TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Markku Mäkitalo

# Exact Unbiased Inverse of the Anscombe Transformation and its Poisson-Gaussian Generalization

Julkaisu 1115 • Publication 1115

Tampere 2013

Tampereen teknillinen yliopisto. Julkaisu 1115
Tampere University of Technology. Publication 1115

Markku Mäkitalo

# Exact Unbiased Inverse of the Anscombe Transformation and its Poisson-Gaussian Generalization

Thesis for the degree of Doctor of Philosophy to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 1st of March 2013, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2013

ISBN 978-952-15-3021-0 (printed)
ISBN 978-952-15-3039-5 (PDF)
ISSN 1459-2045

# Abstract

Digital image acquisition is an intricate process, which is subject to various errors. Some of these errors are signal-dependent, whereas others are signal-independent. In particular, photon emission and sensing are inherently random physical processes, which in turn substantially contribute to the randomness in the output of the imaging sensor. This signal-dependent noise can be approximated through a Poisson distribution. On the other hand, there are various signal-independent noise sources involved in the image capturing chain, arising from the physical properties and imperfections of the imaging hardware. The noise attributed to these sources is typically modelled collectively as additive white Gaussian noise. Hence, we have three common ways of modelling the noise present in a digital image: Gaussian, Poisson, or Poisson-Gaussian.

Image denoising aims at removing or attenuating this noise from the captured image, in order to provide an estimate of the underlying ideal noise-free image. For simplicity, denoising algorithms often assume the noise to be Gaussian, and ignore the signal-dependency. However, in an image corrupted by signal-dependent noise, the noise variance is typically not constant and varies with the expectation of the pixel value. Thus, for the removal of signal-dependent noise, we can either design an algorithm specifically for the particular noise model, or use an indirect three-step variance-stabilization approach. In the indirect approach, the noisy image is first processed with a variance-stabilizing transformation (VST), which renders the noise approximately Gaussian with a known constant variance. Then the transformed image is denoised with a Gaussian denoising algorithm, and finally an inverse VST is applied to the denoised data, providing us with the final estimate of the noise-free image.

For the Poisson and Poisson-Gaussian cases, the most well-known VSTs are the Anscombe transformation and the generalized Anscombe transformation, respectively. The former of these was introduced in the late 1940s, and has been in wide use ever since in all areas of applied statistics where the Poisson distribution is relevant.

In addition to a suitable VST, the above indirect denoising process requires a properly designed inverse VST. Unfortunately, the importance of the inverse VST is often neglected. The simple inverse VST obtained by taking the direct algebraic inverse of the VST generally leads to a biased estimate. Usually the issue of bias is addressed by using an asymptotically unbiased inverse, but also this inverse remains highly inadequate, especially for low-intensity data (e.g., for an image captured in a dark environment or with a very short exposure time).

In this thesis, we address this inadequacy by first proposing an exact unbiased inverse of the Anscombe transformation for the Poisson noise model, and then generalizing it into an exact unbiased inverse of the generalized Anscombe transformation for the Poisson-Gaussian noise model. We show that its role in the denoising process is significant, and that by replacing a traditional inverse with an exact unbiased inverse, we obtain substantial gains in the denoising results, especially for low-intensity data. Moreover, in combination with a state-of-the-art Gaussian denoising algorithm, the proposed method is competitive with the best Poisson and Poisson-Gaussian denoising algorithms. We provide extensive experimental results, and supplement them with rigorous mathematical considerations about the optimality and accuracy of the proposed inverses. In addition, we construct a closed-form approximation for both of these exact unbiased inverses.

In practical applications involving noise removal, identifying a suitable noise model does not guarantee accurate denoising results per se, but for the best results we must also be able to produce reasonable estimates of the noise model parameters. Hence, we conclude the thesis by investigating the effect of inaccurate parameter estimation on variance stabilization; based on the theoretical results, we also devise a novel way of estimating Poisson-Gaussian noise parameters from a single image using an iterative variance-stabilization scheme.

# Foreword

This thesis is based on research performed by the author at the Department of Signal Processing, Tampere University of Technology, Finland, between January 2009 and February 2013.

The research has been carried out under the supervision of Dr. Alessandro Foi, whom I wish to thank for all of his efforts. He has not only supervised and guided me throughout this entire period, but also co-authored all of my publications. Without his ideas and insights, this work would not have been possible in the first place.

I am also grateful to the Tampere Doctoral Programme in Information Science and Engineering (TISE) for providing me with financial support from May 2010 until the present.

Finally, and most importantly, I wish to thank my family for their continuous and unconditional support and interest towards my research and studies.

Tampere, February 2013 Markku Mäkitalo

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

The first digital still camera was developed by an Eastman Kodak engineer Steven Sasson in 1975 [121]. Nowadays digital images are a part of our everyday life, and they are being produced by a variety of devices ranging from digital cameras and mobile phones to medical and astronomical image acquisition devices. In the heart of a typical digital imaging device is an image sensor; an array of light-sensitive photodiodes, gathering the incoming photons focused onto the sensor by a lens or a system of lenses. The most common sensor technologies are *charge-coupled device* [17] (CCD) and *complementary metal-oxide-semiconductor* [117, 139, 23] (CMOS), both dating back to the late 1960s. While the grayscale CCD sensor in Sasson's camera consisted of an array of merely $100 \times 100$ photodiodes (in essence, pixels), modern sensors contain millions or even tens of millions of pixels.

The process of capturing a digital image using a CCD or CMOS sensor is subject to various errors, some of which are signal-dependent and some of which are signal-independent. In particular, photon emission and sensing are inherently random physical processes, which in turn substantially contribute to the randomness in the sensor output; this is also known as shot noise. More specifically, the number of photons in a light beam inevitably fluctuates over time, even if the light source has a constant intensity [88]. As first shown by L. Mandel in 1959 [105], this fluctuation can be considered to follow the well-known Poisson distribution. Thus, a standard way of modelling the noise present in digital images is to assume that the image has been corrupted by Poisson distributed noise (and to implicitly assume that this noise dominates over all the other noise sources).

A more elaborate approach is to also take the signal-independent noise sources into account, modelling the noise with a Poisson-Gaussian distribution. In other words, this noise model employs a Poisson component in order to account for the signal-dependent uncertainty due to random photon emission and sensing, and complementarily, the additive Gaussian component accounts for the other signal-independent noise sources involved in the capturing chain. This modelling has been successfully used in several practical applications, e.g., in noise fitting and denoising of clipped and non-clipped raw CCD data [64, 67], in the denoising of

fluorescence microscopy images [145, 98], in fluorescent-spot detection [145], and in astronomy [126].

Image denoising aims at removing or attenuating this noise from the captured image, in order to provide an estimate of the underlying ideal noise-free image. The strategies for accomplishing this task vary greatly, not only depending on the chosen statistical image model, but also within a particular a priori model.

From the point of view of this thesis, there are two main options for approaching the problem of denoising images corrupted by signal-dependent noise. One option is to directly consider the statistics of the particular noise model, and take advantage of these properties and observations in designing an effective denoising algorithm. Our focus is on the alternative approach, where the problem is tackled in a modular fashion through variance stabilization. This general denoising process involves three steps. First, the noisy data is modified by applying a nonlinear variance-stabilizing transformation (VST) specifically designed for the chosen noise model. Note that in an image corrupted by signal-dependent noise, the noise variance is typically not constant and varies with the expectation of the pixel value. For instance, the variance of a Poisson variable equals its mean. Thus, the rationale of applying a VST is to remove this signal-dependency by rendering the noise variance constant throughout the image. In particular, the transformed data can be assumed to have an approximately Gaussian noise distribution with a known constant (e.g., unitary) variance. Hence, the second step is to treat the noisy data with any algorithm designed for the removal of Gaussian noise. Finally, the desired estimate of the unknown noise-free image is obtained by applying an inverse VST to the denoised data (i.e., by returning the denoised data to the original range). This modular approach has several practical advantages: Not only is the problem of Gaussian noise removal a well studied and widely covered topic, with a plethora of denoising algorithms to choose from, but it also allows the practical implementation of the denoising framework to be divided into self-contained modules, which may be designed and optimized independent of each other.

Considering the three-step denoising procedure laid out above, it is obvious that a properly designed VST is required in order to achieve good variance stabilization. Unfortunately, the importance of the inverse VST is often neglected, even though its role is certainly not insignificant, in particular when denoising low-intensity data. Typical cases where the data is of low intensity are images captured with a regular digital camera in a dark environment or with a very short exposure time. On the other hand, low intensities can be unavoidable in some special applications. For example in fluorescence microscopy, the organism under inspection may exhibit permanent loss of fluorescence after being exposed to a sufficient amount of light; this phenomenon is also known as photobleaching [136]. Perhaps the first intuition is to simply define the inverse VST as the algebraic inverse of the forward VST. However, applying this inverse to the denoised data generally introduces bias to the final estimate, because the forward VST is nonlinear. In general terms, transforming a variable nonlinearly, then computing its expected value (ideal result of denoising), and then applying the algebraic inverse transformation, does not produce the expected value of the original variable. A more technical introduction to the issue of bias and various inverses is presented in Section 2.3.5.

The traditional way of mitigating the issue of bias is to design an inverse VST that is asymptotically unbiased for large expected values. For the denoising of low-intensity data, this kind of an inverse is still highly inadequate. The core contribution of this thesis is to address this inadequacy by constructing a family of exact unbiased inverses, within the scope defined in the next section.

## 1.2 Objectives and scope of the thesis

The main objective of this thesis is to introduce the exact unbiased inverses for the Anscombe variance-stabilizing transformation and its generalization, aimed at Poisson noise and Poisson-Gaussian noise, respectively. Since the concept of exact unbiased inversion applies similarly to other distributions, this thesis also invokes the following question: Is it advantageous to go through the effort of designing separate denoising algorithms for each specific noise distribution, when comparable gains can be attained by perfecting the Gaussian denoising algorithms and addressing various noise distributions through variance stabilization together with a suitable inverse?

In this thesis, we consider the Anscombe transformation and its Poisson-Gaussian generalization, but it is fair to point out that there exist a variety of other variance-stabilizing transformations for Poisson noise, such as the Freeman-Tukey transformation [69], or the optimized ones discussed in [66] and [65]. However, within the scope of this thesis, we investigate the improvement gained through applying a suitable inverse, rather than the improvement gained through optimized variance stabilization. We chose to use the Anscombe transformation because it is in wide use, and because the discussion can then be naturally extended to the generalized Anscombe transformation for the Poisson-Gaussian case. Nevertheless, the proposed concepts could equally well be applied to an alternative transformation.

Even though this thesis is written from the viewpoint of digital imaging, the concepts and procedures presented here can be applied to data of any dimension, including 1-D signals and volumetric data. Moreover, we focus solely on grayscale images, but the use of the proposed inverses can be easily extended to colour image denoising.

## 1.3 Outline of the thesis

Chapter 2 introduces the different noise models considered in the thesis, provides both theoretical background and practical information regarding variance stabilization, and discusses a relevant selection of image denoising algorithms. Chapters 3–5 constitute the core of the thesis. Specifically, Chapter 3 discusses the optimal inversion of the Anscombe transformation for pure Poisson noise, presents the exact unbiased inverse and its closed-form approximation, and also considers a more elaborate minimum mean square inverse. Moreover, it provides thorough experimental results verifying the importance of proper inversion. Chapter 4 proceeds to the more general case of Poisson-Gaussian noise, constructing the exact unbiased inverse of the generalized Anscombe transformation, supporting it with

rigorous theoretical considerations, and again complementing these by performing extensive experiments. Chapter 5 contemplates the practical problem of estimating the parameters of the Poisson-Gaussian noise model, proposing a novel iterative parameter estimation algorithm based on variance stabilization. Finally, Chapter 6 contains the concluding remarks of the thesis.

### 1.3.1 Publications and author's contributions

The core contributions of Chapter 3 were first published in our papers [108], [110] and [112]. Similarly, Chapter 4 presents the results first published in our papers [113] and [114]. The author of the thesis is also the first author of all of these publications, responsible for the writing, practical implementations and the experiments. Alessandro Foi is to be thanked for the original concept of exact unbiased inversion, accurate computation of the $E\{f(z) \mid y\}$ values, the empirical SURE estimate in Section 3.3.2, and the core of the mathematical treatments in Section 4.3.3 and Section 5.2.

## 1.4 Notation and conventions

Various abbreviations are used throughout the thesis for convenience. These abbreviations and their respective meanings are collected into Table 1.1.

Further, the mathematical notation in the thesis adheres to the conventions outlined in Table 1.2.

| | |
|---|---|
| 1-D, 2-D, 3-D | One-Dimensional, Two-Dimensional, Three-Dimensional |
| AWGN | Additive White Gaussian Noise |
| BLS-GSM | Bayesian Least Square - Gaussian Scale Mixture (denoising algorithm [120]) |
| BM3D | Block Matching and 3-D Filtering (denoising algorithm [35]) |
| CCD | Charge-Coupled Device |
| CMOS | Complementary Metal-Oxide-Semiconductor |
| dB | Decibel |
| DCT | Discrete Cosine Transform |
| GAT | Generalized Anscombe Transformation |
| MAD | Mean Absolute Deviation |
| MC | Monte-Carlo |
| ML | Maximum Likelihood |
| MMSE | Minimum Mean Squared Error |
| MS-VST | Multiscale Variance-Stabilizing Transformation (denoising algorithm [146]) |
| MSE | Mean Squared Error |
| NMISE | Normalized Mean Integrated Squared Error |
| PH-HMT | Poisson-Haar Hidden Markov Tree (denoising algorithm [94]) |
| PSNR | Peak Signal-to-Noise Ratio (equation (3.20)) |
| PURE-LET | Poisson(-Gaussian) Unbiased Risk Estimate - Linear Expansion of Thresholds (denoising algorithms [98, 99]) |
| SAFIR | Structure Adaptive Filtering (denoising algorithm [15]) |
| SSIM | Structural Similarity Index Map [137] |
| SURE | Stein's Unbiased Risk Estimate [128] |
| VST | Variance-Stabilizing Transformation |

Table 1.1: A list of abbreviations frequently used in the thesis.

| | |
|---|---|
| $y$ | Noise-free data (the ideal unknown image) |
| $z$ | Data $y$ corrupted by noise (the observed data) |
| $y_i$, $z_i$ | Noise-free pixel and noisy pixel, respectively |
| $E\{z\}$ | Expected value of $z$ |
| $E\{z \mid y\}$ | Conditional expected value of $z$, conditioned by $y$ |
| $\operatorname{std}\{z\}$ | Standard deviation of $z$ |
| $\operatorname{var}\{z\}$ | Variance of $z$ |
| $\mu$ | Mean value of a Gaussian distribution |
| $\sigma$ | Standard deviation of a Gaussian distribution |
| $\mathcal{N}(\mu, \sigma^2)$ | Gaussian distribution with mean $\mu$ and standard deviation $\sigma$ |
| $n_i \sim \mathcal{N}(\mu, \sigma^2)$ | Random Gaussian variable $n_i$ is distributed according to $\mathcal{N}(\mu, \sigma^2)$ |
| $\alpha$ | Positive scaling factor for a Poisson distribution; gain |
| $\mathcal{P}(y_i)$ | Poisson distribution with mean (and variance) $y_i$ |
| $p_i$ | Random Poisson variable |
| $p(z)$ | Probability distribution of $z$ |
| $f$ | Forward variance-stabilizing transformation |
| $D$ | Denoised data before applying an inverse transformation; an approximation of $E\{f(z) \mid y\}$ |
| $\mathcal{I}_0$ | Exact unbiased inverse of the Anscombe transformation |
| $\mathcal{I}_\sigma$ | Exact unbiased inverse of the generalized Anscombe transformation, for parameter $\sigma$ |
| $\widetilde{\mathcal{I}}$ | Closed-form approximation of an inverse transformation $\mathcal{I}$ |
| $\mathcal{I}_{\text{asy}}$ | Asymptotically unbiased inverse transformation of $f$ |
| $\mathcal{I}_{\text{alg}}$ | Algebraic inverse transformation of $f$; $f^{-1}$ |
| $\mathcal{I}_{\text{ML}}$ | Maximum likelihood inverse transformation |
| $\mathcal{I}_{\text{MMSE}}$ | Minimum mean squared error inverse transformation |
| $\sigma_{\text{est}}$ | Estimated value of parameter $\sigma$ |

Table 1.2: A list of mathematical notation frequently used in the thesis.

# Chapter 2

# Background

## 2.1 Noise sources in digital imaging

CCD and CMOS sensors function very similarly when it comes to capturing the incoming light. In both cases, as photons hit the photosensitive silicon layer of a diode, a proportional amount of electrons are released from the silicon (assuming the frequency, and thus energy, of the incoming photons exceed the threshold required for the photoelectric effect [75, 54] to take place). The released electrons are then trapped by a potential well and thus accumulated into electrical charge in a capacitor. On the other hand, there is a major difference in how these competing technologies further convert this charge into voltage. In CCD, the charges are read out line by line. Specifically, the contents of each capacitor is transferred to its neighbour, until it reaches the end of the line, where an amplifier converts the charge into voltage. In contrast, in a CMOS sensor each pixel is read out using an amplifier dedicated to that particular pixel (which is why it is also called an active pixel sensor). In any case, the analogue voltages are then discretized into a 2-D array of numbers representing the intensity of light at each pixel location.

This capturing process is subject to various errors. We can divide the noise sources into two main categories: those causing signal-dependent noise, and those causing signal-independent noise. In particular, the most significant source of signal-dependent noise is the uncertainty of random photon emission and sensing. As mentioned in Section 1.1, even with a light source of constant intensity, there is fluctuation in the number of photons hitting the imaging sensor during a constant time interval (exposure time). For instance, even if we capture ten images in rapid succession under constant illumination, each time using the same exposure time, the number of photons hitting the same location (pixel) in the sensor will vary. This random variation can be modelled by a Poisson distribution [105].

In addition to the randomness of photon emission, the sensor is unable to convert all arriving photons to electrons. The ratio between the number of photons successfully converted to electrical charge, and the total number of photons reaching the sensor, is called the quantum efficiency of the sensor. This percentage depends on the wavelength of the incident light, due to wavelength-dependent absorption characteristics of the silicon-based substances. As a result of this and

various other physical and chemical contributing factors, the quantum efficiency of a standard CCD typically peaks at around 50 % for the red wavelengths, and gradually reduces as the wavelength increases or decreases. For a more thorough discussion about the topic, see, e.g., [125].

Besides the noise directly related to the randomness of the photon behaviour, there are various other noise sources involved in the image capturing chain, arising from the physical properties and imperfections of the imaging hardware. These degradations include thermal noise, dark current shot noise, flicker ($1/f$) noise, readout noise, and several types of fixed-pattern noise.

Thermal noise, also known as Johnson-Nyquist noise, is the statistical fluctuation of electric charge created by thermal agitation [79, 118]. It is unavoidably present in all conductive material (except theoretically at absolute zero temperature, when there is no motion of electrons). When sufficient heat causes these thermal electrons to be freed from the valence band of the conductor, they can be captured by the potential well of a pixel [76]. Thus, the capacitors of the pixels accumulate some charge even when no light enters the sensor (i.e., when the shutter is closed). This level of charge is also called dark current, and the statistical fluctuations within it are known as dark current shot noise (or dark signal non-uniformity, DSNU).

Since dark current depends on the ambient temperature, it can be reduced significantly by cooling down the sensor. This approach is used for example in X-ray imaging [144], neutron radiography [89] and astronomical imaging [76]. A CCD can be cooled even down to -60. . .-100 °C (e.g., with liquid nitrogen), causing the number of thermal electrons per pixel per second to drop from about 25000 (in room temperature) to 0.04–2 [76]. However, even such low amounts of electrons per second are not completely insignificant, because imaging distant astronomical targets usually requires very long exposure times.

Flicker noise ($1/f$ noise, or pink noise) is electronic noise with a pink power spectral density, i.e., the power is inversely proportional to the frequency $f$ of the signal. Hence, flicker noise is most prevalent in the low frequency portion of the signal, but it can be reduced by, e.g., autozeroing and correlated double sampling methods [56]. Its general properties are not perfectly understood, but it typically manifests itself as fluctuation of current (or equivalently, as fluctuation of voltage or resistance). The detailed mechanisms of flicker noise are discussed extensively in [140].

Readout noise is statistical variation in the final readout and discretized intensity values, caused by imperfect electronics of the amplifiers and the analogue-to-digital converter. Fortunately, the effect of this noise is very modest in current CCDs, at most 10 electrons per pixel per read [76]. Since the amplification (gain) throughout the sensor is not exactly constant, this phenomenon is also called photo response non-uniformity (PRNU). It is a type of fixed-pattern noise, meaning it is temporally independent. Other fixed-pattern defects include hot and cold pixels (defective pixels that produce respectively a higher or lower response than expected), the extreme cases of which are stuck pixels being permanently "on" and dead pixels being permanently "off".

Fixed-pattern noise can be compensated via flat-field correction, where the pixel gains and the level of dark current are determined, and then used for cor-

recting the captured image. In particular, a dark frame is produced by capturing an image without actually exposing the shutter to any light. Then the dark frame, which in practise contains noise caused by the dark current and other non-uniformities, can be subtracted from the captured image.

Finally, it is worth mentioning that the captured and discretized raw data often goes through some kind of a post-processing chain inside the imaging device. This chain typically involves demosaicking, denoising, white balancing, gamma correction and JPEG compression, some of which may introduce further artifacts to the final post-processed image.

## 2.2 Noise models

### 2.2.1 Additive white Gaussian noise

As discussed in Section 2.1, there are a multitude of noise sources degrading the quality of a captured digital image, most of which are already compensated to some extent by a careful design of the imaging device and the processing within. Taking each noise source separately into account is usually not feasible in the design of denoising algorithms, so let us recall an important result of probability theory. The Central Limit Theorem (CLT) was defined generally by A. Lyapunov in 1901, but both de Moivre and Laplace have already known it in some form in the previous century [132].

CLT, in Lyapunov's form, goes as follows. Let $X_1, \ldots, X_k$ be $k$ independent random variables with respective mean values $\mu_i$ and variances $\sigma_i^2$. The Lyapunov condition is defined as

$$\lim_{k\to\infty} \frac{1}{s_k^{2+\delta}} \sum_{i=1}^{k} E\left\{|X_i - \mu_i|^{2+\delta}\right\} = 0, \tag{2.1}$$

where $s_k^2 = \sum_{i=1}^{k} \sigma_i^2$. If (2.1) holds for some $\delta > 0$, the normalized sum of the random variables converges in distribution to the standard normal distribution as $k \to \infty$. In other words,

$$\frac{1}{s_k} \sum_{i=1}^{k} (X_i - \mu_i) \to \mathcal{N}(0, 1), \tag{2.2}$$

when $k \to \infty$. This fundamental property suggests that we may try to approximate the combined effect of the various noise sources by a Gaussian distribution.

In image processing, a common stronger assumption is that the noise is not only Gaussian, but additive white Gaussian noise (AWGN); it is signal-independent, normally distributed, independent and identically distributed (the latter implying that $\sigma_i$ is constant over the image), and has a flat power spectrum. In the notation used throughout this thesis, the AWGN model is expressed simply as

$$z = y + n, \tag{2.3}$$

where $z$ is the observed noisy image, $y$ is the underlying (unknown) noise-free image, and the additive term $n$ is the corrupting AWGN noise. In the real-world

digital imaging devices, this is often not a particularly accurate representation of the truth, as it completely ignores the signal-dependent shot noise, which typically contributes to the noise significantly. However, most denoising algorithms traditionally make this simplifying assumption in order to make the design and implementation of the algorithm more accessible and straightforward. On the other hand, there are cases when the assumption of Gaussianity is more justified. For instance, in low-dose computed tomography (CT), the data can be considered Gaussian, although with signal-dependent variance [95].

### 2.2.2 Poisson noise

Instead of assuming the noise to be AWGN, another common approach is to only consider the signal-dependent noise, and thus implicitly assume that the degrading effects of all signal-independent sources are insignificant in comparison. We start off with this assumption in Chapter 3, and then extend the developed inversion method to the more general case of Poisson-Gaussian noise in Chapter 4.

Thus, let us properly define what we mean by Poisson noise. Let $z_i, i = 1, \ldots, N$, be the observed pixel values obtained through an image acquisition device. We consider each $z_i$ to be an independent random Poisson variable whose mean $y_i \geq 0$ is the underlying intensity value to be estimated. Explicitly, the discrete Poisson probability of each $z_i$ is

$$P\left(z_i \mid y_i\right) = \frac{y_i^{z_i} e^{-y_i}}{z_i!}. \tag{2.4}$$

In addition to being the mean of the Poisson variable $z_i$, the parameter $y_i$ is also its variance:

$$E\{z_i \mid y_i\} = y_i = \operatorname{var}\{z_i \mid y_i\}. \tag{2.5}$$

Now, Poisson noise can be formally defined as

$$\eta_i = z_i - E\{z_i \mid y_i\}, \tag{2.6}$$

and thus, we trivially have $E\{\eta_i \mid y_i\} = 0$ and $\operatorname{var}\{\eta_i \mid y_i\} = \operatorname{var}\{z_i \mid y_i\} = y_i$. Since the noise variance depends on the true intensity value, Poisson noise is signal dependent. More specifically, the standard deviation of the noise $\eta_i$ equals $\sqrt{y_i}$. Due to this, the effect of Poisson noise increases (i.e., the signal-to-noise ratio decreases) as the intensity value decreases.

### 2.2.3 Poisson-Gaussian noise

As a natural extension of the previous two models, we consider the Poisson-Gaussian noise model, which takes into account both signal-dependent and signal-independent noise sources. In particular, the model contains a multiplicative scaled Poisson term, and an additive Gaussian term.

Specifically, let $z_i, i = 1, \ldots, N$ be again the observed pixel values obtained through an image acquisition device. In the Poisson-Gaussian model, each $z_i$ is a realization of an independent random Poisson variable $p_i$ with an underlying mean value $y_i$. This variable is then scaled by a positive gain value $\alpha > 0$, and

the scaled result is further corrupted by additive white Gaussian noise $n_i$ of mean $\mu$ and standard deviation $\sigma$. In other words,

$$z_i = \alpha p_i + n_i, \tag{2.7}$$

where $p_i \sim \mathcal{P}(y_i)$ and $n_i \sim \mathcal{N}(\mu, \sigma^2)$. Thus, we can formally define Poisson-Gaussian noise as

$$\eta_i = z_i - \alpha y_i. \tag{2.8}$$

The problem of denoising an image corrupted by Poisson-Gaussian noise is then equivalent to estimating the underlying noise-free image $y$ given the noisy observations $z$.

It is important to mention that this model can still be elaborated in order to more accurately match the physical characteristics and limitations of the imaging hardware. Notably, (2.7) does not take clipping (over- and underexposure) into account. However, the dynamic range of an actual imaging sensor is limited, because each pixel can accumulate only a certain amount of charge. Thus, over- or underexposing an image causes additional nonlinear behaviour in the image model. For simplicity, we will not consider the clipped case here, but we note that a clipped Poisson-Gaussian model has been successfully used, e.g., in noise fitting and denoising of clipped and non-clipped raw CCD data [64, 67].

## 2.3 Variance stabilization

This section explores the theoretical and practical results related to variance stabilization, accompanied with a historical overview on the development of VSTs. We discuss the stabilization of Poisson and Poisson-Gaussian noise in detail, and elucidate the importance of a proper inverse VST.

As mentioned earlier, when it comes to denoising, our focus is in the indirect approach that combines variance stabilization with an AWGN denoising algorithm. A review of some of these denoising algorithms follows in Section 2.4.1, whereas in Sections 2.4.2–2.4.3, we discuss a few effective denoising algorithms designed specifically for signal-dependent noise.

### 2.3.1 Background and motivation

The rationale behind applying a variance-stabilizing transformation to random data $z_i$, $i = 1, \ldots, N$ is to remove the data-dependence of the noise variance, so that it becomes constant throughout the whole data. The stabilizing transformation may also be normalizing (i.e., it results in a Gaussian noise distribution). Generally, this enables an easier analysis and processing of the data, and it has been a well-known tool in statistics for a long time, even though we are mainly interested in it from the perspective of signal and image processing. From our point of view, these properties of the VST allow us to estimate the intensity values $y_i$ with a conventional denoising method designed for additive white Gaussian noise.

The history and development of various VSTs is discussed in Section 2.3.2, but in order to properly review the history, we first derive a well-known heuristic used in the traditional design of VSTs. Let us investigate the removal of the

data-dependence for a general random variable $z$ with mean $E\{z \mid y\} = \mu(y)$ and standard deviation $\operatorname{std}\{z \mid y\} = \sigma(y)$. The ideal variance-stabilizing transformation $f(z)$ ought to be such that $\operatorname{std}\{f(z) \mid y\} = c$, where $c > 0$ is a constant (i.e., the standard deviation, and thus variance, of $f(z)$ is independent of $y$).

By approximating $f(z)$ with its first order Taylor polynomial centered at $\mu$, we get

$$f(z) \approx f(\mu) + (z - \mu)\frac{\partial f}{\partial z}\mu, \tag{2.9}$$

and thus,

$$(f(z) - f(\mu))^2 \approx (z - \mu)^2 \left(\frac{\partial f}{\partial z}\mu\right)^2. \tag{2.10}$$

If we now take the expected value of both sides of (2.10), we have

$$\operatorname{var}\{f(z) \mid y\} = E\left\{(f(z) - f(\mu))^2\right\} \approx E\left\{(z - \mu)^2 \left(\frac{\partial f}{\partial z}\mu\right)^2\right\} = \sigma(y)^2 \left(\frac{\partial f}{\partial z}\mu\right)^2. \tag{2.11}$$

In other words,

$$\operatorname{std}\{f(z) \mid y\} = \sigma(y) \cdot \frac{\partial f}{\partial z}\mu. \tag{2.12}$$

Let us now set a constraint $\operatorname{std}\{f(z) \mid y\} = c$, where $c > 0$ is constant (typically we set $c = 1$). Then, solving (2.12) for $f(z)$ yields

$$f(z) = \int_z \frac{c}{\sigma(y)}\, d\mu(y). \tag{2.13}$$

This is a traditional starting point for the design of practical variance-stabilizing transformations for specific distributions. When the characteristics of the distribution (i.e., $\sigma(y)$ and $\mu(y)$) are known, $f(z)$ is easily obtained by (2.13), as demonstrated in Section 2.3.3. However, it is worth noting that the VST obtained by this procedure is typically suboptimal, and in practise more refined transformations are used.

### 2.3.2 History and development of variance stabilization

The statistical term *variance* has been used at least since 1918, when R. A. Fisher coined the term and defined it as the square of the standard deviation. He found the concept useful in investigating how two independent causes of variability (with standard deviations $\sigma_1$ and $\sigma_2$) act together in Mendelian inheritance; the resulting distribution had a standard deviation equalling $\sqrt{\sigma_1^2 + \sigma_2^2}$ [60].

The idea of rendering the distribution of heteroskedastic data into another form for easier processing or analysis (in other words, the idea of variance stabilization) can be reliably traced back at least to 1930s, but it probably has been known in some form even earlier. In particular, the heuristic rule (2.13) was used by L. H. C. Tippett in 1934, when he applied binomial and Poisson distributions in the statistical treatment of certain random occurrence data in textile industry [134]. However, as the derivation of (2.13) is rather simple, it is not unlikely that it has been independently discovered by a number of people. In the following

years, the topic continued to gain interest, and various articles were published throughout the late 1930s and 1940s (e.g., [9, 143, 13, 30, 27, 31, 11, 33, 3]).

In particular, M. S. Bartlett proposed the transformation $f(z) = 2\sqrt{z + 1/2}$ in 1936 for the stabilization of a Poisson variable, also using (2.13) for the derivation [9]. In 1948, this was improved by F. J. Anscombe to $f(z) = 2\sqrt{z + \frac{3}{8}}$ [3], providing optimal (as defined in Section 2.3.3) asymptotic stabilization; this is the Anscombe transformation that plays a central role also in this thesis. Shortly thereafter, M. Freeman and J. Tukey introduced an alternative transformation $f(z) = \sqrt{z} + \sqrt{z + 1}$ [69], now known as the Freeman-Tukey transformation. Their design principle differed from that of Anscombe, as they emphasized on keeping the error small for low mean values, at the cost of slower asymptotic convergence. The stabilization of Poisson noise is discussed in more detail in Section 2.3.3, where the concepts of this paragraph are elaborated.

Concerning the variance stabilization of general distribution families, we should mention the important and well-known results of J. H. Curtiss, B. Efron, and R. Tibshirani. First, in 1943, Curtiss recognized the lack of general mathematical results in the earlier publications. Thus, in [33], he fills this gap by proving various asymptotic stabilization and normalization properties of general VSTs. In addition, he explores the consequences of his theorems for Poisson and binomial distributions, also confirming the viability of the square root transformation $2\sqrt{z + \text{constant}}$ for the Poisson case (and the viability of the inverse sine transformation for the binomial case). However, he could not determine an optimal value for this constant, or determine the degree of convergence to normality of this type of transformation; as mentioned, the optimal constant 3/8 was then proposed by Anscombe in 1948.

The question of normalizing a general one-parameter family of distributions, and in particular the degree of normalization, was then answered by Efron in 1981 [52]. He derives the conditions under which a family of distributions can be exactly normalized by a transformation, and constructs a measure for the degree of normalization. Moreover, he also provides a way to actually construct the stabilizing and normalizing transformations in practise. Consequentially, these results show that for the Poisson distribution (and for many other distribution families), a perfect transformation (exactly stabilizing the variance or exactly normalizing the distribution) does not exist. Therefore, in practise, approximate or asymptotic results are employed. Curtiss also discusses the discrepancy between stabilizing and normalizing transformations; in general, good stabilization and good normalization are not always mutually compatible properties. For instance, he notes that in the Poisson case, the Anscombe transformation $f(z) = 2\sqrt{z + \frac{3}{8}}$ provides good stabilization, but on the other hand it goes 37 % beyond the ideal normalizer $f(z) = 2\sqrt{z + \frac{1}{3}}$ (in terms of his "strength of transformation" criterion). The latter transformation, with better normalizing properties, was first suggested by Anscombe in 1953 [4], based on the fact that it makes the skewness (i.e., third centered moment) of the resulting distribution nearly zero. Curtiss also argues that both of these transformations stabilize the variance equally well, but despite this observation, the better normalizing transformation has not gained widespread usage in applications, as opposed to the traditional Anscombe transformation. Based

on the results of Curtiss, S. K. Bar-Lev and P. Enis have also proposed a method of using a given normalizing transformation to construct a stabilizing transformation [7], and discussed stabilizing and normalizing transformations for a compound Poisson process [8].

The contribution of Tibshirani is of a different kind, but before discussing that, let us introduce two other methods. In 1964, G. E. P. Box and D. R. Cox proposed a stabilizing power transformation family of the form $z^\lambda = (z^\lambda - 1)/\lambda$ (and another slightly more refined family), where the parameter $\lambda$ defines a specific transformation [16]. They devised these transformations with a particular interest in the linear regression of biological data. Then, in 1985, L. Breiman and J. H. Friedman introduced the two-step iterative ACE (alternating conditional expectation algorithm) for constructing optimal transformations for regression and correlation [18]. In particular, they consider two random variables $z$ and $w$. Their algorithm finds two transformations $f(z)$ and $s(w)$, where $f$ serves as a VST, and $s$ represents a mean function with the goal of satisfying $s(w) = E\{f(z) \mid w\}$. In practise, they alternate between computing $s(w) = E\{f(z) \mid w\}$ and $f(z) = E\{s(w) \mid z\} \cdot \operatorname{var}\{E\{s(w) \mid z\}\}^{-1/2}$, where each conditional expectation is estimated through realizations of the random variables. To put it differently, the goal of ACE is to minimize $E\{f(z) - s(w)\}^2$ and maximize $\operatorname{corr}\{f(z), s(w)\}$, both subject to the condition $\operatorname{var}\{f(z)\} = 1$. As a result, the obtained transformations can be used to regress $z$ on $w$, or vice versa.

Wishing to construct a powerful iterative algorithm like ACE, but specifically targeting regression like the parametric Box-Cox transformations, Tibshirani proposed the nonparametric AVAS (additivity and variance stabilization) procedure in 1988 [131]. It also finds the two transformations $f(z)$ and $s(w)$ by two alternating steps, again with the goal of satisfying $s(w) = E\{f(z) \mid w\}$ and $\operatorname{var}\{f(z) \mid s(w)\} = \text{constant}$. This is accomplished by alternatively computing the mean $s(w) = E\{f(z) \mid w\}$, and updating the VST $f(z)$ by using the knowledge that $h(f(z))$ should have a more constant variance than $f(z)$, where $h$ is another VST obtained by the simple integral heuristic (2.13). Tibshirani demonstrates the usage of AVAS in various regression scenarios, and discusses its convergence with the help of Efron's categorization of distribution families.

Let us mention some of the more recent advances, inspired not only by statistical problems, but also by the growing field of signal and image processing. In particular, a notable approach is to stabilize the wavelet coefficients of the noisy data instead of the data itself. For instance, the Haar-Fisz transform [70] stabilizes Poisson data by first computing the discrete Haar wavelet transform, producing detail coefficients $d_i$ and smooth coefficients $s_i$. Then, the stabilized coefficients are obtained by $d_i/\sqrt{s_i}$, following the asymptotic normality observation by Fisz in [61]. The data-driven Haar-Fisz Transform (DDHFT) [71] is a multiscale generalization of the Haar-Fisz transform, stabilizing the variance of sequences of non-negative independent random variables, where the variance is an unknown non-decreasing function of the expectation. As another example, the stabilization procedure of the MS-VST [146] algorithm is based on combining the Anscombe transformation with a filter bank of wavelets, ridgelets or curvelets (see Section 2.4.2 for a more detailed description of MS-VST).

Finally, the optimization of variance-stabilizing transformations was recently

formulated as an explicit optimization problem by A. Foi [65, 66]. The discrepancy between the standard deviation of the stabilized variable and the target constant value is measured by a nonlinear stabilization functional, which is then iteratively minimized. The minimization can be done either by recursive stabilization (improving on the method used in AVAS), or by direct search. The nonparametric optimized transformations obtained with this procedure are shown to be state-of-the-art [66]. Moreover, the method also allows to relax the typical assumption of monotonicity of the VST $f$, as long as the mapping $E\{z \mid y\} \to E\{f(z) \mid y\}$ remains invertible.

### 2.3.3 Stabilizing Poisson noise

For a Poisson distributed random variable $z$, we have $E\{z \mid y\} = \mu(y) = y$ and $\operatorname{std}\{z \mid y\} = \sigma(y) = \sqrt{y}$. Assuming we wish to obtain a VST that renders the variance to unity (i.e., $c = 1$), (2.13) leads to the root transformation

$$f(z) = \int_z \frac{1}{\sqrt{y}}\, dy = 2\sqrt{z}, \tag{2.14}$$

where the arbitrary constant of integration has been set to zero. As the stabilization obtained through (2.14) is not particularly good especially for low values of $y$ (see Figure 2.1), improvements have been sought by investigating a transformation of the form $f(z) = 2\sqrt{z+b}$, where $b$ is a constant. Specifically, Bartlett proposed the transformation $f(z) = 2\sqrt{z+1/2}$ in 1936 [9], and Anscombe improved this to

$$f(z) = 2\sqrt{z + \frac{3}{8}}. \tag{2.15}$$

in 1948. In fact, setting the constant to 3/8 in (2.15) provides optimal stabilization for large values of $y$ for this type of root transformation [3]. More explicitly, the second order Taylor expansion for the stabilized variance equals

$$\operatorname{var}\{f(z) \mid y\} = 1 + \frac{3-8b}{8y} + \frac{32b^2 - 52b + 17}{32y^2}. \tag{2.16}$$

Hence, setting $b = 3/8$ yields

$$\operatorname{var}\{f(z) \mid y\} = 1 + \frac{1}{16y^2}. \tag{2.17}$$

Anscombe credits this result to A.H.L. Johnson, but (2.15) is nevertheless known as the Anscombe transformation. This is one of the most popular VSTs for Poisson distributed data, although the Freeman-Tukey transformation $f(z) = \sqrt{z} + \sqrt{z+1}$ [69] provides comparable asymptotic stabilization. In particular, both the Anscombe and Freeman-Tukey transformations yield an asymptotic stabilized variance of the order $1 + \mathcal{O}(y^{-2})$ for $y \to \infty$, in comparison to $1 + \mathcal{O}(y^{-1})$ for the simple transformation (2.14).

Applying any of the above variance-stabilizing transformations to Poisson distributed data gives a signal whose noise is approximately additive normal with

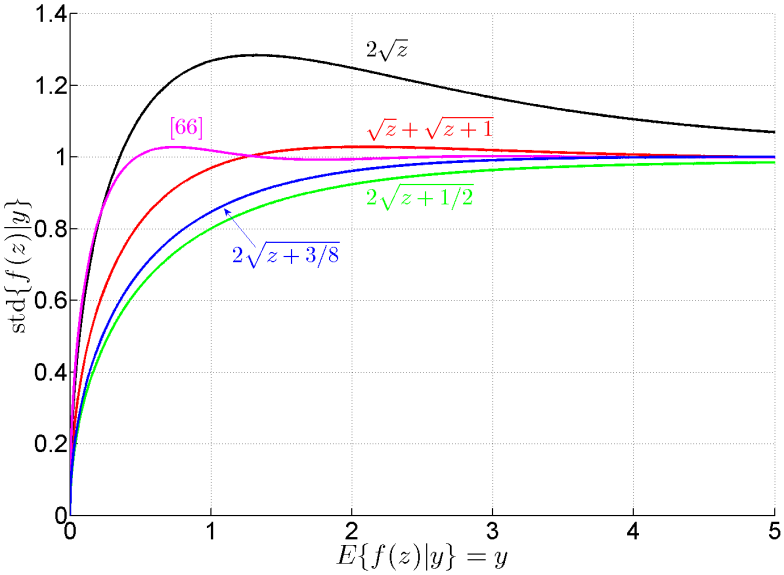



Figure 2.1: Standard deviation of the stabilized Poisson variable $f(z)$, for five different VSTs: the heuristic transformation $f(z) = 2\sqrt{z}$, the Bartlett transformation $f(z) = 2\sqrt{z+1/2}$, the Freeman-Tukey transformation $f(z) = \sqrt{z} + \sqrt{z+1}$, the Anscombe transformation $f(z) = 2\sqrt{z+\frac{3}{8}}$, and a state-of-the-art nonparametric optimized transformation [66] (nonmonotone, direct search).

unitary variance. However, as mentioned in Section 2.3.2, it is not possible to achieve exact stabilization or exact normalization for Poisson distributed data. Thus, all practical transformations are approximately or asymptotically stabilizing and normalizing. The stabilized standard deviation is illustrated in Figure 2.1 for the heuristic transformation $f(z) = 2\sqrt{z}$, the Bartlett transformation $f(z) = 2\sqrt{z+1/2}$, the Freeman-Tukey transformation $f(z) = \sqrt{z} + \sqrt{z+1}$, the Anscombe transformation $f(z) = 2\sqrt{z+\frac{3}{8}}$, and a state-of-the-art nonparametric optimized transformation [66] (nonmonotone, direct search). This comparison shows, for instance, that the Freeman-Tukey transformation stabilizes the variance slightly better for small values ($y < 2.2$) than the Anscombe transformation. On the other hand, the Anscombe transformation provides slightly more accurate stabilization around $2.2 < y < 4.8$. Another noticeable difference in the stabilized variances is that there is some overshoot for the Freeman-Tukey transformation before it approaches the target variance 1 from above, whereas such oscillation does not take place with the Anscombe transformation. In this thesis, our focus is on the Anscombe transformation due to its widespread use in practical applications. In addition, this choice allows us to generalize our results for the Poisson-Gaussian noise, using the generalized Anscombe transformation introduced below; we are not aware of a Poisson-Gaussian generalization for the Freeman-Tukey transformation, for instance.

### 2.3.4 Stabilizing Poisson-Gaussian noise

As explained in Section 2.2.1, the various signal-independent noise sources of a digital acquisition device are typically jointly approximated as Gaussian noise. Combining this with the Poisson noise component produces the Poisson-Gaussian model (2.7), for which the Anscombe transformation is no longer appropriate. In order to address this deficiency, F. Murtagh et al. derived a generalization of the Anscombe transformation, taking into account the three necessary parameters: gain $\alpha$, Gaussian mean $\mu$, and Gaussian standard deviation $\sigma$ [107, 126]. Much of their derivation follows the structure used by Anscombe himself in the derivation of (2.15).

In particular, assuming the noisy data $z$ is distributed according to (2.7), [126] defines the generalized Anscombe transformation (GAT) as

$$f(z) = \begin{cases} \frac{2}{\alpha}\sqrt{\alpha z + \frac{3}{8}\alpha^2 + \sigma^2 - \alpha\mu}, & z > -\frac{3}{8}\alpha - \frac{\sigma^2}{\alpha} + \mu \\ 0, & z \leq -\frac{3}{8}\alpha - \frac{\sigma^2}{\alpha} + \mu. \end{cases} \tag{2.18}$$

Hence, applying (2.18) to $z$ (approximately) stabilizes its variance to unity. More precisely, $\operatorname{var}\{f(z)|y,\sigma\} = 1 + \mathcal{O}(y^{-2})$ for $y \to \infty$, since (2.18) is derived by zeroing the first order term in the Taylor expansion of the stabilized variance [126, p. 265]. This derivation is analogous to how the optimal constant $b = 3/8$ was determined for the traditional Anscombe transformation (2.15). Note that for the pure Poisson case (i.e., $\alpha = 1$, $\sigma = 0$, and $\mu = 0$), the GAT (2.18) coincides with (2.15), as expected.

### 2.3.5 Inverse variance-stabilizing transformations

Especially in the practical context of denoising with the help of variance stabilization, we need not only a properly constructed forward VST in order to achieve good results, but also a suitable inverse transformation; this issue was first discussed in terms of exact unbiased inversion in [67]. Specifically, the denoising of the variance-stabilized data $f(z)$ produces a signal $D$ that can be considered as an estimate of $E\{f(z) \mid y\}$. Then, we need to apply an inverse transformation to $D$ in order to obtain the desired estimate of the underlying noise-free data $y$.

Let us consider the case of pure Poisson noise. The direct algebraic inverse of the Anscombe transformation (2.15) is

$$\mathcal{I}_{\text{alg}}(D) = f^{-1}(D) = \left(\frac{D}{2}\right)^2 - \frac{3}{8}, \tag{2.19}$$

but the resulting estimate of $y$ is biased, because the nonlinearity of the transformation $f$ means we generally have

$$E\{f(z) \mid y\} \neq f(E\{z \mid y\}), \tag{2.20}$$

and, thus,

$$f^{-1}(E\{f(z) \mid y\}) \neq E\{z \mid y\}. \tag{2.21}$$

Another possibility is to use the adjusted inverse [3]

$$\mathcal{I}_{\text{asy}}(D) = \left(\frac{D}{2}\right)^2 - \frac{1}{8}, \tag{2.22}$$

which provides asymptotic unbiasedness for large values of $y$. This is the inverse typically used in applications, but as its name implies, it does not guarantee unbiased results for low-intensity data (low values of $y$, corresponding to the case of low signal-to-noise ratio). Thus, Chapter 3 of this thesis presents an exact unbiased inverse Anscombe transformation that is equally applicable regardless of the values of $y$.

Naturally, the same reasoning applies also for the case of Poisson-Gaussian noise and the GAT. In Chapter 4, we generalize the proposed exact unbiased inverse of the Anscombe transformation into the exact unbiased inverse of the GAT. Moreover, the algebraic and asymptotically unbiased inverses of the GAT are respectively obtained from (2.19) and (2.22) by simply adding the term $-\sigma^2$. Note that these two inverses are denoted by $\mathcal{I}_{\text{alg}}$ and $\mathcal{I}_{\text{asy}}$ in both the Poisson case (Chapter 3) and the Poisson-Gaussian case (Chapter 4). However, in the latter case, the corresponding equations of $\mathcal{I}_{\text{alg}}$ and $\mathcal{I}_{\text{asy}}$ are also given explicitly, so there should be no risk of confusion.

## 2.4 A short overview of image denoising algorithms

Let us briefly review the image denoising algorithms used in the thesis, along with a selection of other algorithms to provide more perspective to the topic. However, as our main focus is on the variance stabilization aspect of image denoising, a detailed survey of the various types of denoising algorithms is out of our scope. For a more extensive summary and categorization of AWGN denoising algorithms, we point the reader to [41, Chapter 3], [82] and the references therein.

Denoising algorithms can be categorized, for instance, by whether they operate in spatial domain or transform domain, or by whether they operate locally or nonlocally, or by whether they estimate only one pixel or a larger patch at a time (see [82, Table 1]). However, for our purposes, it is more natural to divide them primarily according to the targeted noise model: AWGN, Poisson, and Poisson-Gaussian denoising algorithms. Note that some of the AWGN categories are overlapping, as there exist nonlocal transform domain algorithms, local transform domain algorithms, and so forth. Thus, the rough subdivision in Section 2.4.1 should not be taken strictly; in many cases, the introduced algorithms belong to more than one of the categories.

### 2.4.1 AWGN denoising algorithms

#### Local polynomial approximation

Early efforts at estimating the expected value of noisy data were made in 1964 by È.A. Nadaraya [115] and G.S. Watson [138]. With the Nadaraya-Watson estimator, each pixel is estimated by a locally weighted average of the neighbouring pixels. In its simplest form, this kind of *kernel regression* assumes a uniform local

neighbourhood (i.e., all the neighbouring pixels have the same intensity). A natural extension of this idea is to relax the uniformity assumption and model the intensities in a local neighbourhood by a polynomial of a certain degree (see, e.g., [28, 29, 58, 81]). Hence, this method is often called *local polynomial approximation* (LPA).

The LPA method can be further extended by considering an adaptive kernel, in other words letting the weights vary based on the data present in the neighbourhood. This can be accomplished, for instance, by varying the bandwidth of the kernel; see, e.g., [124, 80, 57, 81]. Since images typically contain sharp discontinuities (such as edges), [81] considers also combining several directional LPA estimates in order to better adapt to the image structure; this is known as anisotropic LPA-ICI, where the kernel bandwidth is selected by an intersection of confidence intervals (ICI).

#### Transform-domain algorithms

In general terms, transform-domain denoising is a rather straightforward process: First, the image is transformed into a chosen domain, where the signal can typically be represented more sparsely (i.e., with fewer significant coefficients). Then, the transform domain coefficients are processed in order to attenuate the noise. Finally, the denoised result is obtained by applying an inverse transform. Traditional transforms include the discrete fourier transform (DFT) and the discrete cosine transform (DCT), both of which are *decorrelating* transforms. In other words, the energy of many coefficients is compacted into fewer significant transform domain coefficients. This enables us to threshold (or otherwise modify) the coefficients, thus attenuating the noise without destroying the essential information of the signal. In order to take full advantage of the local correlation between the neighbouring pixels and achieve good energy compaction, all of this processing is typically done separately for small (e.g., $8 \times 8$ or $16 \times 16$) image blocks. This technique is also used in image compression, such as in the lossy JPEG standard.

Another perspective to denoising is offered by Bayesian statistics, where we compute a conditional posterior probability with Bayes' theorem, based on a priori assumptions about the probability densities of the image and noise. An estimate of the noise-free image can then be obtained for example by computing the conditional expectation of the posterior probability, or by finding the value that maximizes the posterior probability. For an extensive and up to date introduction on Bayesian statistics, we refer the reader to [93]. Even though the Bayesian theory is well established, a proper selection of the prior assumptions still remains a major practical challenge (see, e.g., [77]).

One of the denoising algorithms featured throughout this thesis is the BLS-GSM (Bayesian Least Square - Gaussian Scale Mixture) algorithm, proposed by J. Portilla et al. in 2003 [120]. As implied, BLS-GSM performs the denoising in transform domain instead of the spatial domain. The algorithm constructs an overcomplete multiscale oriented basis, and models the local neighbourhoods of these multiscale transform domain coefficients with a Gaussian scale mixture model [2]. In particular, the model states that a vector is a mixture of Gaussian vectors (i.e. GSM), if and only if it is a product of a Gaussian vector and an independent

random scalar multiplier $M > 0$. For the prior density of this multiplier, they have various approaches: a logarithmic prior, a maximum likelihood estimate using the neighbouring vectors, and a noninformative prior $p(M) \sim 1/M$ also known as Jeffrey's prior [59]. Based on its simplicity and the good associated denoising results, they favour the noninformative prior in their experiments.

Multiscale transforms have gained much interest in the field of image processing, with wavelets being a particularly popular subclass of these transforms. In simple terms, wavelets are families of oscillating functions, where each family is generated from one "mother wavelet" by dilations and translations. This wavelet family, which typically forms an orthonormal basis dictionary or an overcomplete dictionary, can then be used to decompose the image into a multiscale representation. As a mathematical construct, the rectangular Haar wavelet was introduced already in 1910 [73], but wavelets began to emerge in the context of image processing (along with the word "wavelet" itself) in the 1980s. The early and important works of that era include those of I. Daubechies [45] and S. Mallat [103, 104]. The wavelet mindset has also inspired many related transforms, which attempt to adapt to specific geometrical image features, such as edges, straight lines or smooth curves. These transforms include directional wavelets [5], complex wavelets [86, 87], curvelets [22, 127], ridgelets [21, 127], wedgelets [50], shearlets [92] and contourlets [48], among others.

Shape adaptivity has also been considered in the SA-DCT algorithm [62, 63], where the support of the 2-D DCT transform is adapted according to the image structure. Specifically, the neighbourhood of the pixel is divided radially into sectors, and the radial length of each sector is determined with the anisotropic LPA-ICI method. These variable length sectors form the neighbourhood, for which the 2-D DCT is applied. The transformed coefficients are then hard-thresholded, after which the inverse 2-D DCT is applied. Note that since the neighbourhoods for different pixels are typically overlapping, the denoised pixels form an overcomplete representation. Thus, the individual estimates are combined by computing their weighted average, where the weights are inversely proportional to the mean variance of the estimates.

Another recent adaptive transform-domain denoising algorithm is K-SVD [55], which aims at representing image blocks as a sparse linear combination of atoms (elements of an overcomplete dictionary). The algorithm works iteratively, alternating between representing the image as a linear combination of the current atoms, and updating the dictionary atoms. In particular, the former step utilizes the K-means clustering algorithm, and the latter step utilizes the singular value decompositon (SVD). The dictionary can be computed beforehand by using an image database, or on the fly from the actual noisy image. The K-SVD algorithm has also been extended to take advantage of a multiscale representation, in the form of MS-K-SVD [102]. Specifically, they use a quadtree decomposition and overlapping image patches for obtaining a multiscale representation.

### Nonlocal algorithms

Nonlocal image denoising is based on the idea of exploiting the mutual similarities between different parts of the image, instead of relying only on information

available in a local neighbourhood. As natural images are typically highly redundant (e.g., an image of the sky, a building, or a grassy field will all most likely contain several highly self-similar small patches all around the image), a nonlocal algorithm can utilize these similarities in estimating the expected value of such an image patch. In theory, all pixels of the image can be taken into account in the estimation process, although it is computationally demanding. Thus, filters of this type are nonlocal.

The concept of nonlocality has been used for example in fractal compression methods already in the 1990s [97], but its applicability in image restoration, especially in denoising, has recently become more apparent due to such methods as nonlocal means [19], exemplar-based estimation [84], and collaborative filtering [34, 35].

To the best of our knowledge, nonlocal means is the first nonlocal image denoising algorithm (although it is interesting to note that the idea of reducing noise by leveraging signal redundancy was also briefly discussed by J.S. De Bonet in 1997 [46]). It was introduced in 2005 by A. Buades et al. [19], and they were inspired by the work of A. Efros and T. Leung [53], who took advantage of nonlocal self-similarities in natural images in the context of texture synthesis. The general concept of nonlocal means is to estimate each pixel of the image as a weighted average of pixels whose neighbourhood is similar to the neighbourhood of the estimated pixel. The similarity of two patches is defined as a Gaussian weighted Euclidean distance. Thus, large Euclidean distances will lead to small weights, and vice versa.

The exemplar-based estimation is an elaborate variant of the nonlocal means. Specifically, instead of fixed size nonlocal estimation neighbourhoods, it defines adaptive-sized neighbourhoods by using pairwise hypothesis testing. This estimation procedure is implemented by the structure adaptive filtering (SAFIR) algorithm [84, 15]. SAFIR is one of the algorithms used in our comparisons in Chapter 3.

#### BM3D

Introduced by the authors of SA-DCT, BM3D [34, 35] belongs to the group of nonlocal algorithms. However, due to its important role in this thesis, we devote a separate section to introducing it. Let us first describe the basic concept behind the algorithm superficially, and then elaborate the process in more detail.

In short, BM3D operates by grouping similar 2-D patches of the image into 3-D arrays, which we refer to as *groups*. Each group is processed by applying a 3-D transform to obtain a sparse representation of the signal, shrinking the transform spectrum and finally applying the corresponding inverse 3-D transform. In other words, 2-D patches are filtered collaboratively, producing a 3-D estimate of the group. This has proven to be an effective way of detecting the details shared by the 2-D patches without losing the distinct features of each patch.

In more detail, the algorithm will complete two rounds of processing. In the first round, the 2-D image blocks are grouped into a 3-D array by first choosing a reference block, and then finding other similar blocks via block matching (BM). The pairwise similarity between the reference and candidate blocks is defined by

using the Euclidean distance between prefiltered blocks. In particular, the blocks are prefiltered (hard-thresholded) in order to diminish the effect of noise to the obtained metric. The distance between the reference block $Z_R$ and the candidate block $Z$ is thus defined as

$$d(Z_R, Z) = \frac{\|\Gamma(\tau_{2D}(Z_R)) - \Gamma(\tau_{2D}(Z))\|_2^2}{N^2}, \tag{2.23}$$

where $\Gamma$ is a hard-thresholding operator, $\tau_{2D}$ is an orthonormal 2-D linear transform, and $N^2$ is the number of pixels in the image block. The 3-D array is then constructed by grouping $Z_R$ with all the candidate blocks $Z$, for which $d(Z_R, Z)$ is smaller than a predefined threshold.

The next step is to apply a 3-D transform to the obtained group. In practise, we take advantage of linear separability, and compute the 3-D transform domain coefficients by first applying a 2-D transform (e.g., 2-D DCT or a biorthogonal wavelet transform), followed by a 1-D transform (e.g., Haar transform) along the stacking dimension. In fact, the choice of these transforms has only a minor effect to the denoising result, as shown in the comparison [35, Table II].

After this, the noise is attenuated by hard-thresholding the obtained sparse 3-D representation of the signal, and finally the corresponding inverse transforms are applied in order to return to spatial domain. The denoised image blocks are then returned to their original places. However, since overlapping blocks are used, and one block may belong to several groups, we have an overcomplete representation of the denoised image. Thus, as in the SA-DCT algorithm, the individual denoised blocks will need to be aggregated in order to produce the final estimate of the block. This is done by weighted averaging of the individual results, where each weight is inversely proportional to the variance of the denoised block.

Now the algorithm has finished the first round, and we have obtained a *basic estimate* of the true image. The second round proceeds with the same general steps as the first round, but with a few crucial differences. First, groups are formed both on the basic estimate and on the original noisy image. However, the actual grouping is exclusively determined from the basic estimate, and the same grouping is then applied on the original noisy image. Consequently, the distance metric (2.23) can also be simplified by removing the thresholding operator $\Gamma$. In other words, we use the simple Euclidean distance between two blocks of the basic estimate. As a result of the grouping stage, we will have groups $\hat{Y}_b$ of the basic estimate, and groups $Z$ of the noisy image (but the location of the blocks in $Z$ is determined from the basic estimate).

After the grouping, the 3-D transform is computed for both groups $\hat{Y}_b$ and $Z$; the 2-D and 1-D transforms need not necessarily be the ones that were used in the first round. Then, instead of hard-thresholding, the noise is attenuated by Wiener filtering. In particular, the empirical Wiener shrinkage coefficients $W$ of a group are defined via the energy of the transformed group $\hat{Y}_b$:

$$W = \frac{|\tau_{3D}(\hat{Y}_b)|^2}{|\tau_{3D}(\hat{Y}_b)|^2 + \sigma^2},$$

where $\tau_{3D}$ is the 3-D transform, and $\sigma^2$ is the variance of the noisy image. Wiener

filtering and inverse 3-D transform are then applied on the *noisy group* $Z$:

$$\hat{Y} = \tau_{3D}^{-1}\left(W\tau_{3D}(Z)\right),$$

where $\hat{Y}$ is the final estimate of the particular block. Finally, as in the first round, these individual block estimates are aggregated by weighted averaging, ultimately producing the final denoised image. For more thorough implementation details, we refer the reader to [35].

Various modifications to BM3D have been proposed in the recent years. For instance, it has been extended for the cross-colour denoising of raw CFA data [42], for the denoising of RGB colour images [36], for video denoising [37], and most recently for the denoising, deblocking and enhancement of volumetric data by using a 4-D transform [100, 101].

In addition, joint sharpening and BM3D denoising was proposed in [38], where the sharpening was achieved via *alpha-rooting* [1]. In other words, the differences both within and between the grouped image blocks are amplified by taking the $\alpha$-root of the magnitude of the 3-D transform domain coefficients, for some $\alpha > 1$. This approach has also been adapted to use a varying sharpening strength $\alpha$ based on the edge and texture information of the image [111]; the edge and texture strengths are estimated as in [135].

Moreover, the shape adaptivity of SA-DCT has been combined with BM3D, in the form of SA-BM3D [39]. This has been further extended to BM3D-SAPCA [40], which employs principal component analysis (PCA) in the transform domain. Specifically, the final 3-D transform is a combination of the PCA applied on each image patch in the group, and the 1-D transform along the stacking dimension.

BM3D, BM3D-SAPCA, MS-K-SVD, K-SVD, SA-DCT and SAFIR are currently among the state-of-the-art algorithms in AWGN image denoising [25, 82]. Other recent efficient algorithms include the clustering-based K-LLD [24] that also uses locally learned dictionaries, the patch-based redundant Wiener filter PLOW [26] by the authors of K-LLD, and locally adaptive regression kernel (LARK) [130] that uses gradients in adapting a canonical kernel to the local image structure. For a deeper insight into some of the fundamental connections between various denoising algorithms, we refer the reader to [106].

Due to its state-of-the-art performance and fast execution speed, BM3D functions as our primary choice of a Gaussian denoising algorithm in all the experiments throughout this thesis. The excellent performance of the BM3D family can also be explained by considering it as an overcomplete sparse frame representation within a variational framework [44]. However, regardless of the algorithm, there is still a gap between the real life denoising performance and the theoretical lower bound for the mean squared error [25], leaving room for future improvements.

### 2.4.2 Poisson denoising algorithms

In the case of Poisson noise, the noise variance equals the expected value of the underlying true signal. This signal-dependency makes the premise for Poisson denoising very different from the case of AWGN denoising, where the variance is assumed to be constant. Hence, it is not surprising that the number of denoising algorithms specifically targeted at Poisson noise is relatively small in comparison

with the AWGN algorithms. Nevertheless, we review some of the best performing Poisson denoising algorithms (MS-VST [146], PH-HMT [94], and PURE-LET [98]), which are also used in our comparisons in Chapter 3.

As far as other Poisson denoising algorithms are concerned, both MS-VST and PURE-LET have been shown [146, 98] to outperform the Haar-Fisz algorithm [70] and platelets [141, 142]. Other algorithms are presented, e.g., in [133, 90, 91, 96], and most recently, [123] presented a nonlocal PCA-based algorithm addressing also extremely low-intensity cases. Classical and Bayesian wavelet shrinkage estimators are reviewed in [12].

### MS-VST

MS-VST, proposed by B. Zhang et al. in [146], is a denoising algorithm that extends the Anscombe transformation for the stabilization of a filtered discrete Poisson process. The obtained VST is then combined with a filter bank of wavelets, ridgelets or curvelets, yielding a multiscale VST. The stabilized coefficients are denoised using hypothesis testing, and the final denoised image is reconstructed via an iterative sparsifying scheme.

Let us review the variance stabilization step in more detail. The filtered Poisson process is defined as $Z_i = \sum_i h(i) z_{j-i}$, where $h$ is a non-negative discrete filter, and each $z_j \sim \mathcal{P}(y_j)$ is an independent Poisson variable. Their reasoning for this prefiltering is that it increases the SNR for the low-intensity cases, which they back up by an asymptotic analysis. In addition, they assume local homogeneity: all intensity values $y_j$ are assumed to be equal within the support of the filter $h$. By denoting $\tau_k = \sum_k h(i)^k$, the Anscombe transformation extended to this filtered case is then defined as

$$f(z) = b\sqrt{Z + c}, \tag{2.24}$$

where

$$b = 2\sqrt{\frac{\tau_1}{\tau_2}} \quad \text{and} \quad c = \frac{7\tau_2}{8\tau_1} - \frac{\tau_3}{2\tau_2}.$$

If no filtering is applied (i.e., $h$ is a Dirac), (2.24) coincides with the standard Anscombe transformation. In practise, (2.24) is combined with a wavelet filter bank in such a way that the smooth wavelet coefficients are stabilized with (2.24), and the detail coefficients are then computed from these stabilized smooth coefficients. Next, the obtained wavelet coefficients are deemed either significant or insignificant by using hypothesis testing [6]. Finally, the inverse MS-VST needs to be applied on the denoised coefficients. The authors also recognize that due to the nonlinearity of the MS-VST, direct inversion is unsuitable. Instead, they formulate the inversion as a convex sparsity-driven minimization problem, solved by iterative steepest descent. This iterative procedure is described more thoroughly in [146, Section III.E].

### PH-HMT

Another approach to Poisson denoising is presented by S. Lefkimmiatis et al. in [94], where the PH-HMT algorithm is introduced. Beginning with the idea of

quadtree multiscale decomposition, they devise a multiscale Poisson-Haar decomposition by adding a new intermediate scale between two regular quadtree scales. This way, they are able to model not only the horizontal and vertical features of the image, but also the the diagonal edges.

The transformed image can now be represented as a vector $y_e$ of rescaled 2-D Haar wavelet detail coefficients. In particular, these coefficients are treated as random variables following a beta distribution (conjugate of the binomial distribution). Since the variables are typically not independent, their interscale dependencies are modelled with a hidden Markov tree (HMT) [32].

The Bayesian posterior means of these random transform domain variables are then estimated by an expectation-maximization (EM) algorithm. The final estimate of the actual image $y$ can then be recursively recovered from these posterior mean values; the transformation from $y$ to $y_e$ is known, and the desired intensities are found at the finest scale of the decomposition.

#### PURE-LET

The third Poisson denoising algorithm featured in our comparisons is PURE-LET, proposed by F. Luisier et al. in [98]. Its concept is built on the idea of estimating the "risk" (MSE) between the noise-free and the denoised images. For a Gaussian distribution, this MSE can be estimated with Stein's unbiased risk estimate (SURE) [128]. However, in order to consider the Poisson case, the authors derive an interscale Poisson unbiased risk estimate (PURE), which minimizes the MSE in the Haar wavelet domain.

Then, as in their similar SURE-LET approach for Gaussian denoising [14], they minimize the estimated MSE over several denoising processes, in order to find the one providing the best SNR. In particular, these denoising processes are wavelet estimators expressed as a linear expansion of thresholds (LET); thresholds in this context mean arbitrary elementary estimators with unknown weights. Thus, the PURE estimate can be minimized by solving a low-dimensional system of linear equations (see [98, Section 3.2]).

### 2.4.3 Poisson-Gaussian denoising algorithms

The problem of directly denoising Poisson-Gaussian corrupted images is even less explored than that of Poisson corrupted images; often, the VST-based approaches are used instead. For instance, the SAFIR algorithm mentioned in Section 2.4.1 was used in combination with the generalized Anscombe transformation (2.18) in [15], for denoising Poisson-Gaussian corrupted fluorescence video-microscopy sequences. However, they presumably use either the algebraic or asymptotically unbiased inverse of the GAT, as they do not elaborate on the inversion details.

Another VST-based approach worth mentioning is the denoising of clipped Poisson-Gaussian corrupted images considered in [67], where the optimized nonparametric VSTs [65, 66] are combined with BM3D and other Gaussian denoising algorithms, providing high quality results.

Alternatively, the Poisson-Gaussian noise model is addressed directly in [99], where the authors of PURE-LET extend their Poisson denoising method to encompass the Poisson-Gaussian case. As described in Section 2.4.2, it operates

by constructing the unbiased PURE estimate (now: Poisson-Gaussian Unbiased Risk Estimate), and minimize this estimate using a linear expansion of thresholds (LET). Specifically, they generalize their method from the Haar wavelet domain to a general redundant transform domain, and update the PURE estimate to take into account both the Poisson and Gaussian noise components. Finally, they show that PURE allows for the global optimization of a LET spanning several different transform domains, such as undecimated wavelet transform (UWT) and block DCT. This UWT/BDCT PURE-LET algorithm is used in the comparisons in Chapter 4.

# Chapter 3

# Optimal inversion of the Anscombe transformation for Poisson noise

## 3.1 Introduction

The viability of variance stabilization for Poisson noise removal has often been questioned in the recent years. This issue has arised because of the poor numerical results achieved with it at the low-count regime, i.e., for low-intensity signals. In this chapter, we show that this disappointing performance, reported in many earlier works (e.g., [146], [94], [98]), is not caused simply by inaccurate stabilization, but rather by using an unsuitable inverse transformation. We show that the inversion plays a major role in ensuring the success of the whole denoising procedure. Specifically, the choice of the proper inverse transformation is crucial in order to minimize the bias error which arises when the nonlinear forward transformation is applied. We observe that both the algebraic inverse (2.19) and the asymptotically unbiased inverse (2.22) of the Anscombe transformation (2.15) lead to a significant bias at low counts. In particular, the latter inverse provides unbiasedness only asymptotically for large counts, while at low counts it actually leads to a larger bias than the former one.

This chapter presents our research results first published in [108], [110] and [112], organized as follows. In Section 3.2, we first propose the exact unbiased inverse of the Anscombe transformation, and derive a closed-form approximation for it. Then we show that the exact unbiased inverse can be interpreted as a maximum likelihood (ML) inverse, and finally we consider a more sophisticated minimum mean square error (MMSE) inverse. Section 3.3 provides an extensive experimental analysis using a few state-of-the-art denoising algorithms and shows that the results can be consistently improved by applying the exact unbiased inverse. In particular, the combination of BM3D [35] and the exact unbiased inverse outperforms some of the best existing algorithms specifically targeted at Poisson noise removal, while maintaining low computational complexity. Finally,

Section 3.4 presents the concluding remarks of this chapter.

## 3.2 Optimal inverse transformations

While the asymptotically unbiased inverse (2.22) provides good results for high-count data, applying it to low-count data leads to a biased estimate, as can be seen, e.g., in [146]. Here we consider three types of optimal inverses.

### 3.2.1 Exact unbiased inverse

Provided a successful denoising, i.e. $D$ is treated as $E\{f(z) \mid y\}$, the exact unbiased inverse of the Anscombe transformation $f$ is an inverse transformation $\mathcal{I}_0$ that maps the values $E\{f(z) \mid y\}$ to the desired values $E\{z \mid y\}$:

$$\mathcal{I}_0 : E\{f(z) \mid y\} \longmapsto E\{z \mid y\}. \tag{3.1}$$

Since $E\{z \mid y\} = y$ for any given $y$, the problem of finding the inverse $\mathcal{I}_0$ reduces to computing the values $E\{f(z) \mid y\}$, which is done by numerical evaluation of the integral corresponding to the expectation operator $E$:

$$E\{f(z) \mid y\} = \int_{-\infty}^{+\infty} f(z) p(z \mid y)\, dz, \tag{3.2}$$

where $p(z \mid y)$ is the generalized probability density function of $z$ conditioned on $y$. In our case we have discrete Poisson probabilities $P(z \mid y)$, so we can replace the integral by summation:

$$E\{f(z) \mid y\} = \sum_{z=0}^{+\infty} f(z) P(z \mid y). \tag{3.3}$$

Further, since here $f(z)$ is the forward Anscombe transformation (2.15), we can write (3.3) as

$$E\{f(z) \mid y\} = 2 \sum_{z=0}^{+\infty} \left( \sqrt{z + \frac{3}{8}} \cdot \frac{y^z e^{-y}}{z!} \right). \tag{3.4}$$

The practical implementation of $\mathcal{I}_0$ is discussed in more detail in Section 3.3.

Figure 3.1 shows the plots of the inverse transformations $\mathcal{I}_{\text{alg}}$, $\mathcal{I}_{\text{asy}}$ and $\mathcal{I}_0$. Since $\mathcal{I}_0$ is unbiased, we see that at low counts the asymptotically unbiased inverse actually leads to a larger bias than the algebraic inverse. This observation is further illustrated in Figure 3.2, where the absolute value of the bias is plotted for the three inverse transformations.

Let us remark that if the exact unbiased inverse (3.1) is applied to the denoised data $D$ with some errors (in the sense that $D \neq E\{f(z)|y\}$), then the estimation error in $\hat{y} = \mathcal{I}_0(D)$ can include variance as well as bias components. In general, the unbiasedness of $\mathcal{I}_0$ holds only provided that $D = E\{f(z)|y\}$ exactly, as it is assumed when defining (3.1).

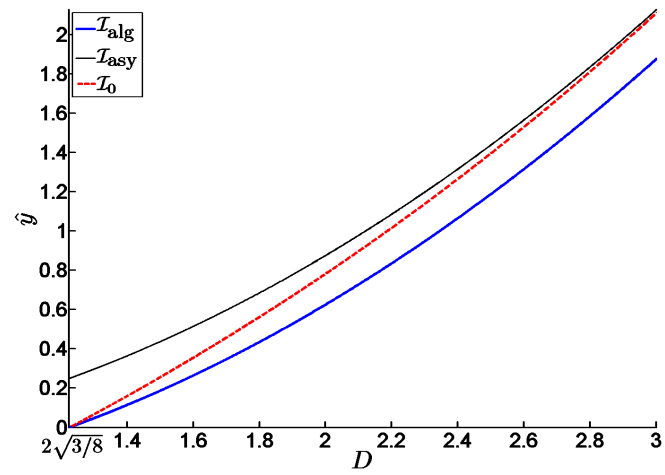



Figure 3.1: Inverse Anscombe transformations $\mathcal{I}_{\text{alg}}$ (algebraic), $\mathcal{I}_{\text{asy}}$ (asymptotically unbiased) and $\mathcal{I}_0$ (exact unbiased). The exact unbiased inverse is designed assuming $D = E\{f(z) \mid y\}$ and setting $\hat{y} = y$. Since $\mathcal{I}_0$ maps $E\{f(z) \mid y\}$ to $y$, it introduces no bias.

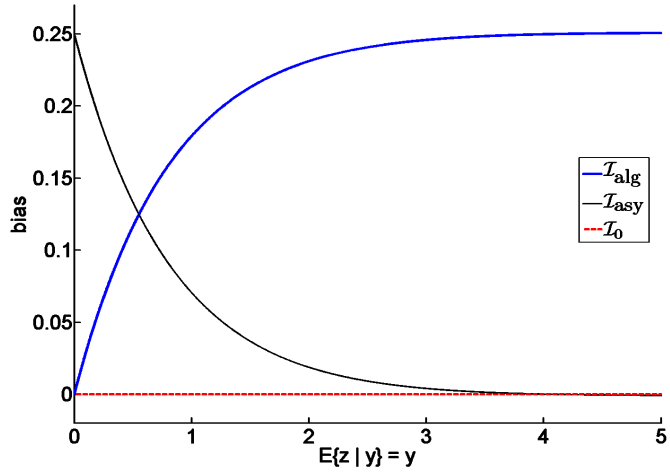



Figure 3.2: Absolute value of the bias of the inverse Anscombe transformations $\mathcal{I}_{\text{alg}}$ (algebraic), $\mathcal{I}_{\text{asy}}$ (asymptotically unbiased) and $\mathcal{I}_0$ (exact unbiased). Since $\mathcal{I}_0$ maps $E\{f(z) \mid y\}$ to $y$, it introduces no bias.

### 3.2.2 Closed-form approximation of the exact unbiased inverse

In order to facilitate the use of the exact unbiased inverse $\mathcal{I}_0$ in applications, let us also construct a closed-form approximation $\widetilde{\mathcal{I}}_0$ for it.

The construction is done by first considering the asymptotically unbiased inverse $\mathcal{I}_{\text{asy}}$, and then subtracting a non-constant correction term from it. Note that

$$\mathcal{I}_0\left(2\sqrt{3/8}\right) = \mathcal{I}_{\text{alg}}\left(2\sqrt{3/8}\right) = \mathcal{I}_{\text{asy}}\left(2\sqrt{3/8}\right) - \frac{1}{4} = 0,$$

and that $\mathcal{I}_0(D) - \mathcal{I}_{\text{asy}}(D) \to 0$, as $D \to \infty$. Thus, our correction term should equal $\frac{1}{4}$ at $2\sqrt{\frac{3}{8}}$, and approach zero as $D$ increases. It is easy to verify that the inverse proposed below satisfies these conditions:

$$\begin{aligned}\widetilde{\mathcal{I}}_C(D) \quad = \quad & \left(\frac{D}{2}\right)^2 - \frac{1}{8} - \frac{1}{4}\left[\alpha\left(D\sqrt{\frac{2}{3}}\right)^{-1} + \right. \\ & \left.\beta\left(D\sqrt{\frac{2}{3}}\right)^{-2} + (1-\alpha-\beta)\left(D\sqrt{\frac{2}{3}}\right)^{-3}\right],\end{aligned} \tag{3.5}$$

where $D \geq 2\sqrt{\frac{3}{8}}$ and $\alpha, \beta \in \mathbb{R}$. As with $\mathcal{I}_0$, for $D < 2\sqrt{\frac{3}{8}}$ we set also $\widetilde{\mathcal{I}}_0(D) = 0$. We choose the values of $\alpha$ and $\beta$ by minimizing the integral criterion

$$\Phi = \int_0^{+\infty} \frac{\left(\widetilde{\mathcal{I}}_0\left(E\left\{f(z) \mid y\right\}\right) - y\right)^2}{y^2}\, dy, \tag{3.6}$$

which can be interpreted as a normalized quadratic fit between $\mathcal{I}_0$ and $\widetilde{\mathcal{I}}_0$. The minimization yields $\alpha \approx -1.0008$ and $\beta \approx 3.6634$, with the corresponding value of $\Phi \approx 7.4368 \cdot 10^{-5}$. As small variations in the coefficient values do not greatly affect the fit, it seems reasonable to choose $\alpha = -1$ and $\beta = \frac{11}{3}$ ($\Phi \approx 7.4658 \cdot 10^{-5}$), and thus simplify (3.5) into the final closed-form approximation

$$\widetilde{\mathcal{I}}_0(D) = \frac{1}{4}D^2 + \frac{1}{4}\sqrt{\frac{3}{2}}D^{-1} - \frac{11}{8}D^{-2} + \frac{5}{8}\sqrt{\frac{3}{2}}D^{-3} - \frac{1}{8}. \tag{3.7}$$

### 3.2.3 ML inverse

In Section 3.2.1 we assumed that the denoising is successful (i.e. we can treat the denoised signal $D$ as $E\{f(z) \mid y\}$), which then lead us to the concept of the exact unbiased inverse. Now we consider a more general scenario, where this assumption does not necessarily hold: instead of the strict equality, we assume that the pointwise mean square error of $D$ as an estimate of $E\{f(z) \mid y\}$ is

$$\varepsilon^2 = E\left\{\left(D - E\{f(z) \mid y\}\right)^2\right\}. \tag{3.8}$$

In practise the distribution of $D$ is unknown. For simplicity, we assume that $D$ is normally distributed around $E\{f(z) \mid y\}$ with variance $\varepsilon^2$:

$$D \sim \mathcal{N}\left(E\{f(z) \mid y\}, \varepsilon^2\right). \tag{3.9}$$

While formally (3.9) implies that $D$ is an unbiased estimate of $E\{f(z) \mid y\}$, in fact also unknown estimation-bias errors can be considered as contributors of $\varepsilon^2$, with the symmetry of the distribution about $E\{f(z) \mid y\}$ reflecting our uncertainty about the sign of the bias.

By treating $D$ as the data, the maximum likelihood (ML) inverse is defined as

$$\mathcal{I}_{\mathrm{ML}}(D) = \arg\max_{y} p(D \mid y), \tag{3.10}$$

where, according to (3.9),

$$p(D \mid y) = \frac{1}{\sqrt{2\pi\varepsilon^2}} e^{-\frac{1}{2\varepsilon^2}(D - E\{f(z)|y\})^2}. \tag{3.11}$$

Under the above assumptions, this equals to

$$\mathcal{I}_{\mathrm{ML}}(D) = \begin{cases} \mathcal{I}_0(D), \text{ if } D \geq 2\sqrt{3/8} \\ 0, \text{ if } D < 2\sqrt{3/8}. \end{cases} \tag{3.12}$$

To prove (3.12), let us consider two cases. First, if $D \geq 2\sqrt{3/8}$, (3.10) can be maximized by choosing $y$ in such a way that the maximum of the probability density function (PDF) of $D$ given $y$ (3.11) coincides with $D$, i.e. $E\{f(z) \mid y\} = D$. Thus, from (3.1) we obtain the first half of (3.12). Second, if $D < 2\sqrt{3/8}$, it is not possible for the maximum of the PDF (3.11) to coincide with $D$, since $y \geq 0$; however, because this PDF is monotonically increasing between $-\infty$ and $E\{f(z) \mid y\}$, and because variations of $y$ correspond to translations of the PDF, $\arg\max_{y} p(D \mid y)$ is achieved with the smallest possible $y$, i.e. $y = 0$. Thus, the exact unbiased inverse coincides with this form of ML inverse.

Note that this proof relies only on the fact that the distribution of $D$ is unimodal with mode at $E\{f(z) \mid y\}$. Note also that $\mathcal{I}_{\mathrm{ML}}(D)$ is independent of $\varepsilon$. The obtained result (3.12) holds for any unimodal distribution whose mode is $E\{f(z) \mid y\}$.

### 3.2.4 MMSE inverse

Under the same hypotheses of Section 3.2.3, we define the minimum mean square error (MMSE) inverse, which is parametrized by $\varepsilon$, as

$$\begin{aligned} \mathcal{I}_{\mathrm{MMSE}}(D, \varepsilon) &= \arg\min_{\hat{y}} E\{(y - \hat{y})^2 \mid D\} \\ &= \arg\min_{\hat{y}} \int_{-\infty}^{+\infty} p(y \mid D)\,(y - \hat{y})^2\, dy. \end{aligned} \tag{3.13}$$

It is worth reminding that we assume $D$ to be normally distributed according to (3.9); if this assumption does not hold, the obtained inverse is not necessarily the true minimum MSE inverse.

Now, let us derive an explicit formula for computing the MMSE inverse (3.13). According to Bayes' theorem,

$$p(y \mid D) = \frac{p(D \mid y)p(y)}{p(D)}, \tag{3.14}$$

where $p(D \mid y)$ is given by (3.11). Thus, (3.13) is equivalent to

$$\begin{aligned} \mathcal{I}_{\mathrm{MMSE}}(D, \varepsilon) &= \arg\min_{\hat{y}} \int_{-\infty}^{+\infty} \frac{p(y)}{p(D)} p(D \mid y)(y - \hat{y})^2 \, dy. \\ &= \arg\min_{\hat{y}} \int_{-\infty}^{+\infty} p(y) p(D \mid y)(y - \hat{y})^2 \, dy. \end{aligned} \tag{3.15}$$

If we additionally assume that the true signal $y$ has an improper uniform distribution over $\mathbb{R}^+$ (noninformative prior), we can further write

$$\mathcal{I}_{\mathrm{MMSE}}(D, \varepsilon) = \arg\min_{\hat{y}} \int_{0}^{+\infty} p(D \mid y)(y - \hat{y})^2 \, dy. \tag{3.16}$$

The $\hat{y}$ minimizing the integral in (3.16) is found by differentiation, giving us

$$\hat{y} = \frac{\int_0^{+\infty} p(D \mid y) y \, dy}{\int_0^{+\infty} p(D \mid y) \, dy}. \tag{3.17}$$

In practise, the same result can be obtained by equivalently assuming $y$ to be uniformly distributed between 0 and $M$, where $M > 0$ is a constant much larger than any of our observations $z_i$, because

$$\int_0^{+\infty} p(D \mid y)(y - \hat{y})^2 \, dy \approx \int_0^{M} p(D \mid y) \, (y - \hat{y})^2 \, dy.$$

The difference between the two integrals is negligible due to our choice of $M$ and the exponential decay of $p(D \mid y)$.

Note that the exact unbiased inverse can be considered a limit case of the MMSE inverse, obtained when $\varepsilon = 0$, because $p(D \mid y)$ becomes a Dirac impulse centered at that particular value of $y$ such that $E\{f(z) \mid y\} = D$. In other words,

$$\mathcal{I}_{\mathrm{MMSE}}(D, 0) = \mathcal{I}_0(D) = \mathcal{I}_{\mathrm{ML}}(D). \tag{3.18}$$

Figure 3.3 shows the MMSE inverse transformations for some values of $\varepsilon$, including the case $\varepsilon = 0$.

## 3.3 Experiments

All of our experiments consist of the same three-step denoising procedure: First we apply the forward Anscombe transformation (2.15) to a noisy image. Then we denoise the transformed image (assuming additive white Gaussian noise of unit variance) with either BM3D [35], SAFIR [15] or BLS-GSM [120], and finally we apply an inverse transformation in order to get the final estimate.

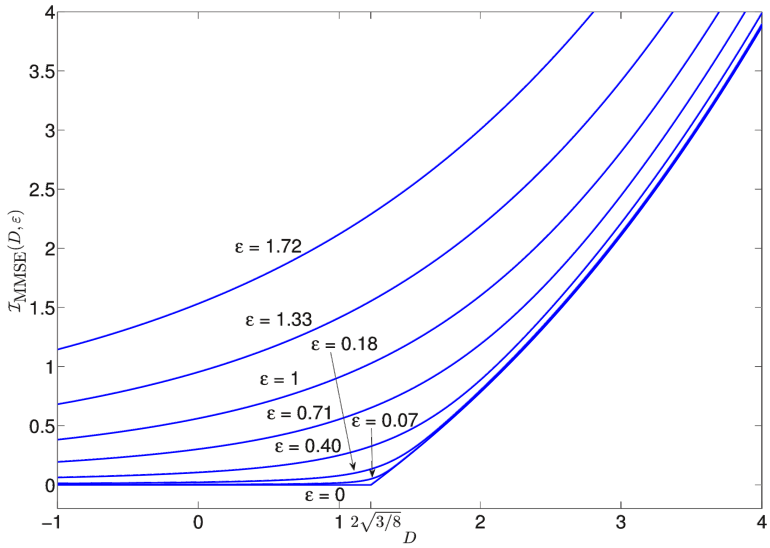



Figure 3.3: MMSE inverse transformations (3.13) for some values of $\varepsilon$. The case $\varepsilon = 0$ corresponds to the exact unbiased inverse transformation (3.1) and the ML inverse transformation (3.12).

To implement the exact unbiased inverse $\mathcal{I}_0$ in practise, it is sufficient to compute (3.4) for a limited set of values[1] $y$. For arbitrary values of $y$ we then use linear interpolation based on these computed values of (3.4), and for large values[2] of $y$ we approximate $\mathcal{I}_0$ by $\mathcal{I}_{\mathrm{asy}}$. In similar fashion, the MMSE inverse can be obtained based on numerical evaluation of the two integrals in (3.17).

Matlab functions implementing these two optimal inverse transformations are available online at http://www.cs.tut.fi/~foi/invansc.

We evaluate the performance either by normalized mean integrated square error (NMISE) or by peak signal-to-noise ratio (PSNR). The NMISE is calculated using the formula

$$\frac{1}{\bar{N}} \sum_{i:y_i>0} \left( (\hat{y}_i - y_i)^2 / y_i \right), \tag{3.19}$$

where $\hat{y}_i$ are the estimated intensities, $y_i$ the respective true values, and the sum is computed over the $\bar{N}$ pixels in the image for which $y_i > 0$. The PSNR is calculated using the formula

$$10 \log_{10} \left( \frac{\max_i (y_i)^2}{\left( \sum_i (\hat{y}_i - y_i)^2 / N \right)} \right), \tag{3.20}$$

where $N$ is the total number of pixels in the image.

[1] We compute (3.4) for 30000 values of $y \in [0, 2500]$ with quadratic spacing, implying a particularly high precision for small values of $y$.

[2] In our implementation, we consider $y$ to be large if $y > 2500$.

In Section 3.3.1 we consider the exact unbiased inverse, and Section 3.3.2 consists of experiments with the MMSE inverse. Section 3.3.3 addresses the computational complexity of the inverse transformations and the denoising algorithms.

### 3.3.1 Exact unbiased inverse

We consider three sets of experiments in order to compare against the three recent works [146], [94] and [98], each of which proposes an algorithm specifically designed for Poisson noise removal (MS-VST, PH-HMT and Interscale PURE-LET, respectively). To the best of our knowledge, these algorithms are currently achieving the best results in denoising Poisson data.

#### NMISE comparison against MS-VST [146] and PH-HMT [94]

For the first set we proceed in the same way as in [146] in order to produce comparable results: The above-mentioned three-step denoising procedure is performed five times for each image, each time with a different realization of the random noise. We evaluate the performance by using NMISE, and the obtained NMISE values are finally averaged over these five replications. This metric was chosen because of the available results for comparison in [146] and [94]. The authors of [146] also kindly provided us with their set of test images (all of them 256×256 in size), shown in Figure 3.4.

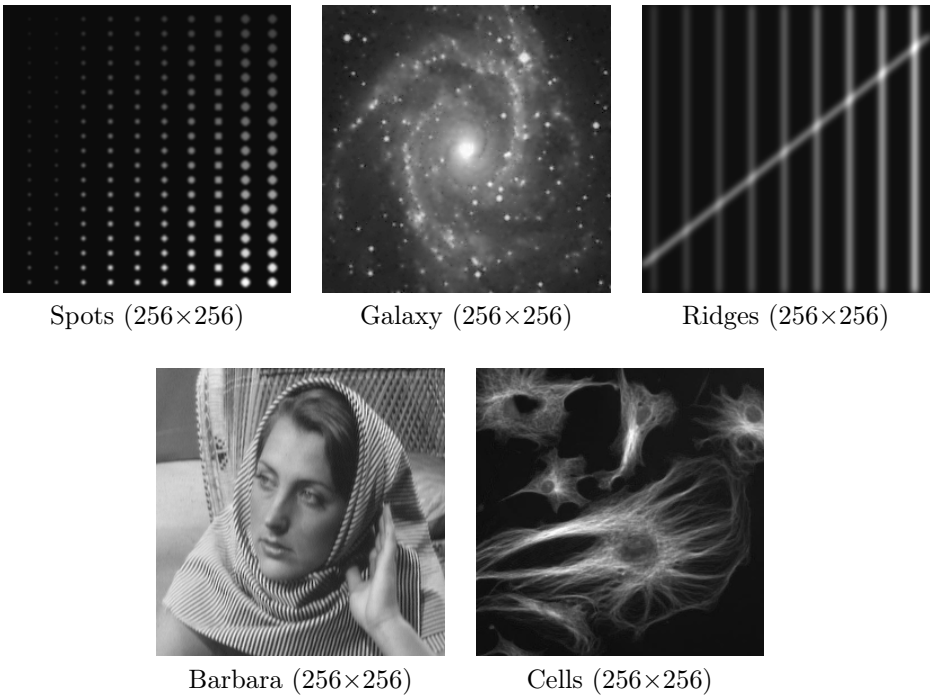



Figure 3.4: The five test images used in the experiments of Sections 3.3.1 and 3.3.2.

The denoising is done with either BM3D, SAFIR or BLS-GSM, and for the inversion of the denoised signal we use the exact unbiased inverse. The same experiments are also done for the asymptotically unbiased inverse (2.22), whose results serve as a point of comparison.

The numerical results of our experiments are presented in Table 3.1, where we also compare them to the state-of-the-art results obtained with the PH-HMT and MS-VST algorithms. In addition, we have included the results obtained in [146] with the asymptotically unbiased inverse Anscombe transformation combined with various undecimated wavelet transforms (here collectively denoted as WT). Table 3.1 shows not only that the exact unbiased inverse produces significantly better results at low counts than the asymptotically unbiased inverse, but also that the method is competitive with both PH-HMT and MS-VST. In particular, the combination of BM3D and the exact unbiased inverse outperforms both of them in terms of NMISE.

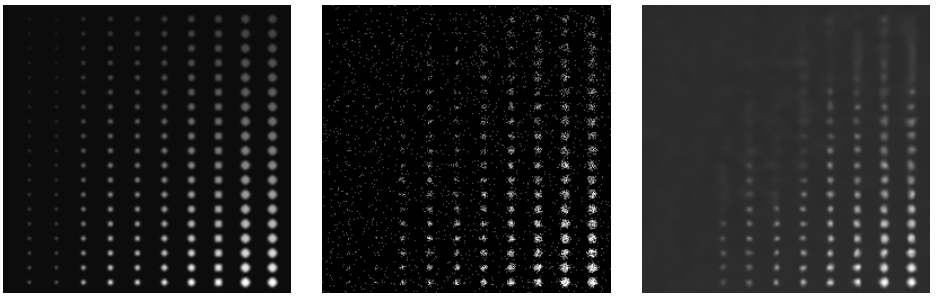
Figures 3.5–3.6 illustrate the improvement that is achieved (especially at low counts) by applying the exact unbiased inverse instead of the asymptotically unbiased inverse, while Figure 3.7 compares the different algorithms for the denoising of the Cells image (the exact unbiased inverse combined with BM3D, SAFIR and BLS-GSM, and the best MS-VST result from [146]). In addition, we present a chosen cross-section (i.e. one row) of some of the test images in Figure 3.8. These plots also clearly demonstrate that at low counts the exact unbiased inverse provides a significant improvement over the asymptotically unbiased inverse, whereas at high counts the difference is expectedly negligible.

We also evaluate the performance of the closed-form approximation $\widetilde{\mathcal{I}}_0$ (3.7) of the exact unbiased inverse $\mathcal{I}_0$. This evaluation is done with the test cases of Table 3.1, by comparing the performance associated with $\mathcal{I}_{\text{asy}}$, $\mathcal{I}_0$ and $\widetilde{\mathcal{I}}_0$, each combined with BM3D. The results are presented in Table 3.2. We observe that in all cases, the denoising results obtained with $\mathcal{I}_0$ and $\widetilde{\mathcal{I}}_0$ are within 0.5 % of each other in terms of NMISE. In other words, we can consider the exact unbiased inverse and its closed-form approximation to be equivalent for practical purposes. Moreover, these results also emphasize the inferior performance of the asymptotically unbiased inverse $\mathcal{I}_{\text{asy}}$ in the low-intensity cases. In summary, also the analytical expression (3.7) can be successfully used as a direct replacement of the traditional inverses $\mathcal{I}_{\text{alg}}$ and $\mathcal{I}_{\text{asy}}$ in all cases, with a notably better performance in low-count Poisson denoising.

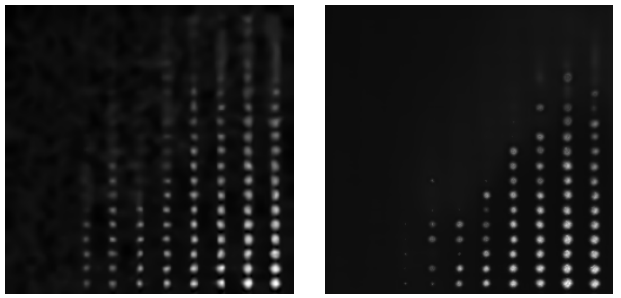
### PSNR comparison against PH-HMT [94]

In the second part of the experiments we use the test images shown in Figure 3.9 and evaluate the performance in terms of PSNR, thus enabling us to compare against the PH-HMT results in [94]. This time we scale each image to seven different peak intensity levels (1, 2, 3, 4, 5, 10 and 20), and for each of them we perform the denoising procedure ten times, with ten different realizations of the random noise.

As above, we use either BM3D, SAFIR or BLS-GSM for the denoising, and the inversion is done with either the exact unbiased inverse or the asymptotically unbiased inverse.

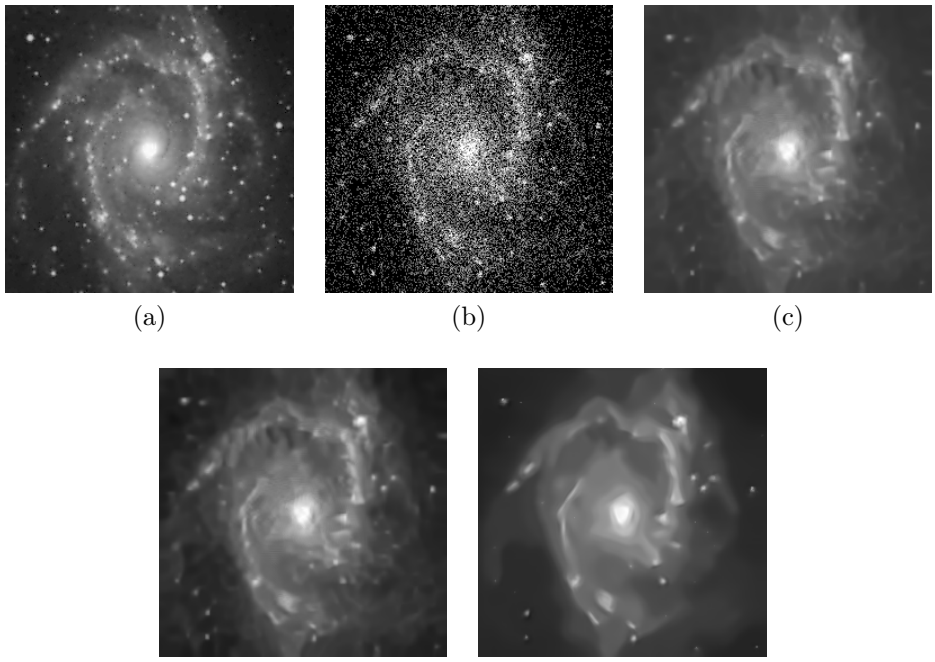


(a) (b) (c)



(d) (e)

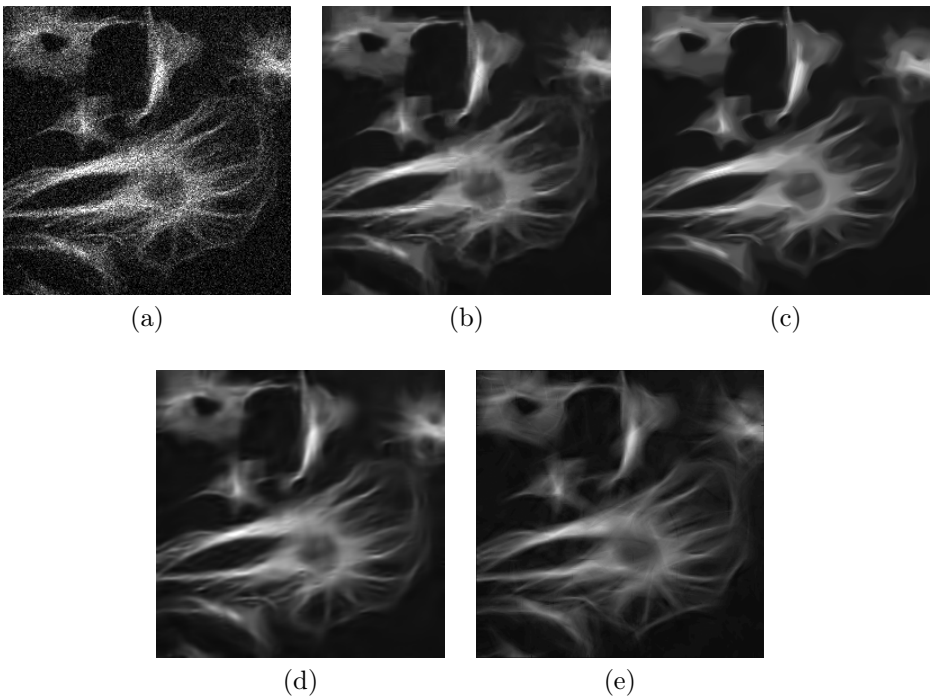
Figure 3.5: (a) Original Spots image (intensity range [0.03, 5.02]), (b) Poisson-count image, (c) image denoised with BM3D and the asymptotically unbiased inverse (average NMISE = 1.7395), (d) image denoised with BM3D and the exact unbiased inverse (average NMISE = 0.0365), (e) image denoised with SAFIR and the exact unbiased inverse (average NMISE = 0.0384). The images shown here are gamma-corrected ($\gamma = 0.6$) for improved visibility of the darker areas. A cross-section of images (a), (c) and (d) is shown in Figure 7(a).



(a) (b) (c)

(d) (e)

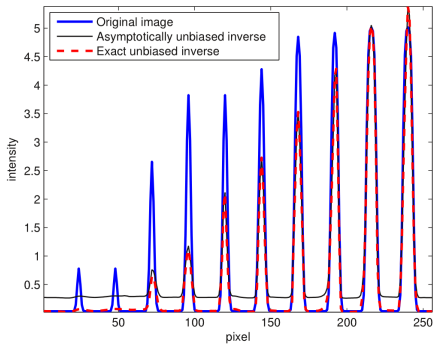
Figure 3.6: (a) Original Galaxy image (intensity range [0, 5]), (b) Poisson-count image, (c) image denoised with BM3D and the asymptotically unbiased inverse (average NMISE = 0.1025), (d) image denoised with BM3D and the exact unbiased inverse (average NMISE = 0.0299), (e) image denoised with SAFIR and the exact unbiased inverse (average NMISE = 0.0301). The images shown here are gamma-corrected ($\gamma = 0.6$) for improved visibility of the darker areas. A cross-section of images (a), (c) and (d) is shown in Figure 7(b).



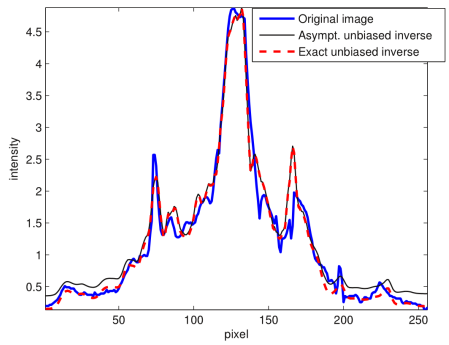
(a) (b) (c)

(d) (e)

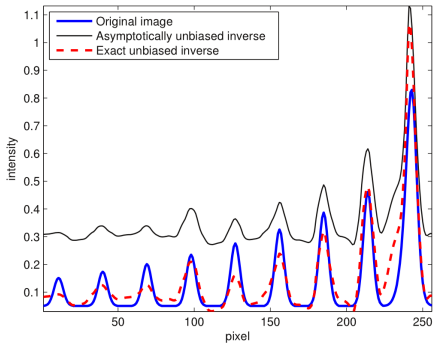
Figure 3.7: (a) Noisy Cells image, denoised with (b) BM3D and the exact unbiased inverse (average NMISE = 0.0649), (c) SAFIR and the exact unbiased inverse (average NMISE = 0.0671), (d) BLS-GSM and the exact unbiased inverse (average NMISE = 0.0707), (e) MS-VST + curvelets (average NMISE = 0.078) [146]. The original image is shown in Figure 3.4.



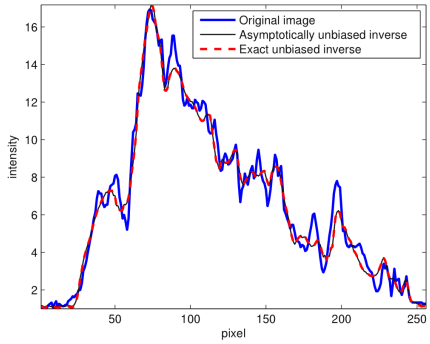

(a) Spots (row 247)




(b) Galaxy (row 130)




(c) Ridges (row 40)




(d) Cells (row 145)

Figure 3.8: Cross-sections of some of the images denoised with BM3D. For Cells the intensities are large enough for the two inverses to practically coincide.

The results, which are averages of ten PSNR values, are reported in Table 3.3. We see again that at low peak intensities we get a substantial improvement by applying the exact unbiased inverse instead of the asymptotically unbiased inverse, regardless of the used denoising algorithm. Indeed, the best results of the table mainly correspond to algorithms combined with the exact unbiased inverse. In particular, the best overall performance is obtained with BM3D, although both SAFIR and PH-HMT provide competitive results especially at the lowest peak intensities. The different performance at low counts is possibly explained by SAFIR exploiting adaptive window sizes, as opposed to BM3D, which uses fixed-size blocks.

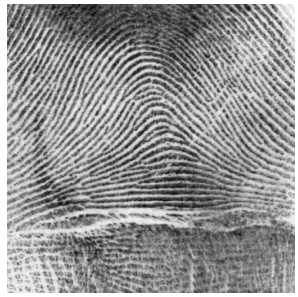
Note that the average PSNR values of the noisy images in Table 3.3 have minor differences to those reported in [94] due to different realizations of the random noise.

Cameraman (256×256)

Lena (512×512)

Boat (512×512)

Barbara (512×512)



Fingerprint (512×512)

Figure 3.9: The five test images used in the experiments of Section 3.3.1.

**PSNR comparison against PURE-LET [98]**

The third part of our experiments is very similar to the second one, with the following differences: Now we use the four test images shown in Figure 3.10 and compare our results against the best results obtained with the Interscale PURE-LET (with two cyclic shifts) [98]. Each image is scaled to the peak intensity levels 1, 5, 10, 20, 30, 60 and 120, so we do not focus at low counts as much as earlier.

As in [98], the denoising performance is evaluated in terms of PSNR. Table 3.4 presents the obtained results (averages of ten values), which are consistent with


Peppers (256×256)


Cameraman (256×256)


MIT (256×256)


Moon (512×512)

Figure 3.10: The four test images used in the experiments of Section 3.3.1.

the results in Table 3.3: the overall performance of BM3D is strong, but it is often outperformed by SAFIR at the lower peak intensity levels. It is interesting to note that for an image like Moon, which presents a large black background area that is completely flat, the performance gap between multiscale (BLS-GSM and PURE-LET) and patch-based (BM3D and SAFIR) methods is reduced, in as much as in a few cases the former methods are producing slightly better numerical results than the latter ones.

**Summary**

All three sets of experiments produce consistent results, showing that at low intensities we obtain significantly better results by applying the exact unbiased inverse instead of the asymptotically unbiased inverse, whereas at high intensities there is expectedly no significant improvement.

The results also show that combined with a state-of-the-art Gaussian denoising algorithm, the exact unbiased inverse is competitive with some of the best algorithms targeted at Poisson noise removal.

### 3.3.2 MMSE inverse

Assuming that (3.9) is valid, the use of the MMSE inverse $\mathcal{I}_{\mathrm{MMSE}}$ (3.13) requires knowledge of the pointwise mean square error (3.8) for the estimate $D$ produced by

the denoising algorithm. In other words, for each pixel, a pair $(D, \varepsilon)$ is used as an argument for $\mathcal{I}_{\mathrm{MMSE}}$. First, in order to illustrate the full potential of this inverse, we show results obtained by employing an oracle estimate of the MSE computed by Monte-Carlo simulations. Second, as an example of the actual performance that can be achieved in practise, we compute an estimate of the MSE using Stein's unbiased risk estimate (SURE) [128].

For both cases, we present the results corresponding to Table 3.1 produced using the BM3D algorithm.

**Oracle Monte-Carlo MSE**

Here we compute the mean square error (3.8) from 50 independent replications of the denoising experiment. This estimate, denoted as $\varepsilon^2_{\mathrm{MC}}$, is an oracle estimate which cannot obviously be produced if $y$ is unknown. Note that very fine structures of the image are visible in $\varepsilon_{\mathrm{MC}}$, as shown in the leftmost images in Figures 3.11–3.12. The corresponding MMSE estimate of $y$ is obtained as $\mathcal{I}_{\mathrm{MMSE}}(D, \varepsilon_{\mathrm{MC}})$. The average NMISE and PSNR results over 5 independent replications of $z$ are reported in Table 3.5.

For some images, the improvement is dramatic (up to almost 2.7 dB PSNR difference for Spots). However, for those images where already the exact unbiased inverse did not provide substantial improvement over the asymptotically unbiased inverse (see Table 3.1), the differences are much smaller. In the case of Barbara the results are even slightly worse. This can be attributed to the failure of the normal model (3.9) in describing the actual estimation errors for this particular image dominated by repeated texture.

**Empirical SURE estimate**

Stein's unbiased risk estimate (SURE) [128] can be used to provide a surrogate for the mean square error (3.8) for an arbitrary denoising algorithm without needing $y$ to be known and without resorting to multiple realizations of the noise. Assuming that the noise corrupting $f(z)$ is zero-mean Gaussian with diagonal covariance matrix $\sigma^2\mathbf{1}$, the SURE for $D$ is

$$\mathrm{SURE}(D_i) = (D_i - f(z_i))^2 + \sigma^2 \left( 2\frac{\partial D_i}{\partial f(z_i)} - 1 \right), \tag{3.21}$$

where $i = 1, \ldots, N$. The Anscombe variance-stabilizing transformation ensures that these assumptions approximately hold with $\sigma^2 = \mathrm{var}\{f(z)\,|y\} \simeq 1$. We compute the partial derivative $\frac{\partial D_i}{\partial f(z_i)}$ in (3.21) as the finite difference $\left(D_i^\delta - D_i\right)/\delta$, where $D^\delta$ is the denoised output obtained after perturbing $f(z_i)$ with a finite increment $\delta = 1.5$ (this value is chosen so that the perturbation can compete with the noise). Thus, calculating (3.21) requires $N$ individual denoising procedures. Although these can be first accelerated by processing only a neighborhood of the perturbed sample and then parallelized, the computational cost of this SURE approach remains obviously very high. Depending on the particular denoising filter, more sophisticated approaches to empirically estimate (3.21) exist (see, e.g., [122]).

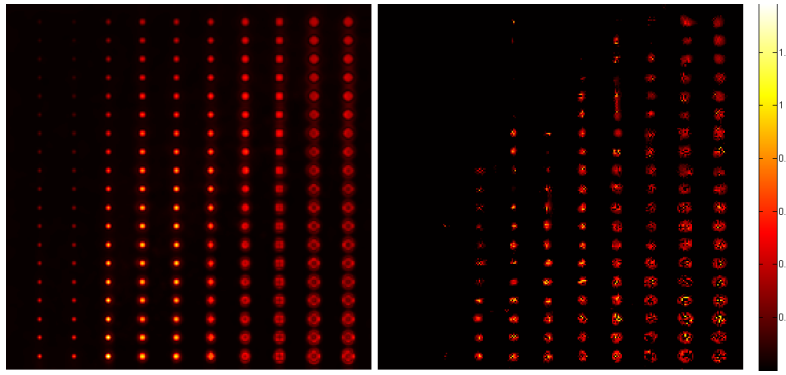



Figure 3.11: Oracle Monte-Carlo estimate $\varepsilon_{\mathrm{MC}}$ for Spots (left) compared with the respective SURE estimate $\varepsilon_{\mathrm{SURE}}$ (right).

This risk estimator is unbiased, but its variance can be high, so it is reasonable to smooth SURE($D$) in order to approximate its expectation, thus improving the estimate. Also for this smoothing we use the BM3D filter. Further, since $\mathrm{var}\{f(z)|y\}$ is constant only approximately, we replace the factor $\sigma^2$ in (3.21) by $\mathrm{var}\{f(z)|\mathcal{I}_0(D)\}$ (this conditional variance of $f(z)$ can be computed numerically as it is done for its conditional expectation). We denote the obtained MSE estimate as $\varepsilon^2_{\mathrm{SURE}}$. In Figures 3.11–3.12 we compare $\varepsilon_{\mathrm{SURE}}$ estimates with oracle Monte-Carlo estimates $\varepsilon_{\mathrm{MC}}$. The average results obtained from the MMSE estimate $\mathcal{I}_{\mathrm{MMSE}}(D, \varepsilon_{\mathrm{SURE}})$ are given in Table 3.5. On a very simple image, such as Spots, $\varepsilon^2_{\mathrm{SURE}}$ can provide a reasonable approximation of the mean square error, and thus a noticeable improvement in the PSNR. However, for all other images this approximation is too coarse and the results do not differ on average from the results of the exact unbiased inverse. Figures 3.13–3.14 compare the results of the three inverses $\mathcal{I}_0(D) = \mathcal{I}_{\mathrm{ML}}(D)$, $\mathcal{I}_{\mathrm{MMSE}}(D, \varepsilon_{\mathrm{MC}})$ and $\mathcal{I}_{\mathrm{MMSE}}(D, \varepsilon_{\mathrm{SURE}})$ for Spots and Barbara, respectively. While visually the three estimates of Barbara are virtually indistinguishable, one can observe that for Spots the MMSE inverses, particularly $\mathcal{I}_{\mathrm{MMSE}}(D, \varepsilon_{\mathrm{MC}})$, provide sharper details than the exact unbiased inverse.

### 3.3.3 Computational complexity

As our Matlab implementation of the exact unbiased inverse takes advantage of precomputed values of $E\{f(z) \mid y\}$ and $E\{z \mid y\}$, the inverse transformation can be executed quickly. Thus, the computation time of the whole denoising procedure mainly depends on the execution time of the chosen denoising algorithm. Table 3.6 shows average computation times of the denoising of Lena ($512 \times 512$, peak 10) and Cameraman ($256 \times 256$, peak 10) with the exact unbiased inverse, for two different CPUs. It is worth noting that for SAFIR we do not use the default parameters, but the ones that should give the best results (no subsampling, patch radius=3, iterations=8, lambda=66, eta=3.7). This significantly increases the computation time by a factor of about 580, but provides an improvement of about 0.6 dB (see [85] for details about the complexity/performance scaling of the algorithm). Also

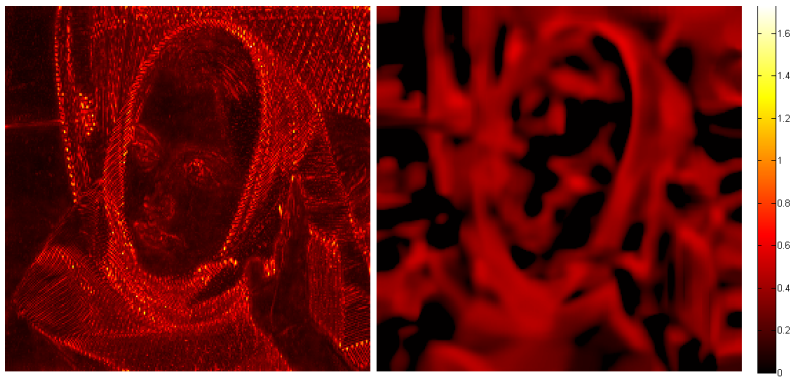



Figure 3.12: Oracle Monte-Carlo estimate $\varepsilon_{\mathrm{MC}}$ for Barbara (left) compared with the respective SURE estimate $\varepsilon_{\mathrm{SURE}}$ (right).

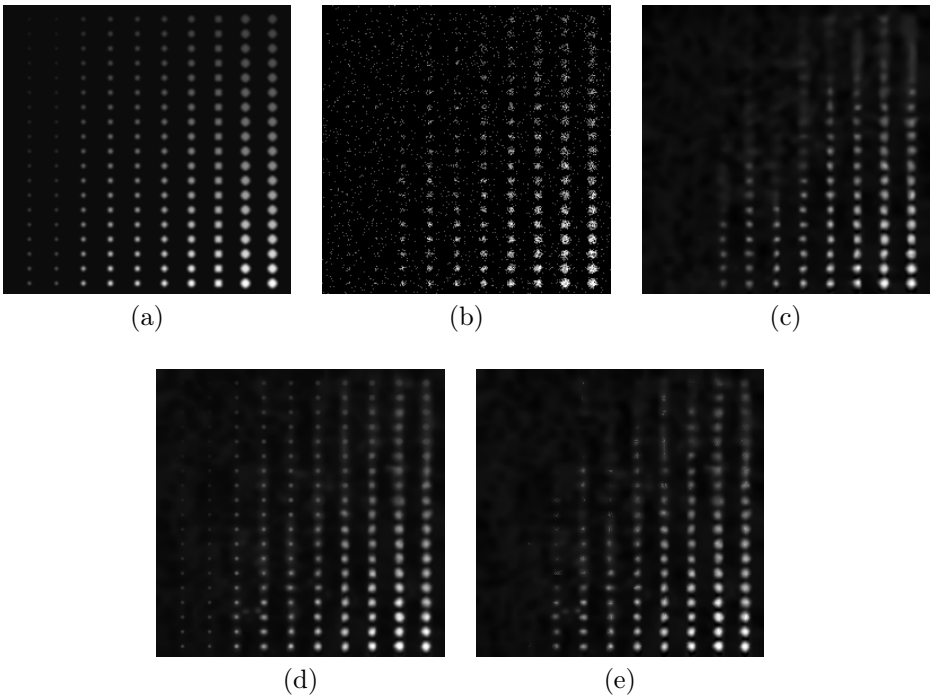
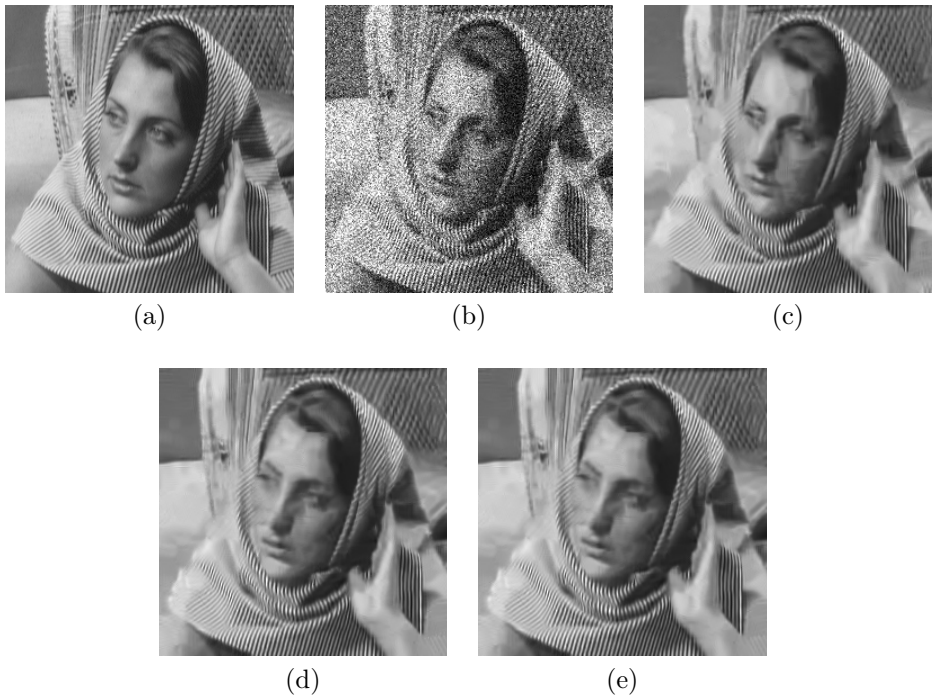



Figure 3.13: (a) Original Spots image (intensity range [0.03, 5.02]), (b) Poisson-count image, (c) image denoised with BM3D and the exact unbiased inverse (average NMISE = 0.0365 and PSNR = 31.96 dB), (d) image denoised with BM3D and the MMSE inverse with the oracle Monte-Carlo estimate $\varepsilon_{\mathrm{MC}}$ (average NMISE = 0.0324 and PSNR = 34.65 dB), (e) image denoised with BM3D and the MMSE inverse with the SURE estimate $\varepsilon_{\mathrm{SURE}}$ (average NMISE = 0.0362 and PSNR = 32.81 dB). The images shown here are gamma-corrected ($\gamma = 0.6$) for improved visibility of the darker areas.



(a) (b) (c)

(d) (e)

Figure 3.14: (a) Original Barbara image (intensity range [0.93, 15.73]), (b) Poisson-count image, (c) image denoised with BM3D and the exact unbiased inverse (average NMISE = 0.0881 and PSNR = 25.92 dB), (d) image denoised with BM3D and the MMSE inverse with the oracle Monte-Carlo estimate $\varepsilon_{\mathrm{MC}}$ (average NMISE = 0.0971 and PSNR = 25.80 dB), (e) image denoised with BM3D and the MMSE inverse with the SURE estimate $\varepsilon_{\mathrm{SURE}}$ (average NMISE = 0.0911 and PSNR = 25.91 dB).

for BLS-GSM we use its full steerable pyramid implementation.

Note that the faster execution times for the dual core T8300 are rather explained by the fact that the CPU is much newer than the Pentium 4, as at least the Matlab implementation of BM3D does not take advantage of more than one CPU core.

In comparison, the authors of [146] report that the MS-VST + curvelet denoising of Cells (see Figure 3.7(e)) required 1287 seconds on a 1.1 GHz PC. The authors of [94] do not specify their hardware, but they report the PH-HMT denoising of Cameraman (peak 20) taking 92 seconds with unoptimized Matlab code. Finally, the authors of [98] report the Interscale PURE-LET denoising of Cameraman at 17.25 dB taking only 0.37 seconds with two cyclic shifts and 4.6 seconds with 25 cyclic shifts (hardware not specified). Note that in Table 3.4 we compare our results against the PURE-LET with two cyclic shifts, as similar results for 25 cyclic shifts are not presented in [98].

Regarding our MMSE inverse, even with exploiting some acceleration, the time needed for computing the empirical MSE estimate $\varepsilon^2_{\mathrm{SURE}}$ is hundreds of times higher than that of a single denoising run. Therefore, based on the minor improvements over $\mathcal{I}_0 = \mathcal{I}_{\mathrm{ML}}$ shown in Table 3.5, the practical use of the MMSE inverse appears extremely limited.

## 3.4 Discussion

In this chapter we showed that the three-step procedure of first stabilizing the noise variance by applying the Anscombe transformation, then denoising with an algorithm designed for Gaussian noise, and finally applying an inverse transformation, can still be considered a viable approach for Poisson noise removal. In particular, the poor performance of the asymptotically unbiased inverse at low counts can be overcome by replacing it with the exact unbiased inverse. The excellent performance achieved through the exact unbiased inverse is justified by the fact that this inverse can be interpreted as a maximum likelihood inverse under rather generic hypotheses.

Further, when combined with a state-of-the-art Gaussian denoising algorithm, this method is competitive with some of the best Poisson noise removal algorithms, such as PH-HMT [94], MS-VST [146] and PURE-LET [98]. While most of the improvement is due to the exact unbiased inverse, the choice of the denoising algorithm does also matter, and of the methods considered here BM3D seems to be the best choice due to its overall strong results combined with relatively low complexity.

Moreover, we approximated the proposed exact unbiased inverse with a closed-form expression, and showed that these two inverses can be used interchangeably, with practically equal denoising results. Thus, the traditional algebraic and asymptotically unbiased inverses can easily be replaced by this convenient closed-form expression in practical applications, providing the excellent performance of the exact unbiased inverse.

We also proposed an MMSE inverse parametrized by the pointwise MSE of the denoised stabilized data. While this inverse is not suitable for practical applications with generic filters, we argue that it can be relevant within specific

implementations where more knowledge about the statistics of the estimates is available.

In connection with our contributions, we would like to mention the work of Neyman and Scott [116] on the unbiased inversion of transformed and stabilized variables, and highlight the aspects that make their results different than ours. First, in [116] it is assumed that the transformed variables are exactly stabilized and normalized, which (as the authors also point out) is not possible for Poisson distributed variables stabilized by a root transformation. Instead, we always provide a fully accurate statistical modelling of the distribution of the stabilized Poisson variables $f(z)$. Second, in the case when the estimation is inaccurate, even though we and they both assume a normal distribution of the estimate $D$, the inverse sought by them is an unbiased one (minimum variance unbiased estimate) whereas we address this case by the ML and MMSE inverses.

Table 3.1: Average NMISE values for the asymptotically unbiased inverse and the exact unbiased inverse, and a comparison to the results obtained in [94] and [146] with algorithms specifically designed for Poisson noise removal. The intensity range of each image is indicated in brackets.

| | Asymptotically unbiased inverse | | | | Exact unbiased inverse | | | Other algorithms | |
|---|---|---|---|---|---|---|---|---|---|
| | WT [146] | BM3D | SAFIR | BLS-GSM | BM3D | SAFIR | BLS-GSM | PH-HMT [94] | MS-VST [146] |
| Spots [0.03, 5.02] | 2.34 | 1.7424 | 1.7495 | 2.0370 | **0.0365** | 0.0384 | 0.1871 | 0.048 | 0.069 |
| Galaxy [0, 5] | 0.15 | 0.1026 | 0.1110 | 0.1253 | **0.0299** | 0.0301 | 0.0385 | 0.030 | 0.035 |
| Ridges [0.05, 0.85] | 0.83 | 0.7025 | 0.7252 | 0.7694 | **0.0128** | 0.0173 | 0.0331 | - | 0.017 |
| Barbara [0.93, 15.73] | 0.26 | **0.0881** | 0.1178 | 0.1122 | **0.0881** | 0.1178 | 0.1123 | 0.159 | 0.17 |
| Cells [0.53, 16.93] | 0.095 | 0.0660 | 0.0683 | 0.0718 | **0.0649** | 0.0671 | 0.0707 | 0.082 | 0.078 |

Table 3.2: Average NMISE values for the asymptotically unbiased inverse ($\mathcal{I}_{\text{asy}}$), exact unbiased inverse ($\mathcal{I}_0$) and its proposed closed-form approximation ($\widetilde{\mathcal{I}}_0$), each combined with BM3D (see also Table 3.1). The intensity range of each image is indicated in brackets.

| | $\mathcal{I}_{\text{asy}}$ | $\mathcal{I}_0$ | $\widetilde{\mathcal{I}}_0$ |
|---|---|---|---|
| Spots [0.03, 5.02] | 1.7424 | **0.0365** | **0.0365** |
| Galaxy [0, 5] | 0.1026 | **0.0299** | **0.0299** |
| Ridges [0.05, 0.85] | 0.7025 | **0.0128** | 0.0129 |
| Barbara [0.93, 15.73] | **0.0881** | **0.0881** | 0.0884 |
| Cells [0.53, 16.93] | 0.0660 | **0.0649** | **0.0649** |

Table 3.3: Average PSNR values (dB) for various peak intensities for the asymptotically unbiased inverse and the exact unbiased inverse, and a comparison to the results obtained in [94] with PH-HMT, an algorithm specifically designed for Poisson noise removal.

| | | | Asympt. unbiased inverse | | | Exact unbiased inverse | | | Other |
|---|---|---|---|---|---|---|---|---|---|
| Image | Peak | Noisy | BM3D | SAFIR | BLS-GSM | BM3D | SAFIR | BLS-GSM | PH-HMT |
| Camera-man | 1 | 3.27 | 14.90 | 14.97 | 14.37 | 19.89 | **20.37** | 18.44 | 20.03 |
| | 2 | 6.26 | 20.49 | 20.21 | 18.94 | **22.10** | 21.88 | 20.11 | 21.41 |
| | 3 | 8.05 | 22.30 | 22.12 | 20.69 | **23.07** | 22.87 | 21.22 | 22.31 |
| | 4 | 9.28 | 23.40 | 23.25 | 21.67 | **23.86** | 23.71 | 21.98 | 22.90 |
| | 5 | 10.27 | 24.12 | 24.01 | 22.36 | **24.42** | 24.31 | 22.57 | 23.37 |
| | 10 | 13.26 | 26.03 | 25.83 | 24.52 | **26.08** | 25.89 | 24.57 | 24.97 |
| | 20 | 16.29 | 27.65 | 27.31 | 26.49 | **27.65** | 27.31 | 26.49 | 26.61 |
| Lena | 1 | 2.96 | 16.13 | 16.26 | 15.79 | 22.22 | **23.41** | 21.52 | 22.66 |
| | 2 | 5.99 | 22.88 | 23.00 | 21.98 | 24.07 | **24.77** | 23.42 | 23.91 |
| | 3 | 7.75 | 24.91 | 25.28 | 24.24 | 25.23 | **25.74** | 24.67 | 24.69 |
| | 4 | 9.00 | 25.96 | 26.32 | 25.43 | 26.06 | **26.45** | 25.59 | 25.29 |
| | 5 | 9.96 | 26.53 | 26.88 | 26.15 | 26.56 | **26.91** | 26.21 | 25.78 |
| | 10 | 12.97 | 28.31 | 28.51 | 28.03 | 28.31 | **28.51** | 28.03 | 27.21 |
| | 20 | 15.98 | 29.99 | 30.03 | 29.60 | 29.99 | **30.03** | 29.59 | 28.66 |
| Boat | 1 | 2.93 | 15.88 | 15.87 | 15.56 | 20.97 | 21.41 | 20.49 | **21.76** |
| | 2 | 5.95 | 21.80 | 21.57 | 20.90 | 22.74 | 22.76 | 21.98 | **22.77** |
| | 3 | 7.71 | 23.36 | 23.28 | 22.69 | **23.67** | 23.62 | 23.04 | 23.45 |
| | 4 | 8.96 | 24.15 | 24.07 | 23.63 | **24.28** | 24.20 | 23.79 | 23.90 |
| | 5 | 9.92 | 24.71 | 24.63 | 24.33 | **24.77** | 24.70 | 24.42 | 24.31 |
| | 10 | 12.94 | 26.27 | 26.06 | 25.93 | **26.28** | 26.07 | 25.94 | 25.57 |
| | 20 | 15.94 | 27.83 | 27.44 | 27.41 | **27.83** | 27.44 | 27.41 | 26.96 |
| Barbara | 1 | 3.20 | 15.28 | 15.33 | 14.95 | 20.43 | **20.78** | 19.64 | 20.48 |
| | 2 | 6.23 | 20.88 | 20.50 | 19.87 | **21.91** | 21.67 | 20.93 | 21.27 |
| | 3 | 7.99 | 22.74 | 21.88 | 21.31 | **23.07** | 22.18 | 21.62 | 21.72 |
| | 4 | 9.24 | 23.71 | 22.38 | 21.90 | **23.84** | 22.48 | 22.01 | 22.07 |
| | 5 | 10.21 | 24.42 | 22.68 | 22.57 | **24.48** | 22.72 | 22.62 | 22.33 |
| | 10 | 13.21 | 26.35 | 24.22 | 24.67 | **26.35** | 24.22 | 24.67 | 23.45 |
| | 20 | 16.22 | 28.18 | 26.91 | 26.50 | **28.18** | 26.91 | 26.50 | 24.92 |
| Finger-print | 1 | 2.55 | 14.61 | 14.28 | 13.89 | 17.12 | 16.61 | 15.98 | **17.39** |
| | 2 | 5.57 | 19.55 | 19.23 | 18.87 | **19.86** | 19.59 | 19.36 | 18.55 |
| | 3 | 7.34 | 20.93 | 20.80 | 20.52 | **20.98** | 20.85 | 20.64 | 19.36 |
| | 4 | 8.58 | 21.68 | 21.54 | 21.34 | **21.69** | 21.54 | 21.39 | 19.94 |
| | 5 | 9.54 | 22.22 | 22.03 | 21.89 | **22.22** | 22.03 | 21.91 | 20.42 |
| | 10 | 12.56 | 23.80 | 23.35 | 23.38 | **23.80** | 23.35 | 23.38 | 21.91 |
| | 20 | 15.57 | 25.37 | 24.62 | 24.94 | **25.37** | 24.62 | 24.94 | 23.46 |

Table 3.4: Average PSNR values (dB) for various peak intensities for the asymptotically unbiased inverse and the exact unbiased inverse, and a comparison to the best results obtained in [98] with the Interscale PURE-LET, an algorithm specifically designed for Poisson noise removal.

| | | | Asympt. unbiased inverse | | | Exact unbiased inverse | | | Other |
|---|---|---|---|---|---|---|---|---|---|
| Image | Peak | Noisy | BM3D | SAFIR | BLS-GSM | BM3D | SAFIR | BLS-GSM | PURE-LET |
| Peppers | 1 | 3.16 | 15.20 | 15.30 | 14.67 | 19.98 | **20.34** | 18.44 | 19.33 |
| | 5 | 10.13 | 24.59 | 24.89 | 23.47 | 24.70 | **25.03** | 23.57 | 22.52 |
| | 10 | 13.15 | 26.41 | 26.49 | 25.39 | 26.43 | **26.51** | 25.41 | 24.29 |
| | 20 | 16.15 | 28.05 | 28.02 | 27.00 | **28.05** | 28.03 | 27.00 | 26.18 |
| | 30 | 17.92 | 29.05 | 28.93 | 27.97 | **29.05** | 28.93 | 27.97 | 27.27 |
| | 60 | 20.92 | 30.75 | 30.48 | 29.62 | **30.75** | 30.48 | 29.62 | 29.07 |
| | 120 | 23.94 | 32.47 | 32.10 | 31.36 | **32.47** | 32.10 | 31.36 | 30.79 |
| Camera-man | 1 | 3.27 | 14.90 | 14.97 | 14.37 | 19.89 | **20.37** | 18.44 | 19.67 |
| | 5 | 10.26 | 24.07 | 23.99 | 22.39 | **24.36** | 24.28 | 22.59 | 22.76 |
| | 10 | 13.26 | 26.05 | 25.87 | 24.52 | **26.11** | 25.93 | 24.57 | 24.32 |
| | 20 | 16.26 | 27.65 | 27.30 | 26.49 | **27.65** | 27.30 | 26.49 | 25.89 |
| | 30 | 18.06 | 28.56 | 28.14 | 27.51 | **28.56** | 28.14 | 27.50 | 26.87 |
| | 60 | 21.05 | 30.04 | 29.41 | 29.10 | **30.04** | 29.41 | 29.10 | 28.56 |
| | 120 | 24.07 | 31.66 | 30.76 | 30.78 | **31.66** | 30.76 | 30.78 | 30.36 |
| MIT | 1 | 5.00 | 13.41 | 13.14 | 12.28 | **19.17** | 17.90 | 15.67 | 17.82 |
| | 5 | 11.97 | 23.81 | 24.19 | 21.19 | 24.43 | **24.88** | 21.52 | 21.63 |
| | 10 | 14.98 | 25.99 | 26.20 | 23.32 | 26.16 | **26.37** | 23.41 | 23.49 |
| | 20 | 18.01 | 27.88 | 27.76 | 25.62 | **27.93** | 27.80 | 25.64 | 25.34 |
| | 30 | 19.78 | 28.96 | 28.64 | 26.84 | **28.98** | 28.65 | 26.84 | 26.55 |
| | 60 | 22.78 | 30.81 | 30.18 | 28.80 | **30.81** | 30.18 | 28.80 | 28.41 |
| | 120 | 25.77 | 32.63 | 31.89 | 30.71 | **32.63** | 31.89 | 30.71 | 30.47 |
| Moon | 1 | 5.46 | 13.96 | 14.03 | 13.98 | 22.64 | **23.34** | 23.08 | 23.19 |
| | 5 | 12.46 | 23.66 | 23.45 | 23.17 | **24.33** | 24.11 | 23.79 | 24.28 |
| | 10 | 15.48 | 24.85 | 24.58 | 24.21 | **25.05** | 24.76 | 24.38 | 24.99 |
| | 20 | 18.47 | 25.86 | 25.56 | 25.50 | 25.92 | 25.62 | 25.56 | **25.97** |
| | 30 | 20.24 | 26.50 | 26.18 | 26.35 | 26.53 | 26.21 | 26.38 | **26.70** |
| | 60 | 23.25 | 27.80 | 27.36 | 28.00 | 27.81 | 27.37 | 28.01 | **28.09** |
| | 120 | 26.25 | 29.45 | 28.82 | 29.80 | 29.45 | 28.83 | **29.80** | 29.77 |

Table 3.5: Average NMISE and PSNR (dB) values for the exact unbiased inverse, MMSE inverse with the oracle Monte-Carlo estimate $\varepsilon_{\text{MC}}$, and MMSE inverse with the SURE estimate $\varepsilon_{\text{SURE}}$. The intensity range of each image is indicated in brackets.

| | NMISE | | | PSNR (dB) | | |
|---|---|---|---|---|---|---|
| | Exact unbiased | MMSE $\varepsilon_{\text{MC}}$ | MMSE $\varepsilon_{\text{SURE}}$ | Exact unbiased | MMSE $\varepsilon_{\text{MC}}$ | MMSE $\varepsilon_{\text{SURE}}$ |
| Spots [0.03, 5.02] | 0.0365 | **0.0324** | 0.0362 | 31.96 | **34.65** | 32.81 |
| Galaxy [0, 5] | 0.0299 | **0.0241** | 0.0308 | 28.05 | **30.06** | 28.16 |
| Ridges [0.05, 0.85] | **0.0128** | 0.0129 | 0.0129 | 25.89 | **25.98** | 25.94 |
| Barbara [0.93, 15.73] | **0.0881** | 0.0971 | 0.0911 | **25.92** | 25.80 | 25.91 |
| Cells [0.53, 16.93] | **0.0649** | 0.0686 | 0.0668 | 30.19 | **30.23** | 30.18 |

Table 3.6: Average computation times of the denoising procedures and transformations for the Lena (512 × 512) and Cameraman (256 × 256) images, for two different CPUs. For SAFIR we use the parameters giving the best denoising results, which significantly increases the computation time, and for BLS-GSM we use its full steerable pyramid implementation.

| CPU | Image | BM3D | SAFIR | BLS-GSM |
|---|---|---|---|---|
| Intel Pentium 4 HT 560 (single core) @ 3.6 GHz | Lena | 6.3 s | 60 min | 68 s |
| | Cameraman | 1.6 s | 14 min | 17 s |
| Intel Mobile Core 2 Duo T8300 @ 2.4 GHz | Lena | 4.1 s | 43 min | - |
| | Cameraman | 1.0 s | 9.5 min | - |

| CPU | Image | Forward Anscombe (transformation only) | Exact unbiased inverse (transformation only) |
|---|---|---|---|
| Intel Pentium 4 HT 560 (single core) @ 3.6 GHz | Lena | 0.07 s | 0.22 s |
| | Cameraman | 0.014 s | 0.063 s |
| Intel Mobile Core 2 Duo T8300 @ 2.4 GHz | Lena | 0.025 s | 0.15 s |
| | Cameraman | 0.006 s | 0.044 s |

# Chapter 4

# Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise

## 4.1 Introduction

In the case of Poisson-Gaussian noise, the generalized Anscombe transformation (2.18) is commonly used for stabilizing the noise variance. Even though the GAT is a well-known transformation, its corresponding exact unbiased inverse transformation has been neglected in the past. In the previous chapter we proposed the exact unbiased inverse of the Anscombe transformation for pure Poisson noise. This chapter generalizes those results into the exact unbiased inverse of the GAT, and shows that it plays an integral part in ensuring accurate denoising results, similarly to the Poisson case.

To the best of our knowledge, this is the first rigorous study of the exact unbiased inverse of the GAT for Poisson-Gaussian noise. Importantly, our results are exact and successfully applicable also for finite parameter values, while previous works [126] have considered only the large-parameter asymptotic case. Moreover, our experimental results demonstrate that our approach leads to state-of-the-art denoising results, when the exact unbiased inverse is combined with a state-of-the-art Gaussian denoising algorithm. We also show that this inverse is optimal in the sense that it can be interpreted as a maximum likelihood inverse under certain reasonable assumptions. Further, we provide a thorough analysis of the behaviour of the proposed inverse. In particular, we show that the exact unbiased inverse of the GAT can be approximated with great accuracy by adding a simple correction term to the exact unbiased inverse of the Anscombe transformation. This also enables us to derive a closed-form approximation for it, by adding the same correction term to the closed-form approximation (3.7) of the exact unbiased inverse of the Anscombe transformation. Finally, we provide asymptotic, as well as global

integral and global supremum error bounds for both of these approximations.

This chapter presents our research results first published in [114] and [113]. It is also worth noting that the exact unbiased inverse of the GAT is conceptually similar to the exact unbiased inverse of the Anscombe transformation. Thus, employing the exact unbiased inverse remains computationally very inexpensive also in this generalization. In other words, the proposed approach does not introduce any notable computational overhead; the choice of the Gaussian denoising algorithm completely dominates the execution time. These observations are consistent with the discussion on computational complexity in Section 3.3.3.

This chapter is organized as follows: Section 4.2 presents a method of reducing the number of parameters in the Poisson-Gaussian model from three to one, aiding in the practical construction of the exact unbiased inverse transformation. In Section 4.3, which is the core of our contribution, we discuss how to construct an exact unbiased inverse of the GAT, the optimality and asymptotic behaviour of this inverse, and how to approximate it with a closed-form expression. Section 3.3 consists of various experiments. In particular, first we examine the case of denoising with known noise parameters, after which we consider the case with unknown parameter values. Then we inspect how the denoising performance changes, when the ratio between the Poisson and Gaussian noise component changes. We conclude the section with a brief commentary on the computational complexity of the proposed method. Finally, in Section 3.4, we discuss the obtained results.

## 4.2 Parameter reduction

As implied by (2.7), the Poisson-Gaussian model is parametrized by three variables: the positive scaling factor $\alpha$ of the Poisson component, the mean $\mu$ of the Gaussian component, and the standard deviation $\sigma$ of the Gaussian component. However, for the construction of a family of exact unbiased inverses for the GAT, dealing with three parameters is rather impractical. Thus, we devise a simple way of modifying the data through linear transformations, allowing us to parametrize the inverse only by $\sigma$.

For notational purposes, we employ temporary variables $\grave{z}_i$, $\grave{\sigma}$ and $\grave{n}_i$ to replace the variables $z_i$, $\sigma$ and $n_i$ in (2.7). In other words, each pixel $\grave{z}_i, i = 1, \ldots, N$ is modelled as a realization of an independent random Poisson variable $p_i$ with an underlying mean value $y_i$, scaled by $\alpha > 0$ and corrupted by additive Gaussian noise $\grave{n}_i$ of mean $\mu$ and standard deviation $\grave{\sigma}$. Hence, (2.7) equals

$$\grave{z}_i = \alpha p_i + \grave{n}_i, \tag{4.1}$$

where $p_i \sim \mathcal{P}(y_i)$ and $\grave{n}_i \sim \mathcal{N}(\mu, \grave{\sigma}^2)$. Thus, we can define Poisson-Gaussian noise as

$$\eta_i = \grave{z}_i - \alpha y_i. \tag{4.2}$$

The problem of denoising an image corrupted by Poisson-Gaussian noise is then equivalent to estimating the underlying noise-free image $y$ given the noisy observations $\grave{z}$.

For clarity, we note that the use of the grave accent in $\grave{z}$, $\grave{\sigma}$ and $\grave{n}_i$ is due to the affine transformations (4.4), which reduce the number of parameters to be

considered from three to one. Thus, the rest of the chapter will only deal with the corresponding transformed variables $z$, $\sigma$ and $n_i$, and with $y$ and $p_i$, unless noted otherwise.

Assuming $\grave{z}$ is distributed according to (4.1), the GAT (2.18) can be written as

$$f(\grave{z}) = \begin{cases} \frac{2}{\alpha}\sqrt{\alpha\grave{z} + \frac{3}{8}\alpha^2 + \grave{\sigma}^2 - \alpha\mu}, & \grave{z} > -\frac{3}{8}\alpha - \frac{\grave{\sigma}^2}{\alpha} + \mu \\ 0, & \grave{z} \leq -\frac{3}{8}\alpha - \frac{\grave{\sigma}^2}{\alpha} + \mu \end{cases} . \tag{4.3}$$

Hence, applying (4.3) to $\grave{z}$ (approximately) stabilizes its variance to unity. However, the number of parameters which define the transformation (4.3) can be reduced significantly by simple variable substitutions

$$z = \frac{\grave{z} - \mu}{\alpha}, \quad \sigma = \frac{\grave{\sigma}}{\alpha}, \tag{4.4}$$

which affinely map each pixel $\grave{z}_i$ to $z_i$, a random (non-scaled) Poisson variable $p_i$ corrupted by additive Gaussian noise $n_i$ of mean 0 and standard deviation $\sigma$:

$$z_i = p_i + n_i, \tag{4.5}$$

where $p_i \sim \mathcal{P}(y_i)$ and $n_i \sim \mathcal{N}(0, \sigma^2)$. In particular, the probability distribution of $z$ is

$$p(z \mid y, \sigma) = \sum_{k=0}^{+\infty} \left( \frac{y^k e^{-y}}{k!} \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-k)^2}{2\sigma^2}} \right). \tag{4.6}$$

Thus, according to (4.3), $z$ can be stabilized with the transformation

$$f_\sigma(z) = \begin{cases} 2\sqrt{z + \frac{3}{8} + \sigma^2}, & z > -\frac{3}{8} - \sigma^2 \\ 0, & z \leq -\frac{3}{8} - \sigma^2 \end{cases} . \tag{4.7}$$

In other words, for any of the parameters $\alpha$ and $\mu$, we can stabilize the variance of $\grave{z}$ by means of variable substitutions (4.4), followed by the transformation (4.7). Then, after applying an inverse transformation $\mathcal{I}$ of (4.7) to the denoised data $D$, we simply return to the original range by inverting (4.4), i.e. setting the final estimate of the expected value of $\grave{z}$ to be $\alpha\mathcal{I}(D) + \mu$. Note that since (4.4) and its inverse are affine, they do not introduce any bias in the estimation. Thus, in the rest of this chapter, we consider only the stabilization of $z$, which is the observed data after the variable substitution (4.4), and by GAT we refer to the corresponding transformation (4.7).

Figure 4.1(a) shows the forward transformation (4.7) for the parameter values $\sigma = 0.01, 1, 2, 3$, and the corresponding standard deviations of the stabilized variables $f_\sigma(z)$ are shown in Figure 4.1(b). Note that there is a particular overshoot in the standard deviation at around $\sigma = 2$ for low values of $y$, but it begins to settle down towards the desired value 1 as $\sigma$ increases. On the other hand, when $\sigma$ is very small (i.e., when the noise is practically pure Poisson), there is a significant undershoot for low values of $y$, which is an inherent limitation of the generalized Anscombe transformation. Regardless of which transformation is used, it is not possible to stabilize the variance to unity for $y = 0$, $\sigma = 0$, since in this case $z$ has zero variance.

## 4.3 Exact unbiased inverse transformation

### 4.3.1 Definition

Assuming the denoising of $f_\sigma(z)$ was successful, we may treat the denoised data $D$ as the expected value $E\{f_\sigma(z) \mid y, \sigma\}$. However, just like in the Poisson case, the nonlinearity of $f_\sigma$ means that we generally have

$$f_\sigma^{-1}\left(E\{f_\sigma(z) \mid y\}\right) \neq E\{z \mid y\}. \tag{4.8}$$

In practise, this means that applying the algebraic inverse $f_\sigma^{-1}$ to the denoised data will, in general, produce a biased estimate of $y$. Once again we see that using an asymptotically unbiased inverse or the algebraic inverse is not sufficient for guaranteeing accurate denoising results, in particular for low-intensity images. Thus, we propose an exact unbiased inverse that can be used in all situations.

The exact unbiased inverse of the generalized Anscombe transformation (4.7) is in fact a family of inverse transformations $\mathcal{I}_\sigma$, parametrized by $\sigma$, that maps the values $E\{f_\sigma(z) \mid y, \sigma\}$ to the desired values $E\{z \mid y, \sigma\}$:

$$\mathcal{I}_\sigma : E\{f_\sigma(z) \mid y, \sigma\} \longmapsto E\{z \mid y, \sigma\}. \tag{4.9}$$

Since we trivially know $E\{z \mid y, \sigma\} = y$ for any given $y$, constructing the inverse requires us to compute the values $E\{f_\sigma(z) \mid y, \sigma\}$, analogously to how the exact unbiased inverse of the Anscombe transformation was computed in Chapter 3. In this more general case, it is computed as

$$\begin{aligned} E\{f_\sigma(z) \mid y, \sigma\} &= \int_{-\infty}^{+\infty} f_\sigma(z)\, p(z \mid y, \sigma)\, dz \qquad (4.10)\\ &= \int_{-\infty}^{+\infty} 2\sqrt{z + \frac{3}{8} + \sigma^2} \sum_{k=0}^{+\infty} \left( \frac{y^k e^{-y}}{k!\sqrt{2\pi\sigma^2}} e^{-\frac{(z-k)^2}{2\sigma^2}} \right) dz. \end{aligned}$$

The exact unbiased inverse transformations $\mathcal{I}_\sigma$ for the parameter values $\sigma = 0.01, 1, 2, 3$ are shown in Figure 4.1(c), along with the corresponding algebraic inverses of (4.7) for the comparison. Note that even though visually $\mathcal{I}_\sigma$ resembles a clipped algebraic inverse to some extent, replacing the standard algebraic inverse by its clipped counterpart does not provide any practical improvement.

### 4.3.2 Optimality

Under certain reasonable assumptions, we can show that the exact unbiased inverse $\mathcal{I}_\sigma$ is optimal in the sense that it coincides with a maximum likelihood (ML) inverse. In Section 3.2.3, we made a similar conclusion about the exact unbiased Anscombe inverse for the pure Poisson case, and thus, the discussion below follows naturally the same lines.

Let us assume that instead of obtaining a perfectly accurate denoising result $D = E\{f_\sigma(z) \mid y, \sigma\}$ (which we assumed in the definition of $\mathcal{I}_\sigma$ in Section 4.3.1), the pointwise mean squared error of our estimate $D$ is

$$\varepsilon^2 = E\left\{ \left(D - E\{f_\sigma(z) \mid y, \sigma\}\right)^2 \right\} > 0. \tag{4.11}$$

In practise the probability density function $p\left(D \mid y\right)$ of $D$ is unknown, but for simplicity we assume that it is symmetric and unimodal, with mode at $E\left\{f_\sigma\left(z\right) \mid y, \sigma\right\}$ and variance $\varepsilon^2$. Note that the actual value of $\varepsilon^2$ has very little to do with the variance of $z$ or $f_\sigma\left(z\right)$, since the estimation accuracy of the denoising algorithm depends mainly on the features of the signal (such as the complexity of details and patterns); thus, it is sensible to treat $\varepsilon^2$ as independent of both $y$ and $\sigma$.

In other words, we assume

$$\xi = D - E\left\{f_\sigma(z)|y,\sigma\right\} \sim U_0, \tag{4.12}$$

where $U_0$ is a unimodal distribution with mode at 0 with probability density $u_0$ (e.g., $U_0$ could be the normal $\mathcal{N}(0,\varepsilon^2)$). This implies that $u_0\left(\bar{\xi}\right) \leq u_0\left(\breve{\xi}\right)$, if either $\bar{\xi} \leq \breve{\xi} \leq 0$ or $\bar{\xi} \geq \breve{\xi} \geq 0$; in particular $u_0\left(0\right) = \max_\xi\left(u_0\right)$, and $u_0\left(D - E\left\{f_\sigma(z)|y,\sigma\right\}\right) = p\left(D|y\right)$. Let us also remark that, for any $\sigma \geq 0$, $E\left\{f_\sigma(z)|y,\sigma\right\}$ is a continuous and monotonically increasing function of $y$.

While (4.12) formally implies that $D$ is an unbiased estimate of $E\left\{f_\sigma(z)|y,\sigma\right\}$, in fact also unknown estimation-bias errors can be considered as contributors of $\varepsilon^2$, with the symmetry of the distribution about $E\left\{f_\sigma(z)|y,\sigma\right\}$ reflecting our uncertainty about the sign of the bias.

By treating $D$ as the data, the ML estimate of $y$ is defined as

$$\mathcal{I}_{\mathrm{ML}}(D) = \arg\max_y p\left(D \mid y\right). \tag{4.13}$$

Under the above assumptions, this equals to

$$\mathcal{I}_{\mathrm{ML}}(D) = \begin{cases} \mathcal{I}_\sigma(D), \text{ if } D \geq E\{f_\sigma\left(z\right) \mid 0,\sigma\} \\ 0, \text{ if } D < E\{f_\sigma\left(z\right) \mid 0,\sigma\} \end{cases}, \tag{4.14}$$

with the proof being essentially identical to the proof of (3.12) presented in Section 3.2.3 for the pure Poisson case. In other words, also the exact unbiased inverse $\mathcal{I}_\sigma$ coincides with this form of ML inverse.

### 4.3.3 Asymptotic behaviour

As we explicitly construct the inverse mapping (4.9) only for a finite grid of values, it is also of interest to examine the asymptotic behaviour of $\mathcal{I}_\sigma$. When the standard deviation $\sigma$ of the Gaussian noise component is large, we may formulate the exact unbiased inverse $\mathcal{I}_\sigma$ in terms of the exact unbiased inverse Anscombe transformation $\mathcal{I}_0$ as

$$\mathcal{I}_\sigma \approx \mathcal{I}_0 - \sigma^2. \tag{4.15}$$

Likewise, when $\sigma$ is very small, we may do the same approximation. A rigorous derivation of this approximation is provided below.

In particular, we analyze the asymptotic behaviour of the exact unbiased inverse $\mathcal{I}_\sigma$ of the generalized Anscombe transformation $f_\sigma$ (4.7) for large and small values of $\sigma$, as well as for large values of its argument $D$. Further, as a result of our analysis, we also obtain a closed-form approximation of $\mathcal{I}_\sigma$, which can be used for any value of $D$ and $\sigma$.

**Expansion about the mean**

Let

$$f_\sigma(z) = \begin{cases} 2\sqrt{z+\frac{3}{8}+\sigma^2}, & z > -\frac{3}{8}-\sigma^2 \\ 0, & z \leq -\frac{3}{8}-\sigma^2 \end{cases} . \tag{4.16}$$

Following [3], we consider the Taylor expansion of $f_\sigma(z)$ about the mean $y = E\{z\} \geq 0$ of $z$. Defining $t = z - y$ and $\breve{y} = y + \frac{3}{8} + \sigma^2$, we have

$$f_\sigma(z) = \begin{cases} 2\sqrt{\breve{y}}\left(1 + \frac{1}{2}\frac{t}{\breve{y}} - \frac{1}{8}\left(\frac{t}{\breve{y}}\right)^2 + \right. & \\ \quad \frac{1}{16}\left(\frac{t}{\breve{y}}\right)^3 - \frac{5}{128}\left(\frac{t}{\breve{y}}\right)^4 + & t > -\breve{y}, \\ \quad \left. \ldots + (-1)^s a_{s-1}\left(\frac{t}{\breve{y}}\right)^{s-1} + R_s\right) & \\ 0 & t \leq -\breve{y}, \end{cases} \tag{4.17}$$

where

$$a_s = (-1)^{s+1}\frac{1\cdot(-1)\cdot(-3)\cdot\ldots\cdot(-2s+3)}{2^s s!} = -\prod_{j=1}^{s} 1 - \frac{3}{2j} = \mathcal{O}\left(\frac{1}{s^{3/2}}\right)$$

and the series converges for $|t| \leq \breve{y}$, with the remainder term $R_s$ being

$$R_s = \mathcal{O}\left(\left(\frac{t}{\breve{y}}\right)^s\right)$$

for all $t \in \mathbb{R}$. This expansion is illustrated in Figure 4.2.

**Moments**

Let $\mu_k$ be the $k$-th centered moment of $z$. In particular,

$$\mu_1 = 0, \quad \mu_2 = y + \sigma^2, \quad \mu_3 = y, \quad \mu_4 = y + 3\left(y + \sigma^2\right)^2 .$$

More in general, $\mu_k$ can be expressed as a polynomial of order at most $\frac{k}{2}$ with non-negative coefficients in the $k-1$ cumulants $\{\kappa_j\}_{j=2}^k$ of $z$ (see [83, Section 3.13, pp. 61–63]). In turn, each cumulant $\kappa_j$ can expressed as the sum of the $j$-th cumulants $\kappa_j^{\mathcal{P}(y)}$ and $\kappa_j^{\mathcal{N}\left(0,\sigma^2\right)}$of the Poisson $\mathcal{P}(y)$ and Gaussian distribution $\mathcal{N}\left(0,\sigma^2\right)$, respectively. In particular, for the Poisson we have $\kappa_j^{\mathcal{P}(y)} = y\ \forall j$, while for the Gaussian $\kappa_2^{\mathcal{N}\left(0,\sigma^2\right)} = \sigma^2$ and $\kappa_j^{\mathcal{N}\left(0,\sigma^2\right)} = 0\ \forall j \neq 2$ [83]. This means that $\kappa_2 = y + \sigma^2$ and $\kappa_j = y\ \forall j \neq 2$, and hence that $\mu_k$ is a polynomial in $y$ and $y + \sigma^2$, which we denote as

$$\mu_k = P_k\left(y, y + \sigma^2\right). \tag{4.18}$$

Let us consider now the order of the centered moments and absolute centered moments. As observed in [3], the $k$-th absolute centered moment $\nu_k^{\mathcal{P}(m)}$ of a Poisson variable with mean $m$ is of order $\mathcal{O}\left(m^{\frac{1}{2}k}\right)$ as $m \to +\infty$. The same

order applies also to the centered moments $\mu_k^{\mathcal{P}(m)}$ of the same Poisson variable. Since $\kappa_j \leq y + \sigma^2 < y + \frac{3}{8} + \sigma^2 = \breve{y}$ and because moments are polynomials with non-negative coefficients in the cumulants, we have that

$$\mu_k^{\mathcal{P}(y)} \leq \mu_k \leq \mu_k^{\mathcal{P}\left(y+\sigma^2\right)} \leq \mu_k^{\mathcal{P}(\breve{y})},$$

whence we obtain that $\mu_k$ is of at most order $\mathcal{O}\left(\breve{y}^{\frac{1}{2}k}\right)$ as $\breve{y} \to +\infty$. The same asymptotic order extends to the absolute moments $\nu_k$, because, for any odd $k$, $\nu_{k-1} = \mu_{k-1}$ and $\nu_k \leq (\mu_{k+1})^{k/(k+1)}$ (see [83, Section 3.6, p. 56]).

**Inverse mappings**

Taking expectations on both sides of (4.17), we obtain

$$E\{f_\sigma(z)\,|y,\sigma\} = 2\sqrt{\breve{y}}\left(1 - \frac{1}{8}\frac{\mu_2}{\breve{y}^2} + \frac{1}{16}\frac{\mu_3}{\breve{y}^3} - \frac{5}{128}\frac{\mu_4}{\breve{y}^4} + \cdots + (-1)^s\, a_{s-1}\frac{\mu_{s-1}}{\breve{y}^{s-1}} + \mathcal{O}\left(\frac{\mu_s}{\breve{y}^s}\right)\right). \tag{4.19}$$

By its very definition, the exact unbiased inverse $\mathcal{I}_\sigma$ of the generalized Anscombe transformation $f_\sigma$ (4.7) is the mapping

$$E\{f_\sigma(z)\,|y,\sigma\} \overset{\mathcal{I}_\sigma}{\longmapsto} y,$$

i.e.,

$$2\sqrt{y+\tfrac{3}{8}+\sigma^2}\left(1 - \frac{1}{8}\frac{y+\sigma^2}{\left(y+\frac{3}{8}+\sigma^2\right)^2} + \frac{1}{16}\frac{y}{\left(y+\frac{3}{8}+\sigma^2\right)^3} - \frac{5}{128}\frac{y+3\left(y+\sigma^2\right)^2}{\left(y+\frac{3}{8}+\sigma^2\right)^4} + \right.$$
$$\left. \ldots + (-1)^s\, a_{s-1}\frac{P_{s-1}\left(y, y+\sigma^2\right)}{\left(y+\frac{3}{8}+\sigma^2\right)^{s-1}} + \mathcal{O}\left(\frac{1}{\left(y+\frac{3}{8}+\sigma^2\right)^{s/2}}\right)\right) \overset{\mathcal{I}_\sigma}{\longmapsto} y. \tag{4.20}$$

For the particular case $\sigma = 0$, the above inverse reduces to the exact unbiased inverse $\mathcal{I}_0$ of the Anscombe transformation $f_0$, which is hence the mapping

$$2\sqrt{y+\tfrac{3}{8}}\left(1 - \frac{1}{8}\frac{y}{\left(y+\frac{3}{8}\right)^2} + \frac{1}{16}\frac{y}{\left(y+\frac{3}{8}\right)^3} - \frac{5}{128}\frac{y+3y^2}{\left(y+\frac{3}{8}\right)^4} + \right.$$
$$\left. \ldots + (-1)^s\, a_{s-1}\frac{P_{s-1}\left(y, y\right)}{\left(y+\frac{3}{8}\right)^{s-1}} + \mathcal{O}\left(\frac{1}{\left(y+\frac{3}{8}\right)^{s/2}}\right)\right) \overset{\mathcal{I}_0}{\longmapsto} y. \tag{4.21}$$

If we substitute $y$ with $y + \sigma^2$ in (4.21), we obtain

$$2\sqrt{y+\tfrac{3}{8}+\sigma^2}\left(1 - \frac{1}{8}\frac{y+\sigma^2}{\left(y+\frac{3}{8}+\sigma^2\right)^2} + \frac{1}{16}\frac{y+\sigma^2}{\left(y+\frac{3}{8}+\sigma^2\right)^3} - \frac{5}{128}\frac{y+\sigma^2+3\left(y+\sigma^2\right)^2}{\left(y+\frac{3}{8}+\sigma^2\right)^4} + \right.$$
$$\left. \ldots + (-1)^s a_{s-1}\frac{P_{s-1}\left(y+\sigma^2, y+\sigma^2\right)}{\left(y+\frac{3}{8}+\sigma^2\right)^{s-1}} + \mathcal{O}\left(\frac{1}{\left(y+\frac{3}{8}+\sigma^2\right)^{s/2}}\right)\right) \overset{\mathcal{I}_0}{\longmapsto} y+\sigma^2. \tag{4.22}$$

Of course, (4.21) and (4.22) define the same inverse $\mathcal{I}_0$, but the similarities between $\mathcal{I}_\sigma$ and $\mathcal{I}_0$ become more evident when comparing the mapping (4.20) with (4.22).

Indeed, if we subtract $\sigma^2$ from $\mathcal{I}_0$, we obtain a mapping which coincides with $\mathcal{I}_\sigma$ except for some high-order terms in the argument of the mapping:

$$2\sqrt{y+\tfrac{3}{8}+\sigma^2}\left(1-\frac{1}{8}\frac{y+\sigma^2}{\left(y+\frac{3}{8}+\sigma^2\right)^2}+\frac{1}{16}\frac{y+\sigma^2}{\left(y+\frac{3}{8}+\sigma^2\right)^3}-\frac{5}{128}\frac{y+\sigma^2+3\left(y+\sigma^2\right)^2}{\left(y+\frac{3}{8}+\sigma^2\right)^4}+\right.$$
$$\left.\ldots+(-1)^s a_{s-1}\frac{P_{s-1}\left(y+\sigma^2,y+\sigma^2\right)}{\left(y+\frac{3}{8}+\sigma^2\right)^{s-1}}+\mathcal{O}\left(\frac{1}{\left(y+\frac{3}{8}+\sigma^2\right)^{s/2}}\right)\right)\overset{\mathcal{I}_0-\sigma^2}{\longmapsto} y. \quad (4.23)$$

The mappings (4.20) and (4.23) can be written in more compact form as

$$\mathcal{I}_\sigma\left(E\left\{f_\sigma\left(z\right)|y,\sigma\right\}\right)=y, \quad (4.24)$$
$$\mathcal{I}_0\left(E\left\{f_\sigma\left(z\right)|y,\sigma\right\}+\delta\left(y,\sigma\right)\right)-\sigma^2=y, \quad (4.25)$$

where $\delta\left(y,\sigma\right)$ is the difference between the left-hand sides of (4.23) and (4.20):

$$\delta\left(y,\sigma\right)=\frac{1}{8}\frac{\sigma^2}{\left(y+\frac{3}{8}+\sigma^2\right)^{5/2}}-\frac{5}{64}\frac{\sigma^2}{\left(y+\frac{3}{8}+\sigma^2\right)^{7/2}}+$$
$$\ldots+(-1)^s a_{s-1}\cdot 2\cdot\frac{P_{s-1}\left(y+\sigma^2,y+\sigma^2\right)-P_{s-1}\left(y,y+\sigma^2\right)}{\left(y+\frac{3}{8}+\sigma^2\right)^{s-3/2}}+\mathcal{O}\left(\frac{1}{\left(y+\frac{3}{8}+\sigma^2\right)^{(s-1)/2}}\right). \quad (4.26)$$

From (4.26) we can immediately see that $\delta\left(y,\sigma\right)\to 0$ as $y+\frac{3}{8}+\sigma^2\to+\infty$. In particular, $\delta\left(y,\sigma\right)=\mathcal{O}\left(\sigma^{-3}\right)$ as $\sigma\to+\infty$ and $\delta\left(y,\sigma\right)=\mathcal{O}\left(y^{-5/2}\right)$ as $y\to+\infty$ for a fixed $\sigma$.

To consider the case of small $\sigma$, let us first show that we can factor out $\sigma^2$ from the difference $P_k\left(y+\sigma^2,y+\sigma^2\right)-P_k\left(y,y+\sigma^2\right)$. It suffices to examine the differences $d_{i,j}$ between the corresponding monomials of equal order,

$$d_{i,j}=\left(y+\sigma^2\right)^{i+j}-y^i\left(y+\sigma^2\right)^j\qquad i+j\le k-1$$

and observe that, upon expanding the powers, both the left term (minuend) and the right term (subtrahend) yield only one monomial which does not have a factor $\sigma^2$, namely $y^{i+j}$, and that this monomial vanishes in the subtraction. Thus,

$$P_k\left(y+\sigma^2,y+\sigma^2\right)-P_k\left(y,y+\sigma^2\right)=\sigma^2 Q_k\left(y,\sigma^2\right),$$

where $Q_k\left(y,\sigma^2\right)$ is polynomial in $y$ and $\sigma^2$, again of the order $\mathcal{O}\left(\left(y+\frac{3}{8}+\sigma^2\right)^{\frac{1}{2}k}\right)$ as $y+\frac{3}{8}+\sigma^2\to+\infty$ and $\mathcal{O}(1)$ as $\sigma\to 0$. Then, for small $\sigma$

$$\delta(y,\sigma)=\sigma^2\left(\frac{1}{8}\frac{1}{\left(y+\frac{3}{8}+\sigma^2\right)^{5/2}}-\frac{5}{64}\frac{1}{\left(y+\frac{3}{8}+\sigma^2\right)^{7/2}}+\right.$$
$$\left.\ldots+(-1)^s a_{s-1}\frac{2Q_{s-1}\left(y,\sigma^2\right)}{\left(y+\frac{3}{8}+\sigma^2\right)^{s-3/2}}+\mathcal{O}(1)\right), \quad (4.27)$$

and therefore $\delta\left(y,\sigma\right)=\mathcal{O}\left(\sigma^2\right)$ as $\sigma\to 0$.

In conclusion, noting that $\mathcal{I}_0$ is a smooth function with derivative $\mathcal{I}_0'$, (4.24) and (4.25) yield

$$\mathcal{I}_\sigma(E\{f_\sigma(z)\,|y,\sigma\}) = \mathcal{I}_0(E\{f_\sigma(z)\,|y,\sigma\}) - \sigma^2 + \mathcal{O}(\mathcal{I}_0'(E\{f_\sigma(z)\,|y,\sigma\}))\,\mathcal{O}(\delta(y,\sigma))\,,$$

where the orders for $\delta\,(y,\sigma)$ found above apply.

Equation (4.20) shows also that $E\{f_\sigma(z)\,|y,\sigma\}$ is large only if $y$ or $\sigma$ are large, i.e., $E\{f_\sigma(z)\,|y,\sigma\} \to +\infty$ if and only if $\breve{y} = y + \frac{3}{8} + \sigma^2 \to +\infty$. Therefore, the above analysis implies that $\mathcal{I}_\sigma(D)$ approaches $\mathcal{I}_0(D) - \sigma^2$ for large $D$ as well as for large or small $\sigma$. We make this statement more precise in the next section.

**Direct asymptotics with respect to $D$ and $\sigma$**

Let $D = E\{f_\sigma(z)\,|y,\sigma\}$. For large $\breve{y}$, we see from (4.19) that $\breve{y} \sim D^2$. Hence, (4.26) yields $\delta(y,\sigma) = \mathcal{O}\left(\sigma^2 D^{-5}\right)$. Since the derivative $\mathcal{I}_0'(D)$ approaches $\frac{D}{2}$ for large $D$, we then have

$$y = \mathcal{I}_\sigma(D) = \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(\sigma^2 D^{-4}\right). \tag{4.28}$$

For a given fixed $\sigma$ (and thus considering large $y$), this becomes

$$y = \mathcal{I}_\sigma(D) = \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(D^{-4}\right), \tag{4.29}$$

whereas for a fixed $y$ (and thus considering large $\sigma$), we have $\sigma^2 \sim \breve{y} \sim D^2$ and, consequently,

$$\begin{aligned} y = \mathcal{I}_\sigma(D) &= \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(D^{-2}\right) \\ &= \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(\sigma^{-2}\right). \end{aligned} \tag{4.30}$$

Finally, for fixed $y$ and $\sigma \to 0$, we have

$$y = \mathcal{I}_\sigma(D) = \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(\sigma^2\right). \tag{4.31}$$

Note that there is no analogous equation for the case $\sigma \neq 0$ and $D$ approaching $E\{f_\sigma(z)\,|0,\sigma\}$ (i.e., small $y$), as for this case $\mathcal{I}_\sigma(D) - \mathcal{I}_0(D) - \sigma^2$ converges to a number that, although quite small (as shown by (4.40) in Section 4.3.6), is typically non-zero. This notwithstanding, (4.30) and (4.31) are valid also for $y = 0$.

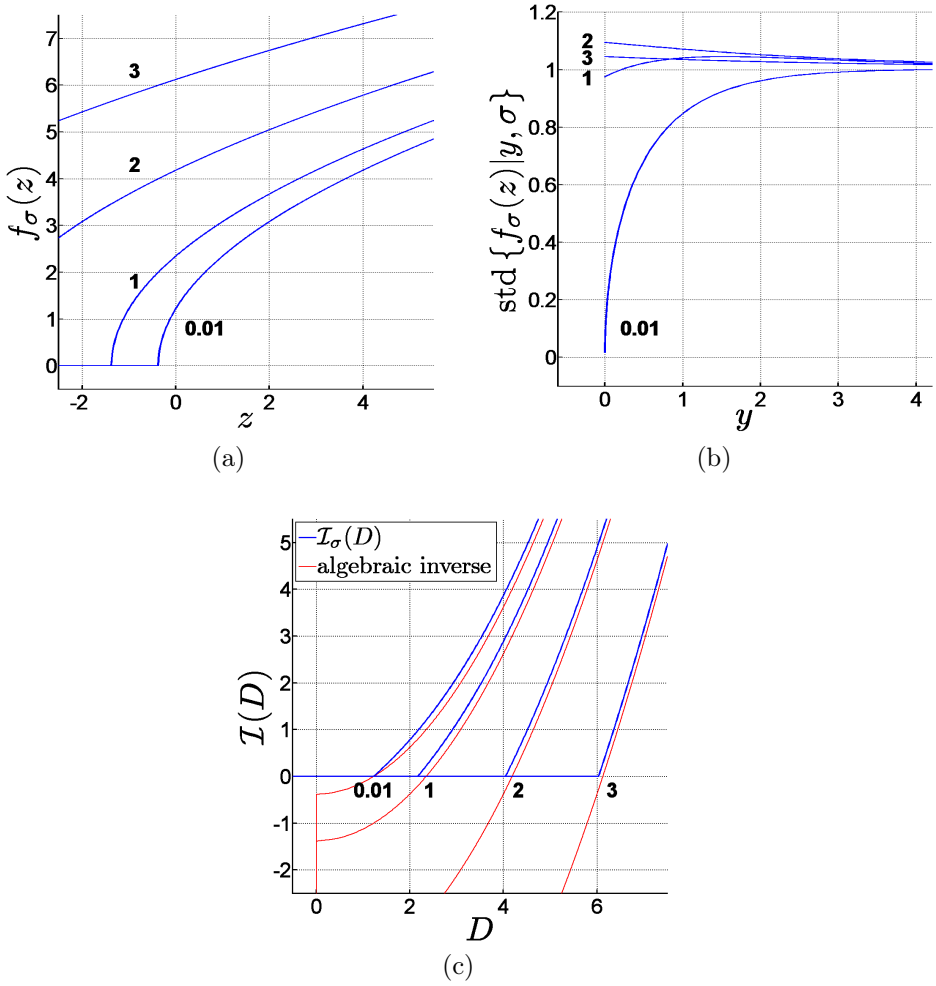



Figure 4.1: The generalized Anscombe transformation (4.7) for the parameter values $\sigma = 0.01, 1, 2, 3$. (a) The forward transformations $f_\sigma(z)$, (b) The standard deviations of the stabilized variables $f_\sigma(z)$, (c) The exact unbiased inverse transformations $\mathcal{I}_\sigma$, compared with the corresponding algebraic inverses of (4.7).

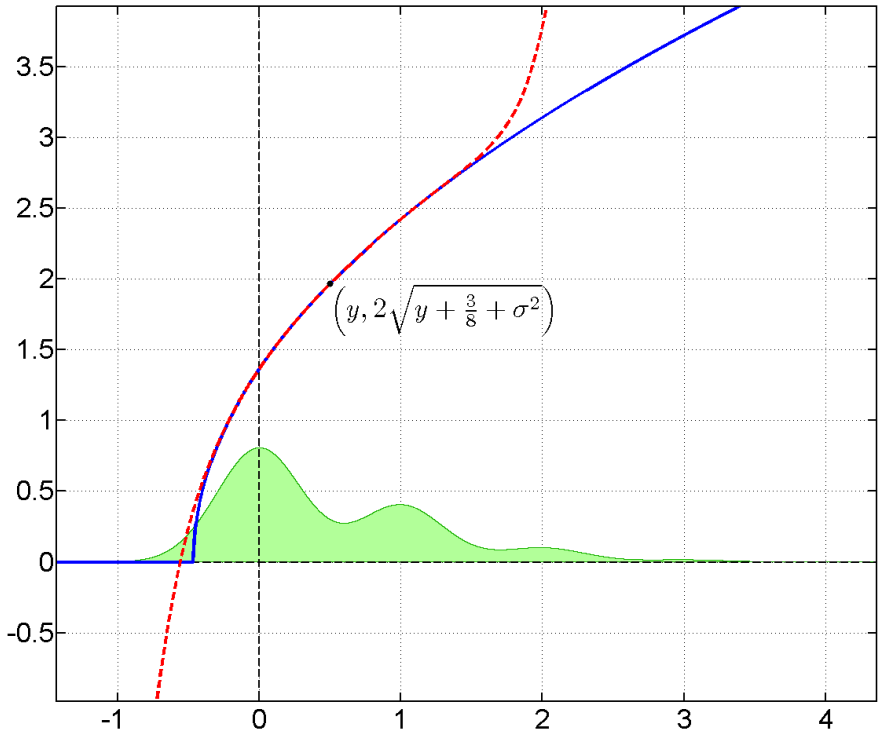



Figure 4.2: Probability density function of $z$ when $y = 0.5$ and $\sigma = 0.3$ (green area plot); generalized Anscombe transformation $f_\sigma$ (blue solid line) and its Taylor expansion of order 9 centered at $y$ (red dashed line).

**Summary**

Let us briefly summarize the main asymptotic results derived above. Considering the direct asymptotics with respect to $D = E\{f_\sigma(z)\,|y,\sigma\}$ and $\sigma$, we showed that for a given fixed $\sigma$ (and thus considering large $y$),

$$y = \mathcal{I}_\sigma(D) = \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(D^{-4}\right). \tag{4.32}$$

For a fixed $y$ (and thus considering large $\sigma$), we have $\sigma^2 \sim D^2$ and, consequently,

$$\begin{aligned} y = \mathcal{I}_\sigma(D) &= \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(D^{-2}\right) \\ &= \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(\sigma^{-2}\right). \end{aligned} \tag{4.33}$$

Finally, for fixed $y$ and small $\sigma$, we have

$$y = \mathcal{I}_\sigma(D) = \mathcal{I}_0(D) - \sigma^2 + \mathcal{O}\left(\sigma^2\right). \tag{4.34}$$

The orders $\mathcal{O}\left(\sigma^{-2}\right)$ and $\mathcal{O}\left(\sigma^2\right)$ of the error terms in (4.33) and (4.34), for various fixed values of $y$, are illustrated in Figure 4.3. Similarly, Figure 4.4 visualizes the order $\mathcal{O}\left(D^{-4}\right)$ of the error term in (4.32).

### 4.3.4 Global accuracy

Apart from the above asymptotic results, we studied the global accuracy of the approximation (4.15) in terms of the variance-normalized integral criterion (weighted $L^2$ squared), yielding

$$\begin{aligned} &\left\| \frac{\mathcal{I}_0(E\{f_\sigma\,(z)\,|y,\sigma\}) - \sigma^2 - y}{\operatorname{std}\{z|y,\sigma\}} \right\|_2^2 = \\ &\qquad = \int_0^{+\infty}\!\!\int_0^{+\infty} \frac{\left(\mathcal{I}_0(E\{f_\sigma(z)\,|y,\sigma\}) - \sigma^2 - y\right)^2}{y+\sigma^2}\, dy d\sigma = 0.0028 \end{aligned} \tag{4.35}$$

and in terms of the maximum absolute difference ($L^\infty$), yielding

$$\begin{aligned} &\left\|\mathcal{I}_0(E\{f_\sigma(z)\,|y,\sigma\}) - \sigma^2 - y\right\|_\infty = \\ &\qquad = \max_{\sigma\geq 0, y\geq 0} \left|\mathcal{I}_\sigma(E\{f_\sigma(z)\,|y,\sigma\}) - y\right| = 0.0470 \end{aligned} \tag{4.36}$$

with the maximum in (4.36) attained at $\sigma = 0.4$ and $y = 0$. In Figure 4.5, the blue surface (below) shows the error $|\mathcal{I}_0(E\{f_\sigma(z)\,|y,\sigma\}) - \sigma^2 - y|$ associated with (4.15), thus visualizing the overall accuracy and the maximum error of the approximation.

### 4.3.5 Practical implementation

Due to the discussed asymptotic behaviour, and given that $\mathcal{I}_0$ is already available (either in accurate numerical form or as closed-form analytical approximation), to compute $\mathcal{I}_\sigma$ it is sufficient to tabulate $E\{f_\sigma(z)\,|y,\sigma\}$ (4.10) only for a finite grid of values, and resort to interpolation (between the grid values) and to the asymptotic form $\mathcal{I}_0(D) - \sigma^2$ (outside of the grid). In particular, for our experiments, we

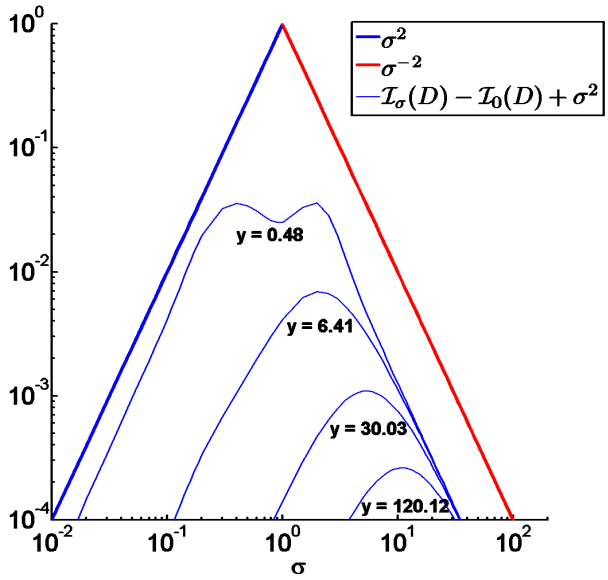



Figure 4.3: The error $\mathcal{I}_\sigma(D) - \mathcal{I}_0(D) + \sigma^2 = \mathcal{O}\left(\sigma^2\right)$ as a function of $\sigma$ for various fixed values of $y$, compared with $\sigma^{-2}$ and $\sigma^2$, confirming the asymptotics in (4.33) and (4.34).

considered 96 non-equispaced values $\sigma \in \{0.01, \ldots, 50\}$ and 1199 non-equispaced values of $y \in \{0, \ldots, 200\}$ and calculated $E\{f_\sigma(z)\,|y, \sigma\}$ on such 96×1199 grid. Our Matlab software implementing this inverse transformation is available online at http://www.cs.tut.fi/~foi/invansc.

### 4.3.6 Closed-form approximation

A closed-form approximation of $\mathcal{I}_\sigma$ can be obtained from the closed-form approximation $\widetilde{\mathcal{I}}_0$ (3.7) of $\mathcal{I}_0$ as

$$\widetilde{\mathcal{I}}_\sigma(D) = \widetilde{\mathcal{I}}_0(D) - \sigma^2 = \tag{4.37}$$
$$= \frac{1}{4}D^2 + \frac{1}{4}\sqrt{\frac{3}{2}}D^{-1} - \frac{11}{8}D^{-2} + \frac{5}{8}\sqrt{\frac{3}{2}}D^{-3} - \frac{1}{8} - \sigma^2.$$

Of course, the negative powers of $D$ become irrelevant when $D$ is large, i.e., when $\sigma$ or $y$ are large, yielding the asymptotic inverse

$$\mathcal{I}_{\text{asy}}(D) = \frac{1}{4}D^2 - \frac{1}{8} - \sigma^2. \tag{4.38}$$

As in Section 4.3.4, we studied the accuracy of the approximation (4.37) in terms of the variance-normalized integral criterion (weighted $L^2$ squared) and the maximum absolute difference $L^\infty$, with the corresponding results being

$$\left\| \frac{\widetilde{\mathcal{I}}_\sigma(E\{f_\sigma(z)\,|y,\sigma\}) - y}{\text{std}\,\{z|y,\sigma\}} \right\|_2^2 = 0.0069, \tag{4.39}$$

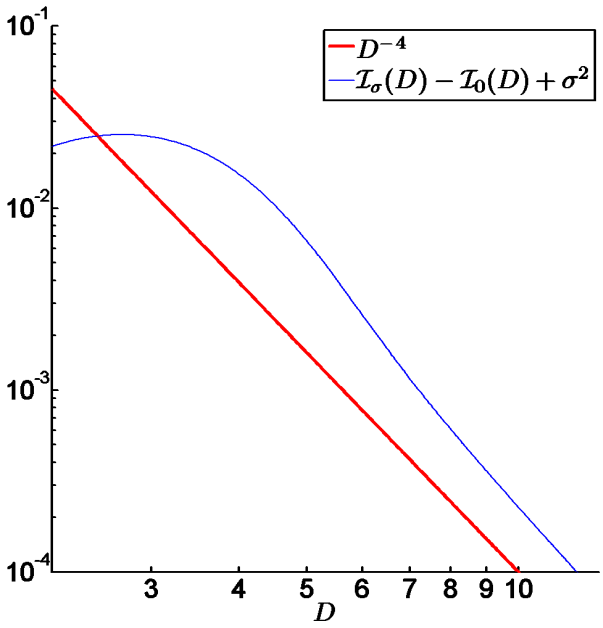



Figure 4.4: The error $\mathcal{I}_\sigma(D) - \mathcal{I}_0(D) + \sigma^2 = \mathcal{O}\left(D^{-4}\right)$ as a function of $D$ (for fixed $\sigma = 1$), compared with $D^{-4}$, confirming the asymptotics in (4.32).

and

$$\left\| \widetilde{\mathcal{I}}_\sigma(E\{f_\sigma(z)\,|y,\sigma\}) - y \right\|_\infty = 0.0468, \tag{4.40}$$

where the maximum of (4.40) is again attained at $\sigma = 0.4$ and $y = 0$. The error surface associated with this criterion is also illustrated in Figure 4.5 (purple surface on top).

In terms of the impact on denoising quality, the discrepancies between $\widetilde{\mathcal{I}}_\sigma$ and $\mathcal{I}_\sigma$ are very small, as affirmed by (4.39) and (4.40), and by the experimental results and observations in the next section. These results and considerations are consistent with those presented in Section 3.3.1.

## 4.4 Experiments

### 4.4.1 Denoising with known parameter values $\alpha$ and $\sigma$

We evaluate the denoising performance associated with the proposed exact unbiased inverse $\mathcal{I}_\sigma$ by considering the Cameraman (256×256), Fluorescent Cells (512×512), and Lena (512×512) test images. For each image, we scale the original image to eight different peak values (1, 2, 5, 10, 20, 30, 60, 120), and corrupt them with Poisson-Gaussian noise ($\alpha = 1$, $\sigma = \text{peak}/10$) according to (4.5), as was done in [98]. Let us remark that since $\alpha = 1$, we have also $\grave{\sigma} = \sigma$, as can be seen from (4.4).

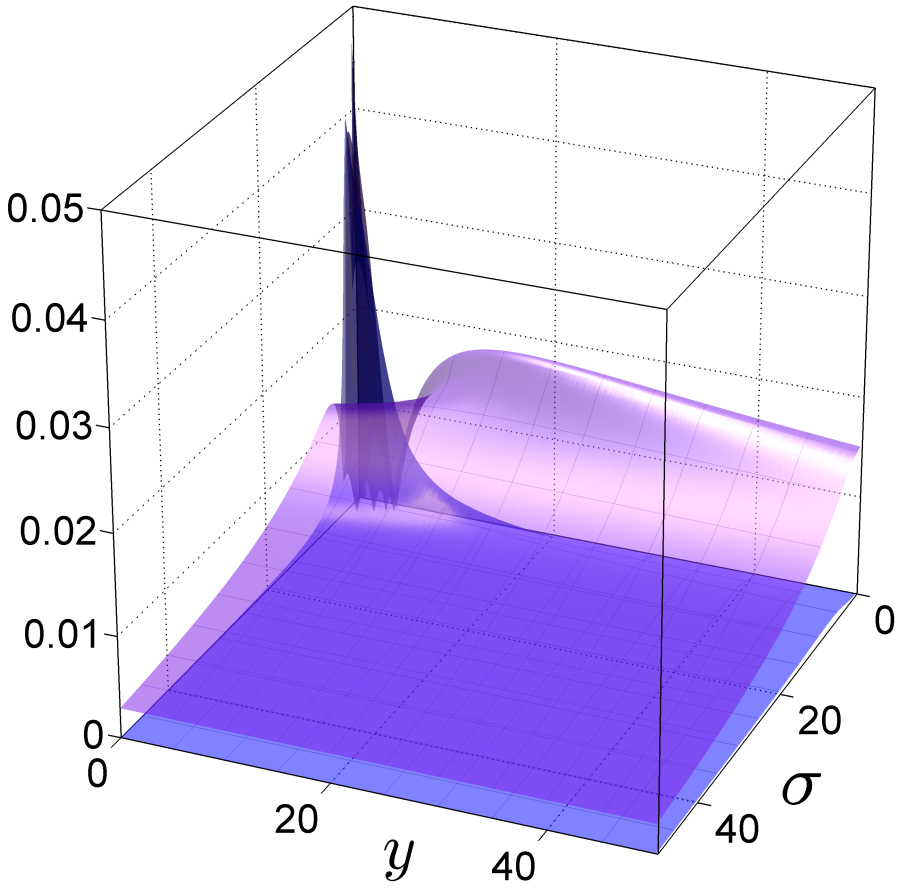



Figure 4.5: The error surface $|\mathcal{I}_0(E\{f_\sigma(z)\,|y,\sigma\}) - \sigma^2 - y|$ associated with (4.15) (blue surface below), and the corresponding surface $|\widetilde{\mathcal{I}}_\sigma(E\{f_\sigma(z)\,|y,\sigma\}) - y|$ associated with the closed-form approximation (4.37) (purple surface on top).

We denoise each image with the three-step variance stabilization approach explained in Section 4.1, using either BM3D or BLS-GSM as the Gaussian denoising algorithm, and inverting the denoised data with each of the following transformations: the exact unbiased inverse $\mathcal{I}_\sigma$, its closed-form approximation $\widetilde{\mathcal{I}}_\sigma$, the asymptotically unbiased inverse $\mathcal{I}_{\text{asy}}(D) = \frac{1}{4}D^2 - \frac{1}{8} - \sigma^2$, or the algebraic inverse $\mathcal{I}_{\text{alg}}(D) = \frac{1}{4}D^2 - \frac{3}{8} - \sigma^2$. When denoising the stabilized data $f_\sigma(z)$, the algorithms assume that $\text{std}\{f_\sigma(z) \mid y,\sigma\}$ is exactly 1. For comparison with the direct approaches, we also denoise each image with the state-of-the-art UWT/BDCT PURE-LET method proposed in [98]. The results are presented in Tables 4.1–4.2, where each PSNR value (Table 4.1), and the respective SSIM [137] value (Table 4.2), is an average of ten individual denoising results (performed on ten random realizations of the Poisson-Gaussian noise).

We see that the proposed method is competitive with the UWT/BDCT PURE-LET algorithm, outperforming it in many cases, in particular when variance sta-

bilization is combined with the BM3D algorithm, which represents the state of the art in additive white Gaussian noise removal. Moreover, there are no major declines in performance for the low-intensity cases, which demonstrates the fact that the poor performance shown in earlier works (e.g., [98]) is not simply due to inadequate variance stabilization associated with the GAT, but mostly a consequence of using an improper inverse transformation. In particular, we see that for the low-intensity cases it is clearly not reasonable to use either the asymptotically unbiased inverse or the algebraic inverse; instead, the proposed exact unbiased inverse $\mathcal{I}_\sigma$ can be used everywhere. Further, its closed-form approximation $\widetilde{\mathcal{I}}_\sigma$ is practically on par with it, introducing minor discrepancies only at the lowest intensities.

Figures 4.6–4.11 present visual comparisons for Fluorescent Cells (peak = 20, $\sigma = 2$), Lena (peak = 100, $\sigma = 2$), and Cameraman (peak = 80, $\sigma = 3$), corroborating the observed good performance of the proposed denoising method. Figure 4.12 shows the denoising results for the low-intensity case of Cameraman with peak = 1 and $\sigma = 0.1$, including a comparison of the different inverses. In addition, Figure 4.13 presents a similar comparison for the Boat ($512 \times 512$) test image with peak = 1.5 and $\sigma = 0.1$. These two figures clearly visualize the previously noted importance of applying a proper inverse transformation to the denoised data.

### 4.4.2 Denoising with estimated parameter values $\alpha_{\mathbf{est}}$ and $\grave{\sigma}_{\mathbf{est}}$

Here we examine the robustness and practical applicability of the proposed method by repeating a subset of the experiments corresponding to Tables 4.1–4.2, but using estimated parameter values $\alpha_{\text{est}}$ and $\grave{\sigma}_{\text{est}}$ instead of the true values $\alpha$ and $\grave{\sigma}$. Note that since we now generally have $\alpha_{\text{est}} \neq 1$, we also have $\grave{\sigma}_{\text{est}} \neq \sigma_{\text{est}}$. Thus, in this part of the experiments, we are making a distinction between $\grave{\sigma}$ and $\sigma$ for clarity, even though $\grave{\sigma} = \sigma$ still holds, as in all the previous experiments. The parameters $\alpha_{\text{est}}$ and $\grave{\sigma}_{\text{est}}$ are estimated from a single noisy image by fitting a global parametric model into locally estimated expectation / standard deviation pairs, as proposed in [64]. Then, the transformations (4.4) and the GAT (4.7) $f_{\sigma_{\text{est}}}$ are applied with $\sigma_{\text{est}} = \grave{\sigma}_{\text{est}}/\alpha_{\text{est}}$, the stabilized data is denoised with BM3D, and the final estimate is obtained by inverting the denoised data with either the exact unbiased inverse $\mathcal{I}_{\sigma_{\text{est}}}$, the asymptotically unbiased inverse $\mathcal{I}_{\text{asy}}(D) = \frac{1}{4}D^2 - \frac{1}{8} - \sigma^2_{\text{est}}$, or the algebraic inverse $\mathcal{I}_{\text{alg}}(D) = \frac{1}{4}D^2 - \frac{3}{8} - \sigma^2_{\text{est}}$.

The results of these experiments are presented in Table 4.3, showing the robustness of the proposed denoising framework, as in the vast majority of the cases the denoising results associated with $\mathcal{I}_{\sigma_{\text{est}}}$ are practically equal to the ones obtained with the exact parameter values $\alpha$ and $\sigma$ (Tables 4.1–4.2); the only notable declines in performance are the approximately 0.3 dB drops for Fluorescent Cells with peaks 1 and 2. However, these drops at low peaks are insignificant in comparison with the major decline (several dBs) in performance caused by using either the asymptotically unbiased inverse $\mathcal{I}_{\text{asy}}$ or the algebraic inverse $\mathcal{I}_{\text{alg}}$ instead of $\mathcal{I}_{\sigma_{\text{est}}}$.

We remark that each result in Table 4.3 is obtained by denoising a single image instead of averaging ten results, since in Table 4.1 we observed that the variations

between individual results are typically minor (in the order of ± 0.1 dB around the average PSNR). Note that even though the true value of $\alpha$ equals 1 in all the experiments, this information is not used in any way in computing the estimates $\alpha_{\text{est}}$ and $\grave{\sigma}_{\text{est}}$. Moreover, using only $\alpha = 1$ is not a fundamental restriction, as the more general case (2.7) can be addressed through simple scaling.

### 4.4.3 Varying the ratio between the Poisson and Gaussian noise components

Finally we analyze how much the denoising performance changes, when the noise distribution gradually changes from pure Poisson to AWGN. In practise, we denoise Cameraman and Lena with either BM3D or BLS-GSM combined with the exact unbiased inverse $\mathcal{I}_\sigma$, and with UWT/BDCT PURE-LET, when the PSNR of the noisy image is kept constant at either 15 dB, 10 dB or 5 dB (with negligible variations depending on the specific realization, of the order ± 0.01 dB), but the ratio $\sigma/\sqrt{\text{peak}}$ is varied from 0 to 10. When this ratio is low, the Poisson noise component is the dominant one, whereas a high ratio means the noise distribution is nearly Gaussian. Specifically, the GAT corresponding to the case $\sigma/\sqrt{\text{peak}} = 10$ is practically affine over the interval [0, peak], and thus, this case is essentially like denoising pure AWGN without any variance stabilization. Conversely, when $\sigma/\sqrt{\text{peak}} = 0$, the Gaussian component is absent, and hence we reduce to the case of denoising pure Poisson data.

Figures 4.14–4.16 show the results of these experiments. First, we see the general trend that the denoising performance increases as the distribution becomes less Gaussian. We explain this perhaps curious behaviour by the fact that a Gaussian distribution can be considered the worst-case scenario, in the sense that it leads to the largest Cramér-Rao lower bound under a rather general class of distributions [129]. Second, it is apparent from Figure 4.15(b) that in the 10 dB case, there is a minor drop in performance for BM3D at about $\sigma/\sqrt{\text{peak}} = 0.5$, but it is not observed for UWT/BDCT PURE-LET. This deficiency could be caused by inaccurate variance stabilization, since the corresponding drop becomes more pronounced in the extreme 5 dB case (Figure 4.16). In particular, as $\sigma/\sqrt{\text{peak}}$ decreases, the stabilization changes from nearly perfect to a slight overshooting (up to about 10 % extra in terms of the standard deviation), and finally undershooting (see Figure 4.1(b)). For low input PSNR values, such as 5 dB or 10 dB, the intensity range of the scaled image is typically concentrated in the region with inaccurate stabilization. However, for higher PSNR values this concerns only the darkest part of the image. This notion is supported by the comparisons in Figure 4.14, showing the corresponding results for the 15 dB case; there is no drop in performance for BM3D due to variance stabilization.

Overall, BM3D and the exact unbiased inverse $\mathcal{I}_\sigma$ outperforms the state-of-the-art UWT/BDCT PURE-LET in all the 10 dB and 15 dB cases, even when inaccurate variance stabilization causes a minor drop in performance for low input PSNR values around $\sigma/\sqrt{\text{peak}} = 0.5$. In the 5 dB cases, BM3D still outperforms the other algorithms for Cameraman (Figure 4.16(a)), but for Lena (Figure 4.16(a)), the more pronounced drop for BM3D around $\sigma/\sqrt{\text{peak}} = 0.5$ causes UWT/BDCT PURE-LET to outperform it around that region. In addition, in

the case of 5 dB Lena, BLS-GSM excels at higher ratios of $\sigma/\sqrt{\text{peak}}$, outperforming BM3D by about 0.5 dB and UWT/BDCT PURE-LET by close to 1 dB. This anomaly at a very low input PSNR can be explained by the fact that unlike BLS-GSM, BM3D relies on a number of fixed internal parameters, which are not optimized for such extreme cases.

On the other hand, the performance of BLS-GSM is much degraded by the mismatch between the actual standard deviation of the stabilized data and its assumed value (which is always 1). This is not a peculiarity of the Poisson-Gaussian case, as drops in performance comparable to those visible in Figures 4.15–4.16 can be experienced with BLS-GSM also in the case of purely Gaussian noise when the standard deviation of the noise is over- or underestimated.

## 4.5 Discussion

We have generalized the results of Chapter 3 in order to encompass the case of Poisson-Gaussian noise. Specifically, we proposed an exact unbiased inverse of the generalized Anscombe transformation for Poisson-Gaussian noise and supplemented it with rigorous mathematical considerations. We also proposed a closed-form approximation of this inverse, based on our closed-form approximation (3.7) of the exact unbiased inverse of the Anscombe transformation.

We showed that the denoising performance associated with the proposed exact unbiased inverse, in conjunction with a state-of-the-art Gaussian noise removal algorithm, is competitive with that of a state-of-the-art algorithm designed specifically for the removal of Poisson-Gaussian noise. Further, we observed that for low peak intensities, the performance gain obtained by using the exact unbiased inverse instead of the algebraic or the asymptotically unbiased inverse is especially significant. In other words, we showed that the poor denoising performance shown in earlier works is not simply due to the inability of the GAT to stabilize the noise variance adequately, but mostly due to applying an unsuitable inverse transformation.

We also wish to point out that the GAT is not the only possible choice for stabilizing the noise variance. For instance, one may instead compute a family of optimized transformations [66] (mitigating the undershoot and overshoot observed in Figure 4.1(b)), to which our denoising framework can be accommodated by recomputing the grid of corresponding expected values (4.10). However, this is not within our scope, as we focus on the major significance of the exact unbiased inverse transformation, not on optimized variance stabilization.

Table 4.1: A comparison of the denoising performance (PSNR, dB) of several denoising algorithms and inverse transformations, using various peak intensities and various standard deviations $\sigma$ of the Gaussian noise component. The results are averages of ten independent realizations, and the variations between individual results are typically in the order of $\pm$ 0.1 dB around the average PSNR.

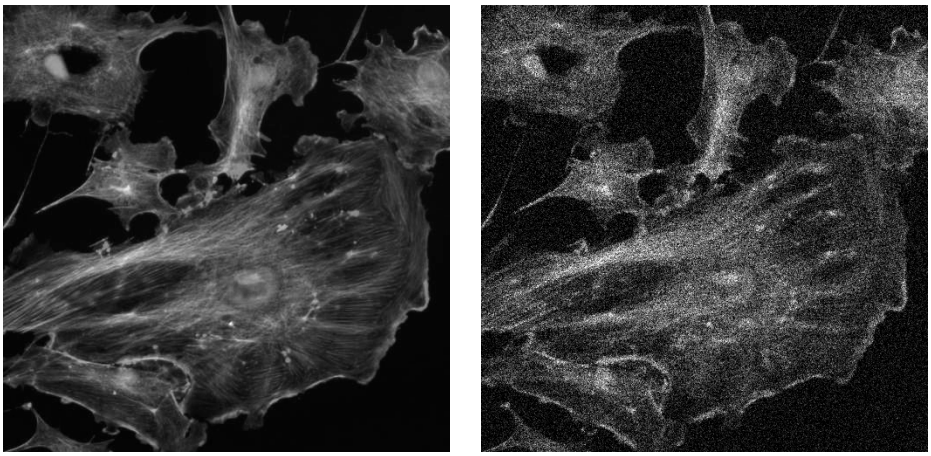
| Image | Peak | $\sigma$ | Noisy | GAT + BM3D | | | | GAT + BLS-GSM | | | | UWT/BDCT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $\mathcal{I}_\sigma$ | $\widetilde{\mathcal{I}}_\sigma$ | $\mathcal{I}_{\text{asy}}$ | $\mathcal{I}_{\text{alg}}$ | $\mathcal{I}_\sigma$ | $\widetilde{\mathcal{I}}_\sigma$ | $\mathcal{I}_{\text{asy}}$ | $\mathcal{I}_{\text{alg}}$ | PURE-LET [98] |
| Cameraman (256×256) | 1 | 0.1 | 3.20 | 20.23 | 20.18 | 15.55 | 15.72 | 18.46 | 18.45 | 14.56 | 15.40 | **20.35** |
| | 2 | 0.2 | 6.12 | **21.93** | 21.85 | 20.70 | 18.24 | 20.28 | 20.23 | 19.41 | 17.36 | 21.60 |
| | 5 | 0.5 | 9.83 | **24.09** | 24.07 | 24.00 | 22.36 | 23.01 | 22.98 | 22.93 | 21.29 | 23.33 |
| | 10 | 1 | 12.45 | **25.52** | **25.52** | **25.52** | 24.80 | 24.36 | 24.36 | 24.35 | 23.62 | 24.68 |
| | 20 | 2 | 14.76 | **26.77** | **26.77** | 26.75 | 26.48 | 25.58 | 25.58 | 25.53 | 25.26 | 25.92 |
| | 30 | 3 | 15.91 | **27.30** | **27.30** | 27.29 | 27.13 | 26.20 | 26.20 | 26.16 | 26.01 | 26.51 |
| | 60 | 6 | 17.49 | **28.07** | **28.07** | 28.06 | 28.01 | 27.02 | 27.02 | 26.98 | 26.93 | 27.35 |
| | 120 | 12 | 18.57 | **28.57** | **28.57** | 28.55 | 28.54 | 27.57 | 27.57 | 27.52 | 27.51 | 27.89 |
| Fluorescent Cells (512×512) | 1 | 0.1 | 7.22 | 24.54 | 24.45 | 13.86 | 20.83 | 22.35 | 22.33 | 13.47 | 20.01 | **25.13** |
| | 2 | 0.2 | 9.99 | 25.87 | 25.60 | 20.99 | 21.96 | 24.20 | 24.07 | 20.25 | 21.23 | **26.25** |
| | 5 | 0.5 | 13.37 | 27.45 | 27.38 | 26.93 | 24.80 | 26.99 | 26.88 | 26.52 | 24.25 | **27.60** |
| | 10 | 1 | 15.53 | 28.63 | **28.64** | 28.61 | 27.20 | 28.05 | 28.04 | 28.03 | 26.50 | 28.59 |
| | 20 | 2 | 17.21 | 29.65 | **29.66** | 29.64 | 29.09 | 29.05 | 29.06 | 28.89 | 28.28 | 29.47 |
| | 30 | 3 | 17.97 | **30.16** | **30.16** | 30.15 | 29.86 | 29.74 | 29.74 | 29.65 | 29.33 | 29.84 |
| | 60 | 6 | 18.86 | **30.77** | **30.77** | **30.77** | 30.68 | 30.52 | 30.52 | 30.48 | 30.38 | 30.42 |
| | 120 | 12 | 19.39 | **31.14** | **31.14** | **31.14** | 31.11 | 30.91 | 30.91 | 30.87 | 30.85 | 30.70 |
| Lena (512×512) | 1 | 0.1 | 2.87 | 22.59 | 22.50 | 16.89 | 16.38 | 21.55 | 21.54 | 16.04 | 16.48 | **22.83** |
| | 2 | 0.2 | 5.82 | **24.34** | 24.21 | 23.31 | 18.91 | 23.56 | 23.46 | 22.52 | 18.58 | 24.16 |
| | 5 | 0.5 | 9.54 | 26.17 | 26.16 | **26.18** | 23.52 | 25.98 | 25.94 | 25.98 | 22.91 | 25.74 |
| | 10 | 1 | 12.19 | 27.72 | 27.71 | **27.73** | 26.50 | 26.85 | 26.84 | 26.84 | 25.57 | 27.27 |
| | 20 | 2 | 14.53 | **29.01** | 29.00 | **29.01** | 28.54 | 28.47 | 28.47 | 28.47 | 27.96 | 28.46 |
| | 30 | 3 | 15.72 | **29.69** | **29.69** | **29.69** | 29.44 | 29.26 | 29.27 | 29.26 | 28.99 | 29.12 |
| | 60 | 6 | 17.35 | **30.51** | **30.51** | **30.51** | 30.43 | 30.12 | 30.12 | 30.12 | 30.03 | 29.91 |
| | 120 | 12 | 18.48 | **31.05** | **31.05** | **31.05** | 31.03 | 30.69 | 30.69 | 30.69 | 30.66 | 30.51 |

Table 4.2: A comparison of the denoising performance (mean SSIM [137]) of several denoising algorithms and inverse transformations, using various peak intensities and various standard deviations $\sigma$ of the Gaussian noise component. The results are averages of ten independent realizations (exactly the same realizations as in Table 4.1).

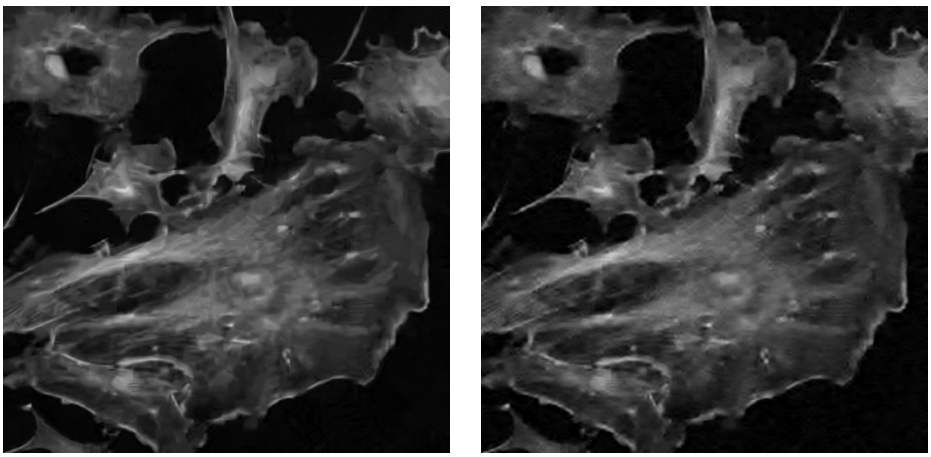
| Image | Peak | $\sigma$ | Noisy | GAT + BM3D | | | | GAT + BLS-GSM | | | | UWT/BDCT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $\mathcal{I}_\sigma$ | $\widetilde{\mathcal{I}}_\sigma$ | $\mathcal{I}_{\text{asy}}$ | $\mathcal{I}_{\text{alg}}$ | $\mathcal{I}_\sigma$ | $\widetilde{\mathcal{I}}_\sigma$ | $\mathcal{I}_{\text{asy}}$ | $\mathcal{I}_{\text{alg}}$ | PURE-LET [98] |
| Cameraman (256×256) | 1 | 0.1 | 0.0442 | **0.5991** | 0.5973 | 0.5071 | 0.5652 | 0.5204 | 0.5177 | 0.4579 | 0.5007 | 0.5232 |
| | 2 | 0.2 | 0.0736 | **0.6512** | 0.6427 | 0.5979 | 0.5992 | 0.5873 | 0.5784 | 0.5470 | 0.5437 | 0.5573 |
| | 5 | 0.5 | 0.1238 | **0.7068** | 0.7044 | 0.6998 | 0.6663 | 0.6569 | 0.6528 | 0.6551 | 0.6131 | 0.6227 |
| | 10 | 1 | 0.1661 | 0.7509 | **0.7514** | 0.7512 | 0.7343 | 0.6689 | 0.6685 | 0.6697 | 0.6419 | 0.6849 |
| | 20 | 2 | 0.2085 | 0.7882 | **0.7883** | 0.7879 | 0.7840 | 0.7135 | 0.7131 | 0.7098 | 0.7012 | 0.7339 |
| | 30 | 3 | 0.2314 | **0.8024** | **0.8024** | 0.8022 | 0.8005 | 0.7414 | 0.7413 | 0.7392 | 0.7348 | 0.7554 |
| | 60 | 6 | 0.2657 | **0.8225** | **0.8225** | 0.8224 | 0.8218 | 0.7696 | 0.7696 | 0.7682 | 0.7666 | 0.7832 |
| | 120 | 12 | 0.2913 | 0.8345 | **0.8346** | 0.8344 | 0.8342 | 0.7865 | 0.7866 | 0.7851 | 0.7845 | 0.7992 |
| Fluorescent Cells (512×512) | 1 | 0.1 | 0.0234 | 0.6090 | 0.5938 | 0.3465 | 0.5189 | 0.5098 | 0.5002 | 0.2939 | 0.4715 | **0.6230** |
| | 2 | 0.2 | 0.0421 | 0.6558 | 0.6031 | 0.4828 | 0.4518 | 0.5759 | 0.5302 | 0.4149 | 0.4169 | **0.6567** |
| | 5 | 0.5 | 0.0806 | **0.7120** | 0.6943 | 0.6634 | 0.4930 | 0.6821 | 0.6558 | 0.6379 | 0.4421 | 0.6982 |
| | 10 | 1 | 0.1184 | 0.7491 | **0.7515** | 0.7464 | 0.6134 | 0.7018 | 0.6993 | 0.7018 | 0.5148 | 0.7288 |
| | 20 | 2 | 0.1569 | 0.7795 | **0.7811** | 0.7788 | 0.7365 | 0.7301 | 0.7292 | 0.7093 | 0.6347 | 0.7566 |
| | 30 | 3 | 0.178 | **0.7980** | **0.7980** | 0.7968 | 0.7750 | 0.7608 | 0.7605 | 0.7488 | 0.7098 | 0.7661 |
| | 60 | 6 | 0.2073 | 0.8157 | **0.8158** | 0.8152 | 0.8082 | 0.7913 | 0.7915 | 0.7861 | 0.7728 | 0.7867 |
| | 120 | 12 | 0.2267 | **0.8241** | **0.8241** | 0.8236 | 0.8207 | 0.8021 | 0.8022 | 0.7973 | 0.7919 | 0.7913 |
| Lena (512×512) | 1 | 0.1 | 0.016 | 0.6288 | **0.6290** | 0.6108 | 0.6068 | 0.6134 | 0.6135 | 0.5899 | 0.5929 | 0.5767 |
| | 2 | 0.2 | 0.0296 | **0.6676** | 0.6670 | 0.6657 | 0.6513 | 0.6606 | 0.6604 | 0.6561 | 0.6443 | 0.6031 |
| | 5 | 0.5 | 0.0606 | **0.7038** | 0.7031 | **0.7038** | 0.6985 | 0.6812 | 0.6804 | 0.6813 | 0.6754 | 0.6406 |
| | 10 | 1 | 0.0946 | **0.7554** | 0.7553 | 0.7553 | 0.7538 | 0.6860 | 0.6855 | 0.6859 | 0.6841 | 0.7227 |
| | 20 | 2 | 0.1343 | **0.7934** | **0.7934** | **0.7934** | 0.7930 | 0.7622 | 0.7621 | 0.7621 | 0.7616 | 0.7662 |
| | 30 | 3 | 0.1591 | **0.8107** | **0.8107** | **0.8107** | 0.8105 | 0.7915 | 0.7915 | 0.7915 | 0.7913 | 0.7882 |
| | 60 | 6 | 0.1965 | **0.8297** | **0.8297** | **0.8297** | **0.8297** | 0.8159 | 0.8159 | 0.8159 | 0.8158 | 0.8088 |
| | 120 | 12 | 0.2266 | **0.8410** | **0.8410** | **0.8410** | **0.8410** | 0.8298 | 0.8298 | 0.8298 | 0.8298 | 0.8264 |

Table 4.3: The denoising performance (PSNR (dB), and mean SSIM [137]) of BM3D, combined with either the exact unbiased inverse $\mathcal{I}_\sigma$, the asymptotically unbiased inverse $\mathcal{I}_{\text{asy}}$ or the algebraic inverse $\mathcal{I}_{\text{alg}}$, using estimated parameter values $\alpha_{\text{est}}$ and $\grave{\sigma}_{\text{est}}$ instead of the true values $\alpha = 1$ and $\grave{\sigma}$. Each result is obtained from a single noisy realization, but as the variations between individual results are typically minor, the results are comparable with the average results presented in Tables 4.1–4.2. Note that since $\alpha_{\text{est}} \neq 1$, we have $\grave{\sigma}_{\text{est}} \neq \sigma_{\text{est}}$, even though $\grave{\sigma} = \sigma$.

| Image | Peak | $\grave{\sigma} = \sigma$ | $\alpha_{\text{est}}$ | $\grave{\sigma}_{\text{est}}$ | $\sigma_{\text{est}} = \frac{\grave{\sigma}_{\text{est}}}{\alpha_{\text{est}}}$ | PSNR | | | SSIM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $\mathcal{I}_{\sigma_{\text{est}}}$ | $\mathcal{I}_{\text{asy}}$ | $\mathcal{I}_{\text{alg}}$ | $\mathcal{I}_{\sigma_{\text{est}}}$ | $\mathcal{I}_{\text{asy}}$ | $\mathcal{I}_{\text{alg}}$ |
| Cameraman (256×256) | 1 | 0.1 | 0.957 | 0.141 | 0.147 | **20.34** | 16.04 | 15.71 | **0.5923** | 0.5039 | 0.5614 |
| | 2 | 0.2 | 0.948 | 0.118 | 0.125 | **21.77** | 21.00 | 18.23 | **0.5994** | 0.5691 | 0.5544 |
| | 5 | 0.5 | 1.015 | 0.459 | 0.453 | **24.19** | 24.10 | 22.38 | **0.7115** | 0.7039 | 0.6705 |
| | 10 | 1 | 1.095 | 0.896 | 0.819 | **25.51** | **25.51** | 24.61 | 0.7600 | **0.7604** | 0.7393 |
| | 20 | 2 | 1.008 | 2.075 | 2.059 | **26.81** | 26.79 | 26.54 | **0.7888** | 0.7887 | 0.7857 |
| | 30 | 3 | 1.053 | 3.038 | 2.885 | **27.24** | 27.23 | 27.09 | **0.8048** | 0.8045 | 0.8028 |
| | 60 | 6 | 0.973 | 6.224 | 6.398 | **28.08** | 28.07 | 28.01 | **0.8203** | 0.8201 | 0.8195 |
| | 120 | 12 | 0.907 | 12.505 | 13.781 | **28.64** | 28.62 | 28.60 | **0.8379** | 0.8378 | 0.8371 |
| Fluorescent Cells (512×512) | 1 | 0.1 | 0.949 | 0.183 | 0.193 | **24.19** | 14.40 | 20.81 | **0.5655** | 0.3541 | 0.5191 |
| | 2 | 0.2 | 0.920 | 0.153 | 0.166 | **25.55** | 22.07 | 21.58 | **0.6184** | 0.5001 | 0.4184 |
| | 5 | 0.5 | 0.984 | 0.483 | 0.491 | **27.54** | 27.06 | 24.78 | **0.7140** | 0.6678 | 0.4947 |
| | 10 | 1 | 0.993 | 0.995 | 1.002 | **28.62** | 28.60 | 27.18 | **0.7503** | 0.7476 | 0.6211 |
| | 20 | 2 | 0.960 | 2.035 | 2.121 | **29.71** | 29.70 | 29.19 | **0.7827** | 0.7813 | 0.7378 |
| | 30 | 3 | 0.928 | 3.067 | 3.305 | **30.26** | **30.26** | 30.01 | **0.8040** | 0.8036 | 0.7879 |
| | 60 | 6 | 0.994 | 5.993 | 6.027 | **30.79** | 30.78 | 30.70 | **0.8182** | 0.8179 | 0.8122 |
| | 120 | 12 | 1.143 | 12.014 | 10.516 | **31.16** | **31.16** | 31.13 | **0.8244** | 0.8239 | 0.8205 |
| Lena (512×512) | 1 | 0.1 | 1.003 | 0.218 | 0.218 | **22.60** | 16.62 | 16.62 | **0.6303** | 0.6107 | 0.6109 |
| | 2 | 0.2 | 1.019 | 0.200 | 0.196 | **24.35** | 23.17 | 18.93 | **0.6697** | 0.6672 | 0.6535 |
| | 5 | 0.5 | 1.074 | 0.285 | 0.265 | 26.09 | **26.12** | 23.31 | 0.7054 | **0.7059** | 0.6995 |
| | 10 | 1 | 0.992 | 1.086 | 1.094 | **27.78** | 27.77 | 26.61 | **0.7571** | 0.7570 | 0.7556 |
| | 20 | 2 | 1.091 | 1.884 | 1.726 | **29.06** | **29.06** | 28.56 | **0.7976** | **0.7976** | 0.7971 |
| | 30 | 3 | 0.994 | 3.100 | 3.118 | **29.81** | **29.81** | 29.61 | **0.8136** | **0.8136** | 0.8134 |
| | 60 | 6 | 1.068 | 5.861 | 5.489 | **30.55** | **30.55** | 30.47 | **0.8308** | **0.8308** | 0.8307 |
| | 120 | 12 | 0.792 | 12.490 | 15.778 | **31.02** | **31.02** | 31.01 | **0.8393** | **0.8393** | **0.8393** |

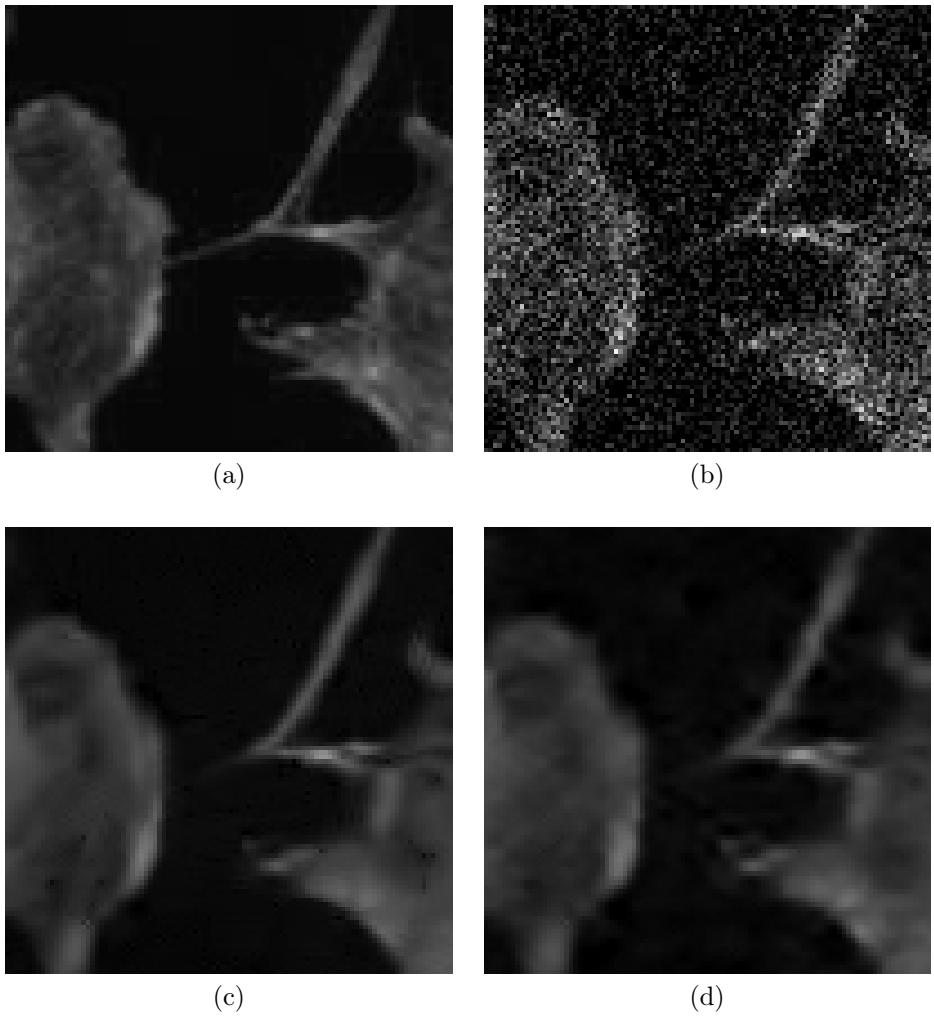


(a) (b)



(c) (d)

Figure 4.6: The denoising of Fluorescent Cells (512×512). (a) Original image, (b) Noisy image (peak = 20, $\sigma = 2$, PSNR = 17.21 dB), (c) Denoised with BM3D and the exact unbiased inverse $\mathcal{I}_\sigma$ (PSNR = 29.65 dB), (d) Denoised with UWT/BDCT PURE-LET (PSNR = 29.47 dB).



(a) (b)

(c) (d)

Figure 4.7: A 100×100 section of each of the images in Figure 4.6. (a) Original image, (b) Noisy image (peak = 20, $\sigma = 2$), (c) Denoised with BM3D and the exact unbiased inverse $\mathcal{I}_\sigma$, (d) Denoised with UWT/BDCT PURE-LET.

(a) (b)

(c) (d)

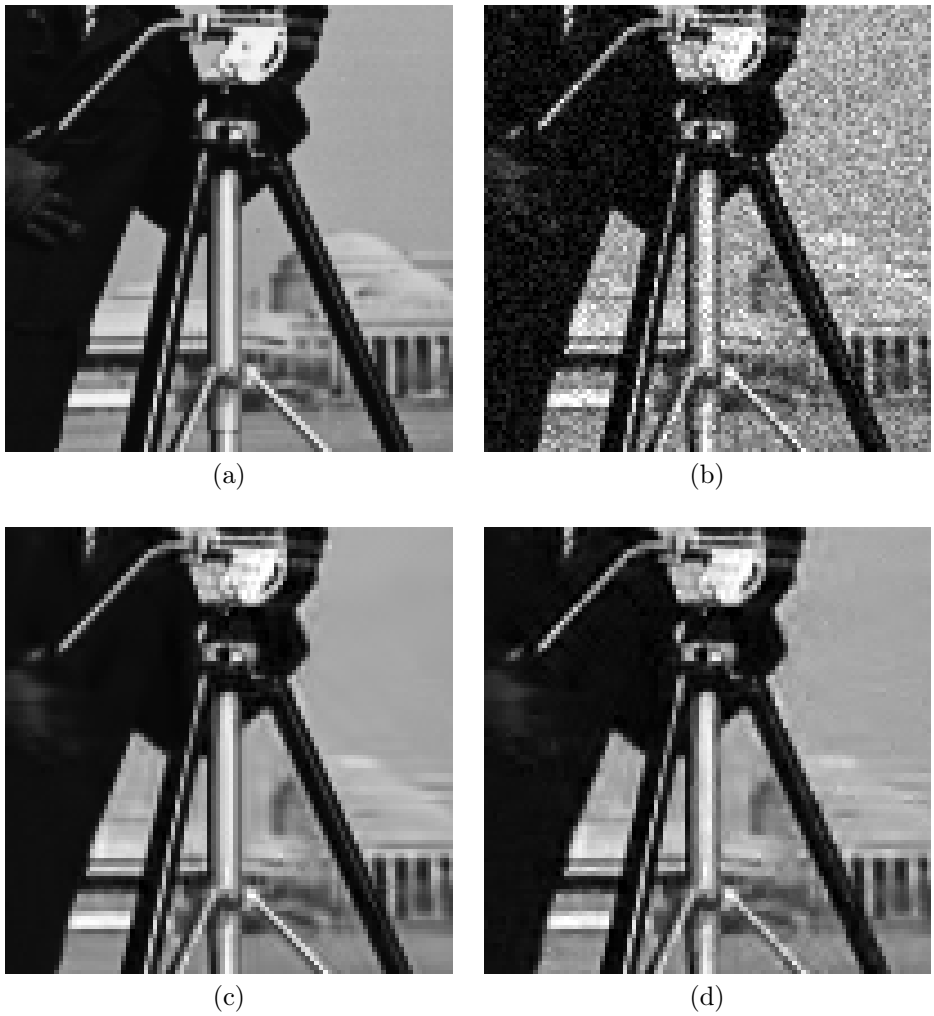
Figure 4.8: The denoising of Lena (512×512). (a) Original image, (b) Noisy image (peak = 100, $\sigma = 2$, PSNR = 22.64 dB), (c) Denoised with BM3D and the exact unbiased inverse $\mathcal{I}_\sigma$ (PSNR = 33.41 dB), (d) Denoised with UWT/BDCT PURE-LET (PSNR = 32.92 dB).

Figure 4.9: A 100×100 section of each of the images in Figure 4.8. (a) Original image, (b) Noisy image (peak = 100, $\sigma = 2$), (c) Denoised with BM3D and the exact unbiased inverse $\mathcal{I}_\sigma$, (d) Denoised with UWT/BDCT PURE-LET.

(a) (b)

(c) (d)

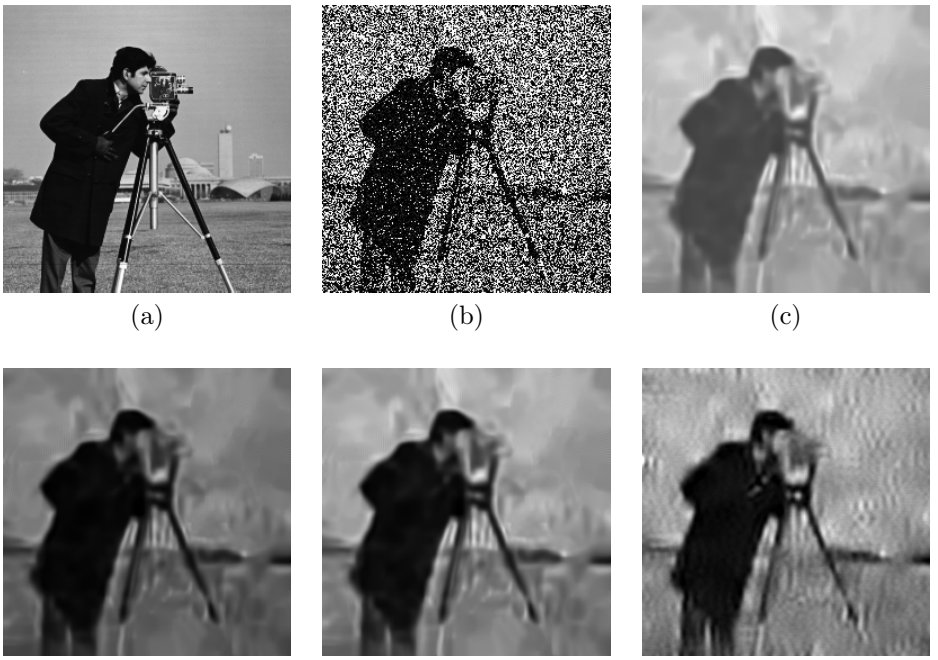
Figure 4.10: The denoising of Cameraman (256×256). (a) Original image, (b) Noisy image (peak = 80, $\sigma = 3$, PSNR = 21.38 dB), (c) Denoised with BM3D and the exact unbiased inverse $\mathcal{I}_\sigma$ (PSNR = 30.15 dB), (d) Denoised with UWT/BDCT PURE-LET (PSNR = 29.68 dB).



(a) (b)

(c) (d)

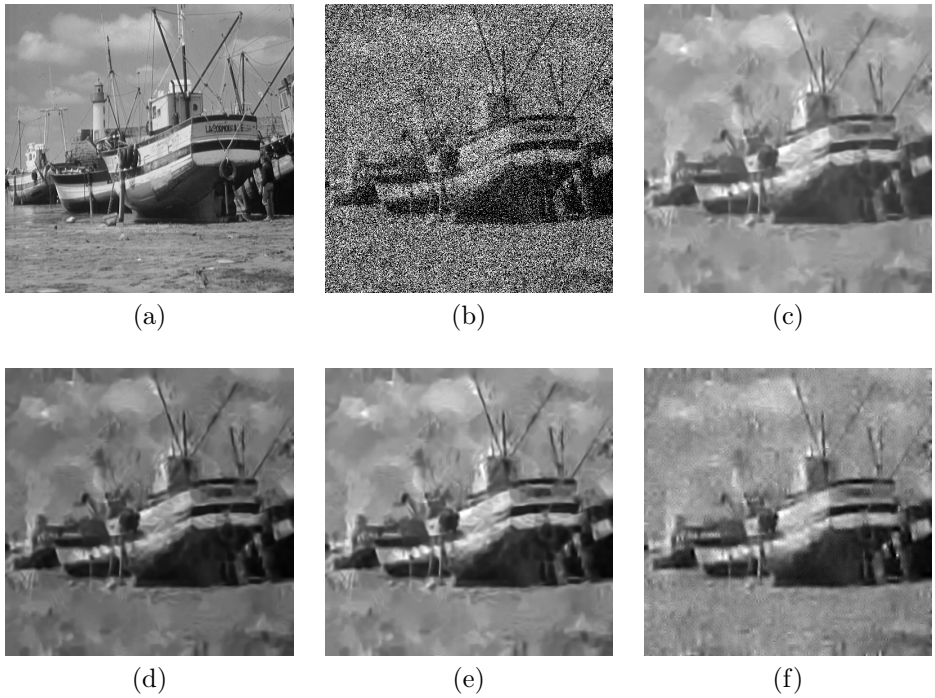
Figure 4.11: A 100×100 section of each of the images in Figure 4.10. (a) Original image, (b) Noisy image (peak = 80, $\sigma = 3$), (c) Denoised with BM3D and the exact unbiased inverse $\mathcal{I}_\sigma$, (d) Denoised with UWT/BDCT PURE-LET.



(a) (b) (c)

(d) (e) (f)

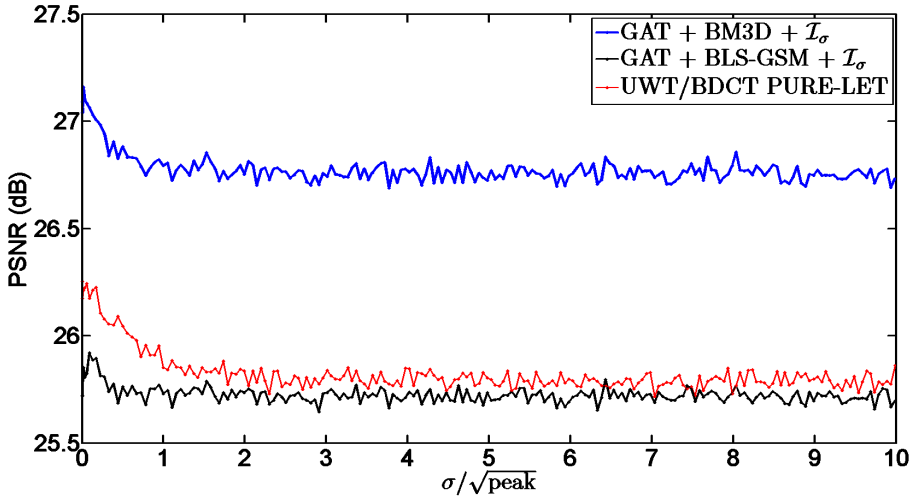
Figure 4.12: The denoising of Cameraman (256×256). (a) Original image, (b) Noisy image (peak = 1, $\sigma = 0.1$, PSNR = 3.20 dB), (c) Denoised with BM3D and the asymptotically unbiased inverse $\mathcal{I}_{\text{asy}}$ (PSNR = 15.55 dB), (d) Denoised with BM3D and the algebraic inverse $\mathcal{I}_{\text{alg}}$ (PSNR = 15.72 dB), (e) Denoised with BM3D and the exact unbiased inverse $\mathcal{I}_{\sigma}$ (PSNR = 20.23 dB), (f) Denoised with UWT/BDCT PURE-LET (PSNR = 20.35 dB).



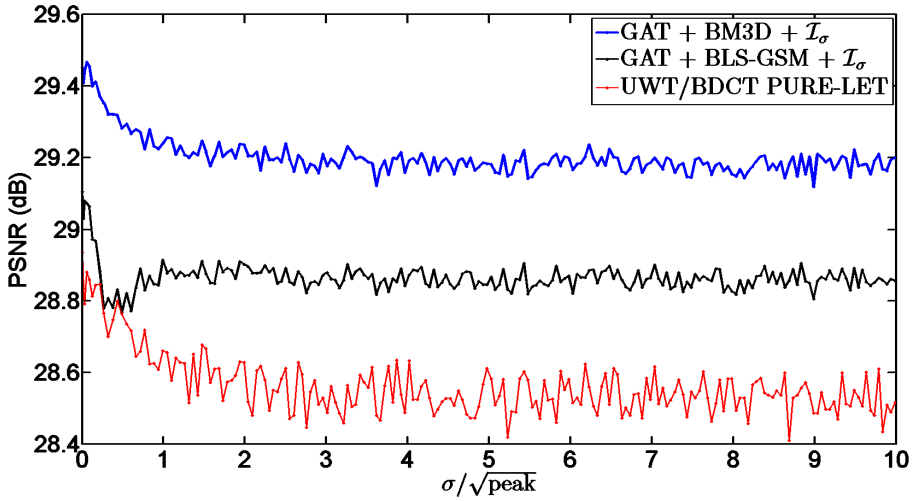
(a) (b) (c)

(d) (e) (f)

Figure 4.13: The denoising of Boat (512×512). (a) Original image, (b) Noisy image (peak = 1.5, $\sigma = 0.1$, PSNR = 4.64 dB), (c) Denoised with BM3D and the asymptotic inverse $\mathcal{I}_{\text{asy}}$ (PSNR = 20.18 dB), (d) Denoised with BM3D and the algebraic inverse $\mathcal{I}_{\text{alg}}$ (PSNR = 17.34 dB), (e) Denoised with BM3D and the exact unbiased inverse $\mathcal{I}_{\sigma}$ (PSNR = 22.30 dB), (f) Denoised with UWT/BDCT PURE-LET (PSNR = 22.31 dB).

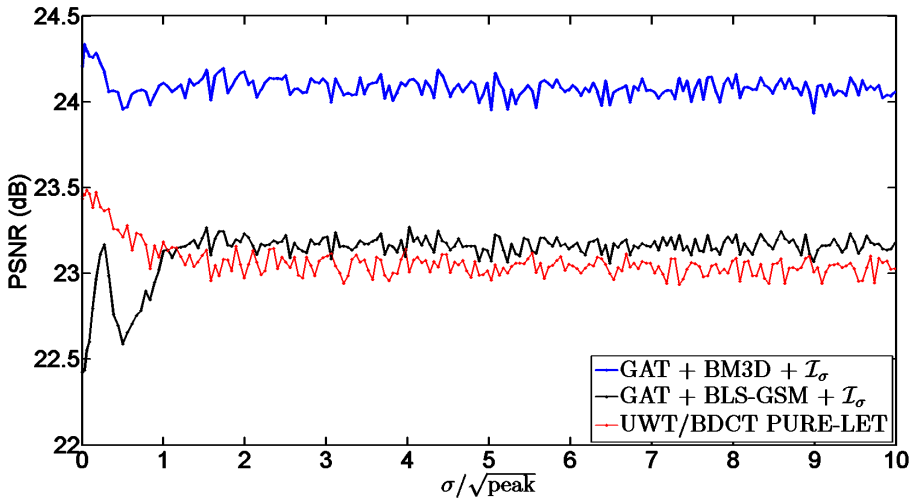



(a)

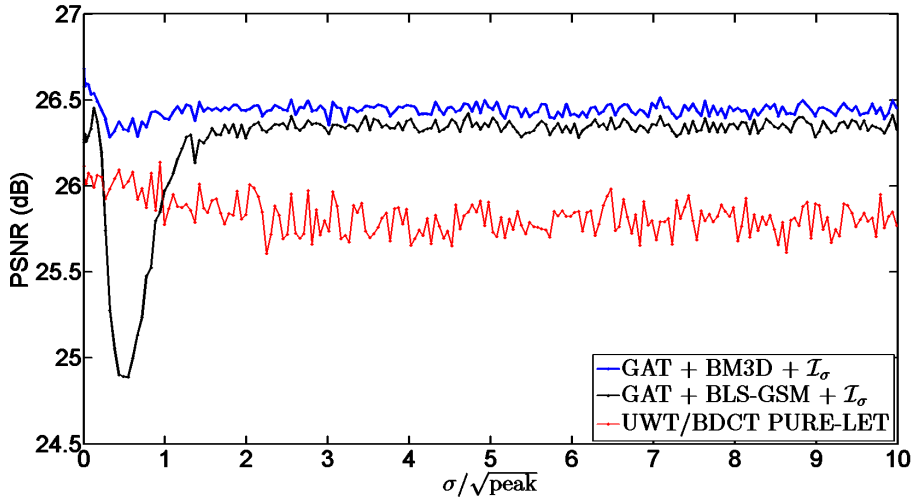



(b)

Figure 4.14: The results of denoising variance-stabilized data with either BM3D or BLS-GSM combined with the exact unbiased inverse $\mathcal{I}_\sigma$, and denoising with UWT/BDCT PURE-LET, when the PSNR of the noisy image is kept constant at 15 dB, but the ratio $\sigma/\sqrt{\text{peak}}$ is varied from 0 (i.e., the Poisson noise component dominates) to 10 (i.e., the Gaussian noise component dominates). Each PSNR value is an average over five random noise realizations. (a) Cameraman (256×256); peak = 14.90 when $\sigma = 0$, (b) Lena (512×512); peak = 15.95 when $\sigma = 0$.

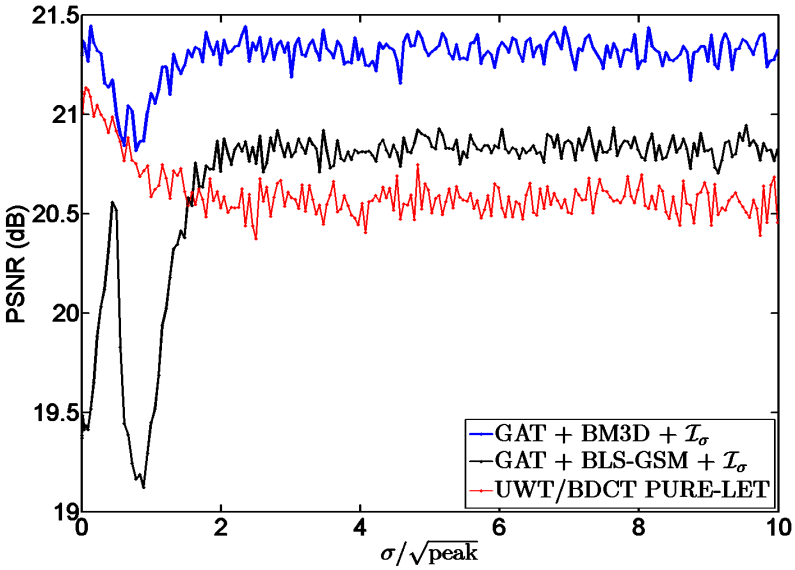



(a)

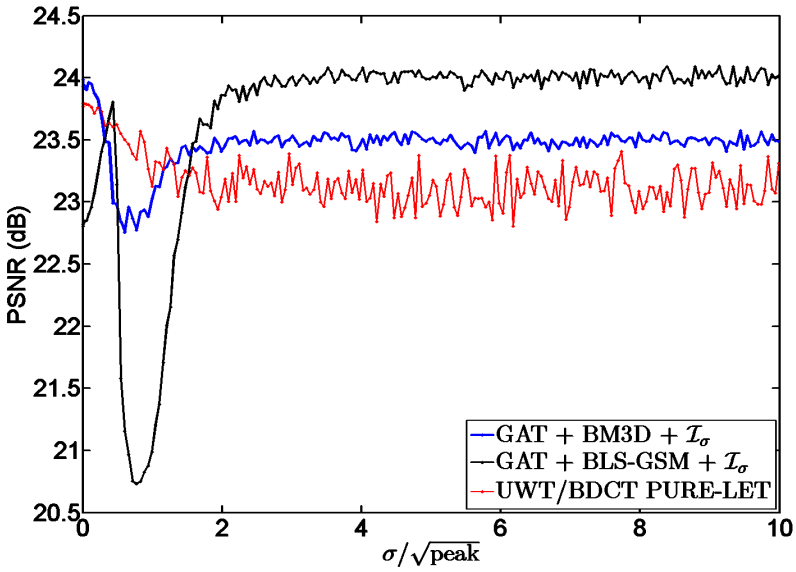



(b)

Figure 4.15: The results of denoising variance-stabilized data with either BM3D or BLS-GSM combined with the exact unbiased inverse $\mathcal{I}_\sigma$, and denoising with UWT/BDCT PURE-LET, when the PSNR of the noisy image is kept constant at 10 dB, but the ratio $\sigma/\sqrt{\text{peak}}$ is varied from 0 (i.e., the Poisson noise component dominates) to 10 (i.e., the Gaussian noise component dominates). Each PSNR value is an average over five random noise realizations. (a) Cameraman (256×256); peak = 4.71 when $\sigma = 0$, (b) Lena (512×512); peak = 5.05 when $\sigma = 0$.




(a)




(b)

Figure 4.16: The results of denoising variance-stabilized data with either BM3D or BLS-GSM combined with the exact unbiased inverse $\mathcal{I}_\sigma$, and denoising with UWT/BDCT PURE-LET, when the PSNR of the noisy image is kept constant at 5 dB, but the ratio $\sigma/\sqrt{\text{peak}}$ is varied from 0 (i.e., the Poisson noise component dominates) to 10 (i.e., the Gaussian noise component dominates). Each PSNR value is an average over five random noise realizations. (a) Cameraman (256×256); peak = 1.49 when $\sigma = 0$, (b) Lena (512×512); peak = 1.60 when $\sigma = 0$.

# Chapter 5

# Estimation of the Poisson-Gaussian noise model parameters

## 5.1 Introduction

As we have noted in many occasions, the Poisson-Gaussian noise model (2.7) can be characterized with three parameters: the scaling factor $\alpha$ of the Poisson component, the mean value $\mu$ of the Gaussian component, and the standard deviation $\sigma$ of the Gaussian component.

In practical applications involving noise removal, identifying a suitable noise model does not guarantee accurate denoising results per se, but for the best results we must also be able to produce reasonable estimates of the noise model parameters. While the specifications of the imaging device may provide some global noise parameters, they are usually not equally applicable to various individual cases. Thus, in some cases, the only practical option may be to estimate the noise parameters from the captured image itself.

Regardless of the particular noise model, the problem of estimating the parameter(s) associated with it can be approached either by taking advantage of multiple images (see, e.g., [74]), or by considering only a single image. For practical purposes, the latter is often necessary, as multiple images of the target may not be available. Thus, we will concentrate on parameter estimation from a single image. A variety of techniques exist for estimating the standard deviation of AWGN. For instance, this estimate can be obtained through mean or median absolute deviation (MAD) of the high frequency spectral coefficients of the image, where the coefficients can be, e.g., the high-detail 2-D wavelet coefficients [49], or an adaptive-size portion of the 3-D transform (DCT or wavelet) coefficients [43]. Alternatively, we may consider an implicit functional relationship between the image intensity and the noise variance [72]; this method is not restricted to AWGN, but can also be applied to other noise distributions. For a more comprehensive discussion on the various estimation methods, see e.g. the review in [20], or the

review of more elementary and older methods in [119].

On the other hand, parameter estimation for the case of Poisson-Gaussian noise is a much less studied problem. A notable approach is to segment the image into approximately uniform regions, locally approximate the expected value and standard deviation of each region, and finally obtain global estimates through regression of these local estimates [47, 64]. The problem has also recently been addressed through an expectation maximization algorithm [78]. While [47] and [78] mainly target parameter estimation from multiple images, [64] is able to provide very good results also for the single image case.

In this chapter, we first consider a general parametric noise model, and investigate the effect of inaccurate parameter estimation on variance stabilization. We present general results concerning this mismatch, and as a proof of concept, we construct a novel algorithm for estimating Poisson-Gaussian noise parameters from a single image using an iterative variance-stabilization scheme. For simplicity, we restrict ourselves to estimating the scaling factor $\alpha$ and the standard deviation $\sigma$ and assume the Gaussian component to have zero mean (i.e., $\mu = 0$). Specifically, each iteration begins with applying a VST to a random sub-block of the image, rendering the noise within the block approximately Gaussian with (ideally) unitary variance. Then we estimate the standard deviation of the stabilized AWGN block, compare it to the assumed standard deviation, and adjust our parameter estimates accordingly via gradient descent. At best, the results obtained with the proposed algorithm are comparable to those obtained with [64]. On the other hand, in low-intensity cases, the proposed algorithm in its current state is not always able to provide as accurate estimates as the latter. However, the main purpose of our algorithm is rather to expose the potential of the novel viewpoint in contrast with the more traditional methods, and to confirm the practical usefulness of the presented theoretical considerations.

This chapter is organized as follows: We begin by considering the mismatch between the estimated and true parameter values for a general noise distribution in Section 5.2. In particular, we show that under certain simplifying assumptions, the unitary contours of a function estimating the standard deviation of stabilized data are smooth in a neighbourhood of the true noise parameter values; we prove this result explicitly for the Poisson-Gaussian noise model. In Section 5.3 we introduce our Poisson-Gaussian parameter estimation algorithm and discuss its implementation details. Section 5.4 consists of experimental results, including comparisons to the regression-based parameter estimation algorithm [64]. Finally, Section 5.5 concludes the chapter and considers the limitations of the proposed algorithm.

## 5.2 Standard deviation contours

Let $z_\theta$ be a noisy image, where the noise model is parametrized by the vector $\theta = (\theta_1, \ldots, \theta_n)$, and $y \geq 0$ is the expected value of $z_\theta$. Further, let $\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_n)$ be the estimated values for the noise parameters $\theta$, and $f_{\hat{\theta}}$ a VST constructed using the estimated parameters. For instance, in the zero-mean Poisson-Gaussian case, $z_{\alpha,\sigma}$ is an image corrupted by (2.7), with $\mu = 0$. Then, we can set $f_{\hat{\alpha},\hat{\sigma}}$ to be, e.g., the GAT (2.18) with estimated parameters $\hat{\alpha}$ and $\hat{\sigma}$.

**Proposition 1** *Assume that*

1. *We can achieve exact stabilization with the correct noise parameters $\theta$:*

$$\operatorname{std}\left\{f_\theta\left(z_\theta\right)|y\right\} = 1 \ \forall y \geq 0. \tag{5.1}$$

2. *For any variance-stabilizing transformation $f_{\hat{\theta}}$ and any choice of parameters $\hat{\theta}$ and $\theta$, the conventional approximation (which follows from a first order Taylor expansion of $f_{\hat{\theta}}\left(z_\theta\right)$)*

$$\operatorname{std}\left\{f_{\hat{\theta}}\left(z_\theta\right)|y\right\} \approx \operatorname{std}\left\{z_\theta|y\right\} f'_{\hat{\theta}}\left(E\left\{z_\theta|y\right\}\right) \tag{5.2}$$

   *holds exactly (i.e., that the above symbol $\approx$ can be replaced by $=$).*

3. *The estimator $\mathfrak{E}_B$ of the standard deviation of AWGN noise successfully returns the mean standard deviation of the stabilized image block $B$.*

*When these assumptions hold, the AWGN estimator $\mathfrak{E}_B$ can be written as*

$$\mathfrak{E}_B\left\{f_{\hat{\theta}}\left(z_\theta\right)\right\} = \int \frac{\operatorname{std}\left\{z_\theta|y\right\}}{\operatorname{std}\left\{z_{\hat{\theta}}|y\right\}} p_B\left(y\right) dy =: F\left(\hat{\theta}\right), \tag{5.3}$$

*where $p_B\left(y\right)$ is the probability density (i.e., the normalized histogram) of $y$ over block $B$.*

**Proof.** Applying (5.2) to (5.1), we can write

$$\operatorname{std}\left\{f_\theta\left(z_\theta\right)|y\right\} = \operatorname{std}\left\{z_\theta|y\right\} f'_\theta\left(y\right) = 1,$$

and hence

$$f'_\theta\left(y\right) = \frac{1}{\operatorname{std}\left\{z_\theta|y\right\}}. \tag{5.4}$$

Thus, for the VST $f_{\hat{\theta}}$ with the estimated parameters $\hat{\theta}$, combining (5.1) and (5.4) leads to

$$\begin{aligned}
\operatorname{std}\left\{f_{\hat{\theta}}\left(z_\theta\right)|y\right\} &= \operatorname{std}\left\{z_\theta|y\right\} f'_{\hat{\theta}}\left(y\right) \\
&= \operatorname{std}\left\{z_\theta|y\right\} f'_\theta\left(y\right) \frac{f'_{\hat{\theta}}\left(y\right)}{f'_\theta\left(y\right)} \\
&= \operatorname{std}\left\{f_\theta\left(z_\theta\right)|y\right\} \frac{f'_{\hat{\theta}}\left(y\right)}{f'_\theta\left(y\right)} \\
&= \frac{f'_{\hat{\theta}}\left(y\right)}{f'_\theta\left(y\right)} = \frac{\operatorname{std}\left\{z_\theta|y\right\}}{\operatorname{std}\left\{z_\theta|y\right\}}.
\end{aligned} \tag{5.5}$$

Let us now consider an image block $B$, with $p_B\left(y\right)$ being the probability density of $y$ over this block. It then follows from (5.5) and assumption 3 that

$$\begin{aligned}
\mathfrak{E}_B\left\{f_{\hat{\theta}}\left(z_\theta\right)\right\} &= \int \operatorname{std}\left\{f_{\hat{\theta}}\left(z_\theta\right)|y\right\} p_B\left(y\right) dy \\
&= \int \frac{\operatorname{std}\left\{z_\theta|y\right\}}{\operatorname{std}\left\{z_{\hat{\theta}}|y\right\}} p_B\left(y\right) dy.
\end{aligned} \tag{5.6}$$

■

As an example, for Poisson-Gaussian noise, we can explicitly write $\text{std}\{z_{\hat{\alpha},\hat{\sigma}}|y\} = \sqrt{\hat{\alpha}y + \hat{\sigma}^2}$. Thus, we can express (5.3) as

$$\mathfrak{E}_B\left\{f_{\hat{\alpha},\hat{\sigma}}\left(z_{\alpha,\sigma}\right)\right\} = \int \sqrt{\frac{\alpha y + \sigma^2}{\hat{\alpha}y + \hat{\sigma}^2}} p_B\left(y\right) dy = F\left(\hat{\alpha},\hat{\sigma}\right). \tag{5.7}$$

In what follows, we study the behaviour of $F\left(\hat{\alpha},\hat{\sigma}\right)$ in the Poisson-Gaussian case. Our aim is to show that, given the assumptions in Proposition 1, a function $F\left(\hat{\alpha},\hat{\sigma}\right)$ estimating the standard deviation of a stabilized block (i.e., an AWGN standard deviation estimator) has a well-behaving unitary contour near the true parameter values $\alpha$, $\sigma$. More precisely, we show that such a contour is locally a simple smooth curve in a neighbourhood of $(\alpha,\sigma)$ in the $(\hat{\alpha},\hat{\sigma})$ plane.

Clearly, $F\left(\alpha,\sigma\right) = 1$, when all of our assumptions hold. Let us consider the implicit function theorem for $F$ at $(\alpha,\sigma)$. In particular, the theorem leads us to the following proposition:

**Proposition 2** *Let $F$ be defined as in (5.7). According to the implicit function theorem, the solution of $F\left(\hat{\alpha},\hat{\sigma}\right) = 1$ is locally a simple smooth curve in a neighbourhood of $(\alpha,\sigma)$ in the $(\hat{\alpha},\hat{\sigma})$ plane, provided that either $\frac{\partial F}{\partial \hat{\alpha}}\left(\alpha,\sigma\right) \neq 0$ or $\frac{\partial F}{\partial \hat{\sigma}}\left(\alpha,\sigma\right) \neq 0$.*

**Proof.** Let us show that either $\frac{\partial F}{\partial \hat{\alpha}}\left(\alpha,\sigma\right) < 0$ or $\frac{\partial F}{\partial \hat{\sigma}}\left(\alpha,\sigma\right) < 0$ hold. With straightforward calculations, we see that

$$\begin{aligned}\frac{\partial F}{\partial \hat{\alpha}} &= \int \frac{\partial}{\partial \hat{\alpha}} \sqrt{\frac{\alpha y + \sigma^2}{\hat{\alpha}y + \hat{\sigma}^2}} p_B\left(y\right) dy \\ &= -\int \frac{y}{2} \frac{\sqrt{\left(\alpha y + \sigma^2\right)\left(\hat{\alpha}y + \hat{\sigma}^2\right)}}{\left(\hat{\alpha}y + \hat{\sigma}^2\right)^2} p_B\left(y\right) dy,\end{aligned}$$

and similarly

$$\frac{\partial F}{\partial \hat{\sigma}} = -\int \sigma \frac{\sqrt{\left(\alpha y + \sigma^2\right)\left(\hat{\alpha}y + \hat{\sigma}^2\right)}}{\left(\hat{\alpha}y + \hat{\sigma}^2\right)^2} p_B\left(y\right) dy.$$

Thus,

$$\frac{\partial F}{\partial \hat{\alpha}}\left(\alpha,\sigma\right) = -\int \frac{y}{2} \frac{1}{\alpha y + \sigma^2} p_B\left(y\right) dy, \tag{5.8}$$

$$\frac{\partial F}{\partial \hat{\sigma}}\left(\alpha,\sigma\right) = -\sigma \int \frac{1}{\alpha y + \sigma^2} p_B\left(y\right) dy. \tag{5.9}$$

The partial derivative (5.9) equals 0 if and only if $\sigma = 0$. However, if $\sigma = 0$, then $\frac{\partial F}{\partial \hat{\alpha}}\left(\alpha,\sigma\right) = -\frac{1}{2\alpha} < 0$. Therefore we have that either $\frac{\partial F}{\partial \hat{\alpha}}\left(\alpha,\sigma\right) < 0$ or $\frac{\partial F}{\partial \hat{\sigma}}\left(\alpha,\sigma\right) < 0$ hold, and hence $F\left(\hat{\alpha},\hat{\sigma}\right) = 1$ is locally a simple smooth curve in a neighbourhood of $(\alpha,\sigma)$. ■

The implicit function theorem also states that the slope of the curve at $(\alpha,\sigma)$ is given by the ratio of the above partial derivatives (5.8) and (5.9). This highlights the fact that, for different blocks $B$, we have differently oriented (and thus stable

and non-degenerate) curves intersecting each other, provided that the different blocks have different enough normalized histograms $p_B(y)$. In order to illustrate the above results, Figure 5.1(b) shows ten contours $F(\hat{\alpha}, \hat{\sigma}) = 1$ computed from ten randomly selected $32 \times 32$ blocks of the Piecewise test image (peak = 120) shown in Figure 5.1(a). Figure 5.2 shows a similar visualization for a low-intensity case (Lena, peak = 5); in this case it is clear that our initial assumptions do not all hold, and the contours fail to intersect at a point.

Note that while we proved Proposition 2 only in the Poisson-Gaussian case, we expect the proposition to hold also for other common distribution families, with the proof proceeding analogously (showing that at least one partial derivative of $F(\hat{\theta})$ does not vanish in a neighbourhood of $\theta$).

## 5.3 The iterative parameter estimation algorithm

Inspired by the theoretical results presented in Section 5.2, we construct an algorithm estimating the parameters $\alpha$ and $\sigma$ of zero-mean Poisson-Gaussian noise through variance stabilization of different sub-blocks of an image, iteratively updating the respective estimates $\hat{\alpha}$ and $\hat{\sigma}$.

In particular, the proposed algorithm considers a different random image block in each successive step, stabilizes its variance, and adjusts the current estimates $\hat{\alpha}_1$ and $\hat{\sigma}_1$ according to a steepest descent criterion. Moreover, in order to regulate the behaviour of the estimates, the adjustments also take into account a number of previous estimates, with weights inversely proportional to their variances. In what follows, we refer to the latter procedure as *dampening*, and the dampened estimates are denoted by $\bar{\alpha}_1$ and $\bar{\sigma}_1$.

Let us first present a general overview of the estimation process, after which we look at each step in more detail. The main steps of the algorithm are as follows:

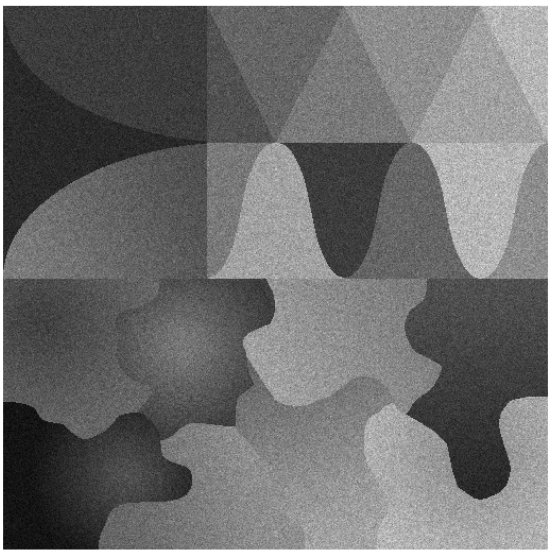
1. **Iteration k = 1**: Initialize the variables (the undampened estimates $\hat{\alpha}_1$ and $\hat{\sigma}_1$, the dampened estimates $\bar{\alpha}_1$ and $\bar{\sigma}_1$, and their respective variance estimates).

2. **Iteration k > 1**: Choose a random block $B$ from the noisy image $z_{\alpha,\sigma}$ and apply a variance-stabilizing transformation $f_{\bar{\alpha}_{k-1},\bar{\sigma}_{k-1}}(z_{\alpha,\sigma})$ to it (for instance, the generalized Anscombe transformation (2.18) with the parameters $\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}$).

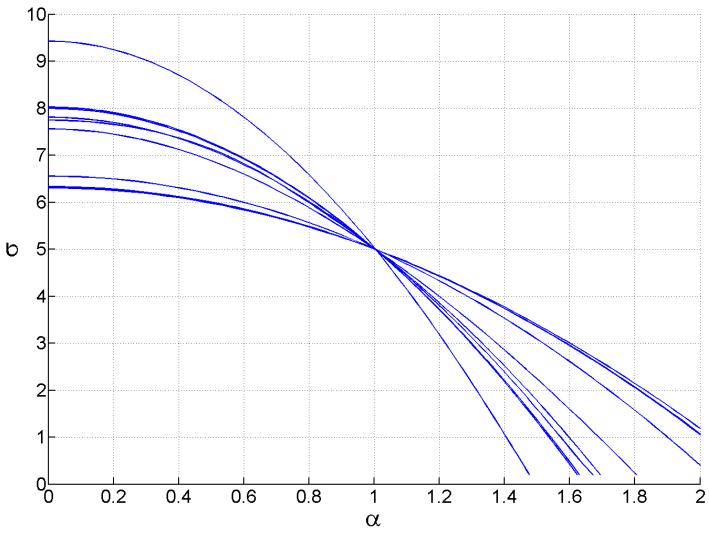
3. Compute an estimate $F(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}) = \mathfrak{E}_B\left\{f_{\bar{\alpha}_{k-1},\bar{\sigma}_{k-1}}(z_{\alpha,\sigma})\right\}$ for the standard deviation of the stabilized image block (using an AWGN standard deviation estimator $\mathfrak{E}$, such as the MAD estimator).

4. Move the estimates $\hat{\alpha}$, $\hat{\sigma}$ towards the negative gradient:

$$\begin{aligned} \hat{\alpha}_k &= \bar{\alpha}_{k-1} - \gamma_k \frac{\partial F}{\partial \bar{\alpha}_{k-1}}(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}), \\ \hat{\sigma}_k &= \bar{\sigma}_{k-1} - \gamma_k \frac{\partial F}{\partial \bar{\sigma}_{k-1}}(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}), \end{aligned}$$



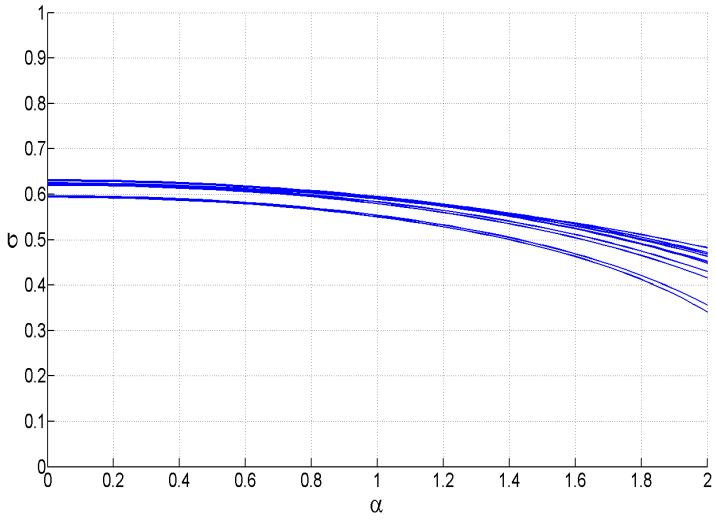
(a)




(b)

Figure 5.1: (a) A piecewise smooth noisy $512 \times 512$ test image ("Piecewise"), with peak intensity value 120, and noise parameters $\alpha = 1$, $\sigma = 5$, $\mu = 0$. (b) Ten standard deviation contours $F(\hat{\alpha}, \hat{\sigma}) = 1$ computed from ten randomly selected $32 \times 32$ blocks of the test image (a).

(a)




(b)

Figure 5.2: (a) Noisy Lena ($512 \times 512$), with peak intensity value 5, and noise parameters $\alpha = 1$, $\sigma = 0.5$, $\mu = 0$. (b) Ten standard deviation contours $F(\hat{\alpha}, \hat{\sigma}) = 1$ computed from ten randomly selected $32 \times 32$ blocks of the test image (a).

where $\gamma_k$ is a step parameter (in particular, we use a step size defined by Eq. (5.12)).

5. Use the undampened estimates $\hat{\sigma}_n, \ldots, \hat{\sigma}_k$ for updating the dampened estimate $\bar{\sigma}_k$:

$$\bar{\sigma}_k = \sum_{i=n}^{k} \mu_i^{(\sigma)} \hat{\sigma}_i,$$

where each weight $\mu_i^{(\sigma)}$ is inversely proportional to the variance of the undampened estimate $\hat{\sigma}_i$. Similarly, compute the dampened estimate

$$\bar{\alpha}_k = \sum_{i=n}^{k} \mu_i^{(\alpha)} \hat{\alpha}_i.$$

6. Check if either the maximum number of iterations (e.g., $k = 8000$) is reached or the relative difference satisfies the inequality

$$\Delta = \frac{|\bar{\alpha}_k - \bar{\alpha}_{k-1}|}{\bar{\alpha}_k} + \frac{|\bar{\sigma}_k - \bar{\sigma}_{k-1}|}{\bar{\sigma}_k} < \tau,$$

where the pre-defined tolerance is, e.g., $\tau = 10^{-7}$. If either of these conditions are met, set the final values for the estimated parameters as $\hat{\alpha} = \bar{\alpha}_k$ and $\hat{\sigma} = \bar{\sigma}_k$. Otherwise, move on to iteration $k + 1$ (return to step 2).

Now, let us examine each of the above steps and their implementation details more carefully.

### 5.3.1 Initialization of the estimates

We initialize $\hat{\sigma}$ by estimating the standard deviation of the non-stabilized whole image $z_{\alpha,\sigma}$ (e.g., with the MAD estimator):

$$\hat{\sigma}_1 = \mathfrak{E}\{z_{\alpha,\sigma}\} \approx \operatorname{std}\{z_{\alpha,\sigma}|y\}.$$

We also set the initial dampened estimate to $\bar{\sigma}_1 = \hat{\sigma}_1$, since at this point there are no previous estimates to be weighed in. The variance of $\bar{\sigma}_1 = \hat{\sigma}_1$ can be approximated using Eq. (58) in [64], yielding

$$\bar{v}_1^{(\sigma)} = \operatorname{var}\{\hat{\sigma}_1\} = \hat{\sigma}_1^2 \cdot \frac{1.35}{|z_{\alpha,\sigma}| + 1.5},$$

where $|z_{\alpha,\sigma}|$ is the number of pixels in the whole image $z_{\alpha,\sigma}$.

For $\hat{\alpha}$, we fully rely on the iterative updating and dampening procedure to steer the estimate towards the right direction, and simply initialize it to a value close to zero (e.g., $\hat{\alpha}_1 = 10^{-4}$). This is done for the purpose of not making implicit assumptions about the data (such as assuming $\alpha$ being close to unity). Due to this initialization, we also set the estimated variance $\bar{v}_1^{(\alpha)}$ to zero.

### 5.3.2 Stabilizing the variance of a random block

As shown in Section 5.2, the unitary variance contours corresponding to different stabilized image blocks are locally smooth in the $(\hat{\alpha}, \hat{\sigma})$ plane. Moreover, they are differently oriented and intersect each other, provided that the probability densities of these image blocks are different enough. Thus, at each iteration $k > 1$ after the initialization step, we process only a random sub-block $B$ of the image $z_{\alpha,\sigma}$. In the most straightforward approach, a fixed block size (e.g., $32 \times 32$) is used throughout the iterations, but adapting the size based on a suitable heuristic may be considered in order to speed up the computations.

After extracting the random block, it is processed with a VST $f$ in order to stabilize its variance to (approximately) unity. Specifically, we use the generalized Anscombe transformation (2.18), with $\alpha$ and $\sigma$ replaced by the dampened parameter estimates $\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}$ from the previous iteration. Alternatives to the generalized Anscombe transformation do exist, such as an optimized nonparametric transformation proposed in [66].

### 5.3.3 Estimating the standard deviation

After applying the VST on the random block $B$, the data can be treated as Gaussian with unitary variance. Thus, we can compute the standard deviation estimate $F\left(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}\right)$ for the stabilized image block $f_{\bar{\alpha}_{k-1},\bar{\sigma}_{k-1}}\left(z_{\alpha,\sigma}\right)$ as

$$F\left(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}\right) = \mathfrak{E}_B\left\{f_{\bar{\alpha}_{k-1},\bar{\sigma}_{k-1}}\left(z_{\alpha,\sigma}\right)\right\},$$

where $\mathfrak{E}$ is any standard deviation estimator $\mathfrak{E}$ designed for the additive white Gaussian noise case. In particular, our estimate is obtained as the mean absolute deviation (MAD) of the wavelet detail coefficients of the block; this is the same MAD estimator that was used in computing the initial estimate $\hat{\sigma}_1$ in Section 5.3.1.

### 5.3.4 Iterative updating of the estimates

In order to regulate the behaviour of the estimates, the algorithm uses two sets of estimates for $\sigma$: the undampened estimates $\hat{\sigma}$, and the dampened estimates $\bar{\sigma}$. Analogously, two sets of estimates $\hat{\sigma}$ and $\bar{\alpha}$ are used for $\alpha$. Thus, for simplicity, we do not repeat the similar equations below for $\alpha$, but consider only the case of $\sigma$.

The algorithm stores the undampened estimates $\hat{\sigma}_1, \ldots, \hat{\sigma}_k$ from each iteration, from which the dampened estimate is computed as

$$\bar{\sigma}_k = \sum_{i=n}^{k} \mu_i^{(\sigma)} \hat{\sigma}_i,$$

where each weight $\mu_i^{(\sigma)}$ is inversely proportional to the variance of $\hat{\sigma}$, and $n$ is a free tuning parameter that can be thought of as the "forgetting factor" of the algorithm. The practical computations regarding the weighting are elaborated in Section 5.3.5, but in order to proceed chronologically, let us now look at how we update the undampened estimates $\hat{\sigma}$ and $\hat{\alpha}$ with a steepest descent (i.e., gradient descent) approach.

Our goal here is to update the estimates $\hat{\sigma}$ and $\hat{\alpha}$ in such a way that we move in the direction of the negative gradient, ideally towards a target value $\text{std}\left\{f_{\bar{\alpha},\bar{\sigma}}\left(z_{\alpha,\sigma}\right)|y\right\}$. With exact stabilization, this target value would be a level set $\text{std}\left\{f_{\bar{\alpha},\bar{\sigma}}\left(z_{\alpha,\sigma}\right)|y\right\} = 1$ (assuming that the standard deviation function is a plane with respect to $\alpha$ and $\sigma$). However, in the case of Poisson-Gaussian distributed data, the stabilization is bound to be inexact (see [33] and [52]), in particular for low values of $y$. Moreover, as in practise we don't know the parameter values defining $z_{\alpha,\sigma}$, we replace it by $z_{\bar{\alpha},\bar{\sigma}}$ when estimating the standard deviation. Explicitly, the gradient descent step is performed by setting

$$\hat{\alpha}_k = \bar{\alpha}_{k-1} - \gamma_k \frac{\partial F}{\partial \bar{\alpha}}\left(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}\right), \tag{5.10}$$

$$\hat{\sigma}_k = \bar{\sigma}_{k-1} - \gamma_k \frac{\partial F}{\partial \bar{\sigma}}\left(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}\right), \tag{5.11}$$

where the step parameter $\gamma_k$ equals

$$\gamma_k = \frac{F\left(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}\right) - \text{std}\left\{f_{\bar{\alpha}_{k-1},\bar{\sigma}_{k-1}}\left(z_{\bar{\alpha},\bar{\sigma}}\right)|y\right\}}{\frac{\partial F}{\partial \bar{\alpha}_{k-1}}\left(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}\right)^2 + \frac{\partial F}{\partial \bar{\sigma}_{k-1}}\left(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}\right)^2}. \tag{5.12}$$

This step parameter is constructed in such a way that (5.10) and (5.11) move the estimates $\bar{\alpha}_{k-1}$, $\bar{\sigma}_{k-1}$ directly to the closest intersection between the $F\left(\bar{\alpha}_{k-1}, \bar{\sigma}_{k-1}\right)$ plane and the $\text{std}\left\{f_{\bar{\alpha}_{k-1},\bar{\sigma}_{k-1}}\left(z_{\bar{\alpha},\bar{\sigma}}\right)|y\right\}$ plane.

It is worth noting that the *dampened* estimates $\bar{\sigma}_{k-1}, \bar{\alpha}_{k-1}$ from the previous iteration determine the new *undampened* estimates $\hat{\sigma}_k, \hat{\alpha}_k$; these are then further used in computing the dampened estimates for the $k$'th iteration. As a final detail, in order to keep the estimates strictly positive, we set $\hat{\sigma}_k = \max\left(\epsilon, \hat{\sigma}_k\right)$ and $\hat{\alpha}_k = \max\left(\epsilon, \hat{\alpha}_k\right)$, where, e.g., $\epsilon = 10^{-5}$. This prevents possible complications in the generalized Anscombe transformation (2.18), where division by zero might otherwise occur.

### 5.3.5 Computing the dampened estimates

Returning to the concept of dampening the estimates by using a linear combination of the undampened estimates, we define the undampened variance $v_k^{(\sigma)}$ as

$$v_k^{(\sigma)} = \text{var}\left\{\hat{\sigma}_n, \hat{\sigma}_{n+1}, \ldots, \hat{\sigma}_{k-1}, \hat{\sigma}_k\right\}, \tag{5.13}$$

where $1 \leq n \leq k$. In other words, the undampened variance $v_k^{(\sigma)}$ is computed over the $k - n + 1$ newest undampened estimates of $\sigma$. We can choose the forgetting factor $n$ as we see best; for instance, we may use the 50 newest estimates, or we may vary $n$ as a function of $k$.

Before updating the dampened variance $\bar{v}_k^{(\sigma)} = \text{var}\left\{\bar{\sigma}_k\right\}$, let us define $\lambda_k^{(\sigma)}$ as

$$\lambda_k^{(\sigma)} = \frac{1}{\bar{v}_{k-1}^{(\sigma)} + \varepsilon^{(\sigma)} v_k^{(\sigma)}} \Big/ \left(\frac{1}{\bar{v}_{k-1}^{(\sigma)} + \varepsilon^{(\sigma)} v_k^{(\sigma)}} + \frac{1}{v_k^{(\sigma)}}\right), \tag{5.14}$$

where the additional adaptive weight parameter is set to $\varepsilon^{(\sigma)} = \max\left(10^{-6}, 4/k\right)$. Now we can express the update of $\bar{v}_k^{(\sigma)}$ as

$$\bar{v}_k^{(\sigma)} = \bar{v}_{k-1}^{(\sigma)} \left(\lambda_k^{(\sigma)}\right)^2 + v_k^{(\sigma)} \left(1 - \lambda_k^{(\sigma)}\right)^2. \tag{5.15}$$

Finally, we obtain the updated dampened estimate $\bar{\sigma}_k$ itself by

$$\bar{\sigma}_k = \bar{\sigma}_{k-1}\lambda_k^{(\sigma)} + \hat{\sigma}_k \left(1 - \lambda_k^{(\sigma)}\right). \tag{5.16}$$

As in the previous section, the case for $\alpha$ is completely analogous, so we omit the explicit equations describing the updating of $\bar{\alpha}_k$.

### 5.3.6 Obtaining the final estimates

In the end of each iteration, we must decide whether our current estimates are good and stable enough, or if further iterations are required. Specifically, we consider the estimates to have reached adequate stability, if the relative difference $\Delta$ satisfies the inequality

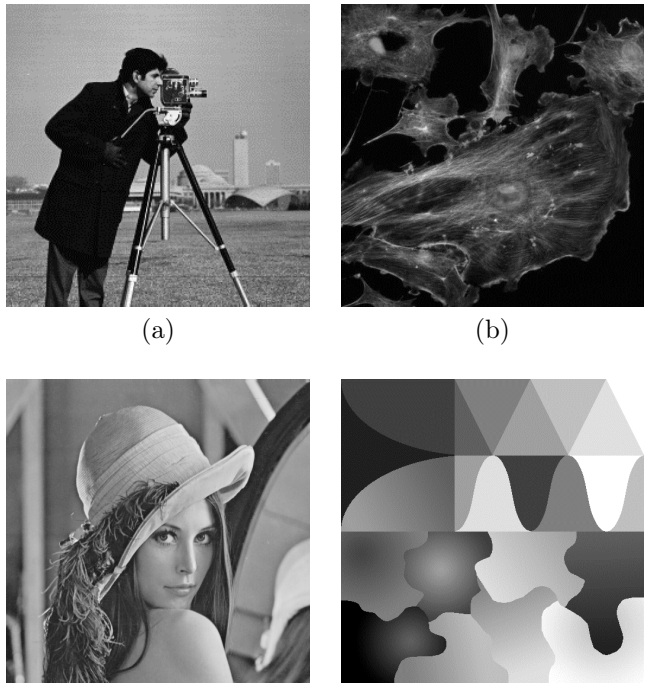
$$\Delta = \frac{|\bar{\alpha}_k - \bar{\alpha}_{k-1}|}{\bar{\alpha}_k} + \frac{|\bar{\sigma}_k - \bar{\sigma}_{k-1}|}{\bar{\sigma}_k} < \tau, \tag{5.17}$$

where the pre-defined tolerance is, e.g., $\tau = 10^{-7}$. In addition to this stopping criterion, we also define an upper limit for the number of iterations (e.g., $k = 8000$) to ensure the halting of the algorithm. If either of these conditions are met, we consider the newest dampened estimates $\bar{\alpha}_k, \bar{\sigma}_k$ to be our final estimates, and thus set $\hat{\alpha} = \bar{\alpha}_k$ and $\hat{\sigma} = \bar{\sigma}_k$. Otherwise, we choose a new random image block and move on to iteration $k + 1$.

## 5.4 Experiments

We investigate the accuracy and effectiveness of the proposed parameter estimation algorithm with the Cameraman ($256 \times 256$), Fluorescent Cells ($512 \times 512$), Lena ($512 \times 512$), and Piecewise ($512 \times 512$) test images shown in Figure 5.3, each scaled to eight peak intensity values $(1, 2, 5, 10, 20, 30, 60, 120)$ and corrupted by various combinations of the Poisson-Gaussian noise parameters $\alpha$ and $\sigma$. As noted earlier, we assume the Gaussian noise component to have zero mean (i.e., $\mu = 0$).

The parameters $\alpha$ and $\sigma$ are estimated from each noisy image with the proposed algorithm, and with the algorithm presented in [64] for a comparison. For the former, we use a constant block size of $32 \times 32$ pixels, forgetting factor $n = \lceil 3\sqrt{k} \rceil + 1$ (where $\lceil \cdot \rceil$ is the ceiling function), and stopping criteria $\tau = 10^{-7}$, $k = 8000$. We also compare the denoising performance associated with the obtained parameter estimates by relaying them to BM3D, which is combined with variance stabilization. More specifically, we use the familiar three-step denoising process: First, the GAT is applied to the noisy image. Then, the stabilized image is denoised with BM3D, using the estimated parameters as input. Finally, the denoised data is processed with the exact unbiased inverse of the GAT, producing the final estimate of the noise-free image. The denoising result is then evaluated by two performance



(a) (b)

(c) (d)

Figure 5.3: The test images used in the experiments. (a) Cameraman ($256 \times 256$), (b) Fluorescent Cells ($512 \times 512$), (c) Lena ($512 \times 512$), (d) Piecewise ($512 \times 512$).

metrics: the peak signal-to-noise ratio (PSNR), and the structural similarity index value (SSIM) [137].

Table 5.1 presents the parameter estimation and denoising results for the Cameraman and Fluorescent Cells images, when $\alpha = 1$ is kept constant. Corresponding results for the Lena and Piecewise images are presented in Table 5.2. Further, Table 5.3 consists of similar results for various other values of $\alpha$. We observe that unless the peak intensity is very low (peak $\leq 2$), both estimation algorithms generally lead to comparable PSNR and SSIM results. Even though [64] is overall slightly more accurate in estimating the parameters, it typically does not help in producing notably better denoising results. On the other hand, the proposed algorithm tends to struggle with low intensity data, often underestimating $\alpha$ in particular. As Table 5.3 shows, this underestimation is especially notable for higher values of $\alpha$. This deficiency is likely caused by the inherent inability of the GAT to perform accurate variance stabilization in regions with low mean intensity values, and could potentially be mitigated by leveraging an optimized nonparametric variance-stabilizing transformation [66] instead of the GAT.

Figure 5.4 illustrates how the estimated parameter values (both undampened and dampened estimates) typically evolve as the proposed algorithm performs the iterative updating. In this particular example, the parameters are estimated from the Cameraman image, with peak $= 20$, $\sigma = 2$, $\alpha = 1$; the estimates converge to

$\bar{\sigma} = 1.978$, $\bar{\alpha} = 1.063$ after 1878 iterations, which takes about 6 seconds with our unoptimized Matlab implementation. Note that the execution time depends on the chosen block size and especially on the convergence threshold $\tau$. For instance, if we choose a less stringent threshold $\tau = 10^{-6}$ instead of the default $\tau = 10^{-7}$, the estimation takes only 1 second (352 iterations), still producing good estimates $\bar{\sigma} = 2.038$, $\bar{\alpha} = 1.034$ and practically identical denoising results. Figure 5.4 also shows the importance of the dampening procedure, as the undampened estimates alone are not stable enough without some form of regularization.

Figure 5.5 shows a similar illustration for the low-intensity case of Fluorescent Cells with peak = 1, $\sigma = 0.1$, $\alpha = 1$. In this case, the convergence threshold $\tau = 10^{-7}$ is not met within the pre-defined limit of $k = 8000$ iterations; the relative difference (5.17) remains in the order of $\Delta = 10^{-4}$, and the estimated parameters are $\bar{\sigma} = 0.172$, $\bar{\alpha} = 0.676$. Despite the notable underestimation of $\alpha$, the denoising performance in this case remains comparable to that achieved using the estimator [64], not only numerically (see Table 5.1). but also visually, as shown in Figure 5.6. On the other hand, in those low-intensity cases where we obtain numerically inferior results, the visual results tend to be inferior as well, containing more artifacts than in the result associated with [64]. This is illustrated in Figure 5.7 for the Piecewise image with peak = 1, $\sigma = 0.1$, $\alpha = 2.5$.

Finally, Figures 5.8–5.10 provide another perspective for the obtained estimation accuracy. In particular, they compare the estimated and true standard deviation curves ($\operatorname{std}\{z_{\bar{\alpha},\bar{\sigma}}|y\} = \sqrt{\bar{\alpha}y + \bar{\sigma}^2}$ and $\operatorname{std}\{z_{\alpha,\sigma}|y\} = \sqrt{\alpha y + \sigma^2}$, respectively) for the three example cases discussed above. In addition, the histograms of the corresponding noisy and noise-free images are overlaid to each comparison. This aids in identifying the intensity ranges, where the accuracy or inaccuracy of the estimates will have practical significance. For instance, Figure 5.9 (Fluorescent Cells, peak = 1, $\sigma = 0.1$, $\alpha = 1$) is in agreement with our observation that the underestimation of $\alpha$ did not translate into a notably worse denoising result; in fact, our estimates are reasonably accurate in the low-intensity range where most of the noisy data is concentrated. Conversely, in the most problematic case of Piecewise with peak = 1, $\sigma = 0.1$, $\alpha = 2.5$ (Figure 5.10), our estimates are inaccurate also in the low-intensity range, where the noisy histogram peaks.

## 5.5 Discussion

We investigated the effect of inaccurate parameter estimation on variance stabilization for a general noise distribution. In particular, we showed that under some simplifying assumptions, the unitary contours of a function estimating the standard deviation of stabilized data are smooth in a neighbourhood of the true parameter values. We derived a general expression for this function, proved the smoothness explicitly for the Poisson-Gaussian noise model, and postulated the result to hold also for other common distribution families.

Based on the results, we proposed a variance-stabilization based iterative algorithm for estimating the parameters of Poisson-Gaussian noise, discussed its implementation details, and compared its estimation performance against an algorithm based on segmentation, local estimates and regression [64]. Further, we compared the corresponding denoising results achieved with the estimated param-

Table 5.1: Estimated parameter values $\hat{\alpha}$ and $\hat{\sigma}$ for the Cameraman and Fluorescent Cells images. The proposed algorithm and the algorithm presented in [64] are compared for various peak intensity values, $\sigma = \text{peak}/10$ and a constant $\alpha = 1$. In addition, the denoising performance (GAT (2.18) + BM3D + exact unbiased inverse $\mathcal{I}_\sigma$ of the GAT) associated with these parameter estimates is compared in terms of both PSNR and SSIM [137].

| Image | Peak | $\sigma$ | Noisy | Proposed algorithm | | | | [64] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PSNR | $\hat{\alpha}$ | $\hat{\sigma}$ | PSNR | SSIM | $\hat{\alpha}$ | $\hat{\sigma}$ | PSNR | SSIM |
| Cameraman (256×256) | 1 | 0.1 | 3.20 | 0.725 | 0.255 | 20.29 | 0.5537 | 0.957 | 0.141 | 20.34 | 0.5923 |
| | 2 | 0.2 | 6.10 | 0.827 | 0.287 | 21.83 | 0.5866 | 0.948 | 0.118 | 21.77 | 0.5994 |
| | 5 | 0.5 | 9.83 | 0.912 | 0.528 | 24.19 | 0.6923 | 1.015 | 0.459 | 24.19 | 0.7115 |
| | 10 | 1 | 12.46 | 1.000 | 0.995 | 25.54 | 0.7567 | 1.095 | 0.896 | 25.51 | 0.7600 |
| | 20 | 2 | 14.75 | 1.063 | 1.978 | 26.81 | 0.7882 | 1.008 | 2.075 | 26.81 | 0.7888 |
| | 30 | 3 | 15.88 | 0.974 | 3.078 | 27.25 | 0.8040 | 1.053 | 3.038 | 27.24 | 0.8048 |
| | 60 | 6 | 17.48 | 1.067 | 6.036 | 28.07 | 0.8197 | 0.973 | 6.224 | 28.08 | 0.8203 |
| | 120 | 12 | 18.58 | 0.840 | 12.782 | 28.62 | 0.8372 | 0.907 | 12.505 | 28.64 | 0.8379 |
| Fluorescent Cells (512×512) | 1 | 0.1 | 7.24 | 0.676 | 0.172 | 24.04 | 0.5751 | 0.949 | 0.183 | 24.19 | 0.5655 |
| | 2 | 0.2 | 10.02 | 0.643 | 0.296 | 25.35 | 0.6381 | 0.920 | 0.153 | 25.55 | 0.6184 |
| | 5 | 0.5 | 13.40 | 0.687 | 0.605 | 27.23 | 0.7017 | 0.984 | 0.483 | 27.54 | 0.7140 |
| | 10 | 1 | 15.52 | 0.916 | 0.996 | 28.57 | 0.7486 | 0.993 | 0.995 | 28.62 | 0.7503 |
| | 20 | 2 | 17.22 | 0.996 | 1.935 | 29.66 | 0.7773 | 0.960 | 2.035 | 29.71 | 0.7827 |
| | 30 | 3 | 17.97 | 0.961 | 2.993 | 30.26 | 0.8033 | 0.928 | 3.067 | 30.26 | 0.8040 |
| | 60 | 6 | 18.86 | 1.027 | 6.002 | 30.78 | 0.8179 | 0.994 | 5.993 | 30.79 | 0.8182 |
| | 120 | 12 | 19.39 | 1.243 | 11.786 | 31.15 | 0.8237 | 1.143 | 12.014 | 31.16 | 0.8244 |

Table 5.2: Estimated parameter values $\hat{\alpha}$ and $\hat{\sigma}$ for the Lena and Piecewise images. The proposed algorithm and the algorithm presented in [64] are compared for various peak intensity values, $\sigma = \text{peak}/10$ and a constant $\alpha = 1$. In addition, the denoising performance (GAT (2.18) + BM3D + exact unbiased inverse $\mathcal{I}_\sigma$ of the GAT) associated with these parameter estimates is compared in terms of both PSNR and SSIM [137].

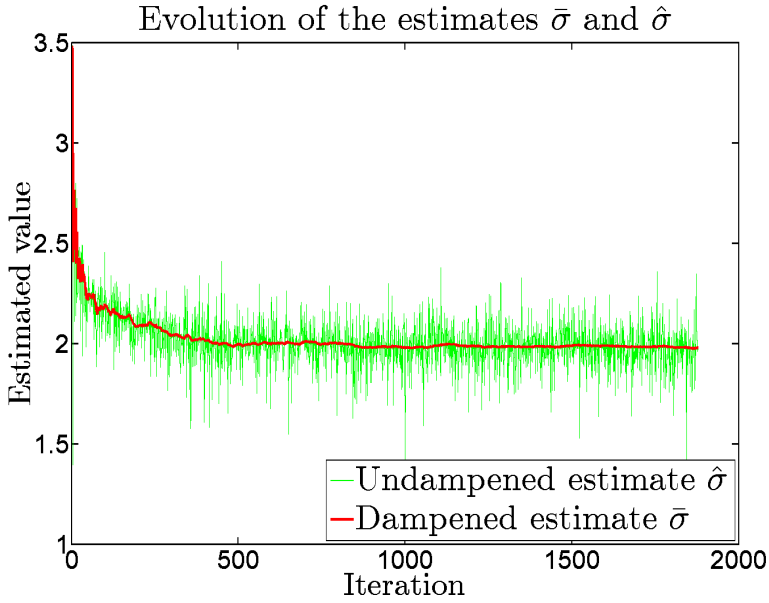
| Image | Peak | $\sigma$ | Noisy | Proposed algorithm | | | | [64] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PSNR | $\hat{\alpha}$ | $\hat{\sigma}$ | PSNR | SSIM | $\hat{\alpha}$ | $\hat{\sigma}$ | PSNR | SSIM |
| Lena (512×512) | 1 | 0.1 | 2.90 | 0.230 | 0.437 | 18.02 | 0.3099 | 1.003 | 0.218 | 22.60 | 0.6303 |
| | 2 | 0.2 | 5.84 | 0.472 | 0.555 | 22.61 | 0.4852 | 1.019 | 0.200 | 24.35 | 0.6697 |
| | 5 | 0.5 | 9.55 | 0.700 | 0.824 | 26.01 | 0.6672 | 1.074 | 0.285 | 26.09 | 0.7054 |
| | 10 | 1 | 12.18 | 0.671 | 1.496 | 27.63 | 0.7355 | 0.992 | 1.086 | 27.78 | 0.7571 |
| | 20 | 2 | 14.52 | 0.715 | 2.549 | 29.05 | 0.7906 | 1.091 | 1.884 | 29.06 | 0.7976 |
| | 30 | 3 | 15.71 | 0.780 | 3.450 | 29.78 | 0.8093 | 0.994 | 3.100 | 29.81 | 0.8136 |
| | 60 | 6 | 17.37 | 0.841 | 6.390 | 30.55 | 0.8297 | 1.068 | 5.861 | 30.55 | 0.8308 |
| | 120 | 12 | 18.49 | 1.007 | 12.041 | 31.04 | 0.8401 | 0.792 | 12.490 | 31.02 | 0.8393 |
| Piecewise (512×512) | 1 | 0.1 | 2.65 | 0.578 | 0.327 | 22.78 | 0.5935 | 0.920 | 0.086 | 24.30 | 0.8029 |
| | 2 | 0.2 | 5.61 | 0.624 | 0.462 | 24.97 | 0.6058 | 0.975 | 0.129 | 26.16 | 0.7705 |
| | 5 | 0.5 | 9.32 | 0.905 | 0.596 | 29.09 | 0.8109 | 1.011 | 0.435 | 29.14 | 0.8293 |
| | 10 | 1 | 11.98 | 0.926 | 1.066 | 31.10 | 0.8735 | 1.038 | 0.927 | 31.08 | 0.8861 |
| | 20 | 2 | 14.36 | 0.944 | 2.136 | 32.79 | 0.9172 | 0.999 | 2.052 | 32.76 | 0.9188 |
| | 30 | 3 | 15.56 | 0.945 | 3.088 | 33.60 | 0.9274 | 1.042 | 2.864 | 33.63 | 0.9270 |
| | 60 | 6 | 17.26 | 0.945 | 6.178 | 34.83 | 0.9479 | 0.975 | 6.081 | 34.84 | 0.9478 |
| | 120 | 12 | 18.42 | 0.989 | 12.044 | 35.63 | 0.9552 | 1.172 | 11.734 | 35.62 | 0.9561 |

Table 5.3: A comparison of the estimated parameter values $\hat{\alpha}$ and $\hat{\sigma}$ for the proposed algorithm and the algorithm presented in [64], for various peak intensity values, $\sigma = \text{peak}/10$ and various values of $\alpha$. In addition, the denoising performance (GAT (2.18) + BM3D + exact unbiased inverse $\mathcal{I}_\sigma$ of the GAT) associated with these parameter estimates is compared in terms of both PSNR and SSIM [137].

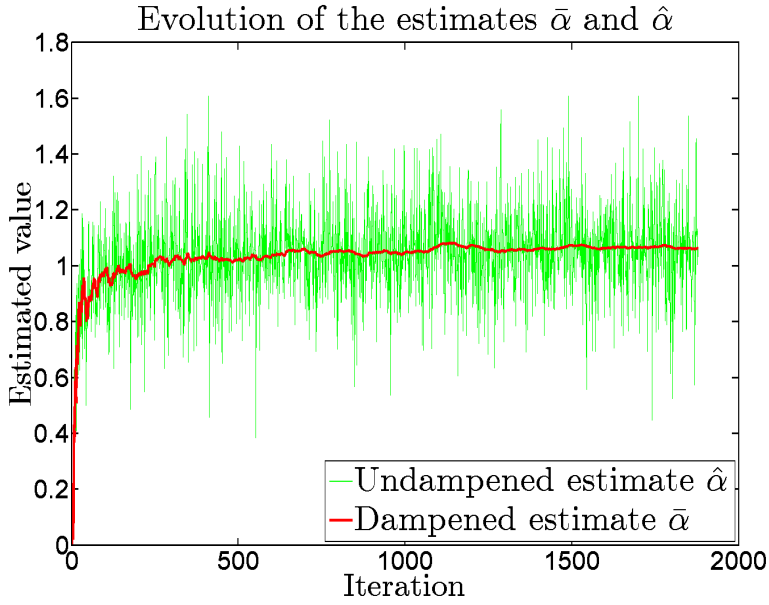
| Image | Peak | $\alpha$ | $\sigma$ | Noisy PSNR | Proposed algorithm | | | | [64] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | $\hat{\alpha}$ | $\hat{\sigma}$ | PSNR | SSIM | $\hat{\alpha}$ | $\hat{\sigma}$ | PSNR | SSIM |
| Cameraman (256×256) | 1 | 0.5 | 0.1 | 2.94 | 0.412 | 0.151 | 20.14 | 0.5643 | 0.479 | 0.063 | 19.58 | 0.5384 |
| | 2 | 0.5 | 0.2 | 5.60 | 0.373 | 0.250 | 21.39 | 0.5294 | 0.471 | 0.199 | 21.44 | 0.5855 |
| | 5 | 0.5 | 0.5 | 8.72 | 0.457 | 0.507 | 23.38 | 0.6485 | 0.522 | 0.504 | 23.36 | 0.6760 |
| | 10 | 0.5 | 1 | 10.60 | 0.513 | 0.983 | 24.36 | 0.7046 | 0.555 | 0.997 | 24.33 | 0.7139 |
| | 20 | 0.5 | 2 | 11.96 | 0.476 | 2.066 | 25.25 | 0.7427 | 0.556 | 1.993 | 25.22 | 0.7421 |
| | 30 | 0.5 | 3 | 12.53 | 0.484 | 3.030 | 25.39 | 0.7484 | 0.509 | 3.029 | 25.39 | 0.7498 |
| | 60 | 0.5 | 6 | 13.19 | 0.535 | 6.031 | 25.80 | 0.7583 | 0.624 | 6.073 | 25.79 | 0.7617 |
| | 120 | 0.5 | 12 | 13.59 | 0.545 | 12.191 | 26.02 | 0.7704 | 0.230 | 12.360 | 26.04 | 0.7681 |
| Fluorescent Cells (512×512) | 1 | 1.5 | 0.1 | 7.37 | 0.733 | 0.206 | 23.09 | 0.5818 | 1.486 | 0.337 | 23.34 | 0.5259 |
| | 2 | 1.5 | 0.2 | 10.27 | 1.534 | 0.335 | 25.94 | 0.6476 | 1.358 | 0.074 | 25.64 | 0.6236 |
| | 5 | 1.5 | 0.5 | 13.96 | 1.263 | 0.610 | 27.76 | 0.7298 | 1.462 | 0.458 | 27.82 | 0.7256 |
| | 10 | 1.5 | 1 | 16.48 | 1.331 | 1.046 | 29.04 | 0.7721 | 1.491 | 0.987 | 29.09 | 0.7731 |
| | 20 | 1.5 | 2 | 18.72 | 1.457 | 1.948 | 30.29 | 0.8042 | 1.486 | 2.045 | 30.32 | 0.8057 |
| | 30 | 1.5 | 3 | 19.81 | 1.506 | 2.890 | 31.00 | 0.8262 | 1.466 | 3.100 | 31.03 | 0.8290 |
| | 60 | 1.5 | 6 | 21.28 | 1.506 | 5.935 | 31.89 | 0.8517 | 1.568 | 6.092 | 31.88 | 0.8510 |
| | 120 | 1.5 | 12 | 22.26 | 1.684 | 11.796 | 32.54 | 0.8649 | 1.633 | 11.940 | 32.53 | 0.8650 |
| Piecewise (512×512) | 1 | 2.5 | 0.1 | 2.71 | 1.025 | 0.881 | 19.74 | 0.4649 | 2.436 | 0.499 | 25.03 | 0.8083 |
| | 2 | 2.5 | 0.2 | 5.74 | 1.541 | 1.084 | 25.04 | 0.6195 | 2.411 | 0.195 | 26.55 | 0.7956 |
| | 5 | 2.5 | 0.5 | 9.65 | 2.233 | 0.999 | 29.40 | 0.8212 | 2.470 | 0.327 | 29.46 | 0.8441 |
| | 10 | 2.5 | 1 | 12.60 | 2.310 | 1.471 | 31.53 | 0.8882 | 2.526 | 0.801 | 31.56 | 0.9001 |
| | 20 | 2.5 | 2 | 15.49 | 2.026 | 3.454 | 33.46 | 0.9161 | 2.529 | 2.023 | 33.60 | 0.9332 |
| | 30 | 2.5 | 3 | 17.12 | 2.254 | 4.006 | 34.79 | 0.9418 | 2.553 | 2.793 | 34.83 | 0.9470 |
| | 60 | 2.5 | 6 | 19.80 | 2.338 | 6.793 | 36.77 | 0.9640 | 2.510 | 6.094 | 36.82 | 0.9654 |
| | 120 | 2.5 | 12 | 22.19 | 1.945 | 14.719 | 38.52 | 0.9734 | 2.719 | 11.577 | 38.63 | 0.9768 |

eters, where the denoising was performed via variance stabilization (GAT (2.18)), a state-of-the art Gaussian denoising algorithm BM3D, and the exact unbiased inverse $\mathcal{I}_\sigma$ of the GAT.

At best, the performance of the proposed algorithm is comparable to that achieved with the estimator presented in [64]. In some low-intensity cases it produces inferior estimation results, but we have demonstrated our novel approach to have definite potential. In particular, we presume these inferior results to be a consequence of inaccurate variance stabilization in low-intensity regions. Thus, it could be beneficial to construct an optimized variance-stabilizing transformation as presented in [66], in order to replace the more inaccurate GAT. On the other hand, optimizing the various heuristics present in the algorithm is another source for possible improvements in performance. For instance, the choice of the aggregation and regularization parameters in the dampening stage ((5.13)–(5.16)) appears to play a significant role in the success of the algorithm.

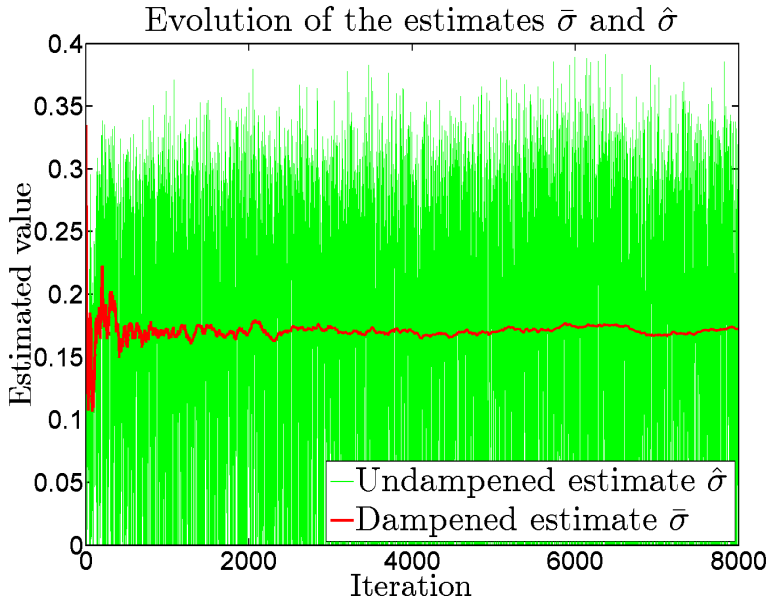



(a)

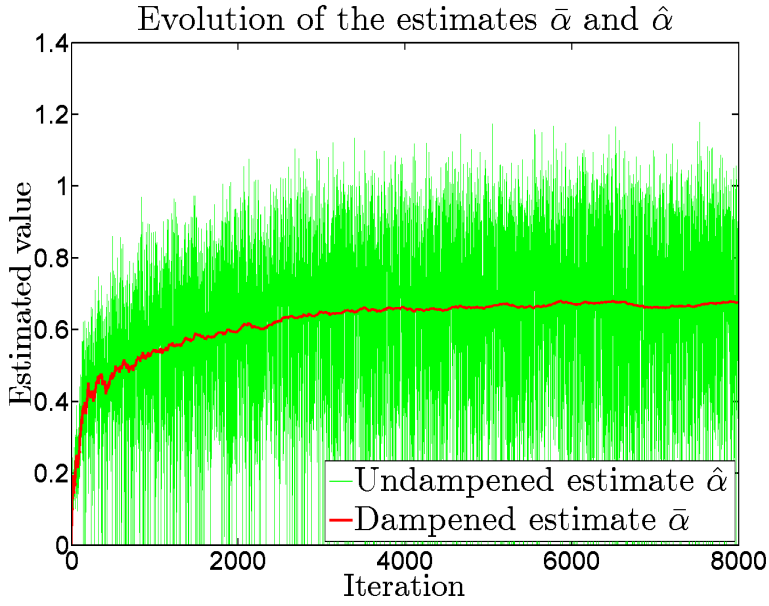



(b)

Figure 5.4: Evolution of the undampened and dampened parameter estimates for Cameraman ($256 \times 256$, peak $= 20$, $\sigma = 2$, $\alpha = 1$). (a) Undampened estimate $\hat{\sigma}$ and dampened estimate $\bar{\sigma}$, converging to $\bar{\sigma} = 1.978$. (b) Undampened estimate $\hat{\alpha}$ and dampened estimate $\bar{\alpha}$, converging to $\bar{\alpha} = 1.063$.

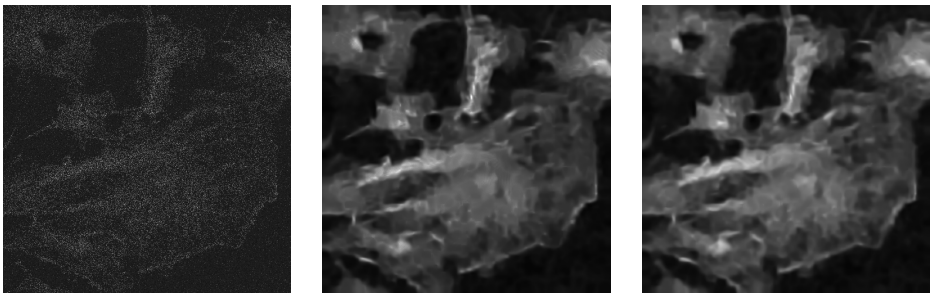



(a)




(b)

Figure 5.5: Evolution of the undampened and dampened parameter estimates for Fluorescent Cells ($512 \times 512$, peak $= 1$, $\sigma = 0.1$, $\alpha = 1$). (a) Undampened estimate $\hat{\sigma}$ and dampened estimate $\bar{\sigma}$, not converging within 8000 iterations, but producing $\bar{\sigma} = 0.172$. (b) Undampened estimate $\hat{\alpha}$ and dampened estimate $\bar{\alpha}$, not converging within 8000 iterations, but producing $\bar{\alpha} = 0.676$.



(a) (b) (c)

Figure 5.6: The denoising of Fluorescent Cells ($512 \times 512$, peak = 1, $\sigma = 0.1$, $\alpha = 1$). (a) Noisy image (gamma-corrected with $\gamma = 0.9$ for improved visibility). (b) Denoised image, obtained with the parameter estimates ($\bar{\sigma} = 0.172$, $\bar{\alpha} = 0.676$) produced by the proposed method (PSNR = 24.04 dB, SSIM = 0.5751). (b) Denoised image, obtained with the parameter estimates ($\hat{\sigma} = 0.183, \hat{\alpha} = 0.949$) produced by [64] (PSNR = 24.19 dB, SSIM = 0.5655).

(a) (b) (c)

Figure 5.7: The denoising of Piecewise ($512 \times 512$, peak = 1, $\sigma = 0.1$, $\alpha = 2.5$). (a) Noisy image (gamma-corrected with $\gamma = 0.9$ for improved visibility). (b) Denoised image, obtained with the parameter estimates ($\bar{\sigma} = 0.881$, $\bar{\alpha} = 1.025$) produced by the proposed method (PSNR = 19.74 dB, SSIM = 0.4649). (b) Denoised image, obtained with the parameter estimates ($\hat{\sigma} = 0.499, \hat{\alpha} = 2.436$) produced by [64] (PSNR = 25.03 dB, SSIM = 0.8083).

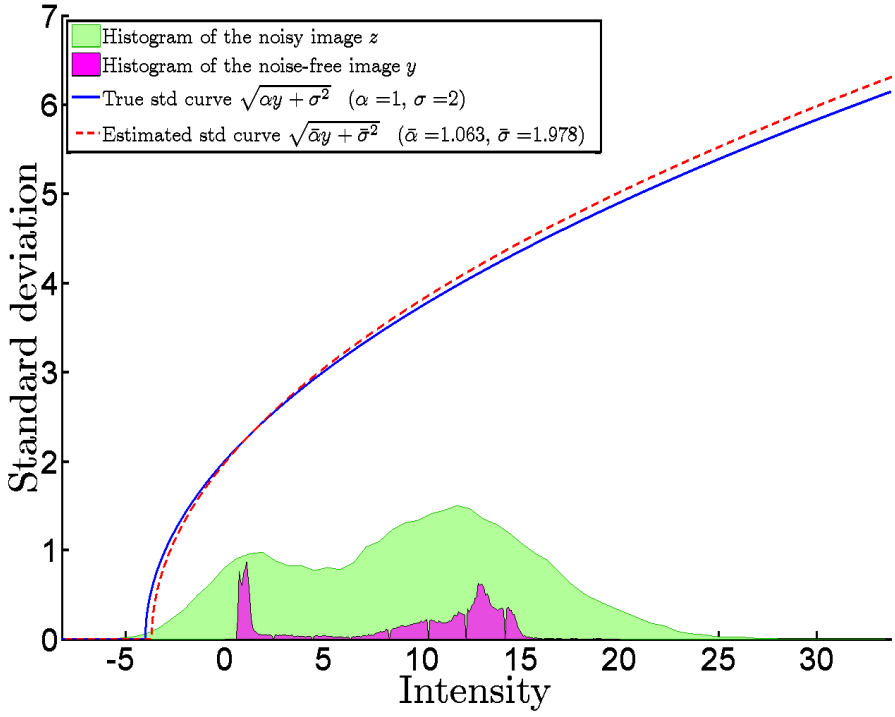



Figure 5.8: Estimated and true standard deviation curves for the Cameraman image ($256 \times 256$, peak $= 20$, $\sigma = 2$, $\alpha = 1$), overlaid with the histograms of the noisy and noise-free images. By "Intensity", we denote $z$ for the noisy image, $y$ for the noise-free image, and $\alpha y$ or $\bar{\alpha} y$ for the standard deviation curves.

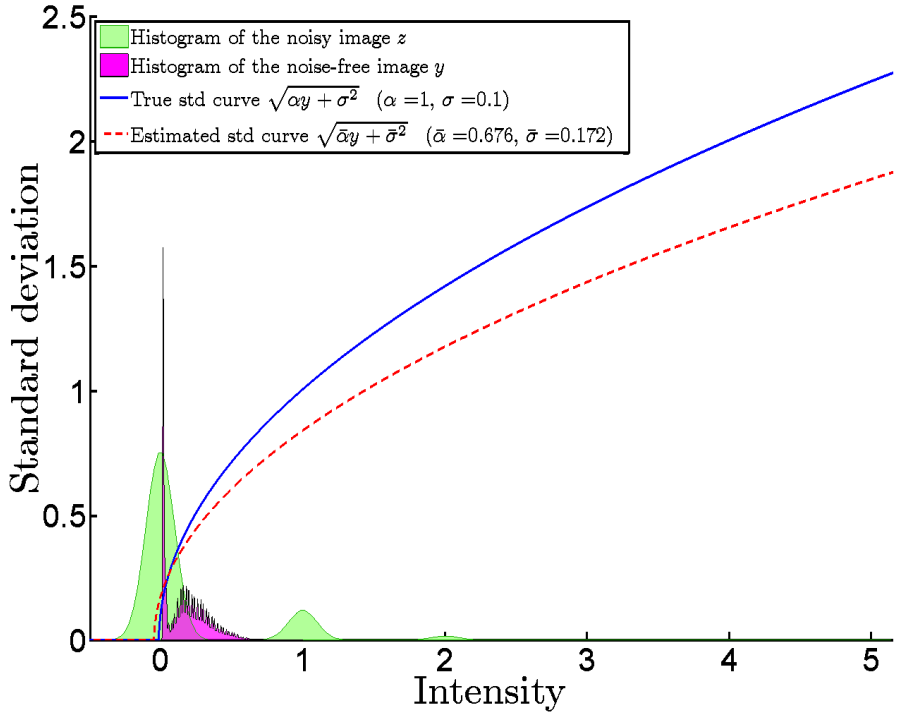



Figure 5.9: Estimated and true standard deviation curves for the Fluorescent Cells image ($512 \times 512$, peak $= 1$, $\sigma = 0.1$, $\alpha = 1$), overlaid with the histograms of the noisy and noise-free images. By "Intensity", we denote $z$ for the noisy image, $y$ for the noise-free image, and $\alpha y$ or $\bar{\alpha} y$ for the standard deviation curves.

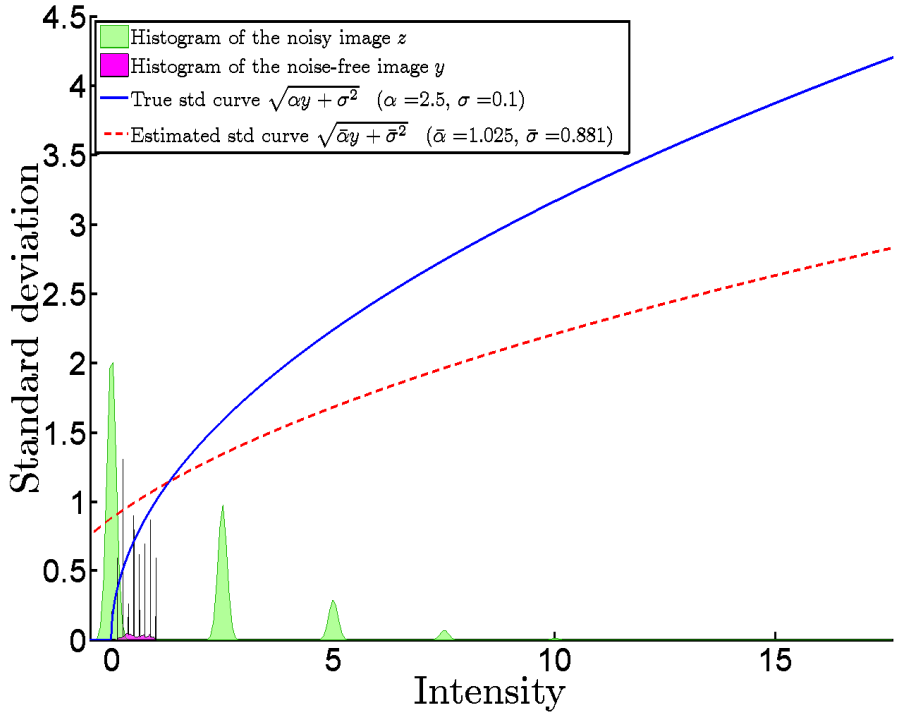



Figure 5.10: Estimated and true standard deviation curves for the Piecewise image (512× 512, peak = 1, $\sigma = 0.1$, $\alpha = 2.5$), overlaid with the histograms of the noisy and noise-free images. By "Intensity", we denote $z$ for the noisy image, $y$ for the noise-free image, and $\alpha y$ or $\bar{\alpha} y$ for the standard deviation curves.

# Chapter 6

# Conclusions to the thesis

In this thesis, we contemplated Poisson and Poisson-Gaussian image denoising, using the Anscombe variance-stabilizing transformation and its Poisson-Gaussian generalization. In particular, we demonstrated the importance of a properly designed inverse transformation, and constructed exact unbiased inverses for these two VSTs.

First, we proposed an exact unbiased inverse for the Anscombe transformation for Poisson noise in Chapter 3. We showed that the poor denoising performance associated with the traditional algebraic and asymptotically unbiased inverses can be overcome by using the proposed inverse instead. Specifically, it provides significant improvements in the denoising of low-count data. We justified the good performance of the exact unbiased inverse by showing that it coincides with a maximum likelihood inverse under rather generic hypotheses.

For a convenient use of the exact unbiased inverse in practical applications, we also provided a closed-form approximation for it, and showed that the denoising results obtained with these two inverses are practically equal. Thus, the traditional algebraic and asymptotically unbiased inverses can easily be replaced by this closed-form expression, providing the excellent performance of the exact unbiased inverse.

By combining the proposed inverse with a state-of-the-art Gaussian denoising algorithm, we achieve results that are competitive with state-of-the-art Poisson noise removal algorithms. While proper inversion is more crucial than the choice of the Gaussian denoising algorithm, the latter bears some significance as well. Out of the algorithms considered here, BM3D seems to be the best choice due to its overall strong results combined with relatively low complexity.

In order to generalize the above results to the case of Poisson-Gaussian noise, we introduced the exact unbiased inverse of the generalized Anscombe transformation in Chapter 4, and demonstrated the excellent denoising performance achieved by combining it with a state-of-the-art Gaussian denoising algorithm. As in the Poisson case, we showed that the proposed inverse can be interpreted as a maximum likelihood inverse. Moreover, we presented a thorough analysis of the asymptotic properties and global accuracy of the inverse. In particular, we showed that it can be approximated very accurately by adding a simple correction term $-\sigma^2$ to

the exact unbiased inverse of the Anscombe transformation. As a byproduct of this analysis, we constructed a closed-form approximation for it by adding this correction term to the closed-form approximation of the exact unbiased inverse of the Anscombe transformation.

Finally, we investigated the effect of inaccurate parameter estimation on variance stabilization for a general noise distribution in Chapter 5. Based on the theoretical results, we constructed a novel variance-stabilization based iterative algorithm for estimating the scaling factor $\alpha$ and the Gaussian standard deviation $\sigma$ of the Poisson-Gaussian noise model. We compared its estimation and denoising performance against that of an estimator based on segmentation, local estimates and regression [64]. At best, our algorithm provided equally good results, thus showing its underlying potential. However, its inferior performance in some low-intensity cases may be caused by inadequate stabilization of the variance, so employing an optimized VST instead of the GAT could be beneficial. Additional improvements may be gained by a careful optimization of the heuristic parts of the estimation algorithm.

In light of our contributions, it seems questionable whether it is advantageous to go through the effort of designing separate denoising algorithms for each specific noise distribution, when comparable gains can be attained by perfecting the denoising algorithms in the AWGN case and addressing various noise distributions through variance stabilization together with a suitable inverse.

Note that even though we only considered the Anscombe transformation and its Poisson-Gaussian generalization, there exist a variety of other variance-stabilizing transformations. In particular, combining the optimized VSTs discussed in [66] and [65] with the proposed framework of exact unbiased inversion would be a good subject for future research. Moreover, even though the thesis deals exclusively with 2-D data (digital images), the concepts and procedures presented here can be applied to data of any dimension, including 1-D signals and volumetric data.

In addition, it is worth mentioning that even though we focused solely on grayscale images, the use of the proposed inverses can be easily extended to colour image denoising. In order to capture a colour image, the light typically passes through a colour filter array (CFA), such as a Bayer filter [10], before reaching the actual image sensor. The Bayer filter is an array composed of $2 \times 2$ blocks, where each block contains two green, one red, and one blue filter. Thus, each pixel in the raw sensor data actually represents the intensity of only red, blue or green colour, and the final RGB colour image is obtained by interpolating the missing samples with a *demosaicking* algorithm. The most simple option of denoising colour images with the proposed framework is to process each colour channel as a separate intensity image. In a more sophisticated manner, one can also combine variance stabilization with, for instance, cross-colour BM3D filtering [42] (for denoising raw CFA data) or C-BM3D [36] (for denoising RGB colour images).

Another possible application of the proposed framework could be to take advantage of it in solving linear inverse problems of the form $z \sim \mathcal{P}(Hy)$, where $H$ is, e.g., a blurring kernel or a projection operator; these inverse problems may be approached through applying a Poisson denoising algorithm. However, adapting the proposed framework to such a problem should be done with care, keeping in

mind that variance stabilization is a nonlinear process. Specifically, we cannot generally assume that after stabilizing $z$, the data can be treated as if it were linearly degraded and corrupted by AWGN. Instead, the nonlinearity of the VST and its inverse should be taken into account, as is done for instance in [51], where the authors consider the Anscombe transformation and its algebraic and asymptotically unbiased inverses.

It is also important to note that the concept of optimal inverse transformations is not restricted to the Poisson and Poisson-Gaussian distributions; the same principles can be used for other distributions just as well. For instance, the stabilization and the exact unbiased inverse transformation for the raw data from digital imaging sensors is studied in [67]; these data are modelled by a doubly censored heteroskedastic normal distribution. Another example is [109], where we denoise single-look synthetic-aperture radar (SAR) images by applying the forward and exact unbiased inverse transformations designed for Rayleigh noise. Finally, similar considerations for the denoising of magnetic resonance images are presented in [68], where the noise distribution is assumed to be Rician.

# Bibliography


[1] Aghagolzadeh, S., and O. K. Ersoy, "Transform image enhancement", *Optical Engineering*, vol. 31, no. 3, pp. 614–626, March 1992.

[2] Andrews, D., and C. Mallows, "Scale mixtures of normal distributions", *Journal of the Royal Statistical Society, Series B*, vol. 36, no. 1, pp. 99–102, 1974.

[3] Anscombe, F. J., "The transformation of Poisson, binomial and negative-binomial data", *Biometrika*, vol. 35, no. 3/4, pp. 246–254, Dec. 1948. http://dx.doi.org/10.1093/biomet/35.3-4.246

[4] Anscombe, F. J., "Contribution to discussion on paper by H. Hotelling", *Journal of the Royal Statistical Society, Series B*, vol. 15, pp. 229–230, 1953.

[5] Antoine, J. P., P. Vandergheynst, and R. Murenzi, "Two-dimensional directional wavelets in image processing", *International Journal of Imaging Systems and Technology*, vol. 7, no. 3, pp. 152–165, 1996.

[6] Antoniadis, A., J. Bigot, and T. Sapatinas, "Wavelet estimators in nonparametric regression: A comparative simulation study", *Journal of Statistical Software*, vol. 6, no. 6, pp. 1–83, 2001.

[7] Bar-Lev, S. K., and P. Enis, "On the classical choice of variance stabilizing transformations and an application for a Poisson variate", *Biometrika*, vol. 75, no. 4, pp. 803–804, 1988.

[8] Bar-Lev, S. K., and P. Enis, "Variance stabilizing and normalizing transformations for the compound Poisson process", *Metrika*, vol. 39, no. 1, pp. 165–175, 1992.

[9] Bartlett, M. S., "The Square Root Transformation in Analysis of Variance", *Supplement to the Journal of the Royal Statistical Society*, vol. 3, no. 1, pp. 68–78, 1936.

[10] Bayer, B. E., "Color imaging array", *United States Patent 3971065*, July 1976.

[11] Beall, G., "The transformation of data from entomological field experiments so that the analysis of variance becomes applicable", *Biometrika*, vol. 32, pp. 243–262, 1942.

[12] Besbeas, P., I. De Feis, and T. Sapatinas, "A Comparative Simulation Study of Wavelet Shrinkage Estimators for Poisson Counts", *International Statistical Review*, vol. 72, no. 2, pp. 209–237, Aug. 2004.

[13] Bliss, C. I., "The transformation of percentages for use in the analysis of variance", *Ohio Journal of Science*, vol. 38, pp. 9–12, 1938.

[14] Blu, T., and F. Luisier, "The SURE-LET Approach to Image Denoising", *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778–2786, Nov. 2007.

[15] Boulanger, J., J. B. Sibarita, C. Kervrann, and P. Bouthemy, "Non-parametric regression for patch-based fluorescence microscopy image sequence denoising", *Proc. IEEE Int. Symp. on Biomedical Imaging (ISBI'08)*, pp. 748–751, May 2008.

[16] Box, G. E. P., and D. R. Cox, "An Analysis of Transformations", *Journal of the Royal Statistical Society, Series B*, vol. 26, no. 2, pp. 211–252, 1964.

[17] Boyle, W. S., and G. E. Smith, "Charge coupled semiconductor devices", *Bell Syst. Tech. J.*, vol. 49, no. 4, pp. 587–593, Apr. 1970.

[18] Breiman, L, and J. H. Friedman, "Estimating Optimal Transformations for Multiple Regression and Correlation", *Journal of the American Statistical Association*, vol. 80, no. 391, pp. 580–598, 1985.

[19] Buades, A., B. Coll, and J. M. Morel, "A Review of Image Denoising Algorithms, with a New One", *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, July 2005.

[20] Buades, A., Y. Lou, J. M. Morel, and Z. Tang, "Multi image noise estimation and denoising", preprint, Aug. 2010.

[21] Candès, E. J., "Harmonic analysis of neural networks", *Applied and Computational Harmonic Analysis*, vol. 6, no. 2, pp. 197–218, March 1999.

[22] Candès, E. J., and D. L. Donoho, "Curvelets: A Surprisingly Effective Nonadaptive Representation for Objects with Edges", *Proc. 4th Int. Conf. on Curves and Surfaces*, vol. 2, pp. 105–120, Saint-Malo, France, July 1999.

[23] Chamberlain, S. G., "Photosensitivity and Scanning of Silicon Image Detector Arrays", *IEEE Journal of Solid-State Circuits*, vol. 4, no. 6, pp. 333–342, Dec. 1969.

[24] Chatterjee, P., and P. Milanfar, "Clustering-Based Denoising With Locally Learned Dictionaries", *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438–1451, July 2009.

[25] Chatterjee, P., and P. Milanfar, "Is Denoising Dead?", *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 895–911, Apr. 2010.

[26] Chatterjee, P., and P. Milanfar, "Patch-Based Near-Optimal Image Denoising", *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1635–1649, Apr. 2012.

[27] Clark, A., and W. H. Leonard, "The analysis of variance with special reference to data expressed as percentages", *Journal of the American Society of Agronomy*, vol. 31, pp. 55–56, 1939.

[28] Cleveland, W. S., "Robust Locally Weighted Regression and Smoothing Scatterplots", *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829–836, 1979.

[29] Cleveland, W. S., and S. J. Devlin, "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting", *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[30] Cochran, W. G., "Some difficulties in the statistical analysis of replicated experiments", *Empire Journal of Experimental Agriculture*, vol. 6, pp. 157–175, 1938.

[31] Cochran, W. G., "The analysis of variance when experimental errors follow the Poisson or binomial laws", *The Annals of Mathematical Statistics*, vol. 9, pp. 335–347, 1940.

[32] Crouse, M., R. Nowak, and G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models", *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 886–902, Apr. 1998.

[33] Curtiss, J. H., "On transformations used in the analysis of variance", *The Annals of Mathematical Statistics*, vol. 14, no. 2, pp. 107–122, June 1943.

[34] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising with Block-Matching and 3D Filtering", *Proc. SPIE Electronic Imaging: Algorithms and Systems V*, no. 6064A-30, San Jose, USA, Jan. 2006. http://www.cs.tut.fi/~foi/3D-DFT

[35] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering", *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[36] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space", *Proc. IEEE Int. Conf. Image Process.*, San Antonio, TX, USA, Sept. 2007.

[37] Dabov, K., A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering", *Proc. 15th European Signal Processing Conference, EUSIPCO 2007*, Poznan, Poland, Sept. 2007.

[38] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Joint image sharpening and denoising by 3D transform-domain collaborative filtering", *Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2007*, Moscow, Russia, Sept. 2007.

[39] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "A nonlocal and shape-adaptive transform domain collaborative filtering", *Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2008*, Lausanne, Switzerland, Aug. 2008.

[40] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D Image Denoising with Shape-Adaptive Principal Component Analysis", *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations, SPARS'09*, Saint-Malo, France, Apr. 2009.

[41] Dabov, K., "Image and Video Restoration with Nonlocal Transform-Domain Filtering", *Thesis for the degree of Doctor of Science in Technology*, Tampere University of Technology, Tampere, Finland, 2010.

[42] Danielyan, A., M. Vehviläinen, A. Foi, V. Katkovnik, and K. Egiazarian, "Cross-color BM3D filtering of noisy raw data", *Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2009*, Tuusula, Finland, pp. 125–129, Aug. 2009.

[43] Danielyan, A., 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, Aug. 2009.

[44] Danielyan, A., V. Katkovnik, and K. Egiazarian, "BM3D Frames and Variational Image Deblurring", *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, Apr. 2012.

[45] Daubechies, I., "Orthonormal bases of compactly supported wavelets", *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, Oct. 1988.

[46] De Bonet, J. S., "Noise Reduction Through Detection of Signal Redundancy", *Rethinking Artificial Intelligence*, MIT AI Lab, 1997.

[47] Delpretti, S., F. Luisier, S. Ramani, T. Blu, and M. Unser, "Multiframe SURE-LET denoising of timelapse fluorescence microscopy images", *Proc. IEEE Int. Symp. on Biomedical Imaging, ISBI 2008*, Paris, France, pp. 149–152, May 2008.

[48] Do, M. N., "The contourlet transform: an efficient directional multiresolution image representation", *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[49] Donoho, D. L., and J. Johnstone, "Ideal spatial adaptation via wavelet shrinkage", *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[50] Donoho, D. L., "Wedgelets: Nearly Minimax Estimation of Edges", *The Annals of Statistics*, vol. 27, no. 3, pp. 859–897, June 1999.

[51] Dupé, F-X., J. M. Fadili, and J-L. Starck, "A Proximal Iteration for Deconvolving Poisson Noisy Images Using Sparse Representations", *IEEE Trans. Image Process.*, vol. 18, no. 2, pp. 310–321, Feb. 2009.

[52] Efron, B., "Transformation theory: How normal is a family of distributions?", *The Annals of Statistics*, vol. 10, no. 2, pp. 323–339, 1982.

[53] Efros, A., and T. Leung, "Texture synthesis by non parametric sampling", *Proc. IEEE International Conference on Computer Vision*, vol. 2, Corfu, Greece, pp. 1033–1038, 1999.

[54] Einstein, A., "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt (On a Heuristic Viewpoint Concerning the Production and Transformation of Light)", *Annalen der Physik*, vol. 322, no. 6, pp. 132–148, June 1905.

[55] Elad, M., and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries", *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[56] Enz, C. C., and G. C. Temes, "Circuit Techniques for Reducing the Effects of Op-Amp Imperfections: Autozeroing, Correlated Double Sampling, and Chopper Stabilization", *Proc. IEEE*, vol. 84, no. 11, pp. 1584–1614, Nov. 1996.

[57] Fan, J., and I. Gijbels, "Data-Driven Bandwidth Selection in Local Polynomial Fitting: Variable Bandwidth and Spatial Adaptation", *Journal of the Royal Statistical Society, Series B*, vol. 57, no. 2, pp. 371–394, 1995.

[58] Fan, J., and I. Gijbels, *Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66*, Chapman & Hall/CRC, 1996.

[59] Figueiredo, M. A. T., and R. D. Nowak, "Wavelet-Based Image Estimation: An Empirical Bayes Approach Using Jeffrey's Noninformative Prior", *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1322–1331, Sept. 2001.

[60] Fisher, R. A., "The Correlation between Relatives on the Supposition of Mendelian Inheritance", *Transactions of the Royal Society of Edinburgh*, vol. 52, pp. 399–433, 1918.

[61] Fisz, M., "The Limiting Distribution of a Function of Two Independent Random Variables and its Statistical Application", *Colloquium Matmematicum*, vol. 3, pp. 138–146, 1955.

[62] Foi, A., K. Dabov, V. Katkovnik, and K. Egiazarian, "Shape-Adaptive DCT for Denoising and Image Reconstruction", *Proc. SPIE Electronic Imaging: Algorithms and Systems V*, no. 6064A-18, San Jose, USA, Jan. 2006.

[63] Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images", *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, May 2007.

[64] Foi, A., 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, Oct. 2008. doi:10.1109/TIP.2008.2001399

[65] Foi, A., "Direct optimization of nonparametric variance-stabilizing transformations", presented at *8èmes Rencontres de Statistiques Mathématiques, CIRM*, Luminy, Dec. 2008.

[66] Foi, A., "Optimization of variance-stabilizing transformations", 2009, preprint available at http://www.cs.tut.fi/~foi/.

[67] Foi, A., "Clipped noisy images: heteroskedastic modeling and practical denoising", *Signal Processing*, vol. 89, no. 12, pp. 2609–2629, Dec. 2009. doi:10.1016/j.sigpro.2009.04.035

[68] Foi, A., "Noise Estimation and Removal in MR imaging: the Variance-Stabilization Approach", *Proc. 2011 IEEE International Symposium on Biomedical Imaging, ISBI 2011*, pp. 1809–1814, Chicago, USA, Apr. 2011.

[69] Freeman, M., and J. Tukey, "Transformations related to the angular and the square root", *The Annals of Mathematical Statistics*, vol. 21, no. 4, pp. 607–611, 1950.

[70] Fryzlewicz, P., and G. P. Nason, "A Haar-Fisz Algorithm for Poisson Intensity Estimation", *Journal of Computational and Graphical Statistics*, vol. 13, no. 3, pp. 621–638, 2004.

[71] Fryzlewicz, P., and V. Delouille, "A Data-Driven Haar-Fisz Transformfor Multiscale Variance Stabilization", *IEEE/SP 13th Workshop on Statistical Signal Processing*, pp. 539–544, July 2005.

[72] Gravel, P., G. Beaudoin, and J. A. De Guise, "A Method for Modeling Noise in Medical Images", *IEEE Trans. Medical Imaging*, vol. 23, no. 10, pp. 1221–1232, Oct. 2004.

[73] Haar, A., "Zur Theorie der orthogonalen Funktionensysteme, Erste Mitteilung", *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910.

[74] Healey, G. E., and R. Kondepudy, "Radiometric CCD Camera Calibration and Noise Estimation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 267–276, March 1994.

[75] Hertz, H., "Ueber einen Einfluss des ultravioletten Lichtes auf die electrische Entladung", *Annalen der Physik*, vol. 267, no. 8, pp. 983–1000, 1887.

[76] Howell, S. B., *Handbook of CCD Astronomy*, 2nd ed., Cambridge University Press, Cambridge, 2006.

[77] Hyvärinen, A., J. Hurri, and P. O. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, Springer-Verlag, London, June 2009.

[78] Jezierska, A., C. Chaux, J-C. Pesquet, and H. Talbot, "An EM approach for Poisson-Gaussian noise modeling", *19th European Signal Processing Conference, EUSIPCO 2011*, Barcelona, Spain, Sept. 2011.

[79] Johnson, J. B., "Thermal Agitation of Electricity in Conductors", *Physical Review*, vol. 32, no. 1, pp. 97–109, July 1928.

[80] Jones, M. C., J. S. Marron, and S. J. Sheather, "Progress in data based bandwidth selection for kernel density estimation", *Mimeo Series 2088*, Department of Statistics, University of North Carolina, Chapel Hill, 1992.

[81] Katkovnik, V., K. Egiazarian, and J. Astola, *Local Approximation Techniques in Signal Processing*, SPIE Press, Washington, 2006.

[82] Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising", *Int. J. Computer Vision*, vol. 86, no. 1, pp. 1–32, Jan. 2010.

[83] Kendall, M., *The advanced theory of statistics*, *Volume 1*, 2nd ed., Charles Griffin and co., 1945.

[84] Kervrann, C., and J. Boulanger, "Optimal Spatial Adaptation for Patch-Based Image Denoising", *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

[85] Kervrann, C., and J. Boulanger, "Local adaptivity to variable smoothness for exemplar-based image denoising and representation", *International Journal of Computer Vision*, vol. 79, no. 1, pp. 45–69, Aug. 2008.

[86] Kingsbury, N., "Image processing with complex wavelets", *Philosophical Transactions of the Royal Society, Series A*, vol. 357, no. 1760, pp. 2543–2560, Sept. 1999.

[87] Kingsbury, N., "Complex wavelets for shift invariant analysis and filtering of signals", *Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 234–253, May 2001.

[88] Koczyk, P., P. Wiewiór, and C. Radzewicz, "Photon counting statistics – Undergraduate experiment", *American Journal of Physics*, vol. 64, no. 3, pp. 240–245, March 1996.

[89] Koerner, S., E. Lehmann, and P. Vontobel, "Design and optimization of a CCD-neutron radiography detector", *Nuclear Instruments and Methods in Physics Research, Section A*, vol. 454, no. 1, pp. 158–164, Nov. 2000.

[90] Kolaczyk, E. D., "Bayesian multiscale models for Poisson processes", *Journal of the American Statistical Association*, vol. 94, no. 447, pp. 920–933, 1999.

[91] Kolaczyk, E. D., and D. D. Dixon, "Nonparametric estimation of intensity maps using Haar wavelets and Poisson noise characteristics", *The Astrophysical Journal*, vol. 534, no. 1, pp. 490–505, 2000.

[92] Labate, D., W-Q. Lim, G. Kutyniok, and G. Weiss, "Sparse Multidimensional Representation using Shearlets", *Proceedings of SPIE, Wavelets XI*, vol. 5914, pp. 254–262, 2005.

[93] Lee, P. M., *Bayesian Statistics: An Introduction*, 4th edition, John Wiley & Sons, 2012.

[94] Lefkimmiatis, S., P. Maragos, and G. Papandreou, "Bayesian inference on multiscale models for Poisson intensity estimation: Applications to photon-limited image denoising", *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1724–1741, Aug. 2009.

[95] Lu, H., X. Li, I-T. Hsiao, and Z. Liang, "Analytical Noise Treatment for Low-Dose CT Projection Data by Penalized Weighted Least-Square Smoothing in the K-L Domain", *Proc. SPIE, Medical Imaging 2002*, vol. 4682, pp. 146–152, 2002.

[96] Lu, H., Y. Kim, and J. M. M. Anderson, "Improved Poisson intensity estimation: Denoising application using Poisson data", *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1128–1135, Aug. 2004.

[97] Lu, N., *Fractal Imaging*, Academic Press, New York, 1997.

[98] Luisier, F., C. Vonesch, T. Blu, and M. Unser, "Fast interscale wavelet denoising of Poisson-corrupted images", *Signal Processing*, vol. 90, no. 2, pp. 415–427, Feb. 2010.

[99] Luisier, F., T. Blu, and M. Unser, "Image Denoising in Mixed Poisson-Gaussian Noise", *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 696–708, March 2011.

[100] Maggioni, M., G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4D nonlocal spatiotemporal transforms", *Proc. SPIE Electronic Imaging: Algorithms and Systems IX*, no. 7870-2, San Francisco, USA, Jan. 2011.

[101] Maggioni, M., G. Boracchi, A. Foi, and K. Egiazarian, "Video Denoising, Deblocking and Enhancement Through Separable 4-D Nonlocal Spatiotemporal Transforms", *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 3952–3966, Sept. 2012.

[102] Mairal, J., G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration", *Multiscale Modeling and Simulation*, vol. 7, no. 1, pp. 214–241, 2008.

[103] Mallat, S. G., "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[104] Mallat, S. G., "Multiresolution approximations and wavelet orthonormal bases of L2(R)", *Transactions of the American Mathematical Society*, vol. 315, no. 1, pp. 69–87, Sept. 1989.

[105] Mandel, L., "Fluctuations of Photon Beams: The Distribution of the Photo-Electrons ", *Proc. Phys. Soc.*, vol. 74, no. 3, pp. 233–243, 1959.

[106] Milanfar, P., "A Tour of Modern Image Filtering: New Insights and Methods, Both Practical and Theoretical", *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[107] Murtagh, F., J-L. Starck, and A. Bijaoui, "Image restoration with noise suppression using a multiresolution support", *Astronomy & Astrophysics Supplement Series*, vol. 112, pp. 179–189, July 1995.

[108] Mäkitalo, M., and A. Foi, "On the inversion of the Anscombe transformation in low-count Poisson image denoising", *Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2009*, Tuusula, Finland, pp. 26–32, Aug. 2009. http://dx.doi.org/10.1109/LNLA.2009.5278406

[109] Mäkitalo, M., A. Foi, D. Fevralev, and V. Lukin, "Denoising of single-look SAR images based on variance stabilization and nonlocal filters", *Proc. Int. Conf. Math. Meth. Electromagn. Th., MMET 2010*, Kiev, Ukraine, Sept. 2010. http://dx.doi.org/10.1109/MMET.2010.5611418

[110] Mäkitalo, M., and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising", *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 99–109, Jan. 2011. http://dx.doi.org/10.1109/TIP.2010.2056693

[111] Mäkitalo, M., and A. Foi, "Spatially adaptive alpha-rooting in BM3D sharpening", *Proc. SPIE Electronic Imaging: Algorithms and Systems IX*, no. 7870-39, San Francisco, USA, Jan. 2011.

[112] Mäkitalo, M., and A. Foi, "A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation", *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2697–2698, Sept. 2011. http://dx.doi.org/10.1109/TIP.2011.2121085

[113] Mäkitalo, M., and A. Foi, "Poisson-Gaussian denoising using the exact unbiased inverse of the generalized Anscombe transformation", *Proc. 2012 IEEE Int. Conf. Acoustics, Speech, Signal Process., ICASSP 2012*, Kyoto, Japan, pp. 1081–1084, March 2012.

[114] Mäkitalo, M., and A. Foi, "Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise", *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 91–103, Jan. 2013. http://dx.doi.org/10.1109/TIP.2012.2202675

[115] Nadaraya, È. A., "On estimating regression", *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[116] Neyman, J., and E. L. Scott, "Correction for Bias Introduced by a Transformation of Variables", *The Annals of Mathematical Statistics*, vol. 31, no. 3, pp. 643–655, 1960.

[117] Noble, P. J. W., "Self-Scanned Silicon Image Detector Arrays", *IEEE Trans. Electron Devices*, vol. 15, no. 4, pp. 202–209, Apr. 1968.

[118] Nyquist, H., "Thermal Agitation of Electric Charge in Conductors", *Physical Review*, vol. 32, no. 1, pp. 110–113, July 1928.

[119] Olsen, S. I., "Noise Variance Estimation in Images", *8th Scandinavian Conference on Image Analysis*, Tromsø, Norway, May 1993.

[120] Portilla, J., V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain", *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[121] Präkel, D., *The Visual Dictionary of Photography*, AVA Publishing SA, Lausanne, 2010.

[122] Ramani, S., T. Blu, and M. Unser, "Monte-Carlo SURE: A Black-Box Optimization of Regularization Parameters for General Denoising Algorithms", *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1540–1554, Sept. 2008.

[123] Salmon, J., C-A. Deledalle, R. Willett, and Z. Harmany, "Poisson Noise Reduction with Non-Local PCA", *Proc. 2012 IEEE Int. Conf. Acoustics, Speech, Signal Process., ICASSP 2012*, Kyoto, Japan, March 2012.

[124] Sheather, S. J., and M. C. Jones, "A Reliable Data-based Bandwidth Selection Method for Kernel Density Estimation", *Journal of the Royal Statistical Society, Series B*, vol. 53 no. 3, pp. 683–690, 1991.

[125] Spring, K. R., and M. W. Davidson, "Concepts in Digital Imaging Technology: Quantum Efficiency", available online at http://learn.hamamatsu.com/articles/quantumefficiency.html. Retrieved: 18.9.2012

[126] Starck, J-L., F. Murtagh, and A. Bijaoui, *Image Processing and Data Analysis*, Cambridge University Press, Cambridge, 1998.

[127] Starck, J-L., E. J. Candès, and D. L. Donoho, "The Curvelet Transform for Image Denoising", *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, June 2002.

[128] Stein, C. M., "Estimation of the mean of a multivariate normal distribution", *The Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, Nov. 1981.

[129] Stoica, P., and P. Babu, "The Gaussian Data Assumption Leads to the Largest Cramér-Rao Bound", *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 132–133, May 2011.

[130] Takeda, H., S. Farsiu, and P. Milanfar, "Kernel Regression for Image Processing and Reconstruction", *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

[131] Tibshirani, R., "Estimating Transformations for Regression Via Additivity and Variance Stabilization", *Journal of the American Statistical Association*, vol. 83, no. 402, pp. 394–405, June 1988.

[132] Tijms, H. C., *Understanding Probability: Chance Rules in Everyday Life*, 2nd ed, Cambridge University Press, Cambridge, July 2007.

[133] Timmerman, K., and R. Nowak, "Multiscale modeling and estimation of Poisson processes with application to photon-limited imaging", *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 846–862, 1999.

[134] Tippett. L. H. C., "Statistical Methods in Textile Research, Part 2 — Uses of the Binomial and Poisson Distributions in Analysis of Variance", *Journal of the Textile Institute Transactions*, vol. 26, no. 1, pp. T13–T50, 1935.

[135] Tong, H. H. Y., and A. N. Venetsanopoulos, "A perceptual model for JPEG applications based on block classification, texture masking, and luminance masking", *1998 International Conference on Image Processing, ICIP'98*, vol. 3, 1998.

[136] Vonesch, C., F. Aguet, J-L. Vonesch, and M. Unser, "The colored revolution of bioimaging: An introduction to fluorescence microscopy", *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 20–31, May 2006.

[137] Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[138] Watson, G. S., "Smooth regression analysis", *Sankhyā: The Indian Journal of Statistics, Series A*, vol. 26, no. 4, pp. 359–372, Dec. 1964.

[139] Weimer, P. K., W. S. Pike, G. Sadasiv, F. V. Shallcross, and L. Meray-Horvath, "Multielement self-scanned mosaic sensors", *IEEE Spectrum*, vol. 6, no. 3, pp. 52–65, March 1969.

[140] Weissman, M. B., "$1/f$ noise and other slow, nonexponential kinetics in condensed matter", *Reviews of Modern Physics*, vol. 60, no. 2, pp. 537–571, 1988.

[141] Willett, R. M., and R. D. Nowak, "Platelets: A Multiscale Approach for Recovering Edges and Surfaces in Photon-Limited Medical Imaging", *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 332–350, March 2003.

[142] Willett, R. M., "Multiscale Analysis of Photon-Limited Astronomical Images", *Statistical Challenges in Modern Astronomy (SCMA) IV*, 2006.

[143] Williams, C. B., "The use of logarithms in the interpretation of certain entymological problems", *Annals of Applied Biology*, vol. 24, pp. 404–414, 1937.

[144] Zanella, G., and R. Zannoni, "Design of CCD-based X-ray area detectors in terms of DQE", *Nuclear Instruments and Methods in Physics Research, Section A*, vol. 406, no. 1, pp. 93–102, March 1998.

[145] Zhang, B., J. M. Fadili, J-L. Starck, and J. C. Olivo-Marin, "Multiscale variance-stabilizing transform for mixed-Poisson-Gaussian processes and its applications in bioimaging", *Proc. IEEE Int. Conf. Image Process., ICIP 2007*, pp. VI-234–VI-236, 2007.

[146] Zhang B., J. M. Fadili, and J-L. Starck, "Wavelets, ridgelets, and curvelets for Poisson noise removal", *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1093–1108, July 2008.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

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

ISBN 978-952-15-3021-0
ISSN 1459-2045