TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 710 • Publication 710

Alessandro Foi

Pointwise Shape-Adaptive DCT Image Filtering and Signal-Dependent Noise Estimation

Tampere 2007

Tampereen teknillinen yliopisto. Julkaisu 710
Tampere University of Technology. Publication 710

Alessandro Foi

Pointwise Shape-Adaptive DCT Image Filtering and Signal-Dependent Noise Estimation

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB111, at Tampere University of Technology, on the 5th of December 2007, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2007

ISBN 978-952-15-1895-9 (printed)
ISBN 978-952-15-1922-2 (PDF)
ISSN 1459-2045

Abstract

When an image is acquired by a digital imaging sensor, it is always degraded by some noise. This leads to two basic questions: What are the main characteristics of this noise? How to remove it? These questions in turn correspond to two key problems in signal processing: *noise estimation* and *noise removal* (so-called *denoising*). This thesis addresses both abovementioned problems and provides a number of original and effective contributions for their solution. The *first part* of the thesis introduces a novel image denoising algorithm based on the low-complexity Shape-Adaptive Discrete Cosine Transform (SA-DCT). By using spatially adaptive supports for the transform, the quality of the filtered image is high, with clean edges and without disturbing artifacts. We further present extensions of this approach to image deblurring, deringing and deblocking, as well as to color image filtering. For all these applications, the proposed SA-DCT approach demonstrates a state-of-the-art filtering performance, which is achieved at a very competitive computational cost. The *second part* of the thesis addresses the problem of noise estimation. In particular, we consider noise estimation for raw-data, i.e. the unprocessed digital output of the imaging sensor. We introduce a method for nonparametric estimation of the standard-deviation curve which can be used with non-uniform targets under non-uniform illumination. Thus, we overcome key limitations of the existing approaches and standards, which typically assume the use of specially calibrated uniform targets. Further, we propose a noise model for the raw-data. The model is composed of a Poissonian part, for the photon sensing, and a Gaussian part, for the remaining stationary disturbances in the output data. The model explicitly takes into account the clipping of the data, faithfully reproducing the nonlinear response of the sensor when parts of the image are over- or under-exposed. This model allows for the parametric estimation of the noise characteristics from a single image. For this purpose, a fully automatic algorithm is presented. Numerous experiments with synthetic as well as with real data are presented throughout the thesis, proving the efficiency of the proposed solutions. Finally, illustrative examples, which show how the methods proposed in the first and in the second part can be integrated within a single procedure, conclude the thesis.

Foreword

This thesis is based on part of the research work which has been carried out by the author between May 2005 and August 2007, at the Institute of Signal Processing of Tampere University of Technology (TUT), Finland. Effectively, this work can be seen as a direct continuation and further development of the methods and results presented in the author's earlier doctoral thesis in Mathematics [47], defended in May 2005 at the Department of Mathematics of Politecnico di Milano, Italy.

This work has been accomplished under the supervision of Professors Karen Egiazarian and Vladimir Katkovnik. I am greatly indebted towards both of them, for their highly professional guidance and for being incessantly patient and supportive. The high trust they have placed in me is possibly the greatest reward of my endeavors.

My special thanks go to Professors Jaakko Astola, director of the Signal Processing Algorithm Group (SPAG) and of Tampere International Center for Signal Processing (TICSP), and to Moncef Gabbouj, director of the Institute of Signal Processing, for the excellent research environment which they have offered me since I came to Tampere in November 2002.

The work presented in this thesis has been done within research projects of the Academy of Finland's Finnish Centre of Excellence program, of Nokia Research Center (NRC), Tampere, and of Tekes, the Finnish Funding Agency for Technology and Innovation. Without their financial support, gratefully acknowledged, this research would have not been possible.

This thesis is dedicated to the fragile and inspiring beauty of the Finnish Nature and especially to lake Särkijärvi, where many of the ideas enclosed in this volume took shape.

Tampere, November 2007 Alessandro Foi

Contents

Introduction to the thesis

Outline of the thesis

The thesis is structured in two self-contained parts. The first part deals with a family of image filtering algorithms collectively called "Pointwise Shape-Adaptive DCT filters". The key idea of these algorithms is to use a shape-adaptive transform on a support that adapts to the images features. Specifically, we use the low-complexity SA-DCT transform [154] in conjunction with the Anisotropic Local Polynomial Approximation (LPA) - Intersection of Confidence Intervals (ICI) technique [47],[91],[92], which defines the shape of the transform's support in a pointwise adaptive manner. The thresholded or attenuated SA-DCT coefficients are used to reconstruct a local estimate of the signal within the adaptive-shape support. Since supports corresponding to different points are in general overlapping, the local estimates are averaged together using adaptive weights that depend on the region's statistics.

Besides a "basic" grayscale image denoising filter for additive white Gaussian noise (Chapter 1, Section 1.5, p. 11), we present extensions of the approach to:

- color image filtering (Section 1.7, p. 24);
- deringing and deblocking of B-DCT compressed image (e.g., JPEG images) (Section 1.6, p. 20);
- deconvolution (deblurring) (Chapter 2, p. 49);
- signal-dependent noise removal (Chapter 3, p. 65).

A separate chapter is dedicated to another class of transforms defined on arbitrarily shaped supports, the so-called "shape-adapted" bases (Chapter 4, p. 77). In particular, we evaluate their filtering performance when they are used within the above algorithms in place of the more efficient SA-DCT.

An appendix on the LPA-ICI technique is provided for the reader's convenience (Appendix A, p. 85). Several illustrations of the different basis elements used in the considered shape-adaptive and "shape-adapted" transforms are given at the end of this first part (Appendix B, p. 89).

The second part of the thesis deals with noise estimation for the raw-data of digital imaging sensors. This part comprises two chapters. First (Chapter 5, p. 123), we present a nonparametric method for estimation of the curve which describes the standard-deviation of the noise as a function of the expectation of the pixel raw-data output (so-called “standard-deviation function”). This method relies on the analysis of several images captured under the same fixed acquisition and illumination conditions. However, we do not require the target or the illumination to be uniform. Based on an automatic segmentation of the images, we separate samples with different expected output and estimate their standard-deviations. Thus, while other techniques require a uniform target (e.g., a test-chart), in our approach we benefit from the target non-uniformity by simultaneously estimating the standard-deviation function over a large range of output values.

The most significant contributions of this second part are given in the subsequent chapter (Chapter 6, p. 135), where we propose a noise model for the raw-data and an algorithm for the fully automatic estimation of these parameters from a single noisy image. The model is composed of a Poissonian part, for the photon sensing, and Gaussian part, for the remaining stationary disturbances in the output data. The model explicitly takes into account the clipping of the data, faithfully reproducing the nonlinear response of the sensor when parts of the image are over- or underexposed. The parameter estimation algorithm utilizes a special maximum-likelihood fitting of the parametric model on a collection of local wavelet-domain estimates of mean and standard-deviation. Experiments with synthetic images and real raw-data from camera sensors demonstrate the effectiveness and accuracy of the algorithm in estimating the model parameters and confirm the validity of the proposed model.

Conclusive remarks and few illustrative examples, which show how the methods proposed in the first and in the second part can be integrated within a single procedure, are given at the end of thesis (Conclusions, p. 173).

Publications and author’s contribution

Most of the material presented in this thesis appears in the following publications by the author:

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

- [59]: Foi, A., V. Katkovnik, and K. Egiazarian, “Pointwise Shape-Adaptive DCT denoising with structure preservation in luminance-chrominance space”, *Proc. 2^{nd} Int. Workshop Video Process. Quality Metrics Consum. Electron., VPQM2006*, Scottsdale, AZ, January 2006.

- [60]: Foi, A., V. Katkovnik, and K. Egiazarian, “Pointwise Shape-Adaptive DCT for high-quality deblocking of compressed color images”, *Proc. 14th Eur. Signal Process. Conf., EUSIPCO 2006*, Florence, September 2006.

- [53]: Foi, A., K. Dabov, V. Katkovnik, and K. Egiazarian, “Shape-Adaptive DCT for denoising and image reconstruction”, *Proc. SPIE El. Imaging 2006, Image Process.: Algorithms and Systems V*, 6064A-18, January 2006.

- [54]: Foi, A., and V. Katkovnik, "From local polynomial approximation to pointwise shape-adaptive transforms: an evolutionary nonparametric regression perspective", *Proc. 2006 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006*, Florence, September 2006.
- [61]: Foi, A., V. Katkovnik, and K. Egiazarian, "Signal-dependent noise removal in Pointwise Shape-Adaptive DCT domain with locally adaptive variance", *Proc. 15th Eur. Signal Process. Conf., EUSIPCO 2007*, Poznań, September 2007.
- [48]: Foi, A., S. Alenius, V. Katkovnik, and K. Egiazarian, "Noise measurement for raw-data of digital imaging sensors by automatic segmentation of non-uniform targets", *IEEE Sensors Journal*, vol. 7, no. 10, pp. 1456-1461, October 2007.
- [64]: Foi, A., V. Katkovnik, D. Paliy, K. Egiazarian, M. Trimeche, S. Alenius, R. Bilcu, M. Vehvilainen, "Apparatus, method, mobile station and computer program product for noise estimation, modeling and filtering of a digital image", *U.S. Patent* (Applications no. 11/426,128, June 2006, and no. 11/519,722, September 2006).
- [63]: Foi, A., M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single image raw-data", *to appear in IEEE Trans. Image Process.*

The contents of this thesis are also closely related to the following publications by the author:

- [49]: Foi, A., S. Alenius, M. Trimeche, and V. Katkovnik, "Adaptive-size block transforms for Poissonian image deblurring", *Proc. 2006 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006*, Florence, September 2006.
- [52]: Foi, A., R. Bilcu, V. Katkovnik, and K. Egiazarian, "Adaptive-Size Block Transforms for Signal-Dependent Noise Removal", *Proc. 7th Nordic Signal Processing Symposium, NORSIG 2006*, Reykjavik, Iceland, June 2006.

The whole thesis and the publications cited above represent original work, of which the author has been the main contributor. In particular, all methods and algorithms presented in the thesis have been originally proposed and developed by the author. However, this work would not have been possible without the support, inspiration, criticism, and help of expert co-authors as well as of many (non co-author) colleagues, among whom I wish to mention Dmitriy Paliy, Atanas Gotchev, Ossi Pirinen, and Giacomo Boracchi. The numerous suggestions which came from the anonymous reviewers of the above publications and from the reviewers of the draft of the thesis, Javier Portilla and Ivan Selesnick, have been an additional source of improvement in the presentation of this material. I wish especially to acknowledge Kostadin Dabov for his skillful C implementation of the SA-DCT filters, which permitted the rapid development of the Pointwise SA-DCT algorithms.

All co-authors have confirmed their agreement on the above statement.

Notation and conventions

We tried to use, as much as possible, well-known notation. However, since even for the most basic concepts there exist in the literature various and equivocal notations, we gradually explain in the text the meaning of the used notation. Nevertheless, as a useful reference, we declare here below, some of the most significant conventions that we follow.

The symbols $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}$, and $\mathbb{N}$ indicate, respectively, the real numbers, the non-negative real numbers, the relative numbers (i.e., integers), and the natural numbers. Given a discrete set X, the symbol $|X|$ stands for the cardinality of X, i.e. the number of elements in the set. The same notation is used also to indicate the measure of a continuous set.

The symbol $\circledast$ denotes the convolution, $(g \circledast z)(x) = \int g(x-v)\, z(v)\, dv \quad \forall x$. The central dot $\cdot$ stands for a "mute variable" or "mute argument". For example, the above definition of the convolution can be written also as $g \circledast z = \int g(\cdot - v)\, z(v)\, dv$ (x being the mute variable). On discrete domains the integration is intended with respect to a discrete measure (typically the counting measure). We denote the ℓ^p norms as $\|\cdot\|_p$ and the inner-product as $\langle \cdot, \cdot \rangle$. The Fourier transform is denoted by $\mathcal{F}$. We usually indicate the Fourier transform of a function with the corresponding capital letter: $\mathcal{F}(g) = G$. We use a discrete Fourier transform normalized in such a way that the convolution theorem and Parseval's equality hold in the forms $\mathcal{F}(f \circledast g) = FG$ and $\|f\|_2^2 = \frac{1}{|X|}\|\mathcal{F}(f)\|_2^2$, where f is a function defined on the domain X. This is the standard normalization used in MATLAB.

The conjugate transpose of a matrix or vector is denoted by the superscript ^{T}.

The hat decoration $\widehat{\ }$ denotes estimated values (e.g., "$\hat{y}$ is the estimate of y").

The $\lfloor \cdot \rfloor$ brackets indicate the rounding to the nearest smaller or equal integer, e.g., $\lfloor 7 - \pi \rfloor = 3$.

For a function $f : X \to \mathbb{R}$, we define its support, $\operatorname{supp} f \subseteq X$, as the subset on which the function is non-zero, $\operatorname{supp} f = \{x \in X : f(x) \neq 0\}$. The other way round, for a subset $A \subseteq X$, we define its characteristic function, $\chi_A : X \to \{0, 1\}$, as the binary function that has A as its support: $\chi_A(x) = 1 \iff x \in A$, $\operatorname{supp} \chi_A = A$.

$\mathcal{N}(\mu, \sigma^2)$ and $\mathcal{P}(\lambda)$ respectively denote the normal (i.e., Gaussian) distribution with mean μ and variance σ^2 and the Poisson distribution with mean (and variance) λ. The notation $z \sim \mathcal{P}(\lambda)$ means that the random variable z is distributed according to a $\mathcal{P}(\lambda)$ distribution, which implies that the probability of z being equal to k is $P(z = k) = e^{-\lambda}\frac{\lambda^k}{k!}$, $k \in \mathbb{N}$. Similarly, if $z \sim \mathcal{N}(\mu, \sigma^2)$, we have that the probability density of z is $\wp(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)}{2\sigma^2}}$, $z \in \mathbb{R}$.

For the images used in the many figures and simulations, unless differently noted, we assume that the data-range is $[0, 255]$, where 0 and 255 correspond, respectively, to black and white.

We use the following standard criteria functions to assess the objective quality of an estimate $\hat{y}$ of y, obtained from a noisy observation z (where all the signals are defined on a domain X of size $|X|$):

$$\begin{aligned}
\text{(signal-to-noise ratio)}\ \ \mathrm{SNR} &= 20\log_{10}\left(\frac{\|y\|_2}{\|y-\hat{y}\|_2}\right),\\
\text{(improvement in SNR)}\ \ \mathrm{ISNR} &= 20\log_{10}\left(\frac{\|y-z\|_2}{\|y-\hat{y}\|_2}\right),\\
\text{(peak SNR)}\ \ \mathrm{PSNR} &= 20\log_{10}\left(\frac{255\cdot\sqrt{|X|}}{\|y-\hat{y}\|_2}\right),\\
\text{(mean squared error)}\ \ \mathrm{MSE} &= \frac{\|y-\hat{y}\|_2^2}{|X|}.
\end{aligned}$$

Additionally, to measure the noisiness of a blurred image, we use the blurred SNR (BSNR), defined as

$$\text{(blurred SNR)}\ \ \mathrm{BSNR} = 20\log_{10}\left(\frac{\|y_{\mathrm{blur}}-\mathrm{mean}(y_{\mathrm{blur}})\|_2}{\|y_{\mathrm{blur}}-z\|_2}\right),$$

where y_{blur} is the noise-free blurred image and z is the noisy blurred image.

For convenience, in Table 1, we list the most frequent abbreviations used in the thesis.

1-D, **2-D**, **3-D**	One-, Two-, Three-Dimensional
AC	Alternating Current (non-constant components)
AWGN	Additive White Gaussian Noise
bpp	bits per pixel
BSNR	Blurred Signal-to-Noise Ratio
B-DCT	Block Discrete Cosine Transform (e.g., 8×8 2-D DCT)
CCD	Charge-Coupled Device
CDF	Cumulative Distribution Function
CMOS	Complementary(-symmetry) Metal-Oxide Semiconductor
CR	first column-wise and then row-wise
dB	decibel
DC	Direct Current (constant component)
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
FPN	Fixed-Pattern Noise
GS	Gram-Schmidt orthonormalization procedure

Table 1: List of frequently used abbreviations (continued on next page).

JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loève Transform
ICI	Intersection of Confidence Intervals
IDCT	Inverse Discrete Cosine Transform
i.i.d.	independent and identically distributed
ISNR	Improvement in Signal-to-Noise Ratio
LPA	Local Polynomial Approximation
LS	Least Squares
MAD	Median Absolute Deviation
ML	Maximum Likelihood
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
PCA	Principal Component Analysis
PDF	Probability Density Function
PSF	Point-Spread Function
PSNR	Peak Signal-to-Noise Ratio
RC	first row-wise and then column-wise
RGB	Red, Green, Blue color components
RI	Regularized Inverse
RWI	Regularized Wiener Inverse
SA-DCT	Shape-Adaptive Discrete Cosine Transform
"SA"-DCT	"Shape-Adapted" Discrete Cosine Transform
SNR	Signal-to-Noise Ratio
asvar	average sample variance
std	standard deviation
supp	support of a function
svar	sample variance
tsvar	total sample variance
YUV	luminance (Y) and chrominance (U,V) color components
var	variance

Table 1 (continued from previous page): List of frequently used abbreviations.

Part I

Pointwise Shape-Adaptive DCT filters

Chapter 1

Pointwise Shape-Adaptive DCT denoising and deblocking

1.1 Introduction

The two-dimensional separable block-DCT (B-DCT), computed on a square or rectangular support, is a well established and very efficient transform in order to achieve a sparse representation of image blocks. For natural images, its decorrelating performance is close to that of the optimum Karhunen-Loève transform [141]. Thus, the B-DCT has been successfully used as the key element in many compression and denoising applications. However, in presence of singularities or edges such near-optimality fails. Because of the lack of sparsity, edges cannot be coded or restored effectively, and ringing artifacts arising from the Gibbs phenomenon become visible. For this reason, other transforms with better edge adaptation capabilities (e.g., wavelets [112], curvelets [158], etc.) have been developed and used in denoising, and post-processing (deringing, deblocking) filters are commonly used in MPEG-video decoders [153].

In the last decade, significant research has been made towards the development of region-oriented, or *shape-adaptive*, transforms. The main intention is to construct a system (frame, basis, etc.) that can efficiently be used for the analysis and synthesis of arbitrarily shaped image segments, where the data exhibit some uniform behavior.

Initially, Gilge [70, 71] considered the orthonormalization of a (fixed) set of generators restricted to the arbitrarily shaped region of interest. These generators could be a basis of polynomials or – for example – a B-DCT basis, thus yielding a "shape-adapted" DCT transform. Orthonormalization can be performed by the standard Gram-Schmidt procedure and the obtained orthonormal basis is supported on the region. Because the region-adapted basis needs to be recalculated for each differently shaped region and because the basis elements are typically non-separable, the overall method presents a rather high computational cost. While even today it is considered as one of the best solutions to the region-oriented transforms problem, Gilge's approach is clearly unsuitable for real-time applications, and faster transforms were sought.

A more computationally attractive approach, namely the *shape-adaptive DCT* (SA-DCT), has been proposed by Sikora et al. [154, 156]. The SA-DCT is computed by cascaded application of one-dimensional varying-length DCT transforms first on the columns and then on the rows that constitute the considered region, as shown in Figure 1.1. Thus, the SA-DCT does not require costly matrix inversions or iterative orthogonalizations and can be interpreted as a direct generalization of the classical 2-D B-DCT transform. In particular, the SA-DCT and the B-DCT (which is separable) have the same computational complexity and in the special case of a square the two transforms exactly coincide. Therefore, the SA-DCT has received considerable interest from the MPEG community, eventually becoming part of the MPEG-4 standard [102, 117]. The recent availability of low-power SA-DCT hardware platforms (e.g., [18],[100],[101]) makes this transform an appealing choice for many image- and video-processing tasks.

The SA-DCT has been shown [154, 155, 12, 96] to provide a compression efficiency comparable to those of more computationally complex transforms, such as the "shape-adapted" DCT [71]. The good decorrelation and energy compaction properties on which this efficiency depends are also the primary characteristics sought for any transform-domain denoising algorithm. In this sense, the SA-DCT features a remarkable potential not only for video compression applications, but also for image and video denoising.

However, this potential has been apparently ignored by the image denoising and restoration community. While this indifference may seem rather surprising, there are sound reasons that can justify it.

The use of a transform with a shape-adaptive support involves actually two separate problems: not only the transform should adapt to the shape (i.e. a shape-adaptive transform), but the shape itself must adapt to the image features (i.e. an image-adaptive shape). The first problem has found a very satisfactory solution in the SA-DCT transform. How to deal with the second problem depends on the considered application.

The shape-adaptive coding of *noise-free* video objects always assumes that the boundary of these objects is known. This information can be obtained either from a-priori knowledge (e.g., motion estimation, chroma keying, layered structure), or it can be estimated from the data with one of the many automated segmentation algorithms suitable for this purpose (e.g., [123]). On the contrary, obtaining an accurate and robust segmentation of noisy data constitutes an extremely more complex task than the region-oriented coding itself. Unlike in video coding, such a segmentation cannot be reasonably assumed to be known a-priori. It must be noted that conventional segmentation (or local-segmentation) techniques which are employed for video processing are not suitable for degraded (noisy, blurred, highly compressed, etc.) data. This very aspect may be identified as the principal reason why the SA-DCT had not been used for the restoration of noisy images.

In our approach, we use the SA-DCT in conjunction with the Anisotropic Local Polynomial Approximation (LPA) - Intersection of Confidence Intervals (ICI) [91, 90, 72, 62], a technique purposely designed to work accurately with noisy data.

By comparing varying-scale directional kernel estimates, this technique adaptively selects, for each point in the image, a set of directional adaptive-scales. The length of the support (i.e., the window size) of the corresponding adaptive-scale kernels define the shape of the transform's support in a pointwise-adaptive manner. Examples of such neighborhoods are shown in Figures 1.2, 1.5, 1.7, and 1.19.

For each one of these neighborhoods a SA-DCT is performed. The hard-thresholded SA-DCT coefficients are used to reconstruct a local estimate of the signal within the adaptive-shape support. By using the adaptive neighborhoods as support for the SA-DCT, we ensure that data are represented sparsely in the transform domain, allowing to effectively separate signal from noise using hard-thresholding.

Since supports corresponding to different points are in general overlapping (and thus generate an overcomplete representation of the signal), the local estimates are averaged together using adaptive weights that depend on the local estimates' statistics. In this way we obtain an adaptive estimate of the whole image.

Once this global estimate is produced, it can be used as reference estimate for an empirical Wiener filter [69, 68] in SA-DCT domain. Following the same adaptive averaging procedure as for hard-thresholding, we arrive to the final Anisotropic LPA-ICI-driven SA-DCT estimate. We term our approach "*Pointwise SA-DCT filtering*".

We present this novel approach for the denoising of grayscale as well as of color images. Extension to color images is based on a luminance-chrominance color-transformation and exploits the structural information obtained from the luminance channel to drive the shape-adaptation for the chrominance channels. Such adaptation strategy enables accurate preservation and reconstruction of image details and structures and yields estimates with a very good visual quality. Additionally, we discuss and analyze its application to deblocking and deringing of block-DCT compressed images. Particular emphasis is given to the deblocking of highly-compressed color images.

Since the SA-DCT is implemented as standard in modern MPEG hardware, the proposed techniques can be integrated within existing video platforms as a pre- or post-processing filter.

The chapter is organized as follows. We begin with the considered observation model and notation. In Section 1.3 we recall the main points of the Anisotropic LPA-ICI technique. Various aspects and details of the shape-adaptive DCT transform are given in Section 1.4. The proposed *Pointwise SA-DCT denoising* algorithm is then introduced in Section 1.5, which constitutes the core of the chapter. The application to deblocking and deringing is given in Section 1.6, where we relate the quantization table with the value of the variance to be used for the filtering. In Section 1.7 we present the extension of the proposed methods to color image filtering, describing the employed color-space transformations and the structural constraints which are imposed on the chrominances. The last section of the chapter is devoted to results and discussions: we provide a comprehensive collection of experiments and comparisons which demonstrate the advanced performance of the proposed algorithms.

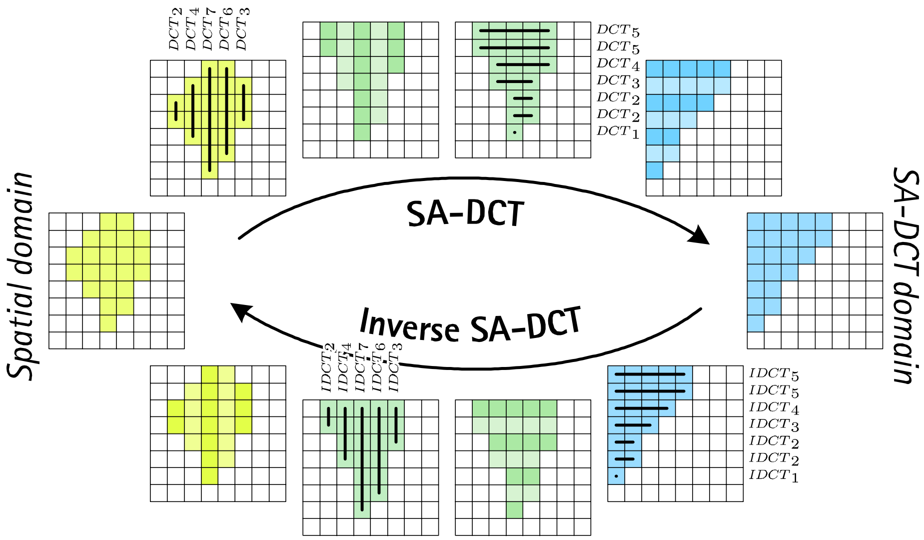


Figure 1.1: Illustration of the shape-adaptive DCT transform and its inverse. Transformation is computed by cascaded application of one-dimensional varying-length DCT transforms, along the columns and along the rows.

1.2 Observation model and notation

We consider noisy observations z of the form

$$z(x) = y(x) + \eta(x), \qquad x \in X, \tag{1.1}$$

where $y : X \to \mathbb{R}$ is the original image, η is i.i.d. Gaussian white noise, $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$, and x is a spatial variable belonging to the image domain $X \subset \mathbb{Z}^2$. At the beginning we restrict ourself to grayscale images (and thus scalar functions), while later (from Section 1.7) we consider also color images.

Given a function $f : X \to \mathbb{R}$, a subset $U \subset X$, and a function $g : U \to \mathbb{R}$, we denote by $f_{|U} : U \to \mathbb{R}$ the restriction of f on U, $f_{|U}(x) = f(x)$ $\forall x \in U$, and by $g^{|X} : X \to \mathbb{R}$ the zero-extension of g to X, $\left(g^{|X}\right)_{|U} = g$ and $g^{|X}(x) = 0$ $\forall x \in X \setminus U$. The characteristic (indicator) function of U is defined as $\chi_U = 1_{|U}{}^{|X}$. We denote by $|U|$ the cardinality (i.e. the number of its elements) of U. The symbol "$\circledast$" stands for the convolution operation.

1.3 Anisotropic LPA-ICI

The approach is based on the Intersection of Confidence Intervals (ICI) rule, a method originally developed for pointwise adaptive estimation of 1-D signals [72, 90]. The idea has been generalized for 2-D image processing, where adaptive-size quadrant windows have been used [93]. Significant improvement of this approach has been achieved on the basis of anisotropic directional estimation [91, 55]. Multidirectional sectorial-neighborhood estimates are calculated for every point

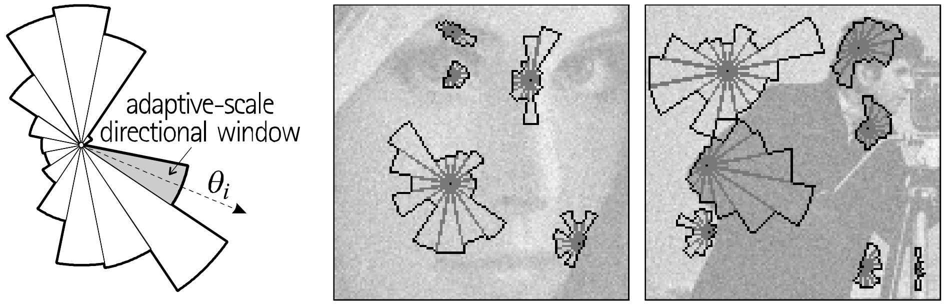


Figure 1.2: Anisotropic LPA-ICI. From left to right: sectorial structure of the anisotropic neighborhood achieved by combining a number of adaptive-scale directional windows; some of these windows selected by the ICI for the noisy *Lena* and *Cameraman* images.

and the ICI rule is exploited for the adaptive selection of the size of each sector. Thus, the estimator is anisotropic and the shape of its support adapts to the structures present in the image. In Figure 1.2 we show some examples of these anisotropic neighborhoods for the *Lena* and *Cameraman* images. The developed anisotropic estimates are highly sensitive with respect to change-points, and allow to reveal fine elements of images from noisy observations.

Let us present the Anisotropic LPA-ICI method in more detail. For every specified direction θ_k, $k = 1, \ldots, K$, a varying-scale family of directional-LPA convolution kernels $\{g_{h,\theta_k}\}_{h\in H}$ is used to obtain a corresponding set of directional varying-scale estimates $\{\hat{y}_{h,\theta_k}\}_{h\in H}$, $\hat{y}_{h,\theta_k} = z \circledast g_{h,\theta_k}$, $h \in H$, where $H \subset \mathbb{R}^+$ is the set of scales. These estimates are then compared according to the ICI rule, and as a result an adaptive scale $h^+(x, \theta_k) \in H$ is defined for every $x \in X$ and for every direction θ_k. The corresponding adaptive-scale estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$ are then "fused" together in an adaptive convex combination in order to yield the final anisotropic LPA-ICI estimate.

However, here we are not interested in this anisotropic estimate. Instead, we consider only the adaptive neighborhood U_x^+, constructed as the union of the supports of the directional adaptive-scale kernels $g_{h^+(x,\theta_k),\theta_k}$,

$$U_x^+ = \bigcup_{k=1}^{K} \operatorname{supp} g_{h^+(x,\theta_k),\theta_k},$$

which we use as the support for a shape-adaptive transform. Observe that, being convolution kernels, the LPA kernels g_{h,θ_k} are always "centered" at the origin, therefore U_x^+ is a neighborhood of the origin. The actual adaptive neighborhood of x, which contains the observations that are used for estimation, is instead

$$\tilde{U}_x^+ = \left\{ v \in X : (x - v) \in U_x^+ \right\}.$$

The neighborhoods shown in Figure 1.2 are in fact examples of $\tilde{U}_x^+$ for a few points $x \in X$.

Let us remark that there is a substantial difference between image segmentation, in which the image is decomposed in a limited number ($\ll |X|$) of non-overlapping subsets (image segments), and the Anisotropic LPA-ICI, which for every $x \in X$ provides an adaptive neighborhood $\tilde{U}_x^+$ of x. In particular, because of the nonparametric nature of the procedure, neighborhoods corresponding to adjacent points do usually overlap.

1.4 Shape-Adaptive DCT transform

The SA-DCT [154, 156] is computed by cascaded application of one dimensional varying-length DCT transforms first on the columns and then on the rows that constitute the considered region. Several improvements over its original definition have been proposed. We exploit the most significant [96], which concern the normalization of the transform and the subtraction of the mean and which have a fundamental impact on the use of the SA-DCT for image filtering. Additionally, an alternative scheme for the coefficients' alignment is also utilized and the possibility of processing first by rows and then by columns is considered.

1.4.1 Orthonormal Shape-Adaptive DCT

The normalization of the SA-DCT is obtained by normalization of the individual one-dimensional transforms that are used for transforming the columns and rows. In terms of their basis elements, they are defined as:

$$\psi_{L,m}^{\text{1D-DCT}}(n) = c_m \cos\left(\tfrac{\pi m(2n+1)}{2L}\right), \quad m,n=0,\dots,L-1, \tag{1.2}$$

$$c_0 = \sqrt{1/L}, \quad c_m = \sqrt{2/L},\ m>0. \tag{1.3}$$

Here L stands for the length of the column or row to be transformed. The normalization in (1.2) is indeed the most natural choice, since in this way all the transforms used are orthonormal and the corresponding matrices belong to the orthogonal group. Therefore, the SA-DCT – which can be obtained by composing two orthogonal matrices – is itself an orthonormal transform. A different normalization of the 1-D transforms would produce, on an arbitrary shape, a 2-D transform that is non-orthogonal (for example as in [154, 156], where $c_0 = \sqrt{2}/L$ and $c_m = 2/L$ for $m > 0$).

Let us denote by $T_U : \mathcal{U} \to \mathcal{V}_U$ the orthonormal SA-DCT transform obtained for a region $U \subset X$, where $\mathcal{U} = \{f : U \to \mathbb{R}\}$ and $\mathcal{V}_U = \{\varphi : V_U \to \mathbb{R}\}$ are function spaces and $V_U \subset \mathbb{Z}^2$ indicates the domain of the transform coefficients. Let $T_U^{-1} : \mathcal{V}_U \to \mathcal{U}$ be the inverse transform of T_U. We indicate the thresholding (or quantization) operator as Υ. Thus, the SA-DCT-domain processing of the observations z on a region U can be written as $\hat{y}_U = T_U^{-1}\left(\Upsilon\left(T_U\left(z_{|U}\right)\right)\right)$, $\hat{y}_U : U \to \mathbb{R}$. From the orthonormality of T and the model (1.1) follows that $T_U\left(z_{|U}\right) = T_U\left(y_{|U}\right) + \bar{\eta}$, where $\bar{\eta} = T_U(\eta_{|U})$ is again Gaussian white noise with variance σ^2 and zero mean.

1.4.2 Mean subtraction

There is, however, an adverse consequence of the normalization (1.2). Even if the signal restricted to the shape $z_{|U}$ is constant, the reconstructed $\hat{y}_U$ is usually

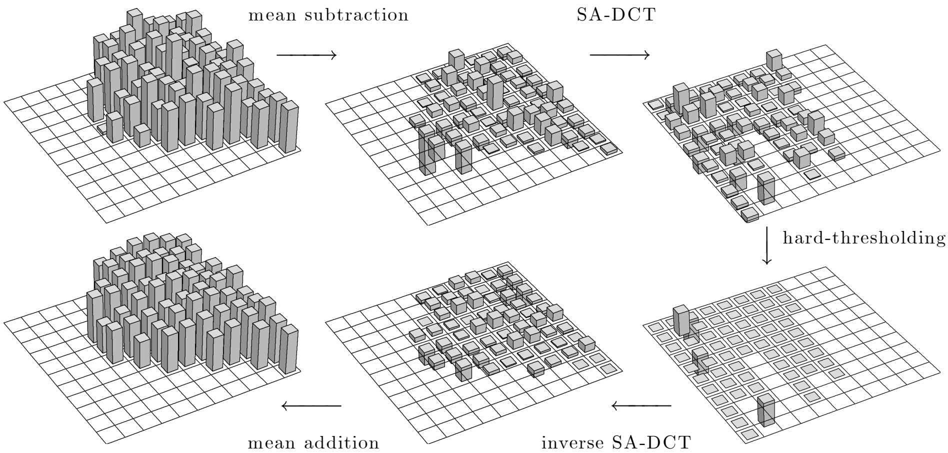


Figure 1.3: Hard-thresholding in SA-DCT domain. The image data on an arbitrarily shaped region is subtracted of its mean. The zero-mean data is then transformed and thresholded. After inverse transformation, the mean is added back.

non-constant. In [95, 96] this behavior is termed as "mean weighting defect", and it is proposed there to attenuate its impact by applying the orthonormal SA-DCT on the zero-mean data which is obtained by subtracting from the initial data z its mean. After the inversion, the mean is added back to the reconstructed signal $\hat{y}_U : U \to \mathbb{R}$:

$$\hat{y}_U = T_U^{-1}\left(\Upsilon\left(T_U\left(z_{|U} - m_U(z)\right)\right)\right) + m_U(z), \tag{1.4}$$

where $m_U(z) = \frac{1}{|U|}\sum_{x\in U} z(x)$ is the mean of z on U.

Although this operation – which is termed "DC separation" – is not fully justified from the approximation theory standpoint (because $m_U(z)$ is calculated from the noisy data, and by subtracting it the noise in the coefficients is no longer white), it produces visually superior results without affecting to the objective restoration performance. The DC separation (together with a special compensation called "ΔDC correction") are also considered in MPEG-4 [117].

1.4.3 Coefficient alignment

To further improve the efficiency of the SA-DCT, it has been proposed to align the coefficients obtained after the first 1-D transformation along the rows in such a way as to maximize their vertical correlation before applying the second transform along the columns. Different strategies, based on different models of the underlying signal y, have been suggested (e.g., [11], [2]). Although they can provide a significant improvement when the data agrees with the assumed signal's model, in practice when dealing with real data only marginal improvement can be achieved over the basic alignment used in [154, 156], where coefficients with the same index m (i.e. all DC terms, all first AC terms, etc.) are aligned in the same columns, regardless of the length L of the current row.

In our implementation we use the following alignment formula, denoting by m and m' the old (i.e. the one coming from (1.2)) and new coefficient index,

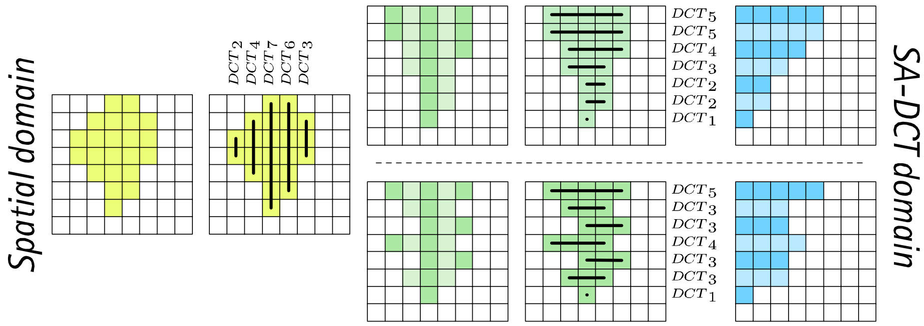


Figure 1.4: Comparison between the basic alignment $m' = m$ (top) and the alignment $m' = \lfloor m L_{\max}/L \rfloor$ (bottom) in the forward SA-DCT transform (see Section 1.4.3).

respectively: $m' = \lfloor m L_{\max}/L \rfloor$, where L is the length of the current row, $m = 0, \ldots, L-1$, $L_{\max}$ is the length of the longest row in U, and the $\lfloor \cdot \rfloor$ brackets indicate the rounding to the nearest integer smaller or equal to $(\cdot)$. Figure 1.4 provides a comparison between these two different ways to align the coefficients after the first 1-D transform.

An illustration of the SA-DCT-domain hard-thresholding, performed according to (1.4) and to the above coefficient alignment formula is given in Figure 1.3.

1.4.4 Non-separability and column-row (CR) vs. row-column (RC) processing

We note that even though the SA-DCT is implemented like a separable 2-D transform, using cascaded 1-D transforms on columns and rows, in general it is not separable, i.e. the corresponding 2-D basis elements[1] cannot always be written as the product of pairs of univariate functions (otherwise the support would necessarily be rectangular). Thus, two possibly different sets of transform coefficients are obtained when processing the considered region first column-wise and then row-wise (CR mode) or first row-wise and then column-wise (RC mode). Similarly, the basis elements and the transform coefficients for a region may differ from the corresponding transposed basis elements and coefficients of the same region transposed[2]. Experimental analysis has shown that the orientation of the shape does have some impact on the efficiency of the SA-DCT and that, typically, slightly sparser decompositions are achieved if the region is transformed processing along its longest orientation first. This fact had been already implicitly discussed in the original paper by Sikora and Makai [156]. The orientation of the adaptive neighborhood $\tilde{U}_x^+$ can be deduced easily from the adaptive scales $\{h^+(x,\theta_k)\}_{k=1}^K$ given by the ICI; thus we use the scales to decide whether to apply the SA-DCT in RC or CR mode. Clearly, the inverse SA-DCT has to be computed accordingly.

[1] A formal definition of the SA-DCT basis elements is given by Eq. (1.12) in Section 1.5.3.
[2] Illustrations of this phenomenon are presented in the Appendix A.2.

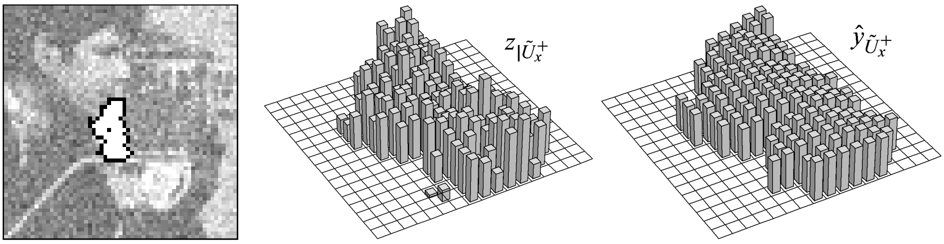


Figure 1.5: From left to right: a detail of the noisy *Cameraman* showing an adaptive-shape neighborhood $\tilde{U}_x^+$ determined by the Anisotropic LPA-ICI procedure, and the image intensity corresponding to this region before and after hard-thresholding in SA-DCT domain.

1.5 Pointwise SA-DCT denoising

We use the anisotropic adaptive neighborhoods $\tilde{U}_x^+$ defined by the LPA-ICI as supports for the SA-DCT, as shown in Figure 1.5.

By demanding the local fit of a polynomial model, we are able to avoid the presence of singularities or discontinuities within the transform support. In this way, we ensure that data are represented sparsely in the transform domain, significantly improving the effectiveness of thresholding.

Before we proceed further, it is worth mentioning that the proposed approach can be interpreted as a special kind of local model selection which is adaptive with respect both to the scale and to the order of the utilized model. Adaptivity with respect to the scale is determined by the LPA-ICI, whereas the order-adaptivity is achieved by hard-thresholding. "Shape-adapted" orthogonal polynomials are the most obvious choice for the local transform, as they are more consistent with the polynomial modeling used to determine its support. However, in practice, cosine bases are known to be more adequate for the modeling of natural images. In particular, when image processing applications are of concern, the use of computationally efficient transforms is paramount, and thus in this chapter we restrict ourself to the low-complexity SA-DCT. "Shape-adapted" transforms (polynomial as well as cosine) are the subject of Chapter 4. We refer the interested reader to [54], where our approach is considered within the more general theoretical framework of nonparametric regression.

1.5.1 Fast implementation of the anisotropic neighborhood

In practice, we do not need a variety of different shapes as broad as in the examples of Figures 1.2 and 1.5. A much simplified neighborhood structure is used in our implementation. Narrow one-dimensional "linewise" directional LPA kernels $\{g_{h,\theta_k}\}_{h\in H}$ are used for $K = 8$ directions with scales $H = \{1, 2, 3, 5, 7, 9\}$. Figure 1.6 shows these kernels for $\theta_1 = 0$; the diagonal kernels $\{g_{h,\theta_2}\}_{h\in H}$, $\theta_2 = \frac{\pi}{4}$, are made by slanting the corresponding horizontal kernels $\{g_{h,\theta_1}\}_{h\in H}$ and the kernels for all remaining six directions are obtained by repeated 90-degrees rotations of these two sets. After the ICI-based selection of the adaptive-scales $\{h^+(x, \theta_k)\}_{k=1}^{8}$, the neighborhood U_x^+ is the octagon constructed as the polygo-

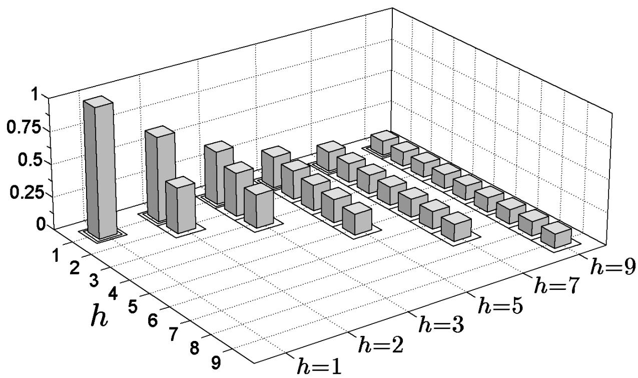


Figure 1.6: "Linewise"one-dimensional directional LPA kernels having scale and length of support equal to $h = 1, 2, 3, 5, 7, 9$ pixels. The origin pixel is marked with a double outline. Kernels for diagonal (e.g., 45 degrees) directions are obtained by slanting the shown kernels; thus, with respect to the Euclidean metric, the diagonal kernels are $\sqrt{2}$ times longer than the corresponding non-diagonal ones.

nal hull of $\left\{\operatorname{supp} g_{h^+(x,\theta_k),\theta_k}\right\}_{k=1}^{8}$. Such neighborhoods are shown in Figure 1.7. The polygonal hull is realized very efficiently as a combination of precalculated triangular binary stencils, each of which is associated to a pair of possible adaptive scales, as illustrated in Figure 1.8. Although the supports obtained in this way have relatively simple shapes when compared to the more general examples of Figure 1.2, we found that this is not a significant restriction. On the contrary, a more regular boundary of the transform's support is known to improve the efficiency of the SA-DCT [12].

We note that in this particular implementation the value of the adaptive-scale $h^+(x, \theta_k)$ coincides with the length (measured in pixels) of the directional window in the direction θ_k (i.e. with the length of the support of the corresponding directional kernel).

For the sake of notation clarity, we remind that the adaptive neighborhood of x used as support for the SA-DCT is $\tilde{U}_x^+$ (with tilde), which is obtained from the adaptive neighborhood U_x^+ (without tilde) by translation and mirroring, as defined in Section 1.3. In both symbols the subscript "x" denotes the point for which the adaptive scales are obtained while the "+" is used to distinguish the adaptive neighborhoods from the non-adaptive ones.

Orientation of the adaptive neighborhood and CR/RC processing

To determine the orientation of the adaptive neighborhood $\tilde{U}_x^+$ and hence decide whether to apply the SA-DCT in RC or CR mode, we use the four adaptive scales for the vertical and horizontal directions and test the following inequality:

$$h^+(x, \theta_1) + h^+(x, \theta_5) > h^+(x, \theta_3) + h^+(x, \theta_7). \tag{1.5}$$

If this inequality is verified, then the neighborhood is considered to be horizontally oriented and the SA-DCT is applied in RC mode. If the inequality (1.5) is not

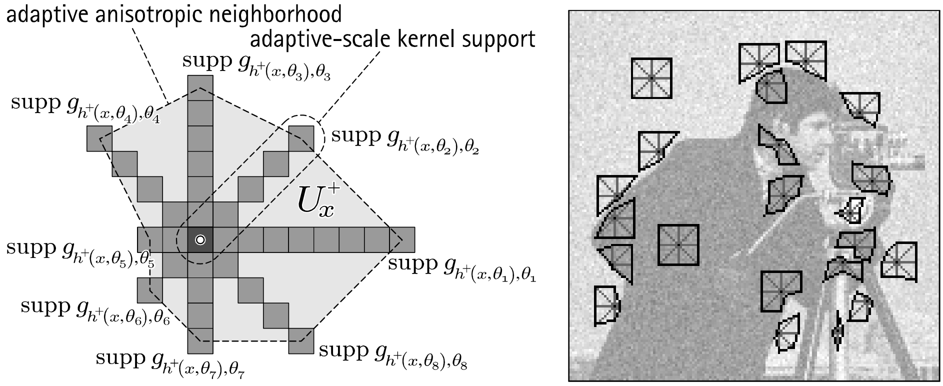


Figure 1.7: Fast implementation of the LPA-ICI anisotropic neighborhoods. "Linewise"one-dimensional directional LPA kernels are used for 8 directions. The anisotropic neighborhood U_x^+ is constructed as the polygonal hull of the adaptive-scale kernels' supports (left). Thus, only the adaptive scales h^+ are needed to construct the neighborhood. Some examples of the anisotropic neighborhoods $\tilde{U}_x^+$ used for SA-DCT filtering of the noisy *Cameraman* image (right), σ=25. In our implementation we use $h \in H = \{1, 2, 3, 5, 7, 9\}$.

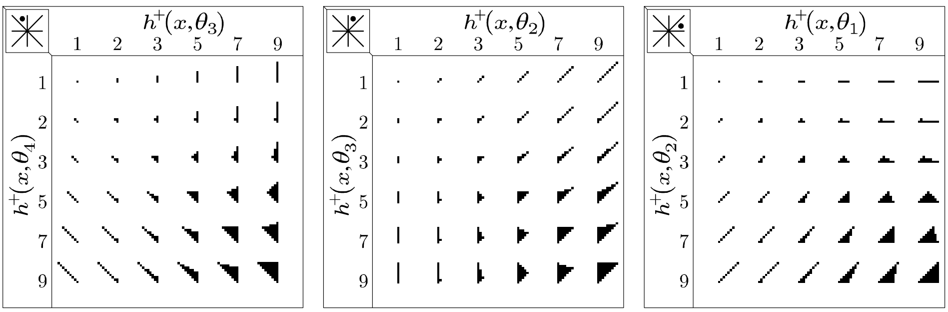


Figure 1.8: The polygonal hull shown in Figure 1.7 is realized combining precalculated triangular stencils, each of which is associated to a pair of possible adaptive scales $\left(h^+(x,\theta_k), h^+\left(x,\theta_k+\frac{\pi}{4}\right)\right)$. Here, we show three sets (look-up tables) of stencils corresponding to the pairs $\left(h^+(x,\theta_3), h^+(x,\theta_4)\right)$, $\left(h^+(x,\theta_2), h^+(x,\theta_3)\right)$, and $\left(h^+(x,\theta_1), h^+(x,\theta_2)\right)$.

verified, then the neighborhood is considered to be vertically oriented and the transform is applied in CR mode.

1.5.2 Local estimates

For every point $x \in X$, we construct a *local* estimate $\hat{y}_{\tilde{U}_x^+} : \tilde{U}_x^+ \to \mathbb{R}$ of the signal y by thresholding in SA-DCT domain as in (1.4),

$$\hat{y}_{\tilde{U}_x^+} = T_{\tilde{U}_x^+}^{-1}\left(\Upsilon_x\left(\varphi_{z,x}\right)\right) + m_{\tilde{U}_x^+}(z), \tag{1.6}$$

where the transform-domain coefficients $\varphi_{z,x} : V_{\tilde{U}_x^+} \to \mathbb{R}$ are calculated as

$$\varphi_{z,x} = T_{\tilde{U}_x^+}\left(z_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}(z)\right), \tag{1.7}$$

and Υ_x is a hard-thresholding operator with threshold

$$\gamma_{\text{thr}}\sigma\sqrt{2\ln|\tilde{U}_x^+| + 1}, \tag{1.8}$$

$\gamma_{\text{thr}} > 0$ being a fixed constant[3]. This threshold is essentially Donoho's "universal" threshold [36, 112] and it increases with the size of the neighborhood. In hard-thresholding, only the coefficients $\varphi_{z,x}$ whose amplitude $|\varphi_{z,x}|$ is larger than the threshold (1.8) are kept, all other smaller coefficients are discarded and replaced by zeros. The sparsity achieved thanks to the adaptive selection of the transform support ensures that most of the energy of the original signal is carried by only few noisy coefficients, which are kept after thresholding, and that the many discarded coefficients contain mostly noise. In what follows, we approximate[4] the variance $\text{var}\left\{\varphi_{z,x}(\cdot)\right\}$ of each transform coefficient as σ^2. An estimate of the total sample variance $\text{tsvar}\{\hat{y}_{\tilde{U}_x^+}\}$ of $\hat{y}_{\tilde{U}_x^+}$ is given as sum of variances of the transform coefficients which are used for reconstruction. It has the form

$$\text{tsvar}\{\hat{y}_{\tilde{U}_x^+}\} = \sigma^2\left(1 + N_x^{\text{har}}\right), \tag{1.9}$$

where N_x^{har} is the number of non-zero coefficients after thresholding (so-called "number of harmonics") and the unit addend accounts for the addition of the mean after the inversion of the transform.

Since the anisotropic neighborhoods corresponding to nearby points are usually overlapping, and since the SA-DCT is a complete system (basis) for an individual support $\tilde{U}_x^+$, the overall approach is obviously overcomplete.

1.5.3 Global estimate as aggregation of local estimates

In order to obtain a single *global* estimate $\hat{y} : X \to \mathbb{R}$ defined on the whole image domain, all the local estimates (1.6) are averaged together using adaptive weights $w_x \in \mathbb{R}$ in the following convex combination:

$$\hat{y} = \frac{\sum_{x\in X} w_x \hat{y}_{\tilde{U}_x^+}{}_{|X}}{\sum_{x\in X} w_x \chi_{\tilde{U}_x^+}}. \tag{1.10}$$

It is a standard approach [37] to use weights w_x that are inversely proportional to the average sample variance[5] of $\hat{y}_{\tilde{U}_x^+}$, $\text{asvar}\{\hat{y}_{\tilde{U}_x^+}\} = \text{tsvar}\{\hat{y}_{\tilde{U}_x^+}\}/|\tilde{U}_x^+|$. As

[3] The constant γ_{thr} is usually chosen to be smaller than 1; it is known that the resulting threshold would be otherwise too large for practical applications (e.g., [112], [86]). In particular, in all our denoising simulations we use $\gamma_{\text{thr}} = 0.77$.

[4] This is only an approximation because of the mean subtraction in (1.7), as discussed in Section 1.4.2. Otherwise, without mean subtraction, the approximation would be an exact equality (as observed in Section 1.4.1), since the noise in $z_{|\tilde{U}_x^+}$ is i.i.d. and the SA-DCT $T_{\tilde{U}_x^+}$ is orthonormal.

[5] Under much simplifying assumptions, such weights would correspond to a maximum-likelihood solution. To see this, let us consider local estimates $\hat{y}_{\tilde{U}_x^+}$, $x \in \Omega_{x_0} =$

shown in [75] for the case of sliding 8×8 block DCT denoising, such a simple weighting enables to attain the same performance achievable with much more involved models of the blocks' statistics.

However, this simple approach is inadequate when instead of fixed-size blocks one is considering adaptive regions with arbitrary shape and size. In particular, not only the size of the regions may vary, but also the number of overlapping shapes may be different for different points. If the inverse of the average variances are used as weights, it can be observed that when regions of significantly different sizes overlap (this may happen along edges or transitions), then the local estimates corresponding to larger regions will inevitably "submerge" the finer details restored by smaller regions.

Crucial compensation of these oversmoothing effects can be obtained by dividing the weights by the square of the size of the support, and we define w_x as

$$w_x = \frac{\mathrm{tsvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}^{-1}}{|\tilde{U}_x^+|} = \frac{\sigma^{-2}}{\left(1+N_x^{\mathrm{har}}\right)|\tilde{U}_x^+|}. \tag{1.11}$$

Let us observe that in areas where the size of the adaptive neighborhood is nearly constant (e.g., within smooth parts of the image) the weights (1.11) are inversely proportional to the average and to the total sample variances of the corresponding local estimates, $w_x \propto \mathrm{tsvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}^{-1}$. Thus, we can use the weights (1.11) for such areas also.

The weights w_x have this form because the total sample variance $\mathrm{tsvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}$ is obviously an upper bound for the pointwise residual-noise variance of the local estimate $\hat{y}_{\tilde{U}_x^+}$ (such pointwise variance is not necessarily uniform over $\tilde{U}_x^+$), while

---

$\left\{x \in X : x_0 \in \tilde{U}_x^+\right\}$, and assume that these estimates are unbiased and independent. Further, we assume that the pointwise sample variance of $\hat{y}_{\tilde{U}_x^+}(x_0)$ coincides with the average sample variance of $\hat{y}_{\tilde{U}_x^+}$ over $\tilde{U}_x^+$. Then, $\hat{y}_{\tilde{U}_x^+}(x_0) \sim \mathcal{N}\left(y(x_0), \mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}\right)$ and the log-likehood L is

$$\begin{aligned} L &= \ln \prod_{x\in\Omega_{x_0}} \left(2\pi\,\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}\right)^{-\frac{1}{2}} e^{-\frac{1}{2\,\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}}\left(\hat{y}_{\tilde{U}_x^+}(x_0)-y(x_0)\right)^2} = \\ &= -\frac{1}{2}\sum_{x\in\Omega_{x_0}} \frac{1}{\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}}\left(\hat{y}_{\tilde{U}_x^+}(x_0)-y(x_0)\right)^2 + \ln\left(2\pi\,\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}\right). \end{aligned}$$

By solving $\frac{\partial L}{\partial y} = 0$, we maximize L and obtain a maximum-likelihood estimate $\hat{y}^{\mathrm{ML}}(x_0)$ of $y(x_0)$ given the local estimates $\hat{y}_{\tilde{U}_x^+}$, $x \in \Omega_{x_0}$:

$$\begin{aligned} 0 &= \sum_{x\in\Omega_{x_0}} \frac{1}{\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}}\left(\hat{y}_{\tilde{U}_x^+}(x_0) - \hat{y}^{\mathrm{ML}}(x_0)\right), \\ \hat{y}^{\mathrm{ML}}(x_0)\sum_{x\in\Omega_{x_0}} \frac{1}{\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}} &= \sum_{x\in\Omega_{x_0}} \frac{1}{\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}}\hat{y}_{\tilde{U}_x^+}(x_0), \\ \hat{y}^{\mathrm{ML}}(x_0) &= \frac{\sum_{x\in\Omega_{x_0}} \frac{1}{\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}}\hat{y}_{\tilde{U}_x^+}(x_0)}{\sum_{x\in\Omega_{x_0}} \frac{1}{\mathrm{asvar}\left\{\hat{y}_{\tilde{U}_x^+}\right\}}}. \end{aligned}$$

The last expression is equivalent to (1.10), provided that the size $|\tilde{U}_x^+|$ is the same for all $x \in \Omega_{x_0}$.

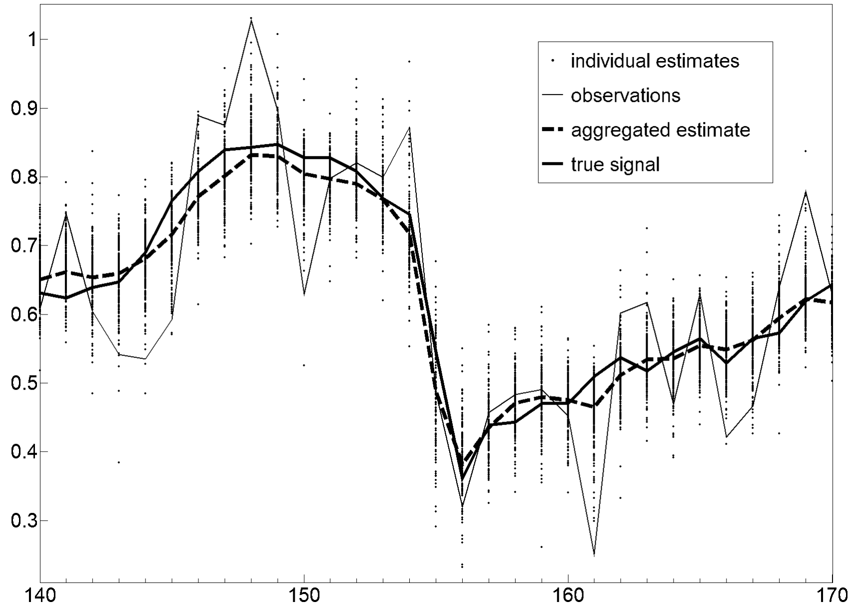


Figure 1.9: Aggregation in action. Details of a cross-section of length 31 pixels from the *Peppers* image (σ=25): the dots show all the individual estimates which are aggregated in order to obtained the final estimates at each position. For each pixel there are about 200 individual estimates.

the extra factor $|\tilde{U}_x^+|$ addresses the correlation that exists between overlapping neighborhoods (the number of overlapping neighborhoods is loosely proportional to their size). Qualitatively speaking, these weights favour estimates which correspond to sparser representations (fewer coefficients survived thresholding, and thus lower variance) and at the same time avoid that estimates with a small support (thus representing image details) are oversmoothed by other overlapping estimates which have a large support (which usually are strongly correlated among themselves and outnumber estimates of a smaller support).

The stabilizing effect of the aggregation (1.10) is clearly depicted in Figure 1.9, for a cross-section of the noisy *Peppers* image. One can observe that the individual estimates are extremely volatile, and that the majority of them is not a particularly good estimate of the true signal. Numerous outliers can be seen. Aggregation aims at combining all these estimates in order to produce an "aggregated estimate" which is not worse (in terms of estimation risk) than the best of them. However, as it can be seen in the figure, the estimates $\hat{y}_{\tilde{U}_x^+}$ are usually noisy and the averaging effect involved in the aggregation yields a remarkable improvement of the accuracy. In practice, one can observe that locally the aggregated estimate $\hat{y}$ is significantly better than any of the individual estimates $\hat{y}_{\tilde{U}_x^+}$ that were aggregated.

Interpretation as frames

Overall, we can interpret the collection of forward SA-DCT transforms on the overlapping adaptive domains as a redundant frame, whose dual frame is realized by the aggregation (1.10) (for a review of frame theory, see, e.g., [77], [43], [112]). For simplicity, in the following explanation, we ignore the DC separation and only consider the orthonormal SA-DCT transform applied on $z_{|\tilde{U}_x^+}$ as $T_{\tilde{U}_x^+}\left(z_{|\tilde{U}_x^+}\right)$. More precisely, for each $x \in X$, let us denote the SA-DCT basis functions defined on $\tilde{U}_x^+$ as $\psi_{\tilde{U}_x^+}^{(i)} : \tilde{U}_x^+ \to \mathbb{R}$, where $i \in V_{\tilde{U}_x^+}$. These functions are defined implicitly by the cascaded row-column structure of the SA-DCT and can be obtained explicitly by inverse transformation of the standard basis $\{e_i\}_{i_x \in V_{\tilde{U}_x^+}}$ as

$$\psi_{\tilde{U}_x^+}^{(i)} = T_{\tilde{U}_x^+}^{-1}\left(e_{i|V_{\tilde{U}_x^+}}\right), \qquad i \in V_{\tilde{U}_x^+}. \tag{1.12}$$

Thus, for each $x \in X$ we have a set $\mathcal{B}_x$ of functions supported on $\tilde{U}_x^+$,

$$\mathcal{B}_x = \left\{\psi_{\tilde{U}_x^+}^{(i)\ |X} : X \to \mathbb{R}, \ \ i \in V_{\tilde{U}_x^+}\right\}.$$

The set $\mathcal{B}_x$ is a basis for $\left\{f \in L^2(X) : f(x) = 0\ \forall x \notin \tilde{U}_x^+\right\}$. The collection $\mathcal{B}$ of all these functions

$$\mathcal{B} = \bigcup_{x \in X} \mathcal{B}_x = \left\{\psi^{(i)}\right\}_{i \in I}$$

is a frame for the function space $L^2(X)$. Here, all the functions of $\mathcal{B}$ are denoted by $\psi^{(i)}$, $i \in I$. For any function $f \in L^2(X)$, the standard analysis-synthesis equation for frames is

$$f = \sum\nolimits_i \langle f, \psi^{(i)} \rangle \breve{\psi}^{(i)}, \tag{1.13}$$

where $\breve{\mathcal{B}} = \{\breve{\psi}^{(i)}\}_i$ is a *dual frame* of $\mathcal{B}$. Because of redundancy (mainly due to the overlapping of the supports $\tilde{U}_x^+$, $x \in X$), the dual frame is not uniquely defined[6] and there exist infinitely many choices for $\breve{\mathcal{B}}$ such that (1.13) holds. In (1.10), we exploit a particular choice of dual frame, defined by the data-adaptive weights (1.11). The dual frame $\breve{\mathcal{B}}$ can be easily expressed, by decomposing it in a number of subcollections $\breve{\mathcal{B}}_x$ of functions supported on $\tilde{U}_x^+$,

$$\breve{\mathcal{B}} = \bigcup_{x \in X} \tilde{\mathcal{B}}_x \qquad \tilde{\mathcal{B}}_x = \left\{\breve{\psi}_{\tilde{U}_x^+}^{(i)} : X \to \mathbb{R}, \quad i \in V_{\tilde{U}_x^+}\right\};$$

then, each dual function $\breve{\psi}_{\tilde{U}_x^+}^{(i)}$ is defined, from (1.10), as

$$\breve{\psi}_{\tilde{U}_x^+}^{(i)} = \frac{w_x}{\sum_{x \in X} w_x \chi_{\tilde{U}_x^+}} \psi_{\tilde{U}_x^+}^{(i)\ |X}.$$

[6] The dual frame $\breve{\mathcal{B}}$ is uniquely defined when $\mathcal{B}$ is a Riesz basis (i.e., a non-redundant frame whose functions are all linearly independent).

In the general case, a dual frame $\{\breve{\psi}^{(i)}\}_i$ can be obtained by $\breve{\boldsymbol{\psi}} = \boldsymbol{\psi}\left(\boldsymbol{\psi}^T \boldsymbol{\psi}\right)^{\dagger}$, where $\boldsymbol{\psi}$ and $\breve{\boldsymbol{\psi}}$ are the *frame* and *dual-frame matrices*, whose columns contain the frame functions $\psi^{(i)}$ and $\breve{\psi}^{(i)}$, respectively, and † denotes the pseudo-inverse. The matrix $\boldsymbol{\psi}^T \boldsymbol{\psi} = \boldsymbol{\Psi}$ is the Gramian matrix, formed by the inner products of the frame elements one against each other.

We remark that, although $\breve{\mathcal{B}}$ is a dual frame of $\mathcal{B}$, in general, $\tilde{\mathcal{B}}_x$ are *not* dual bases of $\mathcal{B}_x$ (the dual basis of $\mathcal{B}_x$ is $\mathcal{B}_x$ itself, since it is orthonormal).

1.5.4 Wiener filtering in SA-DCT domain

Using the same approach as for thresholding, we introduce an empirical Wiener filter [69, 68] in the SA-DCT domain. It assumes that an estimate $\hat{y}$ of y is known. In practice, we can obtain this estimate using the above thresholding technique. For every $x \in X$, let $\varphi_{\hat{y},x} : V_{\tilde{U}_x^+} \to \mathbb{R}$ be the SA-DCT (on $\tilde{U}_x^+$) coefficients of $\hat{y}$ where the mean $m_{\tilde{U}_x^+}(z)$ of z is subtracted before applying the transform:

$$\varphi_{\hat{y},x} = T_{\tilde{U}_x^+}\left(\hat{y}_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}(z)\right). \tag{1.14}$$

The *local* Wiener estimate $\hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+}$ is defined as

$$\hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+} = T^{-1}_{\tilde{U}_x^+}\left(\omega_x \varphi_{z,x}\right) + \varpi_x m_{\tilde{U}_x^+}(z), \tag{1.15}$$

where the SA-DCT coefficients $\varphi_{z,x}$ of z are calculated as in (1.7), and $\omega_x \in \mathcal{V}_{\tilde{U}_x^+}$ and $\varpi_x \in \mathbb{R}$ are respectively the Wiener attenuation factors for $\varphi_{z,x}$ and for the subtracted mean value $m_{\tilde{U}_x^+}(z)$,

$$\omega_x = \frac{\varphi^2_{\hat{y},x}}{\varphi^2_{\hat{y},x} + \sigma^2}, \qquad \varpi_x = \frac{m^2_{\tilde{U}_x^+}(\hat{y})}{m^2_{\tilde{U}_x^+}(\hat{y}) + \sigma^2/|\tilde{U}_x^+|}. \tag{1.16}$$

The *global* estimate $\hat{y}^{\mathrm{wi}}$ can be obtained analogously as in (1.10), using the convex combination with the adaptive weights w_x^{wi}:

$$\hat{y}^{\mathrm{wi}} = \frac{\sum_{x\in X} w_x^{\mathrm{wi}} \hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+}{}^{|X}}{\sum_{x\in X} w_x^{\mathrm{wi}} \chi_{\tilde{U}_x^+}}, \qquad w_x^{\mathrm{wi}} = \frac{\sigma^{-2}}{(\varpi_x^2 + \sum_{V_{\tilde{U}_x^+}} \omega_x^2)|\tilde{U}_x^+|}. \tag{1.17}$$

Similarly to (1.11), the term $\sigma^2\left(\varpi_x^2 + \sum_{V_{\tilde{U}_x^+}} \omega_x^2\right)$ in the adaptive weights corresponds to an estimate of the total sample variance of $\hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+}$.

The Pointwise SA-DCT results which we present in this thesis correspond to the $\hat{y}^{\mathrm{wi}}$ estimate (1.17), obtained using the thresholding estimate $\hat{y}$ (1.10) as a reference for the calculation of the Wiener attenuation factors ω_x,ϖ_x (1.16).

Some denoising examples are shown in Figures 1.10, 1.11, and 1.12. In particular, in Figure 1.12, we show the improvement of the empirical Wiener estimate $\hat{y}^{\mathrm{wi}}$ (1.17) over the hard-thresholding estimate $\hat{y}$ (1.10). In the same figure, we also compare the empirical Wiener estimate $\hat{y}^{\mathrm{wi}}$ with an "oracle" Wiener estimate, obtained in the ideal case where the original image y (which is unknown in practice) is used in place of the estimate $\hat{y}$ (1.10) in the definition of the Wiener attenuation factors ω_x,ϖ_x (1.14-1.16).

Figure 1.10: A fragment of *Lena*. Clockwise, from top-left: original, noisy observation (σ=25, PSNR=20.18dB), BLS-GSM estimate [138] (PSNR=31.69dB), and the proposed Pointwise SA-DCT estimate (PSNR=31.66).

Figure 1.11: A fragment of *Cameraman*. Clockwise, from top-left: original, noisy observation (σ=25, PSNR=20.14dB), BLS-GSM estimate [138] (PSNR=28.35dB), and the proposed Pointwise SA-DCT estimate (PSNR=29.11dB).

1.6 Pointwise SA-DCT for deblocking and deringing of block-DCT compressed images

The scope of the proposed filtering method is not limited to denoising only, and in this section we extend the above Pointwise SA-DCT denoising algorithm into a high-quality image deringing and deblocking filter for B-DCT compressed images.

1.6.1 Motivation

The new wavelet-based JPEG-2000 [157] image compression standard solved many of the drawbacks of its predecessor JPEG [171], which relies on the 8×8 B-DCT. The use of a wavelet transform computed globally on the whole image, as opposed

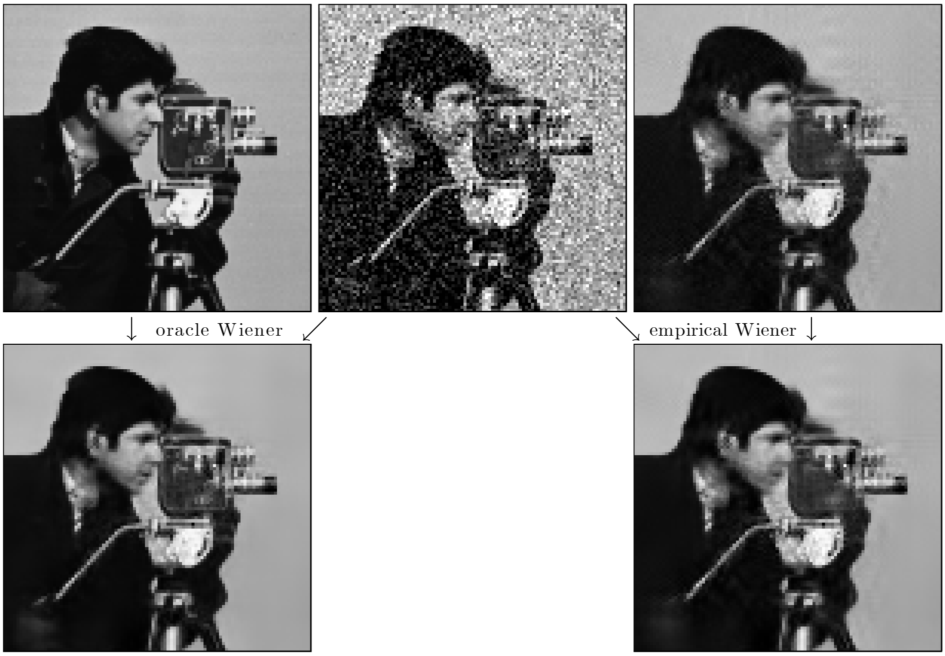


Figure 1.12: A fragment of *Cameraman*. Clockwise, from top-left: original y, noisy observation z (σ=35, PSNR=17.22dB), hard-thresholding estimate $\hat{y}$ (1.10) (PSNR=27.20dB), empirical Wiener estimate $\hat{y}^{\mathrm{wi}}$ (1.17) (PSNR=27.51dB), oracle Wiener estimate (PSNR=30.18dB). The empirical Wiener estimate is obtained filtering z using $\hat{y}$, as from (1.14-1.16); the oracle Wiener estimate represents the ideal case where the original image y (which is unknown in practice) is used in place of the estimate $\hat{y}$.

to the localized B-DCT, does not introduce any blocking artifacts and allows it to achieve a very good image quality even at high compression rates. Unfortunately, this new standard has received so far only very limited endorsement from digital camera manufacturers and software developers. As a matter of fact, the classic JPEG still dominates the consumer market and the near-totality of pictures circulated on the internet is compressed using this old standard. Moreover, the B-DCT is the workhorse on which even the latest MPEG video coding standards rely upon. There are no convincing indicators suggesting that the current trend is about to change any time soon. All these facts, together with the ever growing consumer demand for high-quality imaging, makes the development of advanced and efficient post-processing (deblocking, deringing, etc.) techniques still a very actual and relevant application area.

1.6.2 Modeling

While more sophisticated models of B-DCT-domain quantization noise have been proposed by many authors, we model this degradation as some additive noise. Thus, we use the observation model $z = y + \eta$ of Equation (1.1), where y is the original (non-compressed) image, z its observation after quantization in B-DCT

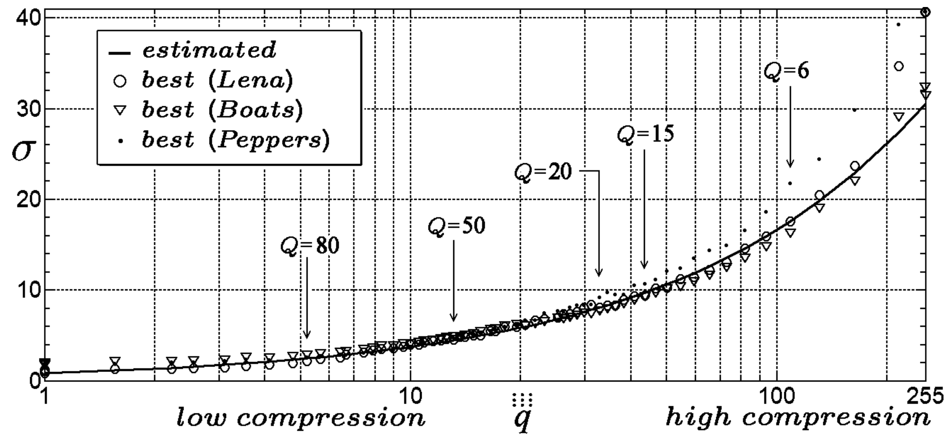


Figure 1.13: Agreement between the values of σ estimated by Equation (1.18) and the best ones (found experimentally), which give the highest PSNR for the filtered *Lena*, *Boats*, and *Peppers* images.

domain, and η is noise with variance σ^2. In order to apply the Pointwise SA-DCT filter we need a suitable value for the variance σ^2. We estimate it directly from the quantization table $\mathbf{Q} = [q_{i,j}]_{i,j=1,\dots,8}$ using the following empirical formula:

$$\sigma^2 = 0.69 \cdot \left(\overset{\vdots\vdots\vdots}{q}\right)^{1.3}, \qquad \overset{\vdots\vdots\vdots}{q} = \tfrac{1}{9} \sum\nolimits_{i,j=1}^{3} q_{i,j}. \tag{1.18}$$

This formula uses only the mean value $\overset{\vdots\vdots\vdots}{q}$ of the nine table entries which correspond to the lowest-frequency DCT harmonics (including the DC-term) and has been experimentally verified to be adequate for a wide range of different quantization tables and images. In Figure 1.13 we show how the values of σ calculated by Equation (1.18) agree with the best values found experimentally for the *Lena*, *Boats*, and *Peppers* images compressed with different quantization tables corresponding to JPEG with quality Q=1,...,100 (and thus $\overset{\vdots\vdots\vdots}{q}$=1,...,255). Note that a higher compression (e.g., JPEG with small Q) corresponds to a larger value for this variance (i.e. Q and $\overset{\vdots\vdots\vdots}{q}$ are inversely related). The standard-deviation σ is not linear with respect to the $q_{i,j}$'s, a fact which reflects the non-uniformity of the distribution of the B-DCT coefficients.

Note that the σ^2 which is calculated by (1.18) is *not* an estimate of the variance of compressed image, *nor* it is an estimate of the variance of the difference between original and compressed images. Instead, it is simply the assumed value for the variance of η in the observation model (1.1). Roughly speaking, it is the variance of some hypothetical noise which, if added to the original image y, would require – in order to be removed – the same level of adaptive smoothing which is necessary to suppress the artifacts generated by the B-DCT quantization with the table $\mathbf{Q}$. Much larger or much smaller values of σ^2 would respectively result in oversmoothing or leave the compression artifacts unfiltered.

Figures 1.14 and 1.15 show fragments of the JPEG-compressed grayscale *Cameraman* image obtained for two different compression levels (JPEG quality Q=6 and Q=15) and the corresponding Pointwise SA-DCT filtered estimates. For these

Figure 1.14: Details of the JPEG-compressed *Cameraman* (Q=6, bpp=0.19, PSNR=25.03dB) and of the corresponding Pointwise SA-DCT estimate (PSNR=26.11dB). The estimated standard deviation for this highly compressed image is σ=17.6.

Figure 1.15: Details of the JPEG-compressed *Cameraman* (Q=15, bpp=0.37, PSNR=27.71dB) and of the corresponding Pointwise SA-DCT estimate (PSNR=28.58dB). The estimated standard deviation for this compressed image is σ=9.7.

two cases the estimated standard-deviations are σ=17.6 and σ=9.7.

Let us observe that the procedure defined by (1.18) can be used in a straightforward manner, because the quantization tables are always (and necessarily) either provided with the coded data, or fixed in advance by the compression standard. It allows to apply the Pointwise SA-DCT denoising algorithm of Section 1.5 as an effective deblocking and deringing filter for B-DCT coded images and videos. The proposed method is particularly relevant for video postprocessing, since it can exploit the SA-DCT hardware of MPEG-4 decoders.

1.7 Pointwise SA-DCT filtering of color images with structural constraint in luminance-chrominance space

The extension from grayscale to color images of our denoising and deblocking approach is based on a very simple, yet powerful strategy. The key idea is the following: the structures (e.g., objects, edges, details, etc.) which determine the adaptive shapes are the same across all three color channels, thus the same shapes should be used for the SA-DCT filtering of the three channels. In order to increase its effectiveness, the method is implemented after transformation in a luminance-chrominance color-space. We call it *structural constraint in luminance-chrominance space* and it fully exploits the shape-adaptive nature of our approach without adding anything to its complexity.

1.7.1 Luminance-chrominance space

We generalize the observation model (1.1) to color data. Let $y = [y_R\ y_G\ y_B]$ be the original color image in the RGB color space. We consider noisy observations $z = [z_R\ z_G\ z_B]$ of the form

$$z_C = y_C + \eta_C, \qquad C = R, G, B, \tag{1.19}$$

where the noise $\eta = [\eta_R\ \eta_G\ \eta_B]$ is independent Gaussian, $\eta_C(\cdot) \sim \mathcal{N}\left(0, \sigma_C^2\right)$, $C = R, G, B$.

In order to deal with color images, we first perform a color-space transformation, aiming at reducing the strong correlation between channels which is typical of the RGB space. In particular, we consider the "*opponent*" and the $YUV/YCbCr$ color spaces [137]. Up to some normalization, the transformation to these color spaces can be expressed by multiplication of a column vector with the R, G, and B components against one of the matrices

$$\mathbf{A}_{opp} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{\sqrt{6}} & 0 & \frac{-1}{\sqrt{6}} \\ \frac{1}{3\sqrt{2}} & \frac{-\sqrt{2}}{3} & \frac{1}{3\sqrt{2}} \end{bmatrix}, \qquad \mathbf{A}_{yuv} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ -0.17 & -0.33 & 0.50 \\ 0.50 & -0.42 & -0.08 \end{bmatrix}.$$

Although purists may consider it an abuse of terminology, we call "luminance" and "chrominances" not only the components the YUV space, but also those of the opponent color space. We denote the luminance channel as Y, and the chrominances as U and V.

In such luminance-chrominance decompositions, the original inter-channel correlation of the RGB space is captured into the luminance channel, which thus enjoys a better signal-to-noise ratio (SNR), whereas the chrominance channels contain the differential information among the RGB channels.

We then come to the following observation model in luminance-chrominance space,

$$z_C = y_C + \eta_C, \qquad C = Y, U, V, \tag{1.20}$$

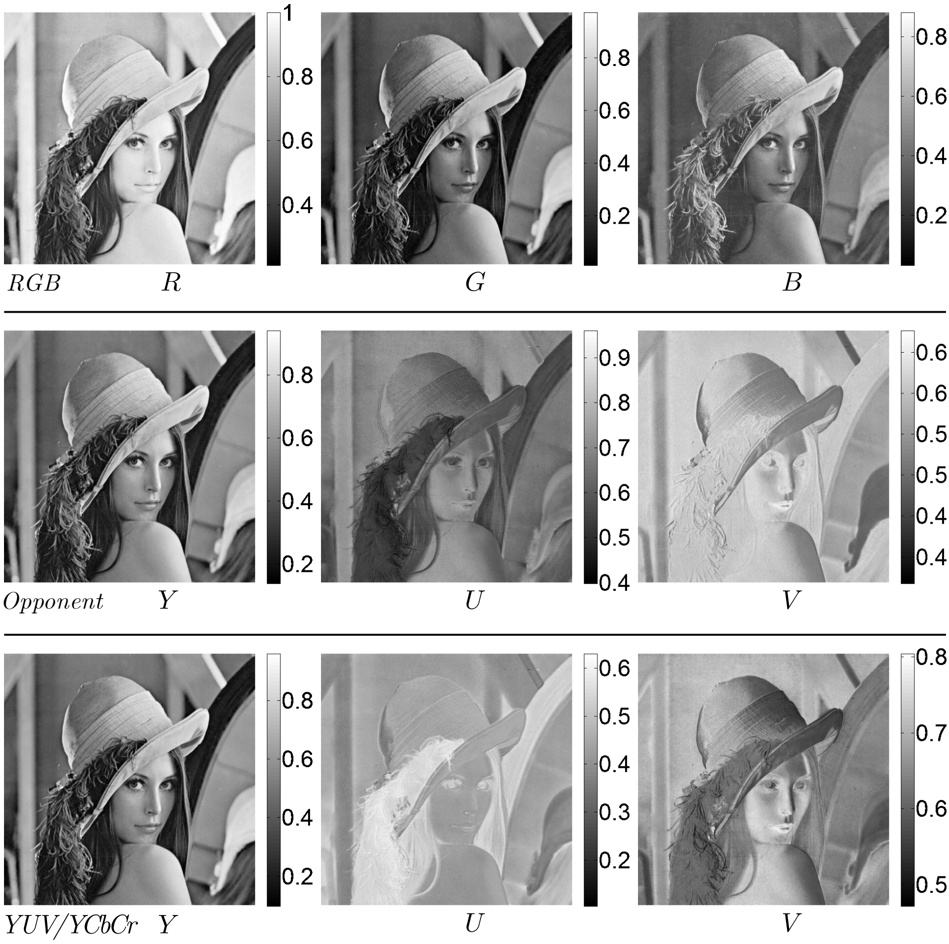


Figure 1.16: Illustration of the structural correlation among the different color channels. R, G, and B channels of the original color *Lena* image, and the luminance and chrominance channels of the same image for the opponent and $YUV/YCbCr$ color transformations. It can be clearly seen that discontinuities and sharp transitions appear in all the nine channels at the same spatial locations.

where $[z_Y \ z_U \ z_V] = [z_R \ z_G \ z_B]\mathbf{A}^T$, $[y_Y \ y_U \ y_V] = [y_R \ y_G \ y_B]\mathbf{A}^T$, and $\eta_C(\cdot) \sim \mathcal{N}(0, \sigma_C^2)$, $C = Y, U, V$.

Ideally, the Y, U, and V channels are considered as independent. Therefore, the common approach for color denoising in luminance-chrominance space is to filter the three channels (i.e. z_Y, z_U, and z_V) separately and independently one from the other.

However, when considering natural images, the different color channels always share some common features which are inherited from the structures and from the objects depicted in the original image. In particular, it can be observed that along the objects' boundaries all color channels usually exhibit some simultaneous discontinuities or sharp transitions. Figure 1.16 illustrates this fact for the *Lena* color image, showing nine different channels for the *RGB*, opponent, and *YUV*/*YCbCr* color spaces.

We exploit this kind of structural correlation by imposing that the three transform's supports which are used for the filtering of z_Y, z_U, and z_V at a particular location have the same adaptive shape. In practice, we use for all three channels the adaptive neighborhoods defined by the Anisotropic LPA-ICI for the luminance channel.

Such a constraint has the effect that whenever some structure is detected (in the luminance channel by the LPA-ICI), it is taken into account (and thus preserved) for the filtering of all three channels. Restricted to the adaptive supports, however, the channels are assumed as independent, and thus the transform-domain hard-thresholding and Wiener filtering are still performed for each channel independently from the others.

After the filtering of the three channels, inverse color-transformation returns the estimate of the original image y in the *RGB* space.

1.7.2 Pointwise SA-DCT denoising in luminance-chrominance space

The noise variances for the Y, U, and V channels can be calculated as the elements of the vector $[\sigma_Y^2\ \sigma_U^2\ \sigma_V^2] = [\sigma_R^2\ \sigma_G^2\ \sigma_B^2]\mathbf{A}^{T2}$, where σ_R^2, σ_G^2, and σ_B^2 are the noise variances for the R, G, and B channels and $\mathbf{A}^{T2}$ is the transposed color transformation matrix with all elements squared. For denoising, the opponent color transformation is preferable because of the orthogonality of the rows of $\mathbf{A}_{opp}$.

The better SNR of the luminance and its higher "information content" are the two main reasons why it is in this channel that we look for structures. There are also other reasons. In natural images it often happens that uniformly colored objects present luminance variations due to non-uniform illumination or shadowing: such transitions cannot be detected from the chrominances. On the other hand, it is quite rare that abrupt changes appear in the chrominances and not in the luminance. Therefore, it is sufficient to perform the LPA-ICI adaptive-scale selection on the luminance channel only.

An example of Pointwise SA-DCT denoising with structural constraint in the opponent color space is shown in Figure 1.17.

1.7.3 Deblocking and deringing of B-DCT compressed color images

The proposed strategy for color image filtering is also particularly effective for deblocking and deringing color images.

When compressing color images or video, the standard approach (e.g., in the JPEG and MPEG), is to first perform the *YUV* color transformation and then compress the resulting three channels separately. According to the modeling in the previous sections, we assume that the original (non-compressed) image y in the *RGB* color space is represented, after B-DCT quantization in *YUV* space, as the z_C in the observation model (1.20), where y_Y, y_U and y_V are the luminance

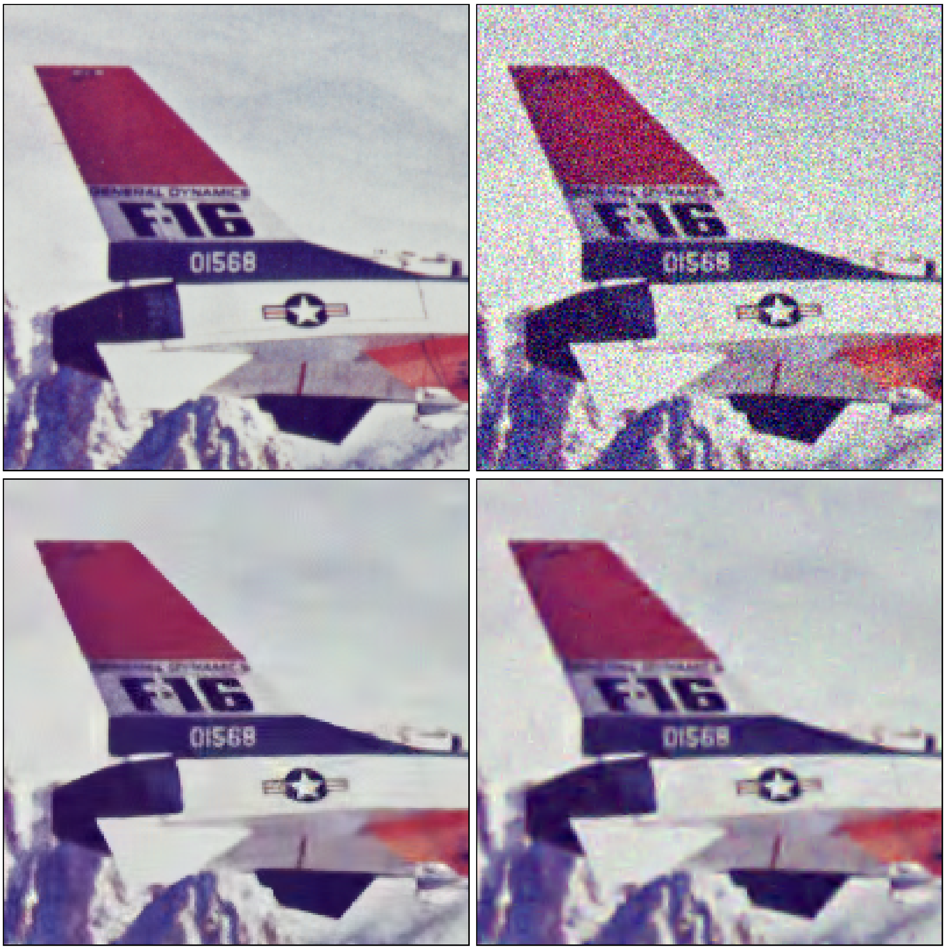


Figure 1.17: Fragments of the original *F-16* image (top-left), of its noisy observation (σ=30, PSNR=18.59dB) (top-right) and of two denoised estimates: our Pointwise SA-DCT estimate (bottom-left), and the ProbShrink-MB [136] estimate (bottom-right). The PSNR for the two estimates is 31.59dB and 30.50dB, respectively.

and chrominance channels of y, and z_Y, z_U and z_V are the corresponding channels after quantization in B-DCT domain.

We estimate the variances σ_Y^2, σ_U^2, and σ_V^2 of η_C, $C = Y, U, V$, from the corresponding quantization tables for the luminance and chrominance channels, using formula (1.18). However, if (as it is commonly done) the chrominance channels are downsampled, then the estimated variances for the chrominances need to be further multiplied by 2, in order to account for the coarser sampling.

Usually, the quantization tables $\mathbf{Q}^U$ and $\mathbf{Q}^V$ used for the two chrominances coincide, $\mathbf{Q}^U = \mathbf{Q}^V = \mathbf{Q}^{UV}$. Following standard models of the human visual system, a higher compression is typically performed on the chrominances than on the luminance. Thus, it is typical that the estimated variances are such that $2\sigma_Y^2 < \sigma_U^2 = \sigma_V^2$. Even at relatively high bit-rates, the compression of the chrominance

Figure 1.18: Fragments of the JPEG-compressed (Q=10, 0.25bpp, PSNR=26.87dB), and restored *F-16* color image (PSNR=28.30dB) using the proposed Pointwise SA-DCT deblocking filter in luminance-chrominance space.

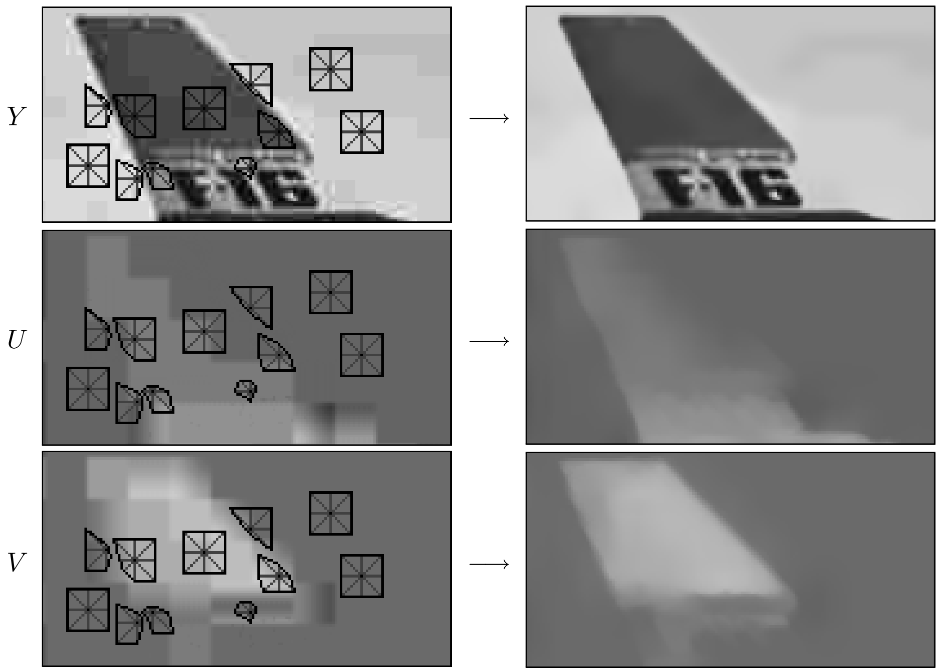


Figure 1.19: The adaptive anisotropic neighborhoods are selected by the LPA-ICI on the luminance channel (left-top). Observe that the neighborhoods are not affected by the blocking artifacts and yet are quite accurate with respect to the image features. These neighborhoods are used for SA-DCT filtering of the luminance as well as of the two chrominances (left-middle and left-bottom). The result of such filtering is shown in the right column. The color estimate obtained after inverse YUV color-transformation is shown in Figure 1.18.

channels can be quite aggressive.

As for color image denoising, we approach color data in a channel-by-channel manner imposing a unique structural constraint among the three channels. This allows to filter the chrominance channels restoring the structural information which was lost due to quantization and coarse sampling. The peculiarity of our approach is easily explained and demonstrated through the next example.

Figures 1.18 and 1.19 present a very common scenario. It can be seen that only very few AC-terms of the chrominance blocks survive to quantization, and the resulting chrominance channels end up with the vast majority of blocks represented by the DC-term only. It results in unpleasant color-bleeding artifacts along edges between differently colored objects. At the same time, on smoother areas the uneven hue due to quantization becomes particularly noticeable. In this example, the values of σ_Y and $\sigma_U = \sigma_V$ calculated according to formula (1.18) are 12.6 and 27.1, respectively.

As shown in Figure 1.19(left), we use for all three channels the adaptive neighborhoods defined by the Anisotropic LPA-ICI for the Y channel, because it is in the luminance that the structural information is usually better preserved after compression.

Figure 1.19(right) shows that the proposed method effectively attenuates ringing and blocking artifacts, faithfully preserving the structures and the salient feature in the image. Moreover, it demonstrates its ability of *reconstructing the missing structural information* in the chrominance channels, where the details of the tail of the plane are clearly revealed, with precise boundaries. The obtained color estimate, shown in Figure 1.18(right), is then quite sharp, with well-defined edges, and the color-bleeding artifacts (clearly visible in the JPEG-compressed image) are accurately corrected.

1.8 Experiments and results

We conclude the chapter with a number of experimental results and comparisons which demonstrate the state-of-the-art performance of the developed algorithms. These experiments have been produced and can be replicated using the Matlab implementation of the Pointwise SA-DCT algorithms, publicly available online at `http://www.cs.tut.fi/~foi/SA-DCT/` . We refer the interested reader to this software for additional details about the implementation and its specific parameters.

1.8.1 Grayscale denoising

Let us start with filtering of grayscale images corrupted by additive Gaussian white noise. In Table 1.1 we compare our results against those reported for other methods. In terms of PSNR, the results of the Pointwise SA-DCT estimates are high, often outperforming all other methods by other authors[7]. Additional results are given in Table 1.2 for more images and levels of noise. PSNR and MSE results which are consistent with those in Table 1.1 are reported in the DenoiseLab package

[7] The Block-Matching 3-D (BM3D) filtering [32] is a newer technique of which the author of this thesis is a co-author. At the time of writing and to the best of the authors' knowledge, the BM3D appears as the best performing image denoising algorithm in the open literature.

	Lena 512×512			*Boats* 512×512		
Method σ	15	20	25	15	20	25
BM3D (Dabov et al.) [32]	**34.27**	**33.05**	**32.08**	**32.14**	**30.88**	**29.91**
Pointwise SA-DCT	33.86	32.62	31.66	31.79	30.49	29.47
BLS-GSM (Portilla et al.) [138]	33.90	32.66	31.69	31.70	30.38	29.37
K-SVD (Elad & Aharon) [38]	33.70	32.38	31.32	31.73	30.36	29.28
Patch-based (Kervrann & Boulanger) [98]	33.70	32.64	31.73	31.44	30.12	29.20
TLS (Hirakawa & Parks) [78]	33.97	32.69	31.69	31.59	30.25	29.21
MGGD (Cho & Bui) [20]	33.70	32.46	31.48	31.46	30.14	29.12
Local bi-shrink, CWT (Şendur&Selesnick) [151]	33.64	32.37	31.37	31.33	30.05	29.03
SI-adaptShr, SI Symm. (Chang et al.) [16]	33.37	32.09	31.07	–	–	–
Recursive Anis. LPA-ICI (Foi et al.) [55, 62]	32.72	31.44	30.43	30.87	29.58	28.58
	Peppers 256×256			*House* 256×256		
σ	15	20	25	15	20	25
BM3D (Dabov et al.) [32]	**32.70**	**31.29**	**30.16**	**34.94**	**33.77**	**32.86**
Pointwise SA-DCT	32.44	31.04	29.92	34.14	32.92	31.92
BLS-GSM (Portilla et al.) [138]	31.74	30.31	29.21	33.63	32.39	31.40
K-SVD (Elad & Aharon) [38]	32.22	30.82	29.73	34.32	33.20	32.15
Patch-based (Kervrann & Boulanger) [98]	32.13	30.59	29.73	34.08	32.90	32.22
TLS (Hirakawa & Parks) [78]	31.61	30.21	29.06	33.82	32.58	31.60
Recursive Anis. LPA-ICI (Foi et al.) [55, 62]	31.78	30.30	29.16	33.18	31.82	30.73

Table 1.1: PSNR (dB) comparison table for the denoising of the grayscale *Lena*, *Boats*, *Peppers*, and *House* test images with different levels of Gaussian noise. Results for *Peppers* and *House* were not reported in [20], [151], and [16]. The results for *Boats* were not reported in [16].

Figure 1.20: Fragments of the Pointwise SA-DCT estimates of the *Boats*, *Peppers*, and *House* images, σ=25.

[104], where the Pointwise SA-DCT is among the benchmark algorithms used for evaluating other denoising algorithms over a standard image database.

We emphasize the outstanding preservation of sharp details which we demonstrate in Figures 1.10, 1.11, 1.12, and 1.20, while almost no visible artifacts are present. Other transform-based estimates, such as those from [138], often display noticeable overshooting on the edges and unpleasant spurious oscillations. These artifacts, which are characteristic of all oscillatory transforms (including

σ	*Lena* 512x512	*Boats* 512x512	*House* 256x256	*Peppers* 256x256	*C'man* 256x256	*Barbara* 512x512	*Hill* 512x512	*Lake* 512x512
5	38.54	37.14	39.38	37.99	38.15	37.47	37.03	36.61
10	35.58	33.63	35.98	34.46	33.98	33.48	33.43	32.81
15	33.86	31.79	34.14	32.44	31.70	31.37	31.60	30.98
20	32.62	30.49	32.92	31.04	30.18	29.99	30.39	29.74
25	31.66	29.47	31.92	29.92	29.11	28.94	29.50	28.79
30	30.86	28.64	31.10	29.03	28.24	28.09	28.80	28.01
35	30.17	27.93	30.39	28.26	27.51	27.35	28.22	27.35
50	28.60	26.26	28.67	26.55	25.88	25.44	26.85	25.83
75	26.76	24.62	26.58	24.57	23.89	23.51	25.44	24.14
100	25.50	23.52	25.07	23.15	22.49	22.49	24.47	22.96

Table 1.2: Grayscale image denoising performance as PSNR (dB) for the proposed Pointwise SA-DCT algorithm.

σ	*Lena*	*Peppers*	*Baboon*	*House*	*F-16*	*Lake*
10	34.95	33.70	30.62	35.67	36.41	32.34
15	33.58	32.42	28.33	34.09	34.67	30.52
20	32.61	31.57	26.89	32.97	33.41	29.40
25	31.85	30.90	25.86	32.12	32.42	28.58
30	31.21	30.33	25.07	31.39	31.59	27.93
35	30.65	29.81	24.44	30.74	30.88	27.38
50	29.27	28.53	23.03	29.13	29.19	26.10
75	27.77	27.07	21.46	27.39	27.43	24.68

Table 1.3: Color image denoising performance as PSNR (dB) for the proposed Pointwise SA-DCT algorithm.

the SA-DCT), do not appear in our estimates thanks to the adaptive selection of the transform support. Further comments and results on grayscale denoising are given in Section 1.8.4, where we concentrate on the perceptual quality of the Pointwise SA-DCT estimates.

1.8.2 Color denoising

For the color denoising experiments, the variance of the additive Gaussian noise is set to be the same for all *RGB* color channels, $\sigma_R^2 = \sigma_G^2 = \sigma_B^2 = \sigma^2$. Filtering is performed after transformation to the opponent color space. Table 1.3 gives the PSNR results for the denoising of the *Lena*, *Peppers*, *Baboon*, *House*, *F-16*, and *Lake* color test-images over a wide range of values of σ.

In Table 1.4 we compare our results against those by other state-of-the-art methods, as reported in [136] and [33]. In particular, the vector-based minimum-mean-squared-error estimator (VMMSE) [147], the multiband wavelet thresholding (MBT) [146], the ProbShrink-multiband wavelet algorithm [136], and the Color Block-Matching 3-D filtering algorithm (C-BM3D) [33] (see footnote 7 on page 29) are considered for comparison.

Method σ	*Color Lena* 512×512			
	10	15	20	25
C-BM3D (Dabov et al.) [33]	**35.22**	**33.94**	**33.02**	**32.27**
Pointwise SA-DCT	34.95	33.58	32.61	31.85
ProbShrink-MB (Pizurica et al.) [136]	34.60	33.03	31.92	31.04
VMMSE (Scheunders and Driesen) [147]	34.02	31.89	30.24	28.88
MBT (Scheunders) [146]	33.84	32.29	31.14	30.15
Method σ	***Color Peppers* 512×512**			
	10	15	20	25
C-BM3D (Dabov et al.) [33]	**33.78**	**32.60**	**31.83**	**31.20**
Pointwise SA-DCT	33.70	32.42	31.57	30.90
ProbShrink-MB (Pizurica et al.) [136]	33.44	32.05	31.12	30.35
VMMSE (Scheunders and Driesen) [147]	33.12	31.13	29.67	28.45
MBT (Scheunders) [146]	31.19	30.22	29.45	28.77
Method σ	***Color Baboon* 512×512**			
	10	15	20	25
C-BM3D (Dabov et al.) [33]	30.64	**28.39**	**26.97**	**25.95**
Pointwise SA-DCT	30.62	28.33	26.89	25.86
ProbShrink-MB (Pizurica et al.) [136]	30.17	27.83	26.38	25.27
VMMSE (Scheunders and Driesen) [147]	**30.68**	28.24	26.63	25.36
MBT (Scheunders) [146]	28.50	26.78	25.53	24.56

Table 1.4: PSNR (dB) comparison table for the denoising of the *Lena*, *Peppers*, and *Baboon* color test images with different levels of Gaussian noise.

Let us note that, with the exception of the C-BM3D, the reference methods which are included in the Table 1.4 are multiband or vector methods, which are specifically designed for the denoising of color or multispectral images. Such algorithms simultaneously filter all channels, exploiting the possible inter-channel correlation, and are thus inherently superior to the simpler strategy where a scalar (grayscale) denoising filter is used independently for each separate channel. The C-BM3D adopts a structural constraint in luminance-chrominance space, mimicking the approach originally proposed for the Pointwise SA-DCT.

We remark that although in our approach the adaptive supports for the SA-DCT at a particular location are the same for all three channels, the SA-DCT-domain filtering is performed for each channel independently. Nevertheless, our results are very competitive and the comparison table shows that in fact the proposed technique outperforms all other reference methods by other authors.

Similarly to the grayscale case, the denoised color estimates produced by our adaptive algorithm are visually very good. A close inspection to Figures 1.17 and 1.21 may reveal the outstanding preservation of sharp details achieved by the shape-adaptive transform. At the same time, almost no visible artifacts (such as blurriness or overshooting) are present. The PSNR difference (0.1 – 0.4 dB) of the C-BM3D algorithm over the Pointwise SA-DCT is consistent with the difference between the grayscale versions of these two methods.

Figure 1.21: Fragments from the noisy (σ=25, PSNR=20.18dB) and denoised color *Peppers* image (PSNR=30.90dB), obtained using the proposed Pointwise SA-DCT algorithm.

$$\mathbf{Q}1 = \begin{bmatrix} 50 & 60 & 70 & 70 & 90 & 120 & 255 & 255 \\ 60 & 60 & 70 & 96 & 130 & 255 & 255 & 255 \\ 70 & 70 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 96 & 120 & 145 & 255 & 255 & 255 & 255 \\ 90 & 130 & 200 & 255 & 255 & 255 & 255 & 255 \\ 120 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

$$\mathbf{Q}2 = \begin{bmatrix} 86 & 59 & 54 & 86 & 129 & 216 & 255 & 255 \\ 64 & 64 & 75 & 102 & 140 & 255 & 255 & 255 \\ 75 & 70 & 86 & 129 & 216 & 255 & 255 & 255 \\ 75 & 91 & 118 & 156 & 255 & 255 & 255 & 255 \\ 97 & 118 & 199 & 255 & 255 & 255 & 255 & 255 \\ 129 & 189 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

$$\mathbf{Q}3 = \begin{bmatrix} 110 & 130 & 150 & 192 & 255 & 255 & 255 & 255 \\ 130 & 150 & 192 & 255 & 255 & 255 & 255 & 255 \\ 150 & 192 & 255 & 255 & 255 & 255 & 255 & 255 \\ 192 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

Table 1.5: The three B-DCT quantization tables **Q**1, **Q**2, **Q**3 [109].

1.8.3 Deblocking and deringing

In order to assess the filtering performance of the proposed deblocking and deringing method, extensive simulation experiments were performed for different types of quantization tables, several level of compression, and for grayscale as well as for color images. We reproduce the same experimental settings used by other authors and we present comparative numerical results collected in three separate tables. The first two tables contain results for grayscale images obtained using three particular quantization tables found in the literature (Table 1.6) and using the standard JPEG (Table 1.7). The third and last table is dedicated to experiments with JPEG compression of color images.

Three quantization tables, usually denoted as **Q**1, **Q**2, and **Q**3, have been used by many authors (e.g., [109] and references therein) in order to simulate various types of B-DCT compression. These three quantization tables are shown Table 1.5. The values of the standard deviation σ corresponding to these three tables – calculated using formula (1.18) – are 12.62, 13.21, and 22.73, respectively. In terms of image degradation, they correspond to a medium to high compression level, similar to what can be obtained by using JPEG with quality Q=11, Q=9, and Q=5, respectively.

Table	*Lena* 512×512			*Peppers* 512×512			*Barbara* 512×512		
	image	others	*P. SA-DCT*	image	others	*P. SA-DCT*	image	others	*P. SA-DCT*
Q1	30.70	31.63	**32.12**	30.42	31.33	**32.02**	25.94	26.64	**26.79**
Q2	30.09	31.19	**31.56**	29.82	30.97	**31.45**	25.59	26.32	**26.45**
Q3	27.38	28.65	**29.03**	27.22	28.55	**29.13**	24.03	24.73	**25.13**

Table 1.6: PSNR (dB) comparison table for restoration from B-DCT quantization for three different quantization matrices. The values under "others" correspond to the best results of any of the methods [109, 79, 116, 127, 179, 177], as reported in [109].

Qual.	*Lena* 512×512			
	JPEG	(bpp)	others	*P. SA-DCT*
4	26.46	(0.11)	27.63	**28.08**
6	28.24	(0.15)	29.22	**29.87**
8	29.47	(0.18)	30.37	**30.99**
10	30.41	(0.22)	31.17	**31.84**
12	31.09	(0.25)	31.79	**32.48**
Qual.	*Barbara* 512×512			
	JPEG	(bpp)	others	*P. SA-DCT*
4	23.48	(0.14)	24.13	**24.65**
6	24.50	(0.18)	25.08	**25.51**
8	25.19	(0.23)	25.71	**26.11**
10	25.79	(0.28)	26.27	**26.61**
12	26.33	(0.32)	26.81	**27.10**
Qual.	"*Green Peppers*"[8] 512×512			
	JPEG	(bpp)	others	*P. SA-DCT*
4	25.61	(0.14)	26.72	**27.41**
6	27.32	(0.18)	28.22	**28.97**
8	28.40	(0.22)	29.28	**29.90**
10	29.16	(0.25)	29.94	**30.51**
12	29.78	(0.28)	30.47	**31.00**

Table 1.7: PSNR (dB) comparison table for restoration from JPEG compression of grayscale images. The values under "others" correspond to the best result obtained by any of the methods [6, 19, 144, 178, 114, 116], as reported in [6].

In Table 1.6 we present results for deblocking from B-DCT quantization performed using these specific quantization tables. We compare the results obtained by our SA-DCT algorithm against the best results obtained by any of the methods [109, 79, 116, 127, 179, 177], as reported in [109]. The results are in favor of our proposed technique, which consistently outperforms all other methods.

Further positive results are shown in Table 1.7 for the case of deblocking from JPEG-compression. In this second table we compare against the best result obtained by any of the methods [6, 19, 144, 178, 114, 116], as reported in [6]. Also in this comparison, the SA-DCT method is found to be superior to all other tech-

[8] In order to replicate the experiments as in [6], the "*Peppers*" image used for Table 1.7 is the green channel of the *RGB* color *Peppers*. Let us note, however, that far more often in the literature the grayscale *Peppers* are found as the luminance channel Y of the *RGB Peppers* image.

	Color Lena 512×512			*Color Peppers* 512×512		
Qual.	JPEG (bpp)	*P. SA-DCT*	ISNR	JPEG (bpp)	*P. SA-DCT*	ISNR
4	23.34 (0.12)	24.79	1.45	22.32 (0.13)	23.77	1.46
6	25.52 (0.16)	27.09	1.57	23.99 (0.17)	25.54	1.54
8	26.64 (0.19)	28.16	1.52	24.99 (0.21)	26.40	1.41
10	27.53 (0.23)	29.06	1.53	25.77 (0.25)	27.11	1.34
15	28.97 (0.31)	30.33	1.35	26.88 (0.33)	27.99	1.11
20	29.83 (0.38)	31.00	1.17	27.57 (0.40)	28.53	0.96
25	30.44 (0.44)	31.46	1.02	28.04 (0.47)	28.90	0.86
30	30.91 (0.50)	31.79	0.88	28.40 (0.54)	29.14	0.74
40	31.54 (0.61)	32.26	0.72	28.83 (0.66)	29.45	0.62
50	32.02 (0.72)	32.63	0.61	29.25 (0.78)	29.81	0.56
75	33.21 (1.13)	33.56	0.35	30.29 (1.23)	30.67	0.52
	Color Baboon 512×512			*Color House* 256×256		
Qual.	JPEG (bpp)	*P. SA-DCT*	ISNR	JPEG (bpp)	*P. SA-DCT*	ISNR
4	19.28 (0.17)	20.00	0.72	22.63 (0.15)	23.76	1.13
6	20.38 (0.26)	21.05	0.67	24.41 (0.19)	25.66	1.24
8	21.12 (0.35)	21.71	0.59	25.16 (0.24)	26.41	1.25
10	21.63 (0.43)	22.13	0.50	26.25 (0.27)	27.54	1.29
15	22.49 (0.62)	22.88	0.38	27.52 (0.34)	28.66	1.14
20	23.07 (0.77)	23.37	0.31	27.87 (0.41)	28.75	0.88
25	23.50 (0.92)	23.75	0.25	28.55 (0.47)	29.44	0.89
30	23.85 (1.05)	24.06	0.21	28.96 (0.54)	29.76	0.80
40	24.40 (1.29)	24.56	0.16	29.51 (0.65)	30.20	0.69
50	24.85 (1.51)	24.97	0.12	29.80 (0.76)	30.40	0.60
75	26.21 (2.33)	26.25	0.04	31.44 (1.18)	32.00	0.56

Table 1.8: PSNR (dB) results for the Pointwise SA-DCT filtering of JPEG-compressed color images. Results are given also in terms of improvement-in-SNR (ISNR, dB).

niques, outperforming them of about 0.5 dB in all experiments.

In Table 1.8 we show results for the SA-DCT filtering of JPEG-compressed color images, from very high (Q=4) to very low (Q=75) compression levels. It can be seen that the improvement is significant especially for very high and moderate compression levels. For very low compression levels (for which the compression artifacts are barely visible and thus there is typically no need for postprocessing) the improvement is still substantial for those images which present some structures or edges.

For the simulations in Table 1.7 and Table 1.8 as well as for all JPEG experiments presented in this thesis, we use the baseline IJG JPEG implementation [83]. Table 1.9 gives the standard JPEG quantization tables for the luminance and chrominance channels for a JPEG-quality parameter Q=50. The corresponding estimated standard-deviations according to (1.18) are $\sigma_Y = 4.4$ and $\sigma_U = \sigma_V = 9.7$.

We conclude with two examples which highlight the very special reconstruction ability enabled by our structural constraint in luminance-chrominance space.

Figure 1.22(left) shows a fragment of the JPEG compressed Lena image (Q=20, 0.38 bpp, PSNR=29.83 dB). The corresponding U and V chrominance channels are shown in Figure 1.23. One can barely recognize the salient features of the

$$
\mathbf{Q}^{Y}_{Q=50} = \begin{bmatrix}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\
72 & 92 & 95 & 98 & 112 & 100 & 103 & 99
\end{bmatrix}
$$

$$
\mathbf{Q}^{UV}_{Q=50} = \begin{bmatrix}
17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\
18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\
24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\
47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\
99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\
99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\
99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\
99 & 99 & 99 & 99 & 99 & 99 & 99 & 99
\end{bmatrix}
$$

Table 1.9: Standard JPEG quantization tables for the luminance and chrominance channels with JPEG-quality parameter Q=50.

image, such as the border of the hat or the contours of the eyes and nose. These structures can be faithfully restored by the use of adaptive-shape supports which are determined from the luminance channel, as shown in Figure 1.24. It is remarkable that even small details such as the iris can be accurately reconstructed from the coarse available information using adaptive transform's supports. The restored color image (PSNR=31.00 dB) is shown in Figure 1.22(right). The ringing and the blocking artifacts disappeared, whereas no details have been oversmoothed, demonstrating the superior adaptivity of the approach. Moreover, thanks to the accurate reconstruction of the structures in the chrominance channels, our estimate does not exhibit any significant chromatic distortion and has a natural appearance.

Although it is well-established that the human visual system is less sensitive to distortions in the chrominances than to those in the luminance, the importance of restoring the chrominances must not be overlooked. In fact, all modern image and video compression standards are designed to exploit the characteristics of the human visual system, and thus adjust the compression rate for the luminance and chrominance channels in such a way to balance the perceptual impact of the distortions among the three channels. Therefore, when visual quality is of concern, the restoration of the different channels deserves equal attention. The downsampling and the coarser quantization of the chrominances makes their accurate restoration a much more difficult and challenging task.

Figure 1.25 provides a final example of the accuracy of the proposed method. First, one can see the sharp reconstruction of contours (e.g., in the legs, shoulders and head). Color-bleeding and blocking artifacts are completely suppressed, not only on smooth regions but even on rather thin details such as the snorkel. Second, the figure shows that the method is still reliable even when no useful structural information can be extracted from the luminance channel. In particular, it can be seen that the swimsuit is composed of three differently colored patches, all

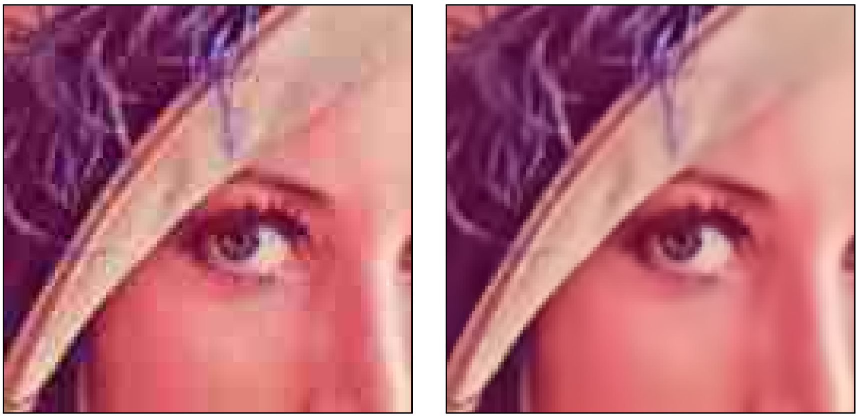


Figure 1.22: Fragment of the JPEG-compressed *Lena* image (Q=20, 0.38bpp, PSNR=29.83dB) and of its Pointwise SA-DCT filtered estimate (PSNR=31.00dB).

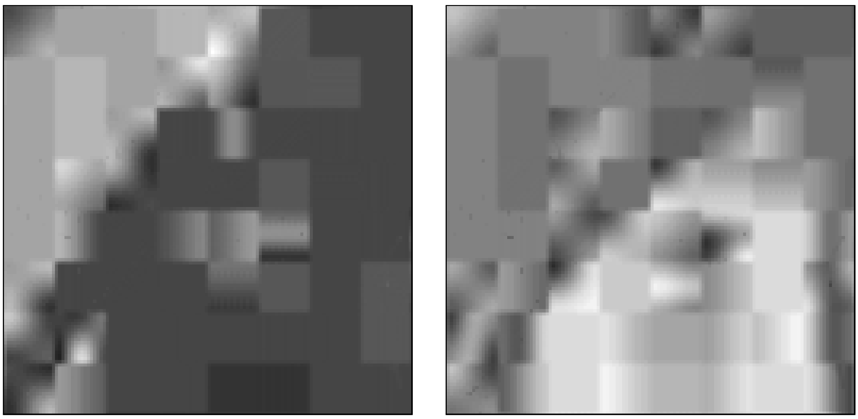


Figure 1.23: The U and V chrominances of the JPEG-compressed *Lena* image shown in Figure 1.22(left). Only very few DCT harmonics survived the aggressive quantization, and the structural information is almost completely lost.

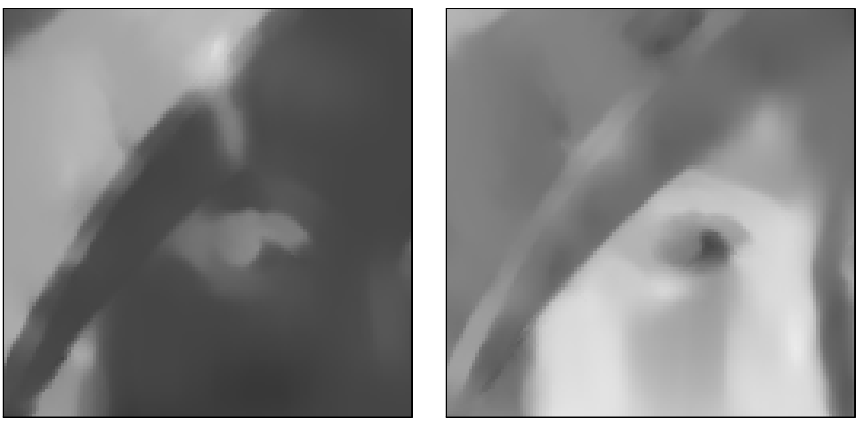


Figure 1.24: The chrominances shown in Figure 1.23 after reconstruction by Pointwise SA-DCT filtering. The blockiness is removed and the structures are faithfully restored.

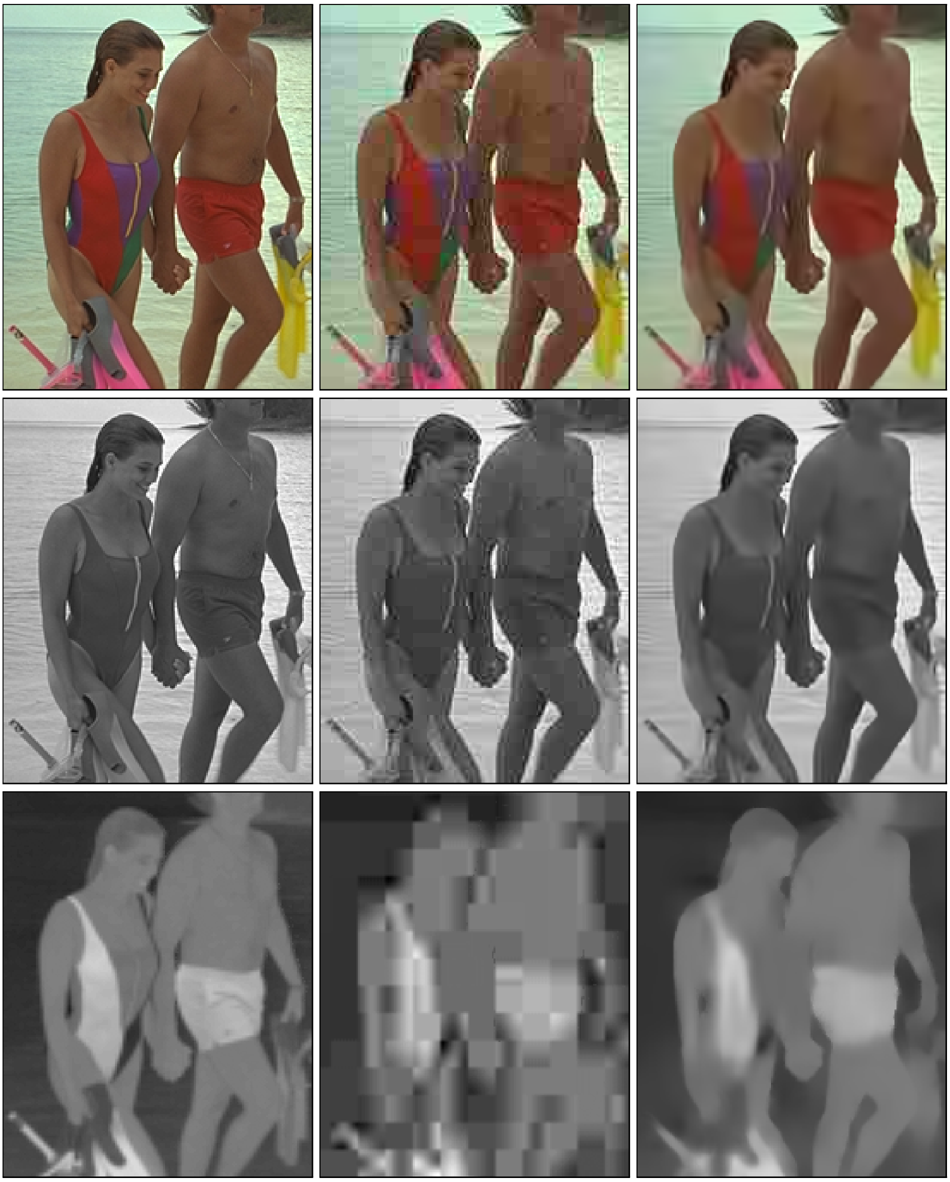


Figure 1.25: Fragments of the original, compressed, and restored *Kodak image 12*. Top to bottom row: *RGB* color, luminance Y channel, chrominance V channel. From left to right: original image, JPEG-compressed (Q=16, 0.25bpp, PSNR=30.45dB), restored by proposed Pointwise SA-DCT method (PSNR=31.45dB).

of which have the same luminance. This makes impossible to reconstruct the boundaries between these patches in a very sharp manner, as the only information available lies in the chrominances. Nevertheless, because the SA-DCT is a basis (complete system), the different colors of these patches are well preserved, while the transform-domain thresholding effectively suppresses the blockiness.

Method σ	*Lena* 512×512 15	20	25
Pointwise SA-DCT	33.86 ***0.891***	32.62 ***0.872***	31.66 ***0.855***
BLS-GSM (Portilla et al.) [138]	33.90 *0.888*	32.66 *0.869*	**31.69** *0.851*
TLS (Hirakawa & Parks) [78]	**33.97** ***0.891***	**32.69** ***0.872***	**31.69** ***0.855***
σ	*Cameraman* 256×256 15	20	25
Pointwise SA-DCT	**31.70** ***0.902***	**30.18** ***0.875***	**29.11** ***0.852***
BLS-GSM (Portilla et al.) [138]	30.91 *0.872*	29.43 *0.838*	28.35 *0.809*
TLS (Hirakawa & Parks) [78]	30.96 *0.882*	29.47 *0.853*	28.39 *0.830*
σ	*Barbara* 512×512 15	20	25
Pointwise SA-DCT	31.37 *0.910*	30.00 *0.886*	28.94 *0.862*
BLS-GSM (Portilla et al.) [138]	31.90 *0.901*	30.35 *0.871*	29.37 *0.842*
TLS (Hirakawa & Parks) [78]	**32.55** ***0.912***	**31.06** ***0.889***	**29.89** ***0.866***

Table 1.10: PSNR (dB, in regular type) and SSIM index [172] (in italic type) comparison table for the denoising of the grayscale *Lena*, *Cameraman*, and *Barbara* test images with different levels of Gaussian noise. Observe that there is no consistency between the PSNR and SSIM for the *Barbara* image.

1.8.4 Subjective perceptual quality

Although the presented figures already show that the Pointwise SA-DCT estimates are visually very good, we wish to mention that an independent research has been conducted [166, 167], aiming at evaluating the perceptual quality of the estimates of many state-of-the-art denoising methods. The research was based on an extensive psychovisual experiment where several noisy images were denoised by various algorithms and then subjectively judged by a large group of observers. Aspects such as blurriness, noisiness, presence of artifacts, etc., were investigated. Our Pointwise SA-DCT algorithm was among the considered denoising algorithms, which included many state-of-the-art methods such as BLS-GSM [138], BiShrink [150], ProbShrink [136], and a preliminary version of our BM3D [30]. The analysis of the results of this experimental research attests that the Pointwise SA-DCT estimates clearly outperformed in terms of overall subjective quality all estimates produced by techniques of other authors. Only the BM3D [30] estimates were judged better than the those obtained by the Pointwise SA-DCT algorithm.

In [167], it is shown that even in those cases where in terms of PSNR the Pointwise SA-DCT estimate might be inferior to the BLS-GSM estimate [138], thanks to the absence of artifacts and the better preservation of details the Pointwise SA-DCT estimate still provides a significantly superior visual quality. Here, we give some examples of this fact using the Structural Similarity (SSIM) index [172] as a reference-based measure of perceptual quality. The SSIM, though simple, is rather popular and has been demonstrated to be well correlated with the perceptual quality assessed by human observers over a large database of degraded images [152]. The SSIM index is a number in the interval $(0, 1]$, where a larger number corresponds to a higher quality and the value 1 represents the ideal per-

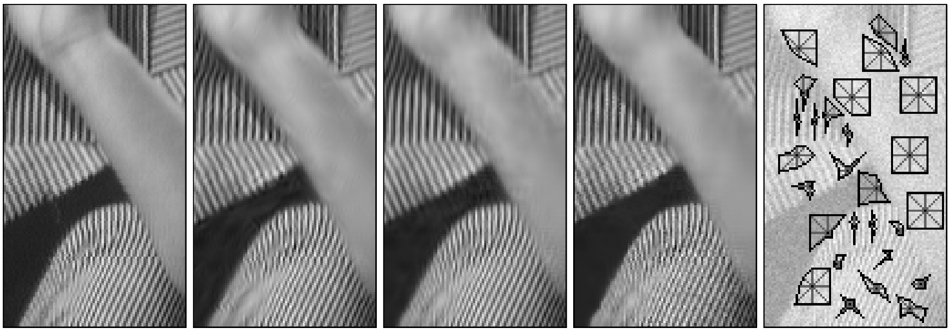


Figure 1.26: A fragment of *Barbara*: original image, BLS-GSM estimate (PSNR=30.35dB SSIM=0.871), TLS estimate (PSNR=31.06dB SSIM=0.889), Pointwise SA-DCT estimate (PSNR=30.00dB SSIM=0.886), and some of the anisotropic neighborhoods $\tilde{U}_x^+$ used by the Pointwise SA-DCT. All estimates have been obtained from the same noisy image with σ=20, shown beneath the adaptive neighborhoods.

fect quality[9]. In Table 1.10 we compare the PSNR and SSIM results obtained by the Pointwise SA-DCT, BLS-GSM [138], and TLS [78] algorithms for denoising the *Lena*, *Cameraman*, and *Barbara* test images with different levels of Gaussian noise. As the table shows, for the *Barbara* image there is no consistency between the PSNR and SSIM and the PSNR differences of 1dB or more between the Pointwise SA-DCT and the TLS estimates does not correspond to a concrete difference in quality, as both methods obtain for this image essentially the same SSIM index. The BLS-GSM, which achieves higher PSNR than the SA-DCT on the *Lena* image and, especially, on the *Barbara* image, has instead always a lower SSIM. Figure 1.26 shows a detail of the denoised *Barbara* for these three methods.

The Pointwise SA-DCT estimate of *Barbara* has a lower PSNR because this image is dominated by a high-frequency regular texture of large amplitude (trousers and scarf). Usually, high-frequency textures of small or moderate amplitude are not detected as singularities by the LPA-ICI. However, in this case the large black-white variations are so tall that the Anisotropic LPA-ICI detects them as edges, and thus selects small scales (i.e. short kernel supports). As a result, the SA-DCT is applied on narrow supports. This can be clearly seen in the rightmost subimage in Figure 1.26, which shows some of the anisotropic neighborhoods $\tilde{U}_x^+$. For such textures, a non-adaptive support of relatively large size (say, a 16×16 block) would be a much better choice when combined with a DCT transform. It means that, because of the LPA-ICI, in this particular case we obtain a poor result in terms of PSNR. This is the price we pay for adaptivity. Fortunately, this loss in PSNR turns out to have little value in terms of perceptual quality, as demonstrated in Table 1.10 and in Figure 1.26. In fact, it is well known that noise and errors are less visible in regions of higher activity such as texture (see, e.g., [174], [44], and references therein).

The results on perceptual quality recently reported in the DenoiseLab package [104] (also based on the SSIM index [172]) fully agree with the above findings:

[9] SSIM is equal to 1 when two identical images are compared (tested for structural) similarity, i.e. in the ideal case when the denoised image coincides with the original reference image.

the Pointwise SA-DCT and the BM3D outperform all other denoising algorithms tested (which include, among others, the Non-local means [15], BLS-GSM [138], and K-SVD [38]). The DenoiseLab results give evidence of another interesting fact: a PSNR vs. SSIM analysis (in a sense, aiming to answer the question "How much is a dB really worth?") shows that, among all tested algorithms, the Pointwise SA-DCT denoising algorithm consistently gives the highest perceptual quality (calculated as SSIM) for a given PSNR value of the denoised image. This suggests that its denoising strategy is "perceptually optimized", mainly because the noise is attenuated most effectively in areas where its visibility would be higher, while the errors (due to artifacts or noise) are mostly left in areas where they do not degrade the perceptual quality.

1.8.5 Complexity

When considering the computational complexity of the proposed algorithm, the first thing to observe is that the LPA-ICI technique is fast, because it is based on convolutions against one-dimensional kernels for a very limited number of directions. It constitutes a negligible computational overhead for the whole Pointwise SA-DCT filtering algorithm, whose complexity is instead essentially determined by the calculation of the forward and inverse SA-DCT transforms for every processed neighborhood.

The complexity of the algorithm is linear with respect to the size of the image and depends on the shapes of the transform supports. Since such shapes are pointwise-adaptive, the complexity depends on the particular image and noise. On average, and without resorting to fast algorithms, the asymptotic computational complexity of processing a single neighborhood is $\mathcal{O}\left(N^3\right)$, with N^2 being the size of the neighborhood. However, fast algorithms for calculation of the shape-adaptive DCT transforms do exists (e.g., [159]), thus the complexity is $\mathcal{O}\left(N^2 \log N\right)$. In our examples N can vary between 1 and 17, with the most frequent values between 6 and 12. On a more practical level, it is important to remark that highly-optimized hardware platforms (designed for real-time SA-DCT coding of high-resolution video) are available.

Our proposed method is also fully scalable in terms of complexity/performance trade-off. Scalability is achieved by decreasing the number of different scales and especially by limiting the number of overlapping neighborhoods (and thus the overcompleteness). In practice, we do not process a neighborhood $\tilde{U}_x^+$ if x belongs to a sufficient number M_{overlap} of already-processed neighborhoods: by decreasing M_{overlap} fewer forward and inverse SA-DCT transforms are computed. In order to count the number of overlaps at each point, we use a buffer $\sum_{x'\in X_{\text{proc}}} \chi_{\tilde{U}_{x'}^+}$, where the sum is calculated, cumulatively, over the set X_{proc} of already-processed neighborhood centers (it is worth comparing this buffer with the denominator of (1.10)). Figure 1.27 shows how a significant acceleration of the algorithm can be achieved in this manner with only a marginal loss in the final estimate's quality. Execution times refer to our current MATLAB implementation of the Pointwise SA-DCT filter (including Anisotropic LPA-ICI and SA-DCT hard-thresholding and Wiener filtering) measured on a 1.5-GHz Pentium M CPU. This MATLAB demonstration software is not optimized in terms of computational efficiency (in particular we do

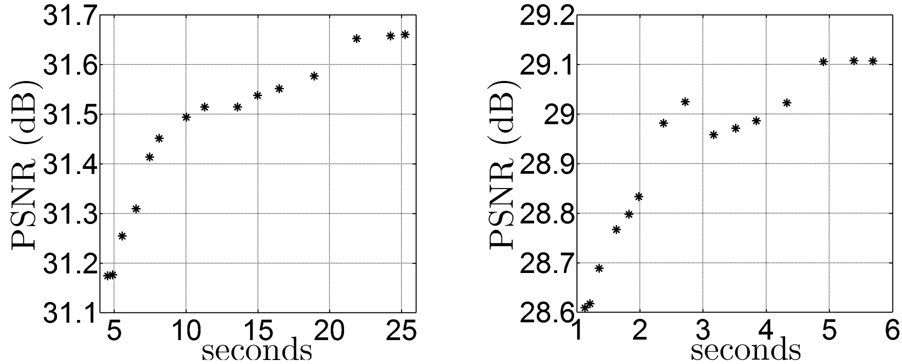


Figure 1.27: Denoising performance (PSNR, dB) vs. execution time (seconds) plots for the grayscale 512×512 *Lena* (left) and 256×256 *Cameraman* (right) images, σ=25. Execution time refers to the current MATLAB implementation measured on a 1.5-GHz Pentium M CPU.

not implement the SA-DCT using fast algorithms), nevertheless its execution time is comparable to other advanced transform-based denoising algorithms. As a comparison, on the same machine it takes about 90 seconds to denoise the grayscale *Lena* image (σ=25) using the BLS-GSM algorithm [138] (PSNR=31.69 dB), about 8 seconds using the ProbShrink algorithm [136] (PSNR=31.21 dB), about 17 seconds using the BM3D algorithm [32] (PSNR=32.08 dB), and about 103 minutes using the TLS algorithm [78] (PSNR=31.69 dB).

When discussing about complexity/performance trade-off, it is important to underline that in terms of PSNR the empirical Wiener filtering estimate (1.17) is usually about 0.3-0.5 dB better than the simpler hard-thresholding estimate (1.10) that is used as reference signal for (1.17). Figure 1.12 gives an example of this improvement. Since the empirical Wiener filtering accounts roughly for half of the overall complexity of the algorithm, the algorithm can be made faster by simply skipping this second-stage filtering. However, it can be seen from the plots in Figure 1.27 that a much more efficient trade-off is achieved using the above scalability strategy: execution time can be halved (from 25 to 12 seconds for *Lena* and from 5.5 to 2.5 seconds for *Cameraman*) sacrificing as little as 0.15 dB in PSNR.

It is worth also mentioning that the adaptive CR/RC processing (Section 1.5.1) can improve the PSNR considerably at a negligible computational cost, which is given by the evaluation of the inequality (1.5). Table 1.11 provides few experimental results on these issues. In the table, one can see that the best results are obtained using the adaptive CR/RC mode and that the non-adaptive CR mode yields (at least on these common test images) mostly better results than the non-adaptive RC mode. We argue that Sikora and Makai purposely decided to use the CR mode in their original SA-DCT [156] because of a similar experimental finding. It may be surprising that randomly selecting CR or RC modes (with equal probability) results in PSNR values competitive to those of the Adaptive CR/RC. This fact is due essentially to the adaptive weights used in the aggregation, which automatically favour the local estimates coming from sparser decompositions: typically

	Lena 512x512	*C'man* 256x256	*Peppers* 256x256	*Boats* 512x512
Reference	**31.66**	**29.11**	**29.92**	**29.47**
Random CR/RC	31.62	29.05	29.85	29.39
CR mode only	**31.66**	29.03	29.77	29.46
RC mode only	31.47	28.93	29.78	29.16
basic alignment	31.65	29.08	29.91	29.45

Table 1.11: PSNR (dB) comparison table for denoising grayscale images (σ=25) with different modes for the row-wise / column-wise processing of the SA-DCT or with the basic coefficient alignment instead of the alignment from Section 1.4.3. The "Reference" method stands for our default implementation, which uses adaptive CR/RC mode (inequality (1.5)) and special alignment (Section 1.4.3).

Figure 1.28: Denoising of *Cameraman* (σ=25, observation shown in Figure 1.11) using B-DCT with fixed block of size 8×8 (left) and 16×16 (right). The PSNRs of the two estimates are 28.83dB and 28.63dB, respectively. Compare with the Pointwise SA-DCT estimate shown in Figure 1.11 (PSNR=29.11dB).

these decompositions are obtained using the same CR or RC mode that would be adaptively selected from the inequality (1.5).

Finally, we wish to note that the impact of the coefficient alignment described in Section 1.4.3 is marginal in terms of PSNR (as can be seen from the last row in Table 1.11) and negligible in terms of complexity (look-up-tables can be utilized), hence its use shall depend exclusively on particular software/hardware design requirements.

1.8.6 Discussion

It is natural to ask to what extent the use of a shape-adaptive transform contributes to the objective and subjective quality achieved by the proposed method. In Figure 1.28 we show two estimates obtained by the denoising algorithm described in

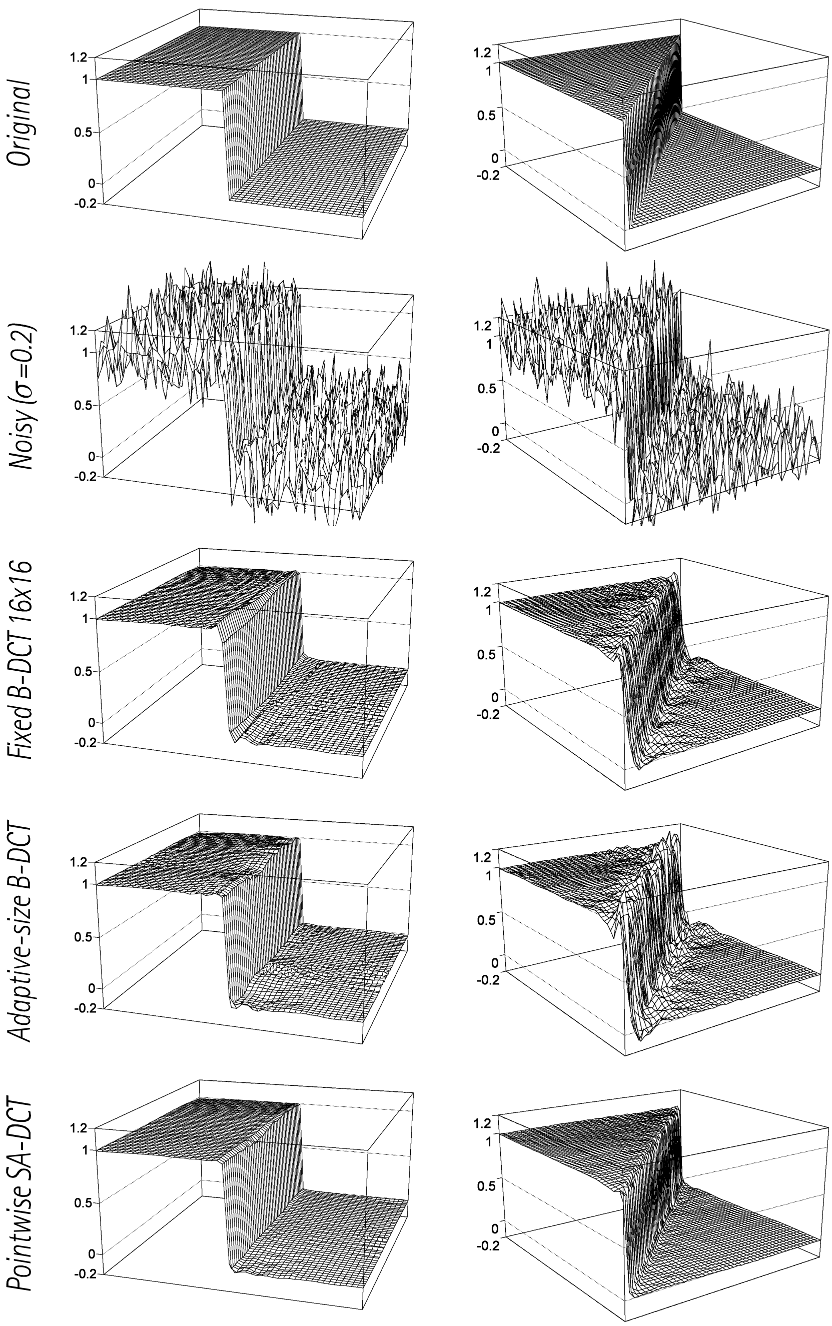


Figure 1.29: Comparison of three different denoising methods based on local (sliding) discrete cosine transforms. From top to bottom: Original signals (unit steps, "cartesian" and "diagonal"); noisy observations ($\sigma = 0.2$); estimates using B-DCT with fixed block of size 16×16, adaptive-size B-DCT estimates [52]; Pointwise SA-DCT estimates. The data have size 64×64, of which we shown only the innermost 50×50 samples.

Sections 1.5.2–1.5.4 (hard thresholding and Wiener filtering in transform-domain) using a square block of fixed size (8×8 and 16×16) (like in [180, 75]) instead of the adaptively shaped support $\tilde{U}_x^+$. Although in terms of PSNR the results are quite satisfactory, the visual quality of the estimates can be disappointing: several artifacts are visible, including blurring and ringing around the edges. For these two estimates, instead of relying on Equation (1.8), we used MSE-optimal values of the hard-threshold. We note that the overall visual quality cannot be really improved by increasing or decreasing this threshold, because that would either accentuate the blur or introduce more artifacts. Thus, these may be considered as the best estimates which can be obtained by this algorithm using square blocks of these fixed sizes. The Pointwise SA-DCT estimate shown in Figure 1.11 not only has higher PSNR, but also exhibits significantly sharper edges and fewer artifacts. Figure 1.29 presents a further denoising experiment where we compare the use of the SA-DCT against the B-DCT with fixed block of size 16×16 and against the B-DCT with adaptive-block size (essentially the algorithm proposed in [52], which improves over the similar older algorithms [124],[93]). In the figure, we can see that the Pointwise SA-DCT filter again provides a much cleaner estimate, free of ringing and with excellent noise attenuation, in smooth regions as well as in the vicinity of the discontinuity.

While the DCT on blocks of fixed size gives essentially only spatial and frequency selectivity, the Pointwise SA-DCT provides an image representation that combines spatial (adaptation is pointwise), frequency (using SA-DCT harmonics), scale (size of transform basis elements is adaptive), and directional (support is anisotropic) selectivity. The locally-adaptive supports can thus be rightly considered as the main reason of the success demonstrated by the Pointwise SA-DCT filter here, in the aforementioned psychovisual experiment [166, 167], and in the DenoiseLab results [104]. Further advantages of our method arise when we consider color image processing, since the *structural constraint in luminance-chrominance space* cannot be realized using blocks of fixed size.

Figure 1.30 provides an illustration of the elements $\psi^{(i)}_{\tilde{U}_x^+}$, $i \in V_{\tilde{U}_x^+}$, (1.12) of an orthonormal SA-DCT basis and of their frequency-domain characteristics. In the same figure we show also a normalized DC element $\chi_{\tilde{U}_x^+}|\tilde{U}_x^+|^{-\frac{1}{2}}$, which can be thought as the element used to perform the mean subtraction. The corresponding Fourier-domain power spectrum is shown below each SA-DCT basis element and next to the DC element. More illustrations of the SA-DCT bases are given in the following chapters and especially in Appendix B.

1.9 Conclusions to the chapter

We presented a novel image filtering approach based on the shape-adaptive DCT transform (SA-DCT). Hard-thresholding and empirical Wiener filtering are performed in SA-DCT domain, with an arbitrarily-shaped transform's support which is adaptively defined for every point in the image. The approach is used for the accurate denoising of grayscale as well as color images. Besides noise removal, the proposed method is also effective in dealing with those artifacts which are often encountered in block-DCT compressed images and videos. Blocking arti-

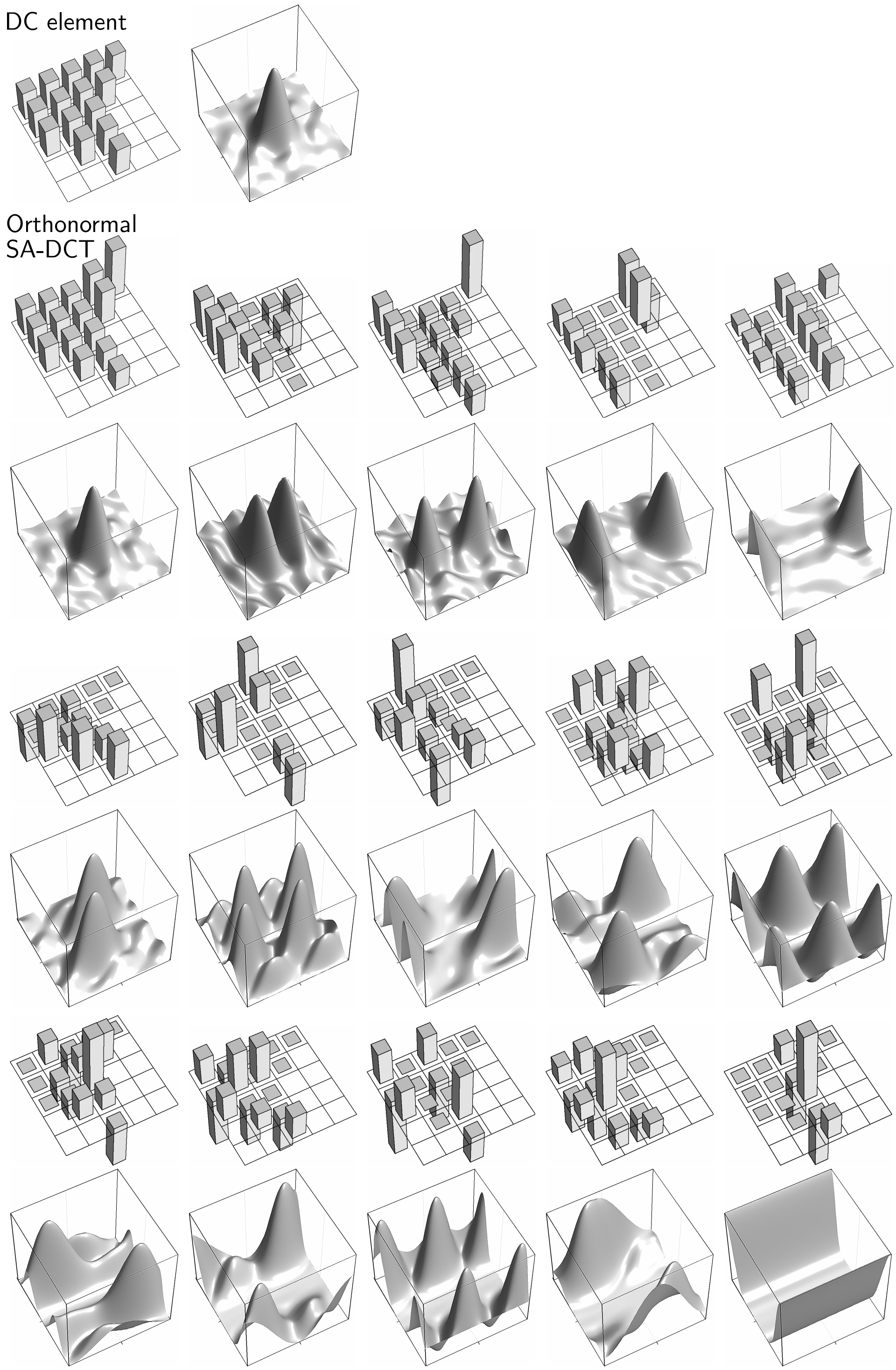


Figure 1.30: A complete orthonormal set of 15 SA-DCT basis elements and a DC (constant) element used for the mean subtraction (top). The respective Fourier-domain power spectrum is shown below each SA-DCT basis element and to the right of the DC element.

facts are suppressed while salient image features are preserved. The luminance-driven shape-adaptive filtering can faithfully reconstruct the missing structural information of the chrominances, thus correcting color-bleeding artifacts. The visual quality of the estimates is high, with sharp detail preservation, clean edges, and without unpleasant artifacts introduced by the fitted transform. Moreover, this is achieved at a very competitive computational cost. Overall, the Pointwise SA-DCT filtering algorithms demonstrate a remarkable performance, often outperforming all previous methods known to the authors.

Chapter 2

Pointwise SA-DCT regularized deconvolution

In this chapter we extend the Pointwise SA-DCT denoising approach to image deblurring. First, let us generalize the observation model (1.1) given in Section 1.2, introducing an optical blurring into the observation process. This distortion is commonly modeled by the convolution $y \circledast v$ of the true image y with the point-spread function (PSF) v of the optical system. Thus, we wish to recover y from the blurred and noisy observations

$$z(x) = (v \circledast y)(x) + \eta(x), \qquad x \in X. \tag{2.1}$$

In the frequency domain the above equation becomes $Z = YV + \tilde{\eta}$, where capital letters are used for the discrete Fourier transform of the corresponding variables and $\tilde{\eta} = \mathcal{F}(\eta)$, $\tilde{\eta}(\cdot) \sim \mathcal{N}\left(0, |X|\,\sigma^2\right)$. It is assumed that the PSF v is known.

Note that the observation model (1.1) is a particular case of (2.1) where v is a Dirac function.

The proposed deblurring algorithm is based on the anisotropic *regularized inverse* (RI) and *regularized Wiener inverse* (RWI) LPA-ICI deconvolution scheme developed in [94, 91]. The same deconvolution technique had also been used for inverse halftoning [56] and Poissonian deblurring [50].

The idea of the proposed SA-DCT domain deblurring is to combine the SA-DCT transform with the regularized inversion operators, using the adaptive supports defined by the anisotropic LPA-ICI regularized deconvolution. The flowchart of the overall deblurring algorithm is shown in Figure 2.1.

2.1 Regularized inverse

The regularized inverse (RI) z^{RI} is computed in the frequency domain as

$$z^{RI} = \mathcal{F}^{-1}\left(T^{RI} Z\right), \qquad T^{RI} = \frac{\overline{V}}{|V|^2 + \varepsilon_1^2}, \tag{2.2}$$

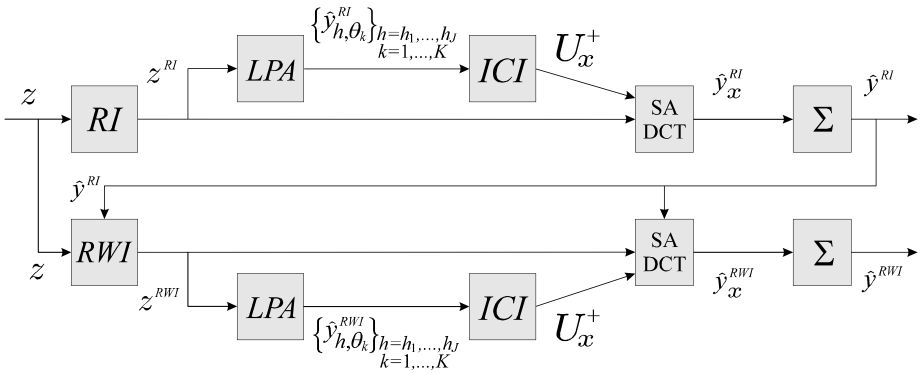


Figure 2.1: Flowchart of the SA-DCT regularized Wiener inverse algorithm. In the upper part of the flowchart a deblurred estimate $\hat{y}^{RI}$ is obtained by combining the regularized-inversion (RI) operator with the SA-DCT-domain hard-thresholding. This estimate is used as reference estimate for the regularized-Wiener stage of the lower part of the flowchart, where a regularized Wiener inverse (RWI) operator is combined with Wiener filtering in SA-DCT domain. The intermediate $\hat{y}^{RI}$ and the final $\hat{y}^{RWI}$ estimates are respectively obtained by averaging with adaptive weights (represented as "Σ") the local SA-DCT estimates $\hat{y}_x^{RI}$ and $\hat{y}_x^{RWI}$, whose supports are defined by the anisotropic LPA-ICI technique on the regularized inverse z^{RI} and regularized Wiener inverse z^{RWI}. Examples of the estimates obtained at the various stages of the algorithm are shown in Figure 2.2.

where T^{RI} is a (Tikhonov) regularized-inversion operator, $\varepsilon_1 > 0$ is a regularization parameter[1], and $\overline{V}$ is the complex conjugate of V. In what follows we make explicit use of the impulse response t^{RI} of the regularized-inversion operator, $t^{RI} = \mathcal{F}^{-1}(T^{RI})$.

2.2 Adaptive anisotropic transform support

The anisotropic neighborhood U_x^+ is constructed from the adaptive scales $\{h^+(x,\theta_k)\}_{k=1}^K$ obtained from the following procedure. The LPA is performed in the spatial domain as a convolution of the regularized inverse z^{RI} against the LPA kernel g_{h,θ_k}, $\hat{y}^{RI}_{h,\theta_k} = z^{RI} \circledast g_{h,\theta_k}$, for each specified direction θ_k, $k = 1,\ldots,K$, and for every scale $h \in H$. All the varying-scale LPA estimates $\{\hat{y}^{RI}_{h,\theta_k}\}_{h\in H}$ obtained for each θ are fed (together with their standard deviations $\{\sigma_{\hat{y}^{RI}_{h,\theta}} = \sigma \left\|t^{RI} \circledast g_{h,\theta}\right\|_2\}_{h\in H}$) into the ICI algorithm, which selects the pointwise-adaptive scale $h^+(x,\theta_k)$. The adaptive scale selection is done *independently* for each direction θ_k.

[1] The regularized inverse estimate (2.2) corresponds to the least-squares problem with penalty on the ℓ^2 norm of the estimate $T^{RI}Z = \operatorname{argmin}_Y \|YV - Z\|_2^2 + \varepsilon_1^2 \|Y\|_2^2$. When $\varepsilon_1 \to 0$, the regularized inverse converges to a pseudo-inverse, typically becoming unbounded.

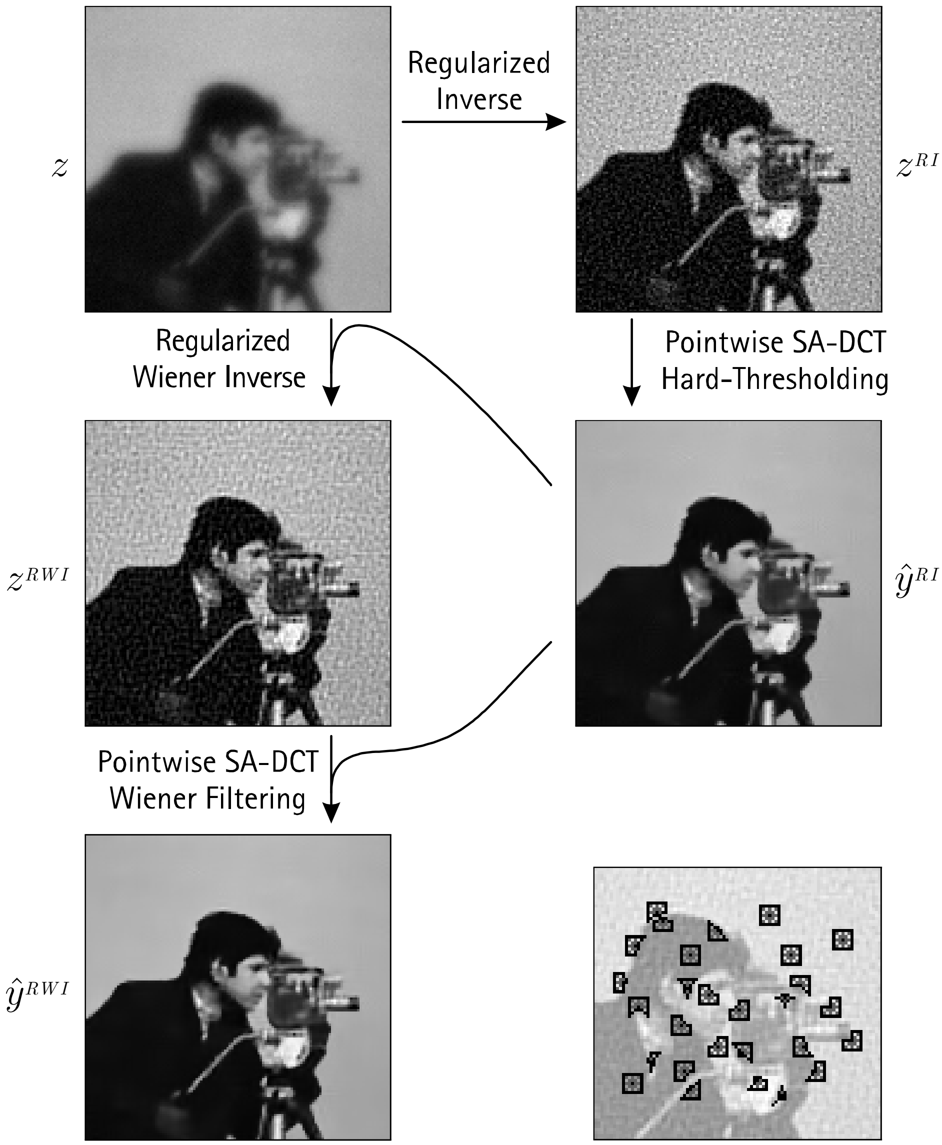


Figure 2.2: Examples of the estimates obtained at the various stages of the Pointwise SA-DCT regularized deconvolution algorithm (see also the flowchart of the algorithm in Figure 2.1). The images correspond to the deblurring Experiment 3, described in Table 2.1. The ISNR of the $\hat{y}^{RI}$ and $\hat{y}^{RWI}$ estimates is 5.57 and 6.34 dB, respectively. Observe the amplification of the noise after the inversion of the blur. Some granularity typical of colored noise is also visible. The subimage at the bottom-right shows some of the anisotropic neighborhoods used as support for the SA-DCT in the Pointwise SA-DCT Wiener filtering stage. Here $H = \{1, 2, 4\}$.

2.3 Pointwise SA-DCT deblurring

A non-linear adaptive deblurring operator is obtained by combining the regularized inversion (2.2) with the shape-adaptive DCT domain shrinkage. While formally it is enough to replace z by z^{RI} in (1.6), several further modifications to the basic denoising procedure of Section 1.5 are required in order to properly take into account the initial regularized inversion.

The principal difference arises from the fact that, because of the blur inversion (2.2), the noise in z^{RI} is colored. Therefore it is not possible anymore to guarantee that the noise variance in the SA-DCT domain is constant (as observed instead in Section 1.4.1). Because the noise variance can possibly vary significantly from a transform coefficient to the other, a constant threshold like (1.7) cannot be used in order to determine which coefficients should be used for the reconstruction of the deblurred estimate.

2.4 Thresholding with adaptive variance in transform domain

We consider the individual variance of the transform coefficients

$$\varphi_{z^{RI},x} = T_{\tilde{U}_x^+}\left(z^{RI}_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}\left(z^{RI}\right)\right)$$

Let $i \in V_{\tilde{U}_x^+}$ be coordinates in SA-DCT domain and $\psi^{(i)}_{\tilde{U}_x^+} : \tilde{U}_x^+ \to \mathbb{R}$ the SA-DCT basis functions, as defined in (1.12). Neglecting the correlating effect of the mean subtraction ("DC separation"), the variance $\sigma^2_{\varphi_{z^{RI},x}}(i)$ of the i-th transform coefficient $\varphi_{z^{RI},x}(i)$ is

$$\sigma^2_{\varphi_{z^{RI},x}}(i) = \sigma^2 \left\| t^{RI} \circledast \psi^{(i)}_{\tilde{U}_x^+}{}_{|X} \right\|_2^2 = \frac{\sigma^2}{|X|} \left\| T^{RI} \mathcal{F}(\psi^{(i)}_{\tilde{U}_x^+}{}_{|X}) \right\|_2^2, \tag{2.3}$$

The appropriate threshold level for each transform coefficient is obtained by replacing the coefficient-invariant σ in the threshold (1.8) with the varying standard deviation $\sigma_{\varphi_{z^{RI},x}} : V_{\tilde{U}_x^+} \to \mathbb{R}$. In this way, we realize an adaptive hard-thresholding rule for the joint deblurring and filtering in SA-DCT domain.

Analogously to (1.6), a local deblurred estimate $\hat{y}^{RI}_{\tilde{U}_x^+}$ is obtained by inverse transformation of the thresholded coefficients.

2.5 Adaptive weights and global estimate

Just as in Section 1.5.3, in order to obtain a global estimate $\hat{y}^{RI} : X \to \mathbb{R}$ all the local estimates $\hat{y}^{RI}_{\tilde{U}_x^+}$ are averaged together using adaptive weights $w_x^{RI} \in \mathbb{R}$ in a convex combination analogous to (1.10). However, because of the different variances of the coefficients used in the reconstruction, the adaptive weights are defined by a different formula. Ignoring the correlation between the noise in the transform domain, the corresponding expression of (1.11) for deblurring is

$$w_x^{RI} = \frac{1}{\left(\sigma^2_{m_{z^{RI},x}} + \sum_{j \in S} \sigma^2_{\varphi_{z^{RI},x}}(j)\right) |\tilde{U}_x^+|}$$

where $S \subseteq V_{\tilde{U}_x^+}$ indicates the coefficients which are non-zero after thresholding and

$$\sigma^2_{m_{z^{RI},x}} = \sigma^2 \left\| t^{RI} \circledast \frac{\chi_{\tilde{U}_x^+}}{\sqrt{|\tilde{U}_x^+|}} \right\|_2^2 = \frac{\sigma^2}{|\tilde{U}_x^+|\,|X|} \left\| T^{RI} \mathcal{F}(\chi_{\tilde{U}_x^+}) \right\|_2^2 \qquad (2.4)$$

is an extra term which accounts for the "DC separation". In (2.3) we interpret the DC as calculated by a normalized basis element of the form $\frac{\chi_{\tilde{U}_x^+}}{\sqrt{|\tilde{U}_x^+|}}$, therefore $\mathrm{var}\{m_{z^{RI},x}\} = \sigma^2_{m_{z^{RI},x}} / |\tilde{U}_x^+|$.

2.6 Regularized Wiener inverse

The regularized Wiener inverse (RWI) z^{RWI} is computed as

$$z^{RWI} = \mathcal{F}^{-1}(T^{RWI} Z), \qquad T^{RWI} = \frac{\overline{V}|Y|^2}{|VY|^2 + \varepsilon_2^2\,|X|\,\sigma^2}, \qquad (2.5)$$

where $\varepsilon_2 > 0$ is a regularization parameter[2]. Since the spectrum $|Y|^2$ of the true image is unknown, the estimate $\hat{y}^{RI}$ from the RI stage is used quite naturally as a "pilot" estimate in the Wiener filtering. It means that $|Y|^2$ in (2.5) is replaced by $|\hat{Y}^{RI}|^2$.

Analogously to Section 2.2, the adaptive neighborhoods $\tilde{U}_x^+$ are defined by the anisotropic LPA-ICI technique applied to z^{RWI}.

On these neighborhoods we perform empirical Wiener filtering in SA-DCT domain in a fashion similar to Section 1.5.4. More precisely, for every $x \in X$, let $\varphi_{\hat{y}^{RI},x} : V_{\tilde{U}_x^+} \to \mathbb{R}$ be the SA-DCT (on $\tilde{U}_x^+$) coefficients of $\hat{y}^{RI}$ where the mean $m_{\tilde{U}_x^+}(z^{RWI})$ of z^{RWI} is subtracted before applying the transform:

$$\varphi_{\hat{y}^{RI},x} = T_{\tilde{U}_x^+}\left(\hat{y}^{RI}_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}(z^{RWI})\right). \qquad (2.6)$$

The *local* Wiener estimate $\hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+}$ is defined as

$$\hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+} = T^{-1}_{\tilde{U}_x^+}\left(\omega_x \varphi_{z^{RWI},x}\right) + \varpi_x m_{\tilde{U}_x^+}(z^{RWI}), \qquad (2.7)$$

where the attenuation coefficients for this filter are

$$\omega_x = \frac{\varphi^2_{\hat{y}^{RI},x}}{\varphi^2_{\hat{y}^{RI},x} + \sigma^2_{\varphi_{z^{RWI},x}}}, \qquad \varpi_x = \frac{m^2_{\tilde{U}_x^+}(\hat{y}^{RI})}{m^2_{\tilde{U}_x^+}(\hat{y}^{RI}) + \sigma^2_{m_{z^{RWI},x}} / |\tilde{U}_x^+|}.$$

[2] If $\varepsilon_2 = 1$, the inverse operator T^{RWI} (2.5) corresponds to the solution of the minimum MSE problem $T^{RWI} = \mathrm{argmin}_T\, E\left\{\|TZ - Y\|_2^2\right\}$. An $\varepsilon_2 \neq 1$ produces inverse operators with a balance between bias and variance different than the optimum one, obtained for $\varepsilon_2 = 1$. In particular, the smaller is ε_2, the smaller is the bias and the larger is the variance (i.e. noise) in the inverse estimate z^{RWI}. Since we are able to further suppress the noise exploiting the Pointwise SA-DCT filter, it is worth choosing $\varepsilon_2 \ll 1$. In this way we are able to achieve lower bias and in the end (after Pointwise SA-DCT filtering) also lower variance (thus, lower MSE) than by the optimum Wiener inverse filter with $\varepsilon_2 = 1$.

The variances $\sigma^2_{\varphi_{zRWI,x}}$ and $\sigma^2_{m_{zRWI,x}}$ are defined as in (2.3) and (2.4), but using the regularized Wiener inverse operator T^{RWI} or its impulse response $t^{RWI} = \mathcal{F}^{-1}(T^{RWI})$ instead of T^{RI} or t^{RI}, respectively. These variances are also used to define the adaptive weights used for the aggregation of the local estimates (2.7):

$$w_x^{RWI} = \frac{1}{\left(\varpi_x^2 \sigma^2_{m_{zRWI,x}} + \sum_{V_{\tilde{U}_x^+}} \omega_x^2 \sigma^2_{\varphi_{zRWI,x}}\right) |\tilde{U}_x^+|}.$$

2.7 Fast computation of coefficients' variance

The calculation of the variance of the transform coefficients as in (2.3) is a rather computationally intense procedure. In order to reduce complexity to an acceptable level, it is necessary to employ some approximations.

Our implementation of (2.3) is based on processing in an undersampled (coarse) FFT domain of size 32×32. Denoting by $\widetilde{\mathcal{F}}$ the corresponding transform, by $\widetilde{X}$ a 32×32 domain, and by $\mathcal{L}$ a decimator with prefiltering, we have the following approximation of the variance of the i-th transform coefficient $\varphi_{zRI,x}(i)$:

$$\begin{aligned} \sigma^2_{\varphi_{zRI,x}}(i) &\simeq \frac{\sigma^2}{|\widetilde{X}|} \left\| \mathcal{L}\{|\mathcal{F}(t^{RI})|^2\} \, |\widetilde{\mathcal{F}}(\psi^{(i)}_{\tilde{U}_x^+}{}^{|\widetilde{X}})|^2 \right\|_1, \\ \sigma^2_{m_{zRI,x}} &\simeq \frac{\sigma^2}{|\tilde{U}_x^+||\widetilde{X}|} \left\| \mathcal{L}\{|\mathcal{F}(t^{RI})|^2\} \, |\widetilde{\mathcal{F}}(1_{|\tilde{U}_x^+}{}^{|\widetilde{X}})|^2 \right\|_1. \end{aligned} \tag{2.8}$$

An obvious modification to the above formula allows to compute approximations of the variances $\sigma^2_{\varphi_{zRWI,x}}$. Figures 2.3 and 2.4 show some SA-DCT basis functions $\psi^{(i)}_{\tilde{U}_x^+}$ and their corresponding power spectrum $|\widetilde{\mathcal{F}}(\psi^{(i)}_{\tilde{U}_x^+}{}^{|\widetilde{X}})|^2$, as used in (2.8). These examples, together with those in Figure 1.30, clearly show that the SA-DCT basis elements are well localized in the frequency plane. This property results in large differences in the variances $\sigma^2_{\varphi_{zRI,x}}(i)$ when $|\mathcal{F}(t^{RI})|$ is non-uniform (e.g., concentrated at particular frequencies).

The approximations (2.8) have a significant impact on the computational performance of the algorithm. On a 1.5-GHz Pentium M CPU, the current MATLAB implementation can process a 256×256 image in about 9 seconds and a 512×512 image in about 38 seconds. This includes RI and RWI stages, comprehensive of LPA-ICI and transform-domain filtering. If (2.3) were used instead of the approximate (2.8), the processing time would be of several minutes, without any noticeable improvement in the restoration quality.

2.8 Experiments

In order to evaluate the restoration ability of the proposed deblurring algorithm, we consider the standard set of four blurred and noisy observations which has been used by many other authors. Table 2.1 presents the description of the observations' parameters. The blur PSFs for these experiments are shown in Figure 2.5. Roughly speaking, these four experiments represent different combinations of

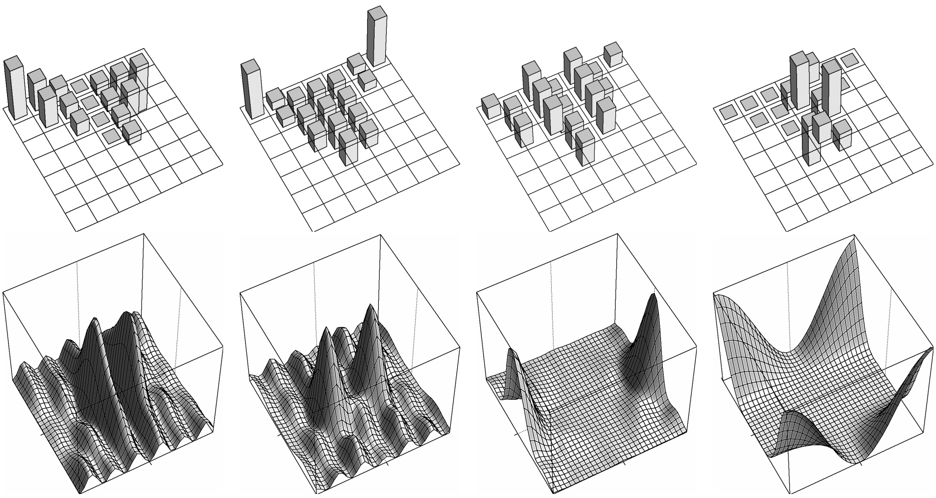


Figure 2.3: Some SA-DCT basis elements $\psi^{(i)}_{\tilde{U}^+_x}$ and their power spectrum $|\widetilde{\mathcal{F}}(\psi^{(i)}_{\tilde{U}^+_x}{}^{|\tilde{X}})|^2$ used in the fast computation of the coefficients' variance (2.8).

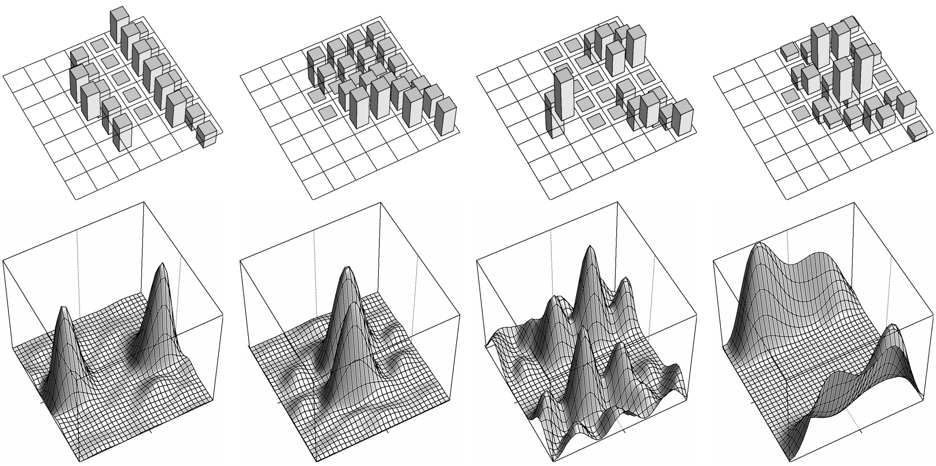


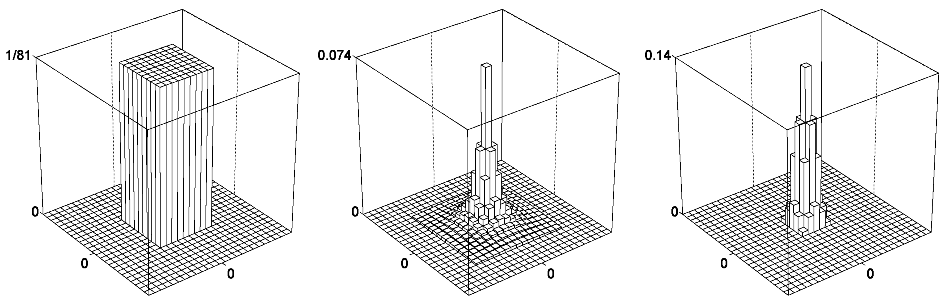
Figure 2.4: Some SA-DCT basis elements $\psi^{(i)}_{\tilde{U}^+_x}$ and their power spectrum $|\widetilde{\mathcal{F}}(\psi^{(i)}_{\tilde{U}^+_x}{}^{|\tilde{X}})|^2$ used in the fast computation of the coefficients' variance (2.8).

blur and noise in a gradation from the strong blur with weak noise (comparable to quantization noise) of Experiment 1, to the mildly blurred image with noticeable noise of Experiment 4. The spectral characteristics of the blur PSFs used in the experiments can be seen in Figure 2.6. In the same figure, we show also the amplitudes $|T^{RI}|$ and $|T^{RWI}|$ of the Fourier-domain regularized inverse (2.2) and regularized Wiener inverse operators (2.5). The regularization parameters ε_1 and ε_2 have been chosen manually for each experiment[3]. For calculating the SA-DCT coefficients' variance (2.8), we use the downsampled $\mathcal{L}\{|T^{RI}|^2\}$ and $\mathcal{L}\{|T^{RWI}|^2\}$,

[3] Adaptive procedures for the automatic selection of the regularization parameters can also be utilized (e.g., [121], [178], [162]).

Exp.	Image, Point-Spread Function v,	noise
1	*Cameraman* 256×256 9×9 boxcar uniform v	BSNR= 40 dB ($\sigma^2 \simeq 0.308$)
2	*Cameraman* 256×256 $v(x_1,x_2) = 1/\left(1+x_1^2+x_2^2\right),\ \ x_1,x_2=-7,\ldots,7$	$\sigma^2=2$
3	*Cameraman* 256×256 $v(x_1,x_2) = 1/\left(1+x_1^2+x_2^2\right),\ \ x_1,x_2=-7,\ldots,7$	$\sigma^2=8$
4	*Lena* 512×512 $v=[1,4,6,4,1]^T\,[1,4,6,4,1]/256$	$\sigma^2=49$

Table 2.1: Description of the observation parameters for the four deblurring experiments.



Exp. 1 Exp. 2 & Exp. 3 Exp. 4

Figure 2.5: Point-spread functions v for the four deblurring experiments of Table 2.1. The amplitude of the Fourier transform of these PSFs is illustrated in Figure 2.6.

shown in Figure 2.7. More precisely, we integrate the pointwise product of the power spectrum $|\widetilde{\mathcal{F}}(\psi^{(i)\,|\widetilde{X}}_{\tilde{U}^+_x})|^2$ (shown in Figures 2.3 and 2.4) with the downsampled power spectrum of the noise $\mathcal{L}\{|\mathcal{F}(t^{RI})|^2\}$ (whose root is shown in Figure 2.7) or $\mathcal{L}\{|\mathcal{F}(t^{RWI})|^2\}$. As already mentioned, the standard-deviation can vary significant from a coefficient to the other. For example, at the RI stage of Experiment 4 and for the same anisotropic support of the basis functions shown in Figure 2.4, the computed standard deviations of the SA-DCT coefficients and of the mean are

$$\sigma_{\varphi_{z^{RI},x}} = \begin{array}{|c|c|c|c|c|c|c|}\hline 12.80 & 15.20 & 21.05 & 20.93 & 19.66 & 8.86 & 3.67 \\ \hline 15.47 & 19.88 & 22.50 & 18.58 & 20.23 & 10.88 & \\ \hline 21.70 & 22.12 & 20.96 & 16.94 & 11.02 & 6.48 & \\ \hline 23.51 & 14.70 & 14.24 & & 16.47 & 6.27 & \\ \hline 12.57 & & & & & & \\ \hline \end{array} \qquad \sigma_{m_{z^{RI},x}} = 12.45.$$

The improvement-in-SNR (ISNR) results for these four experiments are given in Table 2.2. Figure 2.8 show details of the observations and the corresponding restored images. Similarly to denoising, both the objective and subjective quality of our estimates are high. In the case of deblurring, the Pointwise SA-DCT is particularly effective in coping with the ringing artifacts due to the regularization of the inverse operator. To the best of our knowledge, on these standard experiments, the proposed Pointwise SA-DCT regularized deconvolution outperforms all other methods to date.

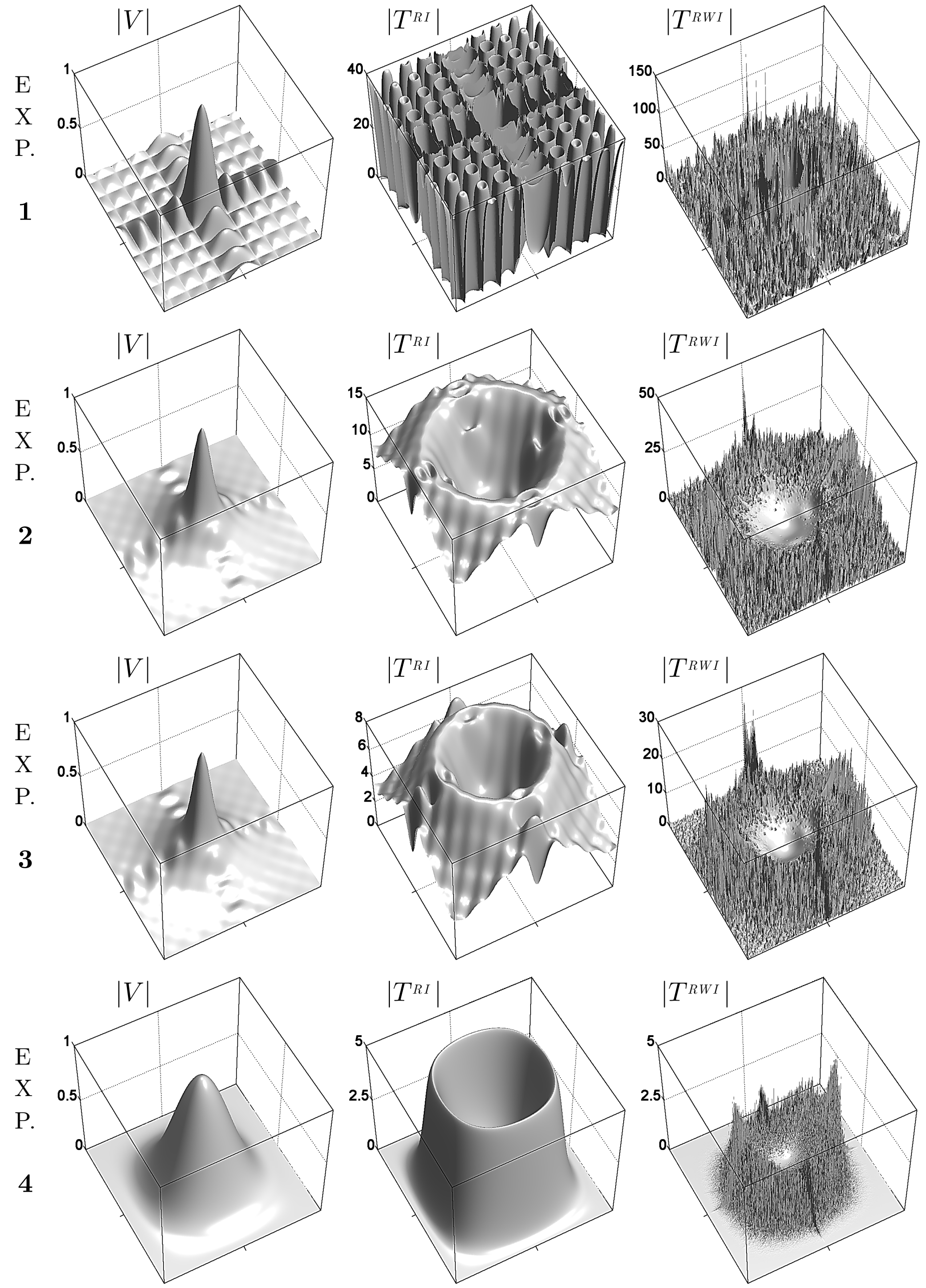


Figure 2.6: Amplitude of the Fourier transform V of the PSF v and of the regularized inverse and Wiener inverse operators T^{RI} and T^{RWI} for each of the four deconvolution experiments (Section 2.8).

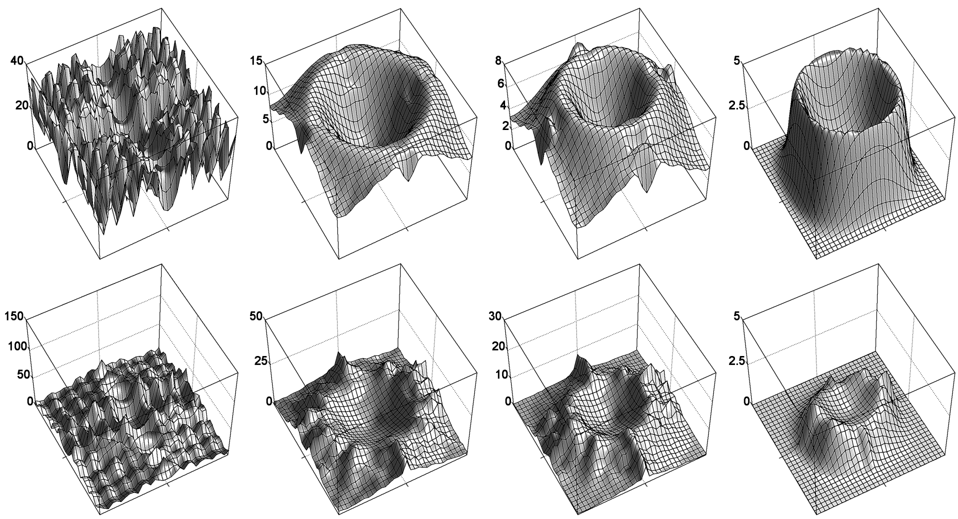


Figure 2.7: Square root of the downsampled power spectrum of the noise for the regularized inverse, $\sqrt{\mathcal{L}\{|\mathcal{F}(t_{\text{RI}})|^2\}}$, and for the regularized Wiener inverse, $\sqrt{\mathcal{L}\{|\mathcal{F}(t_{\text{RWI}})|^2\}}$, in each of the four deconvolution experiments (Section 2.8). These power spectra are used in (2.8). Compare with the corresponding undecimated functions in Figure 2.6.

Method / Experiment	1	2	3	4
Pointwise SA-DCT regularized deconvolution	**8.57**	**8.25**	**6.34**	**4.52**
Anisotropic LPA-ICI RI-RWI [91, 62]	8.29	7.82	5.98	3.90
BOA (Figueiredo and Nowak) [46]	8.16	7.46	5.24	2.84
GEM (Dias) [34]	8.10	7.47	5.17	2.73
Segmentation-based regularization (Mignotte) [115]	8.04	7.23	–	1.34
EM (Figueiredo and Nowak) [45]	7.59	6.93	4.88	2.94
ForWaRD (Neelamani et al.) [121]	7.30	6.75	5.07	2.98

Table 2.2: ISNR (dB) of the proposed deblurring algorithm and of other state-of-the-art methods for four standard experiments.

2.9 Some comments on the approach

Let us give some comments and discussion of the general nature on the deconvolution approach used for the two stages of our algorithm.

Given a frame $\left\{\psi^{(i)}\right\}_i$ and its dual $\{\breve{\psi}^{(i)}\}_i$, we can represent the original signal using the usual analysis-synthesis form

$$y = \sum\nolimits_i \langle y, \breve{\psi}^{(i)} \rangle \psi^{(i)}. \tag{2.9}$$

Let us now be given not y but its blurred (noise-free) observation $y \circledast v$.

There are essentially two strategies which can be followed starting from Equation (2.9).

Figure 2.8: From top to bottom: fragments of the observations z and of the corresponding Pointwise SA-DCT estimates $\hat{y}^{RWI}$ of four deblurring experiments. The ISNRs of these estimates are 8.57, 8.25, 6.34 and 4.52 dB, respectively.

2.9.1 Synthesis with $\{\psi^{(i)}\}_i$

We would like to reconstruct y using the frame elements $\psi^{(i)}$ in the form

$$y = \sum\nolimits_i c_i \psi^{(i)}, \tag{2.10}$$

where the coefficients c_i are computed from $y \circledast v$. Obviously, $c_i = \langle y, \breve{\psi}^{(i)} \rangle$ is a solution of the equation (and is the unique solution if the frame is a basis). For a generic y, the problem can be formally solved under the often unrealistic hypothesis that the blur operator is invertible. In this case we have that

$$\begin{aligned} \langle y, \breve{\psi}^{(i)} \rangle &= \frac{1}{|X|} \left\langle \mathcal{F}(y), \mathcal{F}(\breve{\psi}^{(i)}) \right\rangle = \frac{1}{|X|} \left\langle YV, \tfrac{\mathcal{F}(\breve{\psi}^{(i)})}{\overline{V}} \right\rangle = && (2.11) \\ &= \left\langle y \circledast v, \mathcal{F}^{-1}\left(\tfrac{\mathcal{F}(\breve{\psi}^{(i)})}{\overline{V}}\right) \right\rangle = && (2.12) \\ &= \left\langle y \circledast v, \breve{\psi}^{(i)} \circledast \mathcal{F}^{-1}\left(\tfrac{1}{\overline{V}}\right) \right\rangle, && (2.13) \end{aligned}$$

where $\overline{V}$ is the complex conjugate of V. So the coefficients c_i can be calculated as the inner product between the blurred observations and $\breve{\psi}^{(i)} \circledast \mathcal{F}^{-1}\left(\frac{1}{\overline{V}}\right)$.

2.9.2 Analysis with $\{\breve{\psi}^{(i)}\}_i$

An alternative approach to (2.10) is to seek a solution of the form

$$y = \sum\nolimits_i \langle y \circledast v, \breve{\psi}^{(i)} \rangle \xi^{(i)}, \tag{2.14}$$

where the true signal is analyzed with respect to the dual frame $\{\breve{\psi}^{(i)}\}_i$ and synthesized with an appropriate set of reconstructing functions $\left\{\xi^{(i)}\right\}_i$. In order to get an explicit form for $\left\{\xi^{(i)}\right\}_i$ we convolve the left and right-hand side of the above equation against v, obtaining

$$y \circledast v = \sum\nolimits_i \langle y \circledast v, \breve{\psi}^{(i)} \rangle \left(\xi^{(i)} \circledast v\right). \tag{2.15}$$

This perfect reconstruction formula implies that $\left\{\xi^{(i)} \circledast v\right\}_i$ is a dual frame of $\{\breve{\psi}^{(i)}\}_i$. For example, $\xi^{(i)} \circledast v = \psi^{(i)}$, which gives (assuming invertibility of V)

$$\xi^{(i)} = \mathcal{F}^{-1}\left(\tfrac{\mathcal{F}(\psi^{(i)})}{V}\right). \tag{2.16}$$

2.9.3 Ill-posedness and regularization

Usually, blur operators are not invertible. In the case V has zeros, the simplest ("naïve") approach is to use a generalized inverse of V, i.e.

$$V^{-1}(\cdot) = \begin{cases} 0 & \text{if } V(\cdot) = 0 \\ 1/V(\cdot) & \text{if } V(\cdot) \neq 0 \end{cases}.$$

However, even when V does not have zeros, the problem can still remain ill-posed, with the inverse operator V^{-1} being unbounded. Consequently, when approaching noisy observations of the form (2.1), the typical strategy is to employ regularization at some stage of the inversion procedure.

The simplest form of regularization is the following (Tikhonov) regularized inverse [168],[40]

$$T^{\scriptscriptstyle RI} = \frac{\overline{V}}{|V|^2 + \varepsilon^2}, \tag{2.17}$$

where $\varepsilon > 0$ is the regularization parameter. A larger regularization parameter corresponds to a more stable (lower variance) but also more biased inverse estimate. The regularization (2.17) has been used extensively for deblurring, in the spatial domain (e.g., [10], [91], [94], [50], [168]), as well as in the transform domain (e.g., [121], [53]). Similar forms of regularization combined with transform-domain filtering can be found also in [7] and [120]. In all these approaches, the combination of Fourier-domain regularization with other filtering techniques (e.g., adaptive smoothing, shrinkage, etc.) is shown to lead to an improved inverse estimate.

2.9.4 Intermediate cases; comments; vaguelettes

In addition to the two strategies from Sections 2.9.1 and 2.9.2, there are also infinitely many intermediate cases. Exploiting in (2.9) the fact that formally $\langle y, \breve{\psi}^{(i)} \rangle = \frac{1}{|X|} \left\langle YV^{\alpha}, \mathcal{F}(\breve{\psi}^{(i)})\overline{V}^{-\alpha} \right\rangle$, it is easy to show that $\left\{ \psi^{(i)} \circledast \mathcal{F}^{-1}\left(V^{\alpha}\right) \right\}_i$ and $\left\{ \breve{\psi}^{(i)} \circledast \mathcal{F}^{-1}\left(\overline{V}^{-\alpha}\right) \right\}_i$ are also a pair of frames in duality, for any $\alpha \in \mathbb{R}$. By using this new pair of frames instead of $\left\{ \psi^{(i)} \right\}_i$ and $\{ \breve{\psi}^{(i)} \}_i$ in (2.10) and by varying α, one can obtain the first ($\alpha = 0$) and the second strategy ($\alpha = 1$), all intermediate cases ($0 < \alpha < 1$), as well as other decompositions ($\alpha < 0$, $\alpha > 1$).

Although the two approaches might seem formally equivalent, they become in practice very different as soon as they are considered with respect to a predefined "meaningful" frame.

Equation (2.10) implicitly assumes that $\left\{ \psi^{(i)} \right\}_i$ is a suitable frame to enable a good approximation of y. Since the inverse operator is embedded in the analysis frame, the approach works according to the paradigm "first invert the blur, and then approximate this inverse".

On the contrary, Equation (2.14) assumes that $\left\{ \psi^{(i)} \right\}_i$ is suitable to approximate $y \circledast v$, with the inverse operator embedded in the reconstruction frame $\left\{ \xi^{(i)} \right\}_i$. Thus, it essentially obeys to the paradigm "first approximate the blurred signal, and then invert this approximation".

If not all coefficients are used for the reconstruction, the two approaches lead to significantly different approximations of y.

In our work, we consider frames corresponding to redundant shape-adaptive DCT transforms defined on pointwise adaptive starshaped domains. Redundancy (overcompleteness) comes from the fact that multiple overlapping starshaped domains are used and that for each such domain a complete SA-DCT transform is computed. Originally, this deconvolution approach was studied with $\left\{\psi^{(i)}\right\}_i$ and $\{\breve{\psi}^{(i)}\}_i$ being biorthogonal wavelet bases: the representations (2.10) and (2.14) are respectively known within the wavelet community as *wavelet-vaguelette* [35, 107] and *vaguelette-wavelet* [1] decompositions. "Vaguelette" stands for the combination of the inverse of the blur with the corresponding dual wavelet, such as $\xi^{(i)}$ from (2.16). We refer the reader to [1] for a theoretically-oriented discussion and comparison of the asymptotical risk of estimators based on the two approaches.

Let us note that the frame $\left\{\psi^{(i)}\right\}_i$ and its dual $\{\breve{\psi}^{(i)}\}_i$ can be obviously interchanged in all the above equations.

2.9.5 Practical aspects

Pragmatically, if to be used in estimation from noisy observations, each one the two approaches has its own advantages and disadvantages. In particular, assuming a properly normalized frame and independent noise with constant variance, shrinkage of coefficients which are calculated directly from the blurred observations as in (2.15) is simplified. Note that (diagonal) shrinkage requires the knowledge of the standard-deviation of each coefficient, which can be calculated according to the simple formula

$$\operatorname{std}\{\langle f, g\rangle\} = \sqrt{\left\langle \operatorname{var}\{f\}, |g|^2 \right\rangle},$$

provided the independency of the noise in f. If $\operatorname{var}\{f\}$ is also constant, say, $\operatorname{var}\{f\} = \sigma^2$, then

$$\operatorname{std}\{\langle f, g\rangle\} = \sigma \left\| g \right\|_2. \tag{2.18}$$

Thus, as far as coefficient shrinkage is concerned, the approach is very similar to transform-based denoising from standard Gaussian noise. On the other hand, the convolution against the (regularized) inverse of V can cause that $\left\{\xi^{(i)}\right\}_i$ does not enjoy any of the good decorrelation properties of the original frame $\left\{\psi^{(i)}\right\}_i$. Thus, the approximation ability of the approach (2.14) can be seriously impaired.

In (2.10) the original frame is used for reconstruction, however, even in the simplest case of noise with constant variance, the shrinkage of coefficients obtained by (2.13) is rather involved because the combination of dual frame with the inverse operator.

Nevertheless in practice, since the main motivation to use a transform-based method is its ability of to represent the signal to be recovered with good approximation by using only few transform coefficients (i.e. sparsity), the approach from Section 2.9.1 is more appropriate, despite the calculation of the standard-deviation (required for the shrinkage) is usually more involved.

2.9.6 Decoupling of the transform and the inversion

As mentioned in Section 2.9.4, Equation (2.10) works according to the paradigm "first invert the blur, and then approximate this inverse".

If in the term $\left\langle YV, \frac{\mathcal{F}(\breve{\psi}^{(i)})}{\overline{V}}\right\rangle$ from (2.11) we replace the noise-free data YV with Z and the naive inverse $\frac{1}{V}$ with a regularized inverse (2.17), we obtain that the coefficients c_i are calculated as

$$\begin{aligned}\frac{1}{|X|}\left\langle Z, \mathcal{F}(\breve{\psi}^{(i)})\frac{V}{|V|^2+\varepsilon^2}\right\rangle &= \frac{1}{|X|}\left\langle Z\frac{\overline{V}}{|V|^2+\varepsilon^2}, \mathcal{F}(\breve{\psi}^{(i)})\right\rangle = \\ &= \left\langle z^{RI}, \breve{\psi}^{(i)}\right\rangle,\end{aligned}$$

where $z^{RI} = \mathcal{F}^{-1}\left(\frac{Z\overline{V}}{|V|^2+\varepsilon^2}\right)$ is the regularized-inverse estimate of y. Although, in principle, different forms of regularization (e.g., different regularization parameters) could be used for different coefficients, for computational reasons it is preferable to use a unique regularized inverse z^{RI} for all i. In this way the calculation of the inner products is done exactly as for the analysis of z^{RI} with the frame $\{\breve{\psi}^{(i)}\}_i$. Thus, we come to the following approximation of y in terms of the frame $\left\{\psi^{(i)}\right\}_i$

$$y \simeq \sum\nolimits_i \langle z^{RI}, \breve{\psi}^{(i)}\rangle \psi^{(i)}.$$

The coefficients $\langle z^{RI}, \breve{\psi}^{(i)}\rangle$ are noisy and some shrinkage should be performed in order to obtain a good estimate. The noise in z^{RI} is not white, but colored through inverse filtering with T^{RI} (2.17), thus the thresholds used in the shrinkage are different for different coefficients. In order for this shrinkage to be effective it is important that the frame $\left\{\psi^{(i)}\right\}_i$ represents well y. The redundant shape-adaptive DCT transform on adaptive supports is thus an excellent choice for $\left\{\psi^{(i)}\right\}_i$ when natural images are considered as the original signal to be reconstructed[4].

Finally, the estimate has the generic form

$$\sum\nolimits_i \hat{\varphi}(i)\, \psi^{(i)}, \tag{2.19}$$

where $\hat{\varphi}$ are the filtered coefficients obtained by shrinkage (e.g., hard-thresholding) of $\langle z^{RI}, \breve{\psi}^{(i)}\rangle$.

2.9.7 Global Fourier-domain inverse, periodicity, and localized processing

In our approach, we explicitly use the impulse response t^{RI} of the regularized-inversion operator, $t^{RI} = \mathcal{F}^{-1}(T^{RI})$. The inversion (2.2) is naturally computed globally, assuming periodicity of the processed signal and of the convolutions used in the corresponding space-domain operations. Obviously, such modeling does not

[4] When comparing these equations with those from Section 1.5.3, one should observe that the role of $\breve{\psi}^{(i)}$ and $\psi^{(i)}$ is reversed, as the emphasis is placed on two different parts of the analysis-reconstruction paradigm. Here, we fix the reconstruction frame, whereas there is the analysis frame that is fixed. Overall, the aim is to reconstruct the deblurred signal using the same frame used for denoising.

correspond to the reality, as the translation-invariant blur due to optics or rigid motion does not make the signal periodic. The application of the inverse (2.2) on a non-periodic z results in dramatic boundary artifacts, which often may cause the whole estimate to be degenerate. However, there are a number of efficient *sectioning* techniques (e.g., [3], [142]) which enable the profitable use of the inverse (2.2) on non-periodic signals. The key idea of these techniques is to force z to be periodic, replacing its original "non-periodic boundaries" with synthesized smoothed periodic boundaries (a procedure called *apodization* of the image). In a similar way, this allows to process large images in a localized or windowed fashion (which is necessary when severely limited memory or computational power is available). Whenever t^{RI} has a fast enough decay, such sectioning techniques can be used without producing any visible distortions.

Clearly, the same considerations as above apply to the regularized Wiener inverse (2.5) also.

2.10 Extensions

The proposed deblurring algorithm can be easily extended to observation models more general than (2.1). In particular, we refer the reader to [31], where a straightforward modification of the algorithm–a modification that permits the noise η to be colored instead of white, as in (2.1)–is exploited for inverse-halftoning of error-diffused binary-halftone images.

Chapter 3

Signal-dependent noise removal

This chapter presents a novel and efficient method for denoising of images corrupted by signal-dependent noise based on shrinkage in the shape-adaptive DCT (SA-DCT) domain. The use of an adaptive transform support enables both a simpler modelling of the noise in the transform domain and a sparser decomposition of the signal. Consequently, coefficient shrinkage is very effective and the reconstructed estimate's quality is high, in terms of both numerical error-criteria and visual appearance, with sharp detail preservation and clean edges. Simulation experiments demonstrate the superior performance of the proposed algorithm for a wide class of noise models with a signal-dependent variance, including Poissonian (photon-limited imaging), film-grain, and speckle noise.

3.1 Introduction

In many applications the observed signal is corrupted by a signal-dependent noise. The most widely encountered models are Poisson, film-grain, multiplicative, and speckle noise. Their common feature is that the variance of the noise is directly related to the true-signal's intensity. In particular, because of the inherent "photon-counting" process within digital imaging sensors, the noise found in digital images is signal-dependent, with brighter parts of the image having a larger noise variance, and typically following a Poissonian distribution. These aspects will be discussed more in detail in the second part of the thesis.

Starting with classical filters, such as those by Lee, Kuan, and Frost ([105], [106], [103], [66]), a number of adaptive approaches for signal-dependent noise removal have been developed and proposed, in spatial (e.g., [87], [140], [51]) as well as in transform domain (e.g., [164], [173], [111], [52]).

In this chapter, we extend the Pointwise SA-DCT denoising filter, introduced in Chapter 1, to the filtering of signal-dependent noise. Also in this case, denoising is performed by coefficient shrinkage in the SA-DCT [154] domain, with the Anisotropic Local Polynomial Approximation (LPA) - Intersection of Confidence

Intervals (ICI) technique [91, 92] used to define the shape of the transform's support in a pointwise adaptive manner. On such adaptive supports the signal is smooth and nearly constant, allowing for a simpler modelling of the noise within each transform support and enabling a sparser decomposition of the signal in the transform domain. Thus, coefficient shrinkage (e.g., hard-thresholding and Wiener filtering) can be employed directly, accurately, and very effectively. As a result, the reconstructed estimate's quality is high, in terms of both numerical error-criteria and visual appearance. In particular, thanks to shape-adaptive transform supports, the estimates exhibit sharp detail preservation and clean edges.

The key element of our approach to signal-dependent noise is the use of a locally adaptive estimate of the noise variance, an estimate which is progressively refined during the various stages of the algorithm. The presented method generalizes and extends both the Pointwise SA-DCT filter for denoising of additive white Gaussian noise (AWGN) [58], presented in Chapter 1, and the Adaptive-Size Block-DCT (AS B-DCT) algorithm [52] for signal-dependent noise removal with adaptive-size block transforms, recently proposed by the authors.

The rest of the chapter is organized as follows. In the next section we introduce the general signal-dependent noise model and the notation used throughout the chapter. The algorithm is then presented in detail: construction of the adaptive-shape support, hard-thresholding in SA-DCT-domain, aggregation of overlapping local estimates, and empirical Wiener filtering in SA-DCT domain. The last section of the chapter is devoted to experimental results for removal of Poissonian (photon-limited imaging), film-grain, and speckle noises.

3.2 Signal-dependent noise model

We consider observations $z(x)$, $x \in X \subset \mathbb{Z}^2$, with the expectations $E\{z(x)\} = y(x) \geq 0$, where the errors (noise) $\eta(x) = z(x) - y(x)$ are independent and the variance of these observations is modeled as

$$\sigma_z^2(x) = \mathrm{var}\{z(x)\} = \mathrm{var}\{\eta(x)\} = \rho(y(x)), \tag{3.1}$$

$\rho : \mathbb{R}^+ \to \mathbb{R}^+$ being a given positive function of y called the *variance function*. For example, $\rho(y) = y$, $\rho(y) = y^2$, and $\rho(y) = (Ky^\alpha)^2$ for the Poisson, gamma, and film-grain observation models, respectively.

The problem is to reconstruct the true image y from the noisy observations z.

3.3 Algorithm

Similar to the algorithm for AWGN presented in Section 1.5, the following algorithm for signal-dependent noise comprises two stages: a first stage (Sections 3.3.1-1.5.3) based on hard-thresholding, and a second stage (Sections 3.3.2-3.3.2) based on empirical Wiener filtering, where the hard-thresholding estimate obtained in Section 1.5.3 is used as a reference estimate.

The first step in each stage employs the Anisotropic LPA-ICI technique [47, 91, 92] in order to identify adaptive neighborhoods where the image can be assumed to be locally smooth (polynomial fit) and nearly constant.

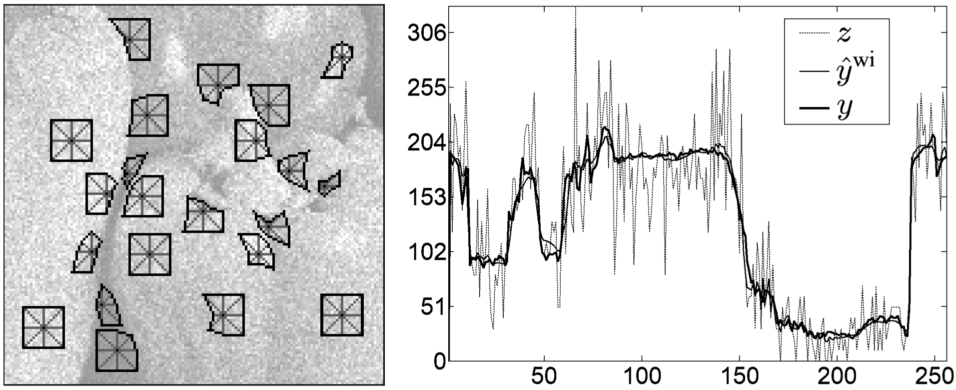


Figure 3.1: LPA-ICI anisotropic neighborhoods. Left: some examples of the anisotropic neighborhoods $\tilde{U}_x^+$ used for SA-DCT filtering of the *Peppers* image corrupted by Poissonian noise (χ=0.1). Right: the corresponding cross-sections (column 87) of the true image y, of the noisy observation z, and of the Pointwise SA-DCT estimate $\hat{y}^{\text{wi}}$. The signal-dependent nature of the noise is clearly visible.

3.3.1 Stage 1: Hard-thresholding in Pointwise SA-DCT domain

Adaptive-shape neighborhood

We initialize the algorithm with a simple, rough estimate $\hat{\sigma}_z^2$ of the variance σ_z^2 obtained as the variance function evaluated on the noisy data, $\hat{\sigma}_z^2 = \rho\left(|z|\right)$.

For every specified direction θ_k, $k = 1, \ldots, K$, a varying-scale family of 1-D directional-LPA convolution kernels $\{g_{h,\theta_k}\}_{h \in H}$ is used to obtain a corresponding set of directional varying-scale estimates $\{\hat{y}_{h,\theta_k}\}_{h \in H}$, $\hat{y}_{h,\theta_k} = z \circledast g_{h,\theta_k}$, $h \in H$, where $H \subset \mathbb{R}^+$ is the set of scales. For these 1-D kernels, the length of the support coincides with the value of the scale. The pointwise standard-deviations of the estimates $\hat{y}_{h,\theta_k}$ are estimated as $\hat{\sigma}_{\hat{y}_{h,\theta_k}} = \sqrt{\hat{\sigma}_z^2 \circledast g_{h,\theta_k}^2}$, $h \in H$. Thus, for each point (pixel) $x \in X$ we obtain a sequence of confidence intervals

$$\left\{\left[\hat{y}_{h,\theta_k}\left(x\right) - \Gamma\hat{\sigma}_{\hat{y}_{h,\theta_k}(x)}, \hat{y}_{h,\theta_k}\left(x\right) + \Gamma\hat{\sigma}_{\hat{y}_{h,\theta_k}(x)}\right]\right\}_{h \in H}$$

where Γ is a positive threshold parameter. These intervals are then compared according to the ICI rule [72, 90], and as a result an adaptive scale $h^+\left(x, \theta_k\right) \in H$ is defined for every $x \in X$. Precisely, the adaptive $h^+\left(x, \theta_k\right)$ is defined as the largest scale h such that the intersection of all confidence intervals corresponding to scales smaller than h is non-empty. The procedure is repeated for all specified directions. Thus, for a fixed $x \in X$ we have K directional adaptive scales $h^+\left(x, \theta_k\right)$, $k = 1, \ldots, K$.

In our implementation, like in Section 1.5.1, we use $K = 8$ directions and construct the adaptive neighborhood $\tilde{U}_x^+$ of x as $\tilde{U}_x^+ = \{v \in X : (x - v) \in U_x^+\}$, where U_x^+ is the polygonal hull of the supports of the adaptive-scale kernels $\left\{g_{h^+(x,\theta_k),\theta_k}\right\}_{k=1}^8$. Figure 3.1(left) shows some examples of these neighborhoods for a noisy image corrupted by Poissonian noise. Observe how these neighborhoods adapt to edges and variations in the image intensity and that in each neighborhood the underlying image is smooth and nearly constant.

Hard-thresholding in SA-DCT domain: local estimates

For every neighborhood $\tilde{U}_x^+$, $x \in X$, we construct a *local* estimate $\hat{y}_{\tilde{U}_x^+} : \tilde{U}_x^+ \to \mathbb{R}$ of the signal y by hard-thresholding in SA-DCT domain.

The typically encountered variance functions ρ, such as those mentioned in Section 3.2, are smooth functions of their argument y. As a consequence, on neighborhoods where the signal is nearly constant, so is its variance. It means that with a good approximation, the variance can be assumed to be constant when the transform's support is restricted to a region of signal uniformity such as the adaptive neighborhood $\tilde{U}_x^+$. Thus, locally, the hard-thresholding can be performed as in the standard case [58] where the variance is everywhere constant, provided that a locally adaptive estimate $\hat{\sigma}_z^2\left(\tilde{U}_x^+\right)$ of the variance of the noise in $\tilde{U}_x^+$ is used.

Such a locally adaptive $\hat{\sigma}_z^2\left(\tilde{U}_x^+\right)$ can be obtained from the mean $m_{\tilde{U}_x^+}(z)$ through the variance function as $\hat{\sigma}_z^2\left(\tilde{U}_x^+\right) = \rho\left(\left|m_{\tilde{U}_x^+}(z)\right|\right)$. For any given neighborhood $\tilde{U}_x^+$, this adaptive variance is a constant. Hence, similar to (1.6-1.7), the local estimate $\hat{y}_{\tilde{U}_x^+}$ is calculated as

$$\hat{y}_{\tilde{U}_x^+} = T_{\tilde{U}_x^+}^{-1}\left(\Upsilon_x\left(T_{\tilde{U}_x^+}\left(z_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}(z)\right)\right)\right) + m_{\tilde{U}_x^+}(z), \tag{3.2}$$

where Υ_x is a hard-thresholding operator with adaptive threshold

$$\gamma_{\text{thr}}\sqrt{\hat{\sigma}_z^2\left(\tilde{U}_x^+\right)\left(2\ln|\tilde{U}_x^+| + 1\right)}, \quad \gamma_{\text{thr}} > 0, \tag{3.3}$$

This threshold is again Donoho's "universal" threshold (1.8), with the same fixed parameter $\gamma_{\text{thr}} > 0$. However, here, instead of an invariant σ^2, we use the locally adaptive variance estimate $\hat{\sigma}_z^2\left(\tilde{U}_x^+\right)$.

Global estimate as aggregation of local estimates

The Anisotropic LPA-ICI provides an adaptive neighborhood $\tilde{U}_x^+$ for every $x \in X$. Neighborhoods corresponding to adjacent points are usually overlapping, thus the local estimates $\left\{\hat{y}_{\tilde{U}_x^+}\right\}_{x \in X}$ (3.2) constitute an overcomplete representation of the image. In order to obtain a single *global* estimate $\hat{y} : X \to \mathbb{R}$ defined on the whole image domain, all the local estimates (3.2) are averaged together using adaptive weights $w_x \in \mathbb{R}$ in the following convex combination:

$$\hat{y} = \frac{\sum_{x \in X} w_x \hat{y}_{\tilde{U}_x^+}{}^{|X}}{\sum_{x \in X} w_x \chi_{\tilde{U}_x^+}}. \tag{3.4}$$

Analogously to Section 3.3.1, the weights w_x depend on the total sample variance of $\hat{y}_{\tilde{U}_x^+}$–thus, on the number N_x^{har} of non-zero coefficients after thresholding and on the local variance $\hat{\sigma}_z^2\left(\tilde{U}_x^+\right)$–and on the size of the neighborhood $|\tilde{U}_x^+|$:

$$w_x = \frac{\hat{\sigma}_z^{-2}\left(\tilde{U}_x^+\right)}{\left(1+N_x^{\mathrm{har}}\right)|\tilde{U}_x^+|}. \tag{3.5}$$

3.3.2 Stage 2: Wiener filtering in Pointwise SA-DCT domain

Improved adaptive-shape neighborhood

The estimate of the variance $\hat{\sigma}_z^2$ used in Section 3.3.1 (calculated from z through the variance function ρ) is indeed rough. A better estimate $\hat{\sigma}_z^2$ can now be calculated from the above $\hat{y}$ as $\hat{\sigma}_z^2 = \rho\left(|\hat{y}|\right)$. The Anisotropic LPA-ICI approach is used once more to deliver more accurate adaptive scales. We modify slightly the procedure from Section 3.3.1, in that the LPA estimates $\{\hat{y}_{h,\theta_k}\}_{h\in H}$ are now calculated not from z but from $\hat{y}$ (3.4), as the convolution $\hat{y}_{h,\theta_k} = \hat{y} \circledast g_{h,\theta_k}$, $h \in H$. The standard-deviations needed for the construction of the confidence intervals are calculated again as $\hat{\sigma}_{\hat{y}_{h,\theta_k}} = \sqrt{\hat{\sigma}_z^2 \circledast g_{h,\theta_k}^2}$, $h \in H$, with $\hat{\sigma}_z^2 = \rho\left(|\hat{y}|\right)$. As a result of the ICI rule, we obtain, for each $x \in X$, the new directional adaptive scales $h^+\left(x,\theta_k\right)$, $k = 1,\ldots,K$, and thus the corresponding adaptive-shape neighborhood $\tilde{U}_x^+$.

Wiener filtering in SA-DCT domain: local estimates

Using the estimate $\hat{y}$ (3.4), the empirical Wiener filter in the SA-DCT domain works as follows. For a fixed x, let $\varphi_{z,x} : V_{\tilde{U}_x^+} \to \mathbb{R}$ and $\varphi_{\hat{y},x} : V_{\tilde{U}_x^+} \to \mathbb{R}$ be, respectively, the SA-DCT (on $\tilde{U}_x^+$) coefficient of z and $\hat{y}$, calculated as

$$\varphi_{z,x} = T_{\tilde{U}_x^+}\left(z_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}(z)\right), \tag{3.6}$$

$$\varphi_{\hat{y},x} = T_{\tilde{U}_x^+}\left(\hat{y}_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}(z)\right), \tag{3.7}$$

where the mean $m_{\tilde{U}_x^+}(z)$ of z is subtracted before applying the transform. The *local* Wiener estimate $\hat{y}_{\tilde{U}_x^+}^{\mathrm{wi}}$ is defined as

$$\hat{y}_{\tilde{U}_x^+}^{\mathrm{wi}} = T_{\tilde{U}_x^+}^{-1}\left(\omega_x \varphi_{z,x}\right) + \varpi_x m_{\tilde{U}_x^+}(z), \tag{3.8}$$

where $\omega_x \in \mathcal{V}_{\tilde{U}_x^+}$ and $\varpi_x \in \mathbb{R}$ are respectively the Wiener attenuation factors for $\varphi_{z,x}$ and for the subtracted mean value $m_{\tilde{U}_x^+}(z)$,

$$\omega_x = \frac{\varphi_{\hat{y},x}^2}{\varphi_{\hat{y},x}^2 + \sigma_x^2}, \qquad \varpi_x = \frac{m_{\tilde{U}_x^+}^2(\hat{y})}{m_{\tilde{U}_x^+}^2(\hat{y}) + \hat{\sigma}_z^2\left(\tilde{U}_x^+\right)/|\tilde{U}_x^+|}, \tag{3.9}$$

and $\hat{\sigma}_z^2\left(\tilde{U}_x^+\right)$ is a local adaptive estimate of the variance of z in $\tilde{U}_x^+$ calculated as $\rho\left(\left|m_{\tilde{U}_x^+}(\hat{y})\right|\right)$.

Final global estimate as aggregation of local estimates

The *global* estimate $\hat{y}^{\text{wi}}$ can be obtained analogously as in (3.4), using the convex combination with the adaptive weights w_x^{wi}:

$$\hat{y}^{\text{wi}} = \frac{\sum_{x\in X} w_x^{\text{wi}} \hat{y}^{\text{wi}}_{\tilde{U}_x^+}|^X}{\sum_{x\in X} w_x^{\text{wi}} \chi_{\tilde{U}_x^+}}, \quad w_x^{\text{wi}} = \frac{\hat{\sigma}_z^{-2}\left(\tilde{U}_x^+\right)}{(\varpi_x^2 + \sum_{V_{\tilde{U}_x^+}} \omega_x^2)|\tilde{U}_x^+|}. \tag{3.10}$$

Similarly to (3.5), the term $\hat{\sigma}_z^2\left(\tilde{U}_x^+\right)\left(\varpi_x^2 + \sum_{V_{\tilde{U}_x^+}} \omega_x^2\right)$ in the adaptive weights corresponds to an estimate of the total sample variance of $\hat{y}^{\text{wi}}_{\tilde{U}_x^+}$.

The Pointwise SA-DCT results which we present in this chapter correspond to the $\hat{y}^{\text{wi}}$ estimate (3.10).

3.3.3 Complexity

The presented method for signal-dependent noise removal inherits the same low computational complexity of the standard Pointwise SA-DCT developed for AWGN (Section 1.8.5). In fact, the extra operations in this generalized algorithm are essentially only the convolutions needed to calculate $\hat{\sigma}_{\hat{y}_{h,\theta_k}}$, and the Anisotropic LPA-ICI in the second stage of the algorithm. Both are computationally negligible compared to the multiple forward and inverse SA-DCT transforms performed in Sections 3.3.1 and 3.3.2. The locally adaptive variance estimates $\hat{\sigma}_z^2\left(\tilde{U}_x^+\right)$ are also calculated "almost for free", since the local means $m_{\tilde{U}_x^+}(z)$ and $m_{\tilde{U}_x^+}(\hat{y})$ are anyway required by standard algorithm for AWGN (in order to accomplish the DC separation).

3.4 Experimental results

We show experimental results for three common types of signal-dependent noise: the "scaled" *Poisson* noise, $\chi z \sim \mathcal{P}(\chi y)$, $\chi \in \mathbb{R}^+$, $\chi \neq 0$, the *generalized film-grain* noise, $z = y + (K_0 + K_1 y^\alpha)\eta$, $K_0, K_1, \alpha \in \mathbb{R}^+$ and $\eta(\cdot) \sim \mathcal{N}(0,1)$, and the "multiple-look" *speckle* noise, $z = L^{-1}\sum_{i=1}^L y\epsilon_i$, $\epsilon_i \sim \mathcal{E}(\beta)$, $\beta \in \mathbb{R}^+$, $\beta \neq 0$. The calligraphic letters $\mathcal{P}$, $\mathcal{N}$, and $\mathcal{E}$ denote, respectively, the Poisson, Gaussian, and exponential distributions. For the above observation models, the variance functions $\rho(y) = \sigma_z^2$ are $\rho(y) = y/\chi$, $\rho(y) = (K_0 + K_1 y^\alpha)^2$, and $\rho(y) = y^2\beta/L$, respectively. The true signal y is assumed to have range [0,255].

Firstly, in Table 3.1 we give a comparison for the simulations presented in [140] for the above noise models with parameters χ=0.1, K_0=0, K_1=3.3, α=0.5, L=4, and β=1. In the table our results are compared against the adaptive-neighborhood filter [140], the noise-updating repeated Wiener filter [87] (as quoted in [140]), the recursive anisotropic LPA-ICI technique [51, 62], and the AS B-DCT algorithm [52].

Next, in Table 3.2 we compare our results for removal of Poissonian noise ($\chi = \frac{30}{255}, \frac{60}{255}, \frac{90}{255}, \frac{120}{255}$) against those obtained by two other transform-based methods [164],[111] recently developed specifically for this type of noise, by the

	noise type	noisy	[87]	[140]	[51, 62]	[52]	P. SA-DCT
Peppers	*Poisson* (χ=0.1)	1243	160	145	120	104	**95**
	Film-grain	1351	169	150	123	109	**97**
	Speckle	4442	372	378	286	225	**193**
Aerial	*Poisson* (χ=0.1)	758	252	179	183	149	**141**
	Film-grain	829	267	188	185	154	**142**
	Speckle	1698	387	318	330	257	**242**

Table 3.1: MSE results for different noise types and denoising algorithms.

	χ	noisy	[164]	[111]	[51, 62]	[52]	P. SA-DCT
Lena	*30 / 255*	1054	168	143	73	61	**55**
	60 / 255	525	117	96	50	42	**38**
	90 / 255	349	93	75	40	34	**31**
	120 / 255	262	81	63	34	30	**27**
Cameraman	*30 / 255*	1015	199	154	136	107	**100**
	60 / 255	504	140	97	89	70	**68**
	90 / 255	336	113	74	70	55	**53**
	120 / 255	254	97	61	60	46	**45**

Table 3.2: MSE comparison against algorithms for Poissonian noise.

recursive anisotropic LPA-ICI technique [51, 62], and by the AS B-DCT algorithm [52].

The results in the tables show that the proposed algorithm outperforms all other methods, for all considered noise models and noise levels. In terms of MSE, the improvement is significant especially for higher noise.

Figures 3.2 and 3.3 provide a demonstration of the visual quality of the proposed Pointwise SA-DCT technique. Edges and small details are restored quite sharply, with very few noticeable artifacts. Figure 3.3 presents also a comparison between denoised estimates obtained by the AS B-DCT and by the Pointwise SA-DCT algorithms. Although the numerical difference is marginal, the two estimates are visually quite different. The AS B-DCT estimate has visible artifacts in the vicinity of sharp edges, and especially along the diagonal ones; the Pointwise SA-DCT estimate presents instead clean-cut edges that are comparable with those of the original image (shown in Figure 3.3(top-left)), thanks to superior spatial adaptivity.

Finally, we compare our Pointwise SA-DCT against the TLS algorithm [78], which is developed to deal with observations of the form $z = y + (K_0 + K_1 y)\,\eta$, $K_0, K_1 \in \mathbb{R}^+$ (i.e., one of the above models, with fixed $\alpha = 1$). In Tables 3.3 and 3.4 we present PSNR and SSIM [172] results for $K_0 = 25$, $K_1 = 0.1$ and for $K_0 = 25$, $K_1 = 0.2$, respectively. The Pointwise SA-DCT achieves the best results on all images, with the notable exception of *Barbara*, to which apply the same considerations made in Section 1.8.4. Figure 3.4 provides an illustration of the visual quality of the estimates. In the figure one can see that the Pointwise SA-DCT estimates are sharper and with fewer visible artifacts.

	Lena 512x512	*Boats* 512x512	*Barbara* 512x512	*House* 256x256	*Peppers* 256x256
Pointwise SA-DCT	**30.04** *0.823*	**27.66** ***0.731***	26.79 *0.798*	**29.99** ***0.816***	**28.08** ***0.824***
TLS (Hirakawa & Parks) [78]	**30.04** ***0.824***	27.40 *0.721*	**27.77** ***0.811***	29.64 *0.808*	27.17 *0.798*

Table 3.3: PSNR (dB, in regular type) and SSIM index [172] (in italic type) comparison table for the denoising of images with signal-dependent noise, where the observations have the form $z = y + (25 + 0.1y)\eta$, $\eta(\cdot) \sim \mathcal{N}(0, 1)$.

	Lena 512x512	*Boats* 512x512	*Barbara* 512x512	*House* 256x256	*Peppers* 256x256
Pointwise SA-DCT	**28.89** ***0.801***	**26.36** ***0.684***	25.61 *0.750*	**28.50** ***0.793***	**26.72** ***0.794***
TLS (Hirakawa & Parks) [78]	28.82 *0.797*	26.22 *0.679*	**26.28** ***0.762***	28.17 *0.781*	25.77 *0.764*

Table 3.4: PSNR (dB, in regular type) and SSIM index [172] (in italic type) comparison table for the denoising of images with signal-dependent noise, where the observations have the form $z = y + (25 + 0.2y)\eta$, $\eta(\cdot) \sim \mathcal{N}(0, 1)$.

Illustrative examples of denoising raw-data of digital imaging sensors are presented at the end of this thesis.

We remark that the proposed method based on locally adaptive variance estimates is superior to the simpler approach where the noisy observations are preprocessed by a non-linear variance-stabilizing transformation (see, e.g., [139]) and then filtered by an algorithm for AWGN. In particular, let us consider the Poissonian noise case and the procedure where the Anscombe transformation [5] is employed to stabilize the variance and denoising is then performed by the Pointwise SA-DCT algorithm [58] for AWGN (presented in Section 1.5), followed by inverse Anscombe transformation. For this simpler approach the MSE results are as follows: $\chi = 0.1$ *Peppers* MSE=103, *Aerial* MSE=153; $\chi = 30/255$ *Lena* MSE=59, *Cameraman* MSE=105. A comparison with the Tables 3.1 and 3.2 shows the improvement achieved by the use of locally adaptive variances.

3.5 Deblurring of images corrupted by signal-dependent noise

As a final remark, we note that the denoising algorithm presented in this chapter can be combined with the deconvolution approach from Chapter 2, resulting in a deblurring algorithm for images corrupted by signal-dependent noise. We refer the reader to our publication [49], where we present a similar procedure based on the block-DCT with adaptive block-size.

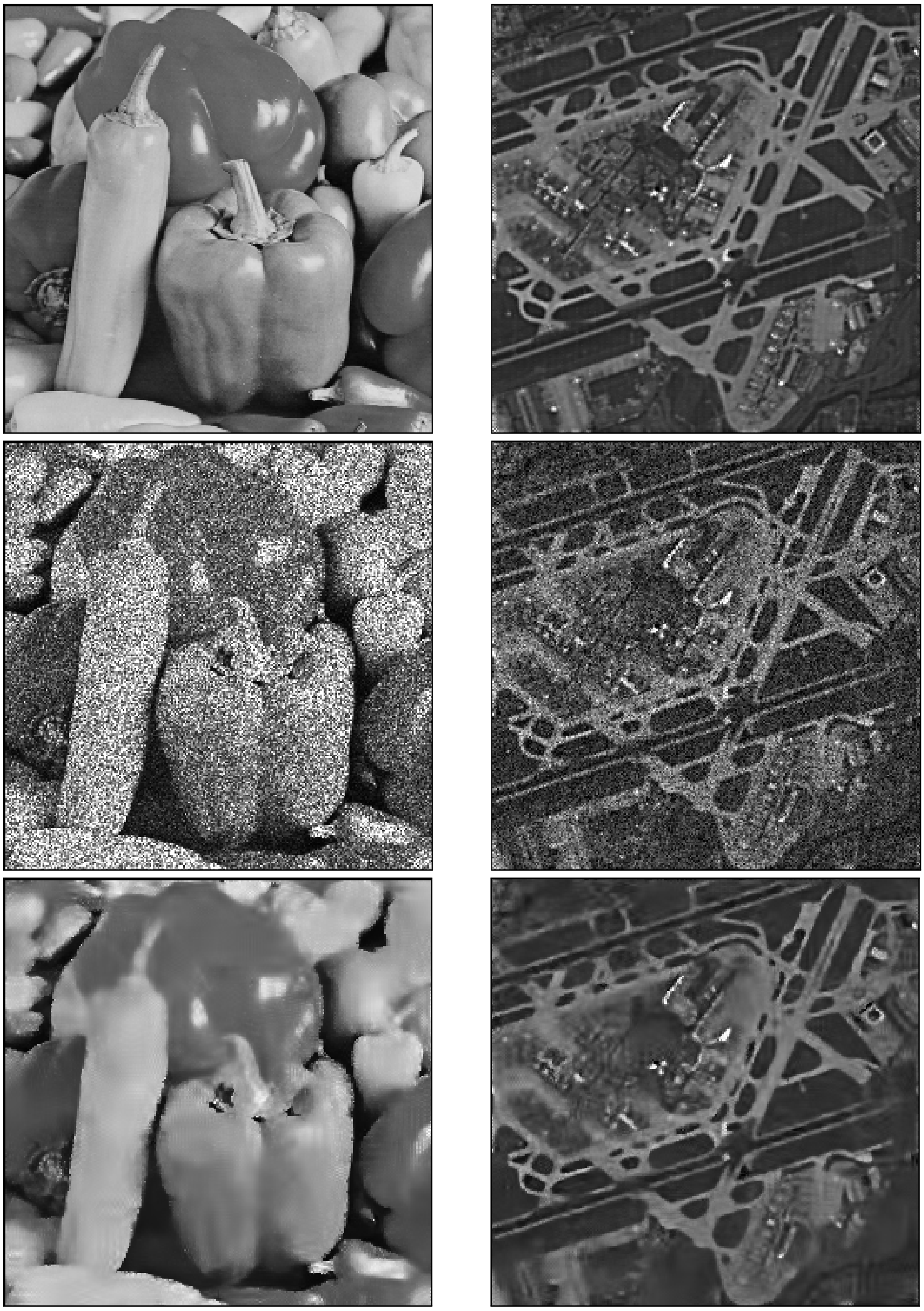


Figure 3.2: *Peppers* image (left), from top to bottom: original image, noisy observation corrupted by speckle noise (MSE=4442, PSNR=11.6dB), and Pointwise Shape-Adaptive DCT estimate $\hat{y}^{\mathrm{wi}}$ (MSE=193, PSNR=25.3dB). *Aerial* image (right), from top to bottom: original image, noisy observation corrupted by film-grain noise (MSE=829, PSNR=18.9dB), and Pointwise Shape-Adaptive DCT estimate $\hat{y}^{\mathrm{wi}}$ (MSE=142, PSNR=26.6dB).

Figure 3.3: *Cameraman* image. Left column, from top to bottom: original image, noisy observation corrupted by Poissonian noise ($\chi = 30/255$, MSE=1015, PSNR=18.1dB), and Pointwise Shape-Adaptive DCT estimate $\hat{y}^{\mathrm{wi}}$ (MSE=100, PSNR=28.1dB). Right column, middle: noisy observation corrupted by Poissonian noise ($\chi = 60/255$, MSE=504, PSNR=18.1dB); top: AS B-DCT estimate [52] (MSE=70, PSNR=29.7dB); bottom: Pointwise Shape-Adaptive DCT estimate $\hat{y}^{\mathrm{wi}}$ (MSE=68, PSNR=29.8dB).

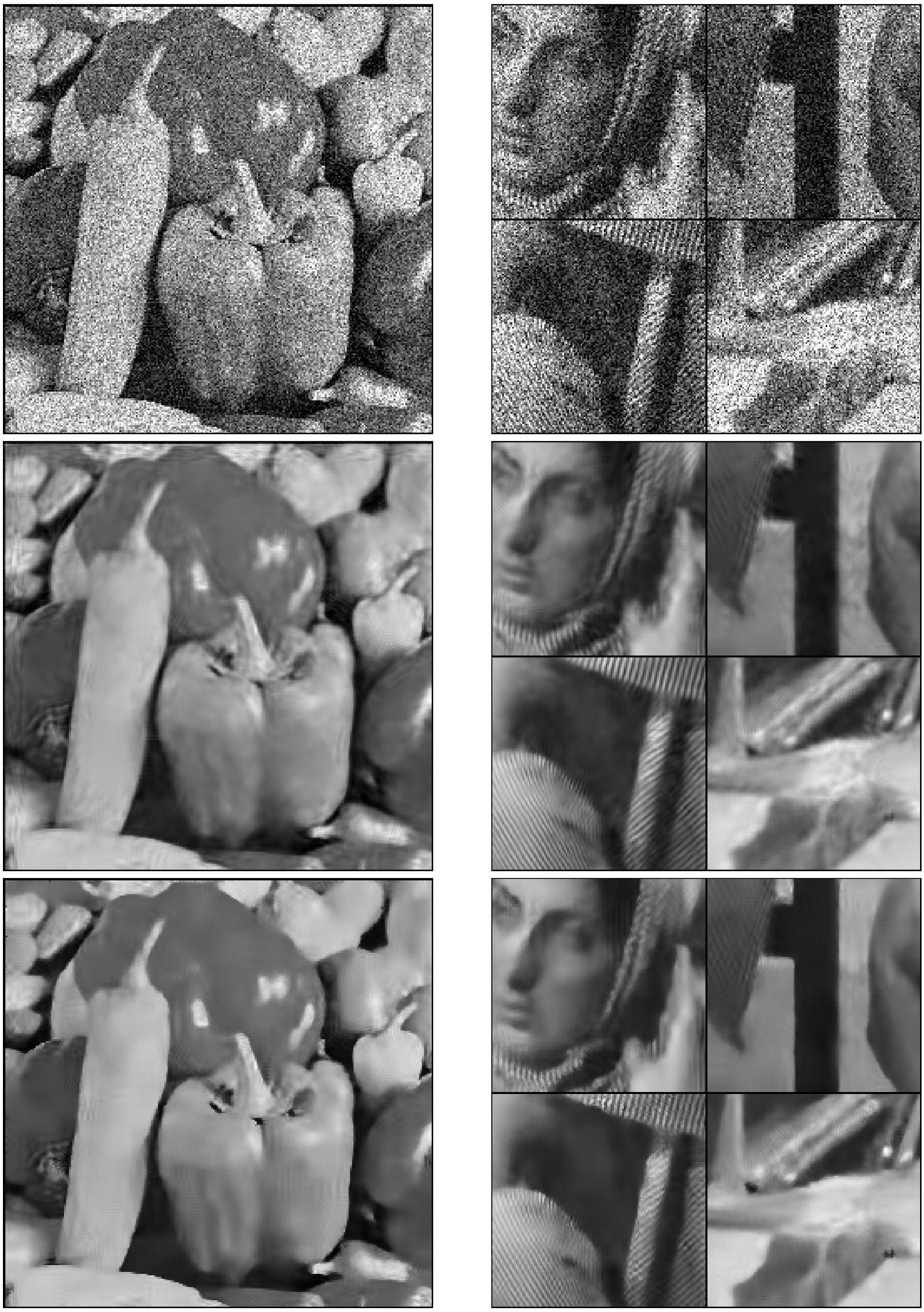


Figure 3.4: Top: noisy *Peppers* image, $z = y+(25+0.1y)\eta$, and noisy *Barbara* (fragments shown), $z = y + (25 + 0.2y)\eta$, $\eta(\cdot) \sim \mathcal{N}(0, 1)$. Center: TLS [78] estimates. Bottom: Pointwise SA-DCT estimates. The corresponding PSNR and SSIM results are given in Tables 3.3 and 3.4.

Chapter 4

“Shape-adapted” bases by orthonormalization

The shape-adaptive DCT proposed by Sikora et al. can be implemented very efficiently with a negligible computational overhead with respect to the usual separable 2-D DCT on a rectangular support. However, for video-coding applications Gilge's “shape-adapted” DCT [70, 71] based on orthonormalization with respect to shape is known to achieve superior results than the approach by Sikora et al. It is considered [155] as a kind of upper bound on the decorrelation ability of the faster method. Thus, if computational complexity is not of primary concern, Gilge's method may be utilized as an alternative method to achieve further improvement in denoising.

4.1 Construction of a “shape-adapted” basis

We remark that the implementation of such “shape-adapted” bases is not trivial. If general enough shapes are allowed[1], precalculation of all possible “shape-adapted” bases is not feasible because the resulting look-up-table would be huge. Hence, the orthogonalization must be performed “on-line”. There are two more problems. First, when restricting a basis to a region smaller than the initial domain of its elements, linear independence is certainly lost. This makes orthonormalization ill-posed. Second, the resulting “shape-adapted” orthonormal basis significantly depends on the order in which the initial elements are orthonormalized. In particular, since an increasing number of orthogonal complements are successively subtracted during the orthogonalization, the similarity (and thus the common decorrelating features) between the resulting orthonormal elements and the initial ones diminishes as the procedure progresses. The combination of these two problems can have a devastating impact, because the attempt to orthonormalize any linearly-dependent element results in a systematic introduction of amplified

[1] In our implementation – for example – despite the rather simplified neighborhood structure (based on $|H| = 6$ scales and $K = 8$ directions, as described in Figure 1.7), a total of $6^8 = 1679616$ different shapes are possible.

rounding-errors in all subsequent elements, which are thus distorted. A rather profound study of these problems, for the case of cosine and polynomial bases, is given in [131], providing criteria for the pruning of the starting basis so to obtain a set of elements which are linearly independent on the smaller region. Unfortunately, such criteria do not fully apply to arbitrary-shaped regions and additional numerical tests for independence must be employed.

We implement "shape-adapted" polynomial and cosine bases relying on the Gram-Schmidt QR-decomposition to perform the orthonormalization. It means that the starting rectangular basis is represented as a matrix A whose column vectors are the individual elements (vectorized, e.g., by raster scan of the support) and Q and R are, respectively, an orthogonal matrix and an upper-triangular matrix, such that $QR = A$. Depending on the particular starting basis, there exist faster (but equivalent) orthonormalization procedures than Gram-Schmidt (e.g., for polynomials see [132]). In order to overcome the difficulties discussed above, we use the criteria from [131] to prune the starting rectangular basis, sorted according to a "zig-zag" order, as shown in Figure 4.1. We note that our "zig-zag" order does not coincide with the natural order used in [132],[70]. Additionally, for computational efficiency and to make sure that no distorted elements are eventually used for the "shape-adapted" basis, we retain only the first $N_x \leq |\tilde{U}_x^+|$ columns of the orthogonal Q matrix for which the magnitude of the corresponding diagonal entry of the upper-triangular matrix R exceeds a predefined threshold. If $N_x < |\tilde{U}_x^+|$, then the resulting set of generators is not complete. However, from the practical point of view, it is still rich-enough to represent most considered signals which – thanks to the LPA-ICI-driven adaptive support – enjoy a particularly sparse decomposition.

Let us remark that since the same "shape-adapted" basis is used for processing all neighborhoods which share the same shape, the above orthonormalization procedure needs to be performed only once for each unique shape. We process neighborhoods with identical shape one after another, simply by following the lexicographic order induced by the directional adaptive scales $h^+(x, \theta_k)$ which define $\tilde{U}_x^+$. Therefore only a single "shape-adapted" basis needs to be kept in memory. Since the number of distinct shapes found in a image is usually much less than the total number of pixels, the computational drawback of Gilge's against Sikora's method is mostly due to the fact that the former transform is performed as multiplication against a $N_x \times |\tilde{U}_x^+|$ matrix, whereas the latter uses transformation by rows and by columns, just like a separable transform.

Illustrations of bases obtained by the above orthogonalization on an arbitrarily shaped domain are given in Appendix B, where these bases are also compared with those corresponding to the shape-adaptive DCT.

4.2 Modifications to the algorithms

The filtering algorithm described in Sections 1.5.2-1.5.4 is valid also if Gilge's "shape-adapted" DCT or "shape-adapted" polynomials are used instead of the SA-DCT. However, since these "shape-adapted" bases have indeed a DC term which truly corresponds to a constant basis element, there is no mean-weighting defect (as described in Section 1.4.2) and the subtraction of the mean is not nec-

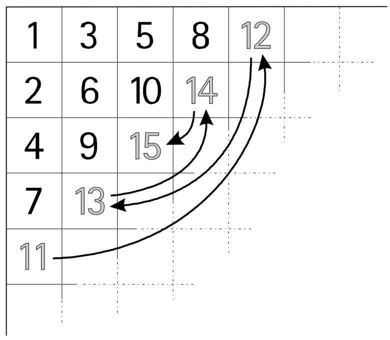


Figure 4.1: Illustration of the particular "zig-zag" ordering of the basis elements used for the Gram-Schmidt orthonormalization procedure.

essary. It means that for Gilge's "shape-adapted" DCT, formulas (1.4-1.17) can take a simpler form, where the mean terms $m_{\tilde{U}_x^+}(z)$ and $m_{\tilde{U}_x^+}(\hat{y})$ are replaced by zeros. Analogous simple modifications apply also to the deconvolution algorithm of Chapter 2.

4.3 Experiments

4.3.1 Denoising experiments

In Table 4.1 we provide PSNR results obtained using the "shape-adapted" DCT instead of the shape-adaptive DCT in the Pointwise SA-DCT denoising algorithm, applied on grayscale images corrupted by additive Gaussian white noise. This table can be directly compared with Table 1.2, which presents the PSNR results for the shape-adaptive DCT. In Table 4.2 we give the gain in PSNR achieved by using the "shape-adapted" DCT. The differences are small, typically about 0.1 dB in favour of the "shape-adapted" DCT. In Figure 4.2 we show fragments of three estimates from the Table 4.1. From a comparison with Figure 1.20, it can be seen that the visual quality achieved by the two approaches is however practically the same. A similar conclusion can be drawn from comparing the estimates of Lena shown in Figures 4.3 and 1.10. Obviously, the increased complexity of the "shape-adapted" approach cannot be justified by such a marginal quality gain. Moreover, for highly textured images, such as *Barbara*, the PSNR gain is even negative, i.e. the SA-DCT outperforms the "shape-adapted" DCT.

Table 4.3 gives PSNR results for the denoising of a few standard images. We consider the basic and recursive Anisotropic LPA-ICI [91, 55, 47, 92] and three realizations of the presented approach using three different transforms for arbitrarily-shaped supports: "shape-adapted" orthonormal polynomials and cosines bases (obtained by successive orthogonalization), and the low-complexity shape-adaptive DCT (SA-DCT). The "shape-adapted" DCT provides the best results in all experiments, thus demonstrating a slightly better approximation ability for natural images than the "shape-adapted" orthonormal polynomials. The low-complexity

σ	*Lena* 512x512	*Boats* 512x512	*House* 256x256	*Peppers* 256x256	*C'man* 256x256	*Barbara* 512x512	*Hill* 512x512	*Lake* 512x512
5	38.62	37.17	39.45	38.14	38.26	37.57	37.04	36.62
10	35.71	33.69	36.09	34.64	34.12	33.50	33.44	32.86
15	34.0	31.85	34.26	32.62	31.84	31.30	31.60	31.06
20	32.75	30.54	33.01	31.19	30.32	29.84	30.40	29.82
25	31.76	29.52	32.01	30.07	29.22	28.75	29.51	28.87
30	30.95	28.68	31.15	29.16	28.33	27.87	28.81	28.10
35	30.25	27.98	30.42	28.41	27.59	27.10	28.24	27.44
50	28.69	26.36	28.78	26.75	25.94	24.94	26.93	25.94
75	26.78	24.70	26.64	24.71	23.92	23.30	25.50	24.26
100	25.44	23.58	25.08	23.30	22.53	22.38	24.49	23.06

Table 4.1: Grayscale image denoising performance as PSNR (dB) for the Pointwise SA-DCT algorithm, using the “shape-adapted” DCT instead of the low-complexity shape-adaptive DCT.

σ	*Lena* 512x512	*Boats* 512x512	*House* 256x256	*Peppers* 256x256	*C'man* 256x256	*Barbara* 512x512	*Hill* 512x512	*Lake* 512x512
5	0.08	0.03	0.07	0.15	0.11	0.10	0.01	0.01
10	0.13	0.06	0.11	0.18	0.14	0.02	0.01	0.05
15	0.14	0.06	0.12	0.18	0.14	-0.07	0.00	0.08
20	0.13	0.05	0.09	0.15	0.14	-0.15	0.01	0.08
25	0.10	0.05	0.09	0.15	0.11	-0.19	0.01	0.08
30	0.09	0.04	0.05	0.13	0.09	-0.22	0.01	0.09
35	0.08	0.05	0.03	0.15	0.08	-0.25	0.02	0.09
50	0.09	0.10	0.11	0.20	0.06	-0.50	0.08	0.11
75	0.02	0.08	0.06	0.14	0.03	-0.21	0.06	0.12
100	-0.06	0.06	0.01	0.15	0.04	-0.11	0.02	0.10

Table 4.2: PSNR gain (dB, ± 0.01) obtained using the “shape-adapted” DCT instead of the low-complexity shape-adaptive DCT in the Pointwise SA-DCT algorithm.

Figure 4.2: Fragments of the estimates of the *Boats*, *Peppers*, and *House* images, σ=25, obtained using the “shape-adapted” DCT instead of the low-complexity shape-adaptive DCT (SA-DCT). Compare with Figure 1.20.

Figure 4.3: A fragment of the estimates of *Lena* (from the noisy observation with σ=25, shown in Figure 1.10) obtained using (left) the low-complexity SA-DCT (PSNR=31.66dB) and (right) using the "shape-adapted" DCT (PSNR=31.76dB). There are almost no visible differences between the two estimates.

SA-DCT achieves a performance which is very close to that of the more complex transforms, but at a much lower computational cost.

Table 4.4 shows results obtained by implementing the proposed approach with the second stage, with empirical Wiener filtering. For this Wiener filtering the reference estimate is the one obtained from the first stage (whose PSNR is given in Table 4.3). The two-stage implementation significantly improves the denoising performance for all three transforms. However, the differences between the estimates obtained using different transform are still small. Thus, we conclude that, in terms of quality of results, the differences between orthonormal polynomials, "shape-adapted" DCT, and SA-DCT is not significant.

4.3.2 Deblurring experiments

We consider the same four blurred experiments of Table 2.1. The improvement-in-SNR (ISNR) results achieved with Gilge's "shape-adapted" DCT for these four experiments are given in Table 4.5, where for convenience we also report the results obtained with the SA-DCT (Table 2.2). From the table we can see that in terms of ISNR the differences (which are in favour of the "shape-adapted" DCT) are negligible.

The four estimates are shown in Figure 4.4. From a visual comparison between this figure and Figure 2.8 we can concluded that the are no relevant visual differences between the "SA"-DCT estimates and the SA-DCT estimates. All the estimates are visually very good, with accurate preservation of details and no disturbing artifacts.

Method (one-stage, hard-thresholding only)	*C'man*	*Peppers*	*Lena*	*Boats*
Anisotropic LPA-ICI (zero order)	28.16	28.34	29.67	27.92
Recursive Anis. LPA-ICI (zero-first order mixture)	28.68	29.16	30.43	28.58
P.wise “shape-adapted” orthon. polynomials	28.89	29.61	31.34	29.08
P.wise “shape-adapted” DCT	**28.99**	**29.69**	**31.43**	**29.20**
P.wise shape-adaptive DCT (SA-DCT)	28.90	29.52	31.27	29.09

Table 4.3: PSNR (dB) results for denoising from additive Gaussian white noise (σ=25) for a few test images with different algorithms for anisotropic nonparametric estimation.

Method (two-stage, with empirical Wiener filtering)	*C'man*	*Peppers*	*Lena*	*Boats*
P.wise “shape-adapted” orthon. polynomials	29.13	30.03	31.68	29.43
P.wise “shape-adapted” DCT	**29.19**	**30.05**	**31.76**	**29.51**
P.wise shape-adaptive DCT (SA-DCT)	29.11	29.92	31.66	29.47

Table 4.4: PSNR (dB) results for denoising with transform-domain empirical Wiener filtering. The reference estimate used in Wiener filtering is the one obtained by the respective algorithm, as shown in Table 4.3. The estimates obtained by this two-stage procedure are noticeably better than those obtained by just a single stage.

Method Experiment	1	2	3	4
Anis. LPA-ICI RI-RWI + Gilge's “SA”-DCT	**8.58**	**8.29**	**6.34**	**4.55**
Anis. LPA-ICI RI-RWI + SA-DCT (i.e. P. SA-DCT)	8.57	8.25	6.34	4.52

Table 4.5: ISNR (dB) of the proposed Pointwise shape-adaptive DCT regularized deconvolution algorithm using either Gilge's “shape-adapted” DCT or the low-complexity SA-DCT. The four deblurring experiments are described in Table 2.1, Section 2.8.

Figure 4.4: Fragments of the estimates $\hat{y}^{RWI}$ obtained using the “shape-adapted” DCT for the four deblurring experiments from Table 2.1, Section 2.8. Top: experiments 1 and 2; bottom: experiments 3 and 4 (ISNR = 8.58, 8.29, 6.34, and 4.55 dB, respectively). Compare with the corresponding SA-DCT estimates shown in Figure 2.8.

Let us remark that because of the need to go through the explicit expression of each one of the SA-DCT basis elements $\psi^{(i)}_{\tilde{U}^+_x}$, $i \in V_{\tilde{U}^+_x}$, the computational advantage of Sikora's SA-DCT over Gilge's "shape-adapted" DCT is substantially reduced. Moreover, when using smaller neighborhoods (for deblurring we use shorter kernels with $H = \{1, 2, 4\}$), the differences between a low-complexity transform and an arbitrary non-separable transform are anyway minor. In fact, our implementations of the proposed deblurring algorithm based on Gilge's "shape-adapted" DCT and on the lower-complexity SA-DCT perform in about the same time.

Appendix A: LPA-ICI

For the readers less familiar with the LPA and ICI techniques utilized in the previous chapters, in this appendix we give more details on these techniques. For the sake of simplicity, we restrict ourself to the 1-D case (thus dropping the subscript θ_k from notation). This simplification is not essential, because as we described in Section 1.5.1 in our implementation we use 1-D kernels supported on line segments. We refer the interested reader to [47],[92], for a complete and in-depth analysis of the LPA-ICI and related algorithms. Details of the various parameters used in the implementation of the LPA-ICI within the Pointwise Shape-Adaptive DCT algorithms can be seen in the publicly available codes at `http://www.cs.tut.fi/~foi/SA-DCT` .

A.1 Local Polynomial Approximation (LPA)

The Local Polynomial Approximation (LPA) (e.g., [42]) is a technique which is applied for nonparametric estimation using a polynomial data fit in a sliding window. The polynomial order m and the window function w characterize the LPA. The LPA estimates are calculated by convolution against a kernel $g = \mathbf{w}\boldsymbol{\phi}\boldsymbol{\Phi}^{-1}\,[1\ 0 \cdots 0]^T$, where $\mathbf{w}=$ diag w is the diagonal matrix composed by the weights w, $\boldsymbol{\phi}$ is a vector of $m+1$ polynomial functions (basis) $\phi_n = \frac{v^n}{n!}$, $n = 0, \ldots, m$, and $\boldsymbol{\Phi}= \boldsymbol{\phi}^T\mathbf{w}\boldsymbol{\phi}$ is the Gramian matrix (formed by the inner products of the basis elements against each other).

Starting from a *basic* window function w, one can obtain LPA's of different bandwidths/scales using scaled windows $w_h = w\left(\cdot/h\right)$, where $h \in \mathbb{R}^+ \setminus \{0\}$ is the *scale* parameter. The corresponding kernels are denoted as g_h. It is common practice to use compactly supported window functions. In this case, by using a basic window w of unit length, we obtain that h coincides with the length of the window w_h. Hence, window length (size), scale, and bandwidth become interchangeable concepts.

The choice of the scale parameter is crucial when dealing with noisy data, because it controls the amount of smoothing introduced by the local approximation. A large h corresponds to a larger window and therefore to smoother estimates, with lower variance and typically increased estimation bias. A small h corresponds to noisier estimates, less biased, and with higher variance. Thus, the scale parameter h controls the trade-off between bias and variance in the LPA estimates.

In practice, the use of a fixed order m can be relaxed, and polynomial order

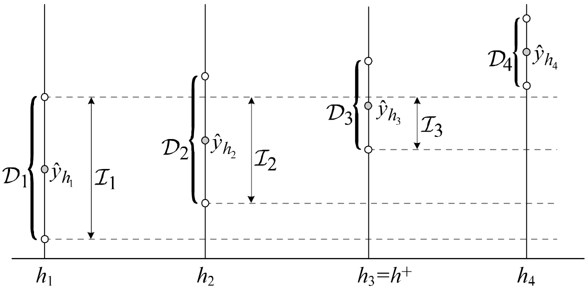


Figure A.1: The Intersection of Confidence Intervals (ICI) rule.

mixtures [22],[47] are most often used, defining LPA kernels of the form $g_h = \sum_m \alpha_{h,m} g_h^{[m]}$, where $g_h^{[m]}$ is an LPA kernel of order m and the coefficients $\alpha_{h,m} \in \mathbb{R}$ are such that $\sum_m \alpha_{h,m} = 1$. The LPA kernels adopted in the Pointwise SA-DCT algorithms and shown in Figure 1.6 are mixtures of orders zero and one with uniform windows $w_h = \chi_{\{0,1,\dots,h-1\}}$.

A.2 Intersection of Confidence Intervals (ICI) rule

The Intersection of Confidence Intervals (ICI) rule [72, 90] is a criterion used for the adaptive selection of the size (length/scale) of the LPA window. Let x be a fixed estimation point/pixel. The LPA estimates $\hat{y}_{h_j}(x) = \left(z \circledast g_{h_j}\right)(x)$ are calculated for a set $H = \{h_j\}_{j=1}^J$ of increasing scales $h_1 < \cdots < h_J$. The goal of the ICI is to select among these given estimates $\left\{\hat{y}_{h_j}(x)\right\}_{j=1}^J$ an adaptive estimate $\hat{y}_{h^+(x)}(x)$, $h^+(x) \in H$, such that $\hat{y}_{h^+(x)}(x)$ is close to an "ideal" estimate $\hat{y}_{h^*(x)}(x)$ which minimizes the MSE with respect to the variation of the scale h (note that $h^*(x)$ does not necessarily belong to H). Roughly speaking, the estimate $\hat{y}_{h^+(x)}(x)$ is the "best" among the given ones.

The ICI rule is as follows:

Consider the intersection of confidence intervals $\mathcal{I}_j = \bigcap_{i=1}^j \mathcal{D}_i$, where

$$\mathcal{D}_i = \left[\hat{y}_{h_i}(x) - \Gamma\sigma_{\hat{y}_{h_i}(x)}, \hat{y}_{h_i}(x) + \Gamma\sigma_{\hat{y}_{h_i}(x)}\right],$$

$\sigma_{\hat{y}_{h_i}(x)} = \operatorname{std}\{\hat{y}_{h_i}(x)\}$ *is the standard deviation of* $\hat{y}_{h_i}(x)$, *and* $\Gamma > 0$ *is a threshold parameter. Let* j^+ *be the largest of the indexes* j *for which* $\mathcal{I}_j$ *is non-empty,* $\mathcal{I}_{j^+} \neq \varnothing$ *and* $\mathcal{I}_{j^++1} = \varnothing$. *The adaptive scale* $h^+(x)$ *is defined as* $h^+(x) = h_{j^+}$ *and the adaptive estimate is thus* $\hat{y}_{h^+(x)}(x)$.

An illustration of the ICI is given in Figure A.1. The standard-deviations of the LPA estimates can be easily calculated from the ℓ^2-norm of the corresponding kernel as $\sigma_{\hat{y}_{h_j}(x)} = \operatorname{std}\left\{\hat{y}_{h_j}(x)\right\} = \sigma \left\|g_{h_j}\right\|_2$. Since the scales are increasing, the standard-deviations are decreasing and the confidence intervals shrink as j increases. Therefore, in the intersections we are testing estimates with progressively lower variance. The rationale behind the ICI is that the estimation bias is not too large as long as the intersections are non-empty. In practice this means that the ICI adaptively allows the maximum level of smoothing, stopping before oversmoothing begins. Asymptotically, the LPA-ICI adaptive estimator allows to get a near-optimal quality of signal recovery [72].

Appendix B: A gallery of basis elements

In this appendix we show basis elements corresponding to few different bases defined on arbitrarily shaped supports. In particular, we compare the SA-DCT basis elements with the basis elements of the "shape-adapted" DCT and "shape-adapted" polynomials. The starting generators used in the Gram-Schmidt orthonormalization which produces the "shape-adapted" bases are also shown. Besides giving a visual impression of the kind of bases used in our algorithms, these many figures have the purpose of demonstrating that these bases, despite constructed by very different procedures, enjoy a remarkable similarity. This similarity explains why the filtering results obtained using these different bases are practically the same (see Tables 4.1, 4.2, 4.3 and 4.4).

B.1 Supports

The illustrations are given for six different supports, shown in Figure B.1, each of which is a subset of the 9×9 square. The coordinates within the 9×9 square support are denoted as (n_1, n_2), $n_1, n_2 = 0, \ldots, 8$. Thus, the centre will have coordinates $(4, 4)$.

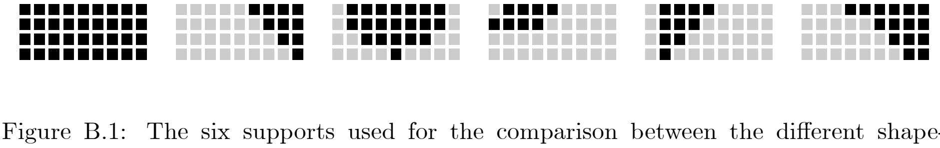


Figure B.1: The six supports used for the comparison between the different shape-adaptive and "shape-adapted" bases. All supports are subsets of a 9×9 square, on which the starting polynomial and cosine bases are defined.

B.2 Arrangement of the figures

For each shape, the figures of the basis elements are arranged as in a long horizontal table (spanning four pages), where the rows are organized as follows:

indicators (in transform domain)
polynomial generators
cosine (DCT) generators
“shape-adapted” polynomials
“shape-adapted” DCT
shape-adaptive DCT (SA-DCT)

.

In each column, the SA-DCT basis element is obtained by applying the inverse SA-DCT on the indicator. Whereas for the “shape-adapted” bases, the indicator coincides with that of the last generator used in the linear combination that yields the orthonormalized basis element. In accordance with the “zig-zag” scan (Figure 4.1 of Section 4.1), these generators are arranged column by column in the following manner (for polynomials and DCT):

1	$n_1 - 4$	$n_2 - 4$
1	$\cos\left(\frac{\pi(2n_1+1)}{18}\right)$	$\cos\left(\frac{\pi(2n_2+1)}{18}\right)$

$(n_1 - 4)^2$	$(n_2 - 4)^2$
$\cos\left(\frac{2\pi(2n_1+1)}{18}\right)$	$\cos\left(\frac{2\pi(2n_2+1)}{18}\right)$

$(n_1 - 4)(n_2 - 4)$	$(n_1 - 4)^3$
$\cos\left(\frac{\pi(2n_1+1)}{18}\right)\cos\left(\frac{\pi(2n_2+1)}{18}\right)$	$\cos\left(\frac{3\pi(2n_1+1)}{18}\right)$

$(n_2 - 4)^3$	$(n_1 - 4)^2(n_2 - 4)$	
$\cos\left(\frac{3\pi(2n_2+1)}{18}\right)$	$\cos\left(\frac{2\pi(2n_1+1)}{18}\right)\cos\left(\frac{\pi(2n_2+1)}{18}\right)$	$\dots$

In the above formulas we neglected all scaling factors (which are irrelevant since within the Gram-Schmidt procedure each generator is automatically scaled as necessary).

For the figures in this appendix, we did not exploit any special alignment of the coefficients of the SA-DCT and resorted instead to the basic alignment (see Section 1.4.3). Further, the SA-DCT is applied in non-adaptive RC mode (see Sections 1.4.4 and 1.5.1).

All the images are drawn with the same scale (given by the unit indicators on the top row). For this reason, some polynomial generators of high degree may appear as identically zero in a neighborhood of the origin, or even globally (e.g., the restricted polynomial generators on p. 103).

For each shape of the support, we show the first ten and the last two basis elements, according to the zig-zag ordering shown in Figure 4.1 (see also Section 4.1). Roughly speaking, these are the ten elements with the lowest frequency/degree and the two elements with the highest frequency/degree.

Only the last three rows present, in general, elements of orthonormal bases. However, we note that in the special case of the 9×9 square support (pp. 92-95), for which the support-restriction is trivial, the restricted DCT is itself obviously already orthonormal as it coincides with the DCT, the "shape-adapted" DCT, as well as with the shape-adaptive DCT.

B.3 Interesting sights in the gallery

We list here some important aspects portrayed by the many figures on the following pages.

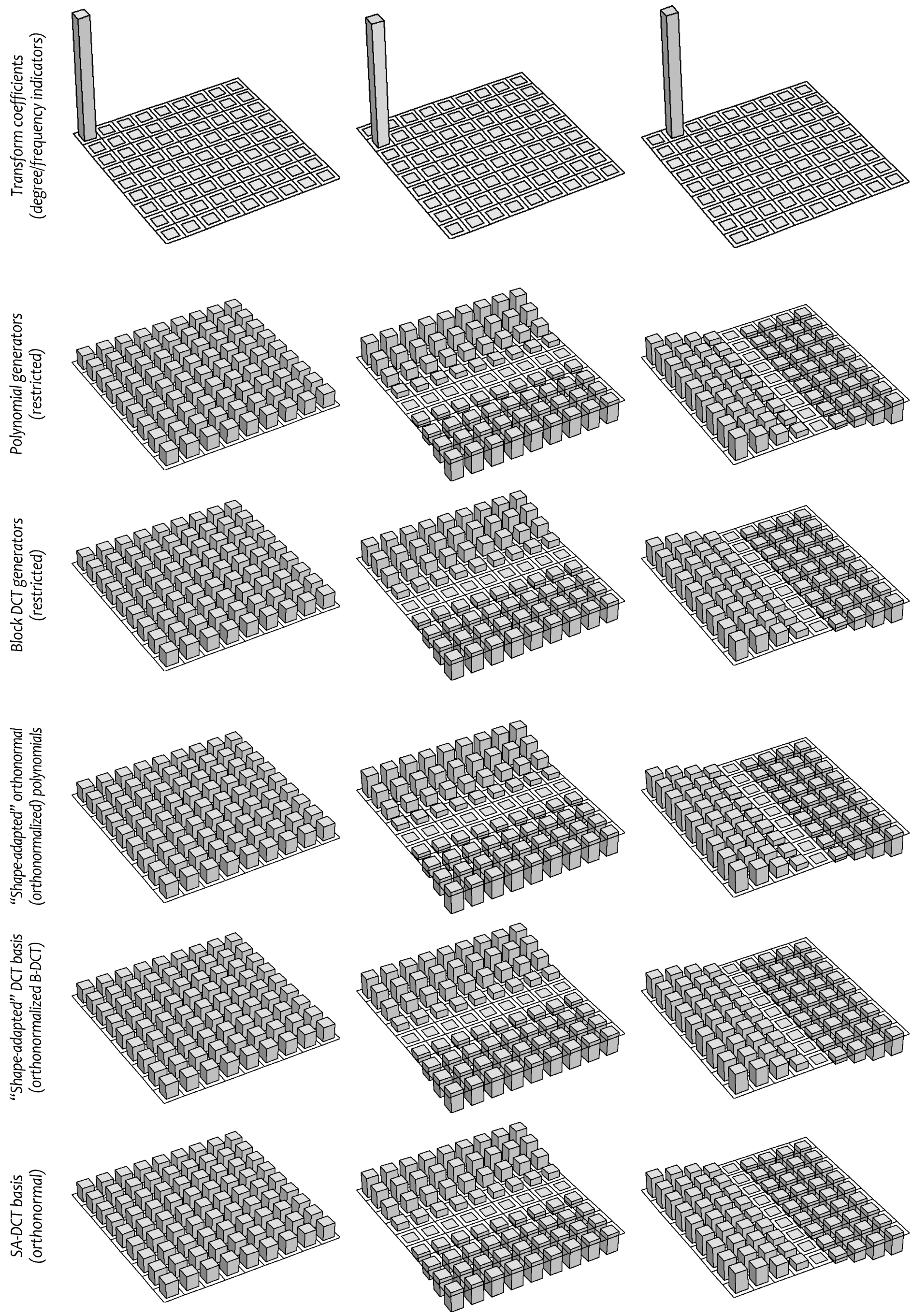
- Even though the polynomial and cosine generators are often radically different, the resultant orthonormal bases (obtained after orthonormalization) are always very similar, and sometimes almost identical, with a clear one-to-one correspondence between the basis elements. This is possible thanks to the zig-zag ordering followed in the Gram-Schmidt orthonormalization (see Section 4.1, esp. Figure 4.1).

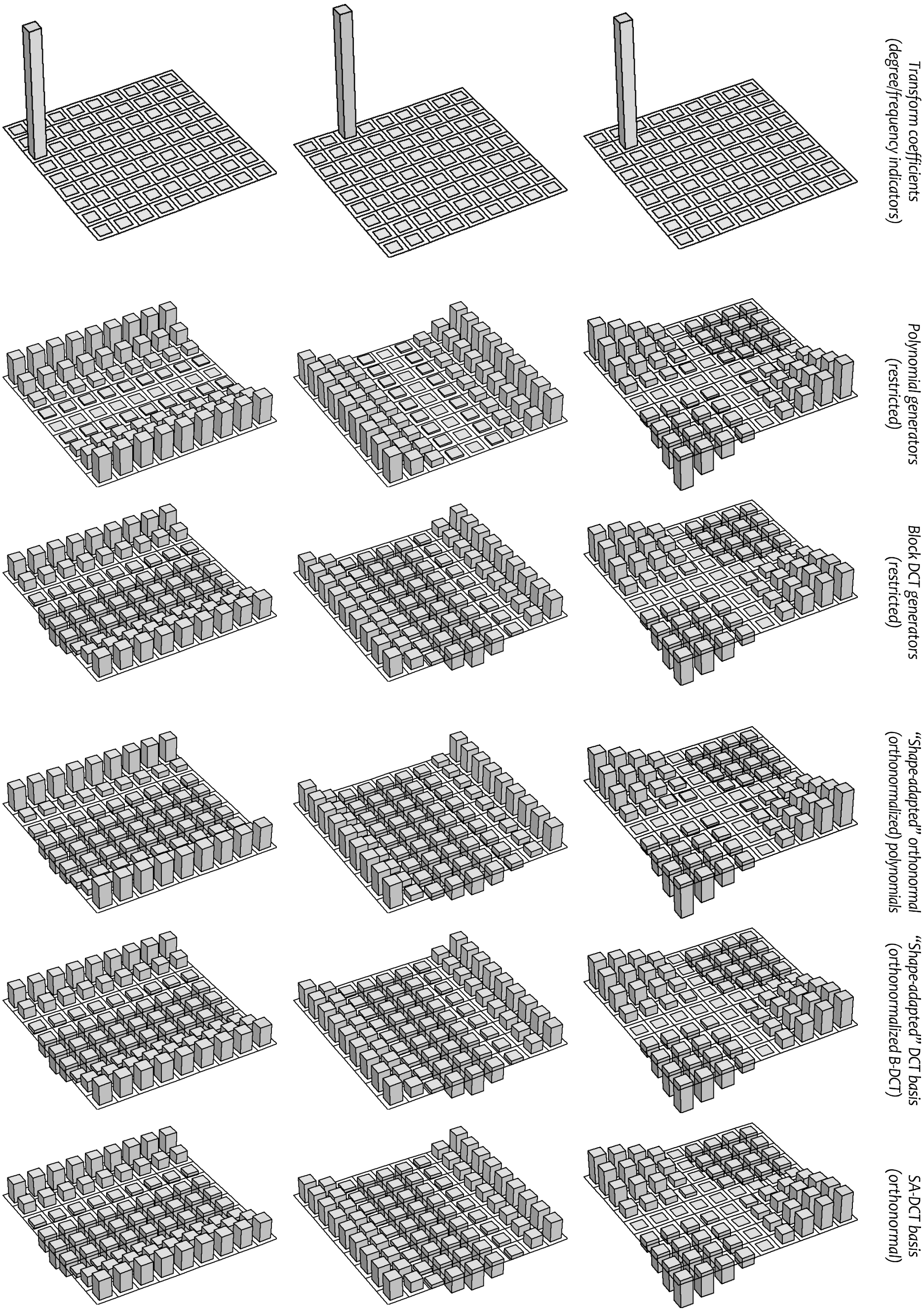
- Square support (pp. 92-95): (as anticipated above) restricted B-DCT, "SA"-DCT, and SA-DCT coincide because restriction is trivial; orthonormalized polynomials are nearly identical to DCT bases.

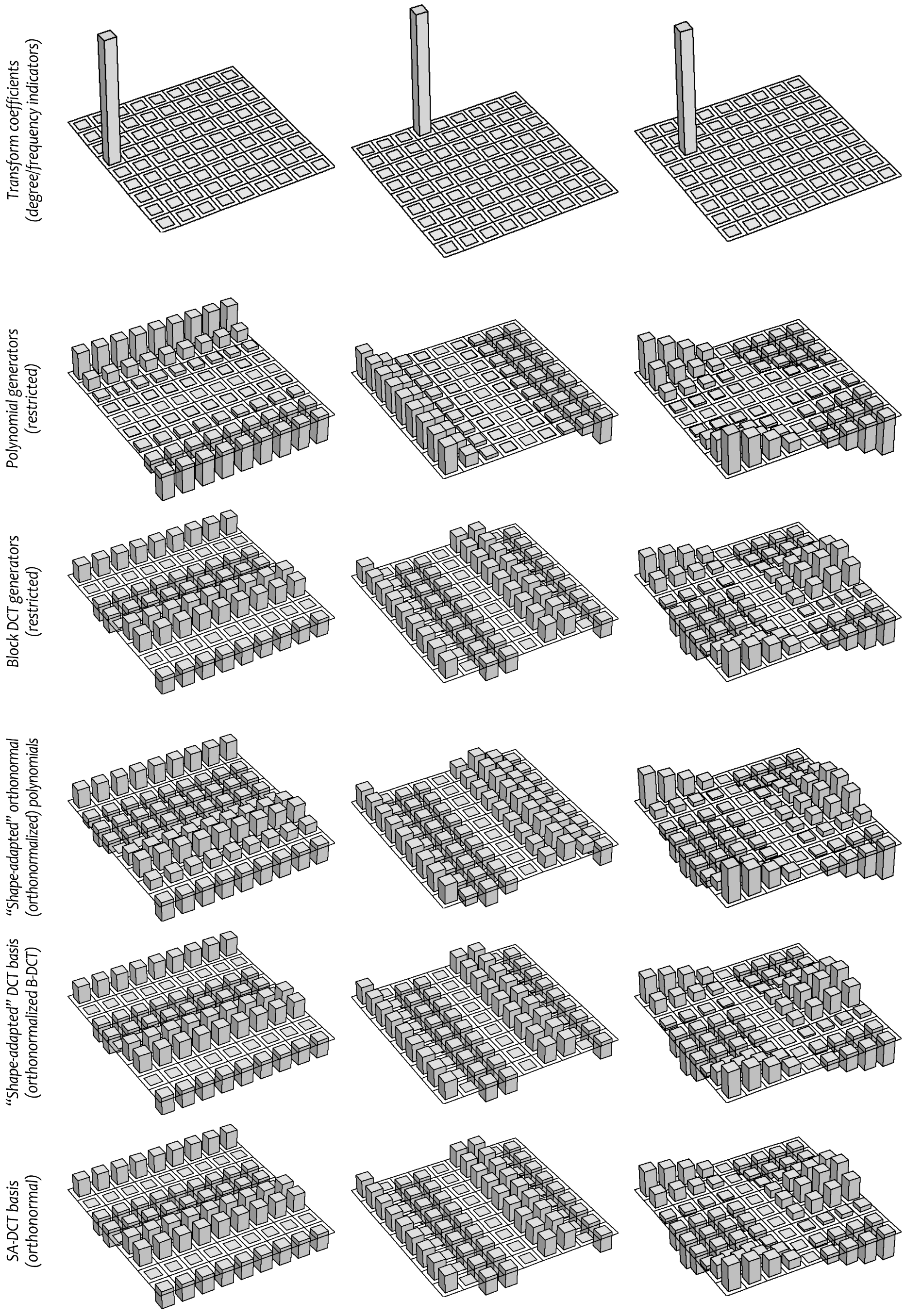
- Differences between CR and RC processing: the first parallelogram coincides with the second parallelogram transposed (as can be clearly seen in Figure B.1), but the SA-DCT basis elements obtained for the first parallelogram (pp. 104-107) are different from those obtained for the second parallelogram (pp. 108-111) as well as from their transposed copies.

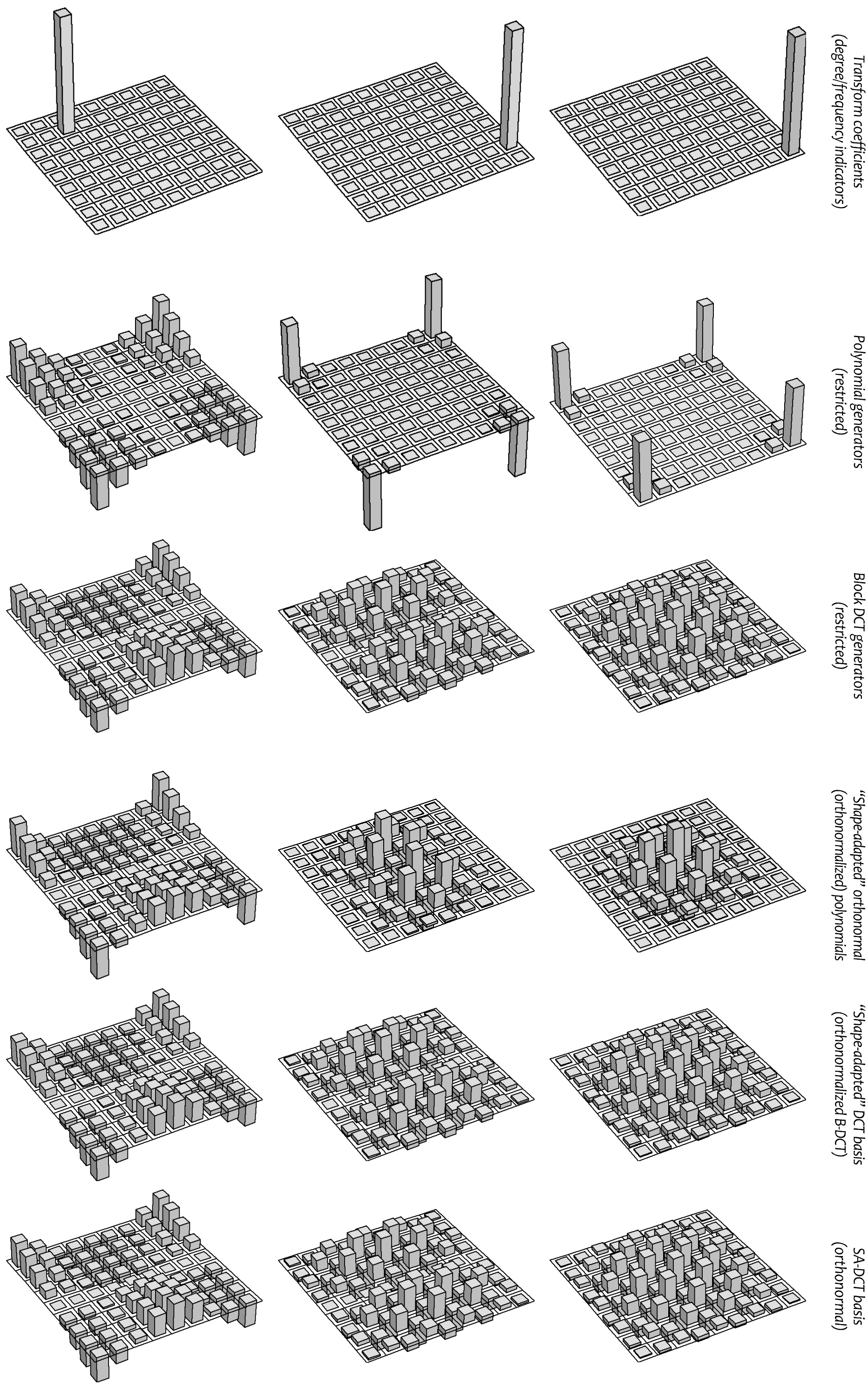
- Mean weighting defect is clearly visible in the first SA-DCT basis elements (bottom left on pp. 100, 112, 96, 108) for all supports except the square (p. 92) and the first parallelogram (p. 104). Observe also the different structure of the domain of the transform coefficients (top row).

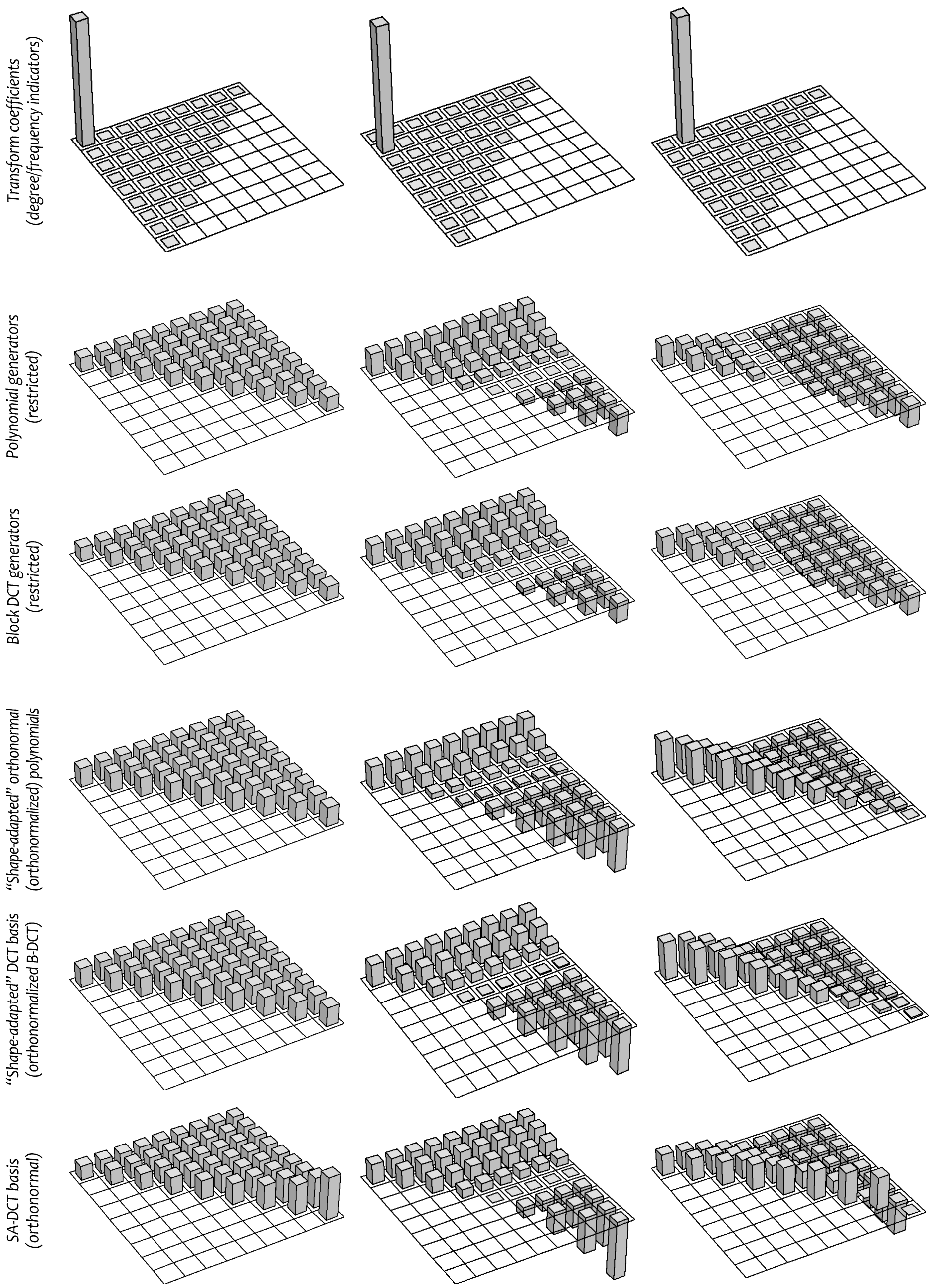
B.4 The gallery

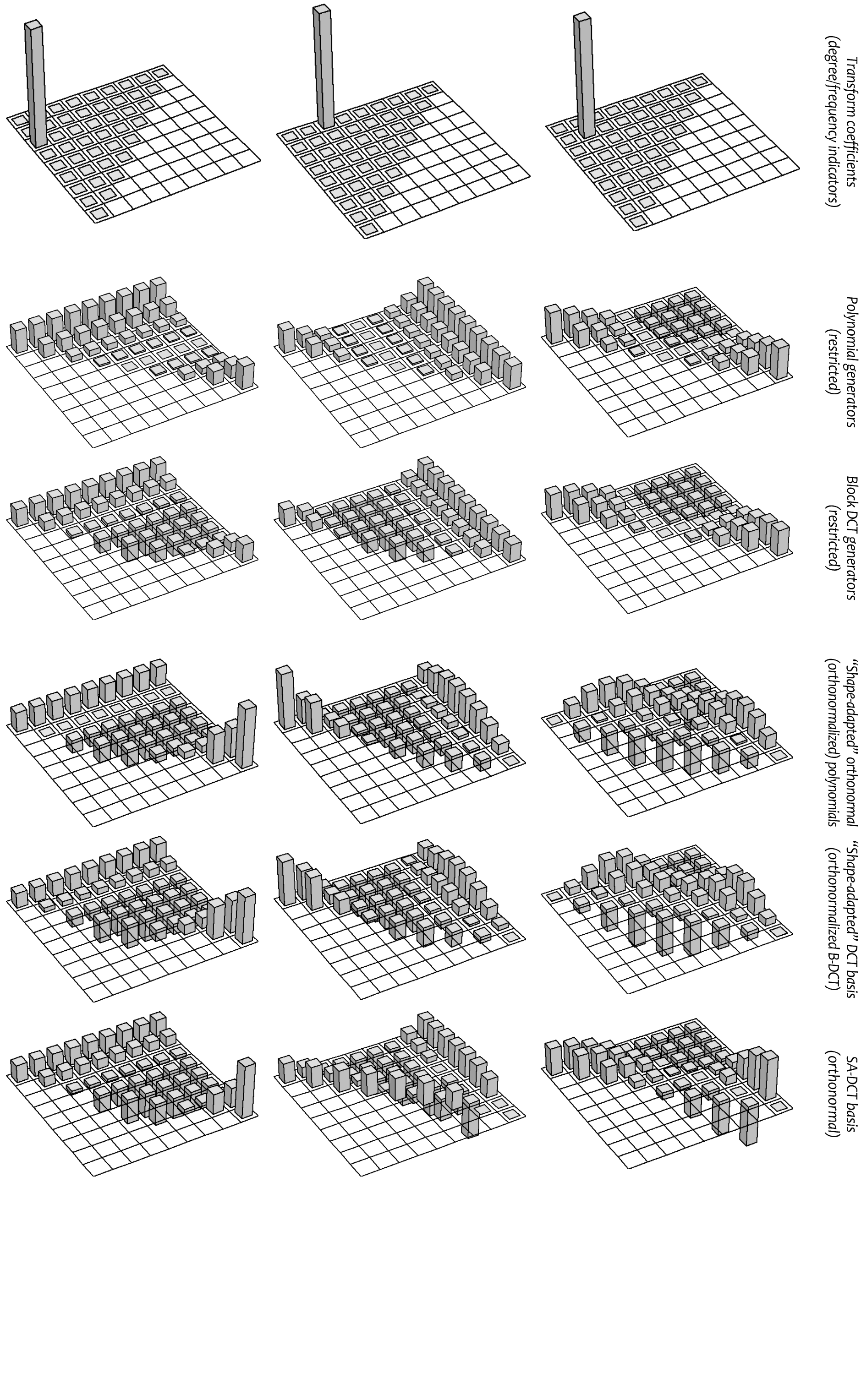


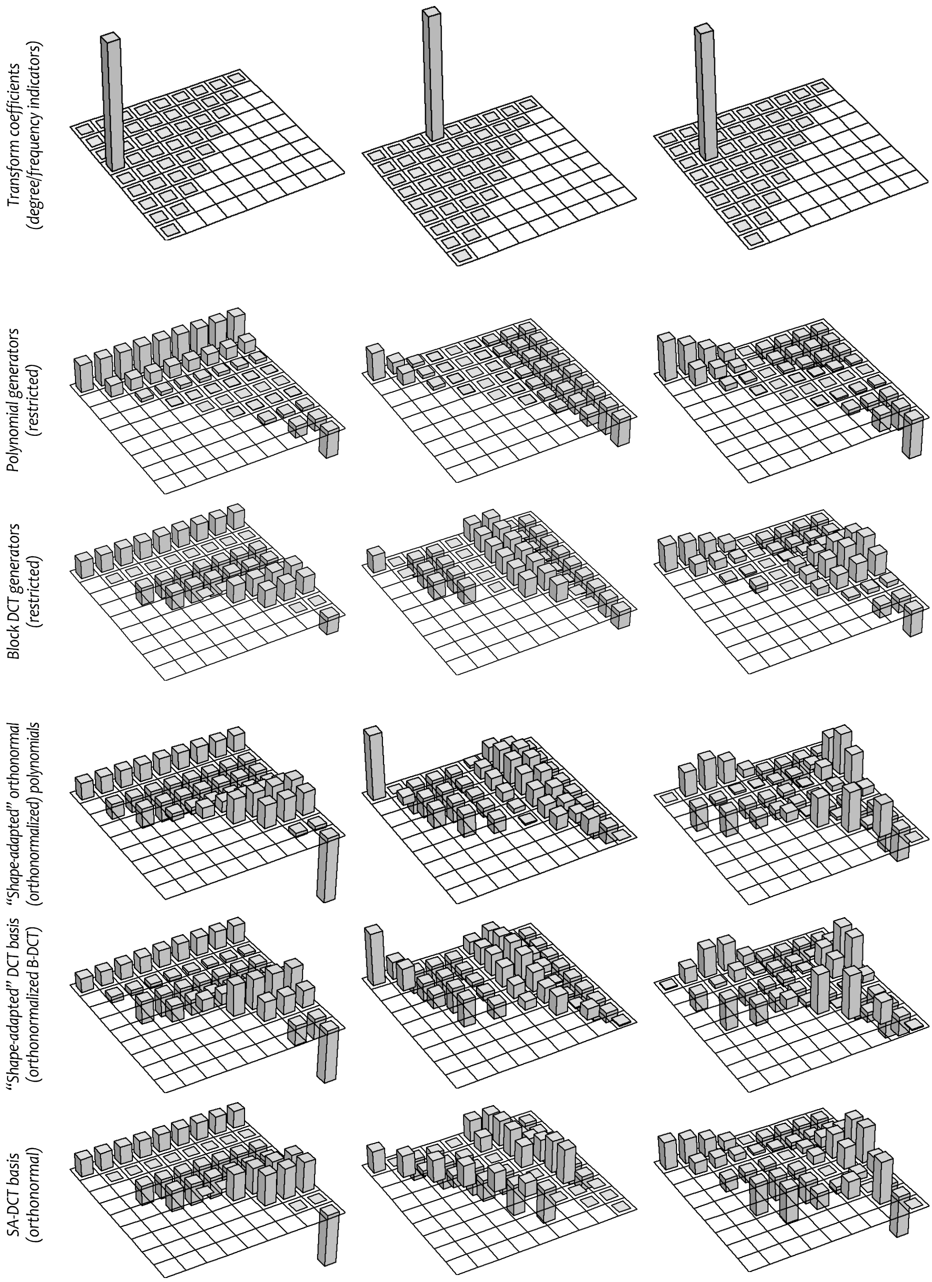

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
“Shape-adapted” orthonormal
(orthonormalized) polynomials
“Shape-adapted” DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

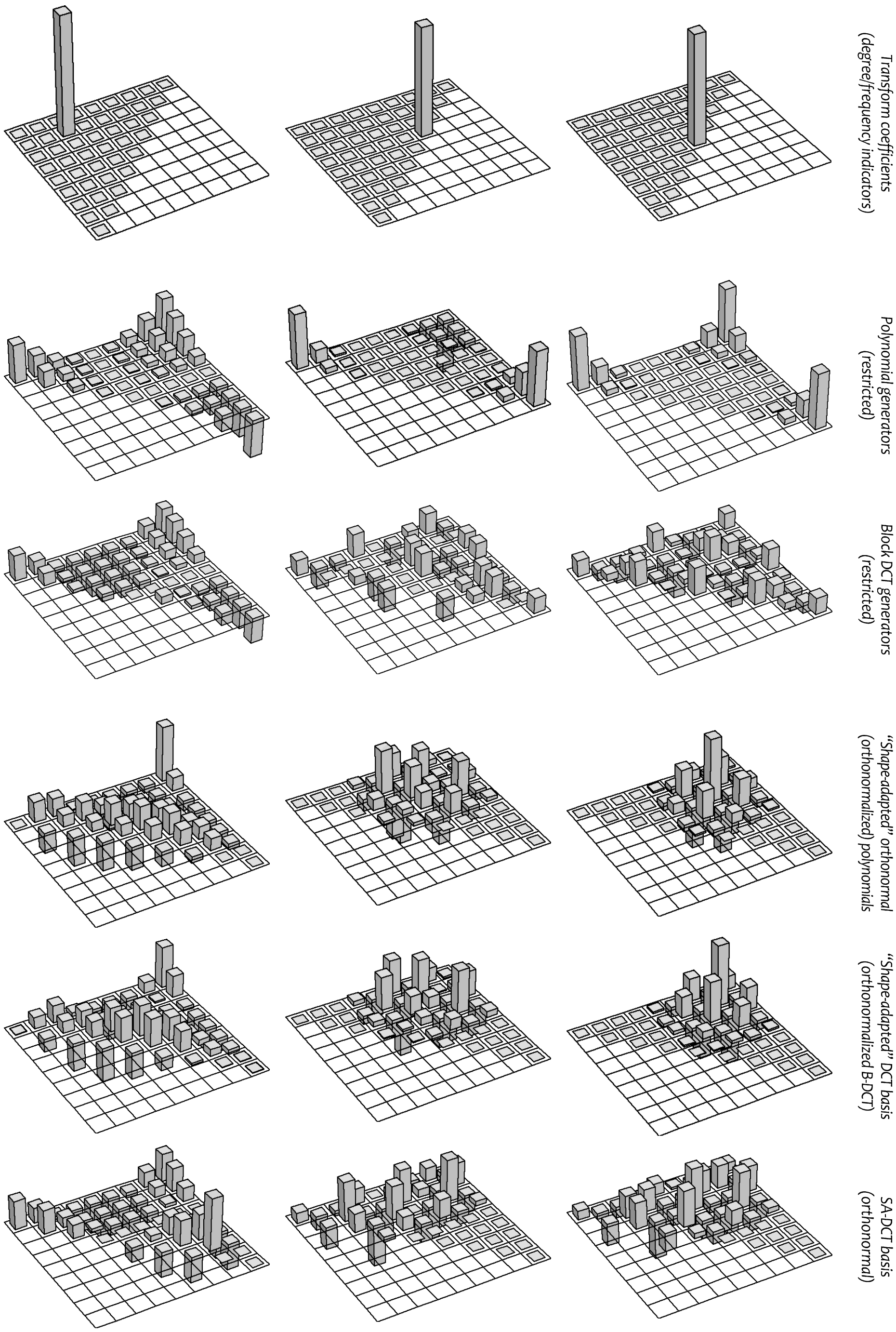

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

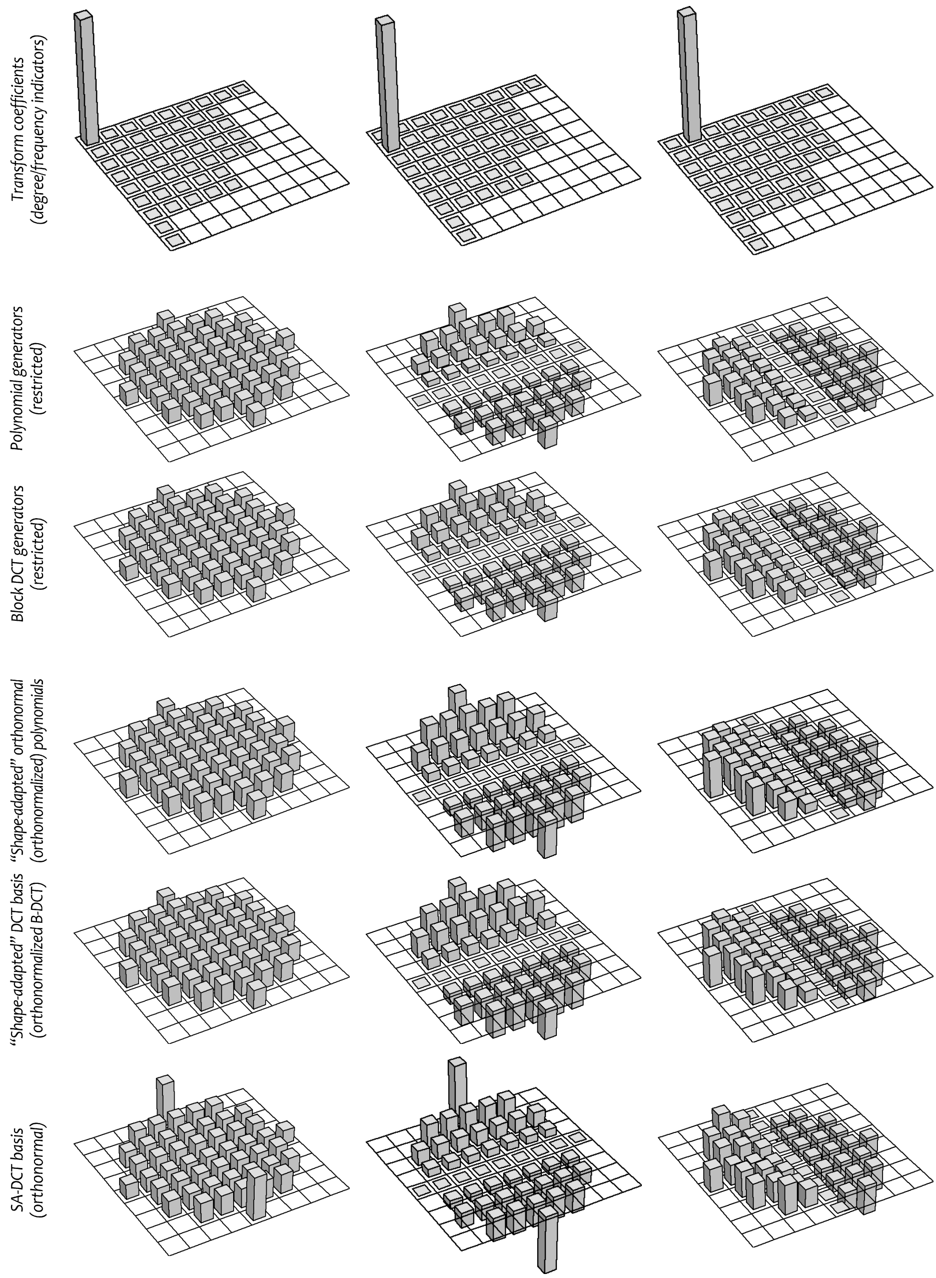

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

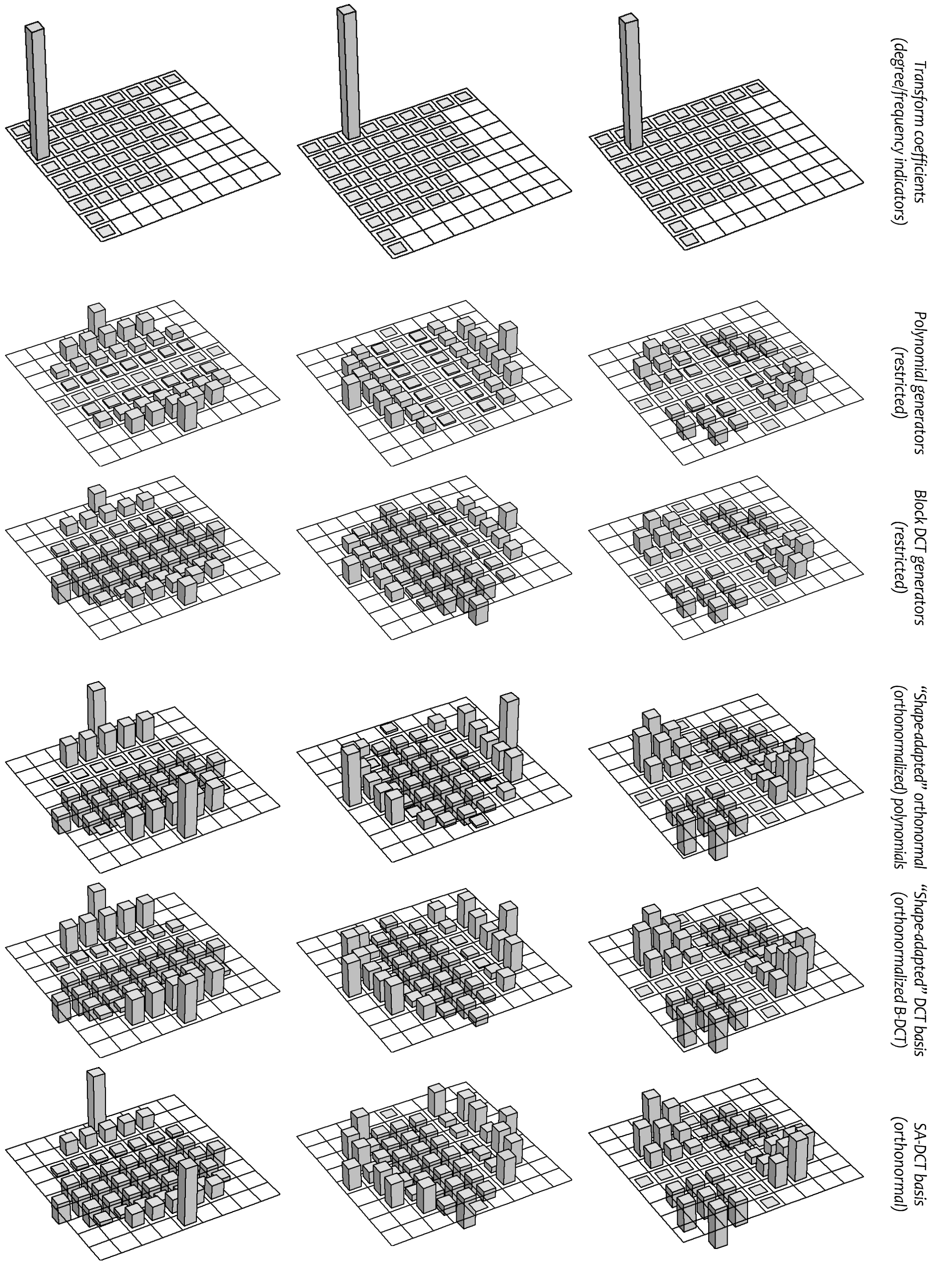

Transform coefficients (degree/frequency indicators)
Polynomial generators (restricted)
Block DCT generators (restricted)
"Shape-adapted" orthonormal (orthonormalized) polynomials
"Shape-adapted" DCT basis (orthonormalized B-DCT)
SA-DCT basis (orthonormal)

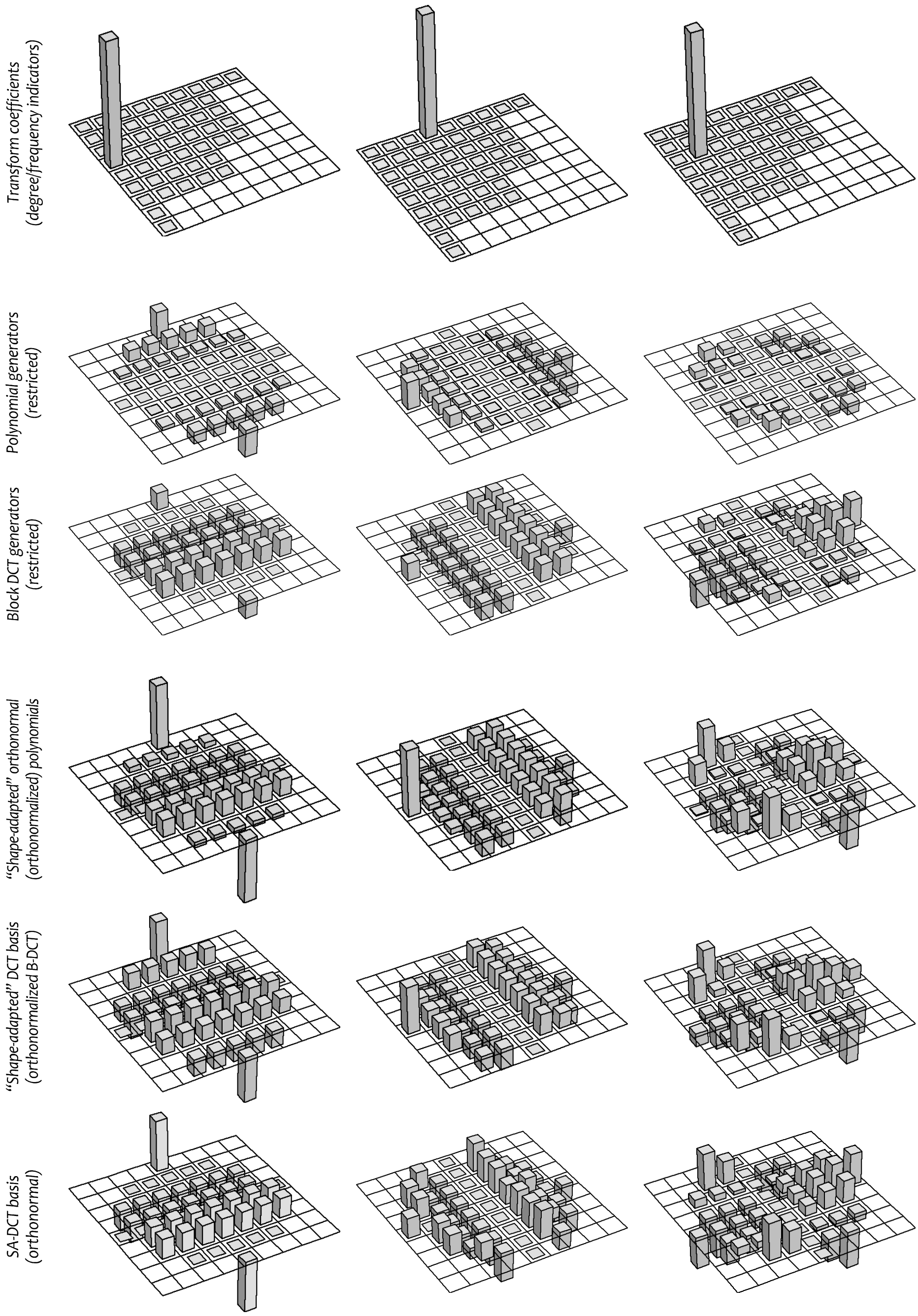

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

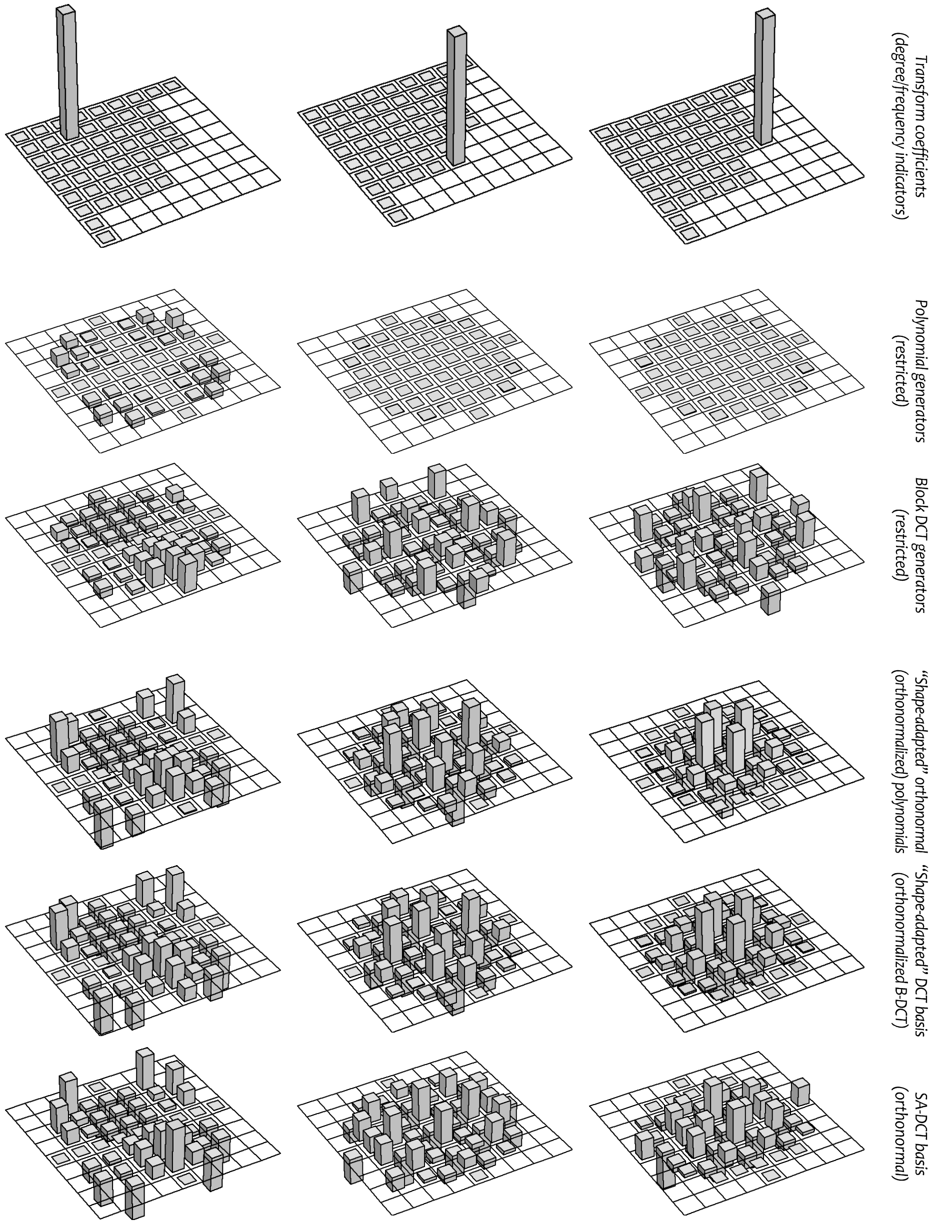

Transform coefficients (degree/frequency indicators)
Polynomial generators (restricted)
Block DCT generators (restricted)
“Shape-adapted” orthonormal (orthonormalized) polynomials
“Shape-adapted” DCT basis (orthonormalized B-DCT)
SA-DCT basis (orthonormal)

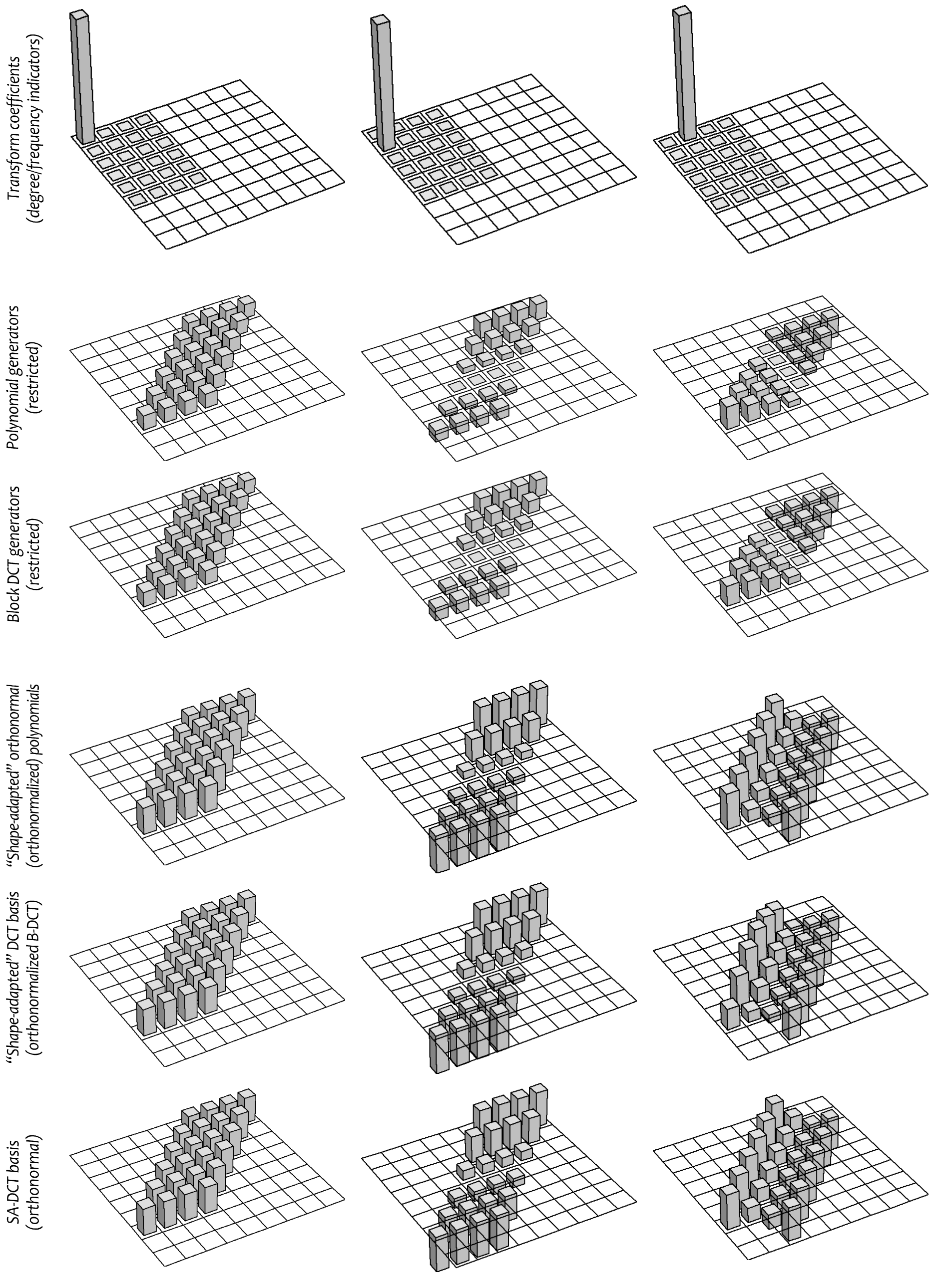

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

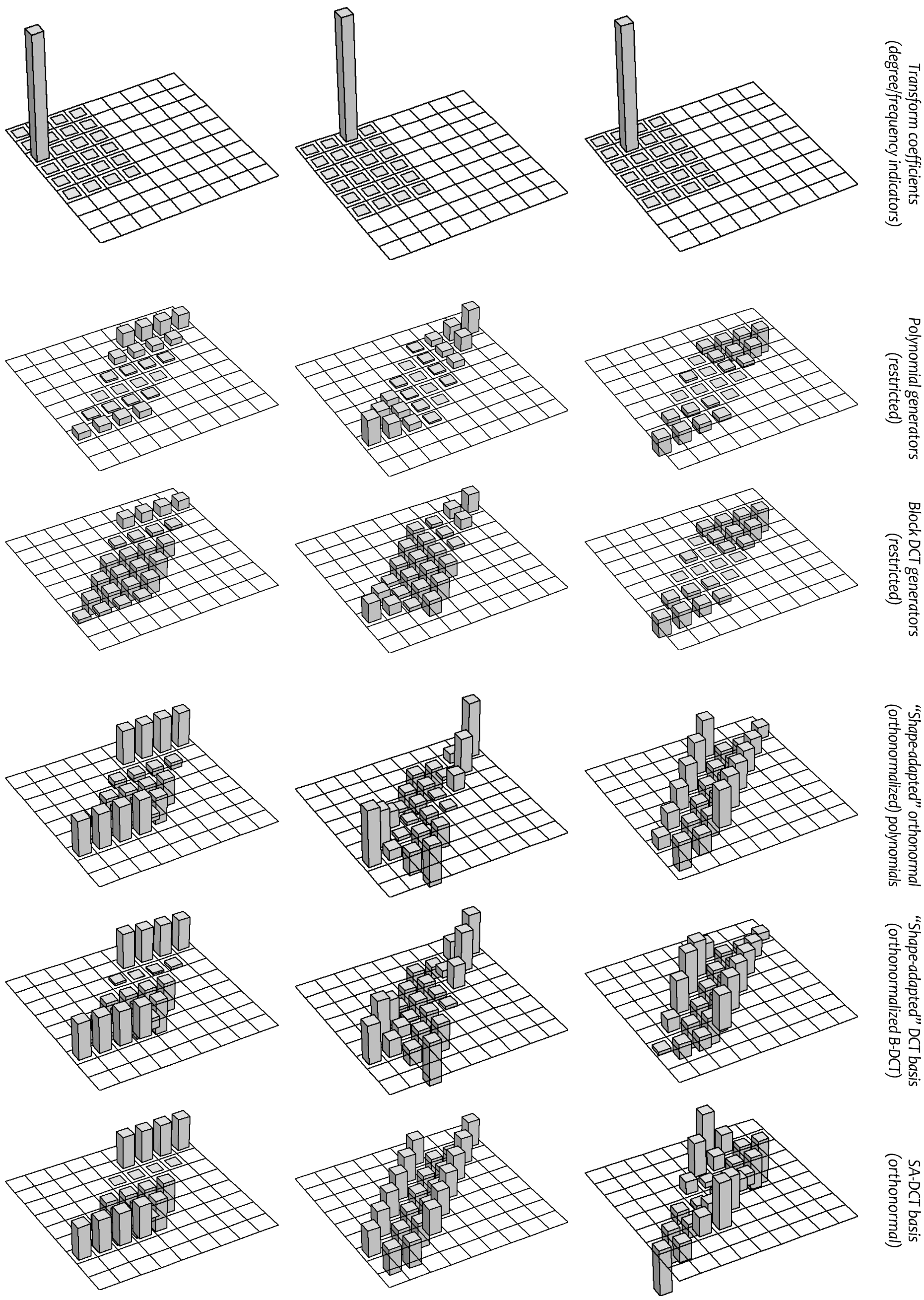

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

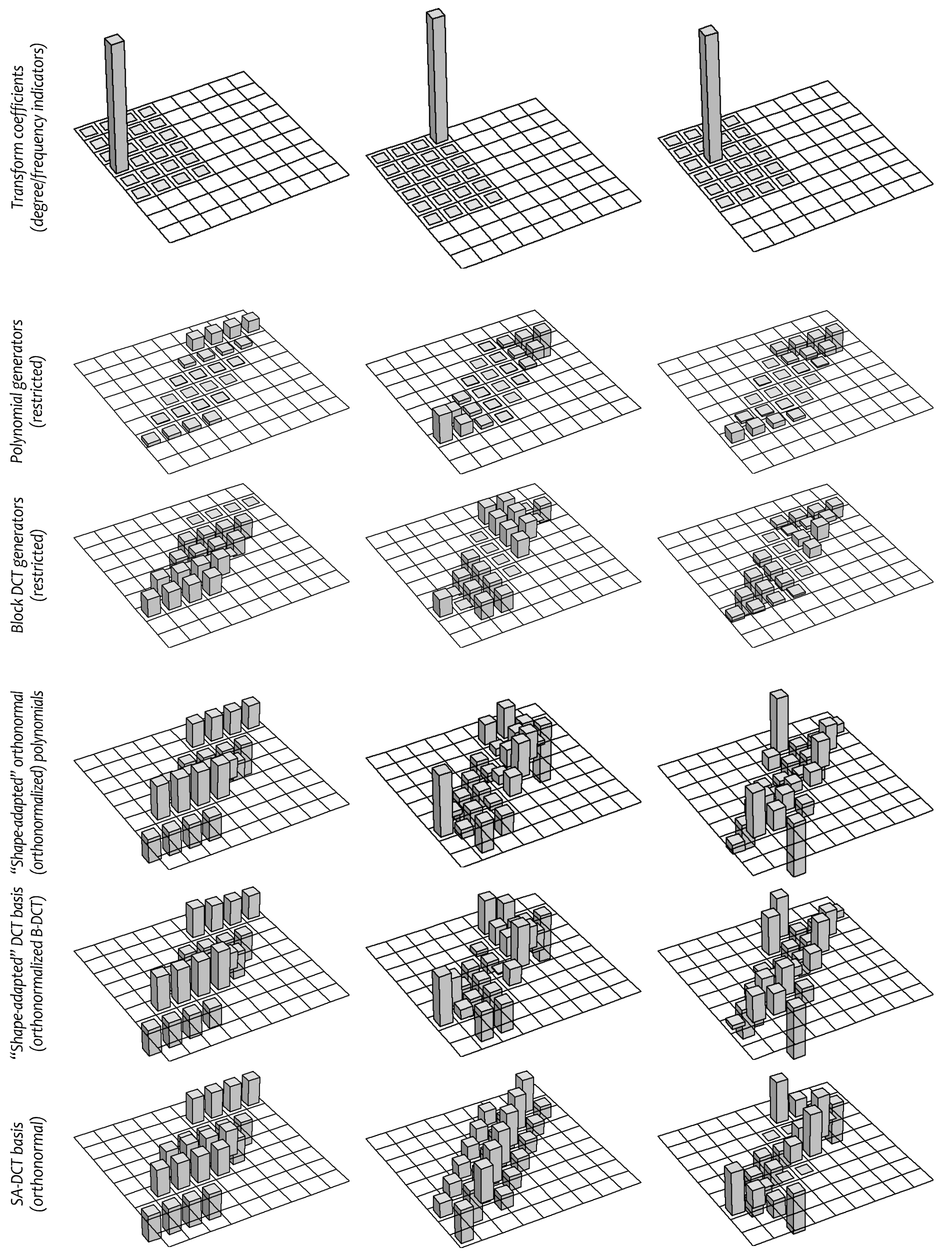

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
“Shape-adapted” orthonormal
(orthonormalized) polynomials
“Shape-adapted” DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

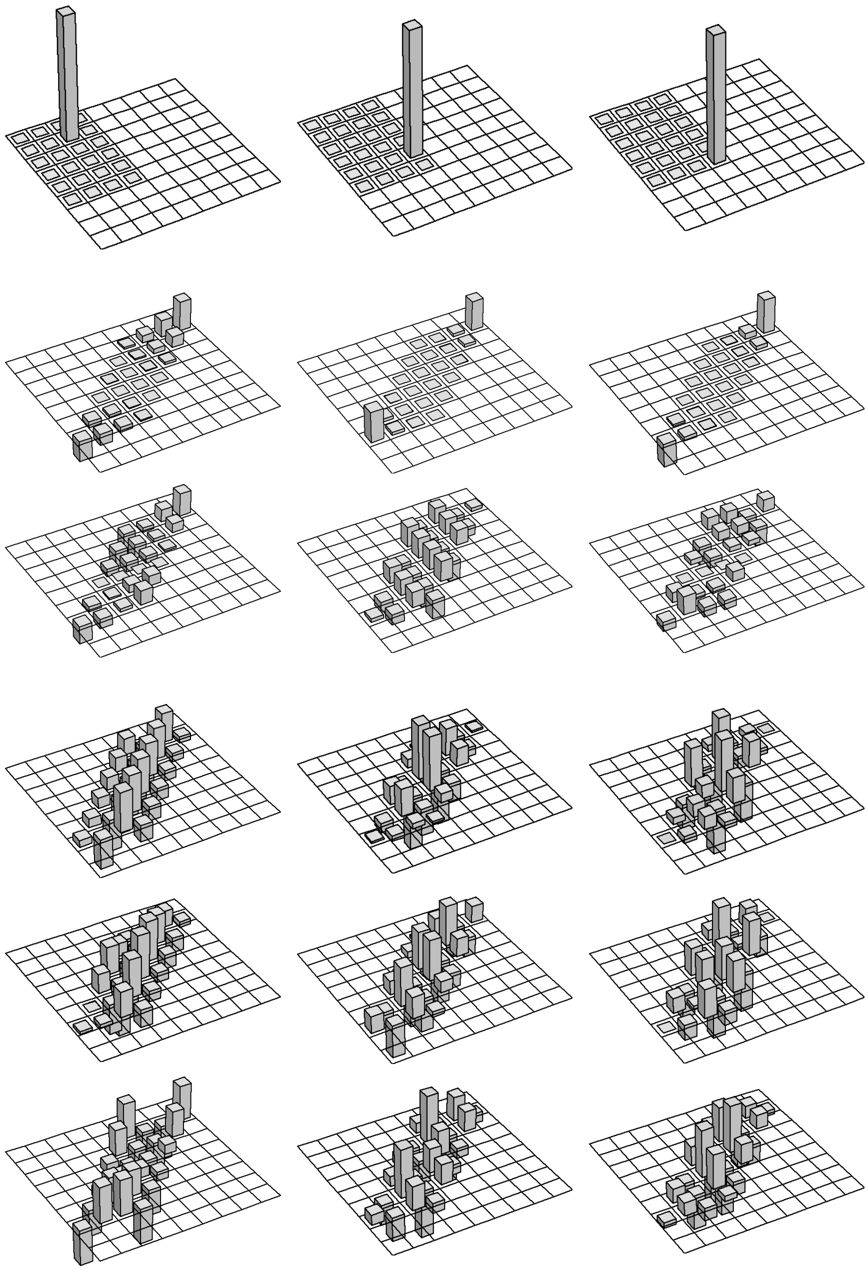

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
“Shape-adapted” orthonormal
(orthonormalized) polynomials
“Shape-adapted” DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)



Transform coefficients (degree/frequency indicators)

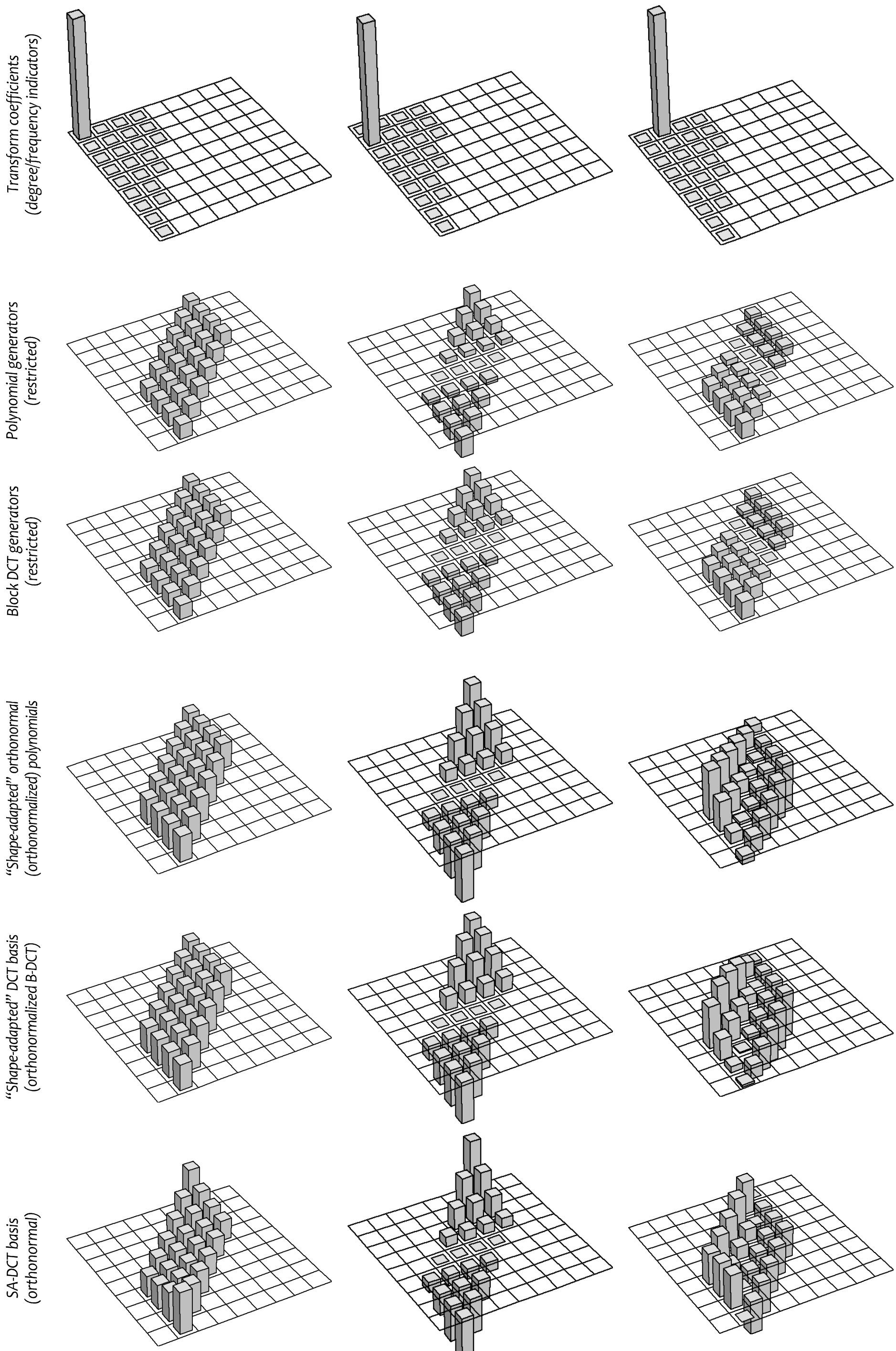
Polynomial generators (restricted)

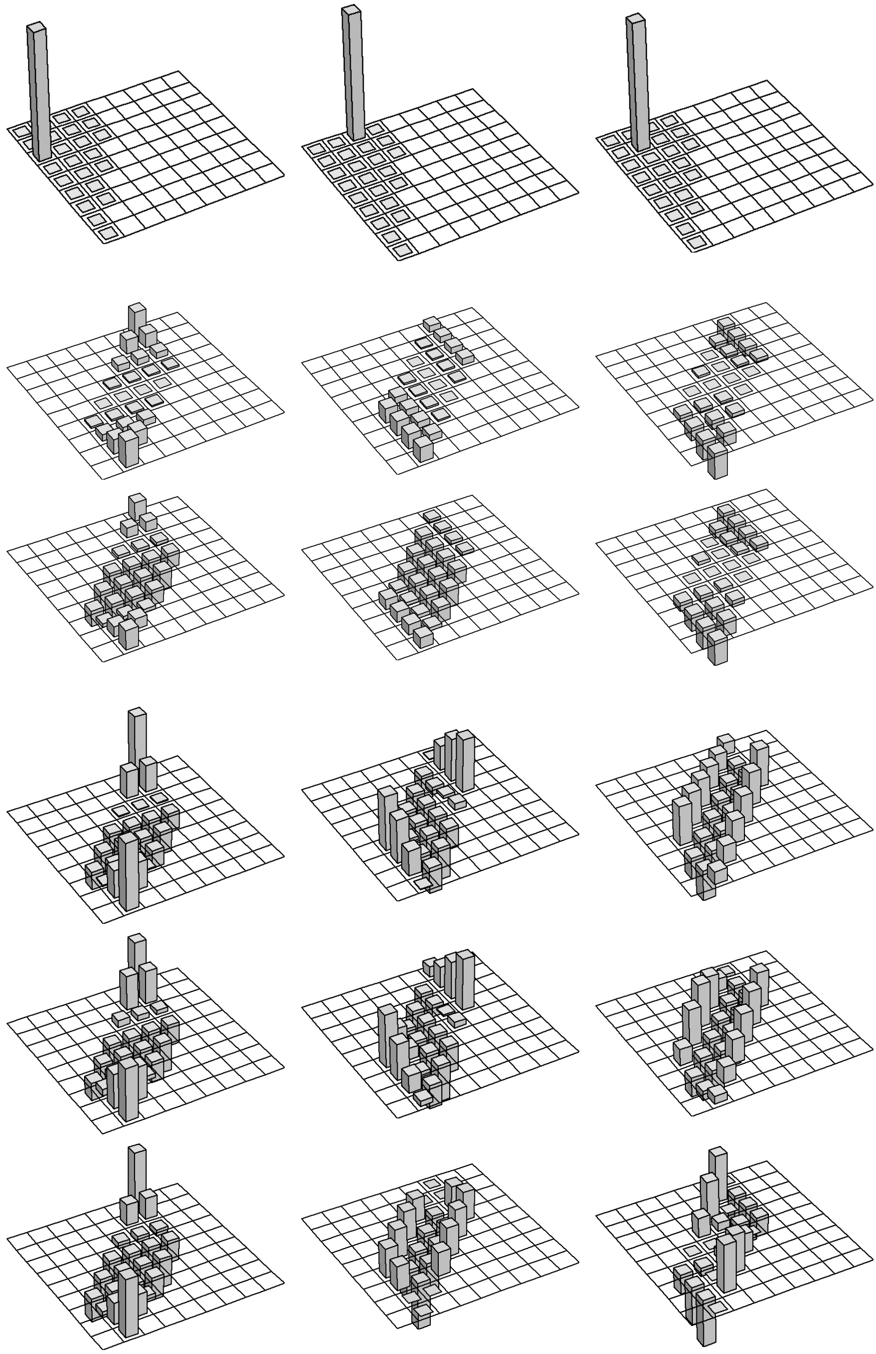
Block DCT generators (restricted)

"Shape-adapted" orthonormal (orthonormalized) polynomials

"Shape-adapted" DCT basis (orthonormalized B-DCT)

SA-DCT basis (orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)



Transform coefficients (degree/frequency indicators)

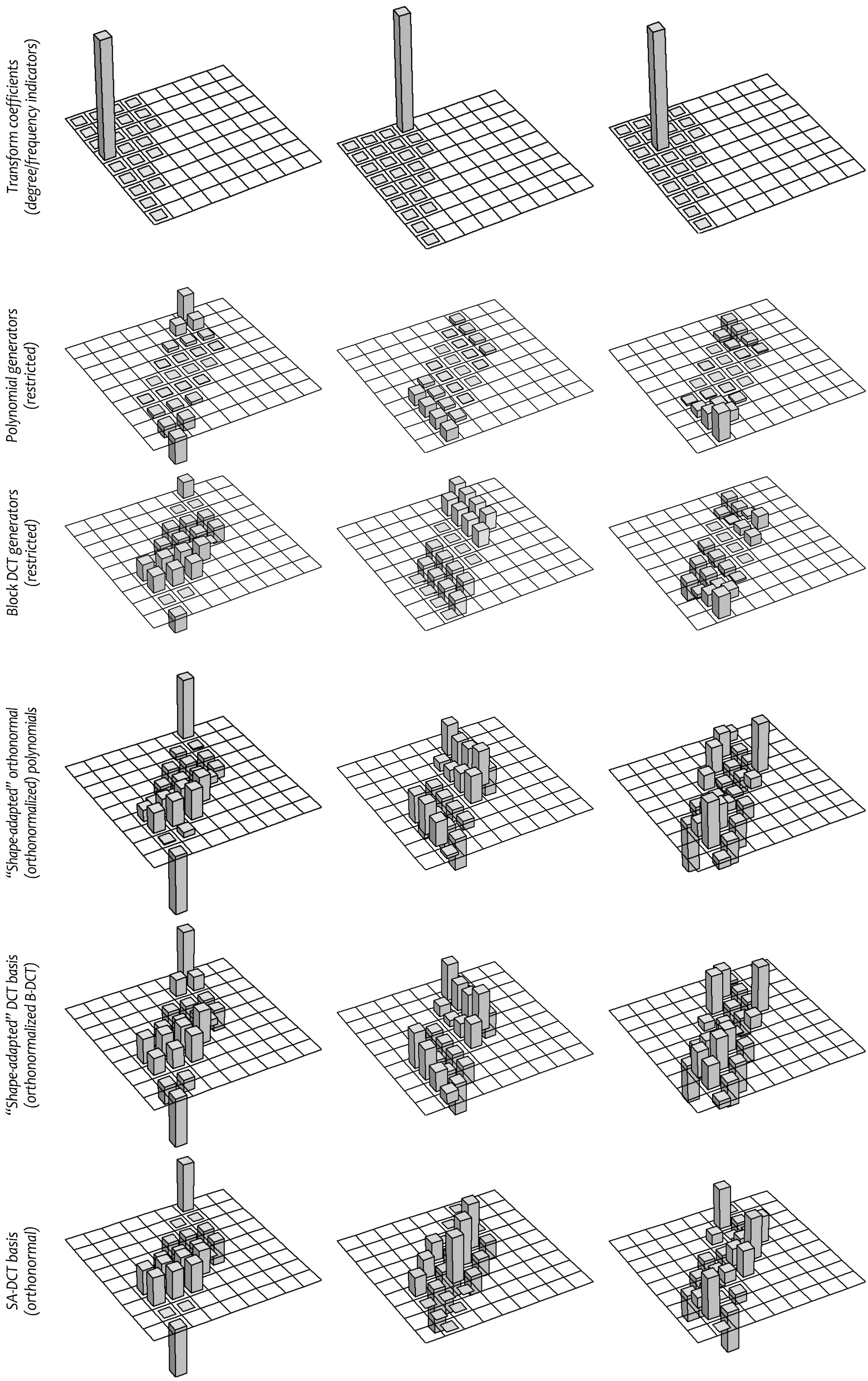
Polynomial generators (restricted)

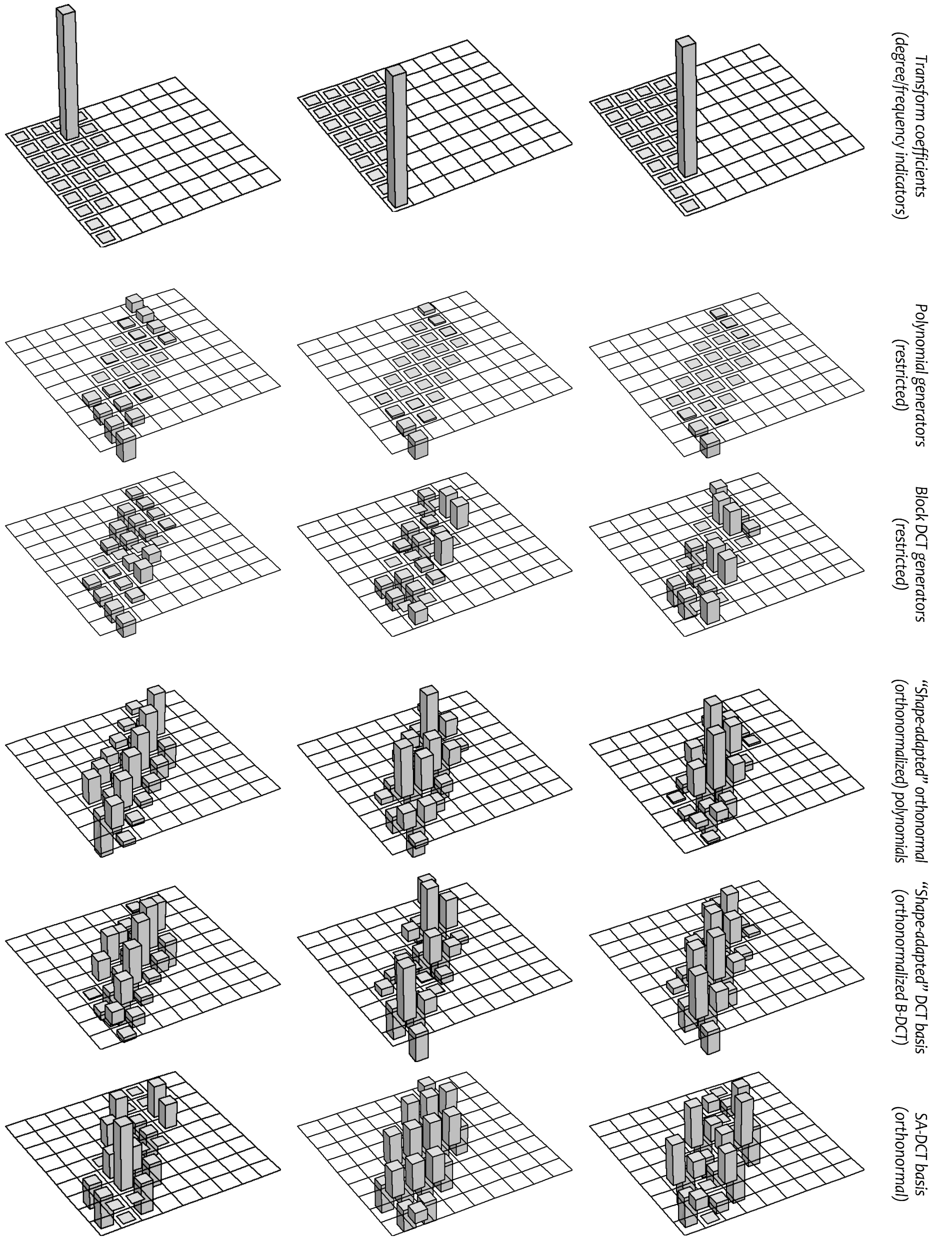
Block DCT generators (restricted)

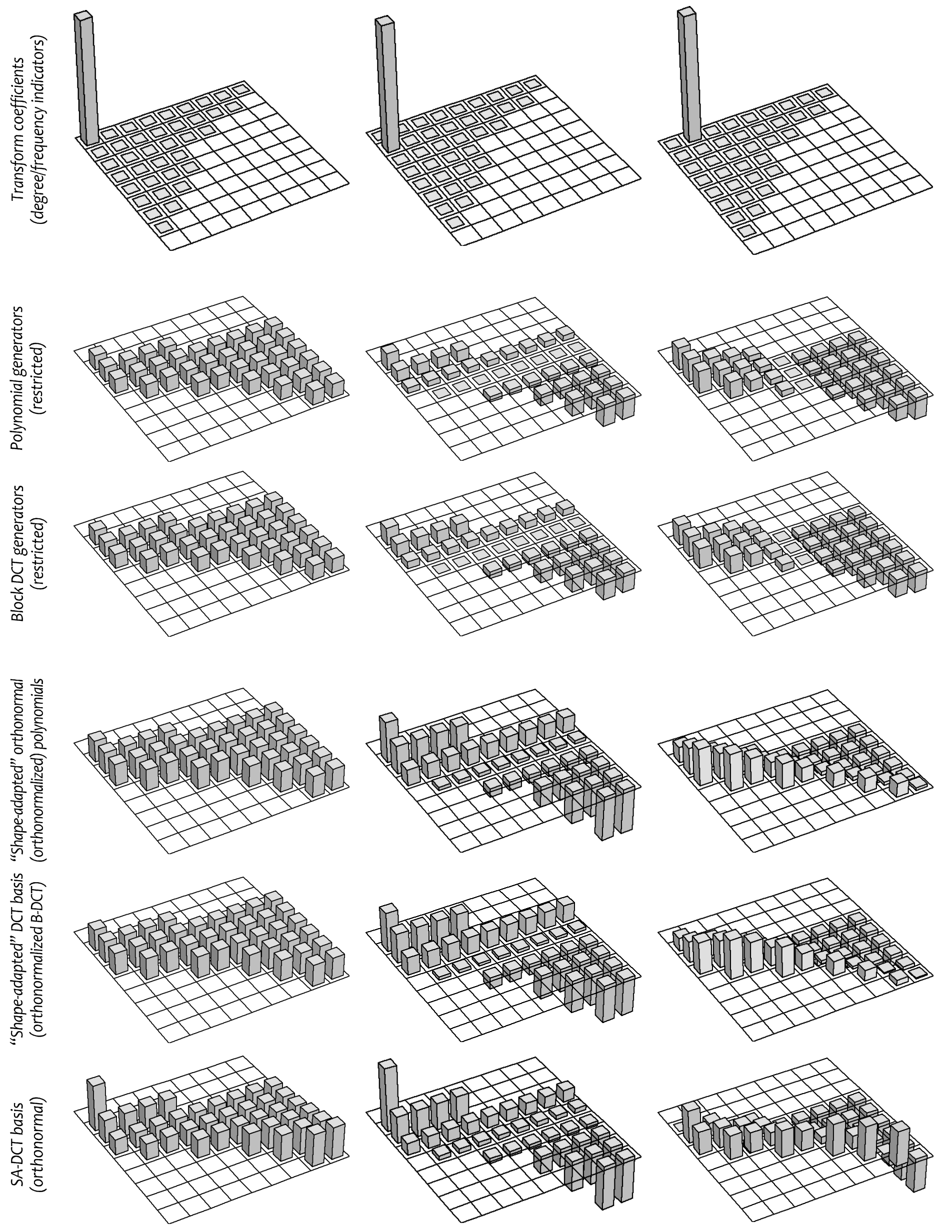
"Shape-adapted" orthonormal (orthonormalized) polynomials

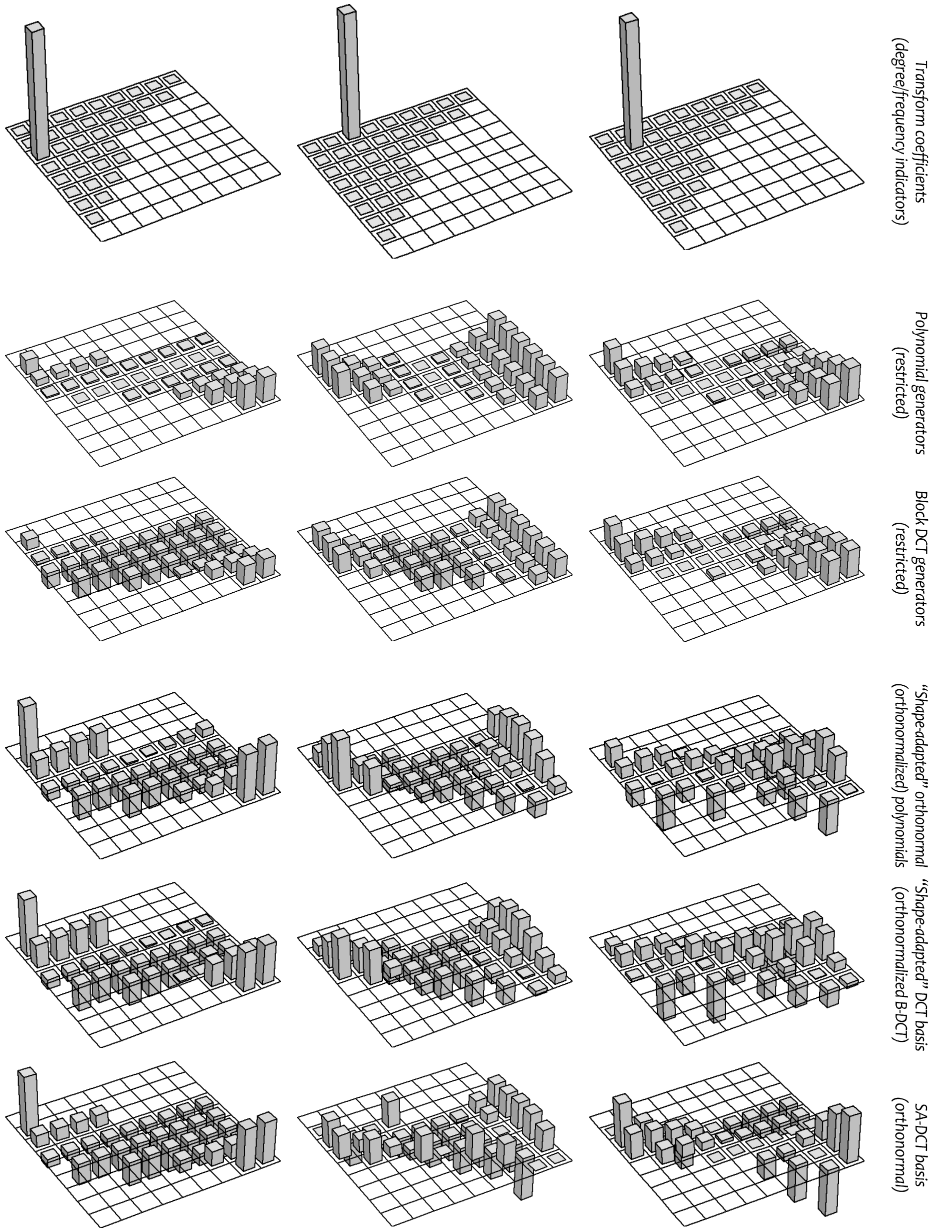
"Shape-adapted" DCT basis (orthonormalized B-DCT)

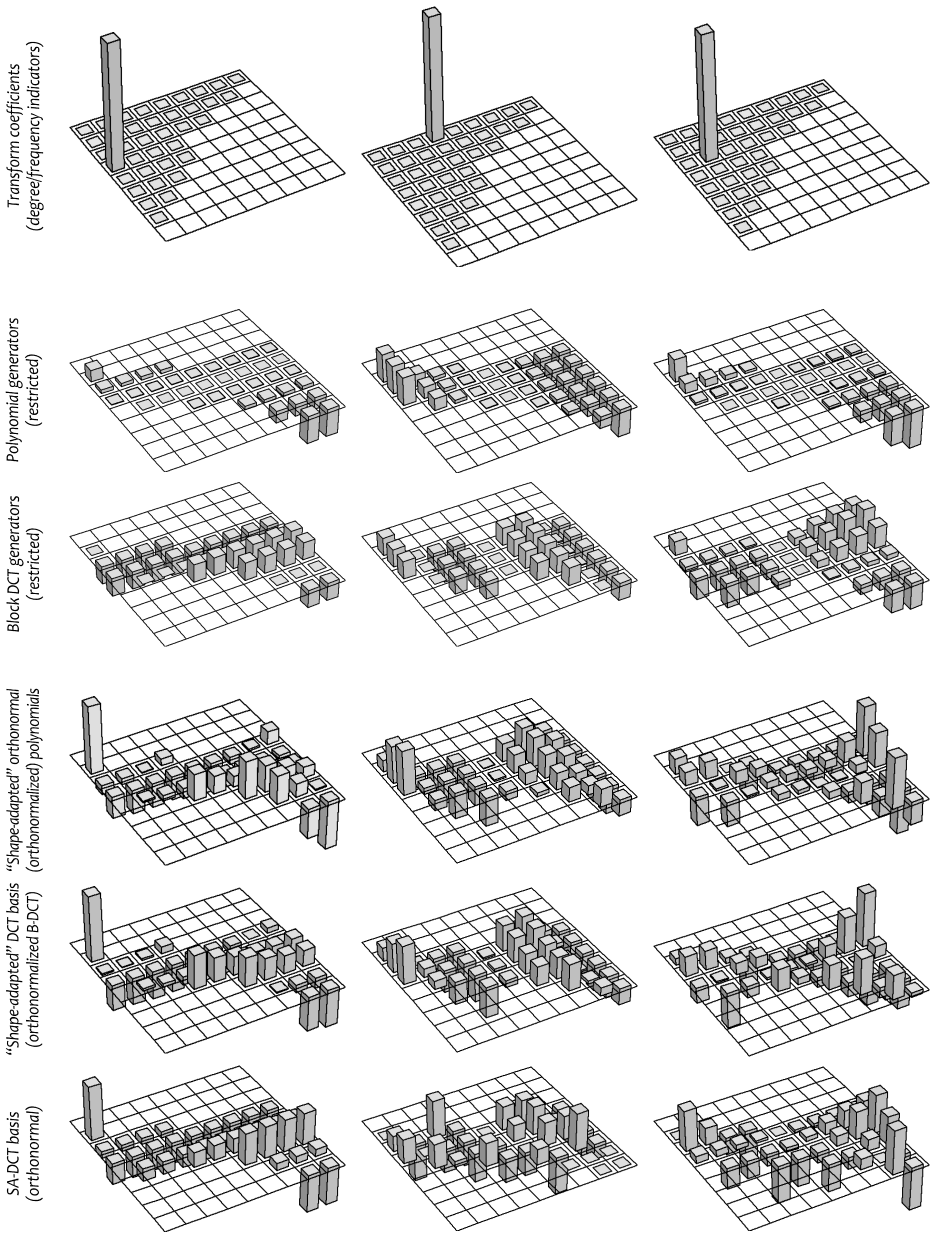
SA-DCT basis (orthonormal)

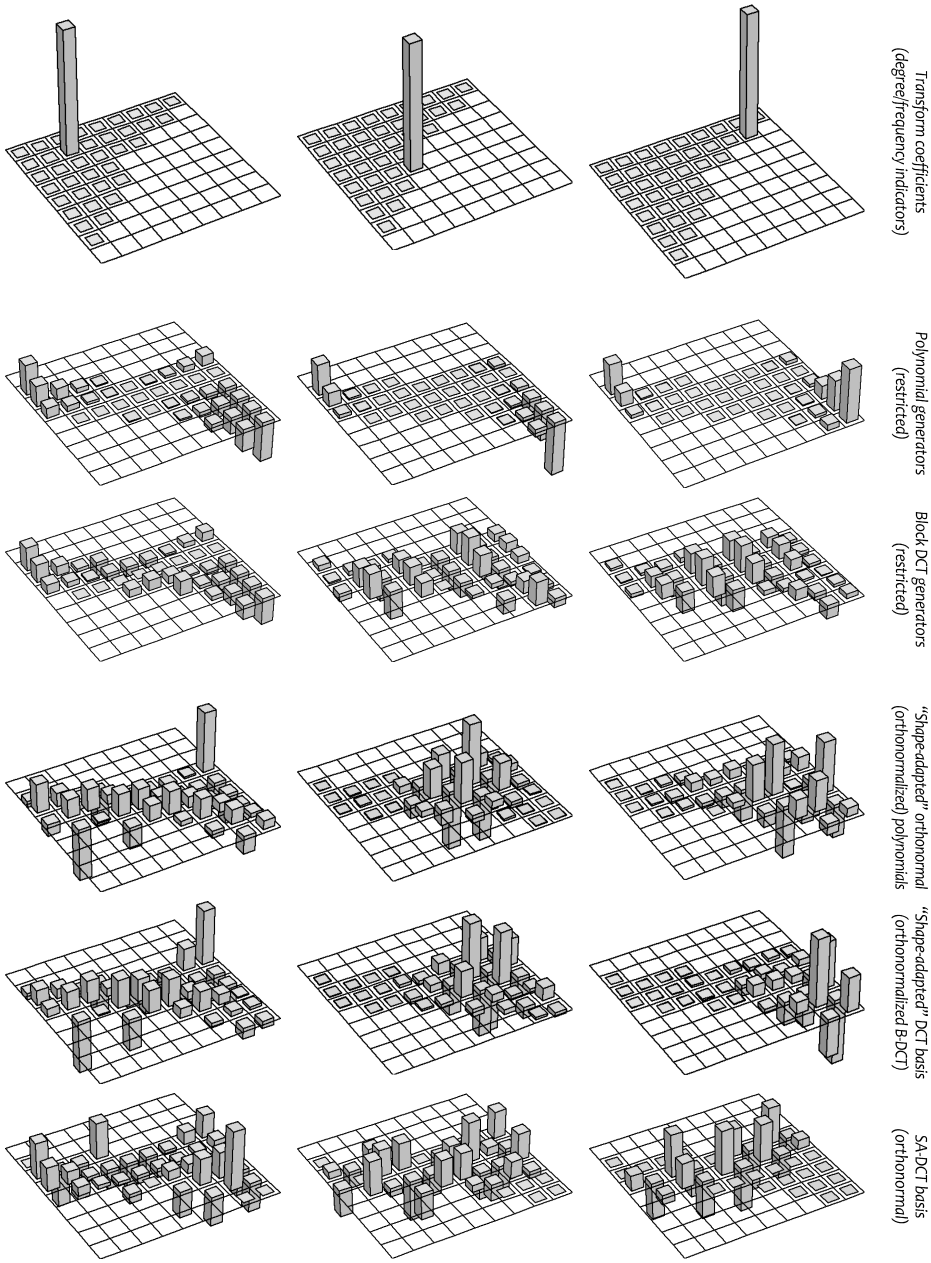

Transform coefficients (degree/frequency indicators)
Polynomial generators (restricted)
Block DCT generators (restricted)
"Shape-adapted" orthonormal (orthonormalized) polynomials
"Shape-adapted" DCT basis (orthonormalized B-DCT)
SA-DCT basis (orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
“Shape-adapted” orthonormal
(orthonormalized) polynomials
“Shape-adapted” DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
“Shape-adapted” orthonormal
(orthonormalized) polynomials
“Shape-adapted” DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
“Shape-adapted” orthonormal
(orthonormalized) polynomials
“Shape-adapted” DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)


Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
“Shape-adapted” orthonormal
(orthonormalized) polynomials
“Shape-adapted” DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

Part II

Signal-dependent noise measurement, modeling, and fitting for digital imaging sensors

Preface

Progress in hardware design and manufacturing has introduced digital imaging sensors having a dramatically increased resolution. This is mainly achieved by an increase of the pixel density. Although the electrical and thermal characteristics of the sensors have noticeably improved in the last decade [175],[143], with the size of each pixel becoming smaller and smaller the sensor output signal's susceptibility to photon noise has become greater and greater. As of now, this source of noise appears as the most significant contributor of the overall noise in a digital imaging sensor [14]. This makes the noise component of the raw-data output of the sensor markedly signal-dependent, thus far from the conventional additive white Gaussian noise (AWGN) modeling so widely used in image processing. Further, with the intention of making full use of the rather limited dynamic range of digital sensors, pictures are usually taken with some areas purposely overexposed or *clipped*, i.e. accumulating charge beyond the full-well capacity of the individual pixels. These pixels obviously present highly nonlinear noise characteristics, which are completely different than that of normally exposed pixels.

The raw-data which comes from a sensor always undergoes various processing stages (e.g., denoising, demosaicking, deblurring, white-balancing, gamma-correction, compression) before the final "cooked" image reaches the user. In order to process the data and/or to attenuate the noise in the most efficient and effective way, it is vital that a proper modeling of the noise is considered during the various stages of digital image processing. However, on the one hand, the technical datasheets of the devices usually provide vague and inadequate figures for the noise that are of a global nature (i.e., "average" values which are meant to be valid for the whole sensor) [161]. One the other hand, the existing models for the sensor noise (e.g., [13], [28], [65], [85], [163], [170]) are not directly applicable to the raw-data, as they depend on multiple parameters which are unknown and need to be estimated.

Consequently, raw-data filtering algorithms either assume independent stationary noise models or, if a signal-dependent model is assumed, the correct parameters for the noise are often not specified. Such rough noise estimates are inadequate for the high-quality image processing filters which are rapidly becoming an integral part of the imaging chain.

The strong market demand for low-cost high-resolution sensors ("Higher resolutions, initially featured in smartphones, will trickle down into lower-tier products, emulating the trends witnessed in the digital still camera market" [21]), and the ever growing consumer attention towards high-quality imaging, makes the development of efficient and noise-resilient digital image processing filters (for color interpolation, color balance, contrast enhancement, sharpening, etc.) very important. In many occasions denoising algorithms need to be inserted in the imaging chain for the image quality to be acceptable. A need therefore exists for an accurate and especially *practical* noise model for the raw-data which can be exploited in the imaging chain.

In the next two chapters we present two different approaches to raw-data noise estimation. First, in Chapter 5, we introduce a nonparametric method for estimating the standard-deviation function (i.e., the function which gives the standard-deviation of the noise as a function of the expectation of the pixel raw-data output). This method relies on the analysis of several images captured under the same fixed acquisition and illumination conditions. In contrast with other similar techniques (including the established standards [133],[134]), our developed procedure does not require an ideal uniform target. On the contrary, our method benefits from the target non-uniformity, as this allows to estimate at once the standard-deviation function over a large range of output values.

However, the most significant contributions in this second part of the thesis are given in Chapter 6. There, we propose a simple parametric model for the raw-data noise. The model includes a Poissonian and a Gaussian component and it explicitly considers the nonlinearity of the sensor response due to under- and over-exposure. An algorithm for the fully automatic estimation of the model parameters from a single noisy image is also developed. Experiments with synthetic images and real raw-data from camera sensors demonstrate the effectiveness and accuracy of the algorithm in estimating the model parameters and confirm the validity of the proposed model.

A remark on the availability and use of raw data

The algorithms presented in what follows are meant to be applied on the raw data, i.e. the unprocessed digital output of the imaging sensor, and not on the processed images (typically stored as TIFF or JPEG files) that are obtained at the end of the digital image processing chain which takes place within the camera. Nearly all professional or high-end consumer digital cameras do provide the user with direct access to the raw data. Low-end cameras and cameraphones do not usually provide this feature and only processed images can be downloaded from the device. The availability or unavailability of a "raw-data saving" feature in a device is dictated mainly by market-positioning strategies, a not by hardware limitations. As a matter of fact, raw data does always exists inside the device and can be accessed using special software provided by the manufacturer. There exist also a number of "hacked" firmwares (see, e.g., [17]) which can be installed on consumer cameras enabling various hidden features, including raw-data saving. These unofficial firmwares have been developed by enthusiasts through reverse engineering of the official camera firmwares and are neither supported nor endorsed by the manufacturer. Even though the use of "hacked" firmwares voids the warranty of the camera, they are rather popular because accurate processing of the raw-data allows the users to obtain high-quality images that would not otherwise be obtainable by postprocessing the standard processed output TIFF or JPEG files. Plugins for raw-data processing exist for the common consumer image processing softwares like Adobe Photoshop or GIMP. Thus, even though the algorithms presented in this chapter have been developed mainly for a scientific or industrial usage, they are directly relevant also for end-user applications.

Chapter 5

Noise measurement for raw-data by automatic segmentation of non-uniform targets

5.1 Introduction

Since the sensor noise is signal-dependent, when measuring the standard-deviation from the recorded data it is necessary to either have uniform data, or take into account for the data non-uniformity.

However, in practice, it is very difficult (sometimes even impossible) to guarantee uniform recordings. Even with perfectly uniform targets and illumination, the lens of the camera introduce a systematic "vignetting" effect where the center of the image is much brighter (and hence noisier) than the peripheral area. As a result, the measurements taken under these assumptions are inherently biased. Unfortunately, the current international standards ISO 15739 and ISO 14524 [134],[133] and proposed approaches (e.g., [14],[13],[82]) for measuring the noise assume known uniform targets and thus provide results that are of a global nature (i.e., "average" values which are meant to be valid for the whole sensor).

In this chapter we present an approach for measuring the temporal noise in the raw-data of digital imaging sensors. The method is specially designed to estimate the curve which describes the standard-deviation of the noise as a function of the expectation of the pixel raw-data output. Based on an automatic segmentation of the recorded images, we separate samples with different expected output and calculate their standard-deviations. Thus, while other techniques require or assume a uniform target, in our approach we benefit from the target non-uniformity by simultaneously estimating the variance function over a large range of output values.

Because of the automatic segmentation embedded in the procedure, our approach has a number of advantages over current noise measurement standards:

- Our approach does not require a specific target (i.e., a test chart). In fact, any fixed target or scene can be used for performing the noise measurements.
- The target and illumination do not need to be known in advance, i.e., no calibration is required before performing the measurements.
- The method is not influenced by the focusing or by the presence of the lens of the camera.
- Illumination does not need to be uniform in space. It is sufficient that the illumination is constant in time.
- The method is applicable without modifications for gray as well as for color sensors (Bayer pattern).
- Fixed-pattern noise does not influence the measured temporal noise.
- With a single experiment (consisting of multiple shots) we measure a whole standard-deviation vs. expectation curve, whereas previous techniques aimed at estimating only a single standard-deviation value for a given uniform intensity.

Overall the method is simple, easy to implement, and allows for accurate measurement with much simplified laboratory equipment. It is thus a cost-effective alternative to other noise measurement techniques.

We emphasize that our method is not intended to characterize the sensor's response (e.g., to derive a camera response function [74]), but rather the temporal noise in the sensor output. In this sense, if the sensor output is interpreted as an estimate of the scene radiance, we consider only the variance of this estimate, and not its bias.

5.2 Observation model

We consider an observation model of the form

$$z(x) = y(x) + \sigma(y(x))\,\xi(x), \qquad x \in X, \tag{5.1}$$

where X is the set of the sensor's active pixel positions, z is the actual raw-data output, y is the ideal output, ξ is zero-mean random noise with standard deviation equal to 1, and σ is a function of y, defining the standard-deviation of the overall noise component. The function σ is called *standard-deviation function* or *standard-deviation curve*. The function σ^2 is called *variance function*. Since $E\{\xi(x)\} = 0$ we have $E\{z(x)\} = y(x)$ and $\operatorname{std}\{z(x)\} = \sigma(E\{z(x)\})$. There are no additional restrictions on the distribution of $\xi(x)$, and different points may have different distributions.

In practice, $z(x)$ is the recorded value of the raw-data at the pixel x, and $y(x)$ is the ideal value to be recorded if no quantization or noise were present. The (signal-dependent) signal-to-noise ratio (SNR) of the imaging sensor can be expressed as $\mathrm{SNR}(y) = \frac{y}{\sigma(y)}$.

A good estimate of $y(x)$ can be obtained as the pointwise average of a large enough number N of observations $z_n(x)$, $n = 1, \ldots, N$, of the form (5.1):

$$\frac{1}{N}\sum\nolimits_{n=1}^{N} z_n(x), \qquad z_n(x) = y(x) + \sigma(y(x))\,\xi_n(x). \tag{5.2}$$

5.3 The method

The experimental realization of the approach described by (5.2) requires that the deterministic terms of the equation are truly invariant with respect to the replication index n. That is, the underlying true signal $y(x)$ must not change over time. In practice this means that during the acquisition process the camera must not move, the acquisition parameters (e.g., exposure, aperture, gain) must be fixed and the illumination is constant in time.

5.3.1 Acquisition and averaging

Under these conditions, we record a number N of images in raw-data format. These shots are averaged, to obtain an approximation $\bar{z}$ of the noise-free y,

$$\bar{z}(x) = \frac{1}{N}\sum\nolimits_{n=1}^{N} z_n(x) = y(x) + \frac{\sigma(y(x))}{\sqrt{N}}\tilde{\xi}(x), \quad x \in X. \tag{5.3}$$

Here $\tilde{\xi}(x)$ is again some zero-mean noise with unitary variance. It is recommended to take a large number of images, so that the factor $1/\sqrt{N}$ in (5.3) is small. In what follows we assume that N is large enough and we consider $\bar{z}(x) = E\{z(x)\} = y(x)$, for all x.

5.3.2 Segmentation

The average image $\bar{z}$ is segmented into a number of uniform regions $\{S\}$, or segments. Ideally, within these regions the value of $\bar{z}(x)$ should be constant:

$$S(y) = \{x : \bar{z}(x) = y\}.$$

However, this may lead to uncertain results as there may be too few (or maybe none) samples (i.e., pixels) that satisfy the equality $\bar{z}(x) = y$. Pragmatically, it is convenient to consider a larger estimation set of the form

$$S_\Delta(y) = \{x : \bar{z}(x) \in [y - \Delta/2, y + \Delta/2)\},$$

where $\Delta > 0$.

Let y and Δ be fixed and denote by x_m, $m = 1, \ldots, M$, the coordinates of the M pixels that constitute the segment $S_\Delta(y)$, $\{x_m\}_{m=1}^{M} = S_\Delta(y)$. On this segment we have that the observations (5.2) satisfy

$$\begin{aligned} z_n(x_m) &= y(x_m) + \sigma(y(x_m))\,\xi_n(x_m) = \\ &= y + \Delta d_{n,m} + \sigma(y(x_m))\,\xi_n(x_m), \end{aligned} \tag{5.4}$$

where $d_{n,m} \in [-1/2, 1/2)$. It is reasonable to assume that $d_{n,m}$ are uniformly distributed. Hence, Equation (5.4) can be rewritten as

$$z_n(x_m) = y + \sqrt{\frac{\Delta^2}{12} + \sigma^2(y(x_m))}\xi'_n(x_m), \tag{5.5}$$

where ξ'_n is again some zero-mean random noise with standard deviation equal to 1.

5.3.3 Measurement of the standard deviation

For any fixed y we proceed as follows in order to compute the estimate $\hat{\sigma}(y)$ of $\sigma(y)$.

Variance for a single image and output value

To contain memory requirements, the variances are first computed independently for each shot z_n. The variance $\mathrm{svar}_n(y)$ corresponding to expected output y and shot n is calculated as the unbiased sample variance estimator on the segment,

$$\mathrm{svar}_n(y) = \frac{\sum_{m=1}^{M}(z_n(x_m) - \tilde{z}_n(y))^2}{M-1}, \tag{5.6}$$

where

$$\{x_m\}_{m=1}^{M} = S_\Delta(y) = \{x : \bar{z}(x) \in [y - \Delta/2, y + \Delta/2)\}$$

and $\tilde{z}_n(y)$ is the average of z_n over $S_\Delta(y)$, $\tilde{z}_n(y) = \frac{1}{M}\sum_{m=1}^{M} z_n(x_m)$.

By (5.5), and assuming that within the segment $\sigma(y(x_m))$ can be well approximated by $\sigma(y)$, we obtain

$$E\{\mathrm{svar}_n(y)\} = \frac{\Delta^2}{12} + \sigma^2(y). \tag{5.7}$$

Averaging over the shots

The estimate $\hat{\sigma}(y)$ of the standard deviation $\sigma(y)$ is given according to (5.7) by the average of the above N estimates:

$$\hat{\sigma}(y) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\mathrm{svar}_n(y) - \frac{\Delta^2}{12}}. \tag{5.8}$$

Here we assume that the expectation under the square-root is non-negative. In practice, one can neglect the extra term due to Δ in (5.7) and (5.8) by choosing Δ significantly smaller than $\sigma(y)$. In this case we simply have

$$\hat{\sigma}(y) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\mathrm{svar}_n(y)}.$$

We remark that while the recorded raw-data has a certain fixed precision (e.g., 10 bits), the values attained by $\bar{z}$ are much denser because of the averaging. Therefore Δ can be taken much smaller than the quantization step of the raw-data. In Section 5.4 we show results obtained for much different choices of Δ which demonstrate that the proposed technique is very stable with respect to this parameter.

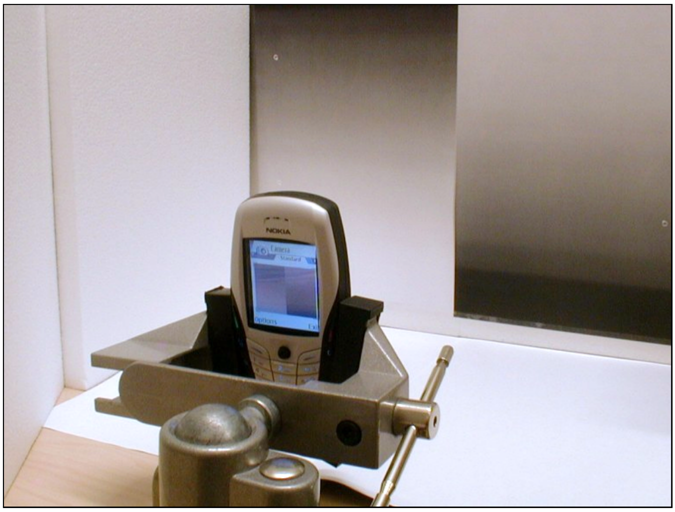


Figure 5.1: Experimental setup.

Standard-deviation curve

The procedure described by the previous two steps is repeated for different values of y. In this way the standard-deviation curve is found. Typically, one would use discrete values $y \in \{y_i,\ i = 1, 2 \ldots\} = \{\Delta j, j \in \mathbb{N}\} \cap [\min\{\bar{z}\}, \max\{\bar{z}\}]$, which ensures that the segments are nonoverlapping and hence that the measurements for different values of y are independent. The estimate of the variance function is simply $\hat{\sigma}^2(y)$, $y \in [\min\{\bar{z}\}, \max\{\bar{z}\}]$.

5.4 Experimental results

The standard-deviation curves $\sigma(y)$, as from Equation (5.1), have been measured from the raw-data of two different CMOS sensors used in Nokia cameraphones. We denote them here as sensor/cameraphone "U" and "V". They are, respectively, an older 660×492 (VGA) 1/4" sensor (5.4μm pixel pitch) with global shutter, and a newer 1296×1040 (1.3 Mpixel) 1/3.3" sensor (3.3μm pixel pitch) with rolling shutter. Both sensors have a Bayer pattern color filter array (CFA) with red, green, and blue filters (R, G_1, G_2, and B).

5.4.1 Setup

The sensors had not been separated from the phones, which were held clamped in a vice. We used a non-uniform target composed by grayscale vertical ramps going from white to black[1]. This simple setup is shown in Figures 5.1 and 5.2.

[1] The target was printed on normal paper using a common office printer and then enlarged to A3-size using a copy machine. Graininess or texture, which may be present in such a target, do not constitute a problem, since the estimation procedure does not rely on any sort of neigh-

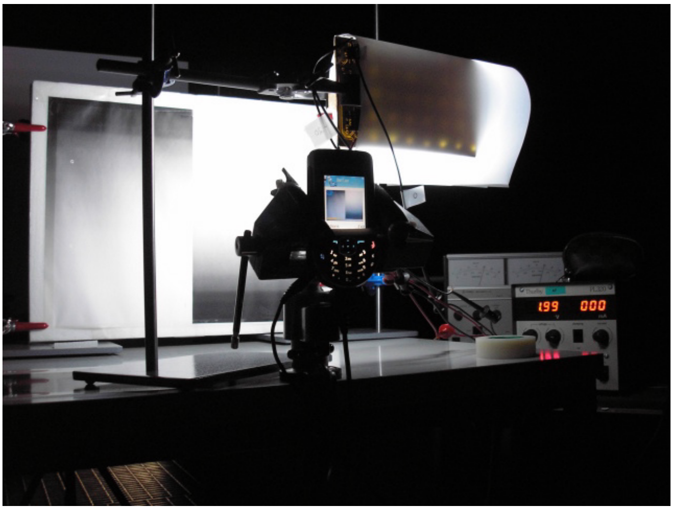


Figure 5.2: Experimental setup.

To ensure constant-in-time (flicker-free) illumination, measurements where taken in a darkroom where the only source of light was an array of white LED lights powered by stabilized DC power supplies.

The devices were configured to take multiple shots automatically and without user intervention (as this would introduce mechanical vibration). A total of 50 shots were taken for each experiment. In Figure 5.3 we show an example of the shots which were taken. Enlarged details of this shot and of the average image $\bar{z}$ (5.3) are shown in Figure 5.4. The raw-data had 10-bit precision, which we normalize on the range $[0,1]$ dividing by 2^{10}=1024. The exposure time, gain, and illumination were fixed in such a way that the recordings were not saturated (i.e., clipped) and thus our raw-data output is typically concentrated inside the range $[0,0.5]$. Furthermore, because of pedestal offset, the output is always larger than a certain minimum value, which for the two considered devices is about 0.05.

5.4.2 Segmentation

In Figures 5.5 and 5.6 we show examples of a segment $S_\Delta(y)$ obtained for $\Delta = 0.01$ and $\Delta = 0.0001$. These values are respectively about ten times and one tenth of the quantization step 1/1024. Observe that the segment corresponding to the smaller Δ is a subset of the other one. It is interesting to observe that despite the target was composed by vertical ramps, because of the nonuniform illumination and the vignetting, the segments do not have a particular horizontal/vertical "shape". Especially for small values of Δ the segments are usually composed of separated (i.e., disconnected) pixels.

---

borhood operation in the spatial domain (e.g., smoothing or local filtering). In this respect, we emphasize that the segments $S_\Delta(y)$ are *not* connected sets.

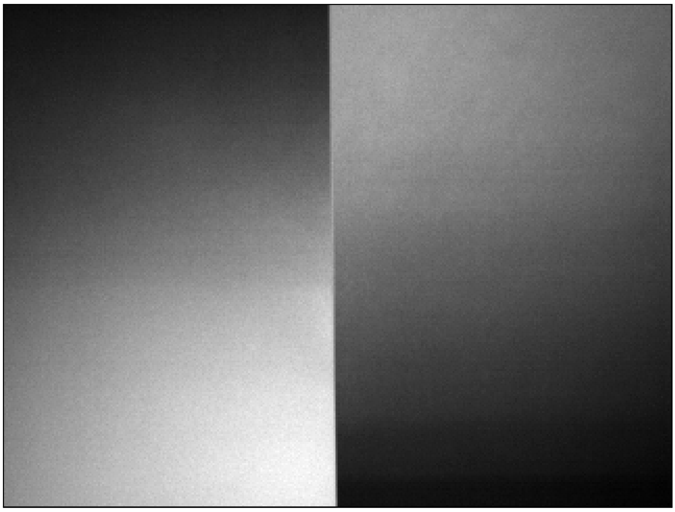


Figure 5.3: One of the acquired shots z_n.

5.4.3 Standard-deviation curves

In Figure 5.7, we show the standard-deviation curves obtained for $\Delta = 0.005$, 0.001, 0.0005, 0.00005. It can be seen that, although the curve becomes noisier for very small Δ, the three plots are essentially the same. This demonstrates the accuracy of (5.7-5.8) and the stability of our procedure with respect to the choice of Δ. We remark also that the noisiness of the plot obtained for $\Delta = 0.00005$ is well compensated by its higher "sampling" density (there are about 10000 samples in the plot) which allows for very accurate smoothing or parametric fitting.

Due to the automatic segmentation, the standard-deviation vs. expectation curve can been measured using the whole sensor at once as well as using each color channel separately. The two approaches are in general equivalent. In Figure 5.8 we show the plots obtained separately for each one of the four color channels (R, G_1, G_2, and B), whereas the plots shown in Figure 5.7 are due to measurements using the whole sensor. The four color channels exhibit the same behavior shown in Figure 5.7.

We note that the sharp vertical drops in the estimated standard-deviation $\hat{\sigma}$ visible in the plots (e.g., for the blue and for the red channels in Figure 5.8(left)) do not correspond to a real drop in the standard deviation of the noise. Instead, they are only due to the segments $S_\Delta(y)$ becoming singletons or empty sets when y approaches or exceeds the bounds of the interval $[\min\{\bar{z}\}, \max\{\bar{z}\}]$. For empty or singleton segments $S_\Delta(y)$, the Equation (5.6) loses its meaning. Such drops in the estimated standard-deviation are therefore to be ignored when analyzing the estimated curve $\hat{\sigma}(y)$.

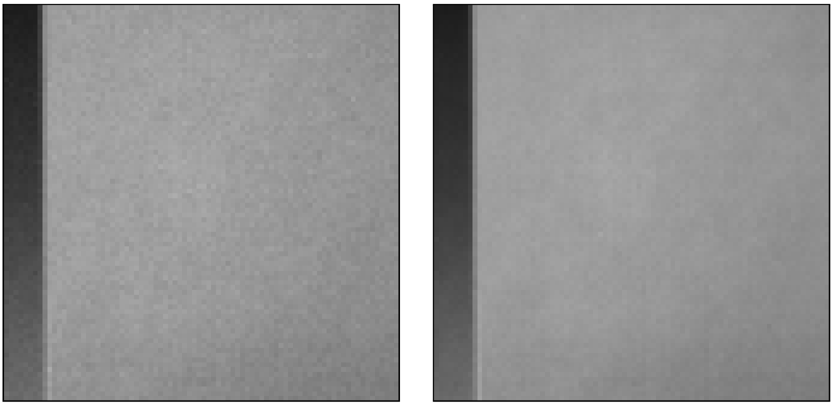


Figure 5.4: Enlarged detail of the acquired shot z_n shown in Figure 5.3 and the corresponding fragment of the image $\bar{z}$, which is obtained by averaging N such z_ns.

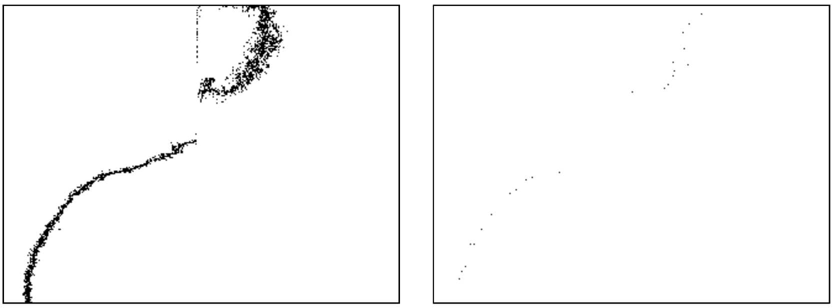


Figure 5.5: Two segments $S_\Delta(y)$ obtained for Δ=0.01 (left) and Δ=0.0001 (right). The value of y is the same for both segments.

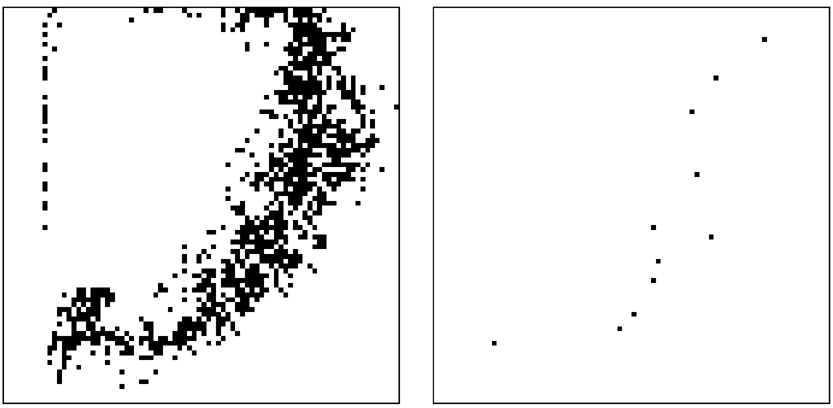


Figure 5.6: Enlarged details of the segments $S_\Delta(y)$ obtained for Δ=0.01 and Δ=0.0001, shown in Figure 5.5. These details correspond to the same enlarged location shown in Figure 5.4.

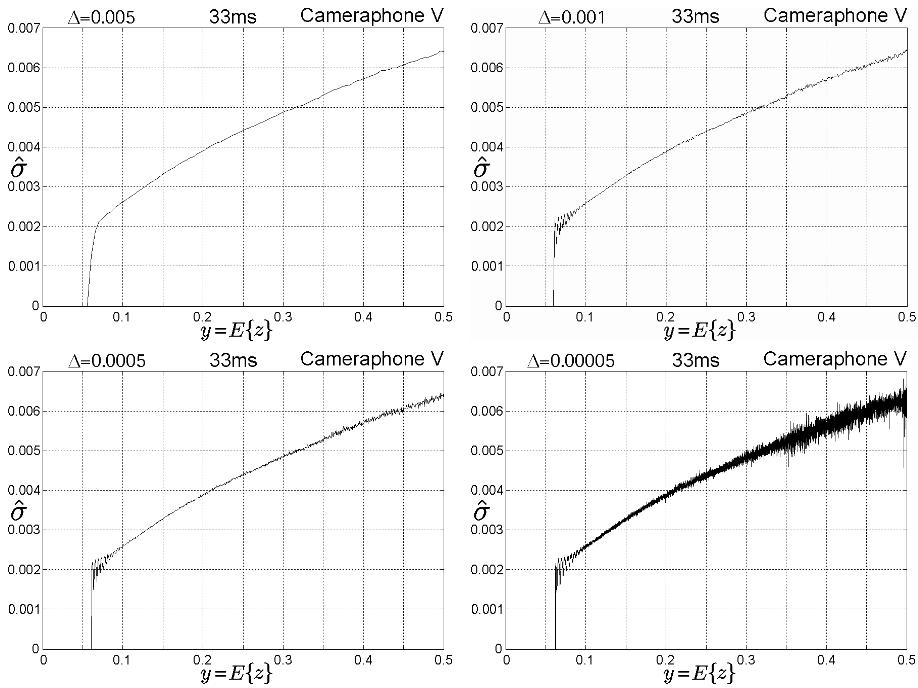


Figure 5.7: Standard-deviation curves for the raw-data from CMOS sensor "V" measured for different values of the segmentation parameter Δ=0.005, 0.001, 0.0005, 0.00005. The curves are calculated using Equation (5.8) and give an estimate $\hat{\sigma}$ of the standard-deviation of the temporal noise in the raw-data as function of the expected (i.e., noise-free) output raw-data value $y = E\{z\}$.

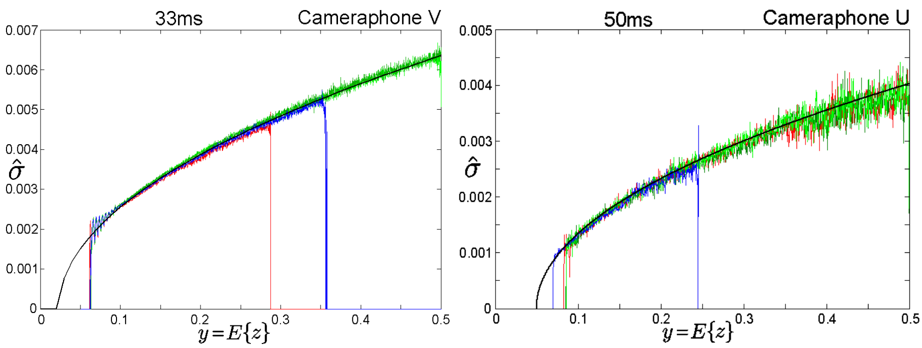


Figure 5.8: Measured standard-deviation curves (and corresponding fitted model) for the raw-data from CMOS sensor "V" (left) and "U" (right). Plots for the different color channels (R, G_1, G_2, B) are drawn in their corresponding colors. The model fits equally well all color channels. A comparison between the two curves shows that for the same expected raw-data values the newer sensor is noisier than the older one.

5.4.4 A parametric model

The following parametric model based on the Poissonian (photon-limited) nature of the sensor (see next chapter and [63]) achieves a near-perfect fit to the experimentally measured data:

$$\sigma(y) = q\sqrt{\max(0, y-p)}, \qquad \begin{array}{ll} q_{\mathrm{U}} = 0.0060, & p_{\mathrm{U}} = 0.050, \\ q_{\mathrm{V}} = 0.0092, & p_{\mathrm{V}} = 0.021, \end{array} \tag{5.9}$$

The plots of these functions are shown, superimposed, in Figure 5.8. We remark again that the same curve $\sigma(y)$ fits equally well all four color channels. It can be seen that the newer sensor "V" presents a lower SNR, a peculiarity that can be justified by the increased pixel density[2].

Weighted least-squares fitting

The above model has been obtained by a weighted least-squares fitting of the variance in the form $\sigma^2(y) = \hat{a}y + \hat{b}$, where

$$\begin{aligned} \left[\begin{array}{cc} \hat{a} & \hat{b} \end{array}\right] &= \underset{a,b}{\operatorname{argmin}} \left([a\ b]\, \mathbf{\Phi}^T - \hat{\mathbf{v}} \right) \mathbf{W} \left([a\ b]\, \mathbf{\Phi}^T - \hat{\mathbf{v}} \right)^T = \\ &= \hat{\mathbf{v}} \mathbf{W} \mathbf{\Phi} \left(\mathbf{\Phi}^T \mathbf{W} \mathbf{\Phi} \right)^{-1}, \end{aligned} \tag{5.10}$$

$$\mathbf{\Phi} = \left[\begin{array}{cc} y_1 & 1 \\ y_2 & 1 \\ \vdots & \vdots \end{array}\right], \quad \hat{\mathbf{v}} = \left[\begin{array}{ccc} \hat{\sigma}^2(y_1) & \hat{\sigma}^2(y_2) & \cdots \end{array}\right], \quad \mathbf{W} = \left[\begin{array}{ccc} w_1 & 0 & \cdots \\ 0 & w_2 & \ddots \\ \vdots & \ddots & \ddots \end{array}\right],$$

and where the weight $w_i = M_i$, i.e. w_i is equal to the number of pixels in the segment $S_\Delta(y_i)$. Equation (5.10) is thus equivalent to

$$\left\{\hat{a}, \hat{b}\right\} = \underset{a,b}{\operatorname{argmin}} \sum_i \left(a y_i + b - \hat{\sigma}^2(y_i) \right)^2 w_i,$$

and the parameters q and p are obtained from $\hat{a}$ and $\hat{b}$ as

$$q = \sqrt{\hat{a}}, \qquad p = -\frac{b}{a}.$$

A much more sophisticated model fitting procedure is presented in the next chapter.

5.5 Conclusions to the chapter

We have presented a novel method for the accurate measurement of the standard-deviation of the temporal noise in the raw-data of digital imaging sensors. The

[2] Note that this does not imply that a picture of the same scene taken with sensor "V" is going to be noisier than that taken with sensor "U".

Plots such as those in Figure 5.8 may serve for a direct, practical, and meaningful comparison of the noisiness of two sensors only if they correspond to measurements taken with a pair of *"matching" analog gain parameters*. We say that the analog gain parameters match if, when acquiring two images (one with each of the sensors) of the *same scene* with *same exposure time* and the *same optical aperture*, the two raw images cover the *same range of normalized raw-data* values (or, equivalently, the two normalized histograms match).

method is based on an automatic segmentation of the recorded images. It allows to use non-uniform targets and illumination, thus can be implemented easily without special calibrated test charts and lighting. While previous techniques estimate only a single value for the standard deviation for a given uniform raw-data, the proposed method can measure a complete standard-deviation vs. expectation curve in a single experiment. The measured standard-deviation curve is nonparametric (pointwise estimation for every value of y). However, as seen from the examples in Figures 5.7 and 5.8, the curve allows for a simple and accurate parametric fit. In the next chapter, we present a parametric model for raw-data noise and an algorithm which can estimate the model parameters from a single noisy image. In this way we overcome the main drawback of the method proposed in the present chapter, i.e., the the need of multiple shots.

Chapter 6

Practical Poissonian-Gaussian noise modeling and fitting for single image raw-data

We present a simple noise model which can accurately be used for the raw-data. Based on the above considerations, it is a signal-dependent noise model based on a *Poissonian* part, modeling the photon sensing, and *Gaussian* part, for the remaining stationary disturbances in the output data. We explicitly take into account the problem of *clipping* (over- and underexposure), faithfully reproducing the nonlinear response of the sensor. Only two parameters are sufficient to fully describe the model. These parameters are explained in relation to the sensor's hardware characteristics (quantum efficiency, pedestal, gain). As a second and most important contribution, we propose an algorithm for the fully automatic estimation of the model parameters given a *single* noisy image.

The chapter is organized as follows. In Section 6.1 we present the model in its basic form, which ignores the clipping. The parameter estimation algorithm is then presented in Section 6.2. The general model with clipping requires more involved mathematics, and it is given in Section 6.3, followed by the modified estimation algorithm in Sections 6.4 and 6.5. Throughout these sections, we demonstrate the accuracy of the algorithm with synthetic test images, for which the exact noise parameters are known. Experiments with real raw-data are presented in Section 6.6; these experiments prove the *practical* applicability of the method and confirm that the raw-data noise can indeed be accurately modeled as a clipped Poissonian-Gaussian process. Further comments and details on the algorithm and its implementation are given in Section 6.7.

6.1 Poissonian-Gaussian modeling

Let us consider again the generic signal-dependent noise observation model of the form

$$z(x) = y(x) + \sigma(y(x))\,\xi(x) \tag{6.1}$$

where $x \in X$ is the pixel position in the domain X, z is the observed (recorded) signal, y is the original (unknown) signal, ξ is zero-mean independent random noise with standard deviation equal to 1, and $\sigma : \mathbb{R} \to \mathbb{R}^+$ is a function of y that gives the standard deviation of the overall noise component. Throughout the chapter, we denote the expected value (or mathematical expectation) of a random variable as $E\{\cdot\}$, its variance as $\mathrm{var}\{\cdot\}$, and its standard deviation as $\mathrm{std}\{\cdot\} = \sqrt{\mathrm{var}\{\cdot\}}$; when any of these operators is applied to a sequence (resp. matrix) of random variables, its output is defined as the sequence (resp. matrix) of the operator's outputs for the individual random variables. The symbol σ is used exclusively to denote this function of the model (6.1). From $E\{\xi(x)\} = 0$ follows that $E\{z(x)\} = y(x)$, i.e. the original signal can be defined as the expected value of the noisy observations. Consequently, we have that $\mathrm{std}\{z(x)\} = \sigma(E\{z(x)\})$, i.e. the standard deviation of the noise is a function, namely σ, of the expectation of the noisy signal.

In our modeling, we assume that the noise term is composed of two mutually independent parts, a *Poissonian* signal-dependent component η_{p} and a *Gaussian* signal-independent component η_{g}:

$$\sigma(y(x))\,\xi(x) = \eta_{\mathrm{p}}(y(x)) + \eta_{\mathrm{g}}(x). \tag{6.2}$$

In terms of distributions, these two components are characterized as follows,

$$\chi\left(y(x) + \eta_{\mathrm{p}}(y(x))\right) \sim \mathcal{P}(\chi y(x)), \quad \eta_{\mathrm{g}}(x) \sim \mathcal{N}(0, b),$$

where $\chi > 0$ and $b \geq 0$ are scalar parameters and $\mathcal{P}$ and $\mathcal{N}$ denote the Poisson and normal (i.e., Gaussian) distributions. From the elementary properties of the Poisson distribution, we obtain the following equation for the mean and variance

$$E\left\{\chi\left(y(x) + \eta_{\mathrm{p}}(y(x))\right)\right\} = \mathrm{var}\left\{\chi\left(y(x) + \eta_{\mathrm{p}}(y(x))\right)\right\} = \chi y(x).$$

Since

$$E\left\{\chi\left(y(x) + \eta_{\mathrm{p}}(y(x))\right)\right\} = \chi y(x) + \chi E\left\{\eta_{\mathrm{p}}(y(x))\right\}$$

and

$$\chi^2\,\mathrm{var}\left\{\eta_{\mathrm{p}}(y(x))\right\} = \chi y(x),$$

it follows that

$$E\left\{\eta_{\mathrm{p}}(y(x))\right\} = 0 \quad \text{and} \quad \mathrm{var}\left\{\eta_{\mathrm{p}}(y(x))\right\} = y(x)/\chi.$$

Thus, the Poissonian η_{p} has varying variance that depends on the value of $y(x)$, $\mathrm{var}\left\{\eta_{\mathrm{p}}(y(x))\right\} = a y(x)$, where $a = \chi^{-1}$. The Gaussian component η_{g} has instead constant variance equal to b.

Consequently, the overall variance of z in (6.1) has the affine form

$$\sigma^2(y(x)) = a y(x) + b, \tag{6.3}$$

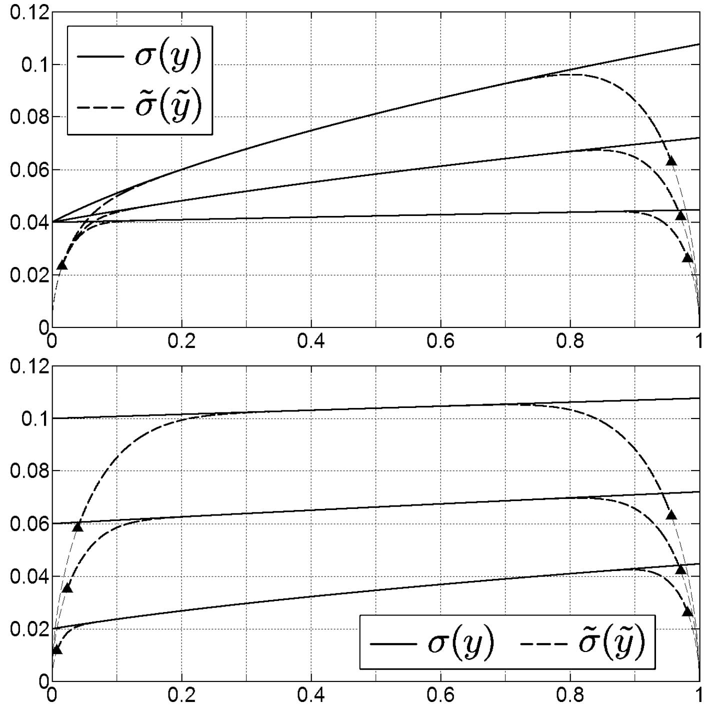


Figure 6.1: Some examples of the standard-deviation functions σ (solid lines) from the model (6.1) for different combinations of the constants a and b of Equation (6.4): (top) $a = 0.02^2, 0.06^2, 0.10^2$, $b = 0.04^2$ and (bottom) $a = 0.4^2$, $b = 0.02^2, 0.06^2, 0.10^2$. The dashed lines show the corresponding functions $\tilde{\sigma}$ of the clipped observation model (6.30), as functions of the clipped $\tilde{y}$ (see Section 6.3). The small black triangles indicate the points $(\tilde{y}, \tilde{\sigma}(\tilde{y}))$ which correspond to $y = 0$ and $y = 1$.

which gives the standard deviation σ as the square root

$$\sigma(y(x)) = \sqrt{ay(x) + b}, \tag{6.4}$$

and, in particular, $\sigma(0) = \sqrt{b}$ and $\sigma(1) = \sqrt{a + b}$.

Some examples of standard-deviation functions σ for different combinations of the constants a and b are shown, as an illustration, in Figure 6.1 (solid lines).

Figure 6.2 presents a simple piecewise smooth image which is degraded by Poissonian and Gaussian noise with parameters $\chi = 100$ ($a = 0.01$) and $b = 0.04^2$. As illustrated in Figure 6.1, these parameters imply that the noise standard-deviation in the brightest parts of the image is more than twice as large as in the darker ones.

6.1.1 Raw-data modeling

The Poissonian-Gaussian model (6.1-6.2) is naturally suited for the raw-data of digital imaging sensors. The Poissonian component η_{p} models the signal-dependent part of the errors, which is essentially due to the photon-counting process, while

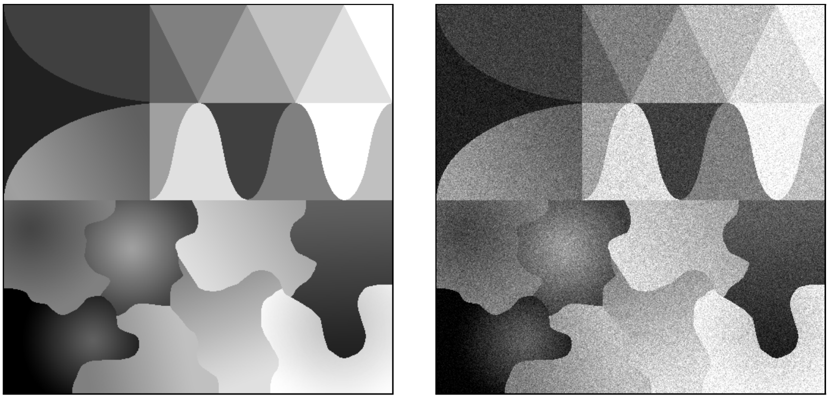


Figure 6.2: A piecewise smooth test image of size 512×512: original y and observation z degraded by Poissonian and Gaussian noise with parameters $\chi = 100$ ($a = 0.01$) and $b = 0.04^2$.

the Gaussian η_{g} accounts for the signal-independent errors such as electric and thermal noise. We briefly mention how the above model parameters relate to elementary aspects of the digital sensor's hardware.

Quantum efficiency

The parameter χ of η_{p} is related to the quantum efficiency of the sensor: the larger the number of photons necessary to produce a response of the sensor (generation of an electron), the smaller the χ.

Pedestal parameter

In digital imaging sensors, the collected charge is always added to some base "pedestal" level $p_0 \in \mathbb{R}^+$. This constitutes an offset-from-zero of the output data and it can be rewritten as a shift in the argument of the signal-dependent part of the noise:

$$\begin{aligned} z(x) &= y(x) + \sigma(y(x) - p_0)\,\xi(x) = \\ &= y(x) + \eta_{\mathrm{p}}(y(x) - p_0) + \eta_{\mathrm{g}}(x)\,. \end{aligned}$$

Analog gain

We model the analog gain as an amplification of the collected charge. Let us denote the variables before amplification by the circle superscript $^{\circ}$,

$$\mathring{z}(x) = \mathring{y}(x) + \mathring{\eta}_{\mathrm{p}}(\mathring{y}(x) - p_0) + \mathring{\eta}_{\mathrm{g}}(x)\,.$$

We formalize the amplification Θ of $\mathring{z}$ as the multiplication of the noise-free signal, of the Poissonian noise, and of a part of the Gaussian noise, by a scaling constant $\theta > 1$,

$$z(x) = \Theta(\mathring{z}(x)) = \theta\left(\mathring{y}(x) + \mathring{\eta}_{\mathrm{p}}(\mathring{y}(x) - p_0) + \mathring{\eta}'_{\mathrm{g}}(x)\right) + \mathring{\eta}''_{\mathrm{g}}(x).$$

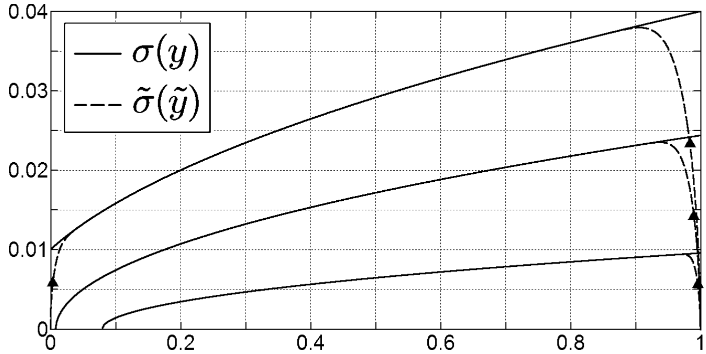


Figure 6.3: Some examples of the standard-deviation functions σ (solid lines) which are often found for the raw data. In these three examples the parameters (a, b) from Equation (6.4) are $\left(1.5 \cdot 10^{-3}, 10^{-4}\right)$, $\left(6 \cdot 10^{-4}, -5 \cdot 10^{-6}\right)$, and $\left(10^{-4}, -8 \cdot 10^{-6}\right)$. The dashed lines show the corresponding functions $\tilde{\sigma}$ of the clipped observation model (6.30), as functions of the clipped $\tilde{y}$ (see Section 6.3). The small black triangles indicate the points $(\tilde{y}, \tilde{\sigma}(\tilde{y}))$ which correspond to $y = 0$ and $y = 1$.

Here, the Gaussian noise term $\mathring{\eta}_g$ has been split in two components $\mathring{\eta}_{\rm g}'$ and $\mathring{\eta}_{\rm g}''$, $\mathring{\eta}_{\rm g}' + \mathring{\eta}_{\rm g}'' = \eta_{\rm g}$, where $\mathring{\eta}_{\rm g}''$ represents the portion of the noise that is introduced after the amplification and thus not affected by the factor θ. The expectation and variance for z are

$$\begin{aligned} E\{z(x)\} &= y(x) = \theta \mathring{y}(x), \\ \operatorname{var}\{z(x)\} &= \theta^2 \chi^{-1} (\mathring{y}(x) - p_0) + \theta^2 \operatorname{var}\left\{\mathring{\eta}_{\rm g}'(x)\right\} + \operatorname{var}\left\{\mathring{\eta}_{\rm g}''(x)\right\}. \end{aligned}$$

Hence, we come again to a model of the form (6.3)-(6.4) with

$$a = \chi^{-1}\theta, \quad b = \theta^2 \operatorname{var}\left\{\mathring{\eta}_{\rm g}'(x)\right\} + \operatorname{var}\left\{\mathring{\eta}_{\rm g}''(x)\right\} - \theta^2 \chi^{-1} p_0.$$

Note that now this b can be negative, provided a large pedestal p_0 and a small variance of $\mathring{\eta}_{\rm g}$. This does not mean that there is a "negative" variance. Indeed, because of the pedestal, $y \geq \theta p_0$ and therefore $ay + b \geq 0$.

In digital cameras, the analog gain (i.e., θ) is usually controlled by the choice of the ISO sensitivity setting. This can be done manually by the user, or automatically by the camera ("auto mode"). Large ISO numbers (e.g., 800 or 1600) correspond to large θ, and thus worse signal-to-noise ratio (SNR). Lower values (e.g., ISO 50) yield a better SNR but at the same time produce darker images, unless these are taken with a longer exposure time (which corresponds to having larger values of $\mathring{y}$ before the multiplication by θ).

Figure 6.3 shows few examples of the standard-deviation functions σ which can typically be found for the raw data. Two of these examples have $b < 0$, which corresponds to a pedestal $p_0 > 0$.

6.1.2 Heteroskedastic normal approximation

Throughout the following sections, we need to derive a few results and relations which depend not only on the mean and variance, but also on the particular

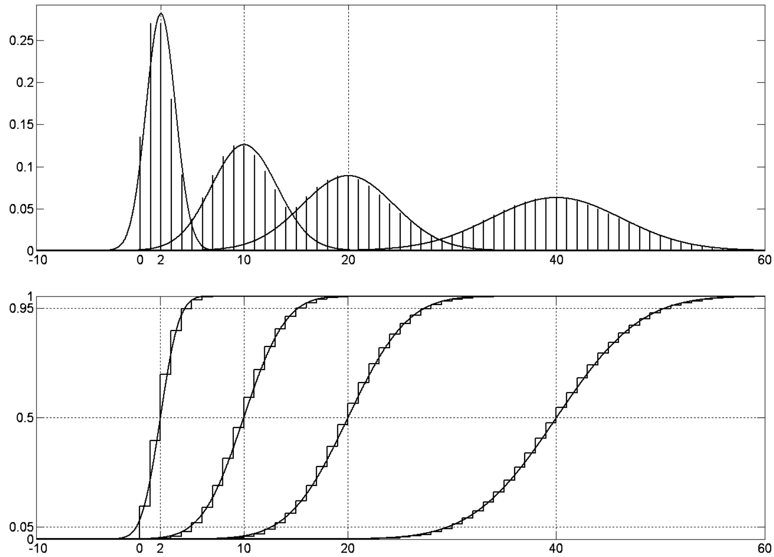


Figure 6.4: Probability densities (top) and distributions (bottom) for Poisson $\mathcal{P}(\lambda)$ and Gaussian $\mathcal{N}(\lambda,\lambda)$ random variables, $\lambda = 2, 10, 20, 40$. Since the Poisson distribution is discrete, in the top chart are shown scaled Dirac-impulses (discrete weights) and not the density itself (which is not defined as an ordinary function).

distribution of the processed samples. For the sake of simplification, we exploit the usual normal approximation of the Poisson distribution, which gives

$$\mathcal{P}(\lambda) \approx \mathcal{N}(\lambda,\lambda). \tag{6.5}$$

The accuracy of this approximation increases with the parameter λ and in practice, for large enough[1] λ, a Poissonian process can be treated as a special heteroskedastic Gaussian one. We thus obtain the following normal approximations of the errors

$$\sigma(y(x))\,\xi(x) = \sqrt{ay(x)+b}\,\xi(x) \simeq \eta_{\mathrm{h}}(y(x)), \tag{6.6}$$

where $\eta_{\mathrm{h}}(x) \sim \mathcal{N}(0, ay(x)+b)$.

Figure 6.4 illustrates the normal approximation (6.5) for a few values of the parameter λ.

[1] How large λ is enough really depends on the considered application and desired accuracy. The fact that the Poisson distribution is discrete is a secondary aspect, because quantization of the digital data makes anyway discrete also the errors due to continuous distributions.

For the considered standard-deviation estimation problem, we found experimentally that already with $\lambda = 10$ (corresponding to $\chi = 20$ for the middle intensity $y = 0.5$) there is virtually no difference between the estimation accuracy of a truly Poissonian variable and that of its Gaussian approximation.

6.2 The algorithm

Our goal is to estimate the function $\sigma : \mathbb{R} \to \mathbb{R}^+$ of the observation model (6.1) from a noisy image z. The proposed algorithm is divided in two main stages: *local estimation* of multiple expectation/standard-deviation pairs and *global parametric model fitting* to these local estimates. An initial preprocessing stage, in which the data is transformed to the wavelet domain and then segmented into non-overlapping level sets where the data is smooth, precedes the estimation.

6.2.1 Wavelet domain analysis

Similar to [36], we facilitate the noise analysis by considering wavelet detail coefficients z^{wdet} defined as the downsampled convolution

$$z^{\text{wdet}} = \downarrow_2 (z \circledast \psi),$$

where ψ is a 2-D wavelet function with zero mean and unity ℓ^2-norm, $\sum \psi = 0$, $\|\psi\|_2 = 1$, and $\downarrow_2$ denotes the decimation operator that discards every second row and every second column. Analogously, we define the normalized approximation coefficients as

$$z^{\text{wapp}} = \downarrow_2 (z \circledast \varphi),$$

where φ is the corresponding 2-D wavelet scaling function, which we specially normalize so that $\sum \varphi = 1$.

For noisy images, the detail coefficients z^{wdet} contain mostly noise and, due to the normalizations of the convolution kernels, we have

$$\begin{aligned} \operatorname{std}\left\{z^{\text{wdet}}\right\} &= \downarrow_2 (\operatorname{std}\{z \circledast \psi\}) = \downarrow_2 \left(\sqrt{\operatorname{var}\{z\} \circledast \psi^2}\right) \simeq &\quad (6.7)\\ &\simeq \downarrow_2 (\operatorname{std}\{z\} \|\psi\|_2) = \downarrow_2 (\operatorname{std}\{z\}) = \\ &= \downarrow_2 (\sigma(y)) = \sigma(\downarrow_2 y) = \sigma\left(\downarrow_2 \left(y \textstyle\sum \varphi\right)\right) \simeq \\ &\simeq \sigma(\downarrow_2 (y \circledast \varphi)) = \sigma\left(E\left\{z^{\text{wapp}}\right\}\right), &\quad (6.8) \end{aligned}$$

with the approximate equalities $\simeq$ becoming accurate at points in regions where y (and hence $\operatorname{std}\{z\}$) is uniform, as we can assume that the distribution of z does not change over the small support of the wavelets. Thus, in particular, at a point x in such uniform regions, we can assume that

$$z^{\text{wdet}}(x) \sim \mathcal{N}\left(0, \sigma\left(E\left\{z^{\text{wapp}}(x)\right\}\right)\right), \tag{6.9}$$

and, because of decimation and orthogonality properties of wavelet functions, that the noise degrading z^{wdet}, as well as the noise degrading z^{wapp}, are independent ones.

Note that, always, $\|\varphi\|_2 \neq 1$. Therefore, when considering $\operatorname{std}\{z^{\text{wapp}}\}$, the above equations can be repeated, replacing ψ with φ, only provided that the factor $\|\varphi\|_2$ is kept. Thus, we come to

$$\operatorname{std}\left\{z^{\text{wapp}}\right\} \simeq \|\varphi\|_2 \, \sigma\left(z^{\text{wapp}}\right).$$

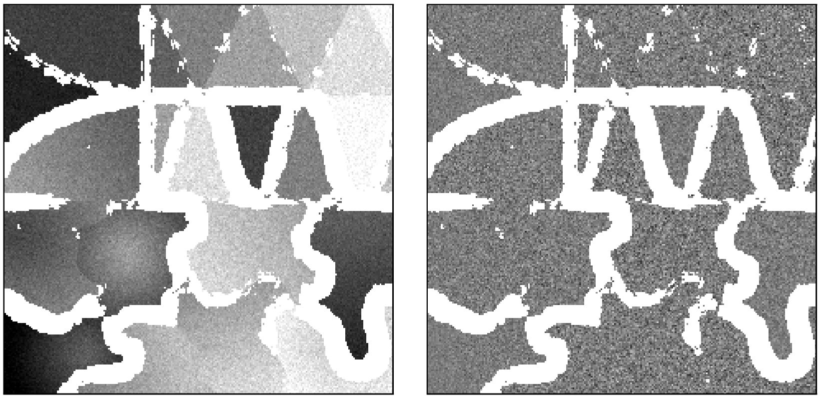


Figure 6.5: Wavelet approximation and detail coefficients z^{wapp} and z^{wdet}, restricted on the set of smoothness X^{smo}. The scale of this figure is half that of Fig. 6.2.

In our implementation, we use separable kernels $\psi = \psi_1 \circledast \psi_1^T$ and $\varphi = \varphi_1 \circledast \varphi_1^T$ where ψ_1 and φ_1 are 1-D Daubechies wavelet and scaling functions

$$\begin{aligned} \psi_1 &= [0.035 \;\; 0.085 \;\; -0.135 \;\; -0.460 \;\; 0.807 \;\; -0.333]\,, \\ \varphi_1 &= [0.025 \;\; -0.060 \;\; -0.095 \;\; 0.325 \; 0.571 \;\; 0.235]\,. \end{aligned} \tag{6.10}$$

6.2.2 Segmentation

Like in our previous work [48], we segment the data into level sets, in each of which the image can be reasonably assumed to be uniformly close to a certain value. Having nothing but a noisy image at our disposal, we shall employ spatial smoothing (as opposed to temporal smoothing, used in [48]) in order to attenuate the noise and an edge-detector in order to stay clear from edges when analyzing the data, thus enabling the conditions (6.7)-(6.8).

There exist a myriad of different methods which can be used for smoothing or for edge detection. However, for our purposes, the following simple and *non-adaptive* methods proved adequate for all considered experimental cases.

Smoothed approximation

From z^{wapp}, we compute a smoothed (low-pass) image z^{smo},

$$z^{\text{smo}} = z^{\text{wapp}} \circledast \varpi, \tag{6.11}$$

where ϖ is positive smoothing kernel, $\varpi \geq 0$ and $\|\varpi\|_1 = 1$. The smoothing action of the kernel should be especially strong, so to effectively suppress most of the noise. In our implementation, we use a uniform 7×7 kernel for ϖ.

In the corresponding regions where y itself is smooth, z^{smo} is approximately equal to $E\{z^{\text{wapp}}\}$, and thus to $\downarrow_2 y$. This is a reasonable assumption provided that the support of ϖ does not intersect edges during the calculation of the convolution (6.11).

Edges and set of smoothness

To detect edges, we use the conventional approach where some smoothed derivatives of the image are thresholded against an estimate of the local standard deviation. Exploiting the fact that the mean of the absolute deviations of $\mathcal{N}(0,1)$ is equal to $\sqrt{2/\pi}$ [88], we can define a rough estimate of the local standard-deviations of z^{wdet} as the map

$$s=\sqrt{\frac{\pi}{2}}\left|z^{\mathrm{wdet}}\right|\circledast\varpi.$$

We define the *set of smoothness* X^{smo} as

$$X^{\mathrm{smo}}=\{x\in\downarrow_2 X:|\nabla(\Lambda(z^{\mathrm{wapp}}))(x)|+|\Lambda(z^{\mathrm{wapp}})(x)|<\tau\cdot s(x)\},\qquad(6.12)$$
$$\Lambda(z^{\mathrm{wapp}})=\nabla^2\,\mathrm{medfilt}(z^{\mathrm{wapp}}),$$

where ∇ and ∇^2 are, respectively, gradient and Laplacian operators, medfilt denotes a 3×3 median filter, $\downarrow_2 X$ is the decimated domain of the wavelet coefficients z^{wapp}, and $\tau>0$ is positive threshold constant. We realize both the Laplacian operator ∇^2 and the gradient operator ∇ as convolutions against 9×9 kernels. Thresholding the sum of the moduli of the Laplacian and of its gradient is a heuristic way to obtain "thickened" edges.

In Figure 6.5, we show the wavelet approximation and detail coefficients z^{wapp} and z^{wdet}, restricted on the set of smoothness X^{smo} (whose complement thus appears as white in the figure), calculated for the test image z of Figure 6.2. Note that some of the weakest edges have not been detected as such.

Level sets (segments)

In the set of smoothness X^{smo}, we can assume that edges of the image did not interfere with the smoothing (6.11), hence, that the conditions (6.7)-(6.8) hold and that, for $x\in X^{\mathrm{smo}}$,

$$z^{\mathrm{smo}}(x)=E\{z^{\mathrm{wapp}}(x)\}=E\{(\downarrow_2 z)(x)\}=(\downarrow_2 y)(x),$$
$$\mathrm{std}\left\{z^{\mathrm{wdet}}(x)\right\}=\mathrm{std}\{(\downarrow_2 z)(x)\}=(\downarrow_2(\sigma(y)))(x).$$

We identify in the smoothness set X^{smo} a collection of non-overlapping level sets (segments) $S_i\subset X^{\mathrm{smo}}$, $i=1,\dots,N$ of the smoothed image z^{smo}. Each level set, characterized by its centre value u_i and allowed deviation $\Delta_i>0$, is defined as

$$S_i=\{x\in X^{\mathrm{smo}}:z^{\mathrm{smo}}(x)\in[u_i-\Delta_i/2,u_i+\Delta_i/2)\}.\qquad(6.13)$$

By non-overlapping we mean that $S_i\cap S_j=\varnothing$ if $i\neq j$. In practice, one can take fixed $\Delta_i\equiv\Delta$ and equispaced $u_i\in\left\{\Delta j,\,j=1,\dots,\bar{N}=\left\lceil\Delta^{-1}\right\rceil\right\}$. Further, we require that the level sets are non-trivial, in the sense that each set S_i must contain at least two samples[2]; thus, $N\leq\bar{N}$ and $\bigcup_{i=1}^{N}S_i\subseteq X^{\mathrm{smo}}\subseteq\downarrow_2 X$. Figure 6.6 shows two of the level sets computed for the example in Figure 6.2 for $\Delta=1/300$. Observe that these sets are meager and quite fragmented.

[2] The smoothness threshold τ (6.12) can be automatically increased in the rare event of $N<2$, i.e. when there are not enough non-trivial level sets for the estimation. Note that X^{smo} is monotonically enlarging to $\downarrow_2 X$ with τ, $X^{\mathrm{smo}}\underset{\tau\to\infty}{\nearrow}\downarrow_2 X$.

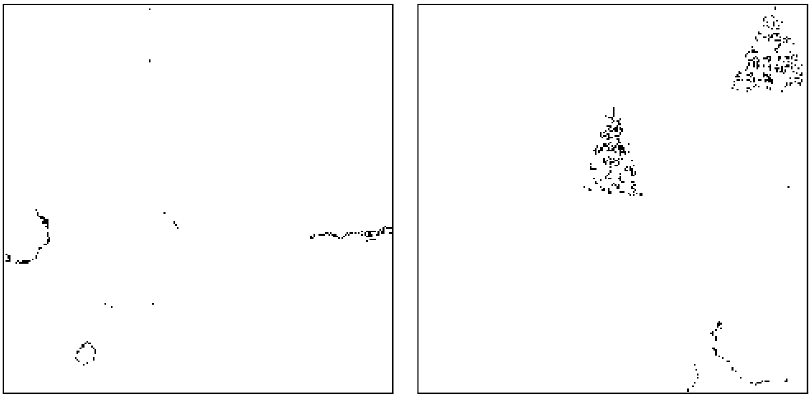


Figure 6.6: Two level-sets S_i (6.13) computed for $\Delta_i = \Delta = 1/300$. The scale of this figure is the same as that Fig. 6.5.

6.2.3 Local estimation of expectation/standard-deviation pairs

For each level set S_i, we define the (unknown) variable

$$y_i = \frac{1}{n_i}\sum_{j=1}^{n_i} E\left\{z^{\mathrm{wapp}}\left(x_j\right)\right\}, \qquad \left\{x_j\right\}_{j=1}^{n_i} = S_i. \tag{6.14}$$

Note that y_i and u_i might not coincide. The level set S_i is used as a one domain for the computation of a pair of estimates $(\hat{y}_i, \hat{\sigma}_i)$, where $\hat{y}_i$ is an estimate of y_i and $\hat{\sigma}_i$ is an estimate of $\sigma\left(y_i\right)$. In what follows, although we shall refer explicitly to y_i, this variable is always used implicitly and, in the final estimation of the function $y \mapsto \sigma\left(y\right)$, the many y_i, $i, \ldots, N$, remain "hidden" variables which are modeled as unknown. Similarly, the smoothed data z^{smo} and the values u_i and Δ_i used for the construction of S_i do not appear in the following estimation, where only z^{wdet}, z^{wapp} and S_i are used in order to compute the estimates $\hat{y}_i$ and $\hat{\sigma}_i$.

Estimation of y_i

We estimate y_i as the sample mean of the approximation coefficients z^{wapp} on S_i

$$\hat{y}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} z^{\mathrm{wapp}}\left(x_j\right), \qquad \left\{x_j\right\}_{j=1}^{n_i} = S_i. \tag{6.15}$$

Estimation of $\sigma\left(y_i\right)$

The estimate $\hat{\sigma}_i$ is calculated as the unbiased sample standard-deviation of the detail coefficients z^{wdet} on S_i

$$\hat{\sigma}_i = \frac{1}{\kappa_{n_i}}\sqrt{\frac{\sum_{j=1}^{n_i}\left(z^{\mathrm{wdet}}\left(x_j\right) - \bar{z}_i^{\mathrm{wdet}}\right)^2}{n_i - 1}}, \tag{6.16}$$

where $\bar{z}_i^{\mathrm{wdet}} = \frac{1}{n_i}\sum_{j=1}^{n_i} z^{\mathrm{wdet}}\left(x_j\right)$ and the factor $\kappa_{n_i}^{-1}$ is defined [88]

$$\kappa_n = \sqrt{\frac{2}{n-1}}\frac{\Gamma\left(\frac{n}{2}\right)}{\Gamma\left(\frac{n-1}{2}\right)} = 1 - \frac{1}{4n} - \frac{7}{32n^2} + O\left(\frac{1}{n^3}\right). \tag{6.17}$$

This factor, which comes from the mean of the chi-distribution with $n-1$ degrees of freedom, makes the estimate unbiased for normally and identically independently distributed (i.i.d.) $z^{\mathrm{wdet}}(x_j)$.

Unbiasedness

Clearly from the definition (6.14), $\hat{y}_i$ is an unbiased estimator of y_i.

The unbiasedness of $\hat{\sigma}_i$ as an estimator of $\sigma(y_i)$ is a more complex issue. As observed above, $\hat{\sigma}_i$ is an unbiased estimator of $\sigma(y_i)$ provided that z^{wdet} is normally i.i.d. on the level set S_i. However, we cannot claim, in general, that z^{wdet} is identically distributed on S_i. We remark that the assumed validity of (6.7)-(6.8) concerns individual points. It does not mean that $\mathrm{std}\left\{z^{\mathrm{wdet}}\right\}$ is constant over S_i. As a matter of fact, especially for large Δ_i, $E\left\{z^{\mathrm{wapp}}(x)\right\}$ is not constant for $x \in S_i$, which implies that the standard deviations of the wavelet detail coefficients (6.8) are not constant over S_i. Lacking any particular hypothesis on the image y, it is nevertheless reasonable to assume that $\left\{E\left\{z^{\mathrm{wapp}}(x)\right\},\ x \in S_i\right\}$ has a symmetric (discrete) distribution centred at y_i (with diameter bounded by Δ_i). Because of (6.3) and (6.9), we have that $\left\{\mathrm{var}\left\{z^{\mathrm{wdet}}(x)\right\},\ x \in S_i\right\}$ has also a symmetrical distribution, which is centred at $\mathrm{var}\{y_i\}$. This makes $\kappa_{n_i}^2\hat{\sigma}_i^2$ an unbiased[3] estimator of $\mathrm{var}\{y_i\}$ and, since $\kappa_n \underset{n\to\infty}{\to} 1$, $\hat{\sigma}_i$ is an asymptotically unbiased estimator of $\sigma(y_i)$. This asymptotic unbiasedness is relevant in the practice, since a large Δ_i corresponds to large n_i.

We further note that, despite the segmentation and removal of edges, the presence of sharp image features, singularities, or even texture in the segment S_i is not completely ruled out. This can be effectively compensated by means of non-linear robust estimators of the standard deviation, such as the well-known median of absolute deviations (MAD) [119]. For the sake of expository simplicity, in the current and in the next section we restrict ourself to the basic estimator (6.16) and postpone considerations on robust estimation of the standard-deviation to Section 6.5.

Variance of the estimates

The variance of the estimates $\hat{y}_i$ and $\hat{\sigma}_i$ depends directly on the variances of the samples used for the estimation, which are degraded by independent noise. With arguments similar to Section 6.2.3, the variances of the estimates can be expressed as

$$\mathrm{var}\{\hat{y}_i\} = \sigma^2(y_i)\,c_i, \quad \mathrm{var}\{\hat{\sigma}_i\} = \sigma^2(y_i)\,d_i, \tag{6.18}$$

$$c_i = \frac{\|\varphi\|_2^2}{n_i}, \quad d_i = \frac{1-\kappa_{n_i}^2}{\kappa_{n_i}^2} = \frac{1}{2n_i} + \frac{5}{8n_i^2} + O\left(\frac{1}{n_i^3}\right), \tag{6.19}$$

where these expressions coincide with those for the perfect case when $\mathrm{var}\left\{z^{\mathrm{wdet}}\right\}$ and $\mathrm{var}\left\{z^{\mathrm{wapp}}\right\}$ are constant on S_i [88].

[3] This can be proved easily since, for $x \in X^{\mathrm{smo}}$, we can treat $E\left\{z^{\mathrm{wdet}}(x)\right\}$ as zero.

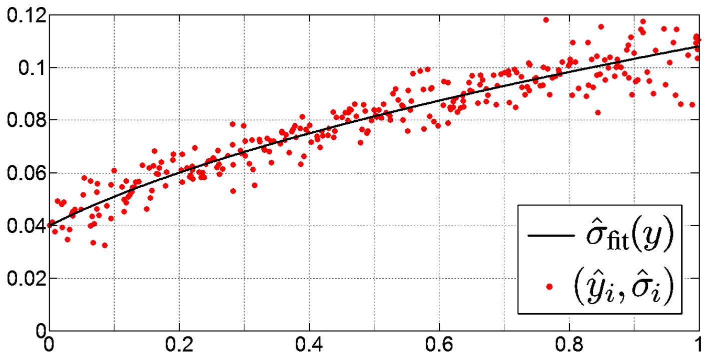


Figure 6.7: Each dot of the scatter plot corresponds to a pair $(\hat{y}_i, \hat{\sigma}_i)$ of estimates of y_i and $\sigma(y_i)$. The solid line shows the maximum-likelihood estimate $\hat{\sigma}_{\rm fit}$ of the true standard-deviation function σ. The plot of $\hat{\sigma}_{\rm fit}$ overlaps perfectly with that of the true σ (shown in Figure 6.1). The estimated parameters are $\hat{a} = 0.01008$ ($\hat{\chi} = 99.20$) and $\hat{b} = 0.001583$ ($\sqrt{\hat{b}} = 0.03979$). The initialization parameters, found as the least-squares solution (6.26), were $\hat{a}_0 = 0.00994$ ($\hat{\chi}_0 = 100.62$) and $\hat{b}_0 = 0.001649$ ($\sqrt{\hat{b}_0} = 0.04061$).

Distribution of the estimates

The estimates $\hat{y}_i$ and $\hat{\sigma}_i$ are distributed, respectively, following a normal distribution and a scaled non-central chi-distribution, which can also be approximated, very accurately for large n_i, as a normal distribution [88]. Thus, in what follows, we treat both $\hat{y}_i$ and $\hat{\sigma}_i$ as normally distributed random variables and, in particular, as

$$\hat{y}_i \sim \mathcal{N}\left(y_i, \sigma^2(y_i)\, c_i\right), \quad \hat{\sigma}_i \sim \mathcal{N}\left(\sigma(y_i), \sigma^2(y_i)\, d_i\right), \tag{6.20}$$

where c_i and d_i are defined as in (6.19).

6.2.4 Maximum-likelihood fitting of a global parametric model

The maximum-likelihood (ML) approach is used to fit a global parametric model of the function σ on the estimates $\{\hat{y}_i, \hat{\sigma}_i\}_{i=1}^N$. Depending on the parameters a and b, we have $\sigma^2(y) = ay + b$. For reasons of numerical consistency (note that formally this $\sigma^2(y)$ may be zero or negative), for the fitting we define a simple regularized variance-function $\sigma^2_{\rm reg}$ as

$$\sigma^2_{\rm reg}(y) = \max\left(\varepsilon^2_{\rm reg}, \sigma^2(y)\right) = \max\left(\varepsilon^2_{\rm reg}, ay + b\right), \tag{6.21}$$

where $\varepsilon_{\rm reg} > 0$ is a small regularization parameter. Hence, the regularized standard-deviation $\sigma_{\rm reg} = \sqrt{\sigma^2_{\rm reg}(y)}$ is always well defined, for any choice of a, b, and y.

As discussed in Section 6.2.3, we can assume normality and unbiasedness for both $\hat{y}_i$ and $\hat{\sigma}_i$. Thus, the conditional probability densities of $\hat{y}_i$ and $\hat{\sigma}_i$ given

$y_i = y$ are, respectively,

$$\begin{aligned} \wp\left(\hat{y}_i | y_i = y\right) &= \frac{1}{\sqrt{2\pi\sigma_{\text{reg}}^2(y)c_i}} e^{-\frac{1}{2\sigma_{\text{reg}}^2(y)c_i}(\hat{y}_i - y)^2}, \\ \wp\left(\hat{\sigma}_i | y_i = y\right) &= \frac{1}{\sqrt{2\pi\sigma_{\text{reg}}^2(y)d_i}} e^{-\frac{1}{2\sigma_{\text{reg}}^2(y)d_i}(\hat{\sigma}_i - \sigma_{\text{reg}}(y))^2}. \end{aligned}$$

Further, we observe that, because of the orthogonality of the wavelets, $\hat{y}_i$ and $\hat{\sigma}_i$ are mutually independent[4]. Hence,

$$\begin{aligned} &\wp\left(\left(\hat{y}_i, \hat{\sigma}_i\right) | y_i = y\right) = \wp\left(\hat{y}_i | y_i = y\right) \wp\left(\hat{\sigma}_i | y_i = y\right) = \\ &= \frac{1}{2\pi\sqrt{c_i d_i}} \frac{1}{\sigma_{\text{reg}}^2(y)} e^{-\frac{1}{2\sigma_{\text{reg}}^2(y)}\left(\frac{(\hat{y}_i - y)^2}{c_i} + \frac{(\hat{\sigma}_i - \sigma_{\text{reg}}(y))^2}{d_i}\right)}. \end{aligned} \tag{6.22}$$

Let us remark that (6.22) is a function of the parameters a and b, which define the regularized variance function σ_{reg}^2 (6.21). The posterior likelihood L is obtained by considering all measurements $\{(\hat{y}_i, \hat{\sigma}_i)\}_{i=1}^N$ and by integrating the densities $\wp\left(\left(\hat{y}_i, \hat{\sigma}_i\right) | y_i = y\right)$ with respect to a prior probability density $\wp_0(y)$ of y,

$$L(a, b) = \prod_{i=1}^{N} \int_{-\infty}^{\infty} \wp\left(\left(\hat{y}_i, \hat{\sigma}_i\right) | y_i = y\right) \wp_0(y)\, dy. \tag{6.23}$$

The integration copes with the fact that y_i and y are unknown. For images in the range $[0, 1]$, the simplest and most obvious choice is $\wp_0$ to be uniform on $[0, 1]$, which implies that (6.23) becomes $L(a, b) = \prod_{i=1}^{N} \int_0^1 \wp\left(\left(\hat{y}_i, \hat{\sigma}_i\right) | y_i = y\right) dy$. In our experiments with synthetic images we use this prior. However, we wish to note that other prior statistics have been shown to be more representative of the histograms of natural images [80]. Let us observe that $(\hat{y}_i, \hat{\sigma}_i)$ and $(\hat{y}_j, \hat{\sigma}_j)$, $i \neq j$, are mutually independent because the corresponding level sets S_i and S_j are non-overlapping.

From (6.23), the sought parameter estimates $\hat{a}$ and $\hat{b}$ are defined as the solution of

$$\begin{aligned} \left(\hat{a}, \hat{b}\right) &= \underset{a,b}{\operatorname{argmax}}\, L(a, b) = \underset{a,b}{\operatorname{argmin}} - \ln L(a, b) = \\ &= \underset{a,b}{\operatorname{argmin}} - \sum_{i=1}^{N} \ln \int \wp\left(\left(\hat{y}_i, \hat{\sigma}_i\right) | y_i = y\right) \wp_0(y)\, dy. \end{aligned} \tag{6.24}$$

Hence, our final estimate of the function σ is

$$\hat{\sigma}_{\text{fit}}(y) = \sqrt{\max\left(0, \hat{a}y + \hat{b}\right)}. \tag{6.25}$$

Figure 6.7 shows the result of the above optimization for the test example shown in Figure 6.2. It can be seen that the procedure estimates the parameters of the noise with great accuracy.

[4] This independence is a general property of the sample mean and sample standard-deviation, which property holds also when the estimates are computed from the very same samples [88]. However, by sampling two independent sets of wavelet coefficients, we have that the two estimates are necessarily independent, regardless of the particular mean and standard-deviation estimators used, a fact that comes useful for the forthcoming sections.

Iterative solution and initialization

In our implementation, we solve the problem (6.24) numerically, using the Nelder-Mead iterative downhill simplex method [122],[125] and evaluating the integrals as finite sums. As initial parameters $\hat{a}_0, \hat{b}_0$ for this iterative optimization we take the least-squares solution

$$\begin{bmatrix} \hat{a}_0 & \hat{b}_0 \end{bmatrix} = \underset{a,b}{\operatorname{argmin}} \left([a\ b]\, \mathbf{\Phi}^T - \hat{\mathbf{v}} \right) \left([a\ b]\, \mathbf{\Phi}^T - \hat{\mathbf{v}} \right)^T = \tag{6.26}$$

$$= \hat{\mathbf{v}} \mathbf{\Phi} \left(\mathbf{\Phi}^T \mathbf{\Phi} \right)^{-1}, \tag{6.27}$$

where

$$\mathbf{\Phi} = \begin{bmatrix} \hat{y}_1 & 1 \\ \hat{y}_2 & 1 \\ \vdots & \vdots \end{bmatrix}, \qquad \hat{\mathbf{v}} = \begin{bmatrix} \breve{\kappa}_{n_1}^2 \hat{\sigma}_1^2 & \breve{\kappa}_{n_2}^2 \hat{\sigma}_2^2 & \cdots \end{bmatrix}, \tag{6.28}$$

with the factors $\breve{\kappa}_n$ defined as $\breve{\kappa}_n = \kappa_n$. The linear problem (6.26) allows a simple direct solution by means of the normal equations (6.27). While in (6.24) we aim at fitting the standard-deviation curve σ_{reg} to the estimates $\{(\hat{y}_i, \hat{\sigma}_i)\}_{i=1}^N$, Equation (6.26) minimizes the residuals with respect to the variances $\hat{\sigma}^2$, treated as a linear function of the parameters a and b. Here, the factor $\breve{\kappa}_{n_i}^2$ makes $\breve{\kappa}_{n_i}^2 \hat{\sigma}_i^2$ an unbiased estimate of the variance (contrary to $\hat{\sigma}_i$ (6.16), which is an unbiased estimate of the standard deviation).

6.3 Clipping (censoring)

6.3.1 Clipped observations model

In practice, the data range, or *dynamic range*, of acquisition, transmission, and storage systems is always limited. Without loss of generality, we consider data given on the normalized range $[0, 1]$, where the extremes correspond to the maximum and minimum pixel values for the considered noisy image (e.g., raw data) format. Even if the noise-free image y is within the $[0, 1]$ range, the noise can cause z to exceed these bounds. We shall assume that values exceeding these bounds are replaced by the bounds themselves, as this corresponds to the behavior of digital imaging sensors in the case of over- or underexposure. Thus, we define the *clipped* (or censored[5]) observations $\tilde{z}$ as

$$\tilde{z}(x) = \max\left(0, \min\left(z(x), 1\right)\right), \qquad x \in X, \tag{6.29}$$

where z is given by the signal-dependent noise model (6.1). Let $\tilde{y}(x) = E\{\tilde{z}(x)\}$. The corresponding noise model for the clipped observations (6.29) is

$$\tilde{z}(x) = \tilde{y}(x) + \tilde{\sigma}(\tilde{y}(x))\, \tilde{\xi}(x), \tag{6.30}$$

[5] Strictly speaking, the form of the so-called censored samples [23] is really $\tilde{z} = z$ if $0 \leq z \leq 1$ and no sample (i.e., censoring) if $z < 0$ or $z > 1$. Usually, the amount of censored samples below and above the extrema are assumed as known (Type-1 censoring). Thus, clipped (6.29) and censored observations can, in a sense, be considered as equivalent. However, the formulas and estimators for censored variables which can be found in the literature cannot be used directly in the case of the clipped observations (6.29).

where again $E\left\{\tilde{\xi}(x)\right\}=0$, $\operatorname{var}\left\{\tilde{\xi}(x)\right\}=1$, and the function $\tilde{\sigma}:[0,1]\rightarrow\mathbb{R}^{+}$ is defined as $\tilde{\sigma}(\tilde{y}(x))=\operatorname{std}\{\tilde{z}(x)\}$.

In general, $\tilde{y}(x)=E\{\tilde{z}(x)\}\neq E\{z(x)\}=y(x)$, $\tilde{\sigma}(\tilde{y}(x))=\operatorname{std}\{\tilde{z}(x)\}\neq\operatorname{std}\{z(x)\}=\sigma(y(x))$, and, even though $\operatorname{var}\left\{\tilde{\xi}(x)\right\}=\operatorname{var}\{\xi(x)\}=1$, the distributions of ξ and $\tilde{\xi}$ are different.

In Figure 6.1 one can compare the standard-deviation functions $\tilde{\sigma}$ (dashed line) and σ (solid line) for different combinations of the constants a and b in (6.4).

In the next sections, we rely on the heteroskedastic normal approximation (6.6) and hence treat z as a purely Gaussian variable. Consequently, we model $\tilde{z}$ as a clipped (censored) normal variable. We note that this normal approximation is especially relevant for values of $y=E\{z\}$ close to 0 or 1, where the clipping effects may be dominant. For y close to 1, we have that (6.5) holds with the largest values of λ, hence is for this values of y that the Gaussianization of the Poissonian component $\eta_{\mathrm{p}}(y(x))$ is most accurate. For y close to 0, although λ in (6.5) might not be large, the approximation holds because the variance $ay(x)$ of $\eta_{\mathrm{p}}(y(x))$ becomes negligible compared to the variance b of the Gaussian part $\eta_{\mathrm{g}}(x)$. This is true provided that $b\neq0$. However, if $b=0$ the noise has only the Poissonian component η_{p}, which is always positive. It means that $z\geq0$ and, thus, $\tilde{z}=\min(1,z)$. Therefore, if $b=0$, for our purposes it is sufficient to consider only the normal approximation for y close to 1, as no clipping happens at 0.

6.3.2 Expectations, standard deviations, and their transformations

To simplify the calculations, we shall assume that the two clippings, the one from below ($z<0$, $\tilde{z}=0$) and the one from above ($z>1$, $\tilde{z}=1$), are not mixed by the randomness of the noise, and can thus be computed independently[6]. In other words, this means that, *in practice*, if y is close enough to 0 so that it is possible that $z<0$, then it is impossible that $z>1$; similarly, if y is close enough to 1, so that z can be larger than 1, then z cannot be smaller than 0; for intermediate values of y, with $0\ll y\ll1$, we have that $0<z<1$, i.e. clipping is not happening.

In what follows, we therefore treat separately the two cases:

- clipping from below (left single censoring): y and z are near 0 and $z<1$, thus, $\tilde{z}=\max(0,z)$;

- clipping from above (right single censoring): y and z are near 1 and $z>0$, thus, $\tilde{z}=\min(z,1)$.

[6]Formally, this corresponds to assuming that, for a given $y(x)$, the product probability $P(z(x)>1)\cdot P(z(x)<0)$ is negligibly small. This condition is satisfied provided, e.g., the stronger condition that $P(z(x)<0|y(x)>1-\epsilon)$ and $P(z(x)>1|y(x)<\epsilon)$ are both negligibly small for $\epsilon=0.5$. These conditions are all surely met in the practical cases, since there the standard deviation $\sigma(y(x))$ of $z(x)$ is always much smaller (in fact, several orders smaller) than 0.5 and its distribution does not have heavy tails (note that $[y-\gamma\sigma(y),y+\gamma\sigma(y)]$ with $\gamma\geq4$ can be a rather "safe" confidence interval for z, with higher than 99.99% confidence for Gaussian distributions).

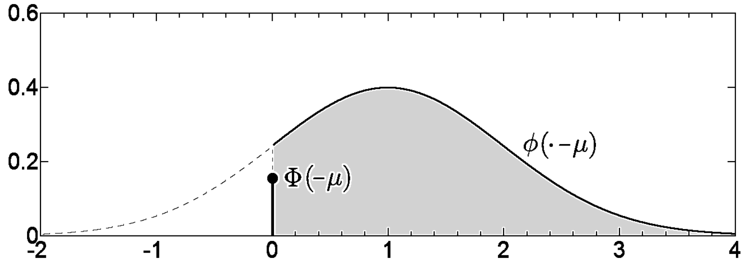


Figure 6.8: The probability density function of $\tilde{\nu} = \max(0, \nu)$, as defined by Equation (6.31). In this illustration $\mu = 1$. The height of the impulse at 0 is equal to the area under the bell curve between $-\infty$ and 0.

Further, we combine the results for the two cases, so to obtain formulas which are valid for the case $\tilde{z} = \max(0, \min(z, 1))$, where clipping can happen from above or below (double censoring).

Clipping from below (left single censoring)

Since $\tilde{z} = \max(0, z)$, we have that $E\{\tilde{z}\} \geq E\{z\} = y$.

Let $\nu \sim \mathcal{N}(\mu, 1)$ be a normally distributed random variable with mean $E\{\nu\} = \mu$ and unitary variance and $\tilde{\nu} = \max(0, \nu)$.

The probability density $f_{\tilde{\nu}}$ of $\tilde{\nu}$ is a generalized function defined as follows

$$f_{\tilde{\nu}}(t) = \begin{cases} \phi(t-\mu) + \Phi(-\mu)\,\delta_0(t) & t \geq 0, \\ 0 & t < 0, \end{cases} \tag{6.31}$$

where ϕ and Φ are the probability density and cumulative distribution functions (p.d.f. and c.d.f.) of the standard normal $\mathcal{N}(0, 1)$ and δ_0 is the Dirac delta impulse. This function is illustrated in Figure 6.8.

Tedious but simple calculations (see, e.g., [88], or [73], Chapter 20) show that the expectation $E\{\tilde{\nu}\}$ and the variance $\operatorname{var}\{\tilde{\nu}\}$ of the clipped $\tilde{\nu}$ are

$$E\{\tilde{\nu}\} = \Phi(\mu)\,\mu + \phi(\mu), \tag{6.32}$$

$$\begin{aligned}\operatorname{var}\{\tilde{\nu}\} = {} & \Phi(\mu) + \phi(\mu)\,\mu - \phi^2(\mu) + \\ & + \Phi(\mu)\,\mu\,(\mu - \Phi(\mu)\,\mu - 2\phi(\mu)).\end{aligned} \tag{6.33}$$

The plots of the expectation $E\{\tilde{\nu}\}$ and of the standard deviation $\operatorname{std}\{\tilde{\nu}\} = \sqrt{\operatorname{var}\{\tilde{\nu}\}}$ as functions $\mathcal{E}_m$ and $\mathcal{S}_m$ of $\mu = E\{\nu\}$ are shown in Figure 6.9. Observe that $E\{\tilde{\nu}\}$ is strictly positive even for negative values of μ; it is convex and increasing, $E\{\tilde{\nu}\} \to 0$ as $\mu \to -\infty$ and $E\{\tilde{\nu}\}$ is asymptotic to $\mu = E\{\nu\}$ as $\mu \to +\infty$. The standard deviation $\operatorname{std}\{\tilde{\nu}\}$ approaches $1 = \operatorname{std}\{\nu\}$ as $\mu \to +\infty$ and goes to zero as μ decreases.

The normal approximation (6.6) gives that $z \sim \mathcal{N}(y, \sigma^2(y))$. "Standardization" of the noise is obtained dividing the variables by $\sigma(y)$, which gives $\frac{z}{\sigma(y)} \sim \mathcal{N}\left(\frac{y}{\sigma(y)}, 1\right)$. It means that, by taking $\mu = \frac{y}{\sigma(y)}$, we can write $z = \sigma(y)\,\nu$, $\tilde{z} = \sigma(y)\,\tilde{\nu}$. It follows that $\tilde{y}(x) = E\{\tilde{z}(x)\} = \sigma(y)\,E\{\tilde{\nu}\}$ and $\tilde{\sigma}(\tilde{y}) = \operatorname{std}\{\tilde{z}\} = \sigma(y)\operatorname{std}\{\tilde{\nu}\}$. Exploiting this standardization, we can formulate the *direct* and *inverse* transformations which link σ and y to $\tilde{y}$ and $\tilde{\sigma}$.

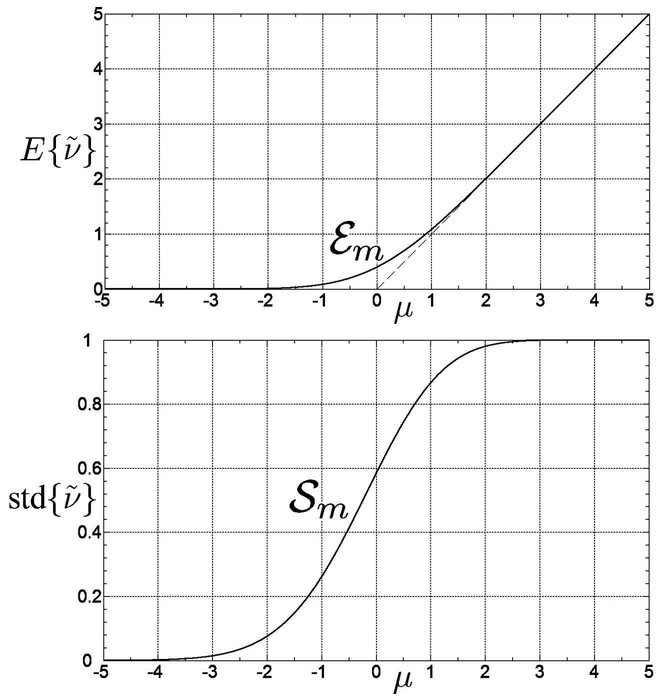


Figure 6.9: Expectation $E\{\tilde{\nu}\}$ and standard deviation std$\{\tilde{\nu}\}$ of the clipped $\tilde{\nu} = \max(0, \nu)$ as functions $\mathcal{E}_m$ and $\mathcal{S}_m$ of μ, where $\mu = E\{\nu\}$ and $\nu \sim \mathcal{N}(\mu, 1)$.

Direct transformation ($\tilde{y}$ and $\tilde{\sigma}$ from y and σ) From the above formulas we obtain

$$\tilde{y} = \sigma(y)\,\mathcal{E}_m\left(\frac{y}{\sigma(y)}\right), \tag{6.34}$$

$$\tilde{\sigma}(\tilde{y}) = \sigma(y)\,\mathcal{S}_m\left(\frac{y}{\sigma(y)}\right), \tag{6.35}$$

which give an explicit expression for the clipped observation model (6.30), provided that $\sigma(y)$ from the basic model (6.1) is known. In particular, (6.34) and (6.35) define the transformations that bring the standard deviation curve $(y, \sigma(y))$ to its clipped counterpart $(\tilde{y}, \tilde{\sigma}(\tilde{y}))$.

The two plots in Figure 6.9 can be unified, plotting std$\{\tilde{\nu}\}$ as a function of $E\{\tilde{\nu}\}$. This is shown by the function $\mathcal{S}_e$ in Figure 6.10. Naturally, between $\mathcal{S}_m$ and $\mathcal{S}_e$ there is only a change of the independent variables, $\mu \longleftrightarrow E\{\tilde{\nu}\}$, hence, from (6.35) follows that

$$\tilde{\sigma}(\tilde{y}) = \sigma(y)\,\mathcal{S}_e\left(\frac{\tilde{y}}{\sigma(y)}\right), \tag{6.36}$$

where $\tilde{y}$ can be obtained from (6.34).

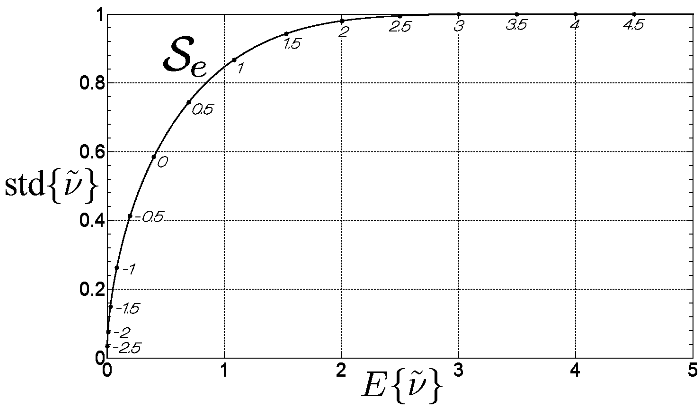


Figure 6.10: Standard deviation $\operatorname{std}\{\tilde{\nu}\}$ of the clipped $\tilde{\nu} = \max(0, \nu)$ as function $\mathcal{S}_e$ of its expectation $E\{\tilde{\nu}\}$. The numbers in italic indicate the corresponding value of μ, where $\mu = E\{\nu\}$ and $\nu \sim \mathcal{N}(\mu, 1)$.

Inverse transformation (σ **and** y **from** $\tilde{\sigma}$ **and** $\tilde{y}$) As clearly seen in Figure 6.9, the plot of $\mathcal{S}_e$ is strictly convex, which implies that the (incremental) ratio $\rho = \frac{E\{\tilde{\nu}\}}{\operatorname{std}\{\tilde{\nu}\}} = \frac{E\{\tilde{\nu}\}-0}{\operatorname{std}\{\tilde{\nu}\}-0}$ is in bijection with μ. This means that μ can be univocally determined given ρ. Note that this ratio is scale-invariant and that, in particular, $\rho = \frac{E\{\tilde{\nu}\}}{\operatorname{std}\{\tilde{\nu}\}} = \frac{\tilde{y}\sigma^{-1}(y)}{\tilde{\sigma}(\tilde{y})\sigma^{-1}(y)} = \frac{\tilde{y}}{\tilde{\sigma}(\tilde{y})}$. Therefore, given both $\tilde{y}$ and $\tilde{\sigma}(\tilde{y})$, we can obtain μ and hence also $E\{\tilde{\nu}\}$ and $\operatorname{std}\{\tilde{\nu}\}$.

In Figure 6.11 we show the plots of $\mathcal{E}_r$ and $\mathcal{S}_r$ which represent $\frac{\mu}{E\{\tilde{\nu}\}} = \frac{E\{\nu\}}{E\{\tilde{\nu}\}}$ and $\operatorname{std}\{\tilde{\nu}\}$ as functions of ρ, respectively.

From the definition of $\mathcal{E}_r$ follows that $y = E\{z\} = \sigma(y) E\{\nu\} = \sigma(y)\mu = \sigma(y) E\{\tilde{\nu}\} \mathcal{E}_r(\rho)$. Substituting $\tilde{y} = E\{\tilde{z}\} = \sigma(y) E\{\tilde{\nu}\}$ in the previous equation (observe that, at this stage, $\sigma(y)$ is considered as unknown) we obtain

$$y = \tilde{y}\mathcal{E}_r(\rho) = \tilde{y}\mathcal{E}_r\left(\frac{\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right). \tag{6.37}$$

Analogously for the standard deviation, $\sigma(y) = \operatorname{std}\{z\} = \sigma(y)\operatorname{std}\{\nu\} = \sigma(y)\frac{\operatorname{std}\{\tilde{\nu}\}}{\mathcal{S}_r(\rho)}$. Substituting $\tilde{\sigma}(\tilde{y}) = \operatorname{std}\{\tilde{z}\} = \sigma(y)\operatorname{std}\{\tilde{\nu}\}$ we have

$$\sigma(y) = \operatorname{std}\{z\} = \frac{\tilde{\sigma}(\tilde{y})}{\mathcal{S}_r(\rho)} = \frac{\tilde{\sigma}(\tilde{y})}{\mathcal{S}_r\left(\frac{\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right)}. \tag{6.38}$$

The Equations (6.37) and (6.38) define the transformation that brings the clipped standard deviation curve $(\tilde{y}, \tilde{\sigma}(\tilde{y}))$ to its non-clipped counterpart $(y, \sigma(y))$.

Clipping from above (right single censoring)

The case of clipping from above, $\tilde{z} = \min(1, z)$, can be treated exactly as the clipping from below, provided simple manipulations and the following obvious

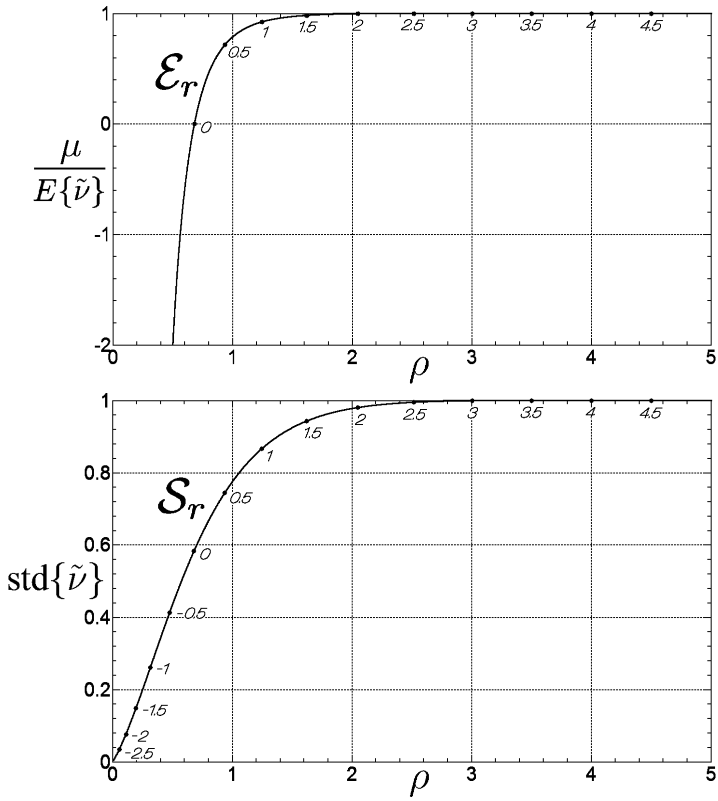


Figure 6.11: Expectation $E\{\tilde{\nu}\}$ and standard deviation $\mathrm{std}\{\tilde{\nu}\}$ of the clipped $\tilde{\nu} = \max(0, \nu)$ as functions $\mathcal{E}_r$ and $\mathcal{S}_r$ of $\rho = \frac{E\{\tilde{\nu}\}}{\mathrm{std}\{\tilde{\nu}\}}$. The numbers in italic indicate the corresponding value of μ.

change of variables:

$$\begin{aligned} y &\longleftrightarrow 1-y, \qquad z \longleftrightarrow 1-z, \\ \tilde{y} &\longleftrightarrow 1-\tilde{y}, \qquad \tilde{z} \longleftrightarrow 1-\tilde{z}. \end{aligned}$$

Direct transformation ($\tilde{y}$ and $\tilde{\sigma}$ from y and σ)

$$\tilde{y} = 1 - \sigma(y)\,\mathcal{E}_m\left(\frac{1-y}{\sigma(y)}\right), \tag{6.39}$$

$$\tilde{\sigma}(\tilde{y}) = \sigma(y)\,\mathcal{S}_m\left(\frac{1-y}{\sigma(y)}\right), \tag{6.40}$$

$$\tilde{\sigma}(\tilde{y}) = \sigma(y)\,\mathcal{S}_e\left(\frac{1-\tilde{y}}{\sigma(y)}\right). \tag{6.41}$$

Inverse transformation (σ and y from $\tilde{\sigma}$ and $\tilde{y}$)

$$y = 1 - (1-\tilde{y})\,\mathcal{E}_r\left(\frac{1-\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right), \tag{6.42}$$

$$\sigma(y) = \operatorname{std}\{z\} = \frac{\tilde{\sigma}(\tilde{y})}{\mathcal{S}_r\left(\frac{1-\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right)}, \tag{6.43}$$

Combined clipping from above and below (double censoring)

The formulas for the two separate clippings, the one from below and the one from above, can be combined into "universal" formulas which can be applied to data which is clipped in any of the two ways. Here, we undertake the assumption, discussed in Section 6.3.2, that the product probability of z being clipped both from above and from below is negligibly small.

Direct transformation ($\tilde{y}$ and $\tilde{\sigma}$ from y and σ) Since only one kind of clipping can happen for a given y, it means that either (6.34) or (6.39) is equal to y. Therefore, Equations (6.37) and (6.42) can be combined by summing the two right-hand sides and subtracting y,

$$\tilde{y} = \sigma(y)\,\mathcal{E}_m\left(\frac{y}{\sigma(y)}\right) - y + 1 - \sigma(y)\,\mathcal{E}_m\left(\frac{1-y}{\sigma(y)}\right). \tag{6.44}$$

Similarly, (6.35) and (6.40) cannot be simultaneously different than $\sigma(y)$. So do (6.36) and (6.41). It means that their combinations are simply the products of the respective factors in the right-hand sides:

$$\tilde{\sigma}(\tilde{y}) = \sigma(y)\,\mathcal{S}_m\left(\frac{y}{\sigma(y)}\right)\mathcal{S}_m\left(\frac{1-y}{\sigma(y)}\right), \tag{6.45}$$

$$\tilde{\sigma}(\tilde{y}) = \sigma(y)\,\mathcal{S}_e\left(\frac{\tilde{y}}{\sigma(y)}\right)\mathcal{S}_e\left(\frac{1-\tilde{y}}{\sigma(y)}\right). \tag{6.46}$$

Inverse transformation (σ and y from $\tilde{\sigma}$ and $\tilde{y}$) Analogous considerations hold also for combining of Equation (6.37) with (6.42) and Equation (6.38) with (6.43). Consequently, we have

$$y = \tilde{y}\mathcal{E}_r\left(\frac{\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right) - \tilde{y} + 1 - (1-\tilde{y})\,\mathcal{E}_r\left(\frac{1-\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right), \tag{6.47}$$

$$\sigma(y) = \operatorname{std}\{z\} = \frac{\tilde{\sigma}(\tilde{y})}{\mathcal{S}_r\left(\frac{\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right)\mathcal{S}_r\left(\frac{1-\tilde{y}}{\tilde{\sigma}(\tilde{y})}\right)}. \tag{6.48}$$

6.3.3 Expectation and standard deviation in the wavelet domain

All the above results are valid also in the more general case where the mean and the standard deviation are not calculated for $\tilde{\nu}$, but rather from the corresponding detail or approximation wavelet coefficients, respectively. More precisely,

$$E\{\tilde{\nu}\} = E\{\tilde{\nu} \circledast \varphi\}, \qquad \mathrm{std}\{\tilde{\nu}\} = \mathrm{std}\{\tilde{\nu} \circledast \psi\},$$

since these equalities follow from the independence of $\tilde{\nu}$ and on the normalizations $\sum \varphi = 1$ and $\|\psi\|_2 = 1$. Therefore, in the next section, we consider the wavelet coefficients calculated from the clipped observations:

$$\tilde{z}^{\mathrm{wdet}} = \downarrow_2 (\tilde{z} \circledast \psi), \qquad \tilde{z}^{\mathrm{wapp}} = \downarrow_2 (\tilde{z} \circledast \varphi).$$

6.4 Algorithm: clipped case

Our goal is to estimate the functions $\tilde{\sigma}$ and σ which correspond to the clipped observation model (6.30) from the clipped image $\tilde{z}$.

Pragmatically, we approach the problem using the estimators $\hat{y}_i$ (6.15) and $\hat{\sigma}_i$ (6.16) of mean and standard-deviation, without any particular modification. Because of clipping, these are no longer unbiased estimators of y_i and $\sigma(y_i)$. However, as discussed below, they can be treated as unbiased estimators of the unknown variable $\tilde{y}_i$, defined analogously to (6.14) as

$$\tilde{y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} E\{\tilde{z}^{\mathrm{wapp}}(x_j)\}, \qquad \{x_j\}_{j=1}^{n_i} = S_i,$$

and of its associated standard deviation $\tilde{\sigma}(\tilde{y}_i)$.

Exploiting the transformations defined in the previous section and by modeling the statistics of the estimates computed from the wavelet coefficients of clipped variables, we modify the likelihood function (6.22) and the least-squares normal equations (Section 6.2.4). Thus, we come to the desired estimates $\hat{\sigma}_{\mathrm{fit}}$ of σ and $\widehat{\tilde{\sigma}}_{\mathrm{fit}}$ of $\tilde{\sigma}$.

6.4.1 Local estimation of expectation/standard-deviation pairs

Estimate of $\tilde{\sigma}(\tilde{y}_i)$

The standard-deviation estimator (6.16) is an asymptotically (for large samples) unbiased estimator of the standard deviation regardless of their particular distribution. However, for finite samples, we can guarantee unbiasedness only when the samples are normally distributed.

On this respect, applying the estimator on the wavelet detail coefficients $\tilde{z}^{\mathrm{wdet}}$ (rather than directly on $\tilde{z}$) has the important beneficial effect of "Gaussianizing" the analyzed data, essentially by the central-limit theorem. In practice, the larger is the support of the filter ψ, the closer to a normal is the distribution of $\tilde{z}^{\mathrm{wdet}}$. To make the issue transparent, let us consider the example of a constant $y(x) \equiv y$, $\forall x \in X$, and restrict our attention to the clipping from below (single left censoring). According to the models (6.1) and (6.30), $\mathrm{var}\{z(x)\} = \sigma^2(y)$ and $\mathrm{var}\{\tilde{z}(x)\} = \tilde{\sigma}^2(\tilde{y})$ are also obviously constant. Then, provided that ψ has zero mean and $\|\psi\|_2 = 1$, we have that $\tilde{z}^{\mathrm{wdet}}$ has a distribution that approaches, for an

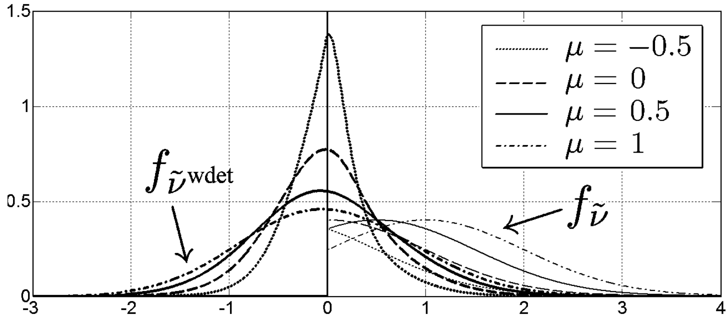


Figure 6.12: Probability densities of the clipped (from below) $\tilde{\nu} = \tilde{z}/\sigma(y)$ (thin lines) and of its wavelet detail coefficients $\tilde{\nu}^{\mathrm{wdet}} = \tilde{z}^{\mathrm{wdet}}/\sigma(y) = \downarrow_2 (\tilde{z} \circledast \psi)/\sigma(y)$ (thick lines) for different values of $\mu = y/\sigma(y)$, when ψ is the 2-D Daubechies wavelet. The "Dirac peaks" at 0, characteristic of these densities, appear here as vertical asymptotes at 0 and cannot thus be seen in the drawing.

enlarging support of ψ, the normal distribution $\mathcal{N}(0, \tilde{\sigma}^2)$. Indeed, the probability density $f_{\tilde{z}^{\mathrm{wdet}}}$ of $\tilde{z}^{\mathrm{wdet}}$ can be calculated as the generalized cascaded convolutions of the densities $f_{\psi(j)\tilde{z}}$, $j = 1, \ldots, n_\psi$ of n_ψ clipped normal distributions, where n_ψ is the number of non-zero elements $\psi(\cdot)$ of the wavelet ψ. We remark that all these densities are generalized functions with a scaled Dirac impulse at 0. From (6.31), we have that the impulse in $f_{\psi(j)\tilde{z}}$ is $\Phi(-y/\sigma(y))\,\delta_0$ (note that the scale of the impulse does not depend on ψ). Because of the independence of $\tilde{z}$, the probability $P(\tilde{z}^{\mathrm{wdet}} = 0)$ is the product probability $\prod_j P(\psi(j)\,\tilde{z} = 0) = \Phi(-y/\sigma(y))^{n_\psi}$, thus the impulse in $f_{\tilde{z}^{\mathrm{wdet}}}$ is $\Phi(-y/\sigma(y))^{n_\psi}\,\delta_0$, showing that the discrete part of the distribution vanishes at exponential rate with n_ψ. The convergence to a normal distribution is rather fast, and even for small wavelet kernels such as $\psi = \psi_1 \circledast \psi_1^T$ with ψ_1 defined as (6.10), for which $n_\psi = 36$, the distribution of $\tilde{z}^{\mathrm{wdet}}$ is very similar to a normal for values $\mu = \frac{y}{\sigma(y)}$ as low as 0 (observe that the larger is μ, the closer is the normal approximation), as shown in Figure 6.12. Note that for $\mu = -0.5, 0, 0.5, 1$ and $n_\psi = 36$, the amplitudes of the step discontinuity at 0 in the distribution of $\tilde{z}^{\mathrm{wdet}}$ are $\Phi(\mu)^{n_\psi} \simeq 1.7 \cdot 10^{-6}, 1.5 \cdot 10^{-11}, 4.1 \cdot 10^{-19}, 1.6 \cdot 10^{-29}$, respectively, thus all these distributions are practically continuous and therefore the plots in the figure are a faithful illustration of the generalized probability densities $f_{\tilde{z}^{\mathrm{wdet}}}$.

The described "Gaussianization" is important, because it ensures that the bias due to finite samples is not significant, allowing to use the same constant κ_n (6.17) as in the non-clipped case. As a rough quantitative figure of the error which may come from this simplification, in Table 6.1 we give the values of the expectation[7] $E\{\hat{\sigma}_i\}/\sigma(y_i)$ for different combinations of $\mu = y/\sigma(y)$ and n_i.

[7] The finite-sample numbers in Tables 6.1 and 6.3 are obtained by Monte Carlo simulations. The simulations were computed with enough replications to have a sample standard-deviation of the averages lower than 0.0001. Thus, the numbers given in the tables can be considered as precise for all shown digits. The taken samples z^{wdet} were contiguous in the set S_i, therefore some dependence was present (exact independence is found only for samples farther than the diameter of the support of ψ, because in this case the distribution of z is not normal), however, as n_i grows the dependence becomes negligible, since z^{wdet} can be split in at most $n_\psi/4 = 9$

n_i	$E\{\hat{\sigma}_i\}/\sigma(y_i)$ for $n<\infty$, $\tilde{\sigma}(\tilde{y})/\sigma(y)=\text{std}\{\tilde{z}\}/\sigma(y)=\mathcal{S}_m(y/\sigma(y))$ for $n=\infty$									
	$\mu=-1$	$\mu=-0.5$	$\mu=0$	$\mu=0.5$	$\mu=1$	$\mu=1.5$	$\mu=2$	$\mu=2.5$	$\mu=3$	$\mu=5$
2	0.226	0.389	0.572	0.741	0.868	0.946	0.982	0.995	0.999	1.000
10	0.247	0.404	0.580	0.743	0.867	0.944	0.981	0.995	0.999	1.000
20	0.253	0.408	0.581	0.743	0.867	0.943	0.980	0.994	0.999	1.000
50	0.258	0.411	0.583	0.744	0.867	0.943	0.980	0.994	0.999	1.000
100	0.260	0.412	0.583	0.744	0.867	0.943	0.980	0.994	0.999	1.000
200	0.261	0.412	0.584	0.744	0.867	0.943	0.980	0.994	0.999	1.000
500	0.261	0.413	0.584	0.744	0.867	0.943	0.980	0.994	0.999	1.000
∞	0.262	0.413	0.584	0.744	0.867	0.943	0.980	0.994	0.999	1.000

Table 6.1: Expectation $E\{\hat{\sigma}_i\}/\sigma(y_i)$ for different combinations of $\mu=y_i/\sigma(y_i)$ and n_i. The cases "$n_i=\infty$" correspond to the true values of the standard deviation $\tilde{\sigma}(\tilde{y})/\sigma(y)=\text{std}\{\tilde{z}\}/\sigma(y)=\text{std}\{\tilde{\nu}\}=\mathcal{S}_m(\mu)=\mathcal{S}_m(y/\sigma(y))$ of the clipped data, calculated from (6.33) and plotted in Figure 6.9.

The cases "$n_i=\infty$" correspond to the true values of the standard deviation $\tilde{\sigma}(\tilde{y})/\sigma(y)=\text{std}\{\tilde{z}\}/\sigma(y)=\text{std}\{\tilde{\nu}\}=\mathcal{S}_m(\mu)=\mathcal{S}_m(y/\sigma(y))$ of the clipped data, calculated from (6.33) and plotted in Figure 6.9. From the table one can see that a handful of samples are sufficient for the finite-sample estimation bias $E\{\hat{\sigma}_i\}-\text{std}\{\tilde{\nu}\}$ to be negligible.

Estimate of $\tilde{y}_i$

Let us now consider the estimates of the mean. Clearly, being a sample average, $\hat{y}_i$ is an unbiased estimate of $\tilde{y}_i$, regardless of the number of samples n_i or of the distribution of $\tilde{z}$. The central-limit theorem and similar arguments as above show that $\tilde{z}^{\text{wapp}}$ and $\hat{y}_i$ are both normally distributed with mean $\tilde{y}_i$.

Variance of the estimates

Ignoring the possible dependence of the noise in the wavelet coefficients (due to non-Gaussianity of the clipped variables), simple estimates of the variances of $\hat{y}_i$ and $\hat{\sigma}_i$ can be obtained from the variances $\|\varphi\|_2^2\,\tilde{\sigma}^2(\tilde{y}_i)$ and $\tilde{\sigma}^2(\tilde{y}_i)$ of the wavelet coefficients $\tilde{z}^{\text{wapp}}$ and $\tilde{z}^{\text{wdet}}$, respectively, as in Section 6.2.3.

Distribution of the estimates

In conclusion, similar to Section 6.2.3, we model the distributions of the estimates $\hat{y}_i$ and $\hat{\sigma}_i$ as the normal

$$\hat{y}_i\sim\mathcal{N}\left(\tilde{y}_i,\tilde{\sigma}^2(\tilde{y}_i)\,c_i\right),\quad \hat{\sigma}_i\sim\mathcal{N}\left(\tilde{\sigma}(\tilde{y}_i),\tilde{\sigma}^2(\tilde{y}_i)\,d_i\right),\tag{6.49}$$

where the factors c_i and d_i are defined as in (6.19).

6.4.2 Maximum-likelihood fitting of the clipped model

It is straightforward to exploit the above analysis for the estimation of the functions $\tilde{\sigma}$ (6.30) and σ (6.1) from the clipped data $\tilde{z}=\max(0,\min(z,1))$. In fact, for

subsets, each with a growing number of fully independent samples.

the ML solution (6.24), it suffices to introduce the functions $\mathcal{E}_m$ and $\mathcal{S}_m$ into the definition of the function to be fitted to the measured data, which are pairs $(\hat{y}_i, \hat{\sigma}_i)$ centered–according to (6.49)–at $(\tilde{y}_i, \tilde{\sigma}(\tilde{y}_i))$. From (6.45) follows that we can define $\tilde{\sigma}_{\text{reg}}(\tilde{y})$ as

$$\tilde{\sigma}_{\text{reg}}(\tilde{y}) = \sigma_{\text{reg}}(y)\, \mathcal{S}_m\left(\frac{y}{\sigma_{\text{reg}}(y)}\right) \mathcal{S}_m\left(\frac{1-y}{\sigma_{\text{reg}}(y)}\right), \tag{6.50}$$

where the argument $\tilde{y}$ is, according to (6.44),

$$\tilde{y} = \sigma_{\text{reg}}(y)\, \mathcal{E}_m\left(\tfrac{y}{\sigma_{\text{reg}}(y)}\right) - y + 1 - \sigma_{\text{reg}}(y)\, \mathcal{E}_m\left(\tfrac{1-y}{\sigma_{\text{reg}}(y)}\right). \tag{6.51}$$

The conditional probability density (6.22) is thus modified into

$$\begin{aligned} \wp\left((\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right) &= \wp\left(\hat{y}_i|\tilde{y}_i = \tilde{y}\right) \wp\left(\hat{\sigma}_i|\tilde{y}_i = \tilde{y}\right) = \\ &= \frac{1}{2\pi\sqrt{c_i d_i}} \frac{1}{\tilde{\sigma}^2_{\text{reg}}(\tilde{y})} e^{-\frac{1}{2\tilde{\sigma}^2_{\text{reg}}(\tilde{y})}\left(\frac{(\hat{y}_i-\tilde{y})^2}{c_i} + \frac{(\hat{\sigma}_i - \tilde{\sigma}_{\text{reg}}(\tilde{y}))^2}{d_i}\right)}. \end{aligned} \tag{6.52}$$

Analogously to (6.23), the posterior likelihood L is obtained by considering all measurements $\{(\hat{y}_i, \hat{\sigma}_i)\}_{i=1}^N$ and by integrating the densities $\wp\left((\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right)$ with respect to a prior $\wp_0(y)$ as

$$\tilde{L}(a, b) = \prod_{i=1}^{N} \int \wp\left((\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right) \wp_0(y)\, dy. \tag{6.53}$$

The integration in (6.53) is still with respect to y and $\wp\left((\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right)$ is itself an explicit function of y, as it is clear from Equations (6.50-6.52). Therefore, (6.53) allows for direct calculation, and by solving (6.24) with the likelihood $\tilde{L}(a, b)$ in place of $L(a, b)$ (6.23) we obtain the parameters $\hat{a}$ and $\hat{b}$, which define both the ML estimate $\hat{\sigma}_{\text{fit}}$ of σ, exactly as in (6.25), and the ML estimate $\hat{\tilde{\sigma}}_{\text{fit}}$ of $\tilde{\sigma}$, which can be obtained from $\hat{\sigma}_{\text{fit}}$ by application of the transformations (6.44) and (6.45). Note that for the clipped raw-data it is unnatural to assume that $\wp_0$ is uniform on $[0, 1]$, because in the case of overexposure the true signal could be much larger than 1. Therefore, for the clipped raw-data, we assume that all positive values of y are equiprobable and we maximize[8] $\tilde{L}(a, b) = \prod_{i=1}^{N} \int_0^{+\infty} \wp\left((\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right) dy$.

Least-squares initialization

Similarly to the non-clipped case, we use a simple least-squares solution as the initial condition for the iterative maximization of the likelihood function. We exploit the inverse transformations (6.47)-(6.48) from Section 6.3.2 to attain a fit of σ^2 with respect to the non-clipped variables. Hence, the initial parameters are

[8] Equivalently, we maximize

$$\tilde{L}(a, b) = \lim_{j \to +\infty} \prod_{i=1}^{N} \int_0^{1+j} \wp\left((\hat{y}_i, \hat{\sigma}_i)\,|\tilde{y}_i = \tilde{y}\right) \wp_j(y)\,(1 + j)\, dy,$$

where $\wp_j$ is a uniform density on $[0, 1 + j]$ and the normalization factor $(1 + j)$ enables the convergence of the sequence of integrals.

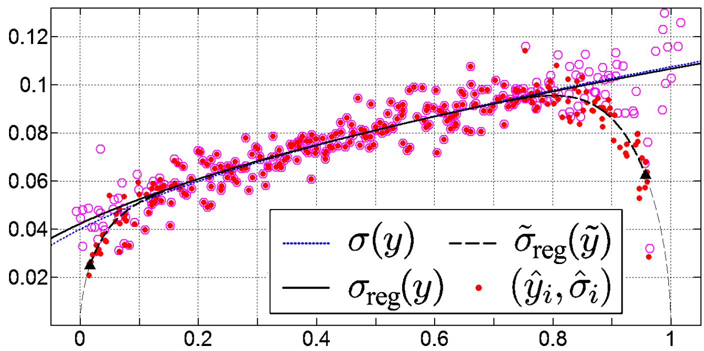


Figure 6.13: Estimation with clipped observations $\tilde{z}$ (Fig. 6.2): Least-squares initialization. Each dot of the scatter plot corresponds to a pair $(\hat{y}_i, \hat{\sigma}_i)$ of estimates of $\tilde{y}_i$ and $\tilde{\sigma}(\tilde{y}_i)$. The circles indicate these pairs of estimates after inverse-transformation (see Eqs. (6.55) and (6.56)). The solid line shows the square root of the least-squares estimate of the variance function σ^2 (see Section 6.4.2), $\hat{a}_0 = 0.00945$ ($\hat{\chi}_0 = 105.82$), $\hat{b}_0 = 0.001822$ ($\sqrt{\hat{b}_0} = 0.04268$). The dotted line is the true σ, while the dashed-line is the function $\tilde{\sigma}_{\text{reg}}$ with parameters $\hat{a}_0, \hat{b}_0$ used as initial condition for the iterative maximization of the likelihood (6.53).

given as $\begin{bmatrix}\hat{a}_0 & \hat{b}_0\end{bmatrix} = \hat{\mathbf{v}}\boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}$, with the dependent and independent variables transformed as

$$\rho_{i,0}^{\text{reg}} = \sqrt{\left(\frac{\hat{y}_i}{\hat{\sigma}_i}\right)^2 + \varepsilon_{\text{reg}}^2}, \quad \rho_{i,1}^{\text{reg}} = \sqrt{\left(\frac{1-\hat{y}_i}{\hat{\sigma}_i}\right)^2 + \varepsilon_{\text{reg}}^2}, \tag{6.54}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \hat{y}_1\mathcal{E}_r\left(\rho_{1,0}^{\text{reg}}\right) - \hat{y}_1 + 1 - (1-\hat{y}_1)\,\mathcal{E}_r\left(\rho_{1,1}^{\text{reg}}\right) & 1 \\ \hat{y}_2\mathcal{E}_r\left(\rho_{2,0}^{\text{reg}}\right) - \hat{y}_2 + 1 - (1-\hat{y}_2)\,\mathcal{E}_r\left(\rho_{2,1}^{\text{reg}}\right) & 1 \\ \vdots & \vdots \end{bmatrix}, \tag{6.55}$$

$$\hat{\mathbf{v}} = \begin{bmatrix} \frac{\breve{\kappa}_{n_1}^2\hat{\sigma}_1^2}{\left(\mathcal{S}_r\left(\rho_{1,0}^{\text{reg}}\right)\mathcal{S}_r\left(\rho_{1,1}^{\text{reg}}\right)\right)^2} & \frac{\breve{\kappa}_{n_2}^2\hat{\sigma}_2^2}{\left(\mathcal{S}_r\left(\rho_{2,0}^{\text{reg}}\right)\mathcal{S}_r\left(\rho_{2,1}^{\text{reg}}\right)\right)^2} & \cdots \end{bmatrix}. \tag{6.56}$$

Figures 6.13 and 6.14 respectively show the initial $\tilde{\sigma}_{\text{reg}}$, which corresponds to the parameters $\hat{a}_0, \hat{b}_0$, and the ML estimates $\hat{\sigma}_{\text{fit}}$ and $\hat{\tilde{\sigma}}_{\text{fit}}$ found using $\tilde{\sigma}_{\text{reg}}$ as initialization in the iterative maximization of the likelihood. In Figure 6.13 we can see that the inverse transformations (6.47)-(6.48) used in (6.55) and (6.56) effectively move the clipped estimates pairs near to their respective “non-clipped” positions. Note also the increased accuracy of the ML estimates compared to that of the least-squares ones.

6.5 Robust estimates

Despite the removal of edges from X^{smo}, small singularities or fine textures and edges of the image can still be present in z^{wdet}, within S_i. The accuracy of the sample standard-deviation estimator (6.16) is consequently degraded, since

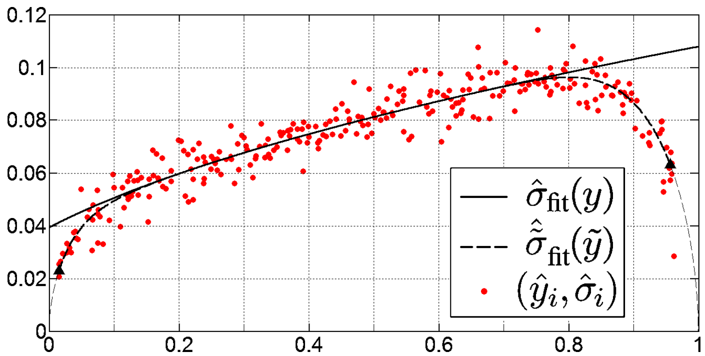


Figure 6.14: Estimation with clipped observations $\tilde{z}$ (Fig. 6.2): ML solution. Each dot of the scatter plot corresponds to a pair $(\hat{y}_i, \hat{\sigma}_i)$ of estimates of $\tilde{y}_i$ and $\tilde{\sigma}(\tilde{y}_i)$. The solid line and dashed line show the maximum-likelihood estimates $\hat{\sigma}_{\text{fit}}$ and $\hat{\tilde{\sigma}}_{\text{fit}}$ of the standard-deviation functions σ and $\tilde{\sigma}$, respectively. $\hat{a} = 0.00995$ ($\hat{\chi} = 100.52$), $\hat{b} = 0.001552$ ($\sqrt{\hat{b}} = 0.03940$). The plot of $\hat{\sigma}_{\text{fit}}$ overlaps perfectly with that of the true σ.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
κ_n^{mad}	0.798		0.732		0.712		0.702		0.696		0.693		0.690		0.688		0.686		0.685	

Table 6.2: Bias factor κ_n^{mad} for the MAD estimator (6.57) for small finite samples of n independent normal variables; $\kappa_{2n}^{\text{mad}} = \kappa_{2n-1}^{\text{mad}}$.

z^{wdet} would contain wild errors of large amplitude, which can cause the distribution of z^{wdet} to become heavy-tailed. This typically leads to an over-estimate of the standard-deviation. It is well-known that robust estimators based on order-statistics can effectively deal with these situations.

6.5.1 Robust standard-deviation estimates

To reduce the influence of these wild errors, we replace the sample standard-deviation estimator (6.16) with the robust estimator based on the median of the absolute deviations (MAD) [119],[76][9]

$$\hat{\sigma}_i^{\text{mad}} = \frac{1}{\kappa_{n_i}^{\text{mad}}} \operatorname*{median}_{x_j \in S_i} \left\{ \left| z^{\text{wdet}}(x_j) \right| \right\}, \tag{6.57}$$

where $\kappa_{n_i}^{\text{mad}}$ is again a scaling factor to make the estimator unbiased. It is well known that, for large normally i.i.d. samples, κ_n^{mad} approaches $\Phi^{-1}(3/4) = 0.6745$, where Φ^{-1} is the inverse c.d.f. of the standard normal. For small finite samples, the values of κ_n^{mad} are larger and up to $\sqrt{2/\pi} = 0.7979$ (mean of absolute deviations

[9] In its general form, this estimator is defined as

$$\frac{1}{\kappa_{n_i}} \operatorname*{median}_{x_j \in S_i} \left| z^{\text{wdet}}(x_j) - \operatorname*{median}_{x_j \in S_i} \left(z^{\text{wdet}}(x_j) \right) \right|.$$

However, when used on wavelet detail coefficients, the subtraction of the median in the deviation is often discarded, since its expected value for these coefficients is typically zero.

of $\mathcal{N}(0,1)$); in Table 6.2 we give the values of κ_n^{mad} for $n = 1, \ldots, 20$. For larger n, we can approximate κ_n^{mad} as $\kappa_n^{\mathrm{mad}} \simeq \frac{1}{5n} + \Phi^{-1}(3/4)$. Note that $\kappa_{2n}^{\mathrm{mad}} = \kappa_{2n-1}^{\mathrm{mad}}$; this is because in a set of $2n$ independent random variables, any individual variable has probability 0.5 of belonging to the subset of n variables smaller (or larger) than the median value. Tables similar to Table 6.2 can be found in [145] for a few other estimators of the standard deviation.

Variance and distribution of the standard-deviation estimates

The variance of the robust estimates (6.57) can be approximated[10] as

$$\mathrm{var}\left\{\hat{\sigma}_i^{\mathrm{mad}}\right\} = \sigma^2(y_i)\, d_i^{\mathrm{mad}}, \qquad d_i^{\mathrm{mad}} \simeq \frac{1.35}{n_i + 1.5}. \tag{6.58}$$

Thus, we pay the increased robustness with respect to outliers with a more than twice as large variance of the estimates, in comparison to (6.19) (this larger variance can be seen clearly by visual comparison of Figures 6.14 and 6.16). However, in practice, when working with many-megapixels images, the variance (6.58) is often quite low, due to the large number of samples n_i. Hence, the use of the robust estimator is ordinarily recommendable.

Like the sample standard-deviation estimates, also the MAD estimates (6.57) have a distribution which can be approximated by a normal[11]. In particular (and analogous to (6.20)),

$$\hat{\sigma}_i^{\mathrm{mad}} \sim \mathcal{N}\left(\sigma(y_i), \sigma^2(y_i)\, d_i^{\mathrm{mad}}\right).$$

Estimates of the variance

An unbiased robust estimate of the variance (as used by the least-squares initialization (6.26)) can be obtained from the squared $\left(\hat{\sigma}_i^{\mathrm{mad}}\right)^2$ (6.57), provided multiplication with a bias correction factor. Using the same symbols of Section 6.2.4, we denote this estimate of the variance as $\left(\breve{\kappa}_{n_i}^{\mathrm{mad}}\hat{\sigma}_i^{\mathrm{mad}}\right)^2$, where the factor $\left(\breve{\kappa}_n^{\mathrm{mad}}\right)^2$ can be approximated as $\left(\breve{\kappa}_n^{\mathrm{mad}}\right)^2 \simeq 1 + \frac{1}{5n}$.

6.5.2 Maximum-likelihood fitting (non-clipped)

The ML solution is found exactly as in Section 6.2.4, provided that the estimates and factors $\hat{\sigma}_i$, d_i, $\breve{\kappa}_{n_i}$ are replaced by their respective "mad" counterparts $\hat{\sigma}_i^{\mathrm{mad}}$, d_i^{mad}, $\breve{\kappa}_{n_i}^{\mathrm{mad}}$ in Equations (6.22-6.24) and (6.28).

---

[10] The approximation of d_i^{mad} in (6.58) can be obtained by Monte Carlo simulations. A table with few of these values is found also in [145].

[11] The normal approximation can be easily verified by numerical simulations. Despite all the necessary ingredients for an analytical proof can be found in [88], it seems that that this result is not explicitly reported in the literature.

n_i	$E\left\{\hat{\sigma}_i^{\mathrm{mad}}\right\}/\sigma(y_i)$									
	$\mu=-1$	$\mu=-0.5$	$\mu=0$	$\mu=0.5$	$\mu=1$	$\mu=1.5$	$\mu=2$	$\mu=2.5$	$\mu=3$	$\mu=5$
2	0.215	0.380	0.566	0.739	0.869	0.946	0.983	0.995	0.999	1.000
10	0.168	0.341	0.543	0.733	0.872	0.952	0.985	0.997	0.999	1.000
20	0.162	0.335	0.540	0.732	0.873	0.952	0.986	0.997	0.999	1.000
50	0.158	0.332	0.538	0.731	0.873	0.953	0.986	0.997	0.999	1.000
100	0.156	0.330	0.537	0.731	0.873	0.953	0.986	0.997	0.999	1.000
200	0.155	0.329	0.536	0.731	0.873	0.953	0.986	0.997	0.999	1.000
500	0.155	0.329	0.536	0.731	0.873	0.953	0.986	0.997	0.999	1.000
500000	0.154	0.329	0.536	0.731	0.873	0.953	0.986	0.997	0.999	1.000

Table 6.3: Expectation $E\left\{\hat{\sigma}_i^{\mathrm{mad}}\right\}/\sigma(y_i)$ for different combinations of $\mu=y_i/\sigma(y_i)$ and n_i.

6.5.3 Clipped observations

Let us now apply the MAD estimator (6.57) to the wavelet coefficients $\tilde{z}^{\mathrm{wdet}}=\downarrow_2(\tilde{z}\circledast\psi)$ of the clipped observations $\tilde{z}$,

$$\hat{\sigma}_i^{\mathrm{mad}}=\frac{1}{\kappa_{n_i}^{\mathrm{mad}}}\operatorname*{median}_{x_j\in S_i}\left\{\left|\tilde{z}^{\mathrm{wdet}}(x_j)\right|\right\}. \tag{6.59}$$

Although robust with respect to outliers of large amplitude, the MAD estimator is sensitive to the asymmetry in the distribution of the samples [145] and even the limiting value $\Phi^{-1}(3/4)$ is, as one can expect from the presence of the inverse c.d.f. of the normal distribution, essentially correct for normally distributed samples only. Thus, contrary to the sample standard-deviation, the MAD estimator is not asymptotically unbiased:

$$\lim_{n_i\to\infty}E\left\{\hat{\sigma}_i^{\mathrm{mad}}\right\}\neq\operatorname{std}\left\{\tilde{z}^{\mathrm{wdet}}\right\}=\lim_{n_i\to\infty}E\left\{\hat{\sigma}_i\right\}. \tag{6.60}$$

Let us investigate this estimation bias for large as well as for small finite samples. As in Section 6.4.1, we restrict ourselves to the case of clipping from below for a constant $y(x)\equiv y$, $\forall x\in X$, and apply the estimator (6.57) to the corresponding $\tilde{z}^{\mathrm{wdet}}$. In Table 6.3, we give the expectations $E\left\{\hat{\sigma}_i^{\mathrm{mad}}\right\}/\sigma(y_i)$ for different combinations of $\mu=y_i/\sigma(y_i)$ and n_i. The same considerations which we made commenting Table 6.1 can be repeated also for Table 6.3.

We use the expectations of large-sample estimates (values with $n_i=500000$ in Table 6.3) as a numerical definition of $\lim_{n_i\to\infty}E\left\{\hat{\sigma}_i^{\mathrm{mad}}\right\}$. In this way, we define the function $\mathcal{S}_m^{\mathrm{mad}}$ which gives $\lim_{n_i\to\infty}E\left\{\hat{\sigma}_i^{\mathrm{mad}}\right\}$ as a function of $E\{\tilde{\nu}\}$. Hence, are also defined $\mathcal{S}_e^{\mathrm{mad}}$, $\mathcal{S}_r^{\mathrm{mad}}$, and $\mathcal{E}_r^{\mathrm{mad}}$ and the corresponding analogs of the direct and inverse transformation formulas (6.45)-(6.48). In particular, we define $\tilde{\sigma}^{\mathrm{mad}}$ by

$$\tilde{\sigma}^{\mathrm{mad}}(\tilde{y})=\sigma(y)\,\mathcal{S}_m^{\mathrm{mad}}\left(\tfrac{y}{\sigma(y)}\right)\mathcal{S}_m^{\mathrm{mad}}\left(\tfrac{1-y}{\sigma(y)}\right).$$

In Figure 6.15, we show the plot of $\mathcal{S}_e^{\mathrm{mad}}$ superimposed on the plot of $\mathcal{S}_e$ (dashed line). The vertical difference between the two plots in the Figure is the bias[12] of

[12] Although it is not insignificant, this asymptotic bias is as not large as it would be

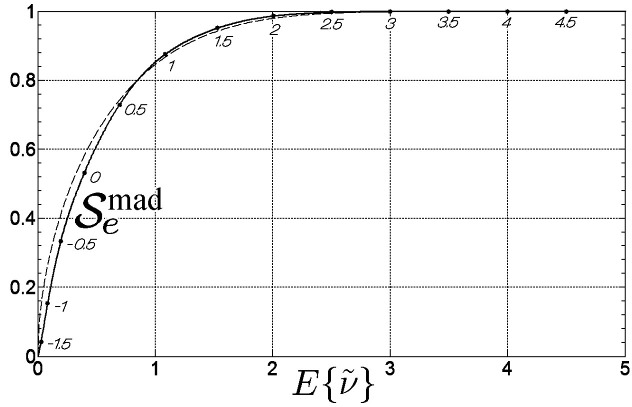


Figure 6.15: Large-sample asymptotic expectation $\lim_{n_i\to\infty} E\left\{\hat{\sigma}_i^{\mathrm{mad}}\right\}$ of the median of the absolute deviations (6.57) as a function $\mathcal{S}_e^{\mathrm{mad}}$ of the expectation $E\{\tilde{\nu}\} = \tilde{y}_i/\sigma(y_i)$ of the clipped variables. The dashed line corresponds to the function $\mathcal{S}_e$ from Fig. (6.10).

(6.57) as an estimator of $\operatorname{std}\{\tilde{\nu}\} = \operatorname{std}\left\{\tilde{z}^{\mathrm{wdet}}\right\}/\sigma(y)$. These differences can also be seen (as a function of μ) by comparing the last rows of the Tables 6.1 and 6.3, for the cases $n_i = \infty$ and $n_i = 500000$, respectively.

For the case of the MAD estimator, formula (6.49) needs therefore to be modified as follows:

$$\hat{\sigma}_i^{\mathrm{mad}} \sim \mathcal{N}\left(\tilde{\sigma}^{\mathrm{mad}}(\tilde{y}_i), \left(\tilde{\sigma}^{\mathrm{mad}}(\tilde{y}_i)\right)^2 d_i^{\mathrm{mad}}\right).$$

The specific $\mathcal{S}_m^{\mathrm{mad}}$ allows us to take into account of the difference $\tilde{\sigma}^{\mathrm{mad}}(\tilde{y}_i) - \tilde{\sigma}(\tilde{y}_i)$ (6.60) here and in the following ML estimation of the functions σ and $\tilde{\sigma}$.

Maximum-likelihood fitting (clipped)

The ML solution is found exactly as in Section 6.4.2, provided that the functions $\mathcal{S}_m^{\mathrm{mad}}$, $\mathcal{S}_r^{\mathrm{mad}}$, and $\mathcal{E}_r^{\mathrm{mad}}$ defined above and estimates and factors $\hat{\sigma}_i^{\mathrm{mad}}$, d_i^{mad}, $\breve{\kappa}_{n_i}^{\mathrm{mad}}$ are used, in place of their respective "non-robust" counterparts, in the Equations (6.50), (6.52), (6.53), (6.54), (6.55), and (6.56). The found parameters $\hat{a}$ and $\hat{b}$ define simultaneously three functions: from (6.25) we obtain $\hat{\sigma}_{\mathrm{fit}}$, a ML estimate of σ; $\hat{\tilde{\sigma}}_{\mathrm{fit}}$, a ML estimate of $\tilde{\sigma}$; and $\hat{\tilde{\sigma}}_{\mathrm{fit}}^{\mathrm{mad}}$, a ML estimate of $\tilde{\sigma}^{\mathrm{mad}}$,

$$\tilde{\sigma}^{\mathrm{mad}}(\tilde{y}) = \sigma(y)\, \mathcal{S}_m^{\mathrm{mad}}\left(\tfrac{y}{\sigma(y)}\right) \mathcal{S}_m^{\mathrm{mad}}\left(\tfrac{1-y}{\sigma(y)}\right),$$

around which are scattered the estimates $\left(\hat{y}_i, \hat{\sigma}_i^{\mathrm{mad}}\right)$.

In Figure 6.16 we show the ML estimates $\hat{\sigma}_{\mathrm{fit}}$ and $\hat{\tilde{\sigma}}_{\mathrm{fit}}^{\mathrm{mad}}$ obtained for the clipped $\tilde{z}$ from Figure 6.2 using the MAD. We can see that, despite the larger variance of the estimates $\hat{\sigma}_i^{\mathrm{mad}}$ (as compared to $\hat{\sigma}_i$ in Figure 6.14), the final estimated parameters and the corresponding $\hat{\sigma}_{\mathrm{fit}}$ are essentially the same as those obtained

---

if applying the MAD (6.57) directly on $\tilde{z}$ instead of z^{wdet}. In fact, it is easy to realize that $\operatorname{median}\{\tilde{\nu}\} = 0$ for $\mu \leq 0$. Since obviously $\operatorname{median}\{|\tilde{\nu}|\} = \operatorname{median}\{\tilde{\nu}\}$, we have that $\operatorname{median}\{|\tilde{\nu}|\} = \operatorname{median}\{|\tilde{\nu} - \operatorname{median}\{\tilde{\nu}\}|\} = 0$ for $\mu \leq 0$.

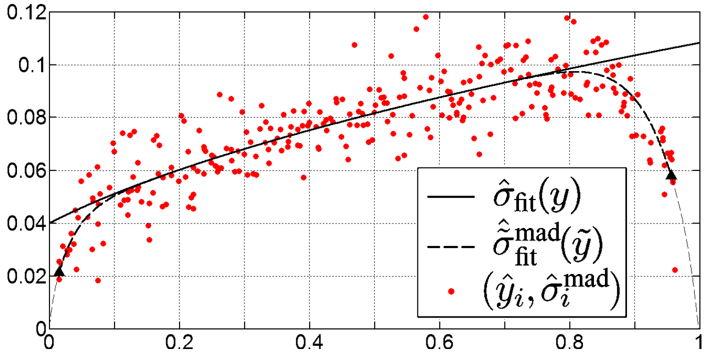


Figure 6.16: Estimation with clipped observations $\tilde{z}$ and MAD estimator (6.59) (Fig. 6.2): Maximum-likelihood solution. Each dot of the scatter plot corresponds to a pair $(\hat{y}_i, \hat{\sigma}_i)$ of estimates of $\tilde{y}_i$ and $\tilde{\sigma}(\tilde{y}_i)$. The solid line and dashed line show the maximum-likelihood estimates $\hat{\sigma}_{\text{fit}}$ and $\hat{\tilde{\sigma}}^{\text{mad}}_{\text{fit}}$ of the standard-deviation functions σ and $\tilde{\sigma}$, respectively. The plot of $\hat{\sigma}_{\text{fit}}$ overlaps perfectly with that of the true σ (shown in Figure 6.1). The estimated parameters are $\hat{a} = 0.01000$ ($\hat{\chi} = 100.04$) and $\hat{b} = 0.001594$ ($\sqrt{\hat{b}} = 0.03992$).

using the sample standard-deviation. In the Figure, note the slightly different shape of the plot of $\hat{\tilde{\sigma}}^{\text{mad}}_{\text{fit}}$ compared to $\hat{\tilde{\sigma}}_{\text{fit}}$.

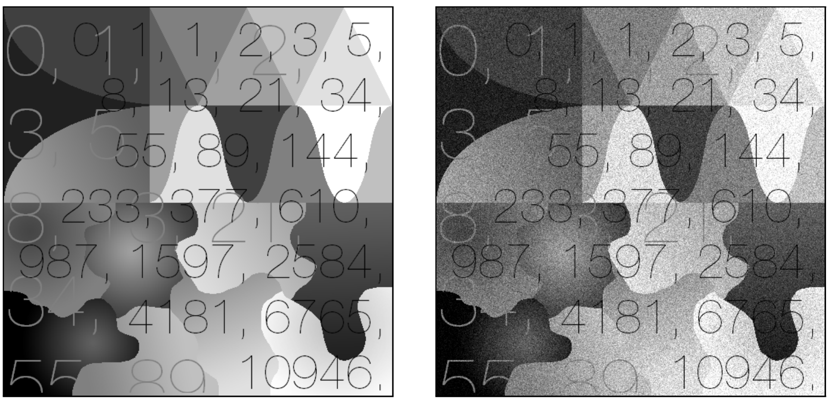


Figure 6.17: The piecewise smooth test image of Fig. 6.2 with thin text superimposed: original y and observation z degraded by Poissonian and Gaussian noise with parameters $\chi = 100$ ($a = 0.01$) and $b = 0.04^2$.

6.5.4 Another example

To demonstrate a situation where the robust estimates are remarkably more accurate than the non-robust ones, we introduce a number of thin and sharp discontinuities in the test image, as shown in Figure 6.17. At many places, due to low contrast (and also due to the simplicity of our edge-detector), these discontinuities cannot be detected properly and are thus eventually incorporated in the smoothness set X^{smo}. In Figure 6.18, we show the estimates $\hat{\sigma}_{\text{fit}}$ and $\widehat{\tilde{\sigma}}_{\text{fit}}$ obtained using

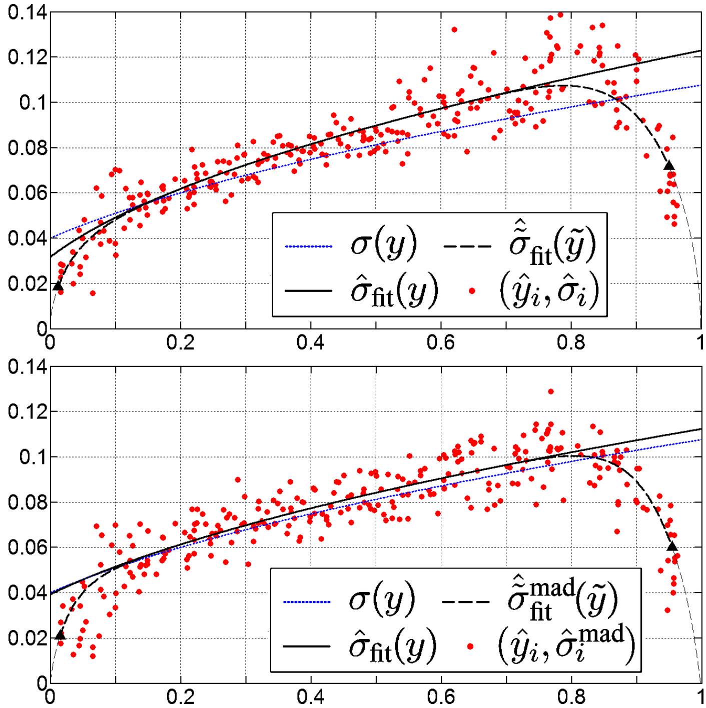


Figure 6.18: Robust (bottom) vs. non-robust (top) estimation with clipped observations $\tilde{z}$ (Fig. 6.17): ML solutions. Each dot of the scatter plot corresponds to a pair $(\hat{y}_i, \hat{\sigma}_i)$ of estimates of $\tilde{y}_i$ and $\tilde{\sigma}\,(\tilde{y}_i)$. The solid line and dashed line show the maximum-likelihood estimates $\hat{\sigma}_{\text{fit}}$ and $\hat{\tilde{\sigma}}^{\text{mad}}_{\text{fit}}$ of the standard-deviation functions σ and $\tilde{\sigma}$, respectively. The dotted line is the true σ. The parameters estimated by the two methods are $\hat{a} = 0.01415$ ($\hat{\chi} = 70.68$), $\hat{b} = 0.000951$ ($\sqrt{\hat{b}} = 0.03084$) and $\hat{a} = 0.01108$ ($\hat{\chi} = 90.29$), $\hat{b} = 0.001524$ ($\sqrt{\hat{b}} = 0.03904$), respectively.

the robust and the non-robust estimator. As easily expected, the estimates $\hat{\sigma}_i$ are inaccurate and typically biased in favour of larger standard-deviation values. As a result, the $\hat{\sigma}_{\text{fit}}$ curve does not match with the true σ. The result obtained from the robust estimates $\hat{\sigma}^{\text{mad}}_i$ is essentially better, with only a mild overestimation of the signal-dependent component of the noise.

6.6 Experiments with raw data

We performed extensive experiments with raw data[13] of various digital imaging sensors under different acquisition parameters. The devices included three CMOS

[13] We reorder the raw-data pixels from color filter array (e.g., Bayer pattern) sensors in such a way to pack pixels of the same color channel together. Thus, the processed frame z is composed by four (Bayer pattern) or three (Fujifilm SuperCCD) subimages, which portray the different chromatic components. The boundaries between the subimages are usually detected as edges, as can be seen in Figure 6.20.

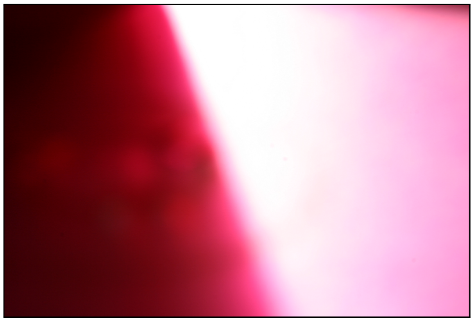


Figure 6.19: Out-of-focus image with under- and overexposure (Canon EOS 350D, ISO 100).

Figure 6.20: Natural image (Canon EOS 350D, ISO 1600) and the wavelet detail coefficients z^{wdet} (of its raw-data) restricted on the set of smoothness X^{smo}. The four subimages are arranged as $[R,B;G_1,G_2]$.

sensors from Nokia cameraphones (300 Kpixel, 1.3 Mpixel, 5 Mpixel), a SuperCCD HR [160] from a Fujifilm FinePix S5600 camera (5.1 Mpixel), and two CMOS from Canon EOS 350D and 400D SLR cameras (8 Mpixel, 10 Mpixel). In all experiments we found near-perfect fit of our proposed clipped Poissonian-Gaussian model to the data. We also compared the parametric curves $\hat{\tilde{\sigma}}_{\mathrm{fit}}$, estimated from a single image by the proposed algorithm, with the nonparametric curves estimated by the algorithm [48] using 50 images; we found the agreement to be very good, with minor differences due to the fact that the present algorithm includes the fix-pattern noise (FPN) in the noise estimate, whereas [48], being a pixelwise procedure, estimates only the temporal noise. Because of length limitation, we present here only few most significant examples of the obtained results.

First, we show the estimated curves for the raw-data of Canon EOS 350D with ISO 100 and 1600 (lowest and highest user-selectable analog-gain options). An out-of-focus, hence smooth, target (shown in Figure 6.19 was used, with under- and overexposed parts, thus providing a complete and reliable coverage of the dynamic range and beyond. Besides the excellent match between the fitted parametric curve $\hat{\tilde{\sigma}}^{\mathrm{mad}}_{\mathrm{fit}}$ and the local estimates $\left(\hat{y}_i, \hat{\sigma}^{\mathrm{mad}}_i\right)$, one should observe that the curve accurately follows the estimates as these approach $(1,0)$, in agreement with our

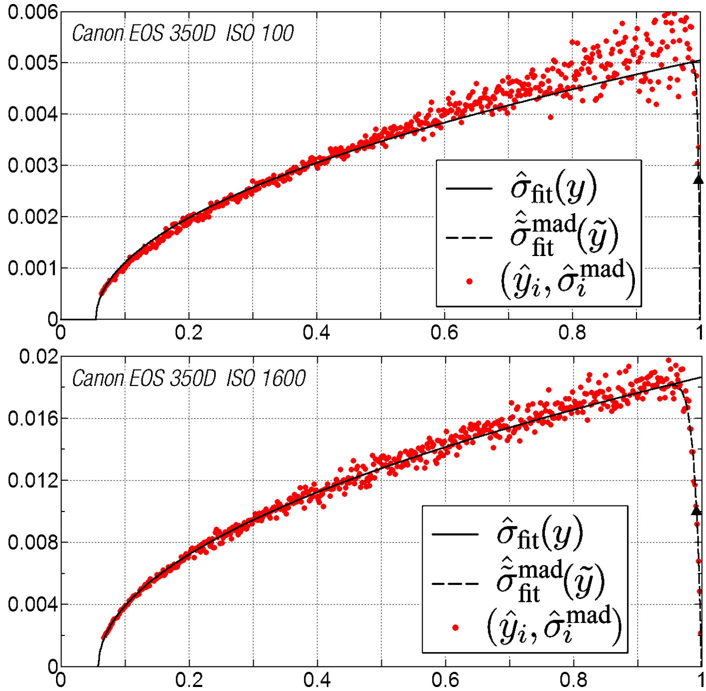


Figure 6.21: Estimation of the standard-deviation function σ from the raw-data of the out-of-focus image (Fig. 6.19,left).

clipped data modeling. Nearly identical curves are found when the smooth out-of-focus target is replaced by one, shown in Figure 6.20(left), which presents various complex structures that may potentially impair the estimation. The wavelet coefficients z^{wdet} are shown in Figure 6.20(right). The estimated curves are shown in Figure 6.22, where one can also observe the wider dispersion of the estimates $\left(\hat{y}_i, \hat{\sigma}_i^{\text{mad}}\right)$ (due to the much smaller number n_i of usable samples in the level sets S_i) and that the $\hat{y}_i$ are not distributed over the full data-range. In Figure 6.23 we show a remarkable example of clipping from above and from below within the same frame, as it can be found with the Fujifilm FinePix S5600 using ISO 1600. Observe that the fit of the model to the data is again nearly perfect. A comparison with the nonparametric estimate $\hat{\sigma}_{\text{np}}$ obtained by the method [48] is given in Figure 6.24 The curve $\hat{\sigma}_{\text{np}}$ was computed analyzing 50 shots of the same target, whereas only one of these 50 images has been used to estimate the function $\hat{\sigma}_{\text{fit}}$ with the proposed algorithm. The shots were acquired by a 1.3-Mpixel CMOS sensor of a cameraphone, with an analog gain of 10 dB. We note that the nonparametric method provides an estimate of $\sigma(y)$ only for the range of values y covered by the used images. Moreover, it produces erroneous results approaching the extrema of this range (about 0.07 and 0.41), due to lack of samples. Within these extrema (i.e., $0.07 < y < 0.41$) the two plots are however very close, with minor differences due to the lack of FPN contribution to the $\hat{\sigma}_{\text{np}}$ estimate.

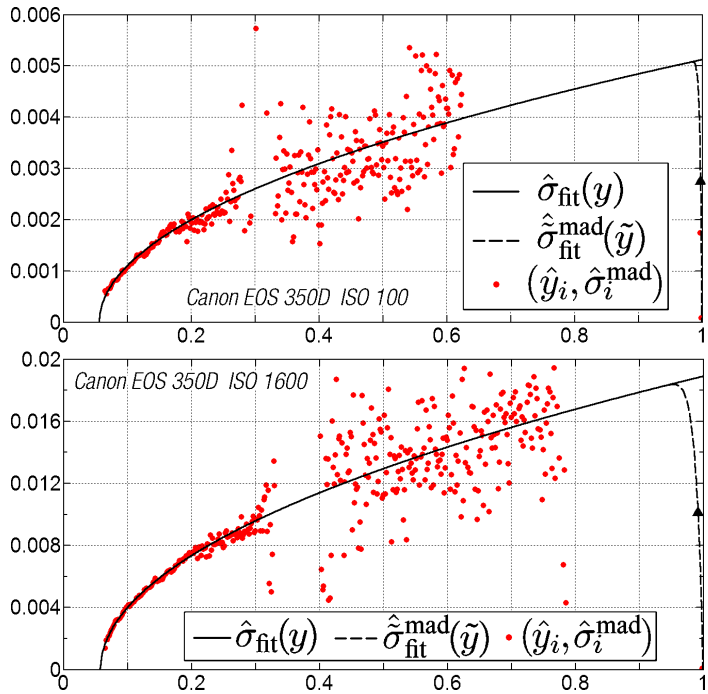


Figure 6.22: Estimation of the function σ from the raw-data of the natural image (Fig. 6.19,center). Compare with the corresponding plots in Fig. 6.21.

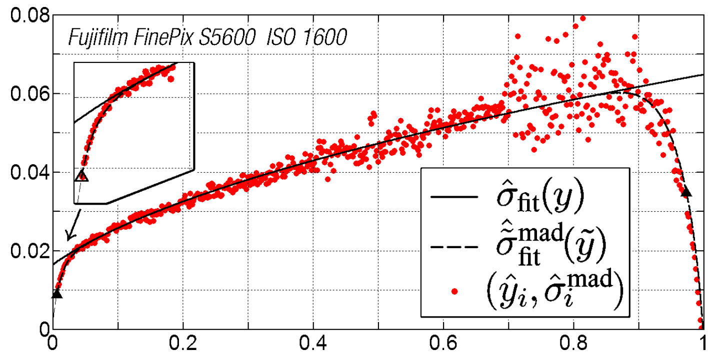


Figure 6.23: Estimation from raw-data which exhibits both clipping from above (overexposure) and from below (underexposure).

6.7 Comments

6.7.1 Different parametric models for the σ function

We remark that the proposed algorithm is not restricted to the particular model (6.4). In fact, the parameters of any other parametric model can be estimated in the same way. It is sufficient to modify the expression of the function σ_{reg} in

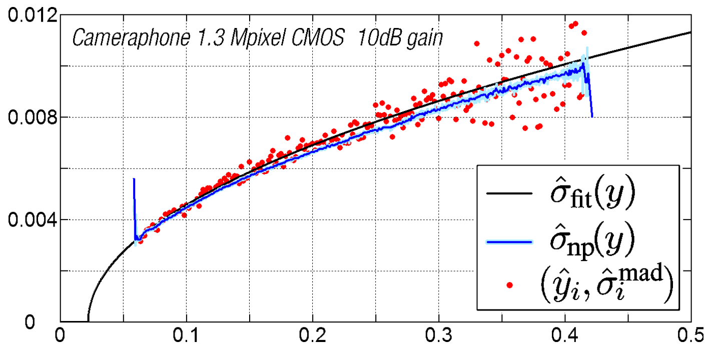


Figure 6.24: Comparison between the parametric $\hat{\sigma}_{\text{fit}}$, estimated from a single image, and the nonparametric curve $\hat{\sigma}_{\text{np}}$ [48] computed from 50 images.

	$\mathcal{S}_m$	$\mathcal{S}_e$	$\mathcal{E}_r$	$\mathcal{S}_r$
p_{10}	–	$-5.7155077 \cdot 10^{-3}$	–	$2.3348876 \cdot 10^{-7}$
p_9	$1.4308530 \cdot 10^{-6}$	$9.9630886 \cdot 10^{-2}$	–	$-8.7619300 \cdot 10^{-6}$
p_8	$3.2172868 \cdot 10^{-7}$	$-7.3358565 \cdot 10^{-1}$	–	$1.2534052 \cdot 10^{-4}$
p_7	$-2.6295693 \cdot 10^{-5}$	$2.9464712 \cdot 10^{0}$	$-5.5320701 \cdot 10^{-3}$	$-7.0907114 \cdot 10^{-4}$
p_6	$-8.5123452 \cdot 10^{-5}$	$-6.9596693 \cdot 10^{0}$	$-5.5542026 \cdot 10^{-2}$	$-1.4943642 \cdot 10^{-3}$
p_5	$-1.7851033 \cdot 10^{-5}$	$9.7330082 \cdot 10^{0}$	$-2.0363415 \cdot 10^{-1}$	$4.5981260 \cdot 10^{-2}$
p_4	$2.0282884 \cdot 10^{-3}$	$-8.3760888 \cdot 10^{0}$	$-3.4651219 \cdot 10^{-1}$	$-2.8638716 \cdot 10^{-1}$
p_3	$2.4377832 \cdot 10^{-2}$	$3.9881199 \cdot 10^{0}$	$-4.1222715 \cdot 10^{-1}$	$8.5412513 \cdot 10^{-1}$
p_2	$3.7234715 \cdot 10^{-2}$	$-1.8598057 \cdot 10^{0}$	$-9.1504182 \cdot 10^{-1}$	$-1.5725702 \cdot 10^{0}$
p_1	$7.0309281 \cdot 10^{-1}$	$-7.0210049 \cdot 10^{-1}$	$-4.3779025 \cdot 10^{0}$	$-5.2653050 \cdot 10^{-1}$
p_0	$1.6923658 \cdot 10^{-1}$	$-3.1389654 \cdot 10^{-6}$	$-1.5498697 \cdot 10^{0}$	$-1.2319839 \cdot 10^{-10}$

Table 6.4: Coefficients of the polynomial $p(t) = \sum_k p_k t^k$ used for indirect interpolation of the functions $\mathcal{S}_m$, $\mathcal{S}_e$, $\mathcal{S}_r$, $\mathcal{E}_r$ as in (6.62).

the likelihood (6.23) according to the assumed parametric model. Therefore, our algorithm has a broader scope of application than shown here and can be applied for parameter estimation of other signal-dependent noise models, which can be approximated as heteroskedastic normal. Heuristic models for σ, such as those found using the principal component analysis in [110], can also be exploited in our estimation framework.

6.7.2 Multiple images

If two or more independent realizations of the image z are available, they can be easily exploited in a fashion similar to [48]. Let us denote the different realizations as $z_1, \ldots, z_J$. From (6.1), we have

$$z_j(x) = y(x) + \sigma(y(x))\,\xi_j(x) \qquad \forall x \in X, \quad j = 1, \ldots, J,$$

where $\xi_1, \ldots, \xi_J$ are mutually independent and, for a fixed x, $\xi_1(x), \ldots, \xi_J(x)$ are i.i.d. random variables, each with variance equal to one. Thus, by averaging we obtain

$$z^{\text{ave}}(x) = \sum_{j=1}^{J} \frac{z_j(x)}{J} = y(x) + \frac{\sigma(y(x))}{\sqrt{J}}\xi'(x), \;\; \forall x \in X, \tag{6.61}$$

	$\mathcal{S}_m^{\mathrm{mad}}$	$\mathcal{S}_e^{\mathrm{mad}}$	$\mathcal{E}_r^{\mathrm{mad}}$	$\mathcal{S}_r^{\mathrm{mad}}$
p_{10}	–	$-2.0882655 \cdot 10^{1}$	–	$-6.8695634 \cdot 10^{-1}$
p_9	$6.8722511 \cdot 10^{-4}$	$1.5778478 \cdot 10^{2}$	$-2.4852997 \cdot 10^{1}$	$1.0659232 \cdot 10^{1}$
p_8	$-3.3132811 \cdot 10^{-3}$	$-4.9232870 \cdot 10^{2}$	$6.2356790 \cdot 10^{1}$	$-7.1938586 \cdot 10^{1}$
p_7	$4.6401970 \cdot 10^{-4}$	$8.1719776 \cdot 10^{2}$	$-3.4588152 \cdot 10^{1}$	$2.7680210 \cdot 10^{2}$
p_6	$1.4193996 \cdot 10^{-2}$	$-7.7148949 \cdot 10^{2}$	$-2.3467375 \cdot 10^{1}$	$-6.6844401 \cdot 10^{2}$
p_5	$-3.3370736 \cdot 10^{-3}$	$4.0631819 \cdot 10^{2}$	$2.2066307 \cdot 10^{1}$	$1.0501179 \cdot 10^{3}$
p_4	$-4.0537889 \cdot 10^{-2}$	$-1.0564855 \cdot 10^{2}$	$1.7531850 \cdot 10^{0}$	$-1.0745079 \cdot 10^{3}$
p_3	$7.8410754 \cdot 10^{-2}$	$8.7374360 \cdot 10^{0}$	$-5.0041080 \cdot 10^{0}$	$6.9444029 \cdot 10^{2}$
p_2	$1.6003810 \cdot 10^{-2}$	$-1.6521034 \cdot 10^{0}$	$1.3236318 \cdot 10^{-1}$	$-2.6249852 \cdot 10^{2}$
p_1	$8.3418294 \cdot 10^{-1}$	$5.2647043 \cdot 10^{-2}$	$-4.7818191 \cdot 10^{0}$	$4.5940232 \cdot 10^{1}$
p_0	$7.0493620 \cdot 10^{-2}$	$-1.3640079 \cdot 10^{-6}$	$-1.4931770 \cdot 10^{0}$	$-1.3097339 \cdot 10^{0}$

Table 6.5: Coefficients of the polynomial $p(t) = \sum_k p_k t^k$ used for indirect interpolation of the functions $\mathcal{S}_m^{\mathrm{mad}}$, $\mathcal{S}_e^{\mathrm{mad}}$, $\mathcal{S}_r^{\mathrm{mad}}$, $\mathcal{E}_r^{\mathrm{mad}}$ as in (6.62).

where $\xi'(x)$ is another random variable with variance equal to one. Applying the proposed estimation procedure on z^{ave} permits to estimate the function $J^{-\frac{1}{2}}\sigma$ and hence σ. In principle, the advantage of the averaging (6.61) lies in the lower variance of the observation z^{ave}, which allows for better edge-removal and results in estimates $(\hat{y}_i, \hat{\sigma}_i)$ with lower variance. However, in practice, J cannot be taken arbitrarily large because a very large J would render the noise-to-signal ratio of z^{ave} too low for the noise $J^{-\frac{1}{2}}\sigma\xi'$ to be measured accurately. Hence, the averaging (6.61) is valuable only provided that the true y is sufficiently smooth and that the computational precision is high.

6.7.3 Denoising clipped signals

A generic denoising procedure can be modeled as an operator whose output is an estimate of the expectation of the noisy input. It means that when we denoise $\tilde{z}$, as the output we do not get an estimate of y, but rather an estimate of $\tilde{y}$. However, by applying (6.47) on the output, we can transform it to an estimate of y. In the same way, we can "take advantage of the noise" to obtain an image with a higher dynamic range, since the range of $\tilde{z}$ and $\tilde{y}$ is always smaller than that of y. An example of such a procedure is shown at the end of this thesis.

6.7.4 Interpolation of the functions $\mathcal{S}_m$, $\mathcal{S}_e$, $\mathcal{S}_r$, $\mathcal{E}_r$

In our algorithm, we use interpolated values for the functions $\mathcal{S}_m$, $\mathcal{S}_e$, $\mathcal{S}_r$, $\mathcal{E}_r$, $\mathcal{S}_m^{\mathrm{mad}}$, $\mathcal{S}_e^{\mathrm{mad}}$, $\mathcal{S}_r^{\mathrm{mad}}$, and $\mathcal{E}_r^{\mathrm{mad}}$ as no closed-form expression is available. For practicality, we resort to indirect (nonlinear) polynomial interpolation with exponential or logarithmic functions. The particular expressions of the used interpolant are as follows,

$$\begin{array}{ll} \mathcal{S}_m(\mu) \approx \frac{1+\tanh(p(\mu))}{2}, & \mathcal{S}_e(\xi) \approx 1 - e^{p\left(\sqrt{\xi}\right)}, \\ \mathcal{E}_r(\rho) \approx 1 - e^{p(\log(\rho))}, & \mathcal{S}_r(\rho) \approx 1 - e^{p(\rho)}, \end{array} \tag{6.62}$$

where $p(t) = \sum_k p_k t^k$ is a polynomial with coefficients p_k as given in Table 6.4. For the MAD estimates $\hat{\sigma}_i^{\mathrm{mad}}$ (6.57) and related functions $\mathcal{S}_m^{\mathrm{mad}}$, $\mathcal{S}_e^{\mathrm{mad}}$, $\mathcal{S}_r^{\mathrm{mad}}$, and $\mathcal{E}_r^{\mathrm{mad}}$, we use the same interpolant expressions as in (6.62) but with different polynomial coefficients, which are given in Table 6.5.

Standard approximations of Φ, or of the Gauss error function erf (being $\Phi(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(x/\sqrt{2}\right)\right)$), which are found in nearly all numerical software, can be used for the functions $\mathcal{E}_m$ and $\mathcal{S}_m$.

We note that, with (6.62) and the coefficients in Tables 6.4 and 6.5, the interpolation achieved for $\mathcal{E}_r^{\mathrm{mad}}$ and $\mathcal{S}_r$ is diverging at $\rho \simeq 0.5$ and $\rho \simeq 11.5$, respectively. However, the interpolation is accurate for $0.65 \leq \rho \leq 11$. Therefore, in our experiments we constrain ρ within these bounds. Since ρ, $\mathcal{E}_r^{\mathrm{mad}}$, or $\mathcal{S}_r$ are used only for the weighted least-squares problem (6.26) and not for the likelihood equation (6.23), the restriction on ρ does not affect the final estimation of the noise model parameters.

The interpolants and tables presented in this thesis complement and extend similar (although not equivalent) numerical data found in the literature [23] (and references therein), [24], [25], [26], [27], [149], [130], [148], [99]. To the best of the authors' knowledge, no other studies of indirect (e.g., in the wavelet domain) and robust (e.g., median-based) estimators of clipped samples have appeared to date and, although limited, the results in Table 6.5 are therefore valuable on their own. Further, we wish to emphasize that the various estimators proposed in the cited publications are developed for censored Gaussian processes with *fixed* mean and variance, and are thus not applicable to the more general estimation problem considered by us.

6.7.5 Alternatives to the MAD estimator

Alternative robust estimators, such as the $\mathrm{S_n}$ or the $\mathrm{Q_n}$ estimators [29, 145] can also be used instead of (6.57). Finally, we suggest the possibility of robust fitting of the function σ_{reg}, replacing the quadratic residuals $(\hat{y}_i - y)^2$ or $(\hat{\sigma}_i - \sigma_{\mathrm{reg}}(y))^2$ in (6.22) with, the corresponding, e.g., absolute or Winsorized difference. However, in such cases, one should take care in ensuring that the mean and standard deviation estimators are "unbiased" with respect to the corresponding metric; in the case of the absolute difference of the residuals, this means that the estimates $\hat{y}_i$ and $\hat{\sigma}_i$ should be median-unbiased rather than (mean-)unbiased.

6.7.6 Alternatives to least-squares initialization

The orthogonal (error-in-variables) least-squares may also be used in place of the standard least-squares. However, this results in only marginal differences in the solution $\left[\hat{a}_0, \hat{b}_0\right]$, because the errors in the variance estimates $\breve{\kappa}_{n_i}^2 \hat{\sigma}_i^2$ are usually much larger than those in the expectation estimates $\hat{y}_i$. Weighted least-squares solutions are not practical in this case, since the variance of the estimates, on which the weights should depend, is unknown.

6.8 Conclusions to the chapter

We presented and analyzed a Poissonian-Gaussian noise model for clipped (and non clipped) raw-data. An algorithm for the estimation of the model parameters from a single noisy image is proposed. The algorithm utilizes a special ML fitting of the parametric model on a collection of local wavelet-domain estimates of mean and standard-deviation. Experiments with synthetic images and real raw-data from camera sensors demonstrate the effectiveness and accuracy of the algorithm in estimating the model parameters and confirm the validity of the proposed model.

Conclusions to the thesis

C.1 Overview

In the first part of the thesis we have presented a very efficient and flexible image filtering approach based on shape-adaptive transforms. The key idea is to perform transform-domain shrinkage (i.e., hard-thresholding and Wiener filtering) with an arbitrarily-shaped transform's support which is adaptively defined for every point in the image. This ensures that the image is represented sparsely in the transform domain, allowing to effectively separate signal from noise. It results in high-quality denoising, with sharp detail preservation and without unpleasant artifacts introduced by the transform. Several extensions of this filtering approach have been shown, including color image filtering, deringing and deblocking for block-DCT compressed images, deconvolution (i.e. deblurring), as well as signal-dependent noise removal. As demonstrated by the extensive experiments, the proposed Pointwise Shape-Adaptive DCT filter achieves state-of-the-art results in all these diverse application scenarios. It is remarkable that these results are obtained using a low-complexity local transform, which makes the overall approach particularly attractive.

The second part of the thesis concerned noise modeling and estimation for the raw-data of digital imaging sensors. We presented a nonparametric method for estimation of the standard-deviation function from a set of images captured under the same fixed acquisition and illumination conditions. The target and the illumination are allowed to be non-uniform and the method does not require any calibration. Based on an automatic segmentation of the images, we separate samples with different expected output and estimate their standard-deviations. In this way, we exploit the non-uniformity by estimating the standard-deviation function over a large range of output values. Further, we proposed a Poissonian-Gaussian parametric model for raw-data noise and an algorithm which can estimate the model parameters automatically from a single noisy image. The proposed noise model explicitly considers the nonlinear response of the sensor in case of under- and over- exposure of the pixels. Experiments with synthetic images and real raw-data from camera sensors demonstrate the effectiveness and accuracy of the algorithm in estimating the model parameters and confirm that the raw-data can indeed be modeled as a clipped Poissonian-Gaussian process.

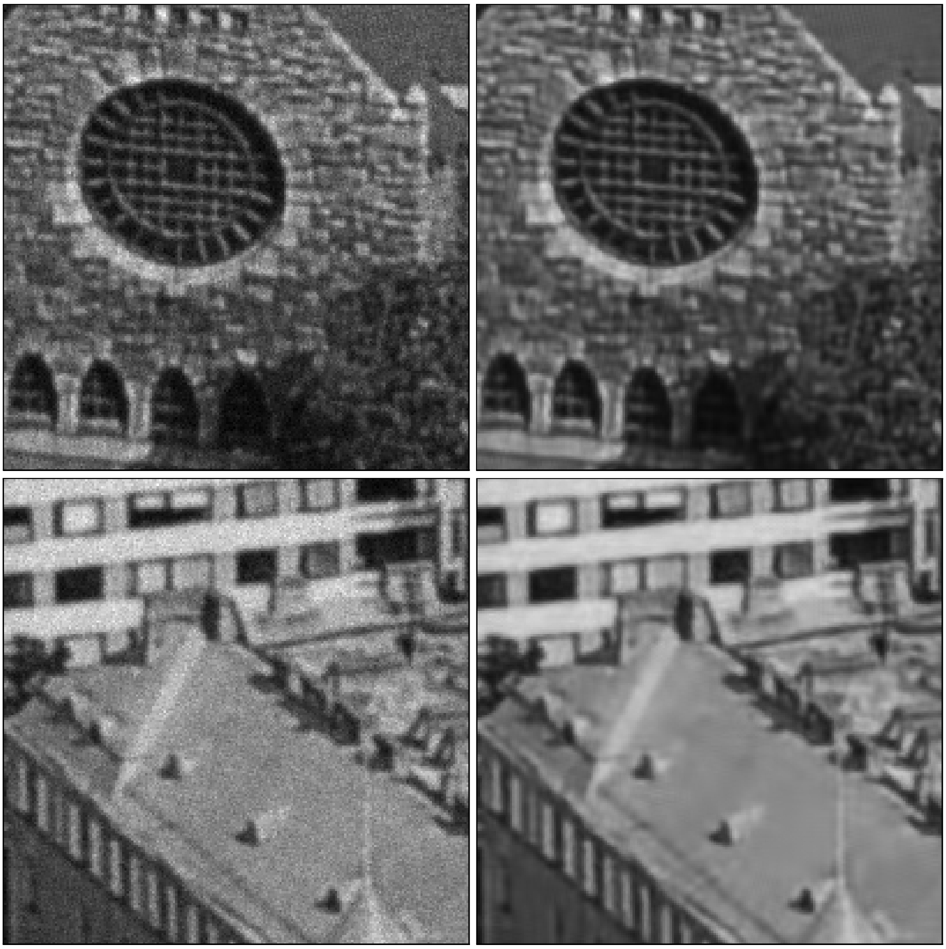


Figure C.1: Fragments of raw-data from CCD sensor before and after Pointwise SA-DCT denoising with noise parameters estimated automatically.

C.2 Future research

The Pointwise SA-DCT filtering algorithms presented in the first part of the thesis represent the current state-of-the-art for what concerns the methods based on local transform-domain estimation. Further extensions of the shape-adaptive approach shall involve non-local estimation, as in the Non-local means algorithm by Buades et al. [15], the Patch-based estimator by Kervrann and Boulanger [98], and our Block-Matching 3D (BM3D) filtering algorithm [32], which is based on simple fixed-size block-transforms.

In terms of applications, as shown by the diverse experiments shown in the thesis, the proposed filtering approach is rather versatile and it can be useful in many other situations than those considered in our work. In particular, a

Figure C.2: Fragments of raw-data from CCD sensor before and after Pointwise SA-DCT denoising with noise parameters estimated automatically.

straightforward extension of our Pointwise SA-DCT algorithm to 3-D data[14] has been developed by Lie et al. and successfully applied for the filtering of tensor-valued magnetic resonance imaging (MRI) data [9], [108].

The proposed Poissonian-Gaussian noise model for clipped raw-data is particularly suitable for describing the sensor response in the image composition stage of high dynamic range (HDR) imaging applications. In such scenario, multiple images acquired under different exposure conditions are combined (i.e., composed) in order to produce a HDR image. A peculiarity of these images is that they include vast underexposed and overexposed portions. It is known [135] that noise has a

[14] We note that the anisotropic LPA-ICI is applicable to multidimensional data and that 3-D implementations have been already exploited for video-denoising [47],[41]. The 2-D SA-DCT can also be naturally generalized to an arbitrary number of dimensions; in particular, the 3-D SA-DCT has been used in the past [113] for the coding of multispectral images.

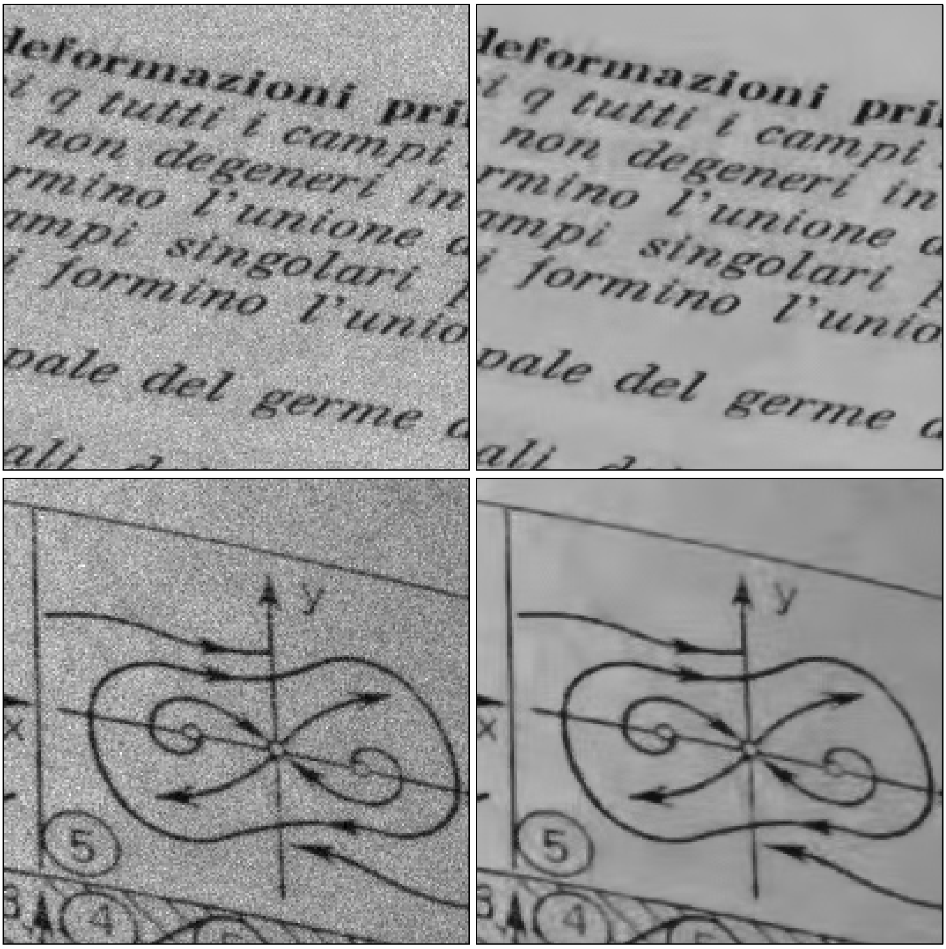


Figure C.3: Fragments of raw-data from CCD sensor before and after Pointwise SA-DCT denoising with noise parameters estimated automatically.

crucial impact on the composition process, as it affects the adaptive weights in the convex combinations used for composing the images. We expect that the accurate modeling of the noise and of the clipping will lead to an optimized design of the HDR composition process and eventually to higher quality in HDR imaging.

C.3 Automatic noise analysis and removal for raw-data

We conclude the thesis with illustrative examples, which show the methods proposed in Part I and in Part II integrated within a single procedure where:

1. the raw-data is analyzed and its noise parameters $\hat{a}$ and $\hat{b}$ are estimated,

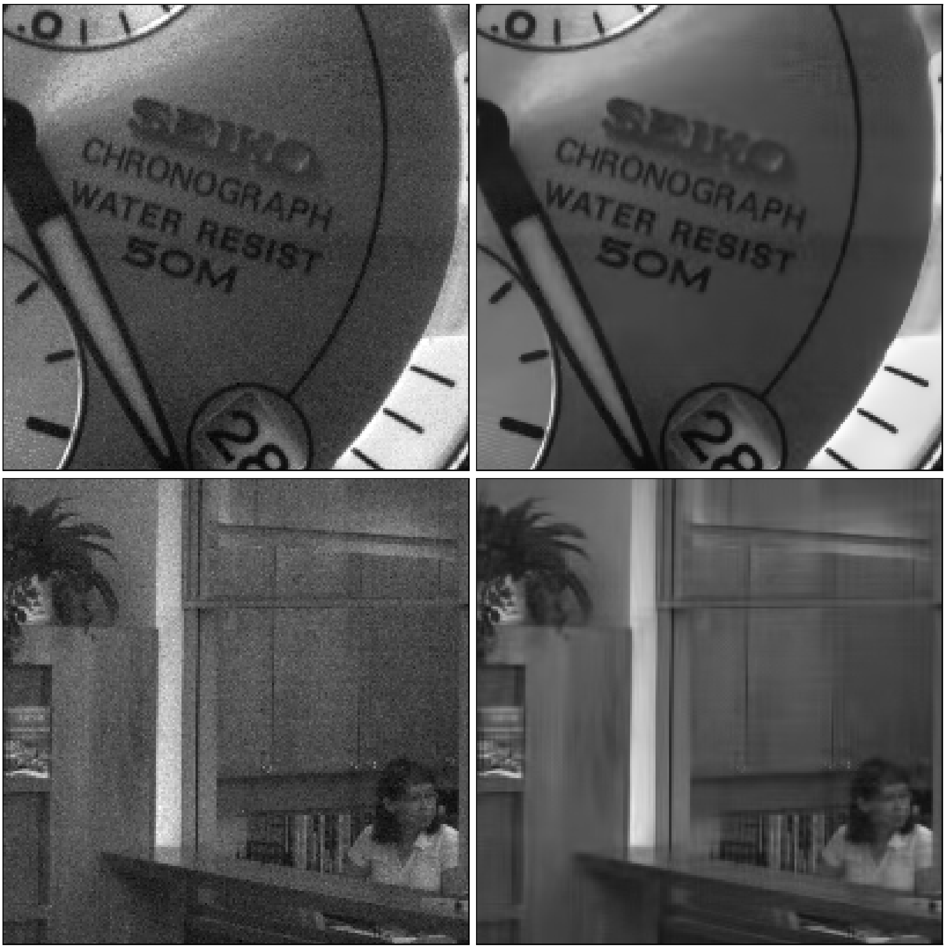


Figure C.4: Fragments of raw-data from CMOS sensor before and after Pointwise SA-DCT denoising with noise parameters estimated automatically.

yielding the standard deviation functions $\hat{\sigma}_{\text{fit}}$ and $\hat{\hat{\sigma}}_{\text{fit}}$ (algorithm from Section 6.5);

2. the raw-data is denoised according to the signal-dependent noise model with the variance function $\rho = \hat{\hat{\sigma}}^2_{\text{fit}}$ (algorithm from Section 3.3).

For these last examples, to avoid the complications due to color filter array interpolation (CFAI), we consider only a single color channel of the raw data[15], thus

[15] i.e., we work on a decimated subset of the sensor's pixels. For example, in the case of the traditional Bayer pattern [8] (where the pixel array has a periodic $\begin{smallmatrix} G_1 & R \\ B & G_2 \end{smallmatrix}$ configuration), it means that we take a fourth of the samples, decimating-by-two on the rows and on the columns. With a Fuji's Super CCD [160] the pixel configuration is octagonal and the whole green channel is already given on a rectangular grid.

Figure C.5: Fragments of raw-data from CMOS sensor before and after Pointwise SA-DCT denoising with noise parameters estimated automatically.

restricting ourself essentially to grayscale applications[16].

In Figures C.1, C.2, and C.3 we show fragments of raw-data from the 5.1-Mpixel SuperCCD HR sensor of a Fujifilm FinePix S5600 camera before and after Pointwise SA-DCT denoising with noise parameters estimated automatically. Figures C.4 and C.5 provide analogous image pairs with raw-data from the 1.3-Mpixel CMOS sensor of a Nokia cameraphone.

Finally, in Figure C.6, we give an illustration of the denoising of clipped raw-data, as suggested in Section 6.7.3. More precisely, in the figure, we show z, the Pointwise SA-DCT estimate $\hat{y}^{\mathrm{wi}}$ (3.10) of Section 3.3.2, and the inverse-

[16] We refer the interested reader to [128] and [129], where LPA-ICI techniques for CFAI of noise-free as well as of noisy raw-data are presented. We note that CFAI algorithms that exploit the SA-DCT can be developed based on these LPA-ICI CFAI techniques.

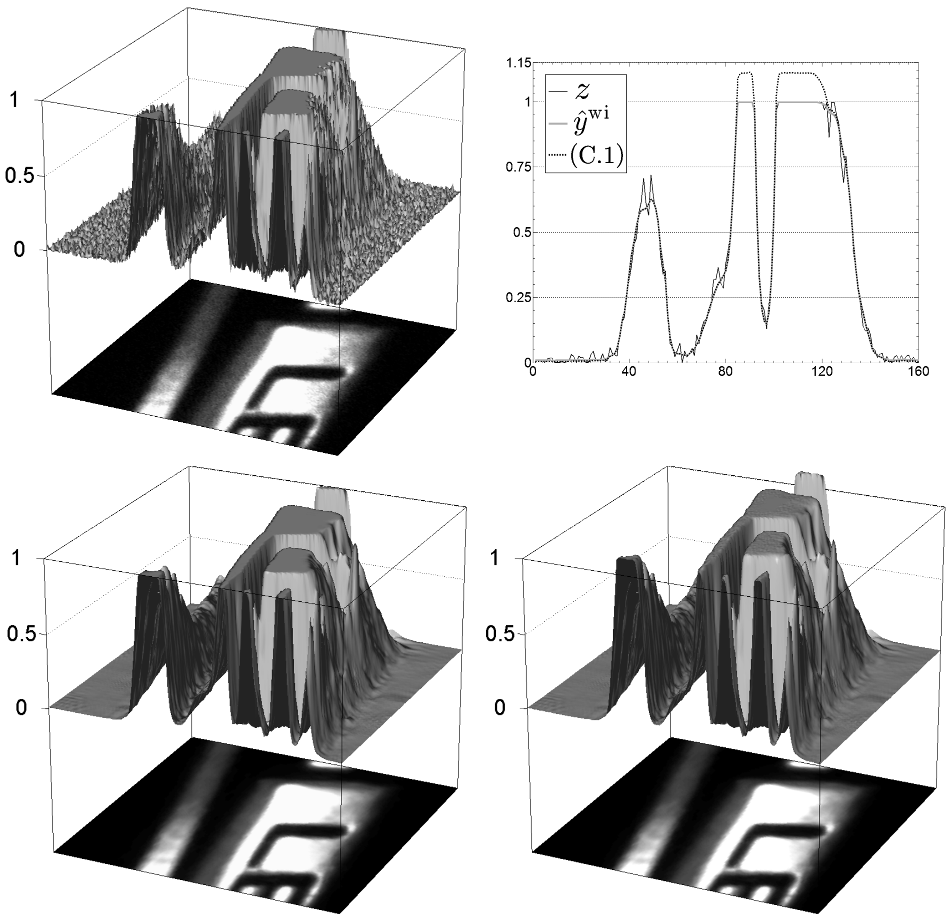


Figure C.6: Top left: fragment of noisy image z with some details overexposed. Observe the clipping of the noise visible in the brightest areas. Bottom: Pointwise SA-DCT estimate before ($\hat{y}^{\mathrm{wi}}$) and after compensation (C.1) of the bias due to clipped noise. Top right: cross-section of the three other subfigures. The compensation stretches the portions of the denoised signal which correspond to overexposed details.

transformed estimate

$$\hat{y}^{\mathrm{wi}}\mathcal{E}_r\left(\frac{\hat{y}^{\mathrm{wi}}}{\hat{\tilde{\sigma}}_{\mathrm{fit}}\left(\hat{y}^{\mathrm{wi}}\right)}\right)-\hat{y}^{\mathrm{wi}}+1-\left(1-\hat{y}^{\mathrm{wi}}\right)\mathcal{E}_r\left(\frac{1-\hat{y}^{\mathrm{wi}}}{\hat{\tilde{\sigma}}_{\mathrm{fit}}\left(\hat{y}^{\mathrm{wi}}\right)}\right), \tag{C.1}$$

which is obtained applying (6.47) of Section 6.3.2 on the estimate $\hat{y}^{\mathrm{wi}}$, with the standard-deviation function $\hat{\tilde{\sigma}}_{\mathrm{fit}}$ from Section 6.5.3 in place of the true $\tilde{\sigma}$. Note that the dynamic range of the inverse-transformed estimate (C.1) is about 11% wider than that of image $\hat{y}^{\mathrm{wi}}$ or z.

Bibliography

[1] Abramovich, F., and B.W. Silverman, "Wavelet Decomposition Approaches to Statistical Inverse Problems", *Biometrika*, vol. 85, no. 1., pp. 115-129, March, 1998.

[2] Acocella, E.C., and A. Alcaim, "Alignment by Phase of Vertical Coefficients in SA-DCT", *IEEE Signal Process. Lett.*, vol. 8, no. 2, pp. 42-44, Feb. 2001.

[3] Aghdasi, F., R.K. Ward, "Reduction of boundary artifacts in image restoration", *IEEE Trans. Image Process.*, vol. 5, no. 4, pp. 611-618, April 1996.

[4] Amemiya, T., "Regression Analysis when the Dependent Variable is Truncated Normal", *Econometrica*, vol. 41, no. 6, pp. 997-1016, Nov. 1973.

[5] Anscombe, F.J., "The Transformation of Poisson, Binomial and Negative-Binomial Data", *Biometrika*, vol. 35, no. 3/4, pp. 246-254, 1948.

[6] Averbuch, A., A. Schclar, and D.L. Donoho, "Deblocking of block-transform compressed images using weighted sums of symmetrically aligned pixels", *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 200-212, Feb. 2005.

[7] Banham, M.R., and A. K. Katsaggelos, "Spatially adaptive wavelet-based multiscale image restoration," *IEEE Trans. Image Process.*, vol. 5, no. 4, pp. 619-634, April 1996.

[8] Bayer, B.E., "Color imaging array", *U.S. Patent*, *no.* 3,971,065, July 1976.

[9] Bergmann, Ø., O. Christiansen, J. Lie, and A. Lundervold, "Shape Adaptive DCT for Denoising of Tensor Valued Images", to appear in *Journal of Digital Imaging*, (preprint available online at `https://bora.uib.no/bitstream/1956/2159/1/Paper+E_Johan+Lie.pdf`), Jan. 2007.

[10] Bertero, M., and P. Boccacci, *Introduction to inverse problems in imaging.* Inst. of Physics Publishing, 1998.

[11] Bi, M., S.H. Ong, and Y.H. Ang, "Coefficient grouping method for shape-adaptive DCT", *Electron. Lett.*, vol. 32, no. 3, pp. 201-202, Feb. 1996.

[12] Bi, M., S.H. Ong, and Y.H. Ang, "Comment on Shape-Adaptive DCT for generic coding of video", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 6, pp. 686-688, Dec. 1996.

[13] Blanksby, A.J., and M.J. Loinaz, "Performance analysis of a color CMOS photogate sensor", *IEEE Trans. Electron Devices*, vol. 47, no. 1, pp. 55-64, Jan. 2000.

[14] Blanksby, A.J., M.J. Loinaz, D.A. Inglis, and B.D. Ackland, "Noise performance of a color CMOS photogate image sensor", *IEEE Int. Electron Devices Meeting 97 Tech. Dig.*, pp. 205-208, 1997.

[15] Buades, A., B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one", *Multisc. Model. Simulat.*, vol. 4, no. 2, pp. 490-530, 2005.

[16] Chang, S.G., B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression", *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1532-1546, Sep. 2000.

[17] *CHDK firmware: Shooting to RAW with Canon PowerShot*, software and documentation, available online (as of Oct. 2007) at `http://chdk.wikia.com` and `http://digicanon.narod.ru`

[18] Chen, K.-H., J. Guo, J.-S. Wang, C.-W. Yeh, and J.-W. Chen, "An energy-aware IP core design for the variable-length DCT/IDCT targeting at MPEG4 shape-adaptive transforms", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 5, pp. 704-714, May 2005.

[19] Chen, T., H.R. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 584-602, Aug. 2001.

[20] Cho, D., and T.D. Bui, "Multivariate statistical modeling for image denoising using wavelet transforms", *Signal Process.: Image Comm.*, vol. 20, no. 1, pp. 77-89, Jan. 2005.

[21] Chute, C., and S.K. Bakhshi, *Worldwide Camera Phone and Videophone 2006-2010 Forecast*, IDC Market Analysis, Doc. #204456, Dec. 2006.

[22] Cleveland, W.S., and C. Loader, "Smoothing by local regression: principles and methods", *Statistical theory and computational aspects of smoothing*, Springer, New York, pp. 10-49, 1996.

[23] Cohen, A.C., *Truncated and Censored Samples*, CRC Press, 1991.

[24] Cohen, A.C., "Estimating the Mean and Variance of Normal Populations from Singly Truncated and Doubly Truncated Samples", *The Annals of Mathematical Statistics*, vol. 21, no. 4., pp. 557-569, Dec. 1950.

[25] Cohen, A.C., "On the Solution of Estimating Equations for Truncated and Censored Samples from Normal Populations", *Biometrika*, vol. 44, no. 1/2, pp. 225-236, June 1957.

[26] Cohen, A.C., "Simplified Estimators for the Normal Distribution When Samples Are Singly Censored or Truncated", *Technometrics*, vol. 1, no. 3, pp. 217-237, Aug., 1959.

[27] Cohen, A.C., "Tables for Maximum Likelihood Estimates: Singly Truncated and Singly Censored Samples", *Technometrics*, vol. 3, no. 4, pp. 535-541, Nov., 1961.

[28] Costantini, R., and S. Süsstrunk, "Virtual Sensor Design", *Proc. IS&T/SPIE Electronic Imaging 2004: Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications V*, vol. 5301, pp. 408-419, 2004.

[29] Croux, C., and P.J. Rousseeuw, "Time-Efficient Algorithms for Two Highly Robust Estimators of Scale", *Computational Statistics, Volume 1*, eds. Y. Dodge and J. Whittaker, Physika-Verlag, Heidelberg, pp. 411-428, 1992.

[30] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering", *Proc. SPIE El. Imaging 2006, Image Process.: Algorithms and Systems V*, 6064A-30, San Jose, California, USA, Jan. 2006.

[31] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Inverse halftoning by pointwise shape-adaptive DCT regularized deconvolution", *Proc. 2006 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006*, Florence, Sep. 2006.

[32] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering", *IEEE Trans. Image Process.*, vol. 16, no. 8, Aug. 2007.
MATLAB software available online at `http://www.cs.tut.fi/~foi/GCF-BM3D`

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

[34] Dias, J.M.B., "Fast GEM wavelet-based image deconvolution algorithm", *Proc. Int. Conf. Image Process., ICIP 2003*, vol. 2, 2003.

[35] Donoho, D., "Nonlinear solution of linear inverse problems by wavelet–vaguelette decomposition", *Appl. Comput. Harmon. Anal.*, vol. 2, pp. 101-126, 1995.

[36] Donoho, D.L., and I.M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage", *Biometrika*, n. 81, pp. 425-455, 1994.

[37] Egiazarian, K., V. Katkovnik, and J. Astola, "Local transform-based image de-noising with adaptive window size selection", *Proc. SPIE Image and Signal Processing for Remote Sensing VI*, vol. 4170, 4170-4, Jan. 2001.

[38] Elad, M., and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries", *IEEE Trans. on Image Process.*, vol. 15, no. 12, pp. 3736-3745, Dec. 2006. MATLAB software available online at `http://www.cs.technion.ac.il/~elad/Various/KSVD_Matlab_ToolBox.zip`

[39] Emery, M., A. Nemirovski, and D. Voiculescu, *Lectures on Probability Theory and Statistics - Ecole d'Eté de Probabilités de Saint-Flour XXVIII - 1998*, ed. P. Bernard, Springer-Verlag, LNM vol. 1738, 2000.

[40] Engl, H.W., M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, 1996.

[41] Ercole, C., A. Foi, V. Katkovnik, and K. Egiazarian, "Spatio-temporal pointwise adaptive denoising of video: 3D non-parametric approach", *Proc. 1st Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics, VPQM2005*, Scottsdale, AZ (USA), Jan. 2005.

[42] Fan, J., and I. Gijbels, *Local polynomial modelling and its application*, Chapman and Hall, London, 1996.

[43] Farnebäck, G., "A Unified Framework for Bases, Frames, Subspace Bases, and Subspace Frames" *Proc. of the 11th Scandinavian Conf. on Image Analysis,* Kangerlussuaq, Greenland, 1999.

[44] Ferwerda, J.A., S.N. Pattanaik, P. Shirley, and D.P. Greenberg, "A model of visual masking for computer graphics", *Proc. SIGGRAPH 97, Annual Conference Series*, *ACM SIGGRAPH*, Addison Wesley, pp. 143-152, Aug. 1997.

[45] Figueiredo, M.A.T., and R.D. Nowak, "An EM algorithm for wavelet-based image restoration", *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906-916, 2003.

[46] Figueiredo, M.A.T., and R.D. Nowak, "A bound optimization approach to wavelet-based image deconvolution", *Proc. IEEE 2005 Int. Conf. Image Process., ICIP 2005*, vol. 2, pp. 782-785, Genova, Sep. 2005.

[47] Foi, A., *Anisotropic nonparametric image processing: theory, algorithms and applications*, Ph.D. Thesis, Department of Mathematics, Politecnico di Milano, id. ERLTDD-D01290, April 2005. (available online at `http://www.cs.tut.fi/~foi/PhD_Thesis_Poli.html`)

[48] Foi, A., S. Alenius, V. Katkovnik, and K. Egiazarian, "Noise measurement for raw-data of digital imaging sensors by automatic segmentation of non-uniform targets", *IEEE Sensors Journal*, vol. 7, no. 10, pp. 1456-1461, Oct. 2007.

[49] Foi, A., S. Alenius, M. Trimeche, and V. Katkovnik, "Adaptive-size block transforms for Poissonian image deblurring", *Proc. 2006 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006*, Florence, Sep. 2006.

[50] Foi, A., S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian, "A spatially adaptive Poissonian image deblurring", *Proc. IEEE 2005 Int. Conf. Image Processing, ICIP 2005*, Sep. 2005.

[51] Foi, A., R. Bilcu, V. Katkovnik, and K. Egiazarian, "Anisotropic local approximations for pointwise adaptive signal-dependent noise removal", *Proc. XIII European Signal Proc. Conf., EUSIPCO 2005*, Sep. 2005.

[52] Foi, A., R. Bilcu, V. Katkovnik, and K. Egiazarian, "Adaptive-Size Block Transforms for Signal-Dependent Noise Removal", *Proc. 7th Nordic Signal Processing Symposium, NORSIG 2006*, Reykjavik, Iceland, June 2006.

[53] Foi, A., K. Dabov, V. Katkovnik, and K. Egiazarian, "Shape-adaptive DCT for denoising and image reconstruction", *Proc. SPIE El. Imaging 2006, Image Process.: Algorithms and Systems V*, 6064A-18, San Jose, California, USA, Jan. 2006.

[54] Foi, A., and V. Katkovnik, "From local polynomial approximation to pointwise shape-adaptive transforms: an evolutionary nonparametric regression perspective", *Proc. 2006 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006*, Florence, Sep. 2006.

[55] Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "A novel anisotropic local polynomial estimator based on directional multiscale optimizations", *Proc. 6th IMA Int. Conf. Math. in Signal Processing*, Cirencester (UK), pp. 79-82, 2004.

[56] Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "Inverse halftoning based on the anisotropic LPA-ICI deconvolution", *Proc. Int. TICSP Workshop Spectral Meth. Multirate Signal Proc., SMMSP 2004,* Vienna, pp. 49-56, Sep. 2004.

[57] Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT as an overcomplete denoising tool", *Proc. 2005 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2005*, pp. 164-170, Riga, June 2005.

[58] Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images", *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395-1411, May 2007. Matlab software available online at `http://www.cs.tut.fi/~foi/SA-DCT`

[59] Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT denoising with structure preservation in luminance-chrominance space", *Proc. 2nd Int. Workshop Video Process. Quality Metrics Consum. Electron., VPQM2006*, Scottsdale, AZ, Jan. 2006.

[60] Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality deblocking of compressed color images", *Proc. 14th Eur. Signal Process. Conf., EUSIPCO 2006*, Florence, Sep. 2006.

[61] Foi, A., V. Katkovnik, and K. Egiazarian, "Signal-dependent noise removal in Pointwise Shape-Adaptive DCT domain with locally adaptive variance", *Proc. 15th Eur. Signal Process. Conf., EUSIPCO 2007*, Poznań, Sep. 2007.

[62] Foi, A., D. Paliy, V. Katkovnik, and K. Egiazarian, "Anisotropic nonparametric image restoration demobox" (MATLAB software), *Local Approximations in Signal and Image Processing (LASIP) Project*, `http://www.cs.tut.fi/~lasip/` , 2005.

[63] Foi, A., M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single image raw-data", to appear in *IEEE Trans. Image Processing.*

[64] Foi, A., V. Katkovnik, D. Paliy, K. Egiazarian, M. Trimeche, S. Alenius, R. Bilcu, M. Vehvilainen, "Apparatus, method, mobile station and computer program product for noise estimation, modeling and filtering of a digital image", *U.S. Patent* (Applications no. 11/426,128, June 2006, and no. 11/519,722, Sep. 2006).

[65] Forbes, L., H. Gopalakrishnan, and W. Wanalertlak, "Analysis and simulation of noise in correlated double sampling imager circuits", *Proc. SPIE, Noise in Devices and Circuits III*, vol. 5844, pp. 238-247, May 2005.

[66] Frost, V. S., J.A. Stiles, K.S. Shanmugan, and J.C. Holtzman, "A Model for Radar Images and its Application to Adaptive Digital Filtering of Multiplicative Noise", *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 4, no. 2, pp. 157-166, March 1982.

[67] Galatsanos, N.P., and K.A. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation", *IEEE Trans. Image Process.*, vol. 1, no. 3, pp. 322-336, July 1992.

[68] Gallaire, J.-P.G., and A.M. Sayeed, "Wavelet-based empirical Wiener filtering", *Proc. IEEE-SP Int. Symp. Time-Frequency and Time-Scale Analysis*, pp. 641-644, Oct 1998.

[69] Ghael, S.P., A.M. Sayeed, and R.G. Baraniuk, "Improved Wavelet Denoising via Empirical Wiener Filtering", *Proc. SPIE Wavelet Applications in Signal and Image Processing V*, vol. 3169, pp. 389-399, July 1997.

[70] Gilge, M., T. Engelhardt, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform", *Signal Process.: Image Comm.*, vol. 1, no. 2, pp. 153-180, Oct. 1989.

[71] Gilge, M., "Region oriented transform coding (ROTC) of images", *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-90*, Albuquerque, USA, vol. 4, pp. 2245-2248, April 1990.

[72] Goldenshluger, A., and A. Nemirovski, "On spatial adaptive estimation of nonparametric regression", *Math. Meth. Statistics*, vol. 6, pp. 135-170, 1997.

[73] Greene, W.H., *Econometric Analysis*, 4$^{\text{th}}$ ed., Prentice Hall, 2000.

[74] Grossberg, M.D., and S.K. Nayar, "Modeling the Space of Camera Response Functions", IEEE Trans. Pattern Analysis and Machine Intelligence, vol.26, no.10, pp. 1272-1282, Oct. 2004.

[75] Guleryuz, O.G., "Weighted overcomplete denoising", *Proc. Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, Nov. 2003.

[76] Hampel, F.R., "The influence curve and its role in robust estimation", *Journal of American Statistical Association*, 62, pp. 1179-1186, 1974.

[77] Hernández, E., and G. Weiss, *A first course on wavelets*, CRC Press, 1996.

[78] Hirakawa, K., and T.W. Parks, "Image denoising using total least squares", *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2730-2742, Sept. 2006. MATLAB software available online at http://www.accidentalmark.com/research/packages/TLSdenoise.zip

[79] Hsung, T.-C., and D.P.-K. Lun, "Application of singularity detection for the deblocking of JPEG decoded images", *IEEE Trans. Circuits Syst. II*, vol. 45, no. 5, pp. 640-644, May 1998.

[80] Huang,. J., and D. Mumford, "Statistics of natural images and models", *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR 1999*, vol. 1., pp. 541-547, June 1999.

[81] Huber, P.J., *Robust statistics*, John Wiley & Sons Inc., 1981.

[82] Hytti, H., "Characterization of digital image noise properties based on RAW data" *Proc. SPIE-IS&T Electronic Imaging, Image Quality and System Performance III*, vol. 6059, 60590A, Jan. 2006.

[83] Independent JPEG Group, http://www.ijg.org

[84] Jain, A.K., *Fundamentals of digital image processing*, Prentice-Hall, 1989.

[85] Janesick, J., "Lux transfer: Complementary metal oxide semiconductors versus charge-coupled devices", *Optical Engineering*, vol. 41, no. 6, pp. 1203-1215, June 2002.

[86] Jansen, M., *Wavelet Thresholding and Noise Reduction*, Ph.D. Thesis, Katholieke Universiteit Leuven, April 2000.

[87] Jiang, S.S., and A.A. Sawchuk, "Noise updating repeated Wiener filter and other adaptive noise smoothing filters using local image statistics", *Appl. Opt.*, vol. 25, pp. 2326-2337, 1986.

[88] Johnson, N.L., S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, 2nd ed., Wiley Series in Probability and Statistics, 1994.

[89] Kalifa, J., and S. Mallat, "Thresholding Estimators for Linear Inverse Problems and Deconvolutions", *Annals of Statistics*, vol. 31, no. 1, pp. 58-109, 2003.

[90] Katkovnik, V., “A new method for varying adaptive bandwidth selection”, *IEEE Trans. Signal Process.,* vol. 47, no. 9, pp. 2567-2571, 1999.

[91] Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, “Directional varying scale approximations for anisotropic signal processing”, *Proc. XII European Signal Proc. Conf., EUSIPCO 2004,* Vienna, pp. 101-104, Sep. 2004.

[92] Katkovnik, V., K. Egiazarian, and J. Astola, *Local Approximation Techniques in Signal and Image Processing*, SPIE Press, Monograph Vol. PM157, Sep. 2006.

[93] Katkovnik, V., K. Egiazarian, and J. Astola, “Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule”, *J. Math. Imaging and Vision*, vol. 16, no. 3, pp. 223-235, 2002.

[94] Katkovnik, V., K. Egiazarian, and J. Astola, “A spatially adaptive nonparametric regression image deblurring”, *IEEE Trans. on Image Process.*, vol. 14, no. 10, pp. 1469-1478, Oct. 2005.

[95] Kauff, P., and K. Schuur, “An extension of shape-adaptive DCT (SA-DCT) towards DC separation and ΔDC correction”, *Proc. of 1997 Picture Coding Symposium*, pp. 647-652, Sep. 1997.

[96] Kauff, P., and K. Schuur, “Shape-adaptive DCT with block-based DC separation and ΔDC correction”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 237-242, 1998.

[97] Kaup, A., and S. Panis, “On the performance of the shape adaptive DCT in object-based coding of motion compensated difference images”, *Proc. of 1997 Picture Coding Symposium*, pp. 653-657, 1997.

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

[99] Khasawneh, M.T., S.R. Bowling, S. Kaewkuekool, B.R. Cho, “Tables of a Truncated Standard Normal Distribution: A Singly Truncated Case”, *Quality Engineering*, vol. 17, no. 1, pp. 33-50, 2005.

[100] Kinane, A., V. Muresan, and N. O’Connor, “Optimal adder-based hardware architecture for the DCT/SA-DCT”, *Proc. SPIE Visual Comm. Image Process. Conf., VCIP 2005*, Beijing, China, 12-15 July 2005.

[101] Kinane, A., A. Casey. V. Muresan, and N. O’Connor, “FPGA-based conformance testing and system prototyping of an MPEG-4 SA-DCT hardware accelerator”, *IEEE 2005 Int. Conf. on Field-Programmable Technology, FPT’05*, Singapore, Dec. 2005.

[102] Koenen, R., “Overview of the MPEG-4 Standard”, ISO/IEC JTC1/SC29/WG11 Doc. N3536, July 2000.

[103] Kuan, D.T., A.A. Sawchuk, T.C. Strand, and P. Chavel, "Adaptive noise smoothing filter for images with signal dependent noise," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 7, pp. 165-177, 1985.

[104] Lansel, S., D. Donoho, and T. Weissman, "DenoiseLab: A Standard Test Set and Evaluation Method to Compare Denoising Algorithms", MATLAB software online at `http://www.stanford.edu/~slansel/DenoiseLab/`.

[105] Lee, J.S., "Digital image enhancement and noise filtering by using local statistics", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, 1980.

[106] Lee, J.S., "Refined filtering of image noise using local statistics", *Comput. Graph. Image Proc.* vol. 15, pp. 380-389, 1981.

[107] Lee, N., *Wavelet-Vaguelette Decompositions and Homogeneous Equations*, Ph.D. thesis, Purdue University, Dec. 1997.

[108] Lie, J., *Mathematical Imaging with Applications to MRI and Diffusion Tensor MRI*, Ph.D. Thesis, Department of Mathematics, The University of Bergen, March 2007. (available online at `https://bora.uib.no/bitstream/1956/2159/6/Main+Thesis_Johan+Lie.pdf`)

[109] Liew, A.W.C., and H. Yan, "Blocking Artifacts Suppression in Block-Coded Images Using Overcomplete Wavelet Representation", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 450-461, Apr. 2004.

[110] Liu, C., W.T. Freeman, R. Szeliski, and S.B. Kang, "Noise estimation from a single image", *Proc. IEEE Conf. Computer Vision and Pattern Recognition, CVPR 2006*, pp. 901-908, 2006.

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

[112] Mallat, S., *A wavelet tour of signal processing*, 2nd edition, Academic Press, 1998.

[113] Markman, D. and D. Malah, "Hyperspectral image coding using 3D transforms", *Proc. 2001 Int. Conf. Image Process., ICIP 2001*, vol. 1, pp. 114-117, Thessaloniki, Greece, Oct. 2001.

[114] Marsi, S., R. Castagno, and G. Ramponi, "A simple algorithm for the reduction of blocking artifacts in images and its implementation", *IEEE Trans. Consum. Electron.*, vol. 44, no. 3, pp. 1062-1070, Aug. 1998.

[115] Mignotte, M., "An adaptive segmentation-based regularization term for image restoration", *Proc 12th IEEE Int. Conf. Image Process., ICIP'05*, vol. 1, pp. 901-904, Italy, Genova, Sep. 2005.

[116] "MPEG-4 video verification model version 18.0 (VM-18)", ISO/IEC JTC1/SC29/WG11, Doc. N3908, 2001.

[117] MPEG-4: Information technology - Coding of audio-visual objects - Part 2: Visual, ISO/IEC 14496-2:2001, Dec. 2001.

[118] Montgomery, D., and G. Runger, *Applied Statistics and Probability for Engineers*, 3rd edition, Wiley and sons, 2003.

[119] Mosteller, F., and J.W. Tukey, *Data Analysis and Regression: A Second Course in Statistics*, Addison Wesley, 1997.

[120] Neelamani, R., H. Choi, and R. Baraniuk, "Wavelet-domain regularized deconvolution for ill-conditioned systems", *Proc. IEEE Int. Conf. Image Process., ICIP '99*, pp. 204-208, Kobe, Japan, Oct. 1999.

[121] Neelamani, R., H. Choi, and R. Baraniuk, "Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems", *IEEE Trans. Image Process.,* vol. 52, no. 2, 2004.

[122] Nelder, J.A., and R. Mead, "A simplex method for function minimization", *The Computer Journal*, vol. 7, pp. 308-313, 1965.

[123] O'Connor, N., S. Sav, T. Adamek, V. Mezaris, I. Kompatsiaris, T.Y. Lu, E. Izquierdo, C. Bennström, and J. Casas, "Region and object segmentation algorithms in the Qimera segmentation platform", *Proc. Third Int. Workshop on Content-Based Multimedia Indexing (CBMI03)*, Rennes, pp. 381-388, 2003.

[124] Öktem, H., V. Katkovnik, K. Egiazarian, and J. Astola, "Local adaptive transform based image de-noising with varying window size", *Proc. IEEE Int. Conf. Image Process., ICIP 2001,* Thessaloniki, Greece, pp. 273-276, Oct. 2001.

[125] O'Neill, R., "Algorithm AS 47: Function Minimization Using a Simplex Procedure", *Applied Statistics*, vol. 20, no. 3, pp. 338-345, 1971.

[126] Ostermann, J., E.S. Jang, J. Shin, and T. Chen, "Coding of arbitrarily shaped video objects in MPEG-4", *Proc. Int. Conf. Image Process., ICIP 1997*, pp. 496-499, 1997.

[127] Paek, H., R.C. Kim, and S.U. Lee, "On the POCS-based postprocessing technique to reduce blocking artifacts in transform coded images," *IEEE Trans. Circuits Sys. Video Technology*, vol. 8, pp. 358-367, June 1998.

[128] Paliy, D., V. Katkovnik, R. Bilcu, S. Alenius, K. Egiazarian, "Spatially Adaptive Color Filter Array Interpolation for Noiseless and Noisy Data", *Int. J. Imaging Systems and Technology (IJIST), Special Issue on Applied Color Image Processing*, vol. 17, no. 3, pp. 105-122, Oct. 2007.

[129] Paliy, D., M. Trimeche, V. Katkovnik, S. Alenius, "Demosaicing of Noisy Data: Spatially Adaptive Approach", *Proc. SPIE-IS&T Electronic Imaging 2007, Computational Imaging IV*, vol. 6497, San Jose, CA, Jan. 2007.

[130] Persson, T., and H. Rootzen, "Simple and Highly Efficient Estimators for a Type I Censored Normal Sample", *Biometrika*, vol. 64, no. 1, pp. 123-128, April 1977.

[131] Philips, W., "Orthogonal base functions on a discrete two-dimensional region", *ELIS Technical Report DG 91-20*, Dept. of Electronics and Inform. Syst., Universiteit Gent, Belgium, Nov. 1993.

[132] Philips, W., "A Fast Algorithm for Orthogonalizing Polynomials on an Arbitrarily Shaped Region (Revised Version)", *Multidimensional Systems and Signal Processing*, vol. 8, no. 4, pp. 409-421, Oct. 1997.

[133] "Photography - Electronic still-picture cameras - Methods for measuring opto-electronic conversion functions (OECFs)", ISO 14524:1999, Dec. 1999.

[134] "Photography - Electronic still-picture imaging - Noise measurements", ISO 15739:2003, May 2003.

[135] Pirinen, O., A. Foi, and A. Gotchev, "Color High Dynamic Range Imaging: The Luminance-Chrominance Approach", *Int. J. Imaging Systems and Technology (IJIST)*, *Special Issue on Applied Color Image Processing*, vol. 17, no. 3, pp. 152-162, Oct. 2007.

[136] Pizurica, A., and W. Philips, "Estimating probability of presence of a signal of interest in multiresolution single- and multiband image denoising", *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 654-665, Mar. 2006.
MATLAB software available online at
`http://telin.ugent.be/~sanja/Sanja_files/WaveletDEN.htm`

[137] Plataniotis, K.N., and A.N. Venetsanopoulos, *Color image processing and applications*, Springer-Verlag New York, Inc., New York, NY, 2000.

[138] Portilla, J., V. Strela, M. Wainwright, and E.P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain", *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338-1351, Nov. 2003.
MATLAB software available online at
`http://decsai.ugr.es/~javier/denoise/software/BLS-GSM_Denoising.zip`

[139] Prucnal, P.R., and B.E.A. Saleh, "Transformation of image-signal-dependent noise into image signal-independent noise", *Optics Letters*, vol. 6, no. 7, July 1981.

[140] Rangarayyan, R.M., M. Ciuc, and F. Faghih, "Adaptive-neighborhood filtering of images corrupted by signal-dependent noise", *Appl. Opt.*, vol. 37, pp. 4477-4487, 1998.

[141] Rao, K.R., and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages and Applications*, Academic Press, 1990.

[142] Reeves, S.J., "Fast image restoration without boundary artifacts", *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1448-1453, Oct. 2005.

[143] Rhodes, H., G. Agranov, C. Hong, U. Boettiger, R. Mauritzson, J. Ladd, I. Karasev, J. McKee, E. Jenkins, W. Quinlin, I. Patrick, J. Li, X. Fan, R. Panicacci, S. Smith, C. Mouli, J. Bruce, "CMOS imager technology shrinks and image performance", *Proc. 2004 IEEE Workshop on Microelectronics and Electron Devices, WMED*, pp. 7-18, April 2004.

[144] Rosenholts, R., and A. Zakhor, "Iterative procedures for reduction of blocking artifacts in transform domain image coding", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 2, pp. 91-95, Mar. 1992.

[145] Rousseeuw, P.J., and C. Croux, "Alternatives to the Median Absolute Deviation", *J. American Statistical Assoc.*, vol. 88, no. 424. pp. 1273-1283, Dec. 1993.

[146] Scheunders, P., "Wavelet thresholding of multivalued images", *IEEE Trans. Image Process.*, vol. 13, pp. 475-483, Apr. 2004.

[147] Scheunders, P., and J. Driesen, "Least squares interband denoising of color and multispectral images", *Proc. IEEE Int. Conf. Image Process., ICIP 2004*, Singapore, Oct. 2004.

[148] Schneider, H., "Simple and Highly Efficient Estimators for Censored Normal Samples", *Biometrika*, vol. 71, no. 2, pp. 412-414, Aug. 1984.

[149] Selvin, S. "A Graphical Estimate of the Population Mean from Censored Normal Data", *Applied Statistics*, vol. 25, no. 1, pp. 8-11, 1976.

[150] Şendur, L., and I. Selesnick, "Bivariate Shrinkage Functions for Wavelet-Based Denoising Exploiting Interscale Dependency", *IEEE Trans. Signal Process.*, vol. 50, no. 11, pp. 2744-2756, Nov. 2002.

[151] Şendur, L., and I. Selesnick, "Bivariate shrinkage with local variance estimation", *IEEE Signal Processing Letters*, vol. 9, no. 12, pp. 438-441, Dec. 2002. MATLAB software available online at
`http://taco.poly.edu/WaveletSoftware/denoise2.html`

[152] Sheikh, H.R., Z. Wang, L. Cormack, and A.C. Bovik, "LIVE Image Quality Assessment Database Release 2", available online at
`http://live.ece.utexas.edu/research/quality`

[153] Shen, M.-Y., and C.-C.J. Kuo, "Review of Postprocessing Techniques for Compression Artifact Removal", *Journal of Visual Communication and Image Representation*, vol. 9, no. 1, pp. 2–14, March, 1998.

[154] Sikora, T., "Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments", *Signal Processing: Image Communication*, vol. 7, pp. 381-395, Nov. 1995.

[155] Sikora, T., S. Bauer, and B. Makai, "Efficiency of shape-adaptive 2-D transforms for coding of arbitrarily shaped image segments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 3, pp. 254-258, June 1995.

[156] Sikora, T., and B. Makai, "Shape-adaptive DCT for generic coding of video", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 1, pp. 59-62, Feb. 1995.

[157] Skodras, A., C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard", *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36-58, Sep. 2001.

[158] Starck, J.L., E.J. Candès, and D.L. Donoho, "The curvelet transform for image denoising", *IEEE Trans. Image Process.,* vol. 11, no.6., pp. 670–684, June 2002.

[159] Stasinski, R., and J. Konrad, "Reduced-complexity shape-adaptive DCT for region-based image coding", *Proc. IEEE Int. Conf. Image Process., ICIP 1998*, Chicago, pp. 114-118, Oct. 1998.

[160] Tamayama, H., K. Ito, and T. Nishimura, "Technology trends of high-definition digital still camera systems", *Sym. VLSI Circuits, Digest of Technical Papers*, pp. 100-105, June 2002.

[161] Theuwissen, A., *Course on camera system*, Lecture notes, CEI-Europe, 2005.

[162] Thompson, A.M., J.C. Brown, J.W. Kay, and D.M. Titterington, "A Study of Methods of Choosing the Smoothing Parameter in Image Restoration by Regularization", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 326-339, April 1991.

[163] Tian, H., B. Fowler, and A. El Gamal, "Analysis of Temporal Noise in CMOS Photodiode Active Pixel Sensor", *IEEE Journal of Solid-State Circuits*, vol. 36, no. 1, pp. 92–101, Jan. 2001.

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

[165] Tobin, J., "Estimation of Relationships for Limited Dependent Variables", *Econometrica*, vol. 26, no. 1., pp. 24-36, Jan. 1958.

[166] Van der Weken, D., E. Kerre, E. Vansteenkiste, and W. Philips, "Evaluation of fuzzy image quality measures using a multidimensional scaling framework", *Proc. 2nd Int. Workshop Video Process. Quality Metrics Consum. Electron., VPQM2006*, Scottsdale, AZ, Jan. 2006.

[167] Vansteenkiste, E., D. Van der Weken, W. Philips, and E. Kerre, "Perceived image Quality Measurement of state-of-the-art Noise Reduction Schemes", *Lecture Notes in Computer Science 4179 - ACIVS 2006*, pp. 114-124, Springer, Sep. 2006.

[168] Vogel, C.R., *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics, vol. 23, SIAM, Philadelphia, PA, 2002.

[169] Wach, H.B., and E.R. Dowski Jr., "Noise modeling for design and simulation of color imaging systems", *Proc. of IS&T/SID's 12th Color Imaging Conference, CIC 12,* 2004.

[170] Wach, H., and E.R. Dowski Jr., “Noise modeling for design and simulation of computational imaging systems”, *Proc. SPIE, Visual Information Processing XIII*, vol. 5438, pp. 159-170, July 2004.

[171] Wallace, G.K., “The JPEG still picture compression standard”, *IEEE Trans. Consum. Electronics*, vol. 38, no. 1, pp. xviii-xxxiv, Feb. 1992.

[172] Wang, Z., A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, “Image quality assessment: From error visibility to structural similarity”, *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, April 2004.

[173] Willett, R.M., and R.D. Nowak, “Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging”, *IEEE Trans. Medical Imaging,* vol. 22, no. 3, pp. 332-350, 2003.

[174] Winkler, S., and S. Süsstrunk, “Visibility of noise in natural images”, *Proc. IS&T/SPIE Electronic Imaging 2004: Human Vision and Electronic Imaging IX*, vol. 5292, pp. 121-129, 2004.

[175] Wong, H.-S., D.J. Frank, P. Solomon, C. Wann, and J. Wesler, “Nanoscale CMOS”, *Proc. of the IEEE*, vol. 87, no. 4, pp. 537-570, April 1999.

[176] Wu, S., H. Yan, and Z. Tan, “An efficient wavelet-based deblocking algorithm for highly compressed images”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 11, pp. 1193-1198, Nov. 2001.

[177] Xiong, Z., M. Orchard, and Y. Zhang, “A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations”, *IEEE Trans. Circuits Syst. Video Techn.*, vol. 7, no. 4, pp. 433-437, Apr. 1997.

[178] Yang, Y., N.P. Galatsanos, and A.K. Katsaggelos, “Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images”, *IEEE Trans. Circuits Syst. Video Techn.*, vol. 3, no. 6, pp. 421-432, Dec. 1993.

[179] Yang, Y., N.P. Galatsanos, and A.K. Katsaggelos, “Projection-based spatially adaptive reconstruction of block-transform compressed images”, *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 896-908, July 1995.

[180] Yaroslavsky, L., K. Egiazarian, and J. Astola, *Transform Domain Image Restoration Methods, Review, Comparison and Interpretation*, TICSP Series no. 9, Tampere, 2000.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland