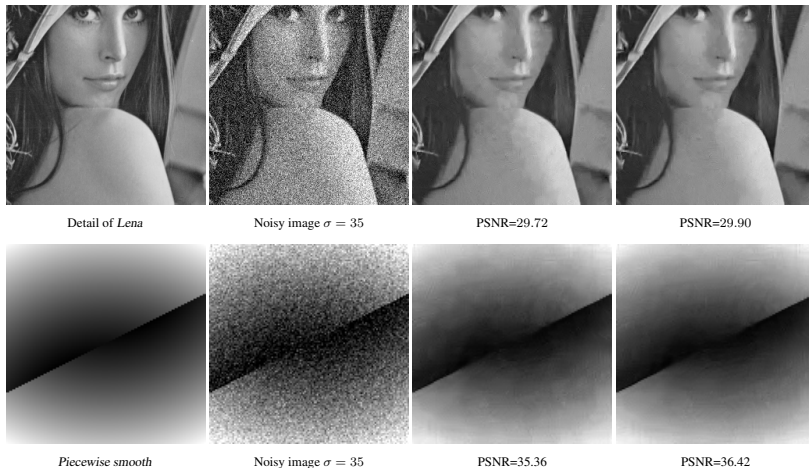


Fig. 4: Denoising of *Lena* (top) and piecewise-smooth test image (bottom), both corrupted by i.i.d. Gaussian noise with $\sigma = 35$. From left to right: original images, noisy observations, images denoised by the standard BM3D algorithm, images denoised by the proposed algorithm with collaborative filtering based on adaptive orthonormal polynomials (see Fig. 3). Notice the improvement, particularly in smooth regions, such as the shoulder of *Lena*.

the hard-thresholding operator Υ in (2), that has been optimized for the two algorithms independently, so to maximize the PSNR average over the dataset of natural images.

The PSNR differences between the proposed and the original implementation are, in terms of mean \pm standard-deviation over the natural images dataset, 0.076 ± 0.063 and 0.12 ± 0.065 for $\sigma = 15$ and $\sigma = 35$, respectively, i.e. there is a marginal but recurrent improvement. For the piecewise-smooth test image, the gain is instead $0.63dB$ and $1.28dB$ for the two noise levels, which is significant and confirms the advantage when the blocks spectra vary smoothly with respect to the group coordinates. Fig. 4 compares the results of the two implementations for *Lena* and for the test image, where one can appreciate the improvement on *Lena*'s shoulder as well as on the gradients of the bottom figure.

Finally, Fig. 2 compares the conventional sharpening algorithm [2] with constant $\alpha = 1.5$ against the proposed one with adaptive $0.7 \leq \alpha(i) \leq 1.8$. Although the two methods yield similar enhancement of the edges, the former one presents spurious artefacts particularly noticeable over smooth regions and in the vicinity of edges. These artefacts are instead mitigated by the proposed adaptive alpha-rooting, because the sharpening is strong only across edges, while softening occurs along edges (due to the different $\alpha(i)$ for differently oriented \mathcal{T}_{2D} -spectrum basis functions) and smooth

areas are merely denoised (due to the axes' length ratio being close to unity).

6. DISCUSSION AND CONCLUSIONS

We introduced an adaptive 1-D transform for 3-D collaborative filtering based on sampling 2-D smooth functions at the positions of grouped blocks. Experiments demonstrate a slight improvement over the standard BM3D on complex natural images and a substantial advantage on images characterized by large piecewise regular regions. These improvements are confirmed both quantitatively and qualitatively. This is consistent with results on the performance limits in image denoising, that predict only a small room for further improvement over BM3D on complex natural images. The proposed approach is further extended to a sharpening algorithm where the group coordinates determine an adaptive alpha-rooting whose strength can vary from group to group as well as within each group spectrum. Unlike the traditional alpha-rooting that amplifies the whole spectrum indiscriminately, the proposed method enables a directional sharpening which is only acting across edges, thus reducing the visibility of artefacts within smooth regions as well as along edges. The proposed scheme is not limited to denoising and sharpening, but it is relevant to any application of 3-D collaborative filtering.

7. REFERENCES

- [1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.
- [2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Joint image sharpening and denoising by 3D transform-domain collaborative filtering," in *Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP*, 2007.
- [3] P. Chatterjee and P. Milanfar, "Is denoising dead?," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 895–911, April 2010.
- [4] A. Levin, B. Nadler, F. Durand, and W.T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *Proc. European Conference on Computer Vision (ECCV 2012)*, pp. 73–86. Springer, 2012.
- [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D image denoising with shape-adaptive principal component analysis," in *Proc. 2009 workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)*, 2009.
- [6] W. Philips, "A fast algorithm for orthogonalizing polynomials on an arbitrarily shaped region (revised version)," *Multidimensional Systems and Signal Processing*, vol. 8, no. 4, pp. 409–421, 1997.

Publication IV

Azzari, L. and Foi, A. (2016c). Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising. *IEEE Signal Processing Letters*, 23(8):1086-1090.

© 2016 Institute of Electrical and Electronics Engineers (IEEE). Reprinted, with permission, from IEEE Signal Processing Letters.

Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising

Lucio Azzari and Alessandro Foi

Abstract—We denoise Poisson images with an iterative algorithm that progressively improves the effectiveness of variance-stabilizing transformations (VST) for Gaussian denoising filters. At each iteration, a combination of the Poisson observations with the denoised estimate from the previous iteration is treated as scaled Poisson data and filtered through a VST scheme. Due to the slight mismatch between a true scaled Poisson distribution and this combination, a special exact unbiased inverse is designed. We present an implementation of this approach based on the BM3D Gaussian denoising filter. With a computational cost at worst twice that of the non-iterative scheme, the proposed algorithm provides significantly better quality, particularly at low SNR, outperforming much costlier state-of-the-art alternatives.

Index Terms—image denoising, photon-limited imaging, Poisson noise, Anscombe transformation, iterative filtering.

I. INTRODUCTION

Denoising of images affected by Poisson noise is commonly executed by: 1) applying a variance stabilizing transformation (VST) to standardize the image noise, 2) denoising the image with an additive white Gaussian noise (AWGN) filter, 3) returning the image to its original range via inverse transformation. The most common VST for this purpose is the Anscombe transformation [1], [2]. Since, at very low counts (*e.g.*, less than one count per pixel, with $\text{SNR} \ll 0\text{dB}$), the Anscombe transformation can be quite inaccurate [3], denoising algorithms specifically designed for Poisson noise [3], [4] were developed to provide better performances than combinations of VST with Gaussian filters.

However, being inexpensive, simple, and independent from the adopted denoising algorithm, the Anscombe transform is still very appealing, and in this letter we propose an iterative algorithm based on the VST framework that is capable of dealing with challenging cases with extremely low SNR, and that outperforms state-of-the-art algorithms, both in terms of image quality and execution time.

At each step we apply the VST approach to a combination of the initial observed image and its most recent estimate to increase the SNR of the signal to be denoised, and consequently to improve the effectiveness of the stabilization and filtering. We analyze the statistics of this combination, which deviate from a Poisson distribution, and introduce the corresponding exact unbiased inverse to be used in this VST framework. We present an implementation of this approach based on the BM3D Gaussian denoising filter [5]. With a computational cost at worst twice that of the non-iterative scheme,

the proposed algorithm provides significantly better quality, particularly at low and extremely low SNR, outperforming much costlier state-of-the-art alternatives.

II. PRELIMINARIES AND MOTIVATION

Let z be an observed noisy image composed of pixels $z(x)$, $x \in \Omega \subset \mathbb{Z}^2$, modeled as independent realizations of a Poisson process with parameter $y(x) \geq 0$:

$$z(x) \sim \mathcal{P}(y(x)), \quad \text{P}(z(x) | y(x)) = \begin{cases} \frac{y(x)^{z(x)} e^{-y(x)}}{z(x)!} & z \in \mathbb{N} \cup \{0\} \\ 0 & \text{elsewhere.} \end{cases}$$

The mean and variance of $z(x)$ coincide and are equal to $y(x)$:

$$\text{E}\{z(x) | y(x)\} = \text{var}\{z(x) | y(x)\} = y(x).$$

For conciseness, henceforth we will omit x from notation.

Our goal is to compute an estimate \hat{y} of y from z . To this purpose, in the archetypal VST framework, the Anscombe forward transformation a [1] yields an image

$$a(z) = 2\sqrt{z + \frac{3}{8}}$$

which can be treated as corrupted by AWGN with unit variance. Thus, it can be denoised using any filter Φ designed for AWGN. If the denoising is ideal, we have

$$\Phi\{a(z)\} = \text{E}\{a(z) | y\}.$$

The so-called *exact unbiased inverse* of a [2]

$$\mathcal{I}_a^{\mathcal{P}} : \text{E}\{a(z) | y\} \mapsto \text{E}\{z | y\} = y,$$

is used to return the denoised image to the original range of z , thus yielding an estimate of y :

$$\hat{y} = \mathcal{I}_a^{\mathcal{P}}(\Phi\{a(z)\}).$$

However, for small y , when the SNR is very low, the stabilization is imprecise and the conditional distribution of $a(z)$ is far from the assumed normal, in terms of both scale and shape, leading to ineffective filtering with Φ . This issue has been commonly addressed either by applying VST after binning, *i.e.* by stabilizing sums of adjacent pixels instead of individual pixels [3], [4], [6]–[10], or by similarly stabilizing transform coefficients [11] (essentially inserting the VST within the denoising method itself). All these stratagems aim at increasing the SNR of the data subject to the VST.

In this letter, we introduce an alternative and more direct way to improve the SNR prior to VST, by combining the noisy observation z with a previously obtained estimate of the noise-free data y , leading to the following simple iterative algorithm.

This work was supported by the Academy of Finland (project no. 252547). The authors are with the Department of Signal Processing, Tampere University of Technology, Tampere, FI-33101, Finland (e-mail: lucio.azzari@tut.fi; alessandro.foi@tut.fi).

III. PROPOSED ITERATIVE ALGORITHM

A subscript index denotes a symbol's instance at a particular iteration, e.g., \hat{y}_i is the estimate of y at iteration i .

We initialize the algorithm by setting $\hat{y}_0 = z$. At each iteration $i = 1, \dots, K$ we compute a convex combination of \hat{y}_{i-1} and z

$$\bar{z}_i = \lambda_i z + (1 - \lambda_i) \hat{y}_{i-1}, \quad (1)$$

where $0 < \lambda_i \leq 1$. Provided we can treat \hat{y}_{i-1} as a surrogate for y , we have $E\{\bar{z}_i|y\} = y = \lambda_i^{-2} \text{var}\{\bar{z}_i|y\}$; thus \bar{z}_i has higher SNR than z for any $\lambda_i < 1$. We then apply a VST f_i to \bar{z}_i and obtain an image $\tilde{z}_i = f_i(\bar{z}_i)$, which we denoise with a filter Φ for AWGN to obtain a filtered image $D_i = \Phi[\tilde{z}_i]$. Assuming $D_i = E\{f_i(\tilde{z}_i)|y\}$, the exact unbiased inverse of f_i , $I_{f_i}^{\lambda_i} : E\{f_i(\tilde{z}_i)|y\} \mapsto E\{\tilde{z}_i|y\} = y$, brings this image to the original range, yielding

$$\hat{y}_i = I_{f_i}^{\lambda_i}(D_i),$$

which is either used for the next iteration if $i < K$, or output as final estimate $\hat{y}_K = \hat{y}$.

Let us provide further details on the above basic procedure.

A. Forward variance-stabilizing transformation

Consider the scaled variable $\lambda_i^{-2}\bar{z}_i$ and let us model \hat{y}_{i-1} as y . Setting $q_i(t) = \lambda_i t - \frac{1-\lambda_i}{\lambda_i} y$, the conditional probability

$$P(\lambda_i^{-2}\bar{z}_i|y) = \begin{cases} \frac{y^{q_i(\lambda_i^{-2}\bar{z}_i)} e^{-y}}{q_i(\lambda_i^{-2}\bar{z}_i)} & q_i(\lambda_i^{-2}\bar{z}_i) \in \mathbb{N} \cup \{0\} \\ 0 & \text{elsewhere.} \end{cases} \quad (2)$$

Unless $\lambda_i = 1$, this is *not* a Poisson distribution. However, the mean and variance of $\lambda_i^{-2}\bar{z}_i$ do always coincide:

$$E\{\lambda_i^{-2}\bar{z}_i|y\} = \text{var}\{\lambda_i^{-2}\bar{z}_i|y\} = \lambda_i^{-2}y.$$

Hence, $\lambda_i^{-2}\bar{z}_i$ resembles $\mathcal{P}(\lambda_i^{-2}y)$ and indeed one can prove [12] that it is asymptotically stabilized by the Anscombe transformation a . Thus, we set $f_i(\cdot) = a(\lambda_i^{-2}(\cdot))$.

B. Exact unbiased inverse transformation

The exact unbiased inverse $I_{f_i}^{\lambda_i}$ is defined upon (2) as

$$E\{f_i(\bar{z}_i)|y\} = \sum_{\bar{z}_i: q_i(\lambda_i^{-2}\bar{z}_i) \in \mathbb{N} \cup \{0\}} a(\lambda_i^{-2}\bar{z}_i) P(\lambda_i^{-2}\bar{z}_i|y) \mapsto E\{\bar{z}_i|y\} = y. \quad (3)$$

We have $I_{f_i}^{\lambda_i} \approx \lambda_i^2 I_a^{\mathcal{P}}$, with $I_a^{\mathcal{P}} = I_a^{\mathcal{P}}$ [2]. The appendix describes how to accurately compute (3) in practice.

C. Binning

It is natural to combine the convex combination (1) with a linear binning; this can be especially beneficial at the first iterations, when \hat{y}_{i-1} is a poor estimate of y . Specifically, a binning operator \mathcal{B}_{h_i} can be applied to \bar{z}_i , yielding a smaller image where each block (i.e. bin) of $h_i \times h_i$ pixels from \bar{z}_i is replaced by a single pixel equal to their sum. \mathcal{B}_{h_i} clearly commutes with (1) and

$$\mathcal{B}_{h_i}[\bar{z}_i] = \lambda_i \mathcal{B}_{h_i}[z] + (1 - \lambda_i) \mathcal{B}_{h_i}[\hat{y}_{i-1}].$$

Algorithm 1 Iterative Poisson Image Denoising via VST

```

1:  $\hat{y}_0 = z$ 
2: for  $i = 1$  to  $K$  do
3:    $\bar{z}_i = \lambda_i z + (1 - \lambda_i) \hat{y}_{i-1}$ 
4:    $\tilde{z}_i = f_i(\mathcal{B}_{h_i}[\bar{z}_i])$ 
5:    $D_i = \Phi[\tilde{z}_i]$ 
6:    $\hat{y}_i = \mathcal{B}_{h_i}^{-1}[I_{f_i}^{\lambda_i}(D_i)]$ 
7: end for
8: return  $\hat{y} = \hat{y}_K$ 

```

Algorithm 2 Debinning $\hat{y}_i = \mathcal{B}_{h_i}^{-1}[I_{f_i}^{\lambda_i}(D_i)]$

```

1:  $\hat{y}_{i,0} = 0$ 
2: for  $j = 1$  to  $J$  do
3:    $r_j = I_{f_i}^{\lambda_i}(D_i) - \mathcal{B}_{h_i}[\hat{y}_{i,j-1}]$ 
4:    $\hat{y}_{i,j} = \max\{0, \hat{y}_{i,j-1} + \mathcal{U}_{h_i}[h^{-2}r_j]\}$ 
5: end for
6: return  $\hat{y}_i = \hat{y}_{i,J}$ 

```

Since $\mathcal{B}_{h_i}[z] \sim \mathcal{P}(\mathcal{B}_{h_i}[y]) = \mathcal{P}(E\{\mathcal{B}_{h_i}[z]|y\})$, and modeling again \hat{y}_{i-1} as y , we have that $\mathcal{B}_{h_i}[\bar{z}_i]$ (resp. $\lambda_i^{-2}\mathcal{B}_{h_i}[\bar{z}_i]$) is subject to the same conditional probability of \bar{z}_i (resp. $\lambda_i^{-2}\bar{z}_i$), which means that the adoption of binning does not interfere with the subsequent VST, denoising, and inverse VST. Thus, we can define $\tilde{z}_i = f_i(\mathcal{B}_{h_i}[\bar{z}_i])$ without modifying f_i .

Debinning: An inverse binning operator $\mathcal{B}_{h_i}^{-1}$ is applied after the exact unbiased inversion,

$$\hat{y}_i = \mathcal{B}_{h_i}^{-1}[I_{f_i}^{\lambda_i}(D_i)],$$

returning a full-size image estimate \hat{y}_i such that

$$\mathcal{B}_{h_i}[\hat{y}_i] = I_{f_i}^{\lambda_i}(D_i). \quad (4)$$

All the above steps are summarized in Algorithm 1 and as

$$\hat{y}_i = \mathcal{B}_{h_i}^{-1}[I_{f_i}^{\lambda_i}(\Phi[f_i(\mathcal{B}_{h_i}[\lambda_i z + (1 - \lambda_i)\hat{y}_{i-1}])])].$$

IV. IMPLEMENTATION AND RESULTS

For Φ we adopt the BM3D denoising algorithm [5]; yet, other AWGN filters such as [13]–[20] may be used as well, and also lead to competitive results as shown in the supplementary material [21].

In the debinning step, to compute $\mathcal{B}_{h_i}^{-1}[I_{f_i}^{\lambda_i}(D_i)]$, $I_{f_i}^{\lambda_i}(D_i)$ is first divided by h_i^2 , i.e. by number of pixels in the bin, and upsampled to the size of z via cubic spline interpolation \mathcal{U}_{h_i} . To enforce the constraint (4), the output of interpolation is recursively binned by \mathcal{B}_{h_i} and subtracted from the target $I_{f_i}^{\lambda_i}(D_i)$, giving a residual which is upsampled and accumulated. This subroutine, summarized in Algorithm 2, is an instance of the recursive shaping regularization with nonnegativity [22], [23].

Our current implementation¹ of Algorithm 1 is determined by four parameters: K (number of iterations), λ_K, h_1, h_K (first and last bin sizes); other values of λ_i and h_i are defined as $\lambda_i = 1 - \frac{i}{K-1}(1-\lambda_K)$ and $h_i = \max\{h_K, h_1 - 2i + 2\}$. We use

¹Matlab software available at <http://www.cs.tut.fi/~foi/invansc/>

Table I
PSNR (dB) DENOISING RESULTS VS [2]–[4], [7], WITH AND WITHOUT 3×3 BINNING. AVERAGES OVER 5 NOISE REALIZATIONS. P^4 IP VALUES FROM [7].

Method	Peak	Flag _{256²}	House _{256²}	Cam _{256²}	Man _{112²}	Bridge _{256²}	Saturn _{256²}	Peppers _{256²}	Boat _{112²}	Couple _{112²}	Hill _{112²}	Time _{256²}
NLSPCA [3]	0.1	14.60	17.63	16.63	18.35	16.71	20.45	16.17	18.14	18.49	18.80	89s
NLSPCA bin [3]		15.32	18.66	17.23	18.41	16.99	18.91	16.22	18.89	18.84	19.47	11s
SPDA [4]		13.51	14.58	14.34	–	14.67	17.57	14.34	–	–	–	8h
SPDA bin [4]		15.27	18.29	16.83	18.72	17.00	21.53	16.15	18.99	18.94	19.39	12min
P^4 IP [7]		13.30	18.30	16.88	–	16.45	21.55	16.28	–	–	–	few mins
VST+BM3D [2]		12.38	15.69	15.44	16.95	15.60	18.40	15.15	16.22	16.50	16.79	0.70s
VST+BM3D bin [2]		13.97	18.22	16.99	18.61	16.93	20.09	15.84	18.91	18.62	19.23	0.11s
Proposed		16.01	18.48	17.45	18.96	17.30	21.64	16.45	19.32	19.31	19.68	0.48s
NLSPCA [3]	0.2	16.47	18.63	17.63	19.18	17.56	21.36	17.21	19.14	19.22	19.74	90s
NLSPCA bin [3]		15.63	19.21	17.87	19.12	17.40	19.67	16.69	19.48	19.37	19.99	12s
SPDA [4]		16.65	17.45	16.75	–	16.96	20.67	16.70	–	–	–	5h
SPDA bin [4]		17.41	18.95	17.80	19.73	17.81	22.90	17.25	19.85	19.72	20.36	27min
P^4 IP [7]		14.82	19.48	17.82	–	17.54	23.05	17.31	–	–	–	few mins
P^4 IP bin [7]		17.26	19.96	18.58	–	17.54	23.79	17.44	–	–	–	~30s
VST+BM3D [2]		13.53	17.79	16.90	18.69	17.12	21.38	16.96	18.23	18.47	18.80	0.69s
VST+BM3D bin [2]		16.85	19.27	17.88	19.82	17.70	22.94	17.19	19.79	19.71	20.09	0.12s
Proposed		17.49	19.68	18.40	19.94	18.13	23.13	17.54	20.09	20.03	20.48	0.83s
NLSPCA [3]	0.5	18.61	20.17	19.20	20.59	18.49	22.89	18.69	20.37	20.42	21.14	96s
NLSPCA bin [3]		15.76	20.48	18.26	19.77	18.17	21.65	17.69	20.11	20.01	20.67	19s
SPDA [4]		20.02	19.96	18.75	–	18.52	25.37	18.55	–	–	–	4h
SPDA bin [4]		18.40	20.57	18.87	20.70	18.57	25.93	18.52	20.84	20.70	21.35	23min
P^4 IP [7]		16.50	20.93	19.27	–	18.47	25.19	18.86	–	–	–	few mins
VST+BM3D [2]		15.58	19.61	18.46	20.39	18.26	23.75	18.41	19.99	20.01	20.74	0.71s
VST+BM3D bin [2]		18.19	21.41	19.47	21.15	18.71	25.81	18.78	20.94	20.83	21.72	0.11s
Proposed		18.60	21.54	19.79	21.25	19.08	25.78	19.05	21.19	21.14	21.84	0.83s
NLSPCA [3]	1	19.68	21.57	20.25	21.46	19.02	24.75	19.50	21.19	21.14	21.94	86s
NLSPCA bin [3]		15.77	20.78	18.40	19.87	18.26	22.83	17.78	20.19	20.11	20.82	16s
SPDA [4]		22.97	22.14	20.15	–	19.30	27.05	19.97	–	–	–	5h
SPDA bin [4]		18.99	20.99	19.43	21.15	18.84	27.40	18.93	21.19	20.97	21.50	25min
P^4 IP [7]		19.07	22.67	20.54	–	19.31	27.05	20.07	–	–	–	few mins
VST+BM3D [2]		18.46	21.64	20.19	21.62	19.43	25.82	19.71	21.47	21.14	21.92	0.78s
VST+BM3D bin [2]		19.28	22.53	20.69	22.07	19.59	27.59	20.22	21.97	21.81	22.72	0.10s
Proposed		19.74	23.04	21.07	22.30	19.86	27.27	20.44	22.17	22.08	22.85	0.82s
NLSPCA [3]	2	19.70	23.16	20.64	22.37	19.43	26.88	20.48	21.83	21.75	22.68	87s
NLSPCA bin [3]		15.52	20.85	18.35	19.87	18.32	21.27	17.78	20.29	20.21	20.98	12s
SPDA [4]		24.72	24.37	21.35	–	20.17	29.13	21.18	–	–	–	6h
SPDA bin [4]		19.26	21.12	19.53	21.66	18.87	28.54	19.17	21.43	21.24	21.94	25min
P^4 IP [7]		21.04	24.65	21.87	–	20.16	28.93	21.33	–	–	–	few mins
VST+BM3D [2]		20.79	23.79	21.97	23.11	20.49	27.95	22.02	22.90	22.65	23.34	0.82s
VST+BM3D bin [2]		19.91	24.10	21.43	23.03	20.36	29.26	21.45	22.92	22.84	23.75	0.10s
Proposed		21.18	24.62	22.25	23.40	20.69	28.85	21.93	23.30	23.12	23.88	0.82s
NLSPCA [3]	4	20.15	24.26	20.97	22.93	20.21	27.99	21.07	22.49	22.33	23.51	123s
NLSPCA bin [3]		15.52	20.94	18.27	19.88	18.32	22.02	17.72	20.29	20.25	20.99	13s
SPDA [4]		25.76	25.30	21.72	–	20.53	31.13	22.20	–	–	–	8h
SPDA bin [4]		19.42	22.07	19.95	22.18	19.26	29.71	20.19	21.76	21.69	22.82	31min
P^4 IP [7]		22.49	26.33	23.29	–	21.11	30.82	23.88	–	–	–	few mins
VST+BM3D [2]		22.93	25.49	23.82	24.32	21.51	29.41	24.01	24.16	24.10	24.47	0.74s
VST+BM3D bin [2]		20.43	25.49	22.22	23.99	21.13	30.87	22.57	23.92	23.84	24.69	0.10s
Proposed		23.51	26.07	24.10	24.52	21.71	30.38	24.04	24.53	24.34	24.82	1.41s

decreasing h_i since binning can cause loss of image details and it becomes progressively less useful when λ_i gets larger and the role of \hat{y}_{i-1} dominates in improving the SNR of the VST input. Obviously, \mathcal{B}_1 and \mathcal{B}_1^{-1} are identity operators.

The Poisson image z is the only input to our algorithm; the parameters K , λ_K , h_1 , h_K are adaptively selected based on the quantiles of z , following a training [21] over 6 images not included in the experiments test dataset; we fix $J = 9$.

PSNR (dB) results of the proposed algorithm and [2]–[4], [7], as well as their versions with binning, are reported in Table I. Table II gives a separate comparison with [8], over the different dataset of 256×256 downsampled images adopted by its authors. The tables demonstrate the superior overall performance of the proposed algorithm, also confirmed by visual inspection of the examples in Figure 1.

The complexity of Algorithm 1 is dominated by the filter Φ and possibly by the debinning operators \mathcal{B}_i^{-1} . The overall execution time depends especially on the number of iterations K and on h_K , which sets the size of the largest image to be filtered by Φ ; all our results have $K \leq 4$. Table I and Table II report also the average execution times for 256×256 images. We ran the proposed algorithm and [2]–[4] on a single thread of a 3.4-GHz Intel i7 CPU; the runtimes for [7], [8] are taken from the respective articles, where [8] uses a 3.3-GHz Intel i7, and [7] also uses an Intel i7. The proposed algorithm and [2] are significantly less expensive than any of the other methods.

V. DISCUSSION AND CONCLUSIONS

We presented an iterative VST framework for Poisson denoising. The iterative combination with a previous estimate refines

Table II
PSNR (dB) DENOISING RESULTS VERSUS THOSE REPORTED IN [8]. AVERAGES OVER 5 NOISE REALIZATIONS.

Method	Peak	Peppers _{256²}	Bridge _{256²}	Boat _{256²}	Couple _{256²}	Hill _{256²}	Mandrill _{256²}	Man _{256²}	Time _{256²}
MMSE est [8]	1	20.38	19.55	20.24	20.26	20.98	18.43	20.49	~14min
Proposed		20.44	19.86	20.65	20.47	21.23	18.56	20.50	0.82s
MMSE est [8]	2	22.26	20.65	21.28	21.22	22.05	18.98	21.60	~14min
Proposed		21.93	20.69	21.46	21.40	22.32	19.14	21.62	0.82s
MMSE est [8]	4	23.92	21.60	22.32	22.26	23.23	19.56	22.79	~14min
Proposed		24.04	21.71	22.53	22.52	23.29	19.65	22.75	1.41s

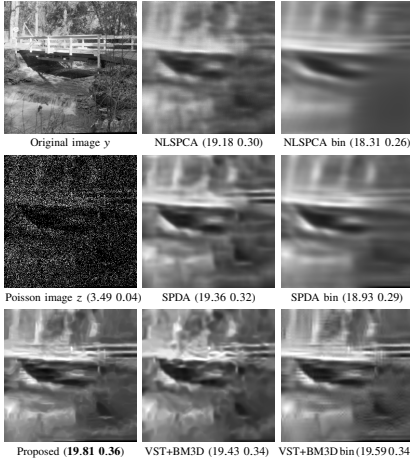


Figure 1. Denoising of *Bridge* at peak 1. PSNR (dB) and SSIM [24] of \hat{y} are given in brackets. For clarity, z is visualized on a compressed range.

the stabilization and helps to cope with extreme low-SNR cases, in which a standard VST approach [2] underperforms even when endowed with binning.

To analyze the importance of embedding the VST framework within the iterations, in Table III we compare our results from Table I with those by a simplified version of Algorithm 1, where the VST is *external* to the loop: f_i and $I_f^{A_i}$ are replaced by identity operators and a and I_a^P are applied outside of the algorithm. The significant gain in the table confirms that the improvement over [2] is not a mere consequence of a better denoising due to iterative filtering at multiple scales.

Also P⁴IP [7] relies on iterative AWGN filtering to denoise the Poisson z . In contrast to P⁴IP, which formulates an optimization problem to be solved upon convergence of ADMM [25] iterations, each iteration of Algorithm 1 attacks the Poisson denoising problem directly, so any \hat{y}_k can be treated as an estimate of y , with \hat{y}_1 already coinciding with [2]. This results in a more efficient, stable, and substantially faster procedure, where Φ (e.g., BM3D) is used as explicit denoiser for AWGN with variance 1 set by the VST without need of empirical tuning.

Table III
PSNR GAIN (AVERAGE OVER ALL IMAGES IN TABLE I) OF ALGORITHM 1 OVER ITS SIMPLIFICATION WITH *external* VST (SEE SECTION V).

Peak	0.1	0.2	0.5	1	2	4
PSNR (dB) gain	0.64	0.66	0.38	0.22	0.11	0.13

The proposed algorithm achieves state-of-the-art quality in only a tiny fraction of the time required by competitive algorithms.

APPENDIX: COMPUTING THE EXACT UNBIASED INVERSE

As in [2], we compute $E\{a(\lambda_i^{-2}\bar{z}_i)|y\}$ numerically over a finite grid of values of y and λ_i , from which we interpolate $I_f^{A_i}$ (3) at values within the grid range. Outside of the grid range, we leverage the available implementation [2] of the exact unbiased inverse for Poisson $I_a^P : E\{a(\lambda_i^{-2}\zeta)|y\} \mapsto \lambda_i^{-2}y$, where $\lambda_i^{-2}\zeta \sim \mathcal{P}(\lambda_i^{-2}y)$, through the composition

$$E\{a(\lambda_i^{-2}\bar{z}_i)|y\} \mapsto E\{a(\lambda_i^{-2}\zeta)|y\} \mapsto \lambda_i^{-2}y \mapsto y. \quad (5)$$

To deal with the first of the three mappings (5), we study the difference between $E\{a(\lambda_i^{-2}\bar{z}_i)|y\}$ and $E\{a(\lambda_i^{-2}\zeta)|y\}$. For $p \sim \mathcal{P}(\mu)$, the mean of a generic $g(p) = 2\sqrt{(p+d)/\gamma}$ is [1], [26]

$$E\{g(p)|\mu\} = 2\sqrt{\frac{\mu+d}{\gamma}} \left(1 - \frac{\mu}{8(\mu+d)^2} + \frac{1}{16} \frac{\mu}{(\mu+d)^3} - \frac{5}{128} \frac{3\mu^2+\mu}{(\mu+d)^4} + \mathcal{O}(\mu^{-3}) \right). \quad (6)$$

It yields $E\{a(\lambda_i^{-2}\bar{z}_i)|y\}$ when $\mu = y$, $\gamma = \lambda_i$, $d = \frac{1-\lambda_i}{\lambda_i}y + \frac{3}{8}\lambda_i$, and $E\{a(\lambda_i^{-2}\zeta)|y\}$ when $\mu = \lambda^{-2}y$, $\gamma = 1$, $d = \frac{3}{8}$. Then

$$E\{a(\lambda_i^{-2}\zeta)|y\} - E\{a(\lambda_i^{-2}\bar{z}_i)|y\} = \frac{\lambda_i^2(\lambda_i-1)}{8}y^{-\frac{3}{2}} + \mathcal{O}(y^{-\frac{5}{2}}), \quad (7)$$

which is however expressed as a function of y , while (5) requires a function of $E\{a(\lambda^{-2}\bar{z})|y\}$. From (6), we can approximate large y as

$$y = \left(\frac{\lambda_i}{2} E\{a(\lambda_i^{-2}\bar{z}_i)|y\} \right)^2 + \mathcal{O}(1). \quad (8)$$

On substituting (8) into (7) we obtain

$$\begin{aligned} E\{a(\lambda_i^{-2}\zeta)|y\} - E\{a(\lambda_i^{-2}\bar{z}_i)|y\} &= \\ &= \frac{\lambda_i-1}{\lambda_i} \left(E\{a(\lambda_i^{-2}\bar{z}_i)|y\} \right)^{-3} + \mathcal{O}\left(E\{a(\lambda_i^{-2}\bar{z}_i)|y\} \right)^{-4}. \end{aligned} \quad (9)$$

Outside of the grid range we can discard the higher-order terms from (9) and compute $I_f^{A_i}(D_i)$ using I_a^P [2] as

$$I_f^{A_i}(D_i) = \lambda_i^2 I_a^P \left(D_i + \frac{\lambda_i-1}{\lambda_i} D_i^{-3} \right).$$

REFERENCES

- [1] F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948. [Online]. Available: <http://www.jstor.org/stable/2332343>
- [2] M. Mäkitalo and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 99–109, Jan 2011. [Online]. Available: <http://dx.doi.org/10.1109/TIP.2010.2056693>. Matlab software at <http://www.cs.tut.fi/~foi/invansc/>.
- [3] J. Salmon, Z. Harnany, C.-A. Delledalle, and R. Willett, "Poisson noise reduction with non-local PCA," *Journal of Mathematical Imaging and Vision*, vol. 48, no. 2, pp. 279–294, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s10851-013-0435-6>. Software at <http://josephsalmon.eu/>.
- [4] R. Giryes and M. Elad, "Sparsity-based Poisson denoising with dictionary learning," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5057–5069, Dec 2014. [Online]. Available: <http://dx.doi.org/10.1109/TIP.2014.2362057>. Software at http://tam-son3.eng.tau.ac.il/~raja/?page_id=32.
- [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007. [Online]. Available: <http://dx.doi.org/10.1109/TIP.2007.901238>. Matlab software at <http://www.cs.tut.fi/~foi/GCF-BM3D/>.
- [6] L. Brown, T. Cai, R. Zhang, L. Zhao, and H. Zhou, "The root-unsroot algorithm for density estimation as implemented via wavelet block thresholding," *Probability Theory and Related Fields*, vol. 146, no. 3–4, pp. 401–433, 2010. [Online]. Available: <http://dx.doi.org/10.1007/s00440-008-0194-2>
- [7] A. Rond, R. Giryes, and M. Elad, "Poisson inverse problems by the plug-and-play scheme," *arXiv e-print*, 2015. [Online]. Available: <http://arxiv.org/abs/1511.02500>
- [8] S. Pyatykh and J. Hesser, "MMSE estimation for Poisson noise removal in images," *arXiv e-print*, 2015. [Online]. Available: <http://arxiv.org/abs/1512.00717>
- [9] T. Remez, O. Litany, and A. Bronstein, "A picture is worth a billion bits: Real-time image reconstruction from dense binary pixels," *arXiv e-print*, 2015. [Online]. Available: <http://arxiv.org/abs/1510.04601>
- [10] W. Feng and Y. Chen, "Fast and accurate Poisson denoising with optimized nonlinear diffusion," *arXiv e-print*, 2015. [Online]. Available: <http://arxiv.org/abs/1510.02930>
- [11] B. Zhang, J. M. Fadili, and J.-L. Starck, "Wavelets, ridgelets, and curvelets for Poisson noise removal," *IEEE Transaction on Image Processing*, vol. 17, no. 7, pp. 1093–1108, 2008. [Online]. Available: <http://dx.doi.org/10.1109/TIP.2008.924386>
- [12] S. K. Bar-Lev and P. Enis, "On the construction of classes of variance stabilizing transformations," *Statistics & Probability Letters*, vol. 10, no. 2, pp. 95–100, 1990. [Online]. Available: [http://dx.doi.org/10.1016/0167-7152\(90\)90002-0](http://dx.doi.org/10.1016/0167-7152(90)90002-0)
- [13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D image denoising with shape-adaptive principal component analysis," in *Proc. SPARS'09 - Signal Processing with Adaptive Sparse Structured Representations*, Saint Malo, France, 2009. [Online]. Available: <http://hal.inria.fr/inria-00369582>. Matlab software at <http://www.cs.tut.fi/~foi/GCF-BM3D/>.
- [14] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007. [Online]. Available: <http://doi.org/10.1109/TIP.2007.891788>. Matlab software at <http://www.cs.tut.fi/~foi/SA-DCT/>.
- [15] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005. [Online]. Available: <http://dx.doi.org/10.1137/040616024>. Matlab software at <http://www.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter>.
- [16] A. Foi and G. Boracchi, "Foveated nonlocal self-similarity," *International Journal of Computer Vision*, pp. 1–33, 2016, in press. [Online]. Available: <http://dx.doi.org/10.1007/s11263-016-0898-1>. Matlab software at <http://www.cs.tut.fi/~foi/FoveatedNL/>.
- [17] J. Boulanger, J.-B. Sibarita, C. Kervrann, and P. Bouthemy, "Non-parametric regression for patch-based fluorescence microscopy image sequence denoising," in *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI 2008)*, 2008, pp. 748–751. [Online]. Available: <http://dx.doi.org/10.1109/ISBI.2008.4541104>. Software at <http://serpico.rennes.inria.fr/doku.php?id=software:nd-safir:index>.
- [18] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1358–1351, Nov 2003. [Online]. Available: <http://dx.doi.org/10.1109/TIP.2003.818640>. Matlab software at <http://www.ncsic.es/PagsPers/Portilla/software/file/3-bis-gsm-image-denoising-toolbox-in-matlab>.
- [19] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006. [Online]. Available: <http://dx.doi.org/10.1109/TSP.2006.881199>. Matlab software at <http://www.cs.technion.ac.il/~elad/software>.
- [20] I. Ram, M. Elad, and I. Cohen, "Image denoising using NL-means via smooth patch ordering," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 1350–1354. [Online]. Available: <http://dx.doi.org/10.1109/ICASSP.2013.6637871>. Matlab software at <http://www.cs.technion.ac.il/~elad/software>.
- [21] L. Azzari and A. Foi, Supplementary to the manuscript "Variance stabilization for noisy-estimate combination in iterative Poisson denoising", 2016. [Online]. Available: <http://www.cs.tut.fi/~foi/invansc/>
- [22] Y. Chen, S. Fomel, and J. Hu, "Iterative deblending of simultaneous-source seismic data using seislet-domain shaping regularization," *Geophysics*, vol. 79, no. 5, pp. V179–V189, 2014. [Online]. Available: <http://reproducibility.org/RSF/book/tccs/deblend/paper.pdf>
- [23] Y. Chen, L. Zhang, and L.-w. Mo, "Seismic data interpolation using nonlinear shaping regularization," *Journal of Seismic Exploration*, 2015. [Online]. Available: <http://reproducibility.org/RSF/book/tccs/ntshape/paper.pdf>
- [24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004. [Online]. Available: <http://dx.doi.org/10.1109/TIP.2003.819861>
- [25] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: <http://dx.doi.org/10.1561/22000000016>
- [26] J.-L. Starck, F. Murtagh, and A. Bijaoui, *Image Processing and Data Analysis: The Multiscale Approach*. Cambridge University Press, 1998, Appendix A. [Online]. Available: <http://www.multiresolution.com/cupbook.pdf>

Supplementary to the manuscript “Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising”

Lucio Azzari and Alessandro Foi

SUPL.I. CONVEX COMBINATION AND VARIANCE STABILIZATION

Here we show that the increase of SNR due to the convex combination (Equation 1, in the manuscript) results in a direct improvement of the stabilization by the Anscombe transformation.

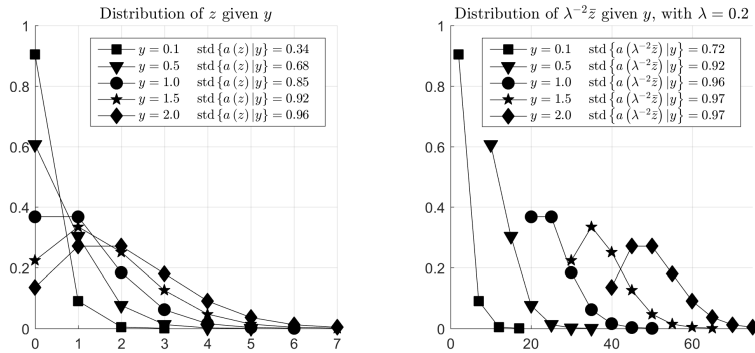


Figure Suppl.I.1. Effect of convex combination on the data distributions and on the standard deviation of the stabilized data.

The plot at the left in Figure Suppl.I.1 shows the Poisson distributions $P(z|y)$ with mean and variance $y = 0.1, 0.5, 1, 1.5, 2$. At the right, we show the distributions $P(\lambda^{-2}\bar{z}|y)$ (Equation 2 in the manuscript) of the data obtained after the convex combination with $\lambda = 0.2$. Note how the convex combination results in a shift of the distributions towards higher mean values and how the overlap between different distributions is reduced. Because of this reduced overlap, different distribution can benefit from the different slopes of the Anscombe transformation at the corresponding locations; this leads to a significantly more accurate stabilization. In particular, in the legends we report the standard deviations of the stabilized distributions, which for the combined data is much closer to the target value 1.

SUPL.II. EXPERIMENTS WITH DIFFERENT GAUSSIAN DENOISING FILTERS

Our manuscript reports extensive Poisson denoising results obtained by the proposed iterative VST framework, adopting the Block-Matching and 3D collaborative filtering (BM3D) algorithm [Suppl.1] as the specific Gaussian denoising filter used inside the iterations. Here we report the corresponding results obtained upon replacing BM3D by each of the following Gaussian denoising filters: BM3D with Shape-Adaptive Principal Components Analysis (SAPCA) [Suppl.2], Pointwise Shape-Adaptive Discrete Cosine Transform filter (SADCT) [Suppl.3], Non-Local Means (NLM) [Suppl.4], Anisotropic Foveated Non-Local Means (FOVNLN) [Suppl.5], Structure-Adaptive Filtering for Image Restoration (SAFIR) [Suppl.6], Bayesian Least Squares - Gaussian Scale Mixture (BLSGSM) [Suppl.7], K-SVD algorithm (KSVD) [Suppl.8], Non-Local Means via Smooth Patch Ordering (NLMPO) [Suppl.9].

Table Suppl.II.1 and Table Suppl.II.2 show that the presented framework gives excellent results consistently over these diverse set of Gaussian denoisers. In fact, most of these results are superior to those by state-of-the-art Poisson filters considered in Table I and Table II of the manuscript. A few examples are visualized in Figure Suppl.III.1.

Table Suppl.II.1
PSNR (dB) DENOISING RESULTS OF THE PROPOSED FRAMEWORK ADOPTING DIFFERENT AWGN DENOISERS. THE NOISY IMAGES ARE THE SAME OF TABLE I IN THE MANUSCRIPT.

Method	Peak	Flag _{256σ}	House _{256σ}	Cam _{256σ}	Man _{512σ}	Bridge _{256σ}	Saturn _{256σ}	Peppers _{256σ}	Boat _{512σ}	Couple _{512σ}	Hill _{512σ}
BM3D [Suppl.1]	0.1	16.01	18.48	17.45	18.96	17.30	21.64	16.45	19.32	19.31	19.68
SAPCA [Suppl.2]		15.88	18.02	17.59	18.99	17.25	21.78	16.52	19.35	19.24	19.66
SADCT [Suppl.3]		13.94	18.41	17.42	18.89	17.16	20.62	16.32	19.41	19.21	19.63
NLM [Suppl.4]		13.24	17.94	17.32	18.52	16.73	21.19	15.95	18.79	18.79	19.25
FOVNLN [Suppl.5]		13.72	18.03	17.51	18.61	16.98	21.40	16.10	19.13	18.91	19.25
SAFIR [Suppl.6]		14.24	18.48	16.97	18.55	16.99	20.73	16.19	19.15	18.90	19.38
BLSGSM [Suppl.7]		13.18	17.72	16.47	18.72	16.89	20.80	16.08	19.06	19.00	19.52
KSVd [Suppl.8]		14.94	18.07	17.03	18.38	16.89	20.55	15.91	19.18	18.81	18.94
NLMPO [Suppl.9]		16.12	18.09	17.44	18.67	17.16	21.80	16.26	19.29	19.12	19.40
BM3D [Suppl.1]		0.2	17.49	19.68	18.40	19.94	18.13	23.13	17.54	20.09	20.03
SAPCA [Suppl.2]	17.35		19.52	18.34	19.88	18.02	23.09	17.60	19.92	19.90	20.32
SADCT [Suppl.3]	15.30		19.41	18.29	19.82	18.00	21.72	17.35	20.16	19.97	20.48
NLM [Suppl.4]	13.89		18.74	17.81	19.32	17.39	22.18	16.86	19.40	19.41	19.91
FOVNLN [Suppl.5]	14.81		19.00	18.25	19.51	17.60	22.93	17.08	19.79	19.62	19.92
SAFIR [Suppl.6]	16.18		19.72	18.16	19.79	17.79	22.54	17.38	19.99	19.82	20.20
BLSGSM [Suppl.7]	14.71		18.76	17.51	19.68	17.81	21.93	17.13	19.91	19.84	20.38
KSVd [Suppl.8]	17.23		18.81	18.14	19.36	17.54	22.28	16.98	19.80	19.59	19.95
NLMPO [Suppl.9]	18.28		19.55	18.27	19.70	17.81	23.44	17.35	19.72	19.78	20.06
BM3D [Suppl.1]	0.5		18.60	21.54	19.79	21.35	19.08	25.78	19.05	21.19	21.14
SAPCA [Suppl.2]		18.46	21.54	19.91	21.26	19.03	25.72	19.21	21.19	21.08	21.78
SADCT [Suppl.3]		17.01	21.07	19.80	21.18	19.03	23.81	19.01	21.18	21.05	21.82
NLM [Suppl.4]		16.97	20.31	19.41	20.82	18.43	25.37	18.50	20.73	20.57	21.09
FOVNLN [Suppl.5]		17.63	21.01	19.78	21.14	18.71	25.53	18.75	20.96	20.76	21.31
SAFIR [Suppl.6]		18.92	21.23	19.59	21.17	18.80	24.98	18.94	21.04	20.93	21.62
BLSGSM [Suppl.7]		16.48	20.42	18.84	20.92	18.80	24.15	18.67	20.96	20.85	21.64
KSVd [Suppl.8]		18.74	21.08	19.71	21.07	18.82	24.87	18.95	21.09	20.87	21.37
NLMPO [Suppl.9]		19.41	21.72	19.90	21.26	18.95	26.16	19.01	21.05	20.94	21.63
BM3D [Suppl.1]		1	19.74	23.04	21.07	22.30	19.86	27.27	20.44	22.17	22.08
SAPCA [Suppl.2]	19.54		23.05	21.17	22.36	19.86	27.39	20.57	22.28	22.08	22.83
SADCT [Suppl.3]	18.44		22.51	20.94	22.20	19.82	25.48	20.40	22.06	21.97	22.79
NLM [Suppl.4]	17.89		21.50	20.46	21.85	19.33	26.95	19.86	21.56	21.41	22.07
FOVNLN [Suppl.5]	18.93		22.30	20.93	22.19	19.54	27.07	20.18	21.88	21.68	22.31
SAFIR [Suppl.6]	20.36		22.90	20.87	22.26	19.65	26.72	20.42	21.94	21.86	22.67
BLSGSM [Suppl.7]	17.53		21.92	19.96	21.93	19.59	25.51	19.98	21.90	21.75	22.62
KSVd [Suppl.8]	20.19		22.26	20.83	22.04	19.67	26.59	20.34	21.90	21.67	22.40
NLMPO [Suppl.9]	20.67		23.07	21.09	22.24	19.70	27.87	20.36	21.94	21.81	22.63
BM3D [Suppl.1]	2		21.18	24.62	22.25	23.40	20.69	28.85	21.93	23.30	23.12
SAPCA [Suppl.2]		20.86	24.65	22.35	23.46	20.67	29.10	22.18	23.38	23.12	23.88
SADCT [Suppl.3]		19.97	23.94	22.02	23.27	20.57	27.20	21.89	23.03	22.93	23.78
NLM [Suppl.4]		18.89	22.84	21.37	22.85	20.12	28.28	21.22	22.41	22.28	23.05
FOVNLN [Suppl.5]		20.30	23.82	21.99	23.20	20.31	28.65	21.66	22.80	22.59	23.34
SAFIR [Suppl.6]		21.74	24.64	22.17	23.35	20.42	28.66	22.09	23.13	22.91	23.79
BLSGSM [Suppl.7]		18.47	23.30	21.00	22.93	20.38	27.08	21.16	22.88	22.73	23.61
KSVd [Suppl.8]		21.49	23.52	21.73	22.92	20.30	28.27	21.52	22.63	22.42	23.31
NLMPO [Suppl.9]		21.84	24.64	22.22	23.21	20.38	29.67	21.82	23.05	22.77	23.61
BM3D [Suppl.1]		4	23.51	26.07	24.10	24.52	21.71	30.38	24.04	24.53	24.34
SAPCA [Suppl.2]	22.85		25.77	24.11	24.57	21.69	30.80	24.25	24.52	24.33	24.77
SADCT [Suppl.3]	21.69		25.46	23.66	24.41	21.61	28.96	23.91	24.27	24.15	24.82
NLM [Suppl.4]	20.81		23.88	22.79	23.89	21.04	29.10	23.18	23.51	23.28	24.02
FOVNLN [Suppl.5]	21.65		25.07	23.61	24.27	21.28	29.97	23.74	24.08	23.70	24.35
SAFIR [Suppl.6]	23.11		26.38	23.99	24.56	21.48	30.70	24.25	24.46	24.17	24.80
BLSGSM [Suppl.7]	20.16		24.81	22.38	24.06	21.41	28.76	22.91	24.10	23.89	24.62
KSVd [Suppl.8]	23.58		24.50	23.35	24.00	21.22	29.32	23.51	23.85	23.48	24.24
NLMPO [Suppl.9]	23.97		26.40	24.11	24.38	21.40	31.41	24.14	24.37	24.06	24.62

SUPL.III. PARAMETER TRAINING AND ADAPTIVE SELECTION

As mentioned in Section IV of the manuscript, our implementation of the algorithm has four parameters, namely K , λ_K , h_1 , and h_K . We have tested multiple combinations of values for these parameters over a training set of 6 images and for different peak values. We remark that none of these images has any overlap with the test images in the manuscript. For each peak, we selected the set of parameter values that yields the highest PSNR average over the training images at that particular peak. We also compute the 21 uniform 5% quantiles of each noisy image, and model the quantile distribution of a generic training image at each peak as a 21-dimensional multivariate normal, whose mean and diagonal covariance are given by the sample means and by the sample variances of the 21 quantiles over all training images at that peak. Thus, for each peak, we have a set of four selected parameter values, and the sample means and sample variances of the image quantiles at that peak. All these data, for all peaks, are stored in a look-up table. Our idea is to leverage the quantile distribution as a proxy for the peak, from which to identify an appropriate set of parameter values for any given input image not found in the training set.

Table Suppl.II.2
PSNR (dB) DENOISING RESULTS OF THE PROPOSED FRAMEWORK ADOPTING DIFFERENT AWGN DENOISERS. THE NOISY IMAGES ARE THE SAME OF TABLE II IN THE ORIGINAL MANUSCRIPT.

Method	Peak	Peppers _{256²}	Bridge _{256²}	Boat _{256²}	Couple _{256²}	Hill _{256²}	Mandrill _{256²}	Man _{256²}
BM3D [Suppl.1]	0.1	16.45	17.30	17.91	17.86	18.43	16.84	17.21
SAPCA [Suppl.2]		16.52	17.25	17.75	17.91	18.48	16.60	17.23
SADCT [Suppl.3]		16.32	17.16	17.75	17.73	18.21	16.87	17.15
NLM [Suppl.4]		15.95	16.73	17.33	17.52	17.95	16.63	16.83
FOVNLN [Suppl.5]		16.10	16.98	17.59	17.57	18.04	16.47	16.92
SAFIR [Suppl.6]		16.19	16.99	17.65	17.36	18.05	16.71	16.80
BLSGSM [Suppl.7]		16.08	16.89	17.16	17.60	17.78	16.66	16.83
KSVL [Suppl.8]		15.91	16.89	17.40	17.34	17.37	16.67	16.65
NLMPO [Suppl.9]		16.26	17.16	17.65	17.45	18.12	16.62	17.07
BM3D [Suppl.1]	0.2	17.54	18.13	18.74	18.62	19.16	17.36	18.17
SAPCA [Suppl.2]		17.60	18.02	18.64	18.53	19.10	17.23	18.19
SADCT [Suppl.3]		17.35	18.00	18.64	18.46	19.04	17.44	18.10
NLM [Suppl.4]		16.86	17.39	17.92	18.13	18.64	17.22	17.69
FOVNLN [Suppl.5]		17.08	17.60	18.27	18.20	18.60	17.13	17.78
SAFIR [Suppl.6]		17.38	17.79	18.41	18.25	18.89	17.32	17.84
BLSGSM [Suppl.7]		17.13	17.81	18.17	18.34	18.85	17.15	17.86
KSVL [Suppl.8]		16.98	17.54	18.36	18.08	18.42	17.17	17.62
NLMPO [Suppl.9]		17.35	17.81	18.64	18.30	18.90	17.38	17.91
BM3D [Suppl.1]	0.5	19.05	19.08	19.80	19.65	20.40	18.08	19.48
SAPCA [Suppl.2]		19.21	19.03	19.75	19.63	20.35	18.05	19.54
SADCT [Suppl.3]		19.01	19.03	19.83	19.50	20.29	18.09	19.45
NLM [Suppl.4]		18.50	18.43	19.29	19.18	19.73	17.89	18.99
FOVNLN [Suppl.5]		18.75	18.71	19.60	19.37	19.90	17.94	19.27
SAFIR [Suppl.6]		18.94	18.80	19.59	19.26	20.06	18.04	19.17
BLSGSM [Suppl.7]		18.67	18.80	19.33	19.33	19.94	17.90	19.21
KSVL [Suppl.8]		18.95	18.82	19.72	19.39	19.88	17.91	19.25
NLMPO [Suppl.9]		19.01	18.95	19.77	19.51	20.23	18.05	19.38
BM3D [Suppl.1]	1	20.44	19.86	20.65	20.47	21.23	18.56	20.50
SAPCA [Suppl.2]		20.57	19.86	20.65	20.43	21.21	18.53	20.56
SADCT [Suppl.3]		20.40	19.82	20.61	20.35	21.19	18.55	20.45
NLM [Suppl.4]		19.86	19.33	20.13	19.96	20.63	18.39	20.02
FOVNLN [Suppl.5]		20.18	19.54	20.43	20.20	20.80	18.48	20.34
SAFIR [Suppl.6]		20.42	19.65	20.44	20.20	20.88	18.51	20.28
BLSGSM [Suppl.7]		19.98	19.59	20.31	20.16	21.02	18.41	20.19
KSVL [Suppl.8]		20.34	19.67	20.50	20.22	20.86	18.42	20.25
NLMPO [Suppl.9]		20.36	19.70	20.50	20.26	21.05	18.50	20.40
BM3D [Suppl.1]	2	21.93	20.69	21.46	21.40	22.32	19.14	21.62
SAPCA [Suppl.2]		22.18	20.67	21.46	21.35	22.28	19.13	21.69
SADCT [Suppl.3]		21.89	20.57	21.39	21.25	22.21	19.10	21.53
NLM [Suppl.4]		21.22	20.12	20.95	20.78	21.57	18.95	21.13
FOVNLN [Suppl.5]		21.66	20.31	21.21	21.03	21.81	19.09	21.50
SAFIR [Suppl.6]		22.09	20.42	21.26	21.09	22.08	19.05	21.49
BLSGSM [Suppl.7]		21.16	20.38	21.13	21.04	22.05	19.03	21.25
KSVL [Suppl.8]		21.52	20.30	21.14	20.90	21.78	18.99	21.17
NLMPO [Suppl.9]		21.82	20.38	21.23	21.00	22.01	19.01	21.38
BM3D [Suppl.1]	4	24.04	21.71	22.53	22.52	23.29	19.65	22.75
SAPCA [Suppl.2]		24.25	21.69	22.59	22.50	23.23	19.68	22.82
SADCT [Suppl.3]		23.91	21.61	22.38	22.36	23.20	19.62	22.67
NLM [Suppl.4]		23.18	21.04	21.85	21.79	22.51	19.45	22.30
FOVNLN [Suppl.5]		23.74	21.28	22.20	22.08	22.71	19.60	22.65
SAFIR [Suppl.6]		24.25	21.48	22.32	22.30	23.10	19.51	22.71
BLSGSM [Suppl.7]		22.91	21.41	22.17	22.13	23.00	19.64	22.32
KSVL [Suppl.8]		23.51	21.22	22.10	21.92	22.61	19.46	22.32
NLMPO [Suppl.9]		24.14	21.40	22.39	22.21	23.00	19.46	22.55

Therefore, our denoising algorithm, given an input unknown noisy Poisson image, proceeds as follows in order to decide the values for the four parameters. First, it computes the input image's 5% quantiles. Next, it compares them against the look-up table: specifically, it computes the likelihoods that the quantiles of the input image belong to the quantile distributions measured in the training for each peak. Finally, each of the four parameters is computed as the weighted average of the parameters in the look-up table, using the above likelihoods as weights over all training peaks.

We emphasize that using the look-up table or computing the quantiles is computationally inexpensive and particularly it is insignificant when compared to other parts of the denoising process.

REFERENCES

[Suppl.1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080-2095, Aug 2007, [Online]. Available: <http://dx.doi.org/10.1109/TIP.2007.901238>. Matlab software at <http://www.cs.tut.fi/~foi/GCF-BM3D/>.

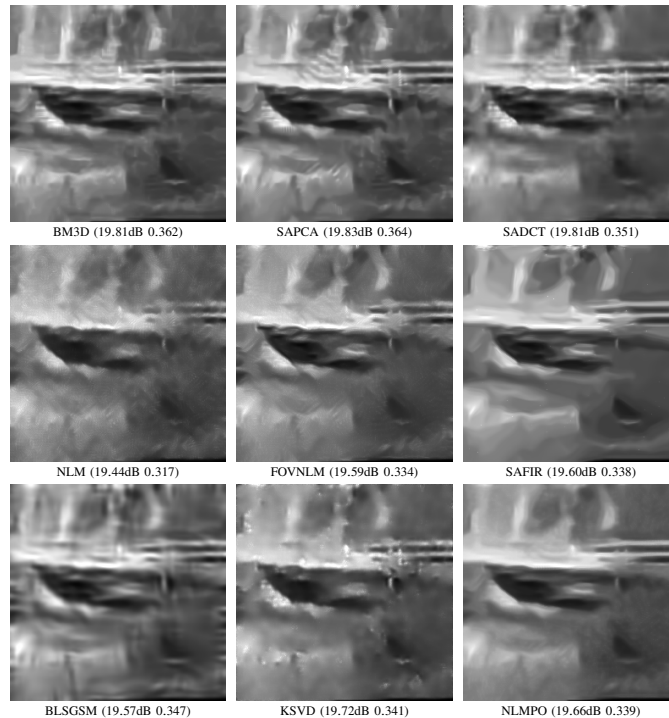


Figure Suppl.III.1. Denoising of *Bridge* at peak 1. PSNR and SSIM of \hat{y} are given in brackets.

- [Suppl.2] —, “BM3D image denoising with shape-adaptive principal component analysis,” in *Proc. SPARS’09 - Signal Processing with Adaptive Sparse Structured Representations*, Saint Malo, France, 2009, [Online]. Available: <http://hal.inria.fr/inria-00369582>. Matlab software at <http://www.cs.tut.fi/~foi/GCF-BM3D/>.
- [Suppl.3] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images,” *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007, [Online]. Available: <http://doi.org/10.1109/TIP.2007.891788>. Matlab software at <http://www.cs.tut.fi/~foi/SA-DCT/>.
- [Suppl.4] A. Buades, B. Coll, and J. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005, [Online]. Available: <http://dx.doi.org/10.1137/040616024>. Matlab software at <http://www.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter>.
- [Suppl.5] A. Foi and G. Boracchi, “Foveated nonlocal self-similarity,” *International Journal of Computer Vision*, pp. 1–33, 2016, in press. [Online]. Available: <http://dx.doi.org/10.1007/s1263-016-0898-1>. Matlab software at <http://www.cs.tut.fi/~foi/FoveatedNL/>.
- [Suppl.6] J. Boulanger, J.-B. Sibarita, C. Kervrann, and P. Bouthemy, “Non-parametric regression for patch-based fluorescence microscopy image sequence denoising,” in *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI 2008)*, 2008, pp. 748–751, [Online]. Available: <http://dx.doi.org/10.1109/ISBI.2008.4541104>. Software at <http://serpico.rennes.inria.fr/doku.php?id=software:nd-safir:index>.
- [Suppl.7] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, “Image denoising using scale mixtures of gaussians in the wavelet domain,” *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, Nov 2003, [Online]. Available: <http://dx.doi.org/10.1109/TIP.2003.818640>. Matlab software at <http://www.io.csic.es/PagsPers/Portilla/software/file/3-bls-gsm-image-denoising-toolbox-in-matlab>.
- [Suppl.8] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006, [Online]. Available: <http://dx.doi.org/10.1109/TSP.2006.881199>. Matlab software at <http://www.cs.technion.ac.il/~elad/software>.
- [Suppl.9] I. Ram, M. Elad, and I. Cohen, “Image denoising using NL-means via smooth patch ordering,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 1350–1354, [Online]. Available: <http://dx.doi.org/10.1109/ICASSP.2013.6637871>. Matlab software at <http://www.cs.technion.ac.il/~elad/software>.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3837-7
ISSN 1459-2045