DISSERTATION
submitted
to the
Combined Faculties for the Natural Sciences and for Mathematics
of the
Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by

Diplom-Inform. Stanislav Pyatykh
Born in Ekaterinburg, Russia

Image Noise Parameter Estimation by Principal Component Analysis

Advisor: Prof. Dr. Jürgen Hesser

Abstract

Noise parameter estimation is an important image processing step, because the noise parameters are often unknown, but many image denoising, compression, and segmentation algorithms take them as input values. The innovation of this thesis is the introduction of a new noise parameter estimation framework. The framework is designed to handle images with signal-independent noise as well as several common types of signal-dependent noise, namely, noise produced by synthetic aperture radars (SAR), magnetic resonance imaging (MRI) devices, charge-coupled device (CCD) sensors, and ultrasound devices.

The framework is based on a sparse representation of the blocks of the original image. Specifically, it is assumed that a part of the original image blocks lies in a proper subspace of the image block vector space, which means that there is a linear dependence between pixels in the blocks. As a result, images without homogeneous areas can be accurately processed, which is a qualitative difference from the state of the art, where homogeneous areas are required to process images with signal-dependent noise with several parameters.

In the case of signal-independent noise, principal component analysis of the blocks of the input image is utilized in order to check the sparsity assumption and estimate the noise variance. Particularly, Bartlett's test and the difference of the sample covariance matrix eigenvalues are used as assumption checks, and the last several sample covariance matrix eigenvalues are utilized to estimate the noise variance. Besides, two strategies to select the part of the blocks, which allows the sparse representation, are suggested.

In order to process images with signal-dependent noise, a variance-stabilizing transformation is applied. An optimization procedure is used to compute the transformation parameters, because they depend on the unknown noise parameters. This procedure analyzes the noise distribution in the transformed image and selects the transformation parameters, which maximize a noise normality measure. After applying the variance-stabilizing transformation, the algorithm designed for signal-independent noise is utilized to estimate the noise variance in the transformed image; and the parameters of the original noise model are calculated.

The noise parameter estimation experiments, which include comparison with 19 state of the art methods, show that the accuracy of the proposed algorithms is the highest in most cases. Speaking of signal-independent noise, the proposed algorithm gives a good compromise between accuracy and execution time: it is at

least 15 times faster compared with the methods with similar accuracy; and it is at least 2 times more accurate than other methods. Regarding signal-dependent noise, the accuracy of the proposed approach is considerably higher for the SAR, CCD, and ultrasound noise models. The denoising experiments demonstrate that the use of noise parameter estimates computed by the proposed method results in considerably higher denoising quality for these three noise models.

Zusammenfassung

Die Schätzung der Parameter für Bildrauschen ist ein wichtiger Bildverarbeitungsschritt. Oft sind diese Parameter unbekannt, obwohl eine ganze Reihe von Methoden der Rauschunterdrückung, Kompression und Segmentierung diese Daten benötigen. In dieser Dissertation wird ein neues Framework für die Schätzung von Bildrauschen vorgestellt. Es werden sowohl Fälle mit signalunabhängigem Rauschen, als auch häufig vorkommende Varianten von signalabhängigem Rauschen (angefangen von Synthetic Aperture Radar, Magnetresonanztomographie (MRT), CCD-Sensoren und Ultraschall) berücksichtigt.

Das Framework basiert auf der Vermutung der spärlichen Codierung von Blocks des Originalbildes, genauer gesagt, auf der Vermutung, dass das Originalbild vollständig in einem echten Untervektorraum eines Block-Vektorraums abgebildet werden kann. Das bedeutet, dass es eine lineare Abhängigkeit zwischen den Pixeln in den Blocks gibt. Deswegen ist es möglich, auch Bilder ohne gleichartige Bereiche zu verarbeiten. Das ist ein qualitativer Unterschied zum Stand der Technik, der gleichartige Bereiche fordert, um Bilder mit signalabhängigem Rauschen mit mehreren Parametern zu verarbeiten.

Im Falle von signalunabhängigem Rauschen wird die Hauptkomponentenanalyse der Blocks des Eingangsbildes genutzt, um die Spärlichkeit zu überprüfen und die Rauschvarianz zu schätzen. Insbesondere werden der Bartlett-Test und die Differenz der Stichproben-Kovarianzmatrix-Eigenwerte zur Überprüfung der Vermutung genutzt. Außerdem werden zwei Strategien zur Auswahl desjenigen Unterraums von Blocks, auf den sich die spärliche Darstellung des Bildes konzentriert, vorgeschlagen.

Um Bilder mit signalabhängigem Rauschen zu verarbeiten wird eine varianzstabilisierende Transformation verwendet. Ein Optimierungsverfahren wird benutzt, um die Transformationsparameter zu berechenen, da sie von den unbekannten Rauschparametern abhängen. Dieses Verfahren analysiert die Rauschverteilung im transformierten Bild und wählt die Transformationsparameter so aus, dass ein Maß der Rauschnormalität maximiert wird. Nach der Anwendung der varianzstabilisierenden Transformation wird der Algorithmus für signalunabhängiges Rauschen genutzt, um die Varianz im transformierten Bild zu schätzen, und somit die Parameter des ursprünglichen Rauschens zu bestimmen.

Die Experimente zur Rauschparameter-Schätzung, die den Vergleich mit den 19 Methoden des Standes der Technik untersuchen, zeigen, dass die vorgeschlagenen Algorithmen fast immer am genauesten sind. Im Falle von signalun-

abhängigem Rauschen ist der vorgeschlagene Algorithmus ein guter Kompromiss zwischen Genauigkeit und Geschwindigkeit gegenüber den besten konkurrierenden Verfahren: Er ist mindestens 15 mal schneller im Vergleich mit den Methoden mit einer ähnlichen Genauigkeit, und er ist mindestens doppelt so genau im Vergleich mit den übrigen Methoden. In Bezug auf signalabhängiges Rauschen ist die Genauigkeit des vorgeschlagenen Algorithmus deutlich höher für SAR, CCD und Ultraschall Rauschmodelle. Die Rauschunterdrückungs-Experimente zeigen, dass die Verwendung der von der vorgeschlagenen Methode bestimmten Rauschparameter zu deutlich höherer Rauschunterdrückungsqualität für diese drei Rauschmodelle führt.

Acknowledgment

This thesis presents the work, which I have done at University Medical Center Mannheim, Heidelberg University, Germany. I would like to thank Prof. Dr. Jürgen Hesser for his excellent guidance, for introducing me to interesting image processing problems, and for providing a pleasant working environment. I would also like to thank Lei Zheng for the fruitful discussions and for his great assistance in preparing the publications.

I am grateful to Dr. Oleg Krivonos, Dzmitry Stsepankou, and Christiane Glasbrenner for their help. I am also grateful to University Medical Center Mannheim and Dr. Sennewald Medizintechnik GmbH for the financial support, which has made this research work possible.

Stanislav Pyatykh
Mannheim, January 2013

Contents

Notation

The following denotations and abbreviations are used in this thesis:

$d \in \mathbb{N}$	image dimension
$\mathbf{p} = (p_1, \ldots, p_d)^{\mathrm{T}} \in \mathbb{N}^d$	pixel location
$(S_1, \ldots, S_d)^{\mathrm{T}} \in \mathbb{N}^d$	image size
$x(\mathbf{p})$	original (noise-free and unknown) image
$n(\mathbf{p})$	white Gaussian noise; $n(\mathbf{p}) \sim \mathcal{N}(0,1)$ for each $\mathbf{p}$
$y(\mathbf{p})$	noisy image
$M_1 \times \cdots \times M_d$	image block size
$M = M_1 \cdot \ldots \cdot M_d$	number of pixels in each image block
$\mathbf{x}_i, \mathbf{n}_i, \mathbf{y}_i \in \mathbb{R}^M$	vectors formed from blocks of x, n, y (see Section 3.2)
$\mathbf{X}, \mathbf{N}, \mathbf{Y} \in \mathbb{R}^M$	random vectors with realizations $\mathbf{x}_i$, $\mathbf{n}_i$, $\mathbf{y}_i$ (see Section 3.2)
$\mathbf{\Sigma_X}, \mathbf{\Sigma_Y} \in \mathbb{R}^{M \times M}$	population covariance matrices of $\mathbf{X}$ and $\mathbf{Y}$
$S_\mathbf{Y} \in \mathbb{R}^{M \times M}$	sample covariance matrix of $\mathbf{Y}$
$\lambda_{\mathbf{X},i}, \lambda_{\mathbf{Y},i} \in \mathbb{R}_{\geq 0}$	eigenvalues of $\mathbf{\Sigma_X}$ and $\mathbf{\Sigma_Y}$
$\mathbf{v}_{\mathbf{X},i} \in \mathbb{R}^M$	normalized eigenvectors of $\mathbf{\Sigma_X}$
$\tilde{\lambda}_{\mathbf{Y},i} \in \mathbb{R}_{\geq 0}$	eigenvalues of $S_\mathbf{Y}$
$\tilde{\mathbf{v}}_{\mathbf{Y},i} \in \mathbb{R}^M$	normalized eigenvectors of $S_\mathbf{Y}$
$\mathbf{w}$	vector of the noise parameters
$h(\cdot; \mathbf{w})$	noise standard deviation as a function of the pixel mean
$g(\cdot; \mathbf{w})$	variance-stabilizing transformation
$s(\cdot)$	sample standard deviation
$s^2(\cdot)$	sample variance
$Q(p, y)$	p-quantile of $\{s^2(\mathbf{y}_i)\}$ (see Section 3.5.2)
$B(p, y)$	block subset $\{\mathbf{y}_i \vert s^2(\mathbf{y}_i) \leq Q(p, y)\}$ (see Section 3.5.2)
I_k	$k \times k$ identity matrix
$\lVert \cdot \rVert_2$	spectral matrix norm
$\mathbf{1}_X$	indicator function of set X
$\lfloor t \rfloor \in \mathbb{Z}$	largest integer not greater than t

AWGN	additive white Gaussian noise
CCD	charge-coupled device
CDF	cumulative distribution function
CMOS	complementary metal-oxide-semiconductor
DCT	discrete cosine transform

MAD	median absolute deviation
MR	magnetic resonance
MRI	magnetic resonance imaging
PCA	principal component analysis
PCA-BS	variance estimation method with block selection (see Section 3.5)
PCA-RS	variance estimation method with region selection (see Section 3.6)
PSNR	peak signal-to-noise ratio
SAR	synthetic aperture radar
VST	variance-stabilizing transformation

Introduction

Many image denoising [12, 18], compression [90], and segmentation [73] algorithms take the parameters of the noise as input values. However, these parameters may not be known beforehand, because they may depend on sensor's operational conditions or the calibration data may not be available [88]. As a result, blind noise parameter estimation is often necessary.

A noise parameter estimator is required to be highly accurate, because it heavily affects the performance of the above-mentioned image processing algorithms. For example, in image denoising, some noise remains in the output image when the noise level is underestimated, whereas overestimating the noise level results in oversmoothing the output image. In image compression, additional data should be stored in order to encode the noise if its level is underestimated. On the other hand, overestimating the noise level leads to loss of useful information.

There are two main reasons, which make the development of a noise parameter estimation algorithm a difficult task. First, the original image can contain only textures and no homogeneous areas, which complicates signal and noise separation. Second, for many imaging devices, the noise variance in each pixel depends on the original image intensity in this pixel, so that the noise significantly deviates from additive white Gaussian noise (AWGN). In this case, several parameters may be required to describe the noise distribution.

Currently available methods can process textured images corrupted with AWGN by applying certain transforms on image blocks, which allow accurate signal and noise separation [65, 21]. Nevertheless, the existence of homogeneous areas is still necessary to process images corrupted with signal-dependent noise, because signal-dependent noise parameter estimation is always realized by the construction of the scatter-plot for the local mean and the local variance.

This work describes a new noise parameter estimation framework, which contains algorithms for processing signal-independent noise as well as signal-dependent noise. Principal component analysis (PCA) of image blocks is utilized for signal and noise separation. First, image blocks of a fixed size are generated from the input image in a sliding window manner. Then, these blocks are rearranged into vectors and PCA of these vectors is done. It is assumed that the last principal component is not affected by the signal and corresponds only to the noise, so that one can:

1. in the case of signal-independent noise: estimate the noise variance as the variance of the last principal component.

2. in the case of signal-dependent noise: assess noise normality by analyzing the distribution of the last principal component; and select the variance-stabilizing transformation (VST) parameters in order to transform the noise into AWGN.

The proposed algorithms do not require the existence of homogeneous areas in the original image, hence images containing only textures can be accurately processed for all considered noise types.

Structure of the thesis

This thesis in organized as follows. Most common noise sources and noise types are described in Chapter 1. For each noise type, the noise distribution is derived using the imaging system properties. Existing noise parameter estimation algorithms are considered in Chapter 2. Since many of them have common ideas, the algorithms are grouped by the assumption about the input image and by the approach. In Chapter 3, the proposed white noise variance estimation method is explained. First, the necessary mathematical framework is developed and analyzed. Next, the algorithm is presented and evaluated. The extension to signal-dependent noise is given in Chapter 4, which contains a description of VSTs and noise normality assessment methods as well as the proposed algorithm and the evaluation. The denoising experiments presented in Chapter 5 show how the usage of noise parameters computed by different methods affects the denoising quality. These experiments confirm the applicability of the proposed noise parameter estimation framework. The conclusion is given in Chapter 6.

Scientific contribution

The following publications have been prepared during the research work presented in this thesis:

1. [69]: S. Pyatykh, L. Zheng, and J. Hesser, "Efficient method of pixel neighborhood traversal", Journal of Visual Communication and Image Representation, vol. 23, issue 5, pp. 719–728, Jul. 2012.

2. [70]: S. Pyatykh, J. Hesser, and L. Zheng, "Image Noise Level Estimation by Principal Component Analysis", IEEE Transactions on Image Processing, vol. 22, issue 2, pp. 687–699, Feb. 2013.

3. [71]: S. Pyatykh, L. Zheng, and J. Hesser, "Fast Noise Variance Estimation by Principal Component Analysis", Image Processing: Algorithms and Systems XI, IS&T/SPIE Electronic Imaging 2013, Feb. 3-7, 2013, to be presented

4. [68]: S. Pyatykh, L. Zheng, and J. Hesser, "Signal-dependent Noise Parameter Estimation by Principal Component Analysis", submitted to IEEE Transactions on Image Processing

The articles [70] and [71] form the basis of Chapter 3, whereas Chapter 4 is based on the work [68]. Chapter 5 contains experiments presented in [70] as well as in [68].

The work [69] presents a flexible and computationally efficient implementation of pixel neighborhood traversal. Since this work is not directly related to noise parameter estimation, it is not included in this thesis.

Chapter 1

Noise models

Noise is a result of different physical processes taking place during the image acquisition, and a precise description of these processes for modern sophisticated imaging systems is complex. On the other hand, digital image processing applications require simple noise models, which can be easily analyzed, while a proper modeling of the acquisition process is not critical. Therefore, a noise model is a trade-off between the simplicity and the accurate description of the noise properties. It provides an abstraction over the underlying physical processes so that one noise model can be often applied to images acquired by essentially different devices.

Let d be the image dimension, $\mathbf{p} = (p_1, \ldots, p_d) \in \mathbb{Z}^d$ be a pixel location, $x(\mathbf{p})$ be the original (noise-free and unknown) image of size $(S_1, S_2, \ldots, S_d)$, and $y(\mathbf{p})$ be the corresponding noisy image. Let us also denote a normal random variable with zero mean and unit variance by $n(\mathbf{p}) \sim \mathcal{N}(0, 1)$. $n(\mathbf{p}_1)$ and $n(\mathbf{p}_2)$ for $\mathbf{p}_1 \neq \mathbf{p}_2$ are statistically independent. Further in this work, pixel location $\mathbf{p}$ is omitted in complex expressions.

1.1 Additive white Gaussian noise

AWGN is defined as

$$y(\mathbf{p}) = x(\mathbf{p}) + \sigma n(\mathbf{p}) \tag{1.1}$$

where σ is the noise standard deviation. That is, AWGN is signal-independent and it is fully characterized by parameter σ. AWGN has become one of the most common noise models because of the following remarkable properties:

1. In many cases, AWGN is a good approximation to the real noise. The central limit theorem implies that a Gaussian random variable can be used to model the result of a large number of independent random effects. For example, electric and thermal noise can be approximated by Gaussian noise [32]. The number of photons arriving the detector, which has a Poisson distribution, can also be approximated by the Gaussian distribution and

the approximation error tends to zero as the average photon count tends to the infinity.

2. The mathematical tractability of the Gaussian distribution allows construction of probabilistic models for various transforms used in image processing algorithms. For instance, the projection of a Gaussian random vector onto a constant vector is a Gaussian random variable, hence the coefficients of an affine transform applied to a Gaussian random vector have Gaussian distribution. This fact is utilized in the algorithms, which apply an affine transform to the vectors formed from image blocks. In particular, it is used in the algorithms [65, 63, 21, 18], which apply DCT of image blocks, as well as in the methods [48, 61, 85, 70, 71, 68], which utilize PCA of image blocks.

For this noise model, the problem is to estimate parameter σ.

1.2 SAR noise

A SAR satellite transmits radar pulses, which are reflected from the Earth and collected by the same SAR [41]. A resolution cell on the Earth's surface, which corresponds to one pixel in the acquired image, contains many randomly distributed scatterers. Waves, which are reflected from these scatterers, form the returned signal, whose strength depends on the interference of the waves: the strength is high, if the waves are in phase and vise versa.

If there are m scatterers in the resolution cell, the returned signal is the sum of the reflected waves [51]:

$$z_k = \sum_{j=1}^{m} z_{kj} \quad z_{kj} \in \mathbb{C} \tag{1.2}$$

where k is the look index (for N-look SAR) and j is the scatterer index. If the scatterers are independent and identically distributed and m is sufficiently large, $\operatorname{Re} z_k$ and $\operatorname{Im} z_k$ are independent and have Gaussian distribution with zero mean and equal variance, which is denoted by $A^2/2$. Then, the signal intensity $y_k = |z_k|^2$ has the exponential distribution with the probability density function

$$f(t) = \frac{1}{A^2} \exp\Big(- \frac{t}{A^2} \Big) \quad t \geq 0. \tag{1.3}$$

In order to obtain an N-look intensity SAR image, the synthetic aperture length is divided into N sections and one intensity SAR image is produced for each section independently [51]. Then, the intensities are averaged:

$$y = \frac{1}{N} \sum_{k=1}^{N} y_k \tag{1.4}$$

Therefore, grayvalues y in the intensity N-look SAR image have the gamma distribution with the probability density function

$$f(t) = \frac{N^N t^{N-1}}{(N-1)!A^{2N}} \exp\Big(-\frac{Nt}{A^2}\Big) \quad t \geq 0. \tag{1.5}$$

For this distribution, the mean equals A^2 and the standard deviation is $A^2/\sqrt{N}$, so that the standard deviation to mean ratio equals $1/\sqrt{N}$ and is independent of A.

In order to obtain an N-look amplitude SAR image, the intensity image is square-rooted, so that grayvalues $\sqrt{y}$ have the generalized gamma distribution with the probability density function

$$f(t) = \frac{2N^N t^{2N-1}}{(N-1)!A^{2N}} \exp\Big(-\frac{Nt^2}{A^2}\Big) \quad t \geq 0. \tag{1.6}$$

For this distribution, the mean is

$$\mathbf{E}(\sqrt{y}) = \frac{A\Gamma(N+1/2)}{\sqrt{N}\Gamma(N)} \tag{1.7}$$

and the standard deviation is

$$\mathrm{std}(\sqrt{y}) = \sqrt{\mathbf{E}(y) - \mathbf{E}^2(\sqrt{y})} = \sqrt{A^2 - \frac{A^2\Gamma^2(N+1/2)}{N\Gamma^2(N)}} \tag{1.8}$$

Hence the standard deviation to mean ratio equals

$$\sqrt{\frac{N\Gamma^2(N)}{\Gamma^2(N+1/2)} - 1} \tag{1.9}$$

so that it is independent of A^2.

As a result, the standard deviation to mean ratio is constant for both N-look intensity SAR images and N-look amplitude SAR images. Hence SAR noise can be modeled as multiplicative noise with unit mean. As one can see in Fig. 1.1, where the probability density functions for N-look intensity SAR images (1.5) and N-look amplitude SAR images (1.6) are compared with the Gaussian probability density functions, the grayvalue distribution for N-look intensity SAR images differs a little from the Gaussian distribution, and the grayvalue distribution for N-look amplitude SAR images fits to the Gaussian distribution very well. Therefore, the multiplicative noise can be assumed to have the normal distribution for sufficiently large N [63]. Signal-independent noise in SAR images is very small and can be neglected [86], so that we have the following noise model:

$$y(\mathbf{p}) = x(\mathbf{p})(1 + an(\mathbf{p})) = x(\mathbf{p}) + ax(\mathbf{p})n(\mathbf{p}) \quad a \geq 0. \tag{1.10}$$

For this model, the problem is to estimate parameter a.

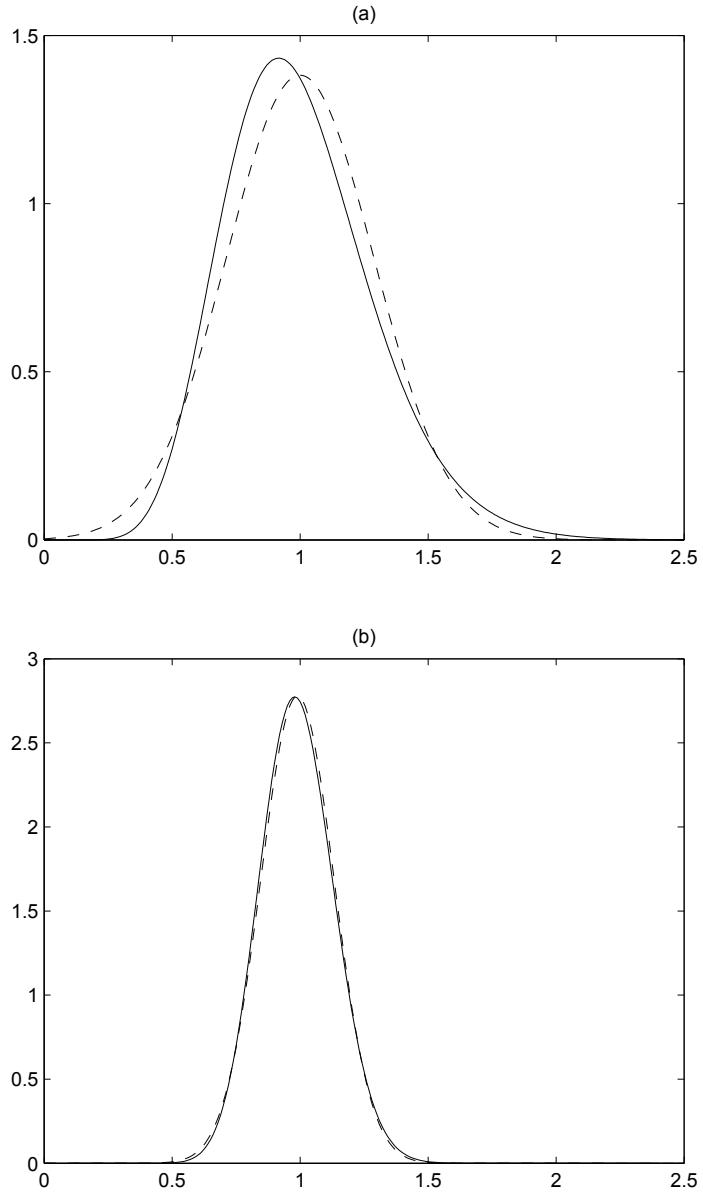

Figure 1.1: Solid lines: The probability density functions for N-look SAR images for $A = 1$ and $N = 12$. Dashed lines: Gaussian probability density functions for the same mean and standard deviation. (a) N-look intensity SAR (1.5). (b) N-look amplitude SAR (1.6).

1.3 MRI noise

It is usually assumed that the real and imaginary parts of the complex raw data acquired by a MR scanner have Gaussian noise. Hence the complex image reconstructed by the Fourier transform is also corrupted with Gaussian noise [23]:

$$\begin{aligned} y_{re}(\mathbf{p}) &\sim \mathcal{N}\big(x(\mathbf{p})\cos\alpha, a^2\big), \\ y_{im}(\mathbf{p}) &\sim \mathcal{N}\big(x(\mathbf{p})\sin\alpha, a^2\big) \end{aligned} \tag{1.11}$$

where y_{re} and y_{im} are the real and imaginary parts respectively and $a \geq 0$ is the noise standard deviation. Then, the magnitude of the complex image $y(\mathbf{p}) = \sqrt{y_{re}^2(\mathbf{p}) + y_{im}^2(\mathbf{p})}$ has Rician noise:

$$y(\mathbf{p}) \sim Rice\big(x(\mathbf{p}), a\big). \tag{1.12}$$

It is defined by the following probability density function [30]:

$$f(t) = \frac{t}{a^2}\exp\Big(-\frac{t^2 + x^2(\mathbf{p})}{2a^2}\Big) I_0\Big(\frac{tx(\mathbf{p})}{a^2}\Big) \quad t \geq 0 \tag{1.13}$$

where I_m is the modified Bessel function of order m:

$$I_m(t) = \sum_{k=0}^{\infty} \frac{(t/2)^{m+2k}}{k!\Gamma(m+k+1)}. \tag{1.14}$$

The mean and the variance of $y(\mathbf{p})$ can be written as [30]

$$\begin{aligned} \mathbf{E}(y(\mathbf{p})) &= a\sqrt{\frac{\pi}{2}} L\Big(-\frac{x^2(\mathbf{p})}{2a^2}\Big) \\ \operatorname{var}(y(\mathbf{p})) &= 2a^2 + x^2(\mathbf{p}) - \frac{\pi a^2}{2} L^2\Big(-\frac{x^2(\mathbf{p})}{2a^2}\Big) \end{aligned} \tag{1.15}$$

where

$$L(t) = e^{t/2}\big((1-t)I_0(-t/2) - tI_1(-t/2)\big) \tag{1.16}$$

is a Laguerre polynomial.

Asymptotic behavior for the mean and the variance for the case when a is fixed and $x(\mathbf{p})$ tends to infinity can be analyzed using the following asymptotic expression for L [27]:

$$L(z) = \frac{2}{\sqrt{\pi}}\sqrt{-z}\Big(1 - \frac{1}{4z} + \frac{1}{32z^2}\Big) + O(|z|^{-5/2}) \quad z \in \mathbb{C},\ \operatorname{Re} z \to -\infty. \tag{1.17}$$

Substituting (1.17) into (1.15), we have

$$\begin{aligned} \mathbf{E}(y) &= x + \frac{a^2}{2x} + \frac{a^4}{8x^3} + O(x^{-5}) \\ \operatorname{var}(y) &= a^2 - \frac{a^4}{2x^2} + O(x^{-4}). \end{aligned} \tag{1.18}$$

Since $\mathbf{E}(y) = x + O(x^{-1})$ for $x \to +\infty$ implies that $x = \mathbf{E}(y) + O(\mathbf{E}^{-1}(y))$ for $\mathbf{E}(y) \to +\infty$, we can also express the variance as a function of the mean:

$$\operatorname{var}(y) = a^2 - \frac{a^4}{2\mathbf{E}^2(y)} + O(\mathbf{E}^{-4}(y)). \tag{1.19}$$

Moreover, when $x(\mathbf{p})$ is sufficiently large compared with a, Rice distribution can be approximated by the normal distribution [62]. This is illustrated in Fig. 1.2, where the probability density functions of Rice distribution with different parameters are compared with the normal probability density functions. Note that due to the scaling property [30]

$$ky(\mathbf{p}) \sim Rice\big(kx(\mathbf{p}), ka\big) \quad k > 0 \tag{1.20}$$

it is sufficient to consider only Rice distributions with fixed parameter a. As it can be seen in the figure, the Gaussian distribution is a very good approximation to Rice distribution already when $x(\mathbf{p})$ is two times larger than a. Therefore, noise in MR images can be approximated by the following model:

$$y(\mathbf{p}) = \mathbf{E}(y(\mathbf{p})) + \operatorname{std}(y(\mathbf{p}))n(\mathbf{p}). \tag{1.21}$$

As follows from (1.18), when a is fixed and $x(\mathbf{p})$ tends to infinity, this model tends to AWGN (1.1).

For this noise model, the problem is to estimate parameter a.

1.4 CCD/CMOS noise

In CCD and CMOS sensors, noise has two main components [32]:

1. signal-dependent Poisson noise, which is explained by the fact that the number of detected photons is random and have Poisson distribution;
2. signal-independent electric and thermal noise added to the signal by the sensor's hardware.

Let $\omega(\mathbf{p}) \sim Pois\big(\lambda(\mathbf{p})\big)$ be the number of detected photons. From the properties of the Poisson distribution, $\lambda(\mathbf{p})$ is the expected value and the variance of $\omega(\mathbf{p})$. Let e_q be the sensor's quantum efficiency, i.e. the reciprocal of the number of photons necessary to generate an electron. Then the collected charge equals $e_q\omega(\mathbf{p})$. This charge is added to pedestal level $p_0 \geq 0$, which results in the output signal

$$e_q\omega(\mathbf{p}) + p_0. \tag{1.22}$$

Signal-independent noise is modeled by addition of Gaussian random variable $\sigma_1 n_1(\mathbf{p}) \sim \mathcal{N}(0, \sigma_1^2)$:

$$e_q\omega(\mathbf{p}) + p_0 + \sigma_1 n_1(\mathbf{p}). \tag{1.23}$$

Next, the signal is amplified by factor $\theta > 0$, which is related to the sensor's analog gain, so that the output signal is

$$\theta e_q\omega(\mathbf{p}) + \theta p_0 + \theta\sigma_1 n_1(\mathbf{p}). \tag{1.24}$$

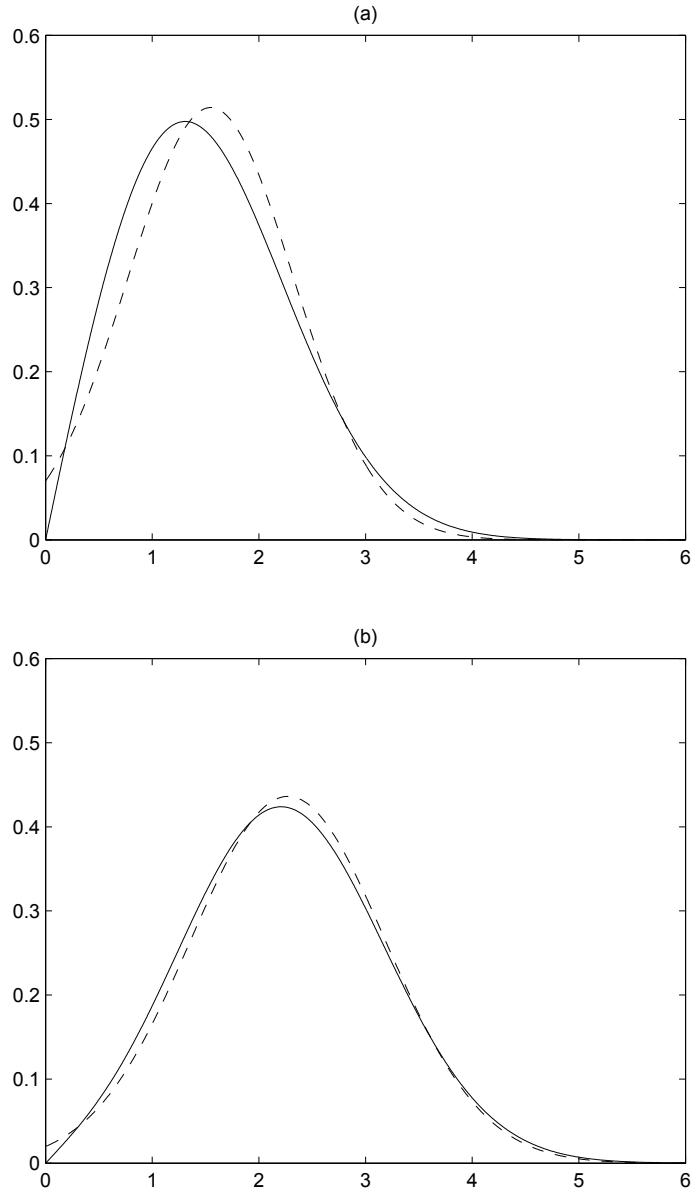

Figure 1.2: Solid lines: The probability density functions of Rice distribution for $a = 1$. Dashed line: Gaussian probability density functions for the same mean and standard deviation. (a) $x(\mathbf{p}) = 1$. (b) $x(\mathbf{p}) = 2$.

Hence the larger the analog gain (which corresponds to the ISO sensitivity in digital cameras) is, the brighter the image is and the higher the noise level is. Finally, Gaussian random variable $\sigma_2 n_2(\mathbf{p}) \sim \mathcal{N}(0, \sigma_2^2)$ representing signal-independent noise, which affects the signal after the amplification, is added, which leads to the following model:

$$y(\mathbf{p}) = \theta e_q \omega(\mathbf{p}) + \theta p_0 + \theta \sigma_1 n_1(\mathbf{p}) + \sigma_2 n_2(\mathbf{p}). \tag{1.25}$$

$$x(\mathbf{p}) = \theta e_q \omega(\mathbf{p}) + \theta p_0 \tag{1.26}$$

Since n_1 and n_2 are independent from the signal and from each other,

$$\mathrm{var}(y) = \theta^2 e_q^2 \lambda + \theta^2 \sigma_1^2 + \sigma_2^2 \overset{(1.26)}{=} \theta e_q x - \theta^2 e_q p_0 + \theta^2 \sigma_1^2 + \sigma_2^2. \tag{1.27}$$

Denoting $a = \theta e_q$ and $b = -\theta^2 e_q p_0 + \theta^2 \sigma_1^2 + \sigma_2^2$,

$$\mathrm{var}(y) = ax + b. \tag{1.28}$$

Therefore, the noise variance linearly depends on the original pixel value. Note that b can be negative, but $ax + b$ is always nonnegative.

For sufficiently large λ, ω can be approximated by a normal random variable with mean λ and variance λ [32, 39]. In this case, y has the normal distribution with mean x and variance $ax + b$ from (1.25); and CCD/CMOS noise can be represented as additive Gaussian noise with signal-dependent variance:

$$y(\mathbf{p}) = x(\mathbf{p}) + \sqrt{ax(\mathbf{p}) + b}\; n(\mathbf{p}). \tag{1.29}$$

Some examples of noise standard deviation functions $\sqrt{ax + b}$ for the Canon PowerShot G10 digital camera [28] are shown in Fig. 1.3.

For this noise model, the problem is to estimate parameters a and b.

1.5 Ultrasound/film-grain noise

Ultrasound imaging model is similar to that of SAR so that noise can be modeled by the Rayleigh distribution with the probability density function

$$f(t) = \frac{2t}{A^2} \exp\Big(- \frac{t^2}{A^2} \Big) \quad t \geq 0. \tag{1.30}$$

The mean of this distribution equals $A\sqrt{\pi}/2$ and the standard deviation equals $A\sqrt{4 - \pi}/2$. That is, the standard deviation to mean ratio is constant and the multiplicative noise model could be used. However, the distribution can be modified by the preprocessing stages in the ultrasound scanner such as logarithmic compression, low-pass filtering, and interpolation [56]. Hence the mean can be proportional not to the standard deviation but to the variance; and the noise model has the following form:

$$y(\mathbf{p}) = x(\mathbf{p}) + ax^b(\mathbf{p}) n(\mathbf{p}) \tag{1.31}$$

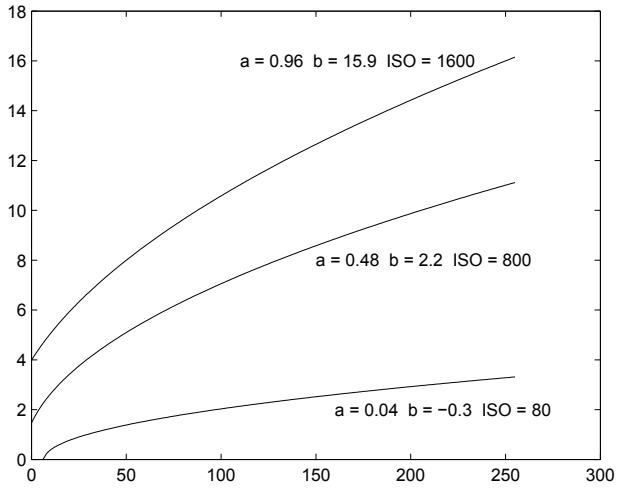

Figure 1.3: Noise standard deviation functions $\sqrt{ax+b}$ for the Canon PowerShot G10 digital camera and different ISO sensitivities [28]. The grayvalue range is $[0, 255]$.

where b equals 0.5 or 1 and $a \geq 0$.

The image formation on photographic film is based on the chemical properties of silver halide grains [94]. Since the silver grains are randomly distributed in the film emulsion and their behavior during the exposure and development is random, photographic film has noise, which is called film-grain noise.

Because film grains are distributed without crowding or clumping and the scanner resolution is much lower than the grain size, the number of grains in a pixel is statistically independent of the number of grains in other pixels. Let S be the grain size, A be the area of the film surface corresponding to one pixel, and m be the number of grains in A. m has the Poisson distribution so that $\mathbf{E}(m) = \mathrm{var}(m)$; and $\mathbf{E}(m)$ is sufficiently large to approximate the distribution of m by the Gaussian distribution. Additionally, assume that the overlapping of the projections of all grains onto a single plane is negligible. Then, the part of area A not covered by grains is $(A - mS)$ and the film transmittance equals [20]

$$T = \frac{A - mS}{A} = 1 - \frac{mS}{A}. \tag{1.32}$$

Hence the optical density is computed as

$$y = -\log_{10} T = -\log_{10}\left(1 - \frac{mS}{A}\right) = -\log_{10} e \cdot \ln\left(1 - \frac{mS}{A}\right). \tag{1.33}$$

If $mS/A \ll 1$ then

$$y \approx \frac{mS}{A} \log_{10} e \tag{1.34}$$

and the noise variance is written as [20]

$$\mathrm{var}(y) = \Big(\frac{S \log_{10} e}{A}\Big)^2 \mathrm{var}(m) = \Big(\frac{S \log_{10} e}{A}\Big)^2 \mathbf{E}(m) = \frac{S \log_{10} e}{A} \mathbf{E}(y). \tag{1.35}$$

This equation suggests that film-grain noise can be modeled by (1.31) with $a = \sqrt{(S \log_{10} e)/A}$ and $b = 0.5$. However, experiments show that exponent b in (1.31) can vary between 0.2 and 0.7 [94]. Hence in order to apply model (1.31) for both ultrasound and film-grain noise, parameters a and b should be taken from intervals $[0, +\infty)$ and $(0, 1]$ respectively.

For this noise model, the problem is to estimate parameters a and b.

1.6 Summary

As one can see, all considered noise types can be approximated by additive Gaussian noise with signal-dependent standard deviation:

$$y = \mathbf{E}(y) + \mathrm{std}(y) n = \mathbf{E}(y) + h(\mathbf{E}(y); \mathbf{w}) n \tag{1.36}$$

where $\mathbf{w}$ is the vector of the model parameters and $h(\cdot; \mathbf{w})$ denotes the standard deviation of y as a function of its mean. h is called the standard deviation function. The parameters and standard deviation functions for all considered models are given in Table 1.1. The noise parameter estimation problem is therefore the problem of estimation of $\mathbf{w}$.

Table 1.1: The considered noise models of the form $y = \mathbf{E}(y) + h(\mathbf{E}(y); \mathbf{w}) n$.

Noise type	$\mathbf{w}$	$h(t; \mathbf{w})$
AWGN	σ	σ
SAR	a	at
MRI	a	$\sqrt{a^2 + O(t^{-2})}$, $t \to +\infty$
		$\sqrt{a^2 - \frac{a^4}{2t^2} + O(t^{-4})}$, $t \to +\infty$
CCD/CMOS	(a, b)	$\sqrt{at + b}$
Ultrasound, film-grain	(a, b)	at^b

Chapter 2

State of the art in noise parameter estimation

Noise parameter estimation algorithms were being developed over the last two decades and they can be classified in different ways:

1. By the approach. The majority of the methods tries to separate the original content from the noise in the input image or its part and, then, to estimate the parameters of the separated noise [77, 80, 1, 87, 52, 15, 9, 75, 95, 55, 76, 83, 24, 38, 4, 89, 35, 65, 21, 93, 7, 3, 59, 2, 51, 30, 16, 37, 72, 81, 32, 33, 10, 86]. There are also algorithms, which assess the noise parameters indirectly, without signal and noise separation [96, 22, 78].

2. By the assumptions about the original image. Most of the methods assume the existence of homogeneous areas in the original image [77, 80, 1, 87, 52, 15, 9, 75, 95, 55, 76, 4, 3, 89, 35, 2, 51, 30, 16, 37, 72, 81, 32, 33, 10, 86]. However, one can also utilize the image self-similarity [21], properties of the block DCT coefficients [65], properties of the kurtosis of marginal bandpass filter response distributions [96], bit-plane randomness [7], ability of a training set to represent the input image [22], and other features.

3. By the noise model. Although AWGN is the most widely used noise model, SAR [2, 51], MRI [30, 16, 37, 72, 81], CCD/CMOS [32, 33, 10], and ultrasound/film-grain [86] noise models have been studied as well. Some authors also analyze models not considered in this work [86, 54].

Since the general idea of each method usually does not significantly depend on the noise model, classification by the first two criteria is used further. In Table 2.1, the methods are grouped by their approach and assumptions. As one can see, to utilize homogeneous areas for signal and noise separation is the most widely used technique. In the following sections, each group is considered separately.

Table 2.1: Classification of the existing noise parameter estimation methods by the approach and assumptions.

		Signal and noise separation is applied	
		yes	no
Existance of homogeneous areas is assumed	yes	[77, 80, 1, 87, 52, 15, 9, 75, 95, 55, 76, 4, 89, 35, 3, 59, 2, 51, 30, 16, 37, 72, 81, 32, 33, 10, 86]	–
	no	[83, 24, 38, 65, 21, 93, 7]	[96, 22, 78]

2.1 Methods based on the signal and noise separation and the homogeneity assumption

The methods from this category include one or several common steps, which are described in the following sections.

2.1.1 Preclassification of homogeneous image blocks

The grayvalue distribution in a homogeneous block is defined only by noise, so that the grayvalue variance in a homogeneous block can be used as a noise variance estimate [77, 80, 1, 87]. For this reason, the noise parameter estimation problem reduces to the homogeneous block selection problem.

For the case of AWGN, Fisher's information for the noise standard deviation estimate can be assigned for each block; and then a block is classified as homogeneous if Fisher's information is above a threshold, and as textured otherwise [87]. Although only the homogeneous blocks are used for noise standard deviation estimation, both the noise and texture parameters are necessary to compute Fisher's information. Therefore, the texture parameters are assessed from the textured blocks using the noise standard deviation estimate computed using the homogeneous blocks. Then Fisher's information is recalculated for each block and both sets of the homogeneous and textured blocks are updated. This procedure repeats several times until convergence is reached, as illustrated in Fig. 2.1. Besides, the blocks, whose standard deviation is close to the minimal standard deviation among all blocks, can be considered as homogeneous [80].

For the ultrasound/film-grain noise model (1.31), a similar iterative block classification procedure can be applied [86]. Let $\mu(z)$ and $\sigma^2(z)$ be respectively the block mean and block variance computed for a given block of some image z. Then

$$\sigma^2(y) = \sigma^2(x) + \mu(x^{2b})a^2. \tag{2.1}$$

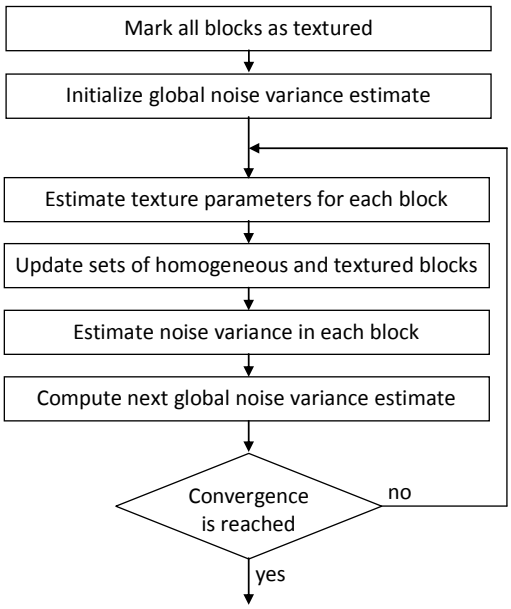

Figure 2.1: Iterative procedure of estimation of the noise and texture parameters used in [87].

Using the Taylor expansion for x^{2b} around $\mu(x^{2b})$

$$x^{2b} \approx \mu(x^{2b}) + \frac{d(x^{2b})}{dx}\Big|_{\mu(x^{2b})}(x - \mu(x^{2b})) + \frac{d^2(x^{2b})}{dx^2}\Big|_{\mu(x^{2b})}\frac{(x - \mu(x^{2b}))^2}{2} \quad (2.2)$$

$\mu(x^{2b})$ can be represented as follows:

$$\mu(x^{2b}) = \mu(x)^{2b} + b(2b-1)\mu(x)^{2b-2}\sigma^2(x). \quad (2.3)$$

Substituting it into (2.1), we have

$$\sigma^2(y) = \sigma^2(x)\big(1 + b(2b-1)\mu(x)^{2b-2}a^2\big) + \mu(x)^{2b}a^2. \quad (2.4)$$

Then, the block, for which the term proportional to $\sigma^2(x)$ is negligible compared with the term $\mu(x)^{2b}a^2$, is classified as homogeneous. Specifically, the following condition is tested:

$$\frac{\mu(x)^{2b}a^2}{\sigma^2(x)\big(1 + b(2b-1)\mu(x)^{2b-2}a^2\big)} > T \quad (2.5)$$

where T is a fixed threshold. This expression can be computed if some estimate of a is known. On the other hand, one should know which blocks are homogeneous in order to estimate a. Because of this, estimation of a and block classification are performed in a loop.

2.1.2 Image filtering

In order to remove low-frequency structures, such as gradual intensity changes, from the input image, a high-pass linear filter (e.g. Laplacian kernel) is applied [9, 15, 30]. Alternatively, the difference between the input image and the response of a low-pass linear filter is computed [80, 52, 76]. Since the filtering result contains the noise as well as object edges, an edge detection algorithm is applied [95, 16] and the pixels, which are not classified as edges, are utilized for noise parameter estimation.

Non-linear filters such as multiparameter piecewise linear sharpener [75] can be applied for signal and noise separation as well.

2.1.3 Analysis of the block variance distribution

The result of the signal and noise separation is often not perfect, therefore the distribution of block variance estimates contains outliers. Thereby, robust statistical methods insensitive to outliers are applied in order to compute the final noise parameter estimates.

For the case of AWGN, one can use the median of block estimates [15], the mode of block estimates [3], and the average of several smallest block estimates [89, 35].

The case of signal-dependent noise is more complex. The block standard deviation is a function of the block mean, hence one cannot consider only the distribution of the block standard deviation. Instead, the scatter-plot for the block mean and block standard deviation should be investigated; and robust fitting of the noise model to the block estimates should be performed. Since the least squares solution is sensitive to outliers, other fitting methods have been considered.

In [51], where the SAR noise model (1.10) is used, the problem is solved from the geometrical point of view. One has to estimate parameter a, which is the ratio of the block standard deviation to the block mean, i.e. it is the slope of the lines connecting the scatter-plot points with the origin. The algorithm counts the scatter-plot points inside circular sectors with a fixed central angle and different slopes (see Fig. 2.2); and the slope of the sector containing the maximal number of points is taken as an estimate of parameter a.

In [32], where the CCD/CMOS noise model (1.29) is considered, the maximum-likelihood approach is used for the fitting. Expected value estimates y_i and standard deviation estimates σ_i are computed for level-sets of the input image ($i = 1, \ldots, N$). If the expected value of y_i is denoted by x_i, the variances of y_i and σ_i can be written as

$$\mathrm{var}(y_i) = h^2(x_i; a, b)c_i; \quad \mathrm{var}(\sigma_i) = h^2(x_i; a, b)d_i \tag{2.6}$$

where $h(t; a, b) = \sqrt{at + b}$ is the standard deviation function of the noise model and factors c_i and d_i are independent of a and b. The distribution of y_i is Gaussian; and σ_i have a scaled noncentral chi-distribution, which can be approximated by Gaussian. Consequently, the conditional probability density functions

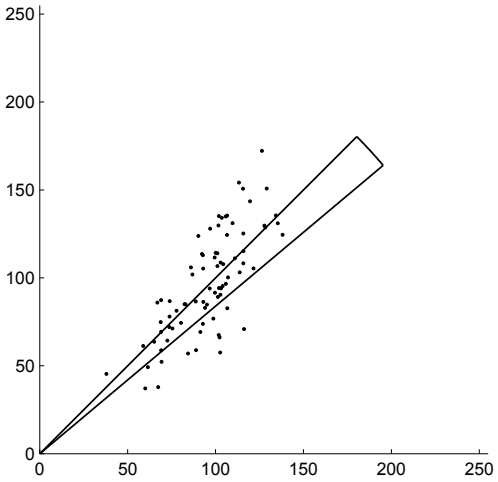

Figure 2.2: Scatter-plot for the block mean and block standard deviation for the SAR noise model and the circular sector, inside which the points are counted [51].

of y_i and σ_i have the following form:

$$f_{y_i}(t|x_i = x) = \frac{1}{\sqrt{2\pi c_i}h(x;a,b)} \exp\Big(-\frac{(t-x)^2}{2h^2(x;a,b)c_i}\Big)$$
$$f_{\sigma_i}(s|x_i = x) = \frac{1}{\sqrt{2\pi d_i}h(x;a,b)} \exp\Big(-\frac{(s-h(x;a,b))^2}{2h^2(x;a,b)d_i}\Big). \tag{2.7}$$

Since y_i and σ_i are mutually independent, the joint conditional probability density function equals the product of the marginal probability density functions:

$$\begin{aligned} f_{y_i,\sigma_i}(t,s|x_i = x) &= f_{y_i}(t|x_i = x) f_{\sigma_i}(s|x_i = x) \\ &= \frac{\exp\Big(-\frac{c_i^{-1}(t-x)^2 + d_i^{-1}(s-h(x;a,b))^2}{2h^2(x;a,b)}\Big)}{2\pi\sqrt{c_i d_i}h^2(x;a,b)}. \end{aligned} \tag{2.8}$$

Posterior likelihood L is

$$L(a,b) = \prod_{i=1}^{N} \int_{-\infty}^{+\infty} f_{y_i,\sigma_i}(t,s|x_i = x) f(x) dx. \tag{2.9}$$

Prior density $f(x)$ can be chosen to be uniform on the grayvalue range. As a

result, parameter estimates a_{est} and b_{est} can be found as

$$\begin{aligned}(a_{est}, b_{est}) &= \arg\max_{a,b} L(a,b) \\ &= \arg\max_{a,b} \sum_{i=1}^{N} \ln \int_{-\infty}^{+\infty} f_{y_i,\sigma_i}(t,s|x_i = x) f(x) dx \end{aligned} \tag{2.10}$$

This maximization problem can be solved numerically taking the least squares solution as the initial estimate.

2.1.4 Analysis of the grayvalue distribution

If the greater part of the input data contains no signal, noise variance can be estimated directly from the grayvalue distribution [59]. Let $(y_1, \ldots, y_N)$ be the input image grayvalues sorted in ascending order. Then, one can find the shortest subsample $(y_{m+k-1}, \ldots, y_m)$ of size k:

$$m = \arg\min_{m}(y_{m+k-1} - y_m). \tag{2.11}$$

Let $r = k/N$ be the fractional subsample size. For the case of AWGN,

$$r = \frac{N_{eff}}{N} \mathrm{erf}\Big(\frac{y_{m+k-1} - y_m}{2\sigma}\Big) \tag{2.12}$$

where N_{eff} is the number of input pixels not affected by the signal, erf is the error function, and σ is the noise standard deviation. This equation has two unknowns: N_{eff} and σ. Therefore, taking two different r, one gets a system of two equations with two unknowns; and σ is computed as the solution of this system. Compared with the median absolute deviation (MAD) estimate, the noise standard deviation estimate obtained using this method is more robust to outliers.

2.1.5 Techniques specific to MRI noise

As pointed out in Section 1.3, the noise in MR images can be approximated by AWGN for large grayvalues. However, this approximation can be too rough for accurate noise level estimation so that the properties of Rice distribution should be taken into account.

In [30], a VST is utilized in order to transform the input image into an image corrupted with AWGN. Since the parameters of the VST depend on the noise level, the author arrives at an iterative algorithm, in which variance stabilization and noise estimation are performed several times in a loop.

The authors of [16] apply the correction procedure [46] to the MAD estimate, for which AWGN is assumed. Correction factor $k(\theta)$ is defined as

$$k(\theta) = 2 + \theta^2 - \frac{\pi}{8} \exp\Big(-\frac{\theta^2}{2}\Big)\Big((2+\theta^2) I_0(\theta^2/4) + \theta^2 I_1(\theta^2/4)\Big)^2 \tag{2.13}$$

where θ is computed by the following iterative procedure:

$$\theta_i = \sqrt{k(\theta_{i-1})\Big(1 + \frac{\overline{m}}{a_{MAD}}\Big) - 2} \tag{2.14}$$

where $\overline{m}$ is the signal mean in the object and a_{MAD} is an estimate of the noise standard deviation computed in the object using the MAD. Finally, the correction factor from the last iteration $k(\theta')$ is applied to estimate the parameter a of Rician noise:

$$a_{est} = \frac{a_{MAD}}{\sqrt{k(\theta')}}. \tag{2.15}$$

In [72], maximum likelihood estimation is utilized. Let $y_1, \ldots, y_N$ be gray-values taken from a small area of the input MR image. Since the image is piecewise constant, we can assume that the expected value x is the same for all y_i. Due to the fact that the noise is independent, the joint probability density function of $y_1, \ldots, y_N$ is the product of the probability density functions (1.13):

$$f_{y_1,\ldots,y_N}(t_1, \ldots, t_N) = \prod_{i=1}^{N} \frac{t_i}{a^2} \exp\Big(-\frac{t_i^2 + x^2}{2a^2}\Big) I_0\Big(\frac{t_i x}{a^2}\Big). \tag{2.16}$$

In order to get the likelihood function $L(x, a)$, one substitutes the observations $y_1, \ldots, y_N$ into the right side of the equation above. Then

$$\ln L(x, a) = \sum_{i=1}^{N} \Big(\ln\Big(\frac{y_i}{a^2}\Big) - \frac{y_i^2 + x^2}{2a^2} + \ln I_0\Big(\frac{y_i x}{a^2}\Big)\Big). \tag{2.17}$$

and the maximum likelihood estimate is

$$(x_{est}, a_{est}) = \arg\max_{x,a} \ln L(x, a). \tag{2.18}$$

Taking the mode of the maximum likelihood estimates a_{est} computed for the neighborhood of each pixel, one obtains a robust estimate of parameter a, which is valid if the assumption $x = \mathbf{E}(y_i)\ \forall i$ holds for the majority of the neighborhoods.

Besides, the noise level can be estimated from the image background, which contains no signal and covers a significant area in some MR images [81].

2.2 Methods based on the signal and noise separation without homogeneity assumption

All methods described in this section consider only AWGN.

Because of the fact that the wavelet coefficients at the finest decomposition level (subband HH_1) correspond almost only to the noise, the noise standard deviation can be estimated as the standard deviation of these coefficients [24,

38, 16]. In order to make the estimate robust against outliers caused by high-frequency structures such as edges, the MAD is utilized instead of the sample standard deviation, hence the noise level estimate is computed as

$$\sigma_{est} = 1.4826 \cdot \text{median}(|t_i|) \quad t_i \in HH_1. \tag{2.19}$$

In [83], it is assumed that only wavelet coefficients with the absolute value smaller than some threshold are caused by the noise. This method is iterative: at each step, the noise variance estimate is found using the threshold proportional to the previous noise variance estimate.

The authors of [93] suggest to estimate the noise level using the derivative orthogonal to the image gradient. Let $\mathbf{n}_x$ be a vector orthogonal to the original image gradient in point $\mathbf{p}$: $\mathbf{n}_x \perp \text{grad } x(\mathbf{p})$. If the level set curvature in point $\mathbf{p}$ is small or if the level set passing through point $\mathbf{p}$ is symmetric about grad $x(\mathbf{p})$ in a neighborhood of $\mathbf{p}$, then the original image derivative $\partial x(\mathbf{p})/\partial \mathbf{n}_x$ in direction $\mathbf{n}_x$ computed by the convolution with kernel k is zero. Consecutively, the input image derivative $\partial y(\mathbf{p})/\partial \mathbf{n}_y$ in direction $\mathbf{n}_y \perp \text{grad } y(\mathbf{p})$ is not affected by the original image and has the normal distribution $\mathcal{N}(0, \|k\|^2\sigma^2)$, where $\|k\|$ is the norm of the convolution kernel coefficient vector and σ^2 is the noise variance. Hence one can make a sample from $\mathcal{N}(0, \|k\|^2\sigma^2)$ by calculating $\partial y(\mathbf{p})/\partial \mathbf{n}_y$ for different points $\mathbf{p}$. Since $\|k\|^2$ is known, σ^2 is then easily estimated from this sample.

In [7], a measure of bit-plane randomness is used for noise level estimation. Let the input image $y(\mathbf{p})$ be digitized with N_{bit} bits of accuracy and let $y(\mathbf{p}, k)$ be the kth bit of pixel $y(\mathbf{p})$ (the bits are counted starting from the least significant bit). Image $y(\mathbf{p}, k)$ is referred to as the kth bit-plane of image $y(\mathbf{p})$. It is supposed that if the noise standard deviation is greater than 2^k, bit-plane $y(\mathbf{p}, k)$ is dominated by noise and looks like salt and pepper noise. In order to assess bit-plane randomness, the differences between pixel $\mathbf{p}$ and its neighbors are computed:

$$\begin{aligned}
\Delta_1(\mathbf{p}, k) &= |y((p_1 + 1, p_2), k) - y((p_1, p_2), k)| \\
\Delta_2(\mathbf{p}, k) &= |y((p_1, p_2 + 1), k) - y((p_1, p_2), k)| \\
\Delta_3(\mathbf{p}, k) &= |y((p_1 + 1, p_2 + 1), k) - y((p_1, p_2), k)| \\
\Delta_4(\mathbf{p}, k) &= |y((p_1 - 1, p_2), k) - y((p_1, p_2), k)|.
\end{aligned} \tag{2.20}$$

If bit-plane $y(\mathbf{p}, k)$ is affected only by noise, the mean of $\Delta_i(\mathbf{p}, k)$ is $1/2$. Otherwise, it is less than $1/2$. Therefore, denoting the average of $\Delta_i(\mathbf{p}, k)$ over $\mathbf{p}$ and i by $\overline{\Delta}$, bit-plane randomness can be represented by $\delta(k) = |1/2 - \overline{\Delta}|$. If $y(\mathbf{p}, k)$ contains only noise, the mean of $\delta(k)$ is zero and the standard deviation of $\delta(k)$ is $4/\sqrt{S_1 S_2}$. Hence bit-plane $y(\mathbf{p}, k)$ is labeled as "noisy", if $\delta(k) < 4m/\sqrt{S_1 S_2}$, where m is a parameter. In order to obtain the final noise standard deviation estimate, $\delta(k)$ for different bit-planes are combined:

$$\sigma_{est} = 0.2 \sum_{k=k_0}^{k_0+3} (2^k - 1)(1 - 2\delta(k))^4 \tag{2.21}$$

where k_0 is the maximal k, for which $y(\mathbf{p}, k)$ is labeled as "noisy".

The method [65] is based on 2D DCT of image blocks. First, the input image is divided into N 8×8 blocks and 2D DCT is applied for each block. This results in a set of N 8×8 arrays of the DCT coefficients $\{D_{ijm}\}$, where $i, j = 1, \ldots, 8$ are the DCT coefficient indices and $m = 1, \ldots, N$ is the block index. A DCT coefficient with indices (i, j) belongs to the low frequency component if $2 < i + j < T$ and to the high frequency component if $i + j \geq T$. The recommended value of T is 11. Then, the variance of the low frequency coefficients is computed for each block:

$$V_m^{(L)} = \frac{1}{k} \sum_{2<i+j<T} D_{ijm}^2 \tag{2.22}$$

where k is the number of low frequency indices. Next, the blocks are sorted in ascending order of $V_m^{(L)}$, which yields tuple $(D_{ij\,\pi(1)}, \ldots, D_{ij\,\pi(N)})$. The first $K < N$ blocks $(D_{ij\,\pi(1)}, \ldots, D_{ij\,\pi(K)})$ correspond to locally passive image regions so that their high frequency components are utilized for noise variance estimation. For each high frequency coefficient, the noise variance estimate is calculated as

$$V_{ij}^{(H)} = \frac{1}{K} \sum_{m=1}^{K} D_{ij\,\pi(m)}^2 \quad i + j \geq T. \tag{2.23}$$

The final noise variance estimate is computed as the median of $\{V_{ij}^{(H)} | i+j \geq T\}$.

In [21], image self-similarity is utilized for noise level estimation. The input image is processed in sliding-block manner. For each block, similar blocks are found and stacked together so that they form a 3D array called group. Due to the fact that the groups are constructed from mutually nonoverlapping blocks, the noise in each group is independent. On the other hand, the original pixel values in each group correlate between each other because

1. neighboring image pixels are correlated;

2. blocks in one group are similar to each other by construction.

For this reason, the original signal in a group can be represented by a small number of coefficients after applying a 3D decorrelating transform to this group. Specifically, if 3D DCT is applied, the original image defines only the low frequency coefficients and the high frequency coefficients are affected only by the noise. Hence the high frequency coefficients of each group are collected into a sample, which is then utilized for noise variance estimation.

2.3 Methods without signal and noise separation

The algorithms described in this section do not construct a sample containing only noise and assess the noise level using statistics computed for this sample. Instead, the noise level is estimated indirectly.

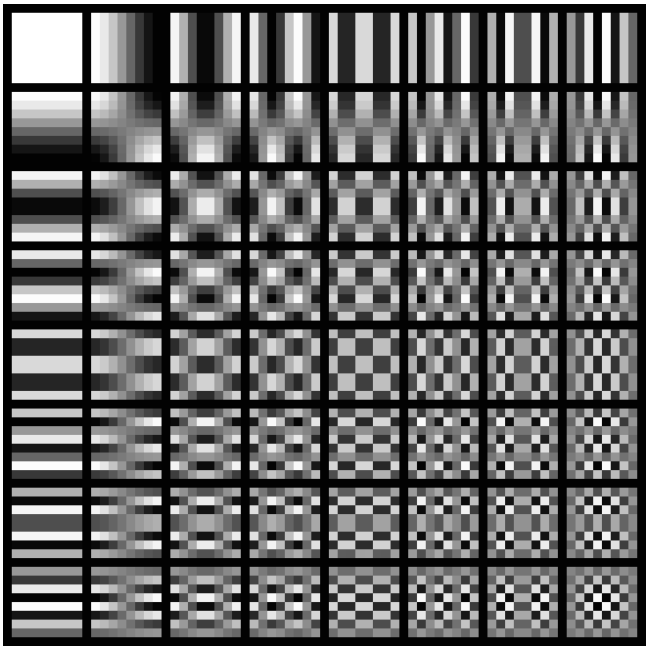

Figure 2.3: 8×8 DCT basis functions.

Like in the previous section, only AWGN is considered.

The method [96] is based on the assumption that the kurtosis of marginal bandpass filter response distributions should be constant for the original image. Let us consider 8×8 DCT basis functions shown in Fig. 2.3 (the basis functions are indexed from 1 to 64 from top to bottom and from left to right). Except for the DC element (in the top-left corner), which is a low-pass filter, all elements of the basis represent bandpass filters of different scales. Let images x_i, n_i, and y_i be, respectively, the results of filtering images x, n, and y with ith basis function ($i = 2, \ldots, 64$). $y_i = x_i + \sigma n_i$, because the filtering operation is linear. It is supposed that the kurtosis of the grayvalue distribution of x_i is independent of the scale, i.e. independent of i. Due to the independence of x_i and n_i,

$$\gamma_{y,i} = \gamma_x \Big/ \Big(1 + \frac{\sigma^2}{\sigma_{y,i}^2 - \sigma^2}\Big)^2 \quad i = 2, \ldots, 64. \tag{2.24}$$

where $\sigma_{y,i}^2$ is the variance of y_i, γ_x is the excess kurtosis of x_i, which is independent of i according to the assumption, and $\gamma_{y,i}$ is the excess kurtosis of y_i. (2.24) is a system with 63 equations and two unknowns: σ and γ_x. It is solved by minimizing the residual:

$$(\sigma_{est}, \gamma_{x,est}) = \arg\min_{\sigma,\gamma_x} \sum_{i=2}^{64} \Big| \gamma_{y,i} - \gamma_{x,i} \Big/ \Big(1 + \frac{\sigma^2}{\sigma_{y,i}^2 - \sigma^2}\Big)^2 \Big|. \tag{2.25}$$

Finally, σ_{est} is taken as a noise standard deviation estimate.

Training methods are considered in [22]. Particularly, the cumulative distribution function (CDF) $c(t)$ of the variances of blocks in the wavelet subband HH_1 is analyzed. In order to make the algorithm computationally efficient, only the value of the CDF at point $t = t_0$ is measured. The mean values of $c(t_0)$ are computed for a training set of images for different noise variances; and a lookup table of $c(t_0)$ against the noise variance is stored. For a new noisy image, the value of $c(t_0)$ is computed and the noise variance is estimated using the lookup table. Point t_0 is selected in such a way that it maximizes a discrimination metric evaluated for the training set:

$$t_0 = \arg\max_t \sum_{i<j} \frac{|m_i(t) - m_j(t)|}{\sqrt{\sigma_i(t) + \sigma_j(t)}} \tag{2.26}$$

where the summation is taken over all considered noise levels l_i, $m_i(t)$ and $\sigma_i(t)$ are, respectively, the mean and the standard deviation of $c(t)$ for the training set for noise level l_i.

In [78], a Bayesian framework for simultaneous deblurring and noise level estimation was proposed. A learned Markov random field prior (Fields of Experts [74]) is used to compute probability density function $f_x(u)$ of original image x; and the noise standard deviation is estimated as

$$\begin{aligned} \sigma_{est} &= \arg\min_a \iint (v - \sigma)^2 f_{x,\sigma}(u, v | y = y_0, K = K_0) du dv \\ &= \mathbf{E}(\sigma | y = y_0, K = K_0) \end{aligned} \tag{2.27}$$

where K_0 is the blur matrix and $y_0 = K_0 x + \sigma n$ is the input image.

2.4 Summary

To utilize homogeneous areas is the simplest and the fasted yet the most restrictive approach to noise level estimation. It is the only method used to estimate the parameters of signal-dependent noise; and if the noise model has several parameters, an even stronger condition is required: homogeneous areas should have different mean intensities, because the dependence of the noise variance on the original intensity should be estimated.

The algorithms, in which signal and noise are separated without the homogeneity assumption, can process a larger class of images, but irregular textures are still a problem, because they cannot be sparsely represented by the transforms utilized in these algorithms. The execution time depends on the transform and can be significantly larger than that of the methods using the image homogeneity. For example, the method [21] includes the search of blocks similar to a reference block, which is done by enumeration of all blocks in the neighborhood of the reference block.

As we will see in the results of the experiments, the approaches, which does not perform signal and noise separation, are in practice not more accurate than

the other methods, although their assumptions about the original image can look weaker. This can be due to the fact that these assumptions are heuristic and they are not satisfied for some images, which can be processed by the methods based on the signal and noise separation.

Chapter 3

Signal-independent noise parameter estimation

In this chapter, the proposed noise parameter estimation framework is described for the case of AWGN (1.1). This noise model contains only one parameter – noise standard deviation σ.

Because of the fact that the proposed method extensively uses properties of the sample covariance matrix eigenvalues, which cannot be illustrated directly, the chapter starts with an explanation of the basic idea on a simple example in Section 3.1. The construction of image blocks is presented in Section 3.2. Population and sample principal component analysis are described in Sections 3.3 and 3.4, which contain all necessary theoretical results and form the basis for two algorithms, which are expounded in Sections 3.5 and 3.6. The experimental results are presented in Section 3.7. The discussion in Section 3.8 and the summary in Section 3.9 conclude the chapter.

3.1 Idea of the method

This section presents a simple 1D example, which demonstrates the ability of image block PCA to estimate the noise variance. Consider original signal $(x_k) = ((-1)^k) = (-1, 1, -1, 1, \ldots)$ and noisy signal $(y_k) = (x_k + \sigma n_k)$, where $\sigma = 1/8$ and n_k are realizations of a random variable with standard normal distribution $\mathcal{N}(0; 1)$. Let us process these signals using a sliding window of size 2. Then, we get two point sets:

1. $\{\mathbf{x}_k\} = \{(x_k, x_{k+1})^{\mathrm{T}}\}$ for the original signal;
2. $\{\mathbf{y}_k\} = \{(y_k, y_{k+1})^{\mathrm{T}}\} = \{(x_k, x_{k+1})^{\mathrm{T}} + (\sigma n_k, \sigma n_{k+1})^{\mathrm{T}}\}$ for the noisy signal.

By construction, points $\mathbf{x}_k$ can have only two values: $(-1; 1)^{\mathrm{T}}$ and $(1; -1)^{\mathrm{T}}$. Points $(\sigma n_k; \sigma n_{k+1})^{\mathrm{T}}$ are realizations of the bivariate normal distribution with

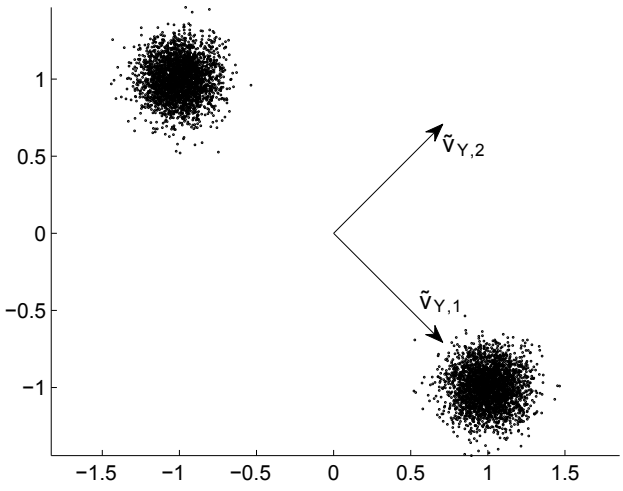

Figure 3.1: Vectors $\mathbf{y}_k$ (dots) and vectors $\tilde{\mathbf{v}}_{\mathbf{Y},1}$, $\tilde{\mathbf{v}}_{\mathbf{Y},2}$ (arrows).

zero mean and scalar covariance matrix

$$\begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}. \tag{3.1}$$

The level sets of the probability density function of this distribution are cycles centered in the origin. Therefore, set $\{(\sigma n_k; \sigma n_{k+1})^{\mathrm{T}}\}$ is a symmetric point cloud centered in the origin; and set $\{\mathbf{y}_k\}$ contains two symmetric point clouds centered in $\mathbf{x}_k$, namely in $(-1;1)^{\mathrm{T}}$ and $(1;-1)^{\mathrm{T}}$. Points $\mathbf{y}_k$ are presented in Fig. 3.1.

Let us now apply PCA to set $\{\mathbf{y}_k\}$. In this 2D case, it is equivalent to finding direction $\tilde{\mathbf{v}}_{Y,1}$, along which the sample variance of $\{\mathbf{y}_k\}$ is maximal:

$$\tilde{\mathbf{v}}_{Y,1} = \arg \max_{\|\mathbf{v}\|=1} s^2(\{\mathbf{v}^{\mathrm{T}}\mathbf{y}_k\}) \tag{3.2}$$

where $s^2(\cdot)$ is the sample variance. Obviously, if $\tilde{\mathbf{v}}_{Y,1}$ is a solution of this problem, then $(-\tilde{\mathbf{v}}_{Y,1})$ is a solution of this problem as well, but it is enough to consider only one of these solutions. As it can be seen from Fig. 3.1, $\tilde{\mathbf{v}}_{Y,1}$ tends to $\left(1/\sqrt{2}, -1/\sqrt{2}\right)^{\mathrm{T}}$ as the number of points $\{\mathbf{y}_k\}$ tends to infinity, hence let us assume that $\tilde{\mathbf{v}}_{Y,1} = \left(1/\sqrt{2}, -1/\sqrt{2}\right)^{\mathrm{T}}$. Then, unit vector $\tilde{\mathbf{v}}_{Y,2}$ orthogonal to $\tilde{\mathbf{v}}_{Y,1}$ can be chosen, for example $\tilde{\mathbf{v}}_{Y,2} = \left(1/\sqrt{2}, 1/\sqrt{2}\right)^{\mathrm{T}}$ (see Fig. 3.1). As a result, vectors $\{\tilde{\mathbf{v}}_{Y,1}, \tilde{\mathbf{v}}_{Y,2}\}$ form a basis.

The first coordinate of vectors $\mathbf{y}_k$ in this basis is calculated as follows:

$$\begin{aligned}\tilde{\mathbf{v}}_{Y,1}^{\mathrm{T}}\mathbf{y}_k &= \tilde{\mathbf{v}}_{Y,1}^{\mathrm{T}}\mathbf{x}_k + \tilde{\mathbf{v}}_{Y,1}^{\mathrm{T}}(\sigma n_k, \sigma n_{k+1})^{\mathrm{T}} = \frac{x_k - x_{k+1}}{\sqrt{2}} + \tilde{\mathbf{v}}_{Y,1}^{\mathrm{T}}(\sigma n_k, \sigma n_{k+1})^{\mathrm{T}} \\ &= \begin{cases} -\sqrt{2} + \tilde{\mathbf{v}}_{Y,1}^{\mathrm{T}}(\sigma n_k, \sigma n_{k+1})^{\mathrm{T}}, & \text{if } \mathbf{x}_k = (-1,1)^{\mathrm{T}} \\ \sqrt{2} + \tilde{\mathbf{v}}_{Y,1}^{\mathrm{T}}(\sigma n_k, \sigma n_{k+1})^{\mathrm{T}}, & \text{if } \mathbf{x}_k = (1,-1)^{\mathrm{T}} \end{cases}. \end{aligned} \tag{3.3}$$

Therefore, the first coordinate is affected by both original signal (x_k) and noise (n_k), which can be also seen in Fig. 3.1: vector $\tilde{\mathbf{v}}_{Y,1}$ is parallel to the line containing points $\{\mathbf{x}_k\}$ so that these points can be represented by their projection onto $\tilde{\mathbf{v}}_{Y,1}$.

The second coordinate of vectors $\mathbf{y}_k$ in basis $\{\tilde{\mathbf{v}}_{Y,1}, \tilde{\mathbf{v}}_{Y,2}\}$ is

$$\begin{aligned}\tilde{\mathbf{v}}_{Y,2}^{\mathrm{T}}\mathbf{y}_k &= \tilde{\mathbf{v}}_{Y,2}^{\mathrm{T}}\mathbf{x}_k + \tilde{\mathbf{v}}_{Y,2}^{\mathrm{T}}(\sigma n_k, \sigma n_{k+1})^{\mathrm{T}} = \frac{x_k + x_{k+1}}{\sqrt{2}} + \sigma\frac{n_k + n_{k+1}}{\sqrt{2}} \\ &= \sigma\frac{n_k + n_{k+1}}{\sqrt{2}} \quad \forall k. \end{aligned} \tag{3.4}$$

Consequently, the second coordinate of $\mathbf{y}_k$ is independent of original signal (x_k). In Fig. 3.1, this is represented by the fact that direction $\tilde{\mathbf{v}}_{Y,2}$ is orthogonal to the line containing points $\{\mathbf{x}_k\}$ so that the projections of $\mathbf{y}_i$ onto $\tilde{\mathbf{v}}_{Y,2}$ are not affected by $\mathbf{x}_k$. Furthermore, the distribution of the second coordinate of $\mathbf{y}_k$ is $\mathcal{N}(0;\sigma^2)$, i.e. the same as that of σn_k, because $\tilde{\mathbf{v}}_{Y,2}$ has unit length. As a result, we can estimate the noise standard deviation as the sample standard deviation of $\tilde{\mathbf{v}}_{Y,2}^{\mathrm{T}}\mathbf{y}_k$:

$$\sigma_{est} = s(\{\tilde{\mathbf{v}}_{Y,2}^{\mathrm{T}}\mathbf{y}_k\}). \tag{3.5}$$

This example shows some properties of the proposed method:

1. The method can be applied if the blocks computed from the original signal can be represented by a number of dimensions smaller than the block size. In the example above, 2-dimensional points $\mathbf{x}_k$ have coordinates $(-\sqrt{2};0)^{\mathrm{T}}$ and $(\sqrt{2};0)^{\mathrm{T}}$ in the new basis. Therefore, they can be represented only by the fist coordinate in the new basis.

2. If the blocks computed from the original signal cannot be represented by a number of dimensions smaller than the block size, we cannot apply PCA directly in order to get the noise variance. In the example above, if the block set had three centroids, which did not lie on one line, then PCA would not provide a coordinate associated only with the noise.

3. No assumption about signal constancy is required. Indeed, in the example above, original signal (x_k) contains no constant parts. In the case of 2D and 3D images, this property means that the existence of homogeneous areas is not required.

3.2 Image block model

Each of images x, n, and y contains

$$N = (S_1 - M_1 + 1) \cdot \ldots \cdot (S_d - M_d + 1) \tag{3.6}$$

blocks of size $M_1 \times \cdots \times M_d$, whose left-top-front corner positions are taken from set

$$\{1, \ldots, S_1 - M_1 + 1\} \times \cdots \times \{1, \ldots, S_d - M_d + 1\}. \tag{3.7}$$

These blocks can be rearranged into vectors $\mathbf{x}_i$, $\mathbf{n}_i$, and $\mathbf{y}_i$ with $M = M_1 \cdot \ldots \cdot M_d$ elements.

Vectors $\mathbf{n}_i$ and $\mathbf{y}_i$ are considered as realizations of random vectors $\mathbf{N}$ and $\mathbf{Y}$ respectively. Since $n(\mathbf{p}) \sim \mathcal{N}(0,1)$, $\mathbf{N} \sim \mathcal{N}_M(0, I_M)$. Although original image x is not random, vectors $\mathbf{x}_i$ are considered as realizations of some random vector $\mathbf{X}$ in this chapter. Due to the fact that the noise is signal-independent, $\operatorname{cov}(\mathbf{X}, \mathbf{N}) = 0$.

3.3 Population principal component analysis

Let $\mathbf{\Sigma_X}$ and $\mathbf{\Sigma_Y}$ be the population covariance matrices of $\mathbf{X}$ and $\mathbf{Y}$ respectively, and $\lambda_{\mathbf{X},1} \geq \cdots \geq \lambda_{\mathbf{X},M}$ be the eigenvalues of $\mathbf{\Sigma_X}$ with corresponding normalized eigenvectors $\mathbf{v}_{\mathbf{X},1}, \ldots, \mathbf{v}_{\mathbf{X},M}$. Let $q_1, \ldots, q_p$ be the multiplicities of the eigenvalues of $\mathbf{\Sigma_X}$ and

$$\begin{aligned}
\delta_{\mathbf{X},1} &= \lambda_{\mathbf{X},1} = \cdots = \lambda_{\mathbf{X},q_1} \\
\delta_{\mathbf{X},2} &= \lambda_{\mathbf{X},q_1+1} = \cdots = \lambda_{\mathbf{X},q_1+q_2} \\
&\vdots \\
\delta_{\mathbf{X},p} &= \lambda_{\mathbf{X},M-q_p+1} = \cdots = \lambda_{\mathbf{X},M}
\end{aligned}$$

where $\delta_{\mathbf{X},1} > \cdots > \delta_{\mathbf{X},p}$. $\operatorname{cov}(\mathbf{X}, \mathbf{N}) = 0$ implies that $\mathbf{\Sigma_Y} = \mathbf{\Sigma_X} + \sigma^2 I_M$. For $i = 1, \ldots, M$,

$$\mathbf{\Sigma_Y}\mathbf{v}_{\mathbf{X},i} = \mathbf{\Sigma_X}\mathbf{v}_{\mathbf{X},i} + \sigma^2\mathbf{v}_{\mathbf{X},i} = \lambda_{\mathbf{X},i}\mathbf{v}_{\mathbf{X},i} + \sigma^2\mathbf{v}_{\mathbf{X},i} = (\lambda_{\mathbf{X},i} + \sigma^2)\mathbf{v}_{\mathbf{X},i}. \tag{3.8}$$

Hence $\lambda_{\mathbf{Y},i} = \lambda_{\mathbf{X},i} + \sigma^2$ are the eigenvalues of $\mathbf{\Sigma_Y}$ with corresponding eigenvectors $\mathbf{v}_{\mathbf{X},i}$, $i = 1, \ldots, M$. The multiplicities of the eigenvalues of $\mathbf{\Sigma_X}$ and $\mathbf{\Sigma_Y}$ are the same; and the distinct values of the eigenvalues of $\mathbf{\Sigma_Y}$ are denoted by $\delta_{\mathbf{Y},1}, \ldots, \delta_{\mathbf{Y},p}$ so that

$$\delta_{\mathbf{Y},i} = \delta_{\mathbf{X},i} + \sigma^2, \quad i = 1, \ldots, p. \tag{3.9}$$

PCA, as a data analysis method, is based on the following properties of the eigenvalues and eigenvectors of $\mathbf{\Sigma_Y}$ [44]:

1. $\mathbf{v}_{\mathbf{X},1}$ is the direction, along which the variance of $\mathbf{Y}$ is maximal:

$$\mathbf{v}_{\mathbf{X},1} = \arg\max\{\operatorname{var}(\mathbf{v}^{\mathrm{T}}\mathbf{Y}) \mid \|\mathbf{v}\| = 1\}. \tag{3.10}$$

   Besides, $\operatorname{var}(\mathbf{v}_{\mathbf{X},1}^{\mathrm{T}}\mathbf{Y}) = \lambda_{\mathbf{Y},1}$.

2. $\mathbf{v}_{\mathbf{X},i}$, $i \geq 2$ is the direction orthogonal to $\mathbf{v}_{\mathbf{X},1}, \ldots, \mathbf{v}_{\mathbf{X},i-1}$, along which the variance of $\mathbf{Y}$ is maximal compared with the variance of $\mathbf{Y}$ along all other directions orthogonal to $\mathbf{v}_{\mathbf{X},1}, \ldots, \mathbf{v}_{\mathbf{X},i-1}$:

$$\mathbf{v}_{\mathbf{X},i} = \arg\max\{\mathrm{var}(\mathbf{v}^{\mathrm{T}}\mathbf{Y}) \mid \|\mathbf{v}\| = 1, \mathbf{v} \perp \mathbf{v}_{\mathbf{X},k}, k = 1, \ldots, i-1\}. \quad (3.11)$$

Again, $\mathrm{var}(\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}}\mathbf{Y}) = \lambda_{\mathbf{Y},i}$.

Random variables $\mathbf{v}_{\mathbf{X},i}^{\mathrm{T}}\mathbf{Y}$ are called the population principal components of $\mathbf{Y}$.

In order to develop the method further, let us consider a class of original images, for which PCA can be applied for noise variance estimation. Such images satisfy the following assumption:

Assumption 1. *Let $m > 1$ be a predefined integer number. The information in original image x is redundant in the sense that all $\mathbf{x}_i$ lie in subspace $V_{M-m} \subset \mathbb{R}^M$, whose dimension $M - m$ is smaller than the number of coordinates M.*

When this assumption holds, it is considered that random vector $\mathbf{X}$ takes its values almost surely only in subspace V_{M-m}. Then, this assumption can be reformulated in the following equivalent forms:

1. $\delta_{\mathbf{X},p} = 0$ and $q_p \geq m$;
2. $\mathbf{X}$ has zero variance along m mutually orthogonal directions.

Assumption 1 also has the following consequences, which explain its meaning:

1. components of $\mathbf{X}$ are almost surely linearly dependent;
2. pixels of x in the image blocks are linearly dependent.

When $\delta_{\mathbf{X},p} = 0$, $\delta_{\mathbf{Y},p}$ equals σ^2, which provides a way for noise variance estimation by estimating $\delta_{\mathbf{Y},p}$ from the realizations of $\mathbf{Y}$. For this reason, Assumption 1 plays a central role in the proposed algorithm. Tests of this assumption are derived; and techniques for selection image block sets, which satisfy Assumption 1, are given.

3.4 Sample principal component analysis

In practice, one has only realizations of $\mathbf{Y}$; and population covariance matrix $\mathbf{\Sigma}_{\mathbf{Y}}$, as well as its eigenvalues, cannot be computed. Therefore, analysis of the corresponding sample quantities is required. Let $S_{\mathbf{Y}}$ be the sample covariance matrix of $\mathbf{Y}$:

$$S_{\mathbf{Y}} = \frac{1}{N-1}\Big(\sum_{i=1}^{N} \mathbf{y}_i\mathbf{y}_i^{\mathrm{T}} - \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i \sum_{i=1}^{N}\mathbf{y}_i^{\mathrm{T}}\Big). \quad (3.12)$$

and $\tilde{\lambda}_{\mathbf{Y},1} \geq \ldots \geq \tilde{\lambda}_{\mathbf{Y},M}$ be the eigenvalues of $S_{\mathbf{Y}}$ with corresponding normalized eigenvectors $\tilde{\mathbf{v}}_{\mathbf{Y},1}, \ldots, \tilde{\mathbf{v}}_{\mathbf{Y},M}$. Then, $\tilde{\mathbf{v}}_{\mathbf{Y},1}^{\mathrm{T}}\mathbf{Y}, \ldots, \tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{Y}$ represent the sample principal components of $\mathbf{Y}$, which have the property [44]

$$s^2(\tilde{\mathbf{v}}_{\mathbf{Y},i}^{\mathrm{T}}\mathbf{Y}) = \tilde{\lambda}_{\mathbf{Y},i} \quad i = 1, \ldots, M. \quad (3.13)$$

When working with the sample principal components, the following problems should be solved:

1. how to check Assumption 1 using $\tilde{\lambda}_{\mathbf{Y},1}, \ldots, \tilde{\lambda}_{\mathbf{Y},M}$;

2. when Assumption 1 holds, how to estimate q_p – the number of the eigenvalues not affected by the original image;

3. when Assumption 1 holds, how to construct an estimator of noise variance σ^2 using $\tilde{\lambda}_{\mathbf{Y},1}, \ldots, \tilde{\lambda}_{\mathbf{Y},M}$.

There are several solutions and they are described below.

3.4.1 Bartlett's test

In order to check Assumption 1 and estimate q_p, Bartlett's test [8], which tests the equality of several consecutive population eigenvalues, can be utilized.

Let

$$H_{0k} : \lambda_{\mathbf{Y},M-k+1} = \cdots = \lambda_{\mathbf{Y},M} \tag{3.14}$$

be the hypothesis that the last k eigenvalues of $\mathbf{\Sigma_Y}$ are equal, and

$$H_{1k} : \lambda_{\mathbf{Y},M-k+1} > \lambda_{\mathbf{Y},M} \tag{3.15}$$

be the alternative hypothesis. H_{0k} can be tested against H_{1k} using Bartlett's test [8, 44], in which H_{0k} is rejected at significance level α if

$$N'\Big(k \ln\Big(\sum_{i=M-k+1}^{M} \tilde{\lambda}_{\mathbf{Y},i}\Big) - \sum_{i=M-k+1}^{M} \ln(\tilde{\lambda}_{\mathbf{Y},i})\Big) \geq \chi^2_{\nu,\alpha} \tag{3.16}$$

where $N' = N - (2M + 11)/6$, $\chi^2_{\nu,\alpha}$ is the value of the inverse CDF of the chi-squared distribution with ν degrees of freedom at point $1 - \alpha$, and $\nu = (k + 2)(k - 1)/2$.

When Assumption 1 holds, $\lambda_{\mathbf{Y},M-m+1} = \cdots = \lambda_{\mathbf{Y},M}$. Therefore, H_{0m} can be used as a necessary condition for the fulfillment of Assumption 1. This condition is formally not sufficient. However, in practice, nonzero eigenvalues of $\mathbf{\Sigma_X}$ are distinct [44], which can be explained by the fact that vector $\mathbf{X}$ represents image structures, and it is very unlikely to have the same variance in different directions. Hence condition H_{0m} practically means that $\delta_{\mathbf{X},p} = 0$ and (3.16) with $k = m$ is a reliable check of Assumption 1.

Repeating test (3.16) for $k = 2, 3, 4, \ldots$ until H_{0k} is rejected allows estimating q_p as the maximal k, for which H_{0k} is accepted. However, since test (3.16) is repeated more than once if H_{02} is not rejected, the overall significance level of the sequence of the tests is not equal to significance level α of each test. Moreover, the tests are not independent and their number is a random variable so that the overall significance level is unknown [44].

3.4.2 Eigenvalue difference

Another possibility to check Assumption 1 is to look at the difference $(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M})$. The properties of the distribution of this difference can be obtained from the following theorem.

Theorem 1. *If Assumption 1 is satisfied then the following asymptotic bound holds for all $i = M - q_p + 1, \ldots, M$:*

$$\mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2|\big) = O(\sigma^2/\sqrt{N}) \quad N \to \infty \tag{3.17}$$

i.e.

$$\exists C' \;\; \exists N' \;\; \forall N \geq N' \quad \mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2|\big) \leq C'\sigma^2/\sqrt{N} \tag{3.18}$$

where C' does not depend on the distributions of $\mathbf{X}$ and $\mathbf{N}$.

The formal proof is given in the Appendix.

Using the result of Theorem 1, let us construct an asymptotic bound for the probability that the difference $(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M})$ is greater than some threshold $T_\lambda\sigma^2/\sqrt{N}$. From Markov's inequality,

$$P\Big(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M} \geq \frac{T_\lambda\sigma^2}{\sqrt{N}}\Big) \leq \frac{\sqrt{N}}{T_\lambda\sigma^2}\mathbf{E}\big(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M}\big). \tag{3.19}$$

Using the triangle inequality, we have

$$\begin{aligned}\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M} &= |\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M}| \\ &= |\tilde{\lambda}_{\mathbf{Y},M-m+1} - \sigma^2 + \sigma^2 - \tilde{\lambda}_{\mathbf{Y},M}| \\ &\leq |\tilde{\lambda}_{\mathbf{Y},M-m+1} - \sigma^2| + |\sigma^2 - \tilde{\lambda}_{\mathbf{Y},M}|. \end{aligned} \tag{3.20}$$

From monotonicity of the expected value,

$$\mathbf{E}(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M}) \leq \mathbf{E}(|\tilde{\lambda}_{\mathbf{Y},M-m+1} - \sigma^2|) + \mathbf{E}(|\sigma^2 - \tilde{\lambda}_{\mathbf{Y},M}|). \tag{3.21}$$

Therefore, when Assumption 1 holds, for $N \geq N_0$

$$\mathbf{E}(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M}) \leq \frac{2C'\sigma^2}{\sqrt{N}} \tag{3.22}$$

and

$$P\Big(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M} \geq \frac{T_\lambda\sigma^2}{\sqrt{N}}\Big) \leq \frac{2C'}{T_\lambda}. \tag{3.23}$$

As a result, we can select T_λ in such a way that $2C'/T_\lambda$ is small; and a necessary condition for the fulfillment of Assumption 1 can be written as follows:

$$\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M} < T_\lambda\sigma^2/\sqrt{N}. \tag{3.24}$$

A question may arise whether condition (3.24) can be made stronger. When the original image x is zero, the limiting joint probability density function of $(\tilde{\lambda}_{\mathbf{Y},1} - \sigma^2)\sqrt{N}$, ..., $(\tilde{\lambda}_{\mathbf{Y},M} - \sigma^2)\sqrt{N}$ is

$$f(t_1, \ldots, t_M; \sigma) = \frac{K(M)}{\sigma^{M(M+1)}} \exp\Big(- \frac{t_1^2 + \cdots + t_M^2}{4\sigma^4} \Big) \prod_{j<k} (t_j - t_k) \tag{3.25}$$

if $t_1 > \cdots > t_M$ and 0 otherwise [5]. Above,

$$K(M) = 2^{-M(M+3)/4} \prod_{i=1}^{M} \big(\Gamma((M+1-i)/2) \big)^{-1} \tag{3.26}$$

and $\Gamma(\cdot)$ is the gamma function. Substituting $t_1' = t_1/\sigma^2$, ..., $t_M' = t_M/\sigma^2$, we have that

$$\begin{aligned} \mathbf{E}((\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2)\sqrt{N}) &= \int t_i f(t_1, \ldots, t_M; \sigma) dt_1 \ldots dt_M \\ &= \sigma^2 \int t_i' f(t_1', \ldots, t_M'; 1) dt_1' \ldots dt_M' = \sigma^2 C_i \end{aligned} \tag{3.27}$$

where coefficients

$$C_i = \int t_i' f(t_1', \ldots, t_M'; 1) dt_1' \ldots dt_M' \tag{3.28}$$

are independent of σ and N. Hence

$$\mathbf{E}(\tilde{\lambda}_{\mathbf{Y},i}) = \sigma^2 + \sigma^2 C_i/\sqrt{N} \tag{3.29}$$

and

$$\mathbf{E}(\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M}) = \frac{(C_{M-m+1} - C_M)\sigma^2}{\sqrt{N}}. \tag{3.30}$$

As a result, (3.22) is a tight upper bound, and (3.24) cannot be improved by changing the exponents of σ or N.

3.4.3 Estimators of the noise variance

Let

$$\sigma^2_{est,k} = \frac{1}{k} \sum_{i=M-k+1}^{M} \tilde{\lambda}_{\mathbf{Y},i} \tag{3.31}$$

where $k \in \{1, \ldots, q_p\}$. According to Theorem 1, when Assumption 1 is satisfied,

$$\begin{aligned} \mathbf{E}\Big(\big| \sigma^2_{est,k} - \sigma^2 \big| \Big) &= \mathbf{E}\Big(\Big| \frac{1}{k} \sum_{i=M-k+1}^{M} (\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2) \Big| \Big) \\ &\leq \frac{1}{k} \sum_{i=M-k+1}^{M} \mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},i} - \sigma^2| \big) \to 0 \quad N \to +\infty. \end{aligned} \tag{3.32}$$

i.e. $\sigma^2_{est,k}$ converges in mean to σ^2. Therefore, the noise variance can be estimated as $\sigma^2_{est,k}$. Due to the fact that convergence in mean implies convergence in probability, $\sigma^2_{est,k}$ is a consistent estimator of the noise variance for any $k \in \{1, \ldots, q_p\}$. By substituting specific values of k, we can obtain two important special cases:

1. $k = 1$:

$$\sigma^2_{est,k} = \tilde{\lambda}_{\mathbf{Y},M} \tag{3.33}$$

   i.e. the last sample eigenvalue is used as a noise variance estimator.

   The advantage of this estimator can be seen, when the estimate of q_p is larger than its actual value, or when the check of Assumption 1 is satisfied, but Assumption 1 does not actually hold. In these cases, the last q_p eigenvalues of $S_{\mathbf{Y}}$ are affected by the original image and correct estimation of the noise variance cannot be guaranteed. $\tilde{\lambda}_{\mathbf{Y},M}$ is less affected by the original image than the other sample eigenvalues, because the variance of $\mathbf{Y}$ among $\tilde{\mathbf{v}}_{\mathbf{Y},M}$ is the smallest among all directions. Hence its is preferable to use $\tilde{\lambda}_{\mathbf{Y},M}$ in order to minimize the estimation error when Assumption 1 does not hold.

   The disadvantage of this estimator is that it is the smallest order statistic of the sample eigenvalues representing the noise. This can be illustrated on the case when original image x is zero. Then, population variance of the projections of $\mathbf{Y}$ onto all directions equals σ^2, i.e. $p = 1$, $q_1 = M$ and $\delta_{\mathbf{Y},1} = \sigma^2$. On the other hand, the sample variances of the projections of $\mathbf{Y}$ for a finite sample of size N cannot be the same in all directions, i.e. the sample eigenvalues are almost surely different. Since $\tilde{\lambda}_{\mathbf{Y},M}$ is the smallest sample eigenvalue, it has a negative bias, i.e. its expected value is smaller than σ^2. As can be seen from Fig. **??**(b), the smaller N is, the larger the spread of the sample eigenvalues is, i.e. the larger the bias of $\tilde{\lambda}_{\mathbf{Y},M}$ is. As a result, the use of $\tilde{\lambda}_{\mathbf{Y},M}$ can cause a considerable underestimation of the noise variance for very small images satisfying Assumption 1.

2. $k = q_p$:

$$\sigma^2_{est,k} = \frac{\tilde{\lambda}_{\mathbf{Y},M-q_p+1} + \cdots + \tilde{\lambda}_{\mathbf{Y},M}}{q_p} \tag{3.34}$$

   i.e. the average of all sample eigenvalues corresponding to the noise is utilized as a noise variance estimator.

   The advantage of this estimator is that it utilizes all information provided by PCA, i.e. all sample eigenvalues corresponding to the noise are used. Consequently, these estimator has the smallest bias among all $\sigma^2_{est,k}$ provided that Assumption 1 holds. This allows accurate estimation of the noise variance for small N, i.e. for small images. Additionally, one can select a small block subset from all image blocks and use only this subset

Table 3.1: The estimates of q_i and $\delta_{\mathbf{Y},i}$ for the image shown in Fig. 3.2. The estimates of $\delta_{\mathbf{Y},i}$ have been computed with the accuracy 10^{-4}.

i	q_i	$\delta_{\mathbf{Y},i}$
1	1	87541.0010
2	1	1115.0010
3	1	612.8945
4	1	61.4036
5	1	46.7131
6	1	25.0071
7	1	3.5403

i	q_i	$\delta_{\mathbf{Y},i}$
8	1	2.3485
9	1	1.2356
10	1	0.7212
11	1	0.1731
12	1	0.1080
13	1	0.0571
14	1	0.0157

i	q_i	$\delta_{\mathbf{Y},i}$
15	1	0.0085
16	1	0.0051
17	1	0.0047
18	1	0.0014
19	1	0.0007
20	1	0.0002
21	5	0.0000

for noise variance estimation without significant loss of accuracy, which results in smaller computation time.

The disadvantage of this estimator is that it relies on the estimate of q_p. If the estimate of q_p is larger than its true value, some sample eigenvalues affected by the original image are used, which leads to overestimation of the noise variance.

3.4.4 Example

Let us consider an example of sample PCA.

Since Bartlett's test can be used to test the equality of any consecutive eigenvalues [44], the estimation of q_i can be continued in the way described in Section 3.4.1 for $i = p-1, \ldots, 1$. Besides, the average of the sample eigenvalues can be used to estimate $\delta_{\mathbf{Y},i}$ for $i = p-1, \ldots, 1$ as well.

The image shown in Fig. 3.2 was taken as a test image. This is the standard test image 'Cameraman', which has been blurred with Gaussian kernel in order to remove possible noise. The estimates of q_i and $\delta_{\mathbf{Y},i}$ computed using 5×5 blocks are shown in Table 3.1; and the nonzero estimates of $\delta_{\mathbf{Y},i}$ are plotted in Fig. 3.3. As one can see, all nonzero eigenvalues are distinct and have approximately exponential decay. The last five eigenvalues equal zero, which means that PCA can be used for noise variance estimation for this image.

3.4.5 Summary

All in all, sample PCA provides the following possibilities for assessment of the population properties:

1. for checking Assumption 1: test (3.16) with $k = m$; check (3.24);
2. for estimating q_p: sequence of tests (3.16) with $k = 2, 3, \ldots$;
3. for estimating σ^2: estimators (3.33) and (3.34).

Figure 3.2: 512×512 'Cameraman' image blurred with Gaussian kernel with standard deviation $\sigma = 2$.

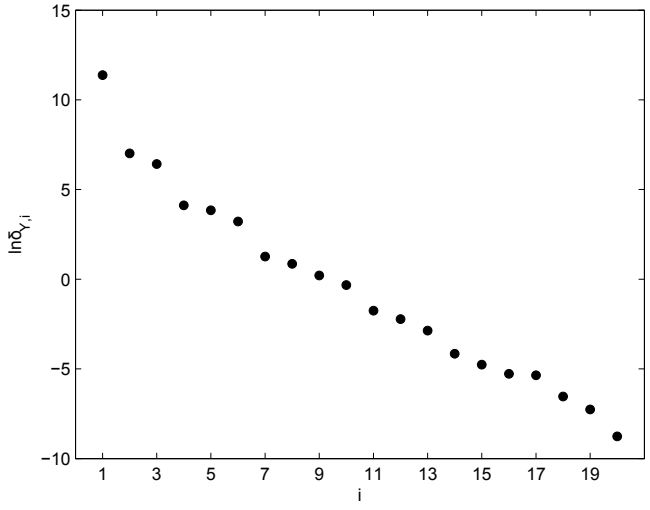

Figure 3.3: Semi-logarithmic plot of the nonzero estimates of $\delta_{\mathbf{Y},i}$ for the image shown in Fig. 3.2.

These tests and estimators can be combined in different ways, which results in a family of methods. In the next two sections, two efficient algorithms from this family, which provide good trade-offs between accuracy and computation time, are described.

3.5 Method based on image block selection

The method described in this section operates as follows. If some estimate σ^2_{est} of the noise variance is available, Assumption 1 is checked. If it holds, σ^2_{est} is taken as the final estimate. Otherwise, the method tries to extract a subset of image blocks, for which Assumption 1 holds, computes a new noise variance estimate using this subset, and repeats the procedure. A strategy to extract a subset of image blocks, which satisfies Assumption 1, is described below.

3.5.1 Image block subset selection

Recall that V_{M-m} is the subspace introduced in Assumption 1. Let d_i be the distances of $\mathbf{x}_i$ to V_{M-m}, $i = 1, \ldots, N$. Assumption 1 holds, i.e. $\mathbf{x}_i \in V_{M-m}$, $i = 1, \ldots, N$, if and only if $d_i = 0$, $i = 1, \ldots, N$. Trying to satisfy this condition, it is reasonable to discard the blocks with the largest d_i from the total N image blocks.

Unfortunately, the values of d_i are not available in practice. Computation of the distances of $\mathbf{y}_i$ to V_{M-m} does not help, since a large distance of $\mathbf{y}_i$ to V_{M-m} can be caused by noise. Several heuristics may be applied in order to select blocks with largest d_i, e.g. to pick blocks with largest variance, largest range, or largest entropy. The first strategy has been chosen, since it is fast to compute and the results are the most accurate in most cases. This strategy is examined below.

Let us consider Spearman's rank correlation coefficient ρ between d_i and $s(\mathbf{x}_i)$, where $s(\mathbf{x}_i)$ is the sample standard deviation of elements of block $\mathbf{x}_i$, $i = 1, \ldots, N$. It is computed as follows. First, d_i and $s(\mathbf{x}_i)$ are sorted independently of each other, which gives the rank of each element is the sorted sequence. Let $r_{d,i}$ be the rank of d_i and $r_{s,i}$ be the rank of $s(\mathbf{x}_i)$. Then, Pearson's correlation coefficient between $r_{d,i}$ and $r_{s,i}$ is calculated:

$$\rho = \frac{\sum_{i=1}^{N}(r_{d,i} - \overline{r}_d)(r_{s,i} - \overline{r}_s)}{\sqrt{\sum_{i=1}^{N}(r_{d,i} - \overline{r}_d)^2 \sum_{i=1}^{N}(r_{s,i} - \overline{r}_s)^2}} \tag{3.35}$$

where

$$\overline{r}_d = \frac{1}{N}\sum_{i=1}^{N} r_{d,i} \qquad \overline{r}_s = \frac{1}{N}\sum_{i=1}^{N} r_{s,i}. \tag{3.36}$$

ρ indicates how well the relationship between d_i and $s(\mathbf{x}_i)$ can be described by a monotonic function. Its values for the reference images from the TID2008

Table 3.2: The values of ρ for the reference images from the TID2008 database (75 grayscale images). The second column is the sample mean and the third column is the sample standard deviation computed across all images. V_{M-m} was computed as the linear span of $\tilde{\mathbf{v}}_{\mathbf{X},1}, \ldots, \tilde{\mathbf{v}}_{\mathbf{X},M-m}$. $M = 5 \times 5$.

m	$\overline{\rho}$	$s(\rho)$
2	0.52	0.140
3	0.59	0.135
4	0.63	0.125
5	0.67	0.116
6	0.70	0.110
7	0.72	0.105
8	0.74	0.102
9	0.76	0.098
10	0.78	0.082
11	0.80	0.078
12	0.81	0.079
13	0.82	0.076
14	0.83	0.074
15	0.84	0.075

database [64] are shown in Table 3.2. As one can see, there is a considerable positive correlation between d_i and $s(\mathbf{x}_i)$. That means large $s(\mathbf{x}_i)$ commonly corresponds to large d_i. Since the noise is signal-independent, $s^2(\mathbf{x}_i)$ approximately equals $s^2(\mathbf{y}_i) - \sigma^2$. Hence large $s^2(\mathbf{y}_i)$ commonly corresponds to large d_i. As a result, we can discard blocks with the largest $s^2(\mathbf{y}_i)$ in order to discard blocks with the largest d_i.

The experiment presented in Table 3.2 shows that image structures, which are different from the general image texture, typically have a large local variance. However, this is not the case for all images. For example, the image shown in Fig. 3.4 consists of two parts: a stripe pattern on the left side, and a complex texture on the right side. $s(\mathbf{x}_i) = 127.5$ and the mean of d_i is 0.1 for the blocks in the stripe pattern, but $s(\mathbf{x}_i) = 49.2$ and the mean of d_i is 9.1 for the blocks in the complex texture even for $m = 1$. This synthetic example is unlikely for real-world images, but it shows that this heuristic cannot be proven.

In 8-bit images with the gray value range [0, 255], the noise is usually clipped. In order to prevent the influence of clipping to the noise level estimation process, blocks, in which more than 10% of pixels have the value 0 or 255, are skipped.

3.5.2 Algorithms

Let $Q(p, y)$ be p-quantile of $\{s^2(\mathbf{y}_i), i = 1, \ldots, N\}$ computed using Definition 3 from [40], i.e. $Q(p, y)$ is the kth order statistic of $\{s^2(\mathbf{y}_i)\}$, where k is the

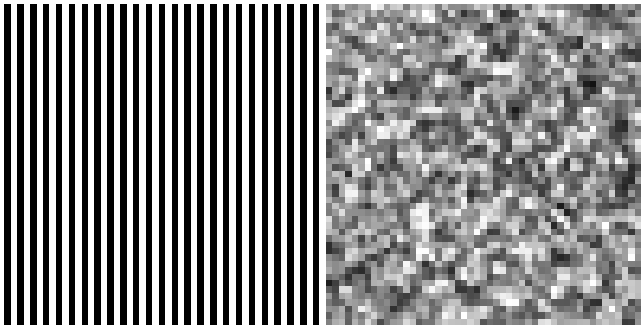

Figure 3.4: A counterexample for the selection of the blocks with the largest variance.

integer nearest to Np. Let $B(p, y)$ be the subset of blocks of image y, whose sample variance is not greater than $Q(p, y)$:

$$B(p, y) = \{\mathbf{y}_i \mid s^2(\mathbf{y}_i) \leq Q(p, y),\ i = 1, \ldots, N\}. \tag{3.37}$$

The first version of the proposed noise variance estimation procedure is presented in the algorithm `EstimateNoiseVariance`. It starts with the whole set of image blocks, which corresponds to quantile level $p = 1$ (line 1). At each iteration, sample PCA is done using image block set $B(p, y)$ (line 4). If Assumption 1 is satisfied, the algorithm stops. Otherwise, p is decreased by Δp, i.e. the blocks with the largest variance are discarded, and the loop continues. Parameter p_{min} defines the minimal block subset size, for which sample PCA produces statistically significant results. Check (3.24) is used in order to test Assumption 1 (line 6), and the noise variance is assessed using estimator (3.33), i.e. as the smallest sample eigenvalue (line 5).

The algorithm `ApplyPCA` computes $\tilde{\mathbf{v}}_{\mathbf{Y},i}$ and $\tilde{\lambda}_{\mathbf{Y},i}$, $i = 1, \ldots, M$ for block set $B(p, y)$.

Check (3.24) depends on current noise variance estimate σ^2_{est}. Therefore, if the noise variance is significantly overestimated at the first iteration of the algorithm `EstimateNoiseVariance` (for $p = 1$), check (3.24) is not reliable and it can be passed even if Assumption 1 does not actually hold. For this reason, the PCA-based method can be initialized with a noise variance estimate, which is computed by some other method, in order to increase the estimator robustness. This strategy is implemented in the second version of the proposed method, which is presented in the algorithm `EstimateNoiseVarianceIterative`. It uses the algorithms `GetUpperBound` and `GetNextEstimate`.

The algorithm `EstimateNoiseVarianceIterative` takes the result of the algorithm `GetUpperBound` as the initial estimate and iteratively invokes algorithm `GetNextEstimate` until convergence is reached. Parameter i_{max} is the maximum number of iterations.

The algorithm `GetUpperBound` computes a noise variance upper bound. It is independent of image block PCA and, like many other noise estimation ap-

Algorithm 1 `EstimateNoiseVariance`

Input: image y corrupted with AWGN
Output: noise variance estimate σ^2_{est}

$p \leftarrow 1$
$\sigma^2_{est} \leftarrow 0$
while $p \geq p_{min}$ **do**
  $\tilde{\mathbf{v}}_{\mathbf{Y},i}, \tilde{\lambda}_{\mathbf{Y},i} \leftarrow$ `ApplyPCA(` $B(p,y)$ `)`
  $\sigma^2_{est} \leftarrow \lambda_{\mathbf{Y},M}$
  if $\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M} < T_\lambda \sigma^2_{est} / \sqrt{|B(p,y)|}$ **then**
    return σ^2_{est}
  end if
  $p \leftarrow p - \Delta p$
end while
return σ^2_{est}

Algorithm 2 `EstimateNoiseVarianceIterative`

Input: image y corrupted with AWGN
Output: noise variance estimate σ^2_{est}

$\sigma^2_{ub} \leftarrow$ `GetUpperBound`(y)
$\sigma^2_{est} \leftarrow \sigma^2_{ub}$
for $i = 1$ **to** i_{max} **do**
  $\sigma^2_{next} \leftarrow$ `GetNextEstimate`$(y, \sigma^2_{est}, \sigma^2_{ub})$
  if $\sigma^2_{est} = \sigma^2_{next}$ **then**
    return σ^2_{est}
  end if
  $\sigma^2_{est} \leftarrow \sigma^2_{next}$
end for
return σ^2_{est}

Algorithm 3 `GetNextEstimate`

Input: image y, previous estimate σ^2_{est}, upper bound σ^2_{ub}
Output: next estimate σ^2_{next}

$p \leftarrow 1$
$\sigma^2_{next} \leftarrow 0$
while $p \geq p_{min}$ **do**
  $\tilde{\mathbf{v}}_{\mathbf{Y},i}, \tilde{\lambda}_{\mathbf{Y},i} \leftarrow$ `ApplyPCA(` $B(p,y)$ `)`
  $\sigma^2_{next} \leftarrow \tilde{\lambda}_{\mathbf{Y},M}$
  if $\tilde{\lambda}_{\mathbf{Y},M-m+1} - \tilde{\lambda}_{\mathbf{Y},M} < T_\lambda \sigma^2_{est} / \sqrt{|B(p,y)|}$ **and** $\sigma^2_{next} \leq \sigma^2_{ub}$ **then**
    return σ^2_{next}
  end if
  $p \leftarrow p - \Delta p$
end while
return σ^2_{next}

proaches, it is based on the analysis of the image block variance distribution. Namely, this algorithm returns $C_0 Q(p_0, y)$, where C_0 and p_0 are parameters.

The algorithm `GetNextEstimate` extracts the subset of the image blocks, which satisfies Assumption 1. Like the algorithm `EstimateNoiseVariance`, it implements the approach described in Section 3.5.1 by taking p-quantiles of the block variance distribution. It starts from the largest possible p equal to 1, which corresponds to the whole set of image blocks. Then it discards blocks with the largest variance by reducing p to $1 - \Delta p$, $1 - 2\Delta p$, and so on, until p is smaller than p_{min}. Again, check (3.24) is applied for testing Assumption 1 in line 6, but, in contrast to the algorithm `EstimateNoiseVariance`, previous estimate σ^2_{est} is utilized here. The smallest sample eigenvalue (estimator (3.33)) is taken as a noise variance estimate (line 5). Upper bound σ^2_{ub} is used as an additional check of the correctness of the computed estimate in line 6.

The values of all algorithm parameters are given in Section 3.7.1.

3.5.3 Efficient implementation

When considering the execution time of the algorithms `EstimateNoiseVariance` and `EstimateNoiseVarianceIterative`, one has to concentrate on sample PCA (the algorithm `ApplyPCA`), because it is invoked inside one loop in the algorithm `EstimateNoiseVariance` and inside two loops in the algorithm `EstimateNoiseVarianceIterative`. Sample PCA consists of two parts:

1. Computation of the sample covariance matrix

$$\frac{1}{|B(p,y)| - 1}\Big(\sum_{\mathbf{y}_i \in B(p,y)} \mathbf{y}_i \mathbf{y}_i^{\mathrm{T}} - \frac{1}{|B(p,y)|} \sum_{\mathbf{y}_i \in B(p,y)} \mathbf{y}_i \sum_{\mathbf{y}_i \in B(p,y)} \mathbf{y}_i^{\mathrm{T}} \Big). \quad (3.38)$$

   The number of operations is proportional to $|B(p,y)|M^2$.

2. Computation of the eigenvalues and eigenvectors of the sample covariance matrix. The number of operations is proportional to M^3 [66].

Since $|B(p,y)| \gg M$, the computation of the sample covariance matrix is the most expensive part of sample PCA, however, this matrix is calculated several times for overlapped block subsets. This computation is redundant and can be avoided using the strategy described below.

Let $C_X = \sum_{\mathbf{y}_i \in X} \mathbf{y}_i \mathbf{y}_i^{\mathrm{T}}$ and $\mathbf{c}_X = \sum_{\mathbf{y}_i \in X} \mathbf{y}_i$. Note that for disjoint sets X_1 and X_2, $C_{X_1 \cup X_2} = C_{X_1} + C_{X_2}$ and $\mathbf{c}_{X_1 \cup X_2} = \mathbf{c}_{X_1} + \mathbf{c}_{X_2}$. Then (3.38) can be represented as

$$\frac{1}{|B(p,y)| - 1}\Big(C_{B(p,y)} - \frac{1}{|B(p,y)|} \mathbf{c}_{B(p,y)} \mathbf{c}_{B(p,y)}^{\mathrm{T}} \Big). \quad (3.39)$$

The algorithm `ApplyPCA` is called only with arguments

$$B(1,y) \supset B(1 - \Delta p, y) \supset \cdots \supset B(1 - n\Delta p, y) \quad (3.40)$$

where $n = \lfloor (1 - p_{min})/\Delta p \rfloor$ and $\lfloor t \rfloor$ is the largest integer not greater than t. For $j = 0, \ldots, n-1$, let us consider sets

$$Y_j = \{\mathbf{y}_i \mid Q(1-(j+1)\Delta p, y) < s^2(\mathbf{y}_i) \leq Q(1-j\Delta p, y)\}. \tag{3.41}$$

Then, $B(1-j\Delta p, y) = B(1-(j+1)\Delta p, y) \cup Y_j$.

At the beginning of the program, matrices $C_{B(1-j\Delta p,y)}$ and vectors $\mathbf{c}_{B(1-j\Delta p,y)}$, $j = 0, \ldots, n$ are precomputed in the following way. Matrices $C_{B(1-n\Delta p,y)}$, C_{Y_0}, ..., $C_{Y_{n-1}}$ and vectors $\mathbf{c}_{B(1-n\Delta p,y)}$, $\mathbf{c}_{Y_0}, \ldots, \mathbf{c}_{Y_{n-1}}$ are calculated by definition and

$$\begin{aligned} C_{B(1-j\Delta p,y)} &= C_{B(1-(j+1)\Delta p,y)} + C_{Y_j} \\ \mathbf{c}_{B(1-j\Delta p,y)} &= \mathbf{c}_{B(1-(j+1)\Delta p,y)} + \mathbf{c}_{Y_j} \end{aligned} \tag{3.42}$$

for $j = n-1, \ldots, 0$. Then, these precomputed matrices and vectors are utilized in the algorithm `ApplyPCA` when computing the sample covariance matrix using (3.39). When the precomputation is applied, the number of operations in (3.39) is proportional to M^2, which is $|B(p,y)|$ times smaller than in the direct implementation. Recursive procedure (3.42) ensures that the precomputation itself is optimal in the sense that expression $\mathbf{y}_i \mathbf{y}_i^{\mathrm{T}}$ is computed only once for each vector $\mathbf{y}_i$.

3.6 Method based on image region selection

The method described in Section 3.5 analyzes each block individually in order to construct a block subset satisfying Assumption 1. This leads to a computationally expansive procedure of sorting all image blocks by their variance, which is necessary for the calculation of quantiles $Q(p,y)$. The complexity of this sorting operation is $O(N \log N)$. Besides, this method starts noise variance estimation process from the whole image. This results in the most accurate estimate but it may be redundant, because the noise is signal-independent and a small part of the input image may be enough to estimate the noise variance.

In this section, a faster but slightly less accurate noise variance estimation method is proposed. The speedup is achieved in two ways:

1. Block sets are constructed using image regions, i.e. all blocks from a region are simultaneously included into the block set used in sample PCA. This allows avoiding the block sorting.

2. Only a small part of the input image is utilized. It helps to reduce the time of the sample covariance matrix computation, which is the most expensive part of the program.

3.6.1 Image region selection

When one chooses number of image blocks N to be used in sample PCA, there is a trade-off between the execution time and the accuracy:

1. A large number of blocks leads to accurate estimates, but large execution time, because the time of the sample covariance matrix computation is proportional to the number of blocks.

2. A small number of blocks leads to small execution time, but the results are not statistically significant.

Therefore, the image domain is partitioned into rectangular regions $\{R_k\}$ of predefined size $W_1 \times \ldots \times W_d$ and image blocks only from several of these regions are used. The predefined region size guarantees that there is a sufficient number of image blocks for accurate noise variance estimation, whereas processing only a small number of the regions allows reduction of the execution time. The regions are defined as

$$R_k = \big\{i_1 W_1 + 1, \ldots, (i_1+1)W_1\big\} \times \cdots \times \big\{i_d W_d + 1, \ldots, (i_d+1)W_d\big\} \quad (3.43)$$

where

$$\begin{aligned} &k = i_1 + i_2 K_1 + \ldots + i_d K_1 \cdot \ldots \cdot K_{d-1} \\ &K_1 = \lfloor S_1/W_1 \rfloor, \ldots, K_d = \lfloor S_d/W_d \rfloor \\ &i_1 \in \{0, \ldots, K_1 - 1\}, \ldots, i_d \in \{0, \ldots, K_d - 1\}. \end{aligned} \quad (3.44)$$

Recall that $S_1 \times \ldots \times S_d$ is the image size. Each region R_k contains

$$(W_1 - M_1 + 1) \cdot \ldots \cdot (W_d - M_d + 1) \quad (3.45)$$

blocks of size $M_1 \times \cdots \times M_d$ with left-top-front corner positions in set

$$\big\{i_1 W_1 + 1, \ldots, (i_1+1)W_1 - M_1 + 1\big\} \times \cdots \times \big\{i_d W_d + 1, \ldots, (i_d+1)W_d - M_d + 1\big\}. \quad (3.46)$$

In the ideal case, regions satisfying Assumption 1 should be selected for processing. However, no information about q_p and $\delta_{\mathbf{X},p}$ is available before applying PCA, hence some heuristic should be utilized for the region selection. Intuitively, regions containing simple structures should be taken, which can be described by the region variance, entropy, or autocorrelation. Similar to the technique used in Section 3.5.1, the region variance is utilized here, because it is fast to compute.

In 8-bit images with gray value range $[0, 255]$, noise clipping usually occurs, as mentioned in Section 3.5.1. Therefore, regions, in which more than 5% of pixels have the value 0 or 255, are skipped.

3.6.2 Algorithm

The fast noise variance estimation method is presented in the algorithm `EstimateNoiseVarianceFast`. First, sample variance s^2 of image pixels is computed for each region R_k. The sample variances are sorted in ascending order, and the sorting result is stored in permutation π.

In lines 3–20, the algorithm tries to compute a noise variance estimate using blocks of size $M_1 \times \cdots \times M_d$. Since the sample covariance matrix computation time is proportional to $NM^2 = NM_1^2 \cdot \ldots \cdot M_d^2$, the algorithm starts with small block size $M_1^{min} \times \ldots \times M_d^{min}$. If Assumption 1 is satisfied for some block size, the algorithm stops (line 19). If the assumption is not satisfied, $\delta_{\mathbf{X},p}$ may be nonzero, i.e. $\mathbf{X}$ may not lie in some proper subspace of $\mathbb{R}^M$. It is supposed that if the vector dimension M is increased, $\mathbf{X}$ will lie in some proper subspace of $\mathbb{R}^M$ and $\delta_{\mathbf{X},p}$ will be zero. Hence the block size is increased in the loop from $M_1^{min} \times \ldots \times M_d^{min}$ to $M_1^{max} \times \ldots \times M_d^{max}$. If the assumption is not satisfied for any block size, the noise variance estimate computed using the largest block size $M_1^{max} \times \ldots \times M_d^{max}$ is taken as the final estimate (line 13). In this case, the highest number of dimensions is used to represent image structures so that the estimation error, which is $\delta_{\mathbf{X},p}$, is expected to be the smallest.

Algorithm 4 `EstimateNoiseVarianceFast`

Input: image y corrupted with AWGN
Output: noise variance estimate σ^2_{est}

```
1:  π ← sort({s²(R_k)})
2:  for (M_1, ..., M_d) = (M_1^min, ..., M_d^min) to (M_1^max, ..., M_d^max) do
3:    σ²_est ← InvalidValue
4:    assumptionIsSatisfied ← false
5:    for k = 1 to k_max do
6:      ṽ_{Y,i}, λ̃_{Y,i} ← ApplyPCA(R_{π(1)}, ..., R_{π(k)})
7:      Estimate q_p using sequence of tests (3.16)
8:      if q_p ≥ m then
9:        σ²_est ← (3.34)
10:       assumptionIsSatisfied ← true
11:     else
12:       if σ²_est = InvalidValue then
13:         σ²_est ← (3.34)
14:       end if
15:       break
16:     end if
17:   end for
18:   if assumptionIsSatisfied then
19:     break
20:   end if
21: end for
22: return σ²_est
```

In lines 6–16, the image blocks from k regions with the smallest variance, namely $R_{\pi(1)}, \ldots, R_{\pi(k)}$, are processed. Final estimate σ^2_{est} is updated if $q_p \geq m$ or if its previous value was not valid (lines 8–16). The algorithm starts with $k = 1$, for which the number of processed image blocks is minimal and, therefore, the chance that $\delta_{\mathbf{X},p} = 0$ is the highest. Then, k is increased in order to process a larger number of blocks and increase the accuracy. k is increased until $q_p < m$ (line 15), i.e. until Assumption 1 does not hold. As a result, the enumeration of k allows computation of the estimate, for which the assumption is satisfied, with the highest possible accuracy.

Obviously, there are several ways to increase the block size from $M_1^{min} \times \ldots \times M_d^{min}$ to $M_1^{max} \times \ldots \times M_d^{max}$ when $d > 1$. When the image resolution and the image content are isotropic, it is reasonable to use only isotropic blocks with $M_1 = M_2 = \ldots = M_d$. In this case, the block size is increased by 1 simultaneously in all dimensions at each iteration of the loop in lines 3–20. Otherwise, a different strategy can be applied, for example, one can fix the block size in all dimensions except for one and vary the size only in that dimension.

Because of the fact that only a small part of the image is used to estimate the noise variance, estimator (3.34), which utilizes all sample eigenvalues corresponding to noise, is applied. Since estimation of q_p is necessary in this case, Bartlett's test (3.16) is used in this algorithm.

However, number of processed blocks N is still much larger than vector dimension M, so that the time to compute the covariance matrix, which is proportional to NM^2, is much larger than the eigenvalue computation time proportional to M^3.

Like for the method described in Section 3.5.3, values $\sum \mathbf{y}_i \mathbf{y}_i^{\mathrm{T}}$ and $\sum \mathbf{y}_i$ computed for regions $R_{\pi(1)}, \ldots, R_{\pi(k)}$ are reused in the calculating the sample covariance matrix for regions $R_{\pi(1)}$, …, $R_{\pi(k)}$, $R_{\pi(k+1)}$ in order to avoid redundant computations.

The values of all algorithm parameters are given in Section 3.7.1.

3.7 Experiments

Further in the experiments, the algorithm `EstimateNoiseVarianceIterative` from Section 3.5 is referred to as PCA-BS (PCA with block selection), and the algorithm `EstimateNoiseVarianceFast` from Section 3.6 is referred to as PCA-RS (PCA with region selection). The accuracy and the speed of these algorithms have been evaluated on two databases: TID2008 [64] and MeasTex [82]. The following recent methods have been included in the comparison:

1. methods which assume that the original image has a sufficient amount of homogeneous areas:
   (a) [87], where Fisher's information is used in order to divide image blocks into two groups: homogeneous areas and textural areas.
   (b) [95], which applies a Sobel edge detection operator in order to exclude the noise-free image content.

(c) [9], which applies Laplacian convolution and edge detection in order to find homogeneous areas.

(d) [3], where the noise variance is estimated as the mode of the distribution of local variances.

(e) [89], which divides the input image into blocks and computes the block standard deviations.

(f) [76], which subtracts low-frequency components detected by a Gaussian filter and edges detected by an edge detector from the input image. Since the method computes the noise variance as a function of the gray value, the estimates for all gray values have been averaged in order to compute the final estimate, as suggested by the authors during the personal discussion.

2. methods which use other assumptions about the input image:

(a) [21], where nonlocal self-similarity of images is used in order to separate the noise from the signal.

(b) [65], where signal and noise separation is achieved with discrete cosine transform.

(c) [78], which treats the noise variance as a parameter of a Bayesian deblurring and denoising framework.

(d) [24], which estimates the noise standard deviation as the MAD of the wavelet coefficients at the finest decomposition level. The Daubechies wavelet of length 8 has been used in the experiments.

(e) [96], where the noise variance is estimated from a kurtosis model under the assumption that the kurtosis of marginal bandpass filter response distributions is constant for noise-free images.

(f) [7], which uses a measure of bit-plane randomness.

(g) [83], which utilizes multiresolution support data structure assuming that small wavelet transform coefficients correspond to the noise.

The proposed algorithms have been implemented in C++ and Matlab in order to compare their execution time with machine code and Matlab implementations of the others. The source code of both C++ and Matlab implementations is available at `http://physics.medma.uni-heidelberg.de/cms/projects/132-pcanle`.

3.7.1 Choice of the parameters

The algorithms PCA-BS and PCA-RS have been tested with different sets of the parameters; and the sets presented in Tables 3.3 and 3.4 are suggested. They have been used in all experiments in this section.

Regarding block size $M_1 \times M_2$, there is a trade-off between the ability to handle complex textures on one hand and the statistical significance of the result

Table 3.3: Parameters of PCA-BS

Parameter	**Denotation**	**Value**
Block width	M_1	5
Block height	M_2	5
Upper bound factor	C_0	3.1
Upper bound quantile level	p_0	0.0005
Noise subspace dimension in Assumption 1	m	7
Threshold in check (3.24)	T_λ	49
Block part step	Δp	0.05
Minimal block part	p_{min}	0.06
Maximal iteration count	i_{max}	10

Table 3.4: Parameters of PCA-RS

Parameter	**Denotation**	**Value**
Minimal block width	M_1^{min}	5
Minimal block height	M_2^{min}	5
Maximal block width	M_1^{max}	8
Maximal block height	M_2^{max}	8
Region width	W_1	64
Region height	W_2	64
Noise subspace dimension in Assumption 1	m	2
Maximal region count	k_{max}	2
Significance level in (3.16)	α	0.01

and the speed on the other hand. In order to satisfy Assumption 1, we need to find correlations between pixels of the image texture. Hence the block size should be large enough and, at least, be comparable with the size of the textural pattern. On the other hand, the block size cannot be arbitrary large for the following reasons:

1. In PCA-BS, since $\tilde{\lambda}_{\mathbf{Y},M}$ is the smallest order statistic of the sample eigenvalues representing the noise, it has a negative bias for a finite number of blocks, which increases with $M = M_1 M_2$.

2. In PCA-RS, the main purpose is to make PCA as fast as possible without significant loss of accuracy. However, the time of the sample covariance matrix computation is proportional to $NM^2 = NM_1^2 M_2^2$.

Blocks sizes from 4×4 to 8×8 are good choices for real-world images of size from 128×128 to 2048×2048. When the horizontal and the vertical resolution of the input image are not equal, nonsquare blocks can be considered as well.

Parameters C_0 and p_0 have been chosen so that $\sigma^2_{ub} = C_0 Q(p_0)$ is an upper bound of the true noise variance, i.e. this value always overestimates the noise level. Similar to [89] and [35], blocks with the smallest variances are used here, i.e. p_0 is close to 0. During the experiments with TID2008 and MeasTex, the output of the algorithm was never equal to σ^2_{ub}, hence it was always a PCA-based estimate.

Check (3.24) is robust when the difference between eigenvalue indices $M - m+1$ and M is large, i.e. when m is large. However, if m is too large, $\tilde{\lambda}_{\mathbf{Y},M-m+1}$ is often influenced by the original image. The selection of T_λ depends on the selection of m.

The results are not sensitive to parameter i_{max}, which can be selected from range $[3, +\infty)$. This parameter is needed only to guarantee that the algorithm always stops.

Maximal region count k_{max} is a compromise between the accuracy and the speed. In order to get high accuracy, one has to increase the number of processed blocks, i.e. increase the number of regions. At the same time, the execution time is proportional to the number of processed blocks, hence one should keep it as small as possible.

3.7.2 Experiments with TID2008

The TID2008 database contains 25 RGB images. 24 of them are real-world scenes and one image is artificial. Each color component has been processed independently, i.e. the results for each noise level have been obtained using 75 grayscale images. Noisy images with the noise variance 65 and 130 are included in the database; and noisy images with the noise variance 25 and 100 have been generated additionally. This database has been already utilized for the evaluation of several noise level estimation methods [21, 87, 57]. Some images from this database are shown in Fig. 3.5.

Though the reference images from TID2008 are considered as noise-free images, they still contain a small level of noise. This level should be estimated in order to compare all methods fairly. This has been done by the following semi-automatic procedure:

1. Rectangular homogeneous area A has been selected manually in each reference image. This area contains almost no structure, i.e. it contains almost only noise with variance σ^2_{ref}. Therefore, the distribution of $x(p_1, p_2)$, $(p_1, p_2) \in A$ can be approximated by $\mathcal{N}(\mu_A, \sigma^2_{ref})$, where μ_A is the mean value of $x(p_1, p_2)$ in A.

2. A high-pass filter has been used in order to remove possible image structures from A. Namely, the following differences between neighbor pixels have been computed:

   (a) $\big(x(p_1+1, p_2) - x(p_1, p_2)\big)/\sqrt{2}$, if width$(A) >$ height(A),

   (b) $\big(x(p_1, p_2+1) - x(p_1, p_2)\big)/\sqrt{2}$, if width$(A) \leq$ height(A),

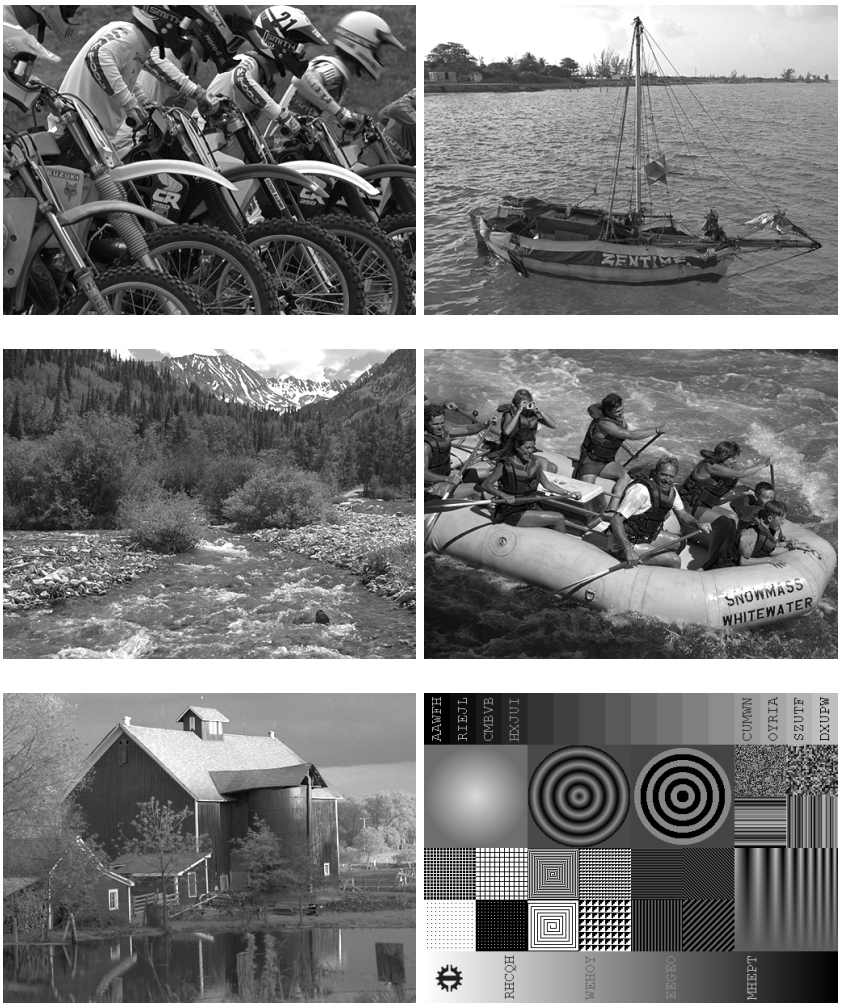

Figure 3.5: Images from the TID2008 database.

where $(p_1, p_2), (p_1 + 1, p_2), (p_1, p_2 + 1) \in A$.

3. Since $x(p_1, p_2) \sim \mathcal{N}(\mu_A, \sigma^2_{ref})$,

$$\begin{aligned} \big(x(p_1 + 1, p_2) - x(p_1, p_2)\big)/\sqrt{2} &\sim \mathcal{N}(0, \sigma^2_{ref}) \\ \big(x(p_1, p_2 + 1) - x(p_1, p_2)\big)/\sqrt{2} &\sim \mathcal{N}(0, \sigma^2_{ref}) \end{aligned} \tag{3.47}$$

where $(p_1, p_2), (p_1 + 1, p_2), (p_1, p_2 + 1) \in A$. Therefore, the noise variance σ^2_{ref} has been estimated as the variance of these differences.

The manually selected areas and the values of σ^2_{ref} are available at `http://physics.medma.uni-heidelberg.de/cms/projects/132-pcanle`.

Then, all noise variance estimates have been corrected according to the following formula:

$$\sigma^2_{corr} = \sigma^2_{est} - \sigma^2_{ref} \tag{3.48}$$

where σ^2_{est} is the algorithm output.

The comparison results are presented in Tables 3.5 and 3.6. Since the execution time of [21] has not been provided by the author, the execution time of the BM3D filter [18], which is a step of [21], has been measured. In the execution time comparison, it is assumed that the CPUs Intel Celeron 1.4 GHz and Intel Core 2 Duo 1.66 GHz used to measure the performance of the methods [65] and [87] are about two times slower than the CPU Intel i7 920 2.67 GHz used to measure the performance of the other methods.

Table 3.5: The accuracy of the considered methods for TID2008. $\overline{\sigma}_{corr} - \sigma$ is the bias of corrected estimates, $s(\sigma_{corr})$ is the standard deviation of corrected estimates, $\max|\sigma_{corr} - \sigma|$ is the maximum difference between a corrected estimate and the true value. The last column is the percentage of the images, for which the method cannot estimate the noise level. The best result in a column is selected with the bold font. For the methods marked with *, the values of σ_{est} have been provided by the authors.

Method	$\overline{\sigma}_{corr} - \sigma$	$s(\sigma_{corr})$	$\max\|\sigma_{corr} - \sigma\|$	% of failures
$\sigma^2 = 25$ ($\sigma = 5$)				
PCA-BS	-0.027	**0.147**	**0.500**	0
PCA-RS	**0.026**	0.223	0.685	0
[87]	–	–	–	–
[95]	0.322	0.547	2.859	0
[9]	0.605	0.882	4.116	0
[3]	0.617	1.530	8.941	0
[89]	-1.499	1.822	5.000	57.3
[76]	4.954	3.408	21.037	0
[21]*	-0.039	0.158	0.525	0
[65]	–	–	–	–

Continued on the next page

Table 3.5 – *Continued from the previous page*

Method	$\overline{\sigma}_{corr} - \sigma$	$s(\sigma_{corr})$	$\max \lvert\sigma_{corr} - \sigma\rvert$	% of failures
[78]	-0.345	0.857	3.507	0
[24]	1.127	1.030	5.194	0
[96]	-0.487	3.323	24.719	1.3
[7]	3.227	2.266	9.158	0
[83]	2.144	2.224	8.903	0
$\sigma^2 = 65$ ($\sigma \approx 8.062$)				
PCA-BS	-0.043	**0.103**	0.486	0
PCA-RS	0.021	0.214	0.822	0
[87]*	-0.074	0.110	**0.401**	0
[95]	0.228	0.430	2.093	0
[9]	0.206	0.769	2.867	0
[3]	0.292	1.526	6.343	0
[89]	-1.467	2.044	8.062	45.3
[76]	4.049	3.290	19.557	0
[21]	–	–	–	–
[65]*	**0.001**	0.209	1.078	0
[78]	-0.858	0.971	4.211	0
[24]	0.724	1.003	4.281	0
[96]	-0.899	1.384	8.062	0
[7]	3.173	1.671	8.968	0
[83]	2.067	2.325	10.160	0
$\sigma^2 = 100$ ($\sigma = 10$)				
PCA-BS	**0.009**	**0.125**	**0.307**	0
PCA-RS	0.052	0.293	1.066	0
[87]	–	–	–	–
[95]	0.232	0.412	1.935	0
[9]	0.269	0.640	3.088	0
[3]	0.582	1.061	6.019	0
[89]	-1.517	2.145	10.000	42.7
[76]	3.553	3.111	20.238	0
[21]*	0.040	0.175	0.717	0
[65]	–	–	–	–
[78]	-0.746	0.750	2.400	0
[24]	0.819	0.900	4.011	0
[96]	-0.395	2.749	20.956	0
[7]	2.204	2.519	9.551	0
[83]	2.248	1.868	7.281	0
$\sigma^2 = 130$ ($\sigma \approx 11.402$)				
PCA-BS	**0.014**	**0.110**	**0.386**	0
PCA-RS	0.083	0.291	1.062	0
[87]*	-0.040	0.136	0.532	0

Continued on the next page

Table 3.5 – *Continued from the previous page*

Method	$\overline{\sigma}_{corr} - \sigma$	$s(\sigma_{corr})$	$\max \lvert\sigma_{corr} - \sigma\rvert$	% of failures
[95]	0.224	0.390	1.943	0
[9]	-0.025	0.777	3.297	0
[3]	0.250	1.464	5.665	0
[89]	-1.467	2.086	6.811	41.3
[76]	3.325	3.506	21.502	0
[21]	–	–	–	–
[65]*	0.094	0.228	1.170	0
[78]	-1.140	1.062	5.351	0
[24]	0.477	0.989	3.711	0
[96]	-0.634	2.634	19.321	0
[7]	2.700	2.510	9.604	0
[83]	2.132	2.279	10.197	0

Table 3.6: The execution time of the considered methods for TID2008. $\min t$ is the minimum execution time, $\bar{t}$ is the average execution time, $\max t$ is the maximum execution time. All implementations are single-threaded. For the methods marked with *, the values have been provided by the authors.

Method	$\min t$	$\bar{t}$	$\max t$	CPU
Machine code implementations (C++, Object Pascal)				
PCA-BS	102 ms	159 ms	169 ms	Intel i7 920 2.67 GHz
PCA-RS	3.4 ms	4.1 ms	28.1 ms	Intel i7 920 2.67 GHz
[95]	1.9 ms	3.1 ms	3.5 ms	Intel i7 920 2.67 GHz
[89]	1.0 ms	1.1 ms	18.8 ms	Intel i7 920 2.67 GHz
[21] ([18])	2628 ms	–	–	Intel i7 920 2.67 GHz
[65]*	$\sim$ 250 ms			Intel Celeron 1.4 GHz
[7]	80 ms	123 ms	140 ms	Intel i7 920 2.67 GHz
[83]	520 ms	805 ms	980 ms	Intel i7 920 2.67 GHz
Matlab implementations				
PCA-BS	911 ms	1491 ms	1583 ms	Intel i7 920 2.67 GHz
PCA-RS	51 ms	81 ms	280 ms	Intel i7 920 2.67 GHz
[87]*	3 min	–	10 min	Intel Core 2 Duo 1.66 GHz
[9]	635 ms	682 ms	1608 ms	Intel i7 920 2.67 GHz
[3]	62 ms	73 ms	402 ms	Intel i7 920 2.67 GHz
[76]	532 ms	681 ms	1621 ms	Intel i7 920 2.67 GHz
[78]	3.7 min	4.8 min	10 min	Intel i7 920 2.67 GHz
[24]	36 ms	38 ms	43 ms	Intel i7 920 2.67 GHz
[96]	2968 ms	2968 ms	3373 ms	Intel i7 920 2.67 GHz

Regarding the algorithm PCA-BS, its accuracy is the highest in most cases. The methods [21] and [87] have comparable results, but they are much slower than PCA-BS: [21] is more than 15 times slower; and [87] is about 50–180 times slower. The method [65] has 2 times larger $s(\sigma_{corr})$ and $\max|\sigma_{corr} - \sigma|$ than PCA-BS has. The methods [95, 9, 3, 89, 76, 78, 24, 96, 7, 83] have more than 3 times larger $s(\sigma_{corr})$ and more than 4 times larger $\max|\sigma_{corr} - \sigma|$ compared with PCA-BS. The bias of these methods is much larger than that of PCA-BS in most cases.

To turn to the algorithm PCA-RS, it has approximately the same bias as the approaches [87, 21, 65]. Its estimate standard deviation and maximum error are comparable with those of the methods [21, 65], but they are larger than those of PCA-BS and [87]. However, PCA-RS is on average more than 30 times faster than the algorithms [87, 21, 65] and PCA-BS. Compared with the algorithms [95, 9, 3, 89, 76, 78, 24, 96, 7, 83], $s(\sigma_{corr})$ and $\max|\sigma_{corr} - \sigma|$ of PCA-RS are more than 2 times smaller for the noise variances 25 and 65 and more than 1.3 times smaller for the noise variances 100 and 130.

3.7.3 Experiments with MeasTex

All images in the TID2008 database contain small or large homogeneous areas. However, this is not the case for all images one can meet. For this reason, the proposed algorithms have been tested on images containing only textures. The MeasTex texture database, which has been already used in many works on texture analysis [82, 47, 79], has been selected. This database contains 236 real textures stored as 512 × 512 grayscale images. Several images from the database are shown in Fig. 3.6.

The comparison results are presented in Table 3.7. Compared with the other methods, the accuracy of the proposed algorithms is significantly better in all cases. For PCA-BS, the standard deviation of the estimates is always more than 2.8 times smaller; and the maximum error is always more than 2.2 times smaller. The bias of this method is much smaller in most cases as well. PCA-RS has more than 2 times smaller $s(\sigma_{est})$ and $\max|\sigma_{est} - \sigma|$ compared with the state of the art, although it has a considerable positive bias. The results of [87], [21], and [65] are not available.

Table 3.7: The accuracy of the considered methods for MeasTex. $\overline{\sigma}_{est} - \sigma$ is the bias of estimates, $s(\sigma_{est})$ is the standard deviation of estimates, $\max|\sigma_{est} - \sigma|$ is the maximum difference between an estimate and the true value. The last column is the percentage of the images, for which the method cannot estimate the noise level. The best result in a column is selected with the bold font.

Method	$\overline{\sigma}_{est} - \sigma$	$s(\sigma_{est})$	$\max\|\sigma_{est} - \sigma\|$	% of failures
$\sigma = 10$				
PCA-BS	**0.283**	**0.845**	7.235	0

Continued on the next page

Table 3.7 – *Continued from the previous page*

Method	$\overline{\sigma}_{est} - \sigma$	$s(\sigma_{est})$	$\max \lvert\sigma_{est} - \sigma\rvert$	% of failures
PCA-RS	0.511	1.057	**5.525**	0
[87]	–	–	–	–
[95]	1.075	2.418	16.520	0
[9]	0.997	2.746	17.689	0
[3]	6.426	10.334	56.555	0
[89]	1.947	8.118	38.551	68.2
[76]	5.129	5.241	38.756	0.4
[21]	–	–	–	–
[65]	–	–	–	–
[78]	0.489	5.033	37.958	0
[24]	1.662	3.486	21.557	0
[96]	-0.648	2.799	23.646	0.4
[7]	6.917	10.215	75.213	0
[83]	6.014	9.328	47.079	0
$\sigma = 15$				
PCA-BS	0.170	**0.592**	**4.868**	0
PCA-RS	0.419	0.960	4.984	0
[87]	–	–	–	–
[95]	0.836	2.007	13.554	0
[9]	0.571	2.266	14.814	0
[3]	5.709	9.294	48.667	0
[89]	1.858	7.238	34.703	57.6
[76]	3.748	4.698	34.986	1.3
[21]	–	–	–	–
[65]	–	–	–	–
[78]	**-0.053**	4.803	36.573	0
[24]	1.291	2.969	18.605	0
[96]	-0.753	2.772	20.942	0
[7]	8.433	10.797	74.195	0
[83]	5.539	8.388	43.381	0
$\sigma = 20$				
PCA-BS	**0.080**	**0.461**	**3.084**	0
PCA-RS	0.373	0.939	5.397	0
[87]	–	–	–	–
[95]	0.625	1.727	11.382	0
[9]	0.239	1.932	12.540	0
[3]	5.054	8.594	46.388	0
[89]	1.238	7.240	31.465	47.0
[76]	2.801	4.641	40.440	3.0
[21]	–	–	–	–
[65]	–	–	–	–

Continued on the next page

Table 3.7 – *Continued from the previous page*

Method	$\overline{\sigma}_{est} - \sigma$	$s(\sigma_{est})$	$\max \lvert\sigma_{est} - \sigma\rvert$	% of failures
[78]	-0.295	4.656	34.860	0
[24]	1.015	2.573	16.240	0
[96]	-0.870	2.398	20.000	1.3
[7]	10.607	12.944	73.093	0
[83]	5.199	7.615	40.076	0

This experiment also shows the limitations of the proposed approach. Among 236 images in the database, there are 4 images, for which the error of PCA-BS is larger than $3s(\sigma_{est})$. These images are textures of fabric and metal (see Fig. 3.7). In order to examine the properties of these images, the following measure has been computed for each image x in the database:

$$\overline{R}_x = \frac{1}{24}\Big(\sum_{\Delta p_1=0}^{4} \sum_{\Delta p_2=0}^{4} |R_x(\Delta p_1, \Delta p_2)| - |R_x(0,0)| \Big) \tag{3.49}$$

where R_x is the autocorrelation function:

$$R_x(\Delta p_1, \Delta p_2) = \frac{1}{\sigma_x^2 N_x} \sum_{p_1,p_2} (x(p_1,p_2) - \mu_x)(x(p_1+\Delta p_1, p_2+\Delta p_2) - \mu_x). \tag{3.50}$$

Above, the sum is computed over all p_1 and p_2 such that (p_1, p_2) and $(p_1 + \Delta p_1, p_2 + \Delta p_2)$ are inside the image domain, N_x is the number of items in this sum, σ_x^2 is the variance of image x, and μ_x is the mean of image x. $\overline{R}_x$ is always in [0, 1]. It reflects the correlation between neighbor image pixels: $\overline{R}_x = 0$ when x is white noise; and $\overline{R}_x = 1$ when the neighbor pixels in x are in an exact linear dependence. For MeasTex, the average value of $\overline{R}_x$ is 0.6 and the standard deviation is 0.2. For the images shown in Fig. 3.7, $\overline{R}_x$ takes its smallest values: 0.08, 0.07, 0.06, and 0.09. Therefore, these images are the closest to white noise in the database, which explains the largest error of PCA-BS for these images.

3.8 Discussion

The proposed methods do not assume the existence of homogeneous areas in the original image. Instead, they belong to the class of algorithms, which are based on a sparse image representation. One of the first methods from this class is [24], whereas [65] and [21] are more recent.

The method [24] utilizes a wavelet transform. This approach does not work well for images with textures, because textures usually contain high frequencies and affect the finest decomposition level, from which the noise variance is estimated [38]. It was outperformed by other techniques, e.g. [9], [65], [21], [38], and [95].

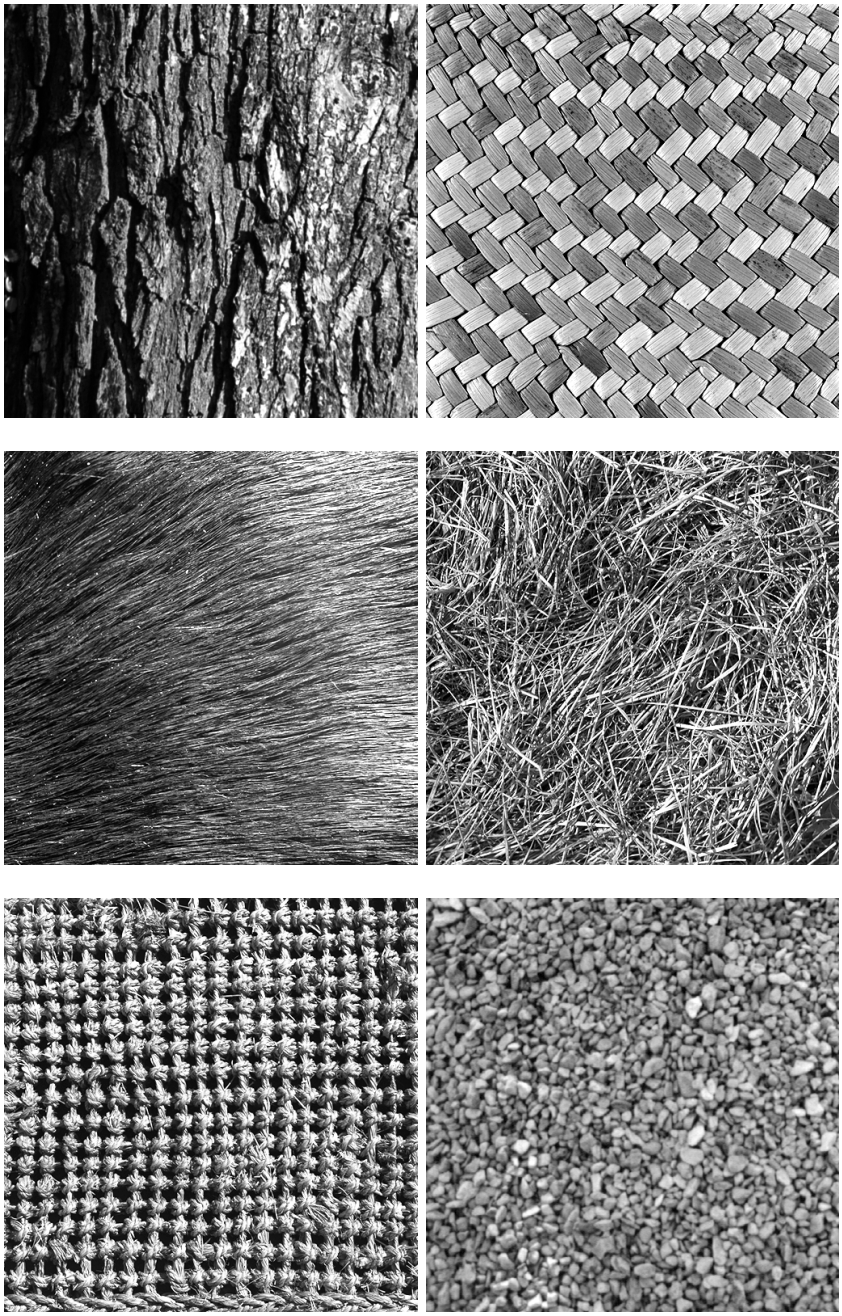

Figure 3.6: Images from the MeasTex database.

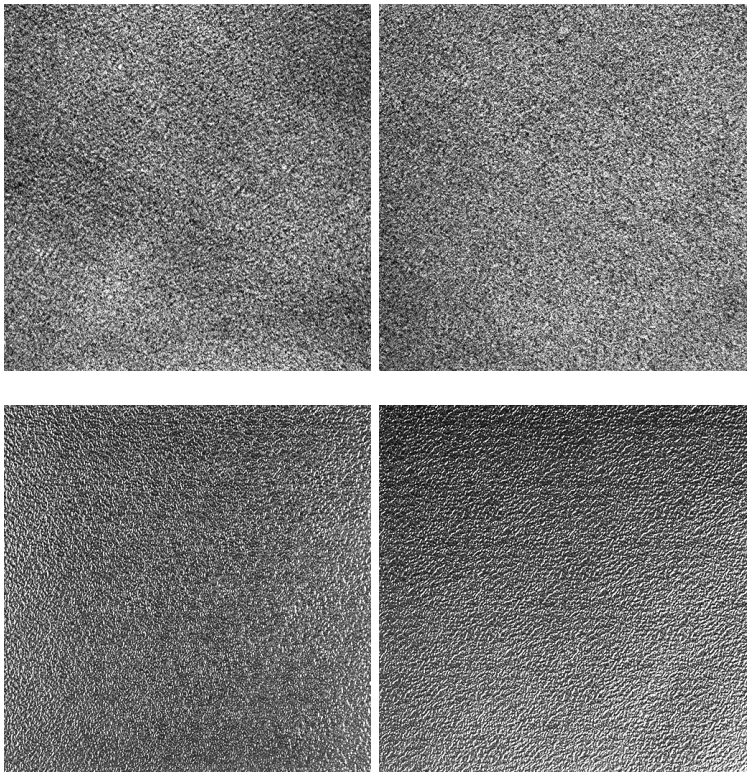

Figure 3.7: Images, for which PCA-BS has an error larger than $3s(\sigma_{est})$. The top row: fabric textures. The bottom row: metal textures.

The method [65] applies 2D DCT of image blocks. It assumes that the image structures occupy only low frequencies and high frequencies contain only noise. Compared with this method, the presented algorithms use PCA instead of DCT. The transform computed by PCA depends on the data in contrast to DCT, which is predefined. Therefore, PCA can efficiently process a larger class of images, including those which contain structures with high frequencies. The evidence can be found in Table 3.5: the maximum error of [65] is more than two times larger than that of PCA-BS, i.e. PCA can handle some images in the TID2008 database much more efficiently than DCT.

Compared with the method [21], which assumes the existence of similar blocks in the image, the proposed approach assumes the correlation between pixels in image blocks. These two assumptions cannot be compared directly, because there are images which satisfy the first and does not satisfy the second and vice versa. Indeed, the experiments with TID2008 demonstrate only a small improvement of the results compared with [21]. A more significant difference is in the execution time: [21] has expensive steps of block matching and image prefiltering, which make it more than 15 times slower than PCA-BS.

3.9 Summary

In this chapter, a new noise level estimation approach implemented in two versions – PCA-BS and PCA-RS – has been presented. The comparison with the several best state of the art methods shows that the both implementations provide good compromises between accuracy and computation time.

Since the proposed methods do not require the existence of homogeneous areas in the original image, they can also be applied to textures. The experiments show that only stochastic textures, whose correlation properties are very close to those of white noise, cannot be successfully processed.

Chapter 4

Signal-dependent noise parameter estimation

In this chapter, the noise parameter estimation method presented in the previous chapter is extended to the case of signal-dependent noise. Particularly, SAR (1.10), MRI (1.12), CCD/CMOS (1.29), and ultrasound/film-grain (1.31) noise models are considered. The idea of the extension is to apply a variance-stabilizing transformation (VST), after which noise can be assumed to be AWGN, and then utilize the noise variance estimation algorithm from the previous chapter. However, the VST depends on the noise parameters, which are unknown. Therefore, the method is extended in such a way that it can check if noise obeys the normal distribution, which allows correct selection of the VST.

This chapter is organized as follows. First, VSTs are briefly described in Section 4.1. The algorithm for VST parameter selection is presented in Section 4.2. This algorithm forms the basis of the model-specific noise parameter estimation methods expounded in Section 4.3. The experiments and the discussion are given in Sections 4.4 and 4.5 respectively. The chapter is summarized in Section 4.6.

4.1 Variance-stabilizing transformations

In this section, the construction of a VST for an arbitrary noise distribution [67] is briefly described.

The goal is to construct transformation $g(t; \mathbf{w})$ such that the standard deviation of the transformed pixel value equals 1 (or some other constant):

$$\operatorname{std}\big(g(y(\mathbf{p}); \mathbf{w})\big) = 1. \tag{4.1}$$

Using the first-order Taylor expansion of g around the mean value

$$g(y; \mathbf{w}) \approx g(\mathbf{E}(y); \mathbf{w}) + g'(\mathbf{E}(y); \mathbf{w})(y - \mathbf{E}(y)) \tag{4.2}$$

(4.1) can be rewritten as

$$g'(\mathbf{E}(y);\mathbf{w})\mathrm{std}(y) = 1 \tag{4.3}$$

or, using function $h(\cdot;\mathbf{w})$,

$$g'(\mathbf{E}(y);\mathbf{w})h(\mathbf{E}(y);\mathbf{w}) = 1. \tag{4.4}$$

Therefore,

$$g(t;\mathbf{w}) = \int \frac{dt}{h(t;\mathbf{w})}. \tag{4.5}$$

VSTs g for models (1.10), (1.12), (1.29), and (1.31) can be easily computed by substituting corresponding functions h, which are listed in Table 1.1, into (4.5). These VSTs are listed in Table 4.1. Since images $g(y;\mathbf{w})$ are utilized for noise variance estimation, the additive constants appearing after the integration in (4.5) do not influence anything, hence they are neglected.

Table 4.1: The signal-dependent noise models and the VSTs. The asymptotic expressions are given for $t \to +\infty$.

Noise type	$\mathbf{w}$	$h(t;\mathbf{w})$	$g(t;\mathbf{w})$
SAR	a	at	$\dfrac{\ln t}{a}$
MRI	a	$\sqrt{a^2 + O(t^{-2})}$	$\approx \dfrac{t}{a}$
		$\sqrt{a^2 - \dfrac{a^4}{2t^2} + O(t^{-4})}$	$\approx \sqrt{\dfrac{t^2}{a^2} - \dfrac{1}{2}}$
CCD/CMOS	(a,b)	$\sqrt{at+b}$	$\dfrac{2}{a}\sqrt{at+b}$
Ultrasound, film-grain	(a,b)	at^b	$\begin{cases} t/a, & \text{if } b = 0 \\ \frac{t^{1-b}}{a(1-b)}, & \text{if } b \in (0,1) \\ \frac{\ln t}{a}, & \text{if } b = 1 \end{cases}$

As discussed in Chapter 1, the noise in the input image can be approximated by additive Gaussian noise with signal-dependent variance for the considered noise models. After applying the VST, the noise variance becomes signal-independent, and the noise can be assumed to be AWGN. For example, let us consider SAR noise model (1.10). Since $\ln(1+t) \approx t$ for sufficiently small t,

$$\begin{aligned} g(y;a) &= \frac{\ln y}{a} = \frac{\ln(x(1+an))}{a} \\ &= \frac{\ln x + \ln(1+an)}{a} \approx \frac{\ln x + an}{a} = \frac{\ln x}{a} + n \end{aligned} \tag{4.6}$$

That means, transformed noisy image $(\ln y)/a$ can be seen as the sum of transformed original image $(\ln x)/a$ and AWGN n.

4.2 Selection of the VST parameters

The VST parameter selection algorithm introduced in this section is independent of the noise model; and it is used in the model-specific noise parameter estimation methods described in Section 4.3.

4.2.1 Noise normality assessment

If Assumption 1 holds, the noise normality can be measured by comparing the distribution of $\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\}$ with the normal distribution, which can be done in several ways:

1. Measuring the difference between the standardized moments of $\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\}$ and the moments of the standard normal distribution. Usually, the third (skewness) and the forth (excess kurtosis) standardized moments are utilized (JarqueBera test [42], D'Agostino's K-squared test [19]). The computation of the moments of $\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\}$ requires linear time.

2. Measuring the difference between the empirical CDF and the normal CDF (the Lilliefors test [53], the Anderson-Darling test [6], the Cramér-von Mises criterion [17]). The empirical CDF is defined as

$$F_N(t) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{C(i)}(t) \tag{4.7}$$

   where $C(i) = (-\infty, \tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i]$ and $\mathbf{1}_X$ is the indicator function of set X. Its computation requires sorting $\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\}$, which has computational complexity $\Theta(N \log N)$.

During the experiments, it was found that the methods based on the empirical CDF does not improve the noise parameter estimation accuracy compared with the methods based on the moments, whereas the time to compute the empirical

CDF is larger. It was also found that, when using the moment-based measures, it is enough to utilize only the forth standardized moment, i.e. the excess kurtosis.

The excess kurtosis of random variable X is

$$\gamma(X) = \frac{\mathbf{E}\big((X-\mu_X)^4\big)}{\mathbf{E}\big((X-\mu_X)^2\big)^2} - 3 \tag{4.8}$$

where μ_X is the mean value of X. If X has the normal distribution, $\gamma(X) = 0$. Excess kurtosis estimator G_2 from [43] is used in the presented algorithm. For sample $\{X_1, \ldots, X_N\}$ of size N, this estimator is calculated as follows:

$$G_2 = \frac{(N-1)((N+1)g_2 + 6)}{(N-2)(N-3)} \tag{4.9}$$

where

$$\begin{aligned} g_2 &= \frac{m_4}{m_2^2} - 3 \\ m_k &= \frac{1}{N}\sum_{i=1}^{N}(X_i - \overline{X})^k \\ \overline{X} &= \frac{1}{N}\sum_{i=1}^{N} X_i. \end{aligned} \tag{4.10}$$

This estimator has the following properties for normally distributed random variable X:

$$\mathbf{E}(G_2) = 0 \tag{4.11}$$

$$\lim_{N\to+\infty} \operatorname{std}(G_2)\sqrt{N} = 2\sqrt{6}. \tag{4.12}$$

Therefore,

$$K\big(\{\mathbf{y}_i\}\big) = -|G_2\big(\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\}\big)|\sqrt{|\{\mathbf{y}_i\}|} \tag{4.13}$$

can be used as a measure of the noise normality, and a necessary condition for the noise normality can be written as

$$K\big(\{\mathbf{y}_i\}\big) > -T_\gamma \tag{4.14}$$

where $T_\gamma > 0$ is a fixed threshold.

Meanwhile, Assumption 1 is not always satisfied for the whole set of the image blocks. For this reason, measure $K(\cdot)$ and condition (4.14) are used without checking Assumption 1, but the procedure is designed in such a way that a wrong result is unlikely for real images:

1. If (4.14) is satisfied, the distribution of $\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\}$ is normal. If Assumption 1 does not hold, this distribution is affected by both the original image and the noise:
$$\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i = \tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{x}_i + \tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}(\mathbf{y}_i - \mathbf{x}_i) \tag{4.15}$$

where vector $\mathbf{x}_i$ corresponds to original image x and vector $(\mathbf{y}_i - \mathbf{x}_i)$ corresponds to noise $(y - x)$. Provided that the original image has no stochastic textures, values $\tilde{\mathbf{v}}^{\mathrm{T}}_{\mathbf{Y},M}\mathbf{x}_i$ are unlikely to be normally distributed. That means the contribution from the original image $\tilde{\mathbf{v}}^{\mathrm{T}}_{\mathbf{Y},M}\mathbf{x}_i$ should not have significant influence, because the distribution of $\tilde{\mathbf{v}}^{\mathrm{T}}_{\mathbf{Y},M}\mathbf{y}_i$ is normal. Therefore, it can be assumed that the normality of $\tilde{\mathbf{v}}^{\mathrm{T}}_{\mathbf{Y},M}\mathbf{y}_i$ implies the normality of $\tilde{\mathbf{v}}^{\mathrm{T}}_{\mathbf{Y},M}(\mathbf{y}_i - \mathbf{x}_i)$, i.e. the normality of the noise.

2. If (4.14) is not satisfied, no conclusions are made about the noise distribution. Instead, a part of the image blocks is skipped as described in Section 3.5.1 and (4.14) is evaluated again.

However, the larger the block subset used in (4.14) is, the more reliable this check is, because a small block subset may not cover the whole intensity range, and the dependence of the noise variance on the original image intensity may not be detected. Hence, when two block subsets are given, the following steps are used in order to select the subset, which is more likely to have AWGN:

1. check (4.14) for both subsets, select the subset for which (4.14) holds;

2. if (4.14) holds for both subsets, select the larger subset;

3. if the subsets have equal size, select one with larger $K(\cdot)$.

That means, only subsets satisfying necessary condition (4.14) are considered; and the subset size has higher priority than $K(\cdot)$.

4.2.2 Algorithm

Let us consider VST $g(t; \mathbf{w})$. The problem is to select $\mathbf{w}$, for which transformed image $g(y; \mathbf{w})$ has AWGN. Since $g(y, \mathbf{w})$ can have AWGN for several values of $\mathbf{w}$, parameter vector $\mathbf{w}$ is computed as a function of vector $\mathbf{q} \in Q$, whose dimension can be smaller than the dimension of $\mathbf{w}$:

$$\mathbf{w} = \mathbf{w}(\mathbf{q}) \quad \mathbf{q} \in Q \tag{4.16}$$

Function $\mathbf{w} = \mathbf{w}(\mathbf{q})$ depends on a particular noise model and its explicit expressions are given in the corresponding sections (Section 4.3.3 and Section 4.3.4). It is selected in such a way that transformed image $g(y, \mathbf{w}(\mathbf{q}))$ has AWGN only for one value of $\mathbf{q}$. By using function $\mathbf{w}(\mathbf{q})$, the dimension of the search space can be reduced, which makes the parameter selection algorithm more computationally efficient.

Therefore, the problem is to select $\mathbf{q}$, for which $y' = g(y; \mathbf{w}(\mathbf{q}))$ has AWGN. It is considered as an optimization problem: an objective function, which implements the selection strategy described in the previous section, is maximized on Q.

Algorithm 5 `ComputeNormality`

Input: image $y(\mathbf{p})$, parameter $\mathbf{q} \in Q$
Output: normality measure

$y' \leftarrow g(y; \mathbf{w}(\mathbf{q}))$
$p \leftarrow 1$
while $p \geq p_{min}$ **do**
  $\tilde{\mathbf{v}}_{\mathbf{Y},i},\ \tilde{\lambda}_{\mathbf{Y},i} \leftarrow$ `ApplyPCA`$(\ B(p, y')\ )$
  $K(B(p, y')) \leftarrow -|G_2(\{\mathbf{v}_{\mathbf{Y},M}^{\mathrm{T}} \mathbf{y}_i\})| \sqrt{|B(p, y')|}$
  if $K(B(p, y')) > -T_\gamma$ **then**
    return $|B(p, y')| + K(B(p, y'))/T_\gamma$
  end if
  $p \leftarrow p - \Delta p$
end while
return -1

The objective function is calculated by the algorithm `ComputeNormality`, which looks for the largest block subset $B(p, y')$, for which (4.14) holds, and returns value

$$|B(p, y')| + K(B(p, y'))/T_\gamma. \tag{4.17}$$

The first term of (4.17) is an integer, and the second term is in $(-1, 0]$, since the algorithm returns only when $K(B(p, y')) \in (-T_\gamma, 0]$. Therefore, objective function values are ordered first by $|B(p, y')|$ and then by $K(B(p, y'))$, which is exactly what is needed in the previous section. The values of all algorithm parameters are given in Section 4.4.2.

Since $|B(p, y')|$ and $K(B(p, y'))$ as functions of $\mathbf{q}$ are not guaranteed to have a single local maximum, a two-step optimization procedure is used:

1. grid search to localize the maximum.

2. a local optimization technique to compute the maximum precisely. The local optimizer should not use objective function derivatives or assume objective function continuity.

Implementation details of the optimization procedure depend on a particular noise model and are described in the corresponding sections (Section 4.3.3 and Section 4.3.4).

Note that an image block is skipped if VST $g(y; \mathbf{w}(\mathbf{q}))$ is undefined in one of its pixels. Since the domain of $g(y; \mathbf{w}(\mathbf{q}))$ depends on $\mathbf{q}$, there is different number of blocks to process for different $\mathbf{q}$. Therefore, $|B(p, y')|$ depends not only on p, but also on $\mathbf{q}$.

4.2.3 Efficient implementation

Computation of projections $\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}} \mathbf{y}_i$ and excess kurtosis estimate G_2 is the most computationally expensive part of the algorithm `ComputeNormality`; and the

aim is to compute $\tilde{\mathbf{v}}^{\mathrm{T}}_{\mathbf{Y},M}\mathbf{y}_i$ and G_2 as rare as possible. Since the optimization consists of consecutive runs of the algorithm `ComputeNormality`, one can use the block subset size N_{prev} computed at the previous execution of this algorithm. If current block subset size $|B(p, y')|$ is smaller than N_{prev}, the current objective function value will not be larger than the previous one, and one can stop without computation of $\tilde{\mathbf{v}}^{\mathrm{T}}_{\mathbf{Y},M}\mathbf{y}_i$ and G_2. This method is implemented in the algorithm `ComputeNormalityFast`. $N_{prev} = 0$ is passed at the first execution.

Algorithm 6 `ComputeNormalityFast`

Input: image $y(\mathbf{p})$,
parameter $\mathbf{q} \in Q$,
previous block subset size N_{prev}
Output: normality measure

$y' \leftarrow g(y; \mathbf{w}(\mathbf{q}))$
$p \leftarrow 1$
while $p \geq p_{min}$ **do**
  if $|B(p, y')| \geq N_{prev}$ **then**
    $\tilde{\mathbf{v}}_{\mathbf{Y},i},\ \tilde{\lambda}_{\mathbf{Y},i} \leftarrow$ `ApplyPCA`$(\ B(p, y')\ )$
    $K(B(p, y')) \leftarrow -|G_2(\{\mathbf{v}^{\mathrm{T}}_{\mathbf{Y},M}\mathbf{y}_i\})|\sqrt{|B(p, y')|}$
    if $K(B(p, y')) > -T_\gamma$ **then**
      return $|B(p, y')| + K(B(p, y'))/T_\gamma$
    end if
  end if
  $p \leftarrow p - \Delta p$
end while
return -1

When using this speedup technique, it is good to reach a large value of N_{prev} (i.e. a large value of the objective function) as early as possible. During the grid search, it can be done by starting with elements of Q, which are far away from each other. When Q is a one-dimensional set, i.e. $\mathbf{q}$ is a scalar, the grid search optimizer can choose the largest possible step between elements of Q and then divide the step by 2. For example, if $Q = \{0, 1, \ldots, 8\}$, its elements are enumerated as follows:

1. $\mathbf{q} = 0, 8$
2. $\mathbf{q} = 0, 4, 8$
3. $\mathbf{q} = 0, 2, 4, 6, 8$
4. $\mathbf{q} = 0, 1, 2, 3, 4, 5, 6, 7, 8$

In practice, executions of the algorithm `ComputeNormalityFast` with the same argument $\mathbf{q}$ are skipped, hence this enumeration method provides no overhead. Meanwhile, it helps to quickly cover different parts of set Q and to have a large N_{prev} from the beginning.

Additionally, efficient computation of the sample covariance matrix $S_{\mathbf{Y}}$ described in Section 3.5.3 is applied in the algorithm `ComputeNormalityFast`.

4.3 Model-specific parameter estimation

4.3.1 SAR noise parameter estimation

Since $g(t;a) = (\ln t)/a$, the noise in the image $\ln y$ is assumed to be AWGN with standard deviation a. Therefore, the VST parameter selection procedure is not needed here, and the estimate of a is computed as

$$a_{est} = \sqrt{\texttt{EstimateNoiseVariance}(\ln y)}. \tag{4.18}$$

4.3.2 MRI noise parameter estimation

For model (1.12), the following two-step estimation procedure is used.

At the first step, the fact that the noise can be approximated by AWGN with variance a^2 for large $x(\mathbf{p})$ is used, and the first estimate is computed as

$$a_{est}^{(1)} = \sqrt{\texttt{EstimateNoiseVariance}(y)}. \tag{4.19}$$

At the second step, the following approximation for h is utilized:

$$h(t;a) = \sqrt{a^2 - \frac{a^4}{2t^2} + O\big(1/t^4\big)} \tag{4.20}$$

for large $x(\mathbf{p})$. The corresponding VST is

$$g(t;a) = \sqrt{\frac{t^2}{a^2} - \frac{1}{2}}. \tag{4.21}$$

Note that the VST depends on noise parameter a, which is unknown. Using estimate $a_{est}^{(1)}$ computed in the first step, the final estimate is computed as

$$a_{est} = \sqrt{\texttt{EstimateNoiseVariance}\big(g(y;a_{est}^{(1)})\big)}. \tag{4.22}$$

At both steps, the minimal object gray value is estimated as the isodata threshold computed for image y; and only the blocks, whose average gray value is larger than this threshold, are processed. This is necessary because the expression for $h(t;a)$ is asymptotic and has low accuracy for small $x(\mathbf{p})$.

4.3.3 CCD/CMOS noise parameter estimation

In contrast to SAR and MRI noise, computation of the VST parameters for noise model (1.29) is more complex.

Let us rewrite the VST in the polar coordinates:

$$\begin{aligned} a &= r\cos\phi \\ b &= r\sin\phi \end{aligned} \tag{4.23}$$

$$g(t; r\cos\phi, r\sin\phi) = \frac{2}{\cos\phi\sqrt{r}}\sqrt{t\cos\phi + \sin\phi}. \tag{4.24}$$

Since $a > 0$, $\phi \in (-\pi/2, \pi/2)$. If the noise in image $g(y; r\cos\phi, r\sin\phi)$ is AWGN, then the noise in image $g(y; \cos\phi, \sin\phi)$ is also AWGN and vise versa, because these two images differ only by factor $\sqrt{r}$. Hence ϕ is selected by maximizing the normality measure computed by the algorithm `ComputeNormalityFast` for image $g(y; \cos\phi, \sin\phi)$. Here, $\mathbf{q} = \phi$ and $\mathbf{w}(\mathbf{q}) = (\cos\phi, \sin\phi)$ (recall that, as given in Table 4.1, $\mathbf{w} = (a, b)$ for this noise model).

Let us define a grid with step $\pi/2^n$ on set $(-\pi/2, \pi/2)$:

$$\Phi_n = \Big(-\frac{\pi}{2}, \frac{\pi}{2}\Big) \cap \Big\{\frac{k\pi}{2^n} | k \in \mathbb{Z}\Big\}. \tag{4.25}$$

The maximization is organized as follows:

1. grid search in set Φ_6, which gives estimate $\phi_{est}^{(1)}$
2. grid search in set
$$\Phi_8 \cap \Big[\phi_{est}^{(1)} - \frac{\pi}{2^6}, \phi_{est}^{(1)} + \frac{\pi}{2^6}\Big] \tag{4.26}$$
which gives estimate $\phi_{est}^{(2)}$
3. golden section search in set
$$\Big(-\frac{\pi}{2}, \frac{\pi}{2}\Big) \cap \Big[\phi_{est}^{(2)} - \frac{\pi}{2^8}, \phi_{est}^{(2)} + \frac{\pi}{2^8}\Big] \tag{4.27}$$
which gives final estimate ϕ_{est}. The search stops when the length of the search interval is smaller than 10^{-4}.

Fig. 4.2 shows that the maximization has a well-defined solution even for a highly textured image.

When angle ϕ_{est} is found, the noise in image $g(y; \cos\phi_{est}, \sin\phi_{est})$ is assumed to be AWGN with standard deviation $\sqrt{r}$; and the polar radius can be estimated as

$$r_{est} = \texttt{EstimateNoiseVariance}(g(y; \cos\phi_{est}, \sin\phi_{est})). \tag{4.28}$$

Then, final estimates a_{est} and b_{est} are

$$\begin{aligned} a_{est} &= r_{est}\cos\phi_{est} \\ b_{est} &= r_{est}\sin\phi_{est}. \end{aligned} \tag{4.29}$$

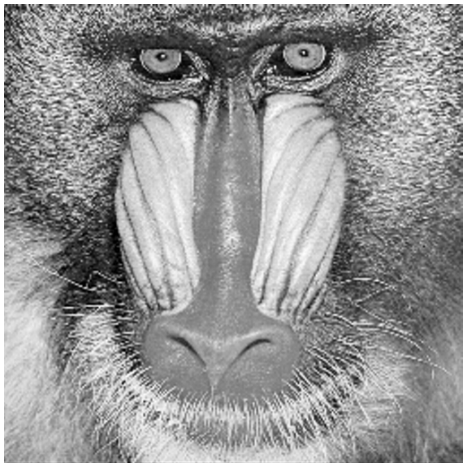

Figure 4.1: The 512×512 "Baboon" test image.

4.3.4 Ultrasound/film-grain noise parameter estimation

The parameter estimation procedure for noise model (1.31) is similar to that for noise model (1.29).

From the form of the VST, if image $g(y; a, b)$ has AWGN, image $g(y; 1, b)$ has AWGN and vise versa, because these two images differ only by factor a. Therefore, the maximization of the normality measure computed by the algorithm `ComputeNormalityFast` for image $g(y; 1, b)$ is used to select b. For this noise model, $\mathbf{q} = b$ and $\mathbf{w}(\mathbf{q}) = (1, b)$.

Let us define a grid with step $1/2^n$ on set $(0, 1]$:

$$B_n = (0, 1] \cap \Big\{ \frac{k}{2^n} | k \in \mathbb{Z} \Big\}. \tag{4.30}$$

The maximization is organized as follows:

1. grid search in set B_4, which gives estimate $b_{est}^{(1)}$
2. grid search in set
$$B_6 \cap \Big[b_{est}^{(1)} - \frac{1}{2^4}, b_{est}^{(1)} + \frac{1}{2^4} \Big] \tag{4.31}$$
which gives estimate $b_{est}^{(2)}$
3. golden section search in set
$$(0, 1] \cap \Big[b_{est}^{(2)} - \frac{1}{2^6}, b_{est}^{(2)} + \frac{1}{2^6} \Big] \tag{4.32}$$
which gives final estimate b_{est}. The search stops when the length of the search interval is smaller than 10^{-3}.

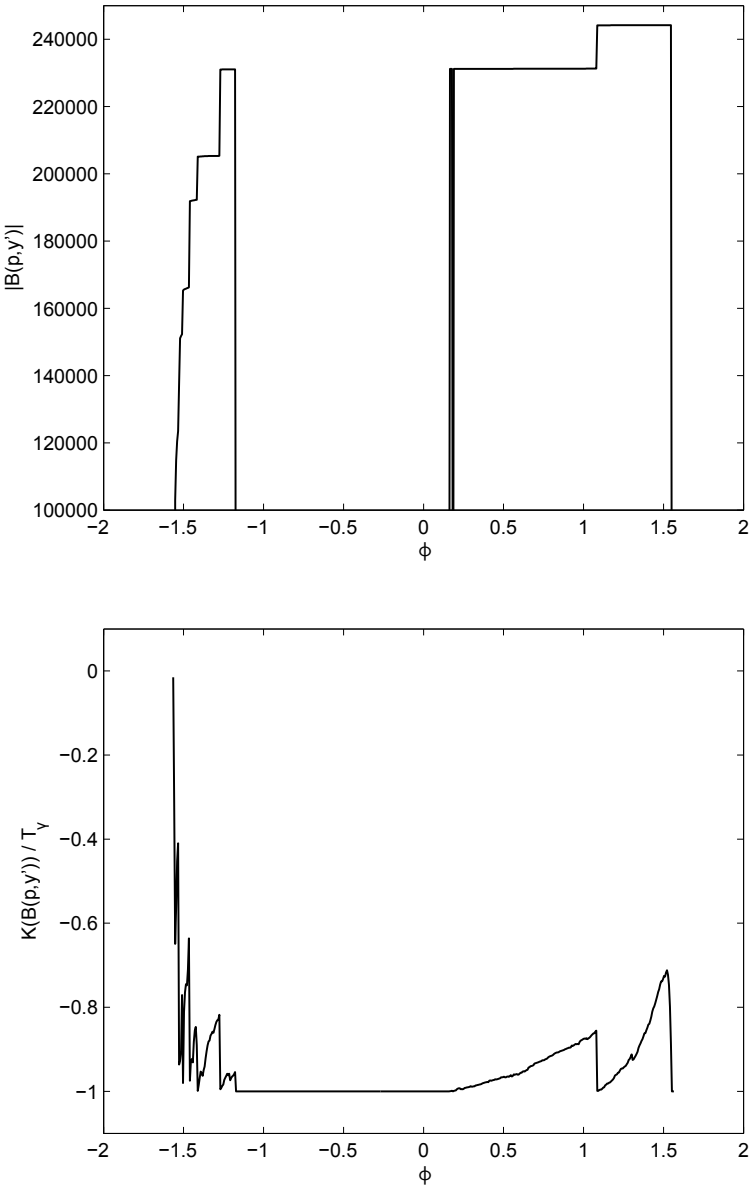

Figure 4.2: $|B(p, y')|$ and $K(B(p, y'))/T_\gamma$ as functions of ϕ for the "Baboon" image (see Fig. 4.1) and noise parameters $a = 1$, $b = 16$. The true value of ϕ is $\arctan b/a \approx 1.508$. The estimated value of ϕ is 1.522.

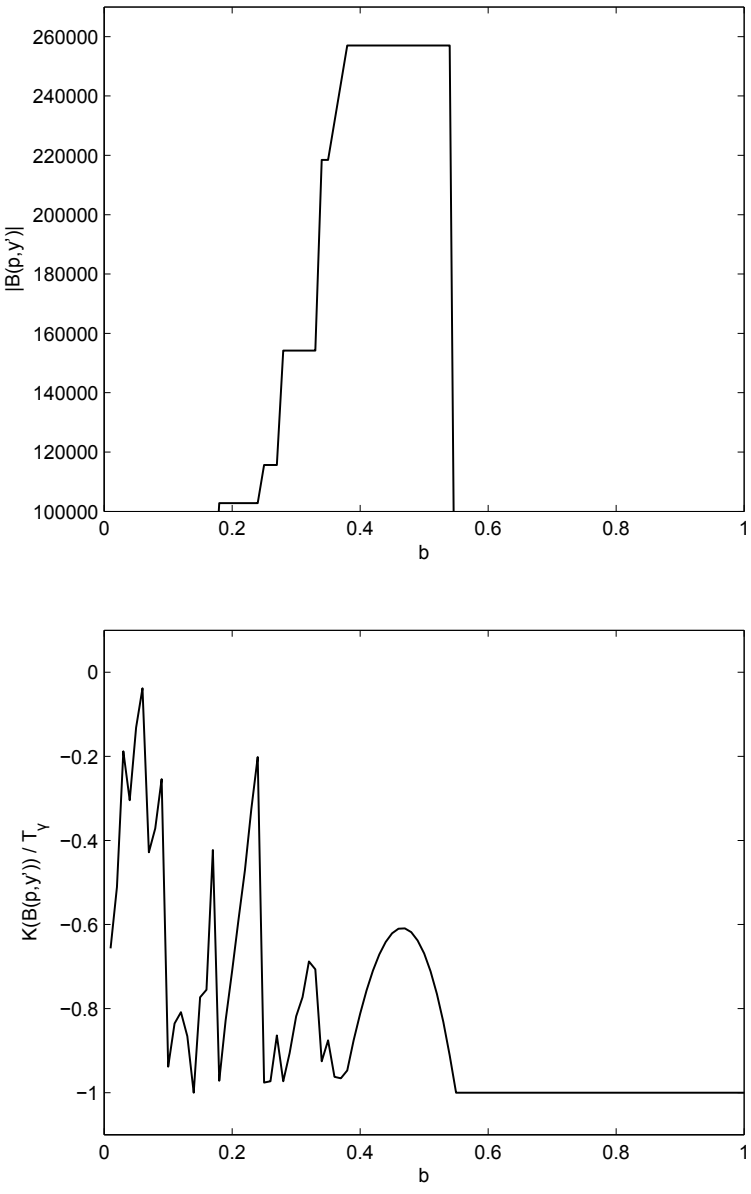

Figure 4.3: $|B(p, y')|$ and $K(B(p, y'))/T_\gamma$ as functions of b for the "Baboon" image (see Fig. 4.1) and noise parameters $a = 1$, $b = 0.5$. The estimated value of b is 0.465.

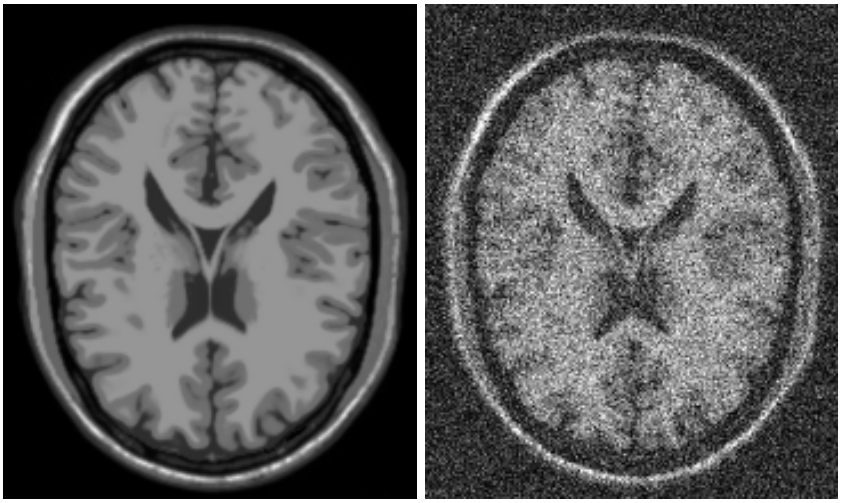

Figure 4.4: The synthetic T1 8-bit image from the Brainweb database (slice 90). Left: the original image. Right: the noisy image with noise level a equal to 15% of the maximum possible grayvalue.

The summands of the objective function are plotted in Fig. 4.3. As one can see, the maximum is well-defined.

When b_{est} is found, it is assumed that image $g(y; 1, b_{est})$ has AWGN with standard deviation a. Therefore, a can be estimated as

$$a_{est} = \sqrt{\texttt{EstimateNoiseVariance}(g(y; 1, b_{est}))}. \tag{4.33}$$

4.4 Experiments

4.4.1 Material

In order to evaluate the algorithm for models (1.10), (1.29), and (1.31), the TID2008 database [64] has been used (see Section 3.7.2). For model (1.12), a synthetic T1 8-bit image of size $181 \times 217 \times 181$ from the Brainweb database [13, 49] shown in Fig. 4.4 has been utilized. This image has been already used for the evaluation of Rician noise level estimators in [16, 30]. The computation time has been measured on a PC with CPU Intel i7 920 2.67 GHz and 3 GB RAM using a single-threaded program.

4.4.2 Choice of the parameters

The set of parameters shown in Table 4.2 has been found applicable to all considered noise models. Parameters M_1, M_2, m, Δp, and p_{min} are the same as those in the algorithm `EstimateNoiseVarianceIterative` (see Table 3.3). M_3 is set to 1, which means that 2D blocks are used for processing of 3D volumes,

Table 4.2: Algorithm parameters

Parameter	Denotation	Value
Block width	M_1	5
Block height	M_2	5
Block slice count	M_3	1
Noise subspace dimension in Assumption 1	m	7
Threshold in check (3.24)	T_λ	49
Threshold in check (4.14)	T_γ	$6\sqrt{6}$
Block part step	Δp	0.05
Minimal block part	p_{min}	0.06

i.e. volumes are processed slice by slice. Since the algorithm `EstimateNoiseVariance` does not use an initial noise level estimate, σ^2 in check (3.24) in this algorithm is usually larger than in `EstimateNoiseVarianceIterative`. Hence threshold T_λ is slightly smaller compared with that in Table 3.3. Threshold T_γ is three times larger than the standard deviation of $G_2\big(\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\}\big)\sqrt{|B(p,y')|}$ (see (4.12)), which means that the probability that check (4.14) fails for Gaussian noise is

$$\begin{aligned} P\big(K(\{\mathbf{y}_i\}) \le -T_\gamma\big) &= P\big(|G_2(\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\})|\sqrt{|B(p,y')|} \ge T_\gamma\big) \\ &\le \frac{\mathrm{var}\big(G_2(\{\tilde{\mathbf{v}}_{\mathbf{Y},M}^{\mathrm{T}}\mathbf{y}_i\})\sqrt{|B(p,y')|}\big)}{T_\gamma^2} \to \frac{1}{9} \quad |B(p,y')| \to +\infty \end{aligned} \tag{4.34}$$

from Chebyshev's inequality.

4.4.3 Measurement of the accuracy

For one-parameter models (1.10) and (1.12), the relative error

$$\eta = \frac{a_{est} - a}{a} \tag{4.35}$$

has been computed for each noisy image y. In order to measure the overall accuracy of an algorithm, the sample mean $\overline{\eta}$ and the sample standard deviation $s(\eta)$ of the relative error over all noisy images have been calculated.

For two-parameter models (1.29) and (1.31), the relative errors of a and b do not always provide a good measure of the accuracy. For example, for constant image $y(\mathbf{p}) = y_0$ and model (1.29), the noise standard deviation is the same for all a and b satisfying equality $ay_0 + b = const$. Hence seminorm ρ_x of function $h(\cdot; a, b)$ is introduced:

$$\rho_x\big(h(\cdot; a, b)\big) = \sqrt{\sum_{\mathbf{p}} h^2(x(\mathbf{p}); a, b)}. \tag{4.36}$$

The summation is done over all pixels in the image domain. Then, the following relation of seminorms, which mimics the relative error, is computed:

$$\delta = \frac{\rho_x\big(h(\cdot; a_{est}, b_{est}) - h(\cdot; a, b)\big)}{\rho_x\big(h(\cdot; a, b)\big)} \tag{4.37}$$

As in the previous case, the sample mean $\overline{\delta}$ and the sample standard deviation $s(\delta)$ over all noisy images have been computed in order to measure the overall accuracy of an algorithm. But in contrast to (4.35), measure (4.37) is always positive and cannot show the bias of the estimates.

4.4.4 SAR noise parameter estimation

For model (1.10), the proposed algorithm has been compared with the first algorithm from [51]. The results are presented in Table 4.3. As one can see, the proposed method has a smaller bias of the estimates, and the standard deviation of the estimates is always more than 4 times smaller. The average computation time of the proposed method implemented in C++ was 169 ms.

Table 4.3: Results for model (1.10). The better result is selected with the bold font.

True a	$\overline{\eta}$		$s(\eta)$	
	[51]	proposed	[51]	proposed
0.05	0.59	**0.07**	0.62	**0.05**
0.10	0.19	**0.05**	0.24	**0.03**
0.15	0.13	**0.06**	0.13	**0.02**
0.20	0.09	**0.08**	0.09	**0.02**

4.4.5 MRI noise parameter estimation

For model (1.12), the presented algorithm has been compared with the methods [16, 30, 81]. The results for each noise level have been obtained using 100 noise realizations. As in [16] and [30], the true values of noise parameter a have been computed as percents of the maximum possible grayvalue:

$$a_k = 255\frac{k\%}{100\%} \quad k = 1, \ldots, 15. \tag{4.38}$$

The results are presented in Table 4.4 and Figure 4.5. The method [81] significantly overestimates a for all noise levels. The bias of the methods [16] and [30] changes gradually from positive for the low noise levels to negative for the high noise levels. Hence these methods have almost no bias for the middle noise levels. However, when the noise level is 1%, the overestimation is

significant: $\overline{\eta} = 4.3\%$ for [16] and $\overline{\eta} = 8.1\%$ for [30]. For the proposed method, the absolute value of bias is always less than 1.3%, and the borders of segment $[\overline{\eta} - 3s(\eta); \overline{\eta} + 3s(\eta)]$ are always within 3%. The average computation time of the proposed method implemented in C++ was 6.5 seconds.

Table 4.4: Results for model (1.12). The best result is selected with the bold font.

True noise level, %	$\overline{\eta} \times 10^3$				$s(\eta) \times 10^3$			
	[81]	[16]	[30]	proposed	[81]	[16]	[30]	proposed
1	37	43	81	**-12**	2	3	**1**	6
2	28	12	20	**-8**	2	3	**1**	4
3	28	**5**	6	-10	2	3	**1**	3
4	28	3	**1**	-9	2	3	**1**	4
5	32	**1**	-2	-5	3	3	**1**	4
6	39	**0**	-4	-1	3	3	**1**	3
7	49	**0**	-6	1	3	3	**1**	3
8	58	**0**	-7	2	3	3	**1**	2
9	67	**-1**	-9	**1**	3	3	**1**	**1**
10	73	**-1**	-9	**1**	3	3	**1**	**1**
11	80	-2	-10	**1**	3	3	**1**	**1**
12	87	-2	-10	**0**	3	3	**1**	**1**
13	93	-2	-10	**0**	3	3	**1**	**1**
14	101	-2	-11	**-1**	3	3	**1**	**1**
15	109	-3	-12	**-2**	3	4	2	**1**

4.4.6 CCD/CMOS noise parameter estimation

For model (1.29), the proposed algorithm has been compared with the method [32]. The pedestal level p_0 (see Section 1.4) has been set to 0. The results are presented in Table 4.5. For the proposed algorithm, both $\overline{\delta}$ and $s(\delta)$ are always at least 3 times smaller. The average computation time of the proposed method implemented in C++ was 24 seconds.

4.4.7 Ultrasound/film-grain noise parameter estimation

For model (1.31), the proposed algorithm has been compared with the method [86]. Because some combinations of fixed parameters a and b do not give realistic noise levels for this model, b and the peak signal-to-noise ratio (PSNR) have been fixed. For 8-bit images, PSNR is computed as

$$PSNR = 20 \log_{10} \frac{255}{s(y - x)} = 20 \log_{10} \frac{255}{a\, s(x^b n)}. \tag{4.39}$$

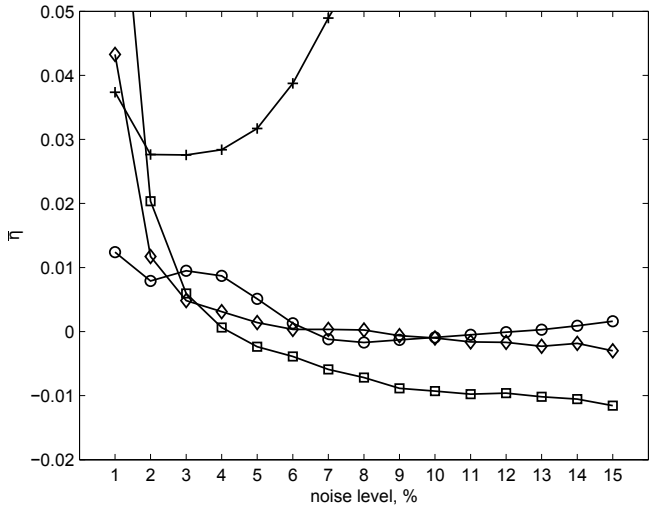

Figure 4.5: Bias $\overline{\eta}$ for model (1.12). +: method [81], ◊ : method [16], □: method [30], ◯: proposed method.

Let μ_x be the average value of image x^{2b}. Then sample standard deviation $s(x^b n)$ equals $\sqrt{\mu_x}$ [36], and a can be computed as

$$a = \frac{255}{\sqrt{\mu_x} 10^{PSNR/20}}. \tag{4.40}$$

The results presented in Table 4.6 show that the accuracy of the proposed method is higher in most cases. Particularly, the small standard deviation of its error indicates that it can handle all images in the database. The average computation time of the proposed method implemented in C++ was 8.6 seconds.

4.5 Discussion

Compared with the scatterplot approach, which is the state of the art in signal-dependent noise parameter estimation, the presented method utilizes a different assumption about the input image, because it is an extension of the PCA-based approach described in the previous chapter. Namely, it selects a block subset, which allows a sparse representation, instead of looking for homogeneous areas. Hence images containing only textures can be efficiently processed.

Since the algorithm is based on the VST, its basic procedure is independent of the noise model. In this chapter, applicability of the method to 4 most common noise models, which have 1 or 2 parameters, is demonstrated, but noise models with 3 and more parameters can be handled as well.

Table 4.5: Results for model (1.29). The better result is selected with the bold font.

True a	True b	$\overline{\delta}$		$s(\delta)$	
		[32]	proposed	[32]	proposed
0.5	16	0.27	**0.03**	0.36	**0.02**
0.5	64	0.17	**0.03**	0.24	**0.04**
1.0	16	0.17	**0.02**	0.26	**0.01**
1.0	64	0.13	**0.03**	0.19	**0.04**
2.0	16	0.10	**0.03**	0.15	**0.04**
2.0	64	0.09	**0.03**	0.15	**0.03**

Table 4.6: Results for model (1.31). The better result is selected with the bold font.

True PSNR (dB)	True b	$\overline{\delta}$		$s(\delta)$	
		[86]	proposed	[86]	proposed
27	0.25	**0.03**	**0.03**	0.02	**0.01**
27	0.50	0.04	**0.03**	0.04	**0.01**
27	0.75	**0.04**	**0.04**	**0.02**	**0.02**
27	1.00	0.05	**0.04**	0.05	**0.02**
32	0.25	0.06	**0.04**	0.06	**0.01**
32	0.50	0.06	**0.04**	0.06	**0.02**
32	0.75	0.06	**0.05**	0.07	**0.03**
32	1.00	0.07	**0.06**	0.07	**0.03**

The use of the VST is also a limitation of the proposed algorithm. Because the VST derivation (see Section 4.1) is based on the first order Taylor expansion of $g(y; \mathbf{w})$ around the mean of y, a VST is not accurate when y is far from its mean, i.e. when the noise variance is large. Additionally, when the noise variance is large, the values of many pixels are outside of the VST domain so that they cannot be processed. As a result, the accuracy of the method decreases when the noise level becomes high. It is seen in Tables 4.3, 4.5, and 4.6: the average relative error $\overline{\delta}$ remains approximately the same as the noise level increases, which means an increase of the absolute error. However, this should not be considered as a very large drawback, because the quality of imaging systems increases and processing of images with a low noise level becomes more and more important.

4.6 Summary

A new signal-dependent noise parameter estimation algorithm is proposed. It is essentially different from the state of the art scatterplot method and allows processing images without homogeneous areas. It can also be simply adapted to various noise models. The experiments show that it is significantly more accurate than the state of the art for all considered noise models.

Chapter 5

Application to image denoising

One of the most important applications of noise parameter estimation is image denoising, which often precedes all other image analysis steps. Most of image denoising algorithms utilize the noise parameters in order to adjust the strength of smoothing; and one can expect that improvement of noise parameter estimation leads to increase of the image denoising quality. In this chapter, an experimental evidence of this fact is given by analyzing the performance of a state of the art denoising method with different noise parameter estimators.

The chapter starts with a brief explanation of the state of the art techniques for signal-independent and signal-dependent noise removal. The experimental results are presented in Section 5.2; and the discussion given in Section 5.3 concludes the chapter.

5.1 Image denoising methods

Recent image denoising methods utilize image self-similarity by applying search of similar blocks, which allows efficient processing of highly textured images. For this reason, these algorithms can benefit from improvement of the noise parameter estimation accuracy on such images.

5.1.1 Signal-independent noise removal

In the denoising experiments presented in this chapter, the denoising method [18] designed for AWGN is utilized. It outperforms the methods [60, 45, 26, 31] and can be considered as the state of the art in image denoising. This algorithm is based on block matching and collaborative filtering. For each image block, similar blocks are found and grouped into a 3D stack. Then, the collaborative filtering is applied, which consists of 3D transformation of this stack, shrinkage of the transform spectrum, and inverse 3D transformation. Because the

blocks are similar, fine details shared by them are preserved during the filtering, whereas the noise is suppressed. The filtered blocks are returned to their original positions in the image. Since they are overlapping, one obtains several estimates for each pixel. These estimates are then combined in order to produce the denoised image.

5.1.2 Signal-dependent noise removal

A denoising method developed for AWGN can be also applied for denoising images corrupted with signal-dependent noise by utilizing the VST [29]:

1. VST g is applied to input image y. It is assumed that transformed image $g(y;\mathbf{w})$ has AWGN.

2. Image $g(y;\mathbf{w})$ is processed by a denoising method designed for AWGN. This operation can be seen as computation of an estimate of $\mathbf{E}\big(g(y;\mathbf{w})\big)$. Let us denote this estimate by $D\big(g(y;\mathbf{w})\big)$.

3. Inversion of the VST is applied in order to get an estimate of $\mathbf{E}(y)$. Note that the usage of g^{-1} leads to a biased estimate of $\mathbf{E}(y)$, because, in general, the expectation operator and g do not commute:

$$g\big(\mathbf{E}(y);\mathbf{w}\big) \neq \mathbf{E}\big(g(y;\mathbf{w})\big) \tag{5.1}$$

and

$$\mathbf{E}(y) = g^{-1}\big(g\big(\mathbf{E}(y);\mathbf{w}\big);\mathbf{w}\big) \neq g^{-1}\big(\mathbf{E}\big(g(y;\mathbf{w})\big);\mathbf{w}\big). \tag{5.2}$$

The correct way to inverse the VST is to consider $\mathbf{E}\big(g(y;\mathbf{w})\big)$ as a function of $\mathbf{E}(y)$ and to solve equation

$$\mathbf{E}\big(g(y;\mathbf{w})\big) = D\big(g(y;\mathbf{w})\big) \tag{5.3}$$

for $\mathbf{E}(y)$. The dependence of $\mathbf{E}\big(g(y;\mathbf{w})\big)$ on $\mathbf{E}(y)$ can be seen from the following expression:

$$\mathbf{E}\big(g(y;\mathbf{w})\big) = \int g(t;\mathbf{w}) f_y(t;\mathbf{w}) dt \tag{5.4}$$

where $f_y(t;\mathbf{w})$ is the probability density function of y and the integration is done over the domain of g. From (1.36), $y \sim \mathcal{N}\big(\mathbf{E}(y), h^2(\mathbf{E}(y);\mathbf{w})\big)$ and

$$f_y(t;\mathbf{w}) = \frac{1}{h(\mathbf{E}(y);\mathbf{w})\sqrt{2\pi}} \exp\Big(\frac{-\big(t-\mathbf{E}(y)\big)^2}{2h^2(\mathbf{E}(y);\mathbf{w})}\Big). \tag{5.5}$$

Because of this, $f_y(t;\mathbf{w})$ and, therefore, $\mathbf{E}\big(g(y;\mathbf{w})\big)$ depend on $\mathbf{E}(y)$. Equation (5.3) usually cannot be solved analytically. In the implementation utilized for the experiments, it is solved using a look-up table of $\mathbf{E}\big(g(y;\mathbf{w})\big)$ vs. $\mathbf{E}(y)$.

4. Estimation of x using the estimate of $\mathbf{E}(y)$. Map $\mathbf{E}(y) \mapsto x$ depends only on the noise model. For SAR (1.10), CCD/CMOS (1.29), and ultrasound/film-grain (1.31) noise models, $x = \mathbf{E}(y)$. For MRI noise model (1.12), x can be computed for sufficiently large $\mathbf{E}(y)$ using (1.18): from

$$\mathbf{E}(y) = x + \frac{a^2}{2x} + O(x^{-3}) \quad x \to +\infty \tag{5.6}$$

one can obtain that

$$x \approx \frac{1}{2}\Big(\mathbf{E}(y) + \sqrt{\big(\mathbf{E}(y)\big)^2 - 2a^2}\Big). \tag{5.7}$$

For small $\mathbf{E}(y)$, a look-up table of $\mathbf{E}(y)$ vs. x can be constructed using the exact expression for the mean (1.15).

In the denoising experiments presented in this chapter, the denoising algorithm [18] is utilized at the step 2.

5.2 Experiments

Like in the noise parameter estimation experiments, the TID2008 database has been utilized in the denoising experiments for AWGN, SAR, CCD/CMOS, and ultrasound/film-grain noise models; and the phantom described in Section 4.4.1 has been used for MRI noise model.

Let $PSNR_T$ be the PSNR of the denoised image produced by the denoising algorithm with the true noise parameters; and $PSNR_E$ be the PSNR of the denoised image produced by the denoising algorithm with the estimated noise parameters. The difference

$$PSNR_D = PSNR_E - PSNR_T \tag{5.8}$$

reflects the decrease of the denoising quality, which occurs when the true noise parameters are substituted with the estimated ones. Therefore, in order to evaluate a noise parameter estimation method on TID2008, the average $PSNR_D$ and the minimal $PSNR_D$ over all noisy images have been computed.

5.2.1 Additive white Gaussian noise removal

The results for AWGN are presented in Table 5.1. The minimal $PSNR_D$ for PCA-BS is always larger than -0.1 dB. Similar results are demonstrated with the methods [21], [87], for which the minimal $PSNR_D$ is always larger than -0.3 dB.

The minimal $PSNR_D$ for PCA-RS is in the range -0.5 dB $-$ -0.1 dB. The method [65] gives similar results: its minimal $PSNR_D$ varies in the interval -0.5 dB $-$ -0.4 dB.

For the other approaches, the minimal $PSNR_D$ is always smaller than -0.7 dB, which indicates a significant loss of the denoising quality for some images in the database.

Since the denoising algorithm [18] achieves the best PSNR at some noise level, which is close to the true value but not exactly equal to it, $PSNR_E$ can sometimes be slightly higher than $PSNR_T$. For example, this is the case for the method [87], which has positive average $PSNR_D$.

Table 5.1: The denoising quality with the considered noise parameter estimation methods for AWGN. The methods which have failures are marked with "F". The best result for each noise level is selected with the bold font.

Method	Average $PSNR_D$ (dB)	Minimal $PSNR_D$ (dB)
$\sigma^2 = 25$ ($\sigma = 5$)		
PCA-BS	**-0.01**	**-0.09**
PCA-RS	-0.02	-0.18
[87]	–	–
[95]	-0.10	-1.87
[9]	-0.19	-2.87
[3]	-0.32	-6.34
[89]	F	F
[76]	-2.87	-10.02
[21]	**-0.01**	-0.11
[65]	–	–
[78]	-0.25	-2.72
[24]	-0.45	-4.33
[96]	F	F
[7]	-1.89	-6.58
[83]	-1.24	-6.44
$\sigma^2 = 65$ ($\sigma \approx 8.062$)		
PCA-BS	0.00	**-0.07**
PCA-RS	-0.01	-0.17
[87]	**0.01**	-0.08
[95]	-0.07	-1.11
[9]	-0.09	-1.45
[3]	-0.31	-3.36
[89]	F	F
[76]	-1.57	-6.46
[21]	–	–
[65]	-0.01	-0.49
[78]	-0.24	-2.63
[24]	-0.25	-2.43
[96]	-0.44	-7.61
[7]	-1.21	-4.50
[83]	-0.91	-5.04
$\sigma^2 = 100$ ($\sigma = 10$)		

Continued on the next page

Table 5.1 – *Continued from the previous page*

Method	Average $PSNR_D$ (dB)	Minimal $PSNR_D$ (dB)
PCA-BS	**0.00**	**-0.05**
PCA-RS	-0.02	-0.41
[87]	–	–
[95]	-0.05	-0.81
[9]	-0.06	-1.01
[3]	-0.17	-2.52
[89]	F	F
[76]	-1.12	-6.53
[21]	-0.01	-0.26
[65]	–	–
[78]	-0.09	-1.05
[24]	-0.20	-1.79
[96]	-0.36	-6.83
[7]	-0.77	-3.83
[83]	-0.72	-3.05
$\sigma^2 = 130$ ($\sigma \approx 11.402$)		
PCA-BS	0.00	**-0.06**
PCA-RS	-0.02	-0.29
[87]	**0.01**	-0.07
[95]	-0.05	-0.72
[9]	-0.05	-0.88
[3]	-0.23	-2.66
[89]	F	F
[76]	-0.92	-5.77
[21]	–	–
[65]	-0.02	-0.41
[78]	-0.16	-2.63
[24]	-0.15	-1.42
[96]	-0.28	-5.25
[7]	-0.79	-3.41
[83]	-0.68	-3.54

5.2.2 SAR noise removal

The results for SAR noise are presented in Table 5.2. On average, the denoising quality for the proposed algorithm is approximately the same as that for the true noise parameters, whereas the method [51] results in significantly lower PSNR for the low noise levels.

The minimal $PSNR_D$ for the proposed algorithm is considerably larger than that for the method [51]; and it shows that the proposed approach is applicable to all images in the database.

Table 5.2: The denoising quality with the considered noise parameter estimation methods for SAR noise. The better result is selected with the bold font.

True a	Average $PSNR_D$ (dB)		Minimal $PSNR_D$ (dB)	
	[51]	proposed	[51]	proposed
0.05	-1.37	**-0.03**	-10.66	**-0.53**
0.10	-0.46	**-0.03**	-5.59	**-0.37**
0.15	-0.26	**-0.04**	-3.24	**-0.33**
0.20	-0.14	**-0.04**	-2.18	**-0.23**

5.2.3 MRI noise removal

For the denoising experiments with the MRI phantom, the denoising algorithm [58] has been utilized. It is based on the method [18] and is designed to process 3D volumes corrupted with Gaussian or Rician noise.

Due to the large execution time of [58], the following evaluation procedure has been applied. For each true noise level and for each estimator, noise level estimates have been computed for 100 noise realizations and the minimal and maximal estimates have been selected. Then, the denoising algorithm [58] has been run with these two estimates resulting in two PSNR values: $PSNR_{E,min}$ and $PSNR_{E,max}$. Finally, the minimal $PSNR_D$ has been computed as

$$\min\{PSNR_{E,min} - PSNR_T, PSNR_{E,max} - PSNR_T\}. \tag{5.9}$$

The results presented in Table 5.3 show that all noise level estimators except for [81] lead to approximately the same denoising quality.

5.2.4 CCD/CMOS noise removal

The results for CCD/CMOS noise are presented in Table 5.4. On average, the denoising quality for the proposed algorithm is approximately the same as that for the true noise parameters. For the method [32], the average denoising quality decreases significantly for the low noise levels.

The minimal values of $PSNR_D$ show that the both methods lead to considerably lower PSNR for some images in the database compared with the PSNR with the true noise parameters. However, the minimal $PSNR_D$ for the proposed method is much larger than that for the method [32].

Table 5.3: The denoising quality with the considered noise parameter estimation methods for MRI noise. The best result is selected with the bold font.

True noise level, %	Minimal $PSNR_D$ (dB)			
	[81]	[16]	[30]	proposed
1	0.07	0.08	**0.14**	-0.05
2	**0.03**	0.01	**0.03**	-0.03
3	**0.01**	0.00	0.00	-0.02
4	-0.01	**0.00**	**0.00**	-0.01
5	-0.05	**0.00**	**0.00**	**0.00**
6	-0.10	-0.01	**0.00**	-0.01
7	-0.19	-0.01	**0.00**	-0.01
8	-0.31	-0.01	**0.01**	-0.01
9	-0.49	-0.02	**0.01**	-0.01
10	-0.68	-0.02	**0.01**	-0.01
11	-0.89	-0.02	**0.02**	-0.01
12	-1.22	-0.02	**0.02**	-0.01
13	-1.50	-0.03	**0.02**	-0.01
14	-1.88	-0.03	**0.03**	-0.01
15	-2.44	-0.04	**0.03**	0.00

Table 5.4: The denoising quality with the considered noise parameter estimation methods for CCD/CMOS noise. The better result is selected with the bold font.

True a	True b	Average $PSNR_D$ (dB)		Minimal $PSNR_D$ (dB)	
		[32]	proposed	[32]	proposed
0.5	16	-0.73	**-0.06**	-5.82	**-2.13**
0.5	64	-0.48	**-0.07**	-3.72	**-2.60**
1.0	16	-0.49	**-0.02**	-4.62	**-0.15**
1.0	64	-0.38	**-0.05**	-3.24	**-1.34**
2.0	16	-0.28	**-0.02**	-2.06	**-0.13**
2.0	64	-0.27	**-0.02**	-3.50	**-0.25**

5.2.5 Ulrasound/film-grain noise removal

The results for ultrasound/film-grain noise are presented in Table 5.5. On average, the both methods result in approximately the same denoising quality as that with the true noise parameters. However, the minimal $PSNR_D$ for the method [86] is always smaller than -0.9 dB, which is a significant decrease of the PSNR. For the proposed algorithm, the minimal $PSNR_D$ is always larger than -0.4 dB.

Table 5.5: The denoising quality with the considered noise parameter estimation methods for ultrasound/film-grain noise. The better result is selected with the bold font.

True $PSNR$ (dB)	True b	Average $PSNR_D$ (dB)		Minimal $PSNR_D$ (dB)	
		[86]	proposed	[86]	proposed
27	0.25	-0.08	**-0.04**	-2.97	**-0.29**
27	0.50	-0.07	**-0.04**	-0.97	**-0.23**
27	0.75	-0.05	**-0.04**	-0.91	**-0.39**
27	1.00	-0.06	**-0.02**	-0.90	**-0.21**
32	0.25	-0.09	**-0.05**	-1.79	**-0.22**
32	0.50	-0.08	**-0.04**	-1.40	**-0.19**
32	0.75	-0.07	**-0.04**	-1.51	**-0.25**
32	1.00	-0.07	**-0.03**	-1.16	**-0.19**

5.3 Discussion

In general, the denoising results show that a higher noise level estimation accuracy leads to a higher denoising quality in most cases and that the increase of noise parameter estimation accuracy provided by the proposed algorithms indeed results in increase of the PSNR of denoised images. This can be explained from several points of view:

1. Compared with the state of the art noise parameter estimation methods, the advantages of the proposed methods are most clearly seen on images containing only textures. As explained in Section 5.1, the denoising method [18] utilizes image self-similarity and sparse image representation in the transform domain so that it can efficiently process textures. Therefore, it is able to benefit from the more accurate noise parameter estimates computed by the proposed algorithms when processing highly textured images.

2. In the denoising algorithm [18], the shrinkage of the transform spectrum is done by thresholding of the transform coefficients, where the threshold is proportional to the noise standard deviation. Hence when the noise level is overestimated, the threshold is higher than its optimal value, and some transform coefficients corresponding to the signal are set to zero, which results in an oversmoothed denoising result. On the other hand, when the noise level is underestimated, the threshold is too low, and transform coefficients corresponding only to the noise are kept, so that some noise presents in the output image. As a result, it is important to provide noise level estimate, which is close to the true noise level.

3. In the case of signal-dependent noise, accurate estimation of the noise parameters is necessary to construct the VST correctly, because the VST de-

pends on the noise parameters. An incorrect VST can be a reason for significant deviations of the noise in transformed image $g(y; \mathbf{w})$ from AWGN, which can make the denoising quality worse. Additionally, the VST domain depends on the noise parameters for CCD/CMOS noise model, therefore, a large noise parameter estimation error can lead to the condition when many input image pixels are outside of the VST domain so that they cannot be efficiently processed by the denoising algorithm.

Chapter 6

Conclusion

In this thesis, a new framework for noise parameter estimation is proposed. It is based on a sparse representation of image blocks, which is analyzed using PCA. The framework consists of the following parts:

1. Checks of the sparsity assumption (Assumtion 1). These checks are presented in Sections 3.4.1 and 3.4.2.

2. Noise variance estimators (see Section 3.4.3).

3. Image block selection strategies, which are described in Sections 3.5 and 3.6.

4. Noise normality check presented in Section 4.2.1.

5. VST parameter selection algorithm (see Sections 4.2.2 and 4.2.3).

These parts can be combined in different ways, leading to different trade-offs between accuracy and computation time, as well as to algorithms designed for different noise models.

The main property, which is preserved in all implementations, is the assumption about the original image (Assumtion 1). It is assumed that a part of the original image blocks lies in a subspace of the image block space, which means a linear dependence of pixels in the blocks. PCA provides a way to check this assumption and makes the proposed algorithms computationally efficient. Indeed, for signal-independent noise, the algorithm PCA-BS is more than 15 times faster than the other methods with similar accuracy (see Section 3.7.2). In contrast to the state of the art methods of signal-dependent noise parameter estimation, Assumption 1 does not require the existence of homogeneous areas in the original image, so that highly textured images can be accurately processed. This can be seen in the experimental results given in Section 4.4.

As shown in the experiments presented in Chapter 5, utilizing the proposed noise parameter estimators in denoising applications results in a significant increase of the denoising quality. Besides, image compression and segmentation

applications, which take the noise level as an input value, can also take advantage of the proposed framework.

The presented work also shows limitations of the proposed methods and possible directions of the further work. In Section 3.7.3, it is demonstrated that stochastic textures, whose correlation properties are close to those of white noise, cannot be accurately processed. For this reason, the further research could be related to the search of new sparsifying transforms, which do not require a linear dependence between pixels in image blocks and can handle more complex relationships between pixels in image texture.

In the case of signal-dependent noise, the use of the VST is also a limitation, because its accuracy decreases as the noise level increases, as pointed out in Section 4.5. Hence other variance-stabilizing techniques, which can accurately handle noise with high variance, could be a topic of the further work as well.

Appendix A

Proof of Theorem 1

In this appendix, the proof of Theorem 1 is given. This theorem is the basis of the assumption check procedure used in the proposed algorithms. Therefore, although the simulation results are given, the formal proof is important to ensure that the algorithms work correctly for any image block set, which satisfies Assumption 1.

The following sections are organized as follows. First, the previous results for the distribution of the sample covariance matrix eigenvalues are reviewed. Then, the relevant results of the eigenvalue perturbation theory, which are necessary for the proof, are presented. Next, information on the variance of the sample covariance is given. Finally, the result of Theorem 1 is derived.

A.1 Previous results for the sample eigenvalue distribution

The properties of the distributions of the sample covariance matrix eigenvalues were being studied over the last six decades.

First, the case, when the parent distribution is multivariate normal, was considered. Expansions of the sample eigenvalues about the distinct population eigenvalues were derived in [50]. In [5], the sample eigenvalue distribution is given for the case when the population eigenvalues have arbitrary multiplicities.

The case of a non-normal population is more complex. For distinct population eigenvalues, the distribution of the sample eigenvalues was analyzed in [91], and the distributions of functions of the sample eigenvalues were considered in [34]. In [25], the authors studied convergence of the sample eigenvalues in distribution for the case when the population eigenvalues have arbitrary multiplicity.

However, the results [50, 5, 91, 34] are not applicable to the proof of Theorem 1, because random vector $\mathbf{X}$ can have arbitrary distribution, not necessarily multivariate normal, and the eigenvalues of matrices $\Sigma_{\mathbf{X}}$ and $\Sigma_{\mathbf{Y}}$ are not distinct under Assumption 1.

Compared with the work [25], a stronger type of convergence is considered in Theorem 1, namely, convergence in mean. This allows, in particular, proving that the average of the last several sample eigenvalues is a consistent noise variance estimator (see Section 3.4.3).

A.2 Eigenvalue perturbation theory

Given population covariance matrix $\mathbf{\Sigma_Y}$, sample covariance matrix $S_\mathbf{Y}$ can be seen as the sum of $\mathbf{\Sigma_Y}$ and small perturbation $S_\mathbf{Y} - \mathbf{\Sigma_Y}$. Therefore, the proof of Theorem 1, like the results given in [50, 91, 34, 25], is based on eigenvalue perturbation theory.

The perturbation of the eigenvalues can be bounded by $\|S_\mathbf{Y} - \mathbf{\Sigma_Y}\|_2$:

Lemma 1. *Let $A \in \mathbb{C}^{M\times M}$ and $B \in \mathbb{C}^{M\times M}$ be Hermitian matrices, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$ be the eigenvalues of A, and $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_M$ be the eigenvalues of perturbed matrix $\tilde{A} = A + B$. Then $\forall i = 1, \ldots, M$ $|\tilde{\lambda}_i - \lambda_i| \leq \|B\|_2$.*

Proof. See [84, p. 203]. □

However, this is not a tight bound when Assumption 1 holds, because the eigenvalues corresponding only to noise are considered in Theorem 1, whereas the difference $(S_\mathbf{Y} - \mathbf{\Sigma_Y})$ is affected by the original image and can be arbitrary large.

Theorem 2.3 in [84, p. 183] gives an estimate with accuracy $\|S_\mathbf{Y} - \mathbf{\Sigma_Y}\|_2^2$, but it can be applied only for eigenvalues with multiplicity 1, which is not the case when Assumption 1 holds. In [92, p. 76], this result was extended to the case of arbitrary eigenvalue multiplicity. However, the formulation in [92] has some restrictions and cannot be applied here directly so that a different formulation (Lemma 2 below) is used, although its proof has the same steps as those in [84] and [92]. Particularly, it is also based on Gerschgorin's result:

Theorem 2 (Gerschgorin)**.** *For matrix $A \in \mathbb{C}^{M\times M}$ with entries a_{ij}, let*

$$a_i = \sum_{j\neq i} |a_{ij}| \tag{A.1}$$

and

$$G_i(A) = \{z \in \mathbb{C} |\ |z - a_{ii}| \leq a_i\}. \tag{A.2}$$

Then all eigenvalues of A lie in $\cup_{i=1}^{M} G_i(A)$. Moreover, if q of Gerschgorin disks $G_i(A)$ are isolated from the other $M - q$ disks, then there are precisely q eigenvalues of A in their union.

Proof. See [84, p. 181]. □

Lemma 2. *Let $A = [a_{ij}] \in \mathbb{C}^{M\times M}$ be a diagonal matrix such that*

$$\begin{aligned} a_{ii} &\neq \lambda \quad i = 1, \ldots, M - q \\ a_{ii} &= \lambda \quad i = M - q + 1, \ldots, M \end{aligned} \tag{A.3}$$

i.e. λ is an eigenvalue of A with multiplicity q. Let $B = [b_{ij}] \in \mathbb{C}^{M\times M}$ be a Hermitian matrix, $\delta > 0$ be the minimum of the distances between λ and the other eigenvalues of A, and $\tilde{A} = A+B$ be a perturbed matrix. If $\|B\|_2 < \frac{\delta}{4M}$ then there are exactly q eigenvalues $\tilde{\lambda}_k$ of $\tilde{A}$, which satisfy the following inequality:

$$|\tilde{\lambda}_k - \lambda| \leq \max_{i=M-q+1,\dots,M} \sum_{j=M-q+1}^{M} |b_{ij}| + \frac{4M^2}{\delta}\|B\|_2^2. \tag{A.4}$$

Proof. Without loss of generality,

$$A = \mathrm{diag}\{\underbrace{\lambda', \dots, \lambda'}_{M-q}, \underbrace{\lambda, \dots, \lambda}_{q}\} \tag{A.5}$$

where $|\lambda' - \lambda| \geq \delta$. Let us consider matrix $\tilde{A}_a = D_a \tilde{A} D_a^{-1}$, where

$$D_a = \mathrm{diag}\{\underbrace{1, \dots, 1}_{M-q}, \underbrace{a, \dots, a}_{q}\} \quad a > 0. \tag{A.6}$$

Matrices $\tilde{A}$ and $\tilde{A}_a$ are similar and have the same eigenvalues.

$$\tilde{A}_a = D_a(A+B)D_a^{-1} = D_aAD_a^{-1} + D_aBD_a^{-1} = A+$$
$$\begin{bmatrix} b_{11} & \cdots & b_{1\,M-q} & a^{-1}b_{1\,M-q+1} & \cdots & a^{-1}b_{1M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ b_{M-q\,1} & \cdots & b_{M-q\,M-q} & a^{-1}b_{M-q\,M-q+1} & \cdots & a^{-1}b_{M-q\,M} \\ ab_{M-q+1\,1} & \cdots & ab_{M-q+1\,M-q} & b_{M-q+1\,M-q+1} & \cdots & b_{M-q+1\,M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ ab_{M1} & \cdots & ab_{M\,M-q} & b_{M\,M-q+1} & \cdots & b_{MM} \end{bmatrix}. \tag{A.7}$$

Note that $|b_{ij}| \leq \|B\|_2$.

The first $M-q$ Gerschgorin disks of $\tilde{A}_a$ have centers $\lambda' + b_{ii}$ and radii bounded by

$$\sum_{j=1}^{M-q} |b_{ij}| - |b_{ii}| + qa^{-1}\|B\|_2 \leq (M-1)\|B\|_2 + Ma^{-1}\|B\|_2. \tag{A.8}$$

Hence these disks lie entirely in circle

$$C_1 = \{z \in \mathbb{C} \mid |z - \lambda'| \leq (1 + a^{-1})M\|B\|_2\}. \tag{A.9}$$

The last q Gerschgorin disks have centers $\lambda + b_{ii}$ and radii bounded by

$$\sum_{j=M-q+1}^{M} |b_{ij}| - |b_{ii}| + (M-q)a\|B\|_2 \leq \sum_{j=M-q+1}^{M} |b_{ij}| - |b_{ii}| + Ma\|B\|_2 \tag{A.10}$$

so that they lie entirely in circle

$$C_2 = \{z \in \mathbb{C} \mid |z - \lambda| \leq t + Ma\|B\|_2\} \tag{A.11}$$

where

$$t = \max_{j=M-q+1,\ldots,M} \sum_{j=M-q+1}^{M} |b_{ij}|. \tag{A.12}$$

Since $|\lambda - \lambda'| \geq \delta$, if

$$(1 + a^{-1})M\|B\|_2 + t + Ma\|B\|_2 < \delta \tag{A.13}$$

circles C_1 and C_2 are disjoint. According to the statement of the lemma, $M\|B\|_2 < \delta/4$. Besides,

$$t \leq q\|B\|_2 \leq M\|B\|_2 < \frac{\delta}{4}. \tag{A.14}$$

As a result, (A.13) is satisfied for

$$a = \frac{4M\|B\|_2}{\delta}. \tag{A.15}$$

Then circle C_2:

1. has the radius bounded by

$$t + \frac{4M^2}{\delta}\|B\|_2^2 \tag{A.16}$$

2. is disjoint from the first $M - q$ Gerschgorin disks of $\tilde{A}_a$,

3. contains the last q Gerschgorin disks of $\tilde{A}_a$.

From Gerschgorin's theorem, C_2 contains exactly q eigenvalues of $\tilde{A}_a$ and, therefore, exactly q eigenvalues of $\tilde{A}$. □

A.3 The variance of the sample covariance

Since elements of matrix $S_{\mathbf{Y}}$ are the sample covariances of elements of $\mathbf{Y}$, the variance of the sample covariance is of interest. It can be taken from the more general results obtained in [14].

Let ξ and η be two random variables. Consider the cumulant generating function

$$\ln \mathbf{E}\big(e^{t_1\xi + t_2\eta}\big) = \frac{\kappa_{10}}{1!0!}t_1 + \frac{\kappa_{01}}{0!1!}t_2 + \ldots + \frac{\kappa_{ij}}{i!j!}t_1^i t_2^j + \ldots. \tag{A.17}$$

Coefficients κ_{ij} are the bivariate cumulants, i.e. the cumulants of the joint distribution of ξ and η. Particularly, κ_{i0} is the ith univariate cumulant κ_i of ξ.

Let $\mu_{ij} = \mathbf{E}\big((\xi - \mathbf{E}(\xi))^i(\eta - \mathbf{E}(\eta))^j\big)$ be the (i,j)th central moment. For example, μ_{i0} is the ith univariate moment μ_i of ξ and μ_{11} is the covariance between ξ and η. The first several bivariate cumulants are related to the central moments as follows [14]:

$$\begin{aligned}
\mu_{20} &= \kappa_{20} \\
\mu_{11} &= \kappa_{11} \\
\mu_{30} &= \kappa_{30} \\
\mu_{21} &= \kappa_{21} \\
\mu_{40} &= \kappa_{40} + 3\kappa_{20}^2 \\
\mu_{31} &= \kappa_{31} + 3\kappa_{20}\kappa_{11} \\
\mu_{22} &= \kappa_{22} + \kappa_{20}\kappa_{02} + 2\kappa_{11}^2.
\end{aligned} \tag{A.18}$$

The sample estimates of κ_{ij} are denoted by k_{ij}. Specifically, $c(\xi,\eta) = k_{11}$ is the sample covariance:

$$k_{11} = c(\xi,\eta) = \frac{1}{N-1}\sum_{i=1}^{N}(\xi_i - \overline{\xi})(\eta_i - \overline{\eta}) \tag{A.19}$$

where $\{\xi_i\}$ and $\{\eta_i\}$ are samples of size N and

$$\overline{\xi} = \frac{1}{N}\sum_{i=1}^{N}\xi_i \qquad \overline{\eta} = \frac{1}{N}\sum_{i=1}^{N}\eta_i. \tag{A.20}$$

As in [14], let us denote the (r,s)th cumulant of the joint distribution of $k_{\alpha\alpha'}$ and $k_{\beta\beta'}$ by

$$\kappa\Big(\underbrace{\begin{matrix}\alpha & \cdots & \alpha \\ \alpha' & \cdots & \alpha'\end{matrix}}_{r}\ \underbrace{\begin{matrix}\beta & \cdots & \beta \\ \beta' & \cdots & \beta'\end{matrix}}_{s}\Big). \tag{A.21}$$

Then the rth cumulant of sample covariance k_{11} is

$$\kappa\Big(\underbrace{\begin{matrix}1 & \cdots & 1 \\ 1 & \cdots & 1\end{matrix}}_{r}\Big). \tag{A.22}$$

and the variance of the sample covariance can be taken from the list given in [14]:

$$\mu_2(c(\xi,\eta)) = \kappa\begin{pmatrix}1 & 1 \\ 1 & 1\end{pmatrix} = \frac{\kappa_{22}}{N} + \frac{\kappa_{20}\kappa_{02} + \kappa_{11}^2}{N-1}. \tag{A.23}$$

A.4 Proof of Theorem 1

Further, let x_i, n_i, and y_i be entries of $\mathbf{X}$, $\mathbf{N}$, and $\mathbf{Y}$ respectively,

$$B = [b_{ij}] = S_{\mathbf{Y}} - \mathbf{\Sigma_Y}$$
$$d = \frac{\delta}{4M}. \tag{A.24}$$

Let $\Lambda_{\mathbf{X}} = Q^{\mathrm{T}}\Sigma_{\mathbf{X}}Q$ be the eigendecomposition of $\Sigma_{\mathbf{X}}$. Q is an orthogonal matrix and $\Lambda_{\mathbf{X}}$ is the diagonal matrix whose diagonal elements are the eigenvalues of $\Sigma_{\mathbf{X}}$. Then the eigenvalues of $\Sigma_{\mathbf{Y}}$ and $Q^{\mathrm{T}}\Sigma_{\mathbf{Y}}Q$ are the same, and the eigenvalues of $S_{\mathbf{Y}}$ and $Q^{\mathrm{T}}S_{\mathbf{Y}}Q$ are the same. For this reason, we can assume without loss of generality that $\Sigma_{\mathbf{X}}$ is a diagonal matrix:

$$\Sigma_{\mathbf{X}} = \begin{bmatrix} \delta_{\mathbf{X},1}I_{q_1} & 0 & \cdots & 0 \\ 0 & \delta_{\mathbf{X},2}I_{q_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_{\mathbf{X},p}I_{q_p} \end{bmatrix}. \tag{A.25}$$

In this case, matrix B can be represented in the following form:

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^{\mathrm{T}} & B_{22} \end{bmatrix} \tag{A.26}$$

where B_{11} is a $(M - q_p + 1) \times (M - q_p + 1)$ symmetric matrix, B_{12} is a $(M - q_p + 1) \times q_p$ matrix, and B_{22} is a $q_p \times q_p$ symmetric matrix.

Lemma 3. *In the conditions of Theorem 1*

$$\mathbf{E}(\|B\|_2^2) = O(1/N). \tag{A.27}$$

Proof. Indeed, from (A.23),

$$\mathrm{var}(b_{ij}) = \mathrm{var}(c(y_i, y_j)) = \kappa \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = O(1/N). \tag{A.28}$$

The spectral norm of B is bounded by its Frobenius norm:

$$\|B\|_2 \leq \sqrt{\sum_{i,j=1}^{M} b_{ij}^2}. \tag{A.29}$$

Additionally, $\mathbf{E}(b_{ij}) = 0$ because $\mathbf{E}(S_{\mathbf{Y}}) = \mathbf{\Sigma_Y}$. Hence, because of monotonicity of the expected value,

$$\begin{aligned} \mathbf{E}(\|B\|_2^2) \leq \mathbf{E}\Big(\sum_{i,j=1}^{M} b_{ij}^2\Big) &= \sum_{i,j=1}^{M} \mathbf{E}(b_{ij}^2) = \sum_{i,j=1}^{M} \Big(\mathrm{var}(b_{ij}) + \big(\mathbf{E}(b_{ij})\big)^2\Big) \\ &= \sum_{i,j=1}^{M} \mathrm{var}(b_{ij}) = O(1/N). \end{aligned} \tag{A.30}$$

□

Lemma 4. *In the conditions of Theorem 1*

$$\mathbf{E}(\|B\|_2 \mathbf{1}_{\|B\|_2 \geq d}) = O(1/N) \tag{A.31}$$

Proof. Let $F(t)$ be the CDF of $\|B\|_2$. From Chebyshev's inequality,

$$1 - F(t) = P(\|B\|_2 > t) \leq \frac{\mathbf{E}(\|B\|_2^2)}{t^2} \tag{A.32}$$

Using integration by parts,

$$\begin{aligned}\mathbf{E}(\|B\|_2 \mathbf{1}_{\|B\|_2 \geq d}) &= \int_d^{+\infty} t dF(t) = -\int_d^{+\infty} t d(1 - F(t)) \\ &= -t(1 - F(t))\big|_d^{+\infty} + \int_d^{+\infty} (1 - F(t)) dt. \end{aligned} \tag{A.33}$$

From (A.32),

$$0 \leq t(1 - F(t)) \leq \frac{\mathbf{E}(\|B\|_2^2)}{t} \quad t > 0 \tag{A.34}$$

and

$$\lim_{t \to +\infty} t(1 - F(t)) = 0. \tag{A.35}$$

As a result (see also [11]),

$$\begin{aligned}\mathbf{E}(\|B\|_2 \mathbf{1}_{\|B\|_2 \geq d}) &= d\big(1 - F(d)\big) + \int_d^{+\infty} (1 - F(t)) dt \\ &\leq \frac{\mathbf{E}(\|B\|_2^2)}{d} + \int_d^{+\infty} \frac{\mathbf{E}(\|B\|_2^2)}{t^2} dt = \frac{2\mathbf{E}(\|B\|_2^2)}{d} \end{aligned} \tag{A.36}$$

Then (A.31) follows from (A.27). □

Proof of Theorem 1. Under Assumption 1, the last $q_p \geq m$ eigenvalues of $\mathbf{\Sigma_X}$ are zeros and the last q_p eigenvalues of $\mathbf{\Sigma_Y}$ equal σ^2. Let $J = \{M - q_p + 1, \ldots, M\}$ be the set of indices of zero eigenvalues of $\mathbf{\Sigma_X}$. Using Lemma 2 with matrices $\mathbf{\Sigma_Y}$ and B, for $k \in J$

$$\|B\|_2 < \frac{\delta}{4M} \;\Rightarrow\; |\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2| \leq \max_{i \in J} \sum_{j \in J} |b_{ij}| + \frac{4M^2}{\delta} \|B\|_2^2. \tag{A.37}$$

Because of the fact that $\max_{i \in J} \sum_{j \in J} |b_{ij}| \leq \sqrt{q_p} \|B_{22}\|_2$, when $\|B\|_2 < \frac{\delta}{4M}$,

$$|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2| \leq \sqrt{q_p} \|B_{22}\|_2 + \frac{4M^2}{\delta} \|B\|_2^2 \tag{A.38}$$

and, using monotonicity of the expected value,

$$\mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2| \mathbf{1}_{\|B\|_2 < d}\big) \leq \sqrt{q_p} \mathbf{E}(\|B_{22}\|_2) + \frac{4M^2}{\delta} \mathbf{E}(\|B\|_2^2). \tag{A.39}$$

Consider the first summand on the right side of (A.39). Using the fact that the spectral norm of B_{22} is bounded by its Frobenius norm and then applying Hölder's inequality, we have

$$\mathbf{E}(\|B_{22}\|_2) \leq \mathbf{E}\Big(\sqrt{\sum_{i,j\in J} b_{ij}^2}\Big) \leq \sqrt{\mathbf{E}\Big(\sum_{i,j\in J} b_{ij}^2\Big)} = \sqrt{\sum_{i,j\in J} \mathbf{E}(b_{ij}^2)}. \tag{A.40}$$

For $i, j \in J$,

$$\begin{aligned}\mathbf{E}(b_{ij}^2) &= \mathrm{var}(b_{ij}) = \mathrm{var}(c(y_i, y_j)) = \mathrm{var}(c(x_i + n_i, x_j + n_j)) \\ &= \mathrm{var}(c(x_i, x_j) + c(x_i, n_j) + c(n_i, x_j) + c(n_i, n_j)) \\ &= \mathrm{var}(c(n_i, n_j))\end{aligned} \tag{A.41}$$

because $x_i = x_j = 0$ almost surely. From [14] and using the fact that $n_i, n_j \sim \mathcal{N}(0, \sigma^2)$,

$$\begin{aligned}\mathrm{var}(c(n_i, n_j)) &= \kappa \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \\ &= \frac{\mathbf{E}(n_i^2 n_j^2)}{N} + \frac{\mathbf{E}(n_i^2)\mathbf{E}(n_j^2)}{N-1} + \frac{(\mathbf{E}(n_i n_j))^2}{N-1} \\ &\leq \frac{\mathbf{E}(n_i^4)}{N} + \frac{(\mathbf{E}(n_i^2))^2}{N-1} + \frac{(\mathbf{E}(n_i^2))^2}{N-1} \\ &= \frac{3\sigma^4}{N} + 2\frac{\sigma^4}{N-1} = O(\sigma^4/N).\end{aligned} \tag{A.42}$$

As a result, $\mathbf{E}(b_{ij}^2) = O(\sigma^4/N)$, and $\mathbf{E}(\|B_{22}\|_2) = O(\sigma^2/\sqrt{N})$.

Applying (A.27) to the second summand on the right side of (A.39), we have

$$\mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|\mathbf{1}_{\|B\|_2<d}\big) = O(\sigma^2/\sqrt{N}) + O(1/N) = O(\sigma^2/\sqrt{N}) \tag{A.43}$$

because $1/N$ is infinitesimal compared with $1/\sqrt{N}$ when $N \to +\infty$.

Utilizing Lemmas 1 and 4,

$$\mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|\mathbf{1}_{\|B\|_2\geq d}\big) \leq \mathbf{E}(\|B\|_2 \mathbf{1}_{\|B\|_2\geq d}) = O(1/N). \tag{A.44}$$

In order to get the final result, we should combine bounds (A.43) and (A.44):

$$\begin{aligned}\mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|\big) &= \mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|\mathbf{1}_{\|B\|_2<d}\big) \\ &\quad + \mathbf{E}\big(|\tilde{\lambda}_{\mathbf{Y},k} - \sigma^2|\mathbf{1}_{\|B\|_2\geq d}\big) \\ &= O(\sigma^2/\sqrt{N}) + O(1/N) = O(\sigma^2/\sqrt{N}).\end{aligned} \tag{A.45}$$

□

Bibliography

[1] S.K. Abramov, V.V. Lukin, B. Vozel, K. Chehdi, and J.T. Astola. Segmentation-based method for blind evaluation of noise variance in images. *Journal of Applied Remote Sensing*, 2:023533, 2008.

[2] B. Aiazzi, L. Alparone, S. Baronti, and A. Garzelli. Coherence estimation from multilook incoherent sar imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(11):2531–2539, 2003.

[3] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, M. Martín-Fernández, and C. Alberola-López. Automatic noise estimation in images using local statistics. additive and multiplicative cases. *Image and Vision Computing*, 27(6):756–770, 2009.

[4] L. Alparone, M. Selva, L. Capobianco, S. Moretti, L. Chiarantini, and F. Butera. Quality assessment of data products from a new generation airborne imaging spectrometer. In *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, volume 4, pages IV–422. IEEE.

[5] T.W. Anderson. Asymptotic theory for principal component analysis. *The Annals of Mathematical Statistics*, 34(1):122–148, 1963.

[6] TW Anderson and D.A. Darling. A test of goodness of fit. *Journal of the American Statistical Association*, 49(268):765–769, 1954.

[7] A. Barducci, D. Guzzi, P. Marcoionni, and I. Pippi. Assessing noise amplitude in remotely sensed images using bit-plane and scatterplot approaches. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(8):2665–2675, 2007.

[8] M.S. Bartlett. Tests of significance in factor analysis. *British Journal of Statistical Psychology*, 3(2):77–85, 1950.

[9] R.C. Bilcu and M. Vehvilainen. New method for noise estimation in images. In *Nonlinear Signal and Image Processing, 2005. NSIP 2005. Abstracts. IEEE-Eurasip*, page 25. IEEE.

[10] A. Bosco, RA Bruna, D. Giacalone, S. Battiato, and R. Rizzo. Signal-dependent raw image denoising using sensor noise characterization via multiple acquisitions. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7537, page 4, 2010.

[11] M.C. Bryson. Heavy-tailed distributions: properties and tests. *Technometrics*, pages 61–68, 1974.

[12] A. Buades, B. Coll, and J.M. Morel. On image denoising methods. *SIAM Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

[13] C.A. Cocosco, V. Kollokian, K.S.K. Remi, G.B. Pike, and A.C. Evans. Brainweb: Online interface to a 3D MRI simulated brain database. In *NeuroImage*. Citeseer, 1997.

[14] MB Cook. Bi-variate k-statistics and cumulants of their joint sampling distribution. *Biometrika*, 38(1/2):179–195, 1951.

[15] BR Corner, RM Narayanan, and SE Reichenbach. Noise estimation in remote sensing imagery using data masking. *International Journal of Remote Sensing*, 24(4):689–702, 2003.

[16] P. Coupé, J.V. Manjķn, E. Gedamu, D. Arnold, M. Robles, and D.L. Collins. Robust rician noise estimation for mr images. *Medical Image Analysis*, 14(4):483–493, 2010.

[17] H. Cramer. On the composition of elementary errors. *Scandinavian Actuarial Journal*, 1928(1):13–74, 1928.

[18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.

[19] R.B. D'agostino, A. Belanger, and R.B. D'Agostino Jr. A suggestion for using powerful and informative tests of normality. *The American Statistician*, 44(4):316–321, 1990.

[20] J.C. Dainty and R. Shaw. *Image science: principles, analysis and evaluation of photographic-type imaging processes*. Academic Press, 1974.

[21] A. Danielyan and A. Foi. Noise variance estimation in nonlocal transform domain. In *Local and Non-Local Approximation in Image Processing, 2009. LNLA 2009. International Workshop on*, pages 41–45. IEEE, 2009.

[22] A. De Stefano, PR White, and WB Collis. Training methods for image noise level estimation on wavelet components. *EURASIP Journal on Applied Signal Processing*, 2004:2400–2407, 2004.

[23] O. Dietrich, J.G. Raya, S.B. Reeder, M. Ingrisch, M.F. Reiser, and S.O. Schoenberg. Influence of multichannel combination, parallel imaging and other reconstruction techniques on MRI noise characteristics. *Magnetic resonance imaging*, 26(6):754–762, 2008.

[24] D.L. Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 1995.

[25] M.L. Eaton and D.E. Tyler. On Wielandt's inequality and its application to the asymptotic distribution of the eigenvalues of a random symmetric matrix. *The Annals of Statistics*, 19(1):260–271, 1991.

[26] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[27] A. Erdélyi, W. Magnus, F. Oberhettinger, F.G. Tricomi, et al. *Higher transcendental functions, Vol. I.* McGraw-Hill, New York, 1953.

[28] A. Foi. Estimated standard deviation functions for Canon PowerShot G10. http://www.cs.tut.fi/~foi/sensornoise.html.

[29] A. Foi. Clipped noisy images: heteroskedastic modeling and practical denoising. *Signal Processing*, 89(12):2609–2629, 2009.

[30] A. Foi. Noise estimation and removal in MR imaging: The variance-stabilization approach. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1809–1814. IEEE, 2011.

[31] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. *Image Processing, IEEE Transactions on*, 16(5):1395–1411, 2007.

[32] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *Image Processing, IEEE Transactions on*, 17(10):1737–1754, 2008.

[33] W. Förstner. Image preprocessing for feature extraction in digital intensity, color and range images. *Geomatic Method for the Analysis of Data in the Earth Sciences*, pages 165–189, 2000.

[34] Y. Fujikoshi. Asymptotic expansions for the distributions of the sample roots under nonnormality. *Biometrika*, 67(1):45–51, 1980.

[35] M. Ghazal and A. Amer. Homogeneity localization using particle filters with application to noise estimation. *Image Processing, IEEE Transactions on*, (99):1–1, 2010.

[36] L.A. Goodman. On the exact variance of products. *Journal of the American Statistical Association*, pages 708–713, 1960.

[37] P. Gravel, G. Beaudoin, and J.A. De Guise. A method for modeling noise in medical images. *Medical Imaging, IEEE Transactions on*, 23(10):1221–1232, 2004.

[38] M. Hashemi and S. Beheshti. Adaptive noise variance estimation in bayesshrink. *Signal Processing Letters, IEEE*, 17(1):12–15, 2010.

[39] M. Hensel, T. Pralow, and R.R. Grigat. Modeling and real-time estimation of signal-dependent noise in quantum-limited imaging. In *Proceedings of WSEAS International Conference on Signal Processing, Robotics and Automatiin*, pages 16–19, 2007.

[40] R.J. Hyndman and Y. Fan. Sample quantiles in statistical packages. *American Statistician*, pages 361–365, 1996.

[41] C.R. Jackson, J.R. Apel, and United States. Dept. of Commerce. *Synthetic aperture radar marine user's manual*. US Department of Commerce, 2004.

[42] C.M. Jarque and A.K. Bera. A test for normality of observations and regression residuals. *International Statistical Review/Revue Internationale de Statistique*, pages 163–172, 1987.

[43] DN Joanes and CA Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998.

[44] I. Jolliffe. Principal component analysis. *Encyclopedia of Statistics in Behavioral Science*, 2002.

[45] C. Kervrann and J. Boulanger. Optimal spatial adaptation for patch-based image denoising. *Image Processing, IEEE Transactions on*, 15(10):2866–2878, 2006.

[46] C.G. Koay and P.J. Basser. Analytically exact correction scheme for signal extraction from noisy magnitude mr signals. *Journal of Magnetic Resonance*, 179(2):317–322, 2006.

[47] PP Koltsov. Comparative study of texture detection and classification algorithms. *Computational Mathematics and Mathematical Physics*, 51(8):1460–1466, 2011.

[48] K. Konstantinides, B. Natarajan, and G.S. Yovanof. Noise estimation and filtering using block-based singular value decomposition. *Image Processing, IEEE Transactions on*, 6(3):479–483, 1997.

[49] R.K.S. Kwan, A.C. Evans, and G.B. Pike. MRI simulation-based evaluation of image-processing and classification methods. *Medical Imaging, IEEE Transactions on*, 18(11):1085–1097, 1999.

[50] DN Lawley. Tests of significance for the latent roots of covariance and correlation matrices. *Biometrika*, 43(1/2):128–136, 1956.

[51] J.S. Lee, K. Hoppel, and S.A. Mango. Unsupervised estimation of speckle noise in radar images. *International Journal of Imaging Systems and Technology*, 4(4):298–305, 1992.

[52] M. Liévin, F. Luthon, and E. Keeve. Entropic estimation of noise for medical volume restoration. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 871–874. IEEE, 2002.

[53] H.W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, 1967.

[54] S.H. Lim. Characterization of noise in digital photographs for image processing. In *Proceedings of SPIE*, volume 6069, page 60690O, 2006.

[55] A. Liu. A fast method of estimating gaussian noise. In *Information Science and Engineering (ICISE), 2009 1st International Conference on*, pages 441–444. IEEE, 2009.

[56] T. Loupas, WN McDicken, and PL Allan. An adaptive weighted median filter for speckle suppression in medical ultrasonic images. *Circuits and Systems, IEEE Transactions on*, 36(1):129–135, 1989.

[57] VV Lukin, SK Abramov, ML Uss, IA Marusiy, NN Ponomarenko, AA Zelensky, B. Vozel, and K. Chehdi. Testing of methods for blind estimation of noise variance on large image database. *Book chapter in Theoretical and practical aspects of digital signal processing in information-telecommunication systems, Shakhty, Russia*, pages 43–70, 2009.

[58] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi. A nonlocal transform-domain filter for volumetric data denoising and reconstruction.

[59] D. Makovoz. Noise variance estimation in signal processing. In *Signal Processing and Information Technology, 2006 IEEE International Symposium on*, pages 364–369. IEEE, 2006.

[60] M. Miller and N. Kingsbury. Image denoising using derotated complex wavelet coefficients. *Image Processing, IEEE Transactions on*, 17(9):1500–1511, 2008.

[61] D.D. Muresan and T.W. Parks. Adaptive principal components and image denoising. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–101. IEEE, 2003.

[62] R.D. Nowak. Wavelet-based Rician noise removal for magnetic resonance imaging. *Image Processing, IEEE Transactions on*, 8(10):1408–1419, 1999.

[63] R. Oktem, K. Egiazarian, V. Lukin, N. Ponomarenko, and O. Tsymbal. Locally adaptive DCT filtering for signal-dependent noise removal. *EURASIP Journal on Advances in Signal Processing*, 2007:10, 2007.

[64] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti. TID2008 - a database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radioelectronics*, 10(4):30–45, 2009.

[65] NN Ponomarenko, VV Lukin, MS Zriakhov, A. Kaarna, and J. Astola. An automatic approach to lossy compression of AVIRIS images. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, pages 472–475. IEEE, 2007.

[66] W.H. Prass. *Numerical recipes: the art of scientific computing*. Cambridge University Press, 2007.

[67] P.R. Prucnal and B.E.A. Saleh. Transformation of image-signal-dependent noise into image-signal-independent noise. *Optics Letters*, 6(7):316–318, 1981.

[68] S. Pyatykh, J. Hesser, and L. Zheng. Signal-dependent noise parameter estimation by principal component analysis. submitted to Image Processing, IEEE Transactions on.

[69] S. Pyatykh, J. Hesser, and Lei Zheng. Efficient method of pixel neighborhood traversal. *Journal of Visual Communication and Image Representation*, 23(5):719–728, 2012.

[70] S. Pyatykh, J. Hesser, and Lei Zheng. Image noise level estimation by principal component analysis. *Image Processing, IEEE Transactions on*, 22(2):687–699, Feb. 2013.

[71] S. Pyatykh, L. Zheng, and J. Hesser. Fast noise variance estimation by principal component analysis. In *Image Processing: Algorithms and Systems XI, IS&T/SPIE Electronic Imaging 2013*. SPIE, 2013. to be presented.

[72] J. Rajan, D. Poot, J. Juntu, and J. Sijbers. Noise measurement from magnitude mri using local estimates of variance and skewness. *Physics in medicine and biology*, 55:N441, 2010.

[73] P. Rosin. Thresholding for change detection. In *Computer Vision, 1998. Sixth International Conference on*, pages 274–279. IEEE, 1998.

[74] S. Roth and M.J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.

[75] F. Russo. Gaussian noise estimation in digital images using nonlinear sharpening and genetic optimization. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–5. IEEE.

[76] M. Salmeri, A. Mencattini, G. Rabottino, and R. Lojacono. Signaldependent noise characterization for mammographic images denoising. In *IMEKO TC4 Symposium (IMEKOTC408)*, 2008.

[77] M. Salmeri, A. Mencattini, E. Ricci, and A. Salsano. Noise estimation in digital images using fuzzy processing. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 517–520. IEEE, 2001.

[78] U. Schmidt, K. Schelten, and S. Roth. Bayesian deblurring with integrated noise estimation. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2632, 2011.

[79] M. Sharma and S. Singh. Evaluation of texture methods for image analysis. In *Intelligent Information Systems Conference, The Seventh Australian and New Zealand 2001*, pages 117–121. IEEE, 2001.

[80] D.H. Shin, R.H. Park, S. Yang, and J.H. Jung. Block-based noise estimation using adaptive gaussian filtering. *Consumer Electronics, IEEE Transactions on*, 51(1):218–226, 2005.

[81] J. Sijbers, D. Poot, A.J. Den Dekker, and W. Pintjens. Automatic estimation of the noise variance from the histogram of a magnetic resonance image. *Physics in medicine and biology*, 52:1335, 2007.

[82] G. Smith and I. Burns. Measuring texture classification algorithms1. *Pattern Recognition Letters*, 18(14):1495–1501, 1997.

[83] J.L. Starck and F. Murtagh. Automatic noise estimation from the multiresolution support. *Publications of the Astronomical Society of the Pacific*, 110:193–199, 1998.

[84] G.W. Stewart and J. Sun. *Matrix perturbation theory*. Academic press San Diego, CA, 1990.

[85] T. Tasdizen. Principal components for non-local means image denoising. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1728–1731. IEEE, 2008.

[86] G. Torricelli, F. Argenti, and L. Alparone. Modelling and assessment of signal-dependent noise for image de-noising. In *11th European Conference on Signal Processing, EUSIPCO*, pages 287–290, 2002.

[87] M. Uss, B. Vozel, V. Lukin, S. Abramov, I. Baryshev, and K. Chehdi. Image informative maps for estimating noise standard deviation and texture parameters. *EURASIP Journal on Advances in Signal Processing, vol. Article ID*, 806516:12.

[88] M.L. Uss, B. Vozel, V.V. Lukin, and K. Chehdi. Local signal-dependent noise variance estimation from hyperspectral textural images. *Selected Topics in Signal Processing, IEEE Journal of*, 5(3):469–486, 2011.

[89] I. van Zyl Marais and W.H. Steyn. Noise estimation algorithms for onboard image quality assessment. In *International Conference on Space Technology*, 2009.

[90] J.S. Walker. Combined image compressor and denoiser based on tree-adapted wavelet shrinkage. *Optical Engineering*, 41:1520, 2002.

[91] C.M. Waternaux. Asymptotic distribution of the sample roots for a non-normal population. *Biometrika*, 63(3):639–645, 1976.

[92] J.H. Wilkinson. *The algebraic eigenvalue problem*. Oxford University Press, USA, 1988.

[93] P. Wyatt and H. Nakai. Developing nonstationary noise estimation for application in edge and corner detection. *Image Processing, IEEE Transactions on*, 16(7):1840–1853, 2007.

[94] J.C.K. Yan. Statistical methods for film grain noise removal and generation. Master's thesis, University of Toronto, 1997.

[95] S.M. Yang and S.C. Tai. Fast and reliable image-noise estimation using a hybrid approach. *Journal of Electronic Imaging*, 19(3):3007, 2010.

[96] D. Zoran and Y. Weiss. Scale invariance and noise in natural images. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2209–2216. IEEE, 2009.