UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Processing wavelet compression artifacts in high-resolution satellite imagery

Mario González Olmedo

Programa de Posgrado en Ingeniería Matemática
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Diciembre de 2016

UNIVERSIDAD DE LA REPUBLICA URUGUAY

FACULTAD DE INGENIERIA

CENUR Litoral Norte Salto

Processing wavelet compression artifacts in high-resolution satellite imagery

Mario González Olmedo

Tesis de Maestría presentada al Programa de Posgrado en Ingeniería Matemática, Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Magister en Ingeniería Matemática.

Directores:

Dr. Andrés Almansa

Dr. Pablo Musé

Director académico:

Dr. Roberto Markarian

Montevideo – Uruguay

Diciembre de 2016

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

Dr. Antoni Buades (Universitat de les Illes Balears, España)

Dr. Roberto Markarián (Fac. de Ingeniería, UdelaR)

Dr. Lionel Moisan (Université Paris Descartes, Francia)

Dr. Álvaro Pardo (Fac. de Ingeniería, UCUDAL)

Dr. Ignacio Ramirez (Fac. de Ingeniería, UdelaR)

Dr. Andrés Almansa (CNRS / Université Paris Descartes, Francia)

Dr. Pablo Musé (Fac. de Ingeniería, UdelaR)

Montevideo – Uruguay
Diciembre de 2016

Acknowledgments

I would like to express my special appreciation and thanks to my advisors Dr. Andrés Almansa and Dr. Pablo Musé, you have been tremendous mentors for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advices on both research as well as on my career have been priceless. Also, I would like to thank my colleagues from Universidad de la República for their wonderful collaboration, especially to Dr. José Vieitez. You supported me greatly and were always willing to help me.

I also want to thank my other committee members, Dr. Antoni Buades, Dr. Roberto Markarián, Dr. Lionel Moisan, Dr. Álvaro Pardo and Dr. Ignacio Ramirez. Thank you for investing time and providing interesting and valuable feedback. I feel proud and honoured that you have accepted to be on my committee.

At the end I would like express appreciation to my beloved wife Dalma Latorre who spent sleepless nights with and was always my support in the moments when there was no one to answer my queries, and also to my son Samuel and my mother Margarita. I must express my very profound gratitude to all of them for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abstract

JPEG and Wavelet compression artifacts leading to Gibbs effects and loss of texture are well known and many restoration solutions exist in the literature. So is denoising, which has occupied the image processing community for decades. However, when a noisy image is compressed, a new kind of artifact may appear from the interaction of both degradations. This new kind of artifact is surprisingly never mentioned or studied in the image processing community, with only a few rare exceptions. Yet the importance of such artifacts in very high resolution satellite imaging has recently been recognized. Indeed, such images are mainly used for highly accurate subpixel stereo vision, an application where the presence of this kind of artifact (even if barely visible) is particularly harmful. In this work we present a thorough probabilistic analysis of the kind of degradation that results from the interaction of noise and compression called *wavelet outliers* , and conclude that their probabilistic nature is characterized by a single parameter q/σ that can be inferred from a noise model and a compression model. This analysis provides the conditional probability for a Bayesian MAP estimator, whereas a patch-based local Gaussian prior model is learnt from the corrupted image iteratively, like in state of the art denoising algorithms (non-local Bayes), albeit with the additional difficulty of dealing with non-Gaussian noise during the learning process. The resulting joint denoising and decompression algorithm has been experimentally evaluated under realistic conditions. The results show its ability to simultaneously denoise, decompress and remove wavelet outliers better than the available alternatives, both from a quantitative and a qualitative point of view. As expected, the advantage of our method is more evident for large values of q/σ, a situation that naturally occurs in satellite images containing very dark areas (shadows).

Contents

Chapter 1

Introduction

Image Processing can be broadly defined as the manipulation of signals which are inherently multidimensional. The most common such signals nowadays are photographs and video sequences taken with cellphones or digital cameras, but the field also includes medical (ultrasound, X-ray tomography) and multispectral images. Some of the goals of image processing can be

- Enhancement or correction for human visualization (*color correction*, *texture editing*)
- Restoration from blur (*deblurring*) or noise (*denoising*).
- Compression for storage or transmission
- Analysis, segmentation, recognition, and understanding

The use of image processing techniques has become almost ubiquitous. They find applications in such diverse areas as astronomy, archaeology, medicine, video communication, and satellite imagery. Its study incorporates and in many cases motivates the development of modern mathematical tools, in order to to accompany the growing demand on this type of processing.

This thesis is about the study of two of the areas mentioned above, denoising and compression, in the context of high-resolution satellite imagery. These satellites generate a great amount of information that must be stored on the satellite and transmitted to the ground stations when possible. Both processes are very expensive in terms of data size and this is the main reason of compressing the data on space before sending it to Earth. As we shall see, high compression ratios are achieved only using an irreversible process called *quantization*, and jointly with the noise, it causes several artifacts in the decompressed image that we propose to correct.

Another ingredient that will be present is the theory of *Wavelets*. This analysis tool arises as an abstraction of various common techniques that have been applied in many contexts, and thus it was widely used as an alternative

of the Fourier transform, in particular in denoising and compression of digital images, as we shall present in the next section.

1.1 Wavelet analysis

In this section, we review the main ideas behind wavelet analysis. We begin by giving the motivation of searching for time and frequency localized functional basis, and how wavelets provide tools that can be useful for analyzing non-stationary signals unlike Fourier transform. For a complete treatment of wavelet analysis, see [23].

1.1.1 Motivation

The search for basis functions for *analyzing* a function f, that is, to decompose f as a combination of basic elements that describes the behavior of the funtion, goes back at least as far as Jean-Baptiste Joseph Fourier (1768-1830) who used complex sinusoids [31]. The Fourier transform of a continuous time signal $f \in L^2(\mathbb{R})$ can be defined by

$$\mathcal{F}(f)(\omega) = \hat{f}(\omega) = \langle f, e^{i\omega\cdot} \rangle = \int_{\mathbb{R}} f(t) e^{-i\omega t}\, dt. \tag{1.1}$$

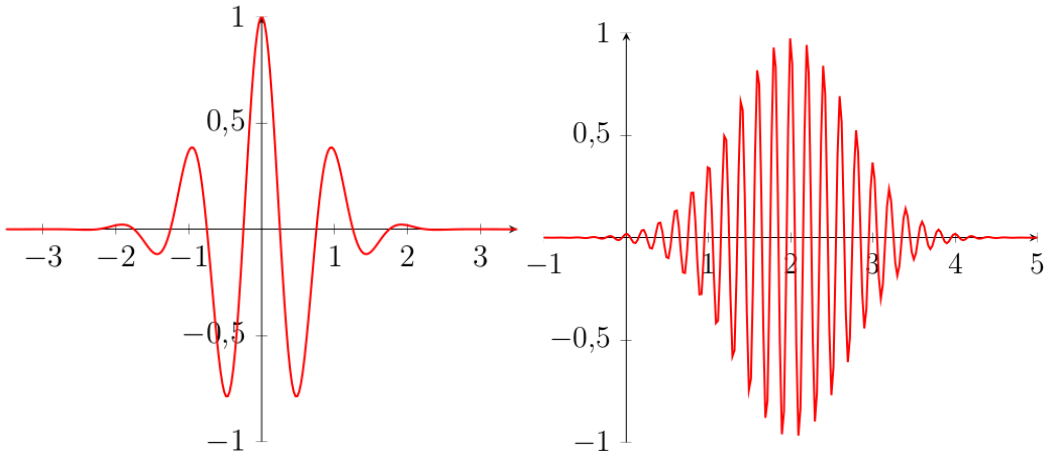
A difficulty that has often been pointed out with this approach is that, because of the infinite extent of the basis function, any time-local information (e.g., an abrupt change in the signal) is spread out over the whole frequency axis. For that reason, the above transform (1.1) is not adequate for frequency analysis localized in time. This is like knowing which notes are present in a song, but not at what moments they sound.

Dennis Gabor (1900-1979) adressed this problem by introducing windowed complex sinusoids as basis functions [17]. This leads to the doubly indexed windowed Fourier transform, or *Short-Time Fourier Transform* (STFT):

$$\mathcal{SF}(f)(\omega, \tau) = \langle f, g_{\omega,\tau} \rangle = \int_{\mathbb{R}} f(t) h(t-\tau) e^{-i\omega t}\, dt \tag{1.2}$$

where $h(\cdot)$ is an appropriate window, like a Gaussian. This can be viewed as projecting f on a basis function $g_{\omega,\tau}(t) = h(t-\tau)e^{i\omega t}$ that consist of a wave having frequency ω but localized in time around τ. Real parts of two basis elements are shown in Figure 1.1.

A limitation of the STFT is that the resolution of the analysis is the same at all locations in the time-frequency plane, because a single time window is used for all frequencies. Of course, the *Heisenberg uncertainty principle* excludes the possibility of having arbitrarily high resolution in both time and



(a) $\omega = 1, \tau = 0$ (b) $\omega = 5, \tau = 2$

Figure 1.1: Real part of $g_{\omega,\tau}(t)$ of the STFT (1.2) using window $h(t) = e^{-t^2}$.

frequency, since it lower bounds the time-bandwidth product of possible basis functions by

$$\Delta T \times \Delta\Omega \geq \frac{1}{4\pi}$$

where $(\Delta T)^2$ and $(\Delta\Omega)^2$ are the variances of the absolute values of the function and its Fourier transform respectively [17]. An illustration of this fact is shown in Figure 1.2. However, by varying the window used, one can trade resolution in time for resolution in frequency. This is what wavelets intend to do.

1.1.2 Continuous and discrete wavelet transform

Suppose we have a function $\psi \in L^2(\mathbb{R})$ (assuming sufficient regularity on ψ) called *wavelet*. Think of ψ as a short wave like in Figure 1.1 (a). For $a \in \mathbb{R}^+, b \in \mathbb{R}$ we can define

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right). \tag{1.3}$$

That is, we generate a scaled/translated version of ψ. For large a, the above function comes as a stretched version of the prototype wavelet, that is a low-frequency function, while for small a the function becomes a contracted version of ψ and thus a high-frequency function (see Figure 1.3).

With these basis functions, we can define the *continuous wavelet transform* as

$$W_c(f)(a,b) = \langle f, \psi_{a,b}\rangle = \frac{1}{\sqrt{a}}\int_{\mathbb{R}} f(t)\psi^*\left(\frac{t-b}{a}\right) dt \tag{1.4}$$

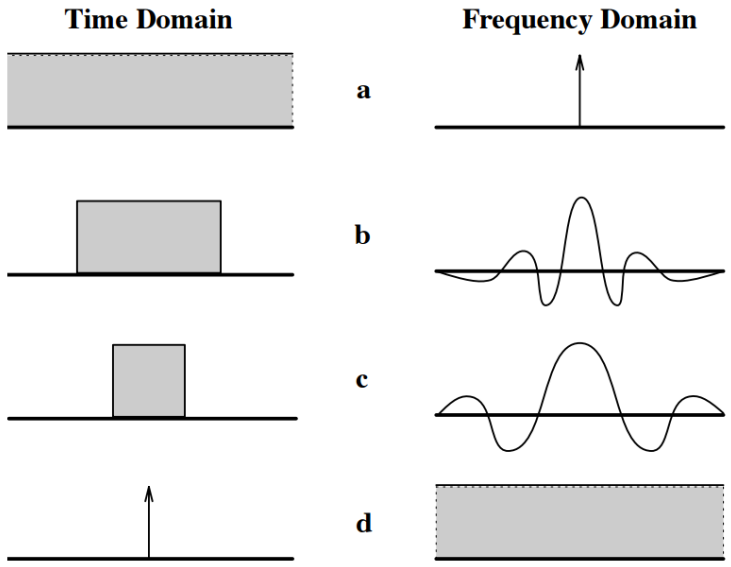


Figure 1.2: As a function support narrows in time domain (left), its frequency support widens (right).

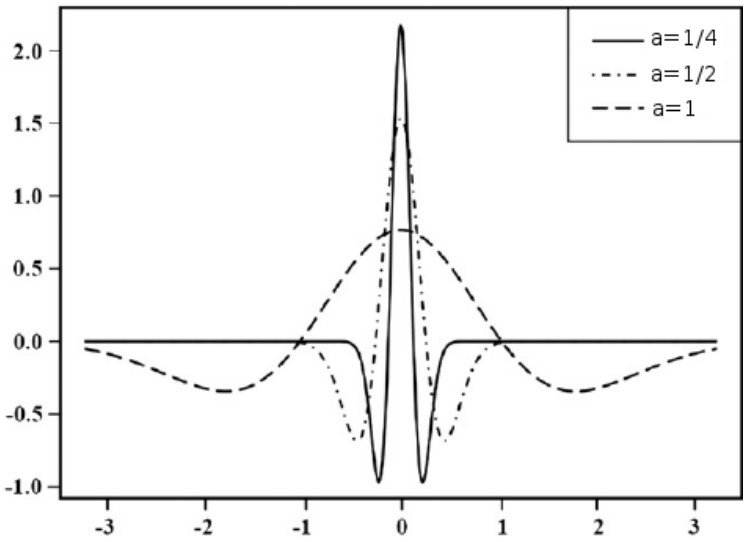


Figure 1.3: Scaled versions of a function ψ (*Mexican hat wavelet*).

where ψ^* is the complex conjugate of ψ. The time-frequency resolution of the wavelet transform (1.4) involves a different tradeoff to the one used by the STFT: the time window is narrowed when we want to detect high frequencies, and widened when we search for low frequencies.

Like the STFT, the wavelet transform is highly redundant when the parameters (a, b) are continuous (or (ω, τ) in STFT). Therefore the transforms are usually evaluated on a discrete grid on the time-frequency plane, corresponding to a discrete set of continuous basis functions. The question that arises is when there is a grid such that the set of basis functions constitutes an orthonormal basis, which in particular implies that there is no redundancy. For the STFT, there is a well known theorem that shows that there is no hope of obtaining an orthonormal Gabor basis with nice time-frequency localization, which is a classical manifestation of the uncertainty principle for Gabor systems [25], [2]:

Theorem 1.1. *Balian-Low*
Let $h \in L^2(\mathbb{R})$ and a *Gabor basis* $\mathcal{G}(h) = \{g_{m,n}\}_{m,n\in\mathbb{Z}}$ defined by

$$g_{m,n}(t) = h(t-n)e^{imt}.$$

If

$$\int_{\mathbb{R}} t^2|h(t)|^2\,dt < \infty \quad \text{and} \quad \int_{\mathbb{R}} \omega^2|\hat{h}(w)|^2\,d\omega < \infty$$

then $\mathcal{G}(h)$ is not an orthonormal basis for $L^2(\mathbb{R})$.

Let's discretize the translation and dilation parameters of the wavelet transform: if we choose $a_0 > 1, b_0 \neq 0$ then we set

$$\psi_{j,k}(t) = a_0^{-j/2}\psi(a_0^{-j}t - kb_0) \tag{1.5}$$

which corresponds to $a = a_0^j$ and $b = ka_0^j b_0$ in (1.3). Note that translation steps depend on the dilation length, since long wavelets are advanced by larger steps, and short ones by smaller steps. On this discrete grid, the *discrete wavelet transform* is thus

$$W_d(f)(j,k) = \langle f, \psi_{j,k}\rangle = a_0^{-j/2}\int_{\mathbb{R}} f(t)\psi^*(a_0^{-j}t - kb_0)\,dt. \tag{1.6}$$

Of particular interest is the discretization on a dyadic grid, which occurs for $a_0 = 2$, and $b_0 = 1$. Now, it is possible to construct wavelets ψ so that the set

$$W = \{\psi_{j,k}\}_{j,k\in\mathbb{Z}}, \qquad \psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) \tag{1.7}$$

is orthonormal, that is

$$\langle \psi_{j,k}, \psi_{j',k'}\rangle = \delta_{jj'}\delta_{kk'}$$

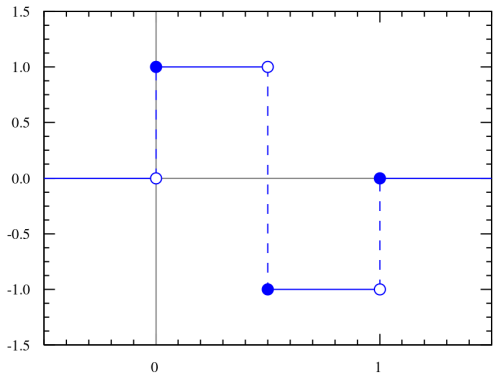


Figure 1.4: Haar wavelet defined by (1.8).

where δ_{ij} is the Kronecker delta. A classic example is the *Haar basis* generated by:

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1/2 \\ -1 & \text{if } 1/2 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

(see Figure 1.4). The orthonormality is easily verified, and the fact that it is a basis of $L^2(\mathbb{R})$ is proved in [30]. Haar [18] used these functions to give an example of an orthonormal system for the space of square-integrable functions on the unit interval $[0, 1]$, and it is now recognised as the first known wavelet basis. However, the Haar function is not continuous, and this is not generally appropriate for signal processing.

Orthonormal and biorthogonal wavelet basis

It can be shown [23] that we can construct other wavelets with more regularity (that is, continuity, smoothness, etc) than the Haar wavelet. This is done by the concept of *Multiresolution Analysis (MRA)* due to Mallat [22] and Meyer [24], that basically consists in a nested set of subspaces $\{V_j, W_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$ such that

$$\ldots \subset V_{j+1} \subset V_j \subset V_{j-1} \subset \ldots \tag{1.9}$$

$$V_j \oplus W_j = V_{j-1} \quad \forall j \in \mathbb{Z} \tag{1.10}$$

and there exists functions ϕ and ψ such that $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$ and $\{\psi_{j,k}\}_{k\in\mathbb{Z}}$ are basis of V_j and W_j respectively, where $\phi_{j,k}$ and $\psi_{j,k}$ are defined as in (1.7). Here, ψ is the wavelet. A function $f \in L^2(\mathbb{R})$ can be projected onto an approximation space V_j and then we can add details (which consist in the projection of f onto W_j) to obtain a higher resolution version in V_{j-1}. See the next subsection for a graphical example.

There are two important cases to be distinguished:

- $V_j \perp W_j$ and $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ is an *orthonormal* basis of $L^2(\mathbb{R})$, thus

$$\langle \psi_{j,k}, \psi_{j,l} \rangle = \delta_{kl} \quad \forall j,k,l \in \mathbb{Z}$$
$$f = \sum_{(j,k)\in\mathbb{Z}^2} \langle f, \psi_{j,k} \rangle \psi_{j,k} \quad \forall f \in L^2(\mathbb{R})$$

- There exists a *dual wavelet* $\tilde{\psi}$ such that $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ and $\{\tilde{\psi}_{j,k}\}_{j,k\in\mathbb{Z}}$ are *biorthogonal Riesz basis* of $L^2(\mathbb{R})$, thus

$$\langle \psi_{j,k}, \tilde{\psi}_{j,l} \rangle = \delta_{kl} \quad \forall j,k,l \in \mathbb{Z}$$
$$f = \sum_{(j,k)\in\mathbb{Z}^2} \langle f, \psi_{j,k} \rangle \tilde{\psi}_{j,k} = \sum_{(j,k)\in\mathbb{Z}^2} \langle f, \tilde{\psi}_{j,k} \rangle \psi_{j,k} \quad \forall f \in L^2(\mathbb{R})$$

We refer the reader to Appendix A.1 for more details about this construction.

1.1.3 Wavelet analysis of finite-length signals

We are mainly interested in processing *finite-length* L signals such as images. Also, we want to do it by computing at least L wavelet coefficients. Otherwise, we are increasing the amount of coefficients needed to represent the same signal, which is obviously undesirable, and this can be achieved analyzing a periodization or symmetrization of the original signal. It can be shown [36] that in this case, orthogonal wavelet filters cause border artifacts because they are non-symmetric (the only symmetric orthogonal wavelet basis is the Haar basis [12]), but we can still obtain biorthogonal wavelet basis with this desirable property. This is why biorthogonal basis are preferable to orthogonal basis when processing images, as we see below.

In the orthogonal case, one can construct discrete filters h and g related to the basis functions ϕ and ψ of the MRA such that the decompositions and reconstructions between two resolutions j and $j-1$ are performed by convolutions and downsampling/upsampling with these filters [23]. In the biorthogonal case, we have filters $\tilde{h}$ and $\tilde{g}$ related to the dual functions $\tilde{\phi}$ and $\tilde{\psi}$ that replace h and g in the reconstruction. The *Cohen-Daubechies-Feauveau* (CDF) wavelets [10] are historically the first family of biorthogonal wavelets, and the CDF 9/7 is experimentally the better suited for image analysis [37]. Although this is not an orthogonal wavelet, it is *almost orthogonal* in the sense that is almost energy preserving, so for practical purposes it can be considered as orthogonal.

Finally, we can decompose a signal by applying this process several times to the samples of the signal. A *wavelet decomposition* of a function $f \in V_L$, which is caracterized by the coefficients $a_L = \{\langle f, \phi_{L,k} \rangle\}_{k\in\mathbb{Z}}$ is composed of detail wavelet coefficients $d_j = \{\langle f, \psi_{j,k} \rangle\}_{k\in\mathbb{Z}}$ of f at scales $L < j \leq J$ plus the remaining approximation at the largest scale J:

$$[a_J, d_J, d_{J-1}, \ldots, d_{L-1}] \tag{1.11}$$

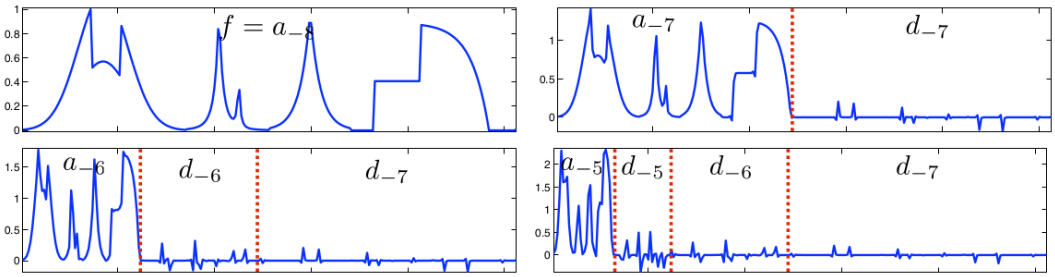


Figure 1.5: Decomposition of a signal in multiple approximation levels. The a_j and d_j are the coefficients of the expansion of f in the basis of V_j and W_j respectively (from [29]).

(recall that the resolution increases as the scale parameter j decreases). A diagram of this decomposition/reconstruction process is shown in Figure A.4. In Figure 1.5 we show an example of multi-level decomposition of a signal using the decomposition (1.11) for several values of J.

1.2 Image processing

In this section, we present the fundamental concepts of *digital image processing* and set the context for this thesis. We begin by formalizing the definition of digital image, reviewing some of the common problems concerning images, and finally we explain how images are processed efficiently using wavelets.

1.2.1 What is a digital image?

An *image* may be defined as a two-dimensional function $f(x, y)$, where x and y are *spatial (plane) coordinates*, and the amplitude of f at any pair of coordinates is called the *intensity* of the image at that point. The term *gray level* is often used to refer to the intensity of monochromatic images. Color images are formed by a combination of individual images. For example, in the RGB color system a color image consists of three individual monochrome images, referred to as the *red (R), green (G) and blue (B) primary images*. For this reason, many of the techniques developed for monochromatic images can be extended to color images by processing the three component images individually, although in most applications better results can be obtained using vectorial approaches (working on each channel separately and then composing the color image may lead to aberrant colors, for instance). In this thesis we only work with monochromatic (gray-valued) images.

Theoretically, an image may be continuous with respect to the x and y coordinates, and also in amplitude. Converting such an image to digital form requires that the coordinates, as well as the amplitude, be digitized. Digitizing the coordinate values is called *sampling*, and digitizing the amplitude values

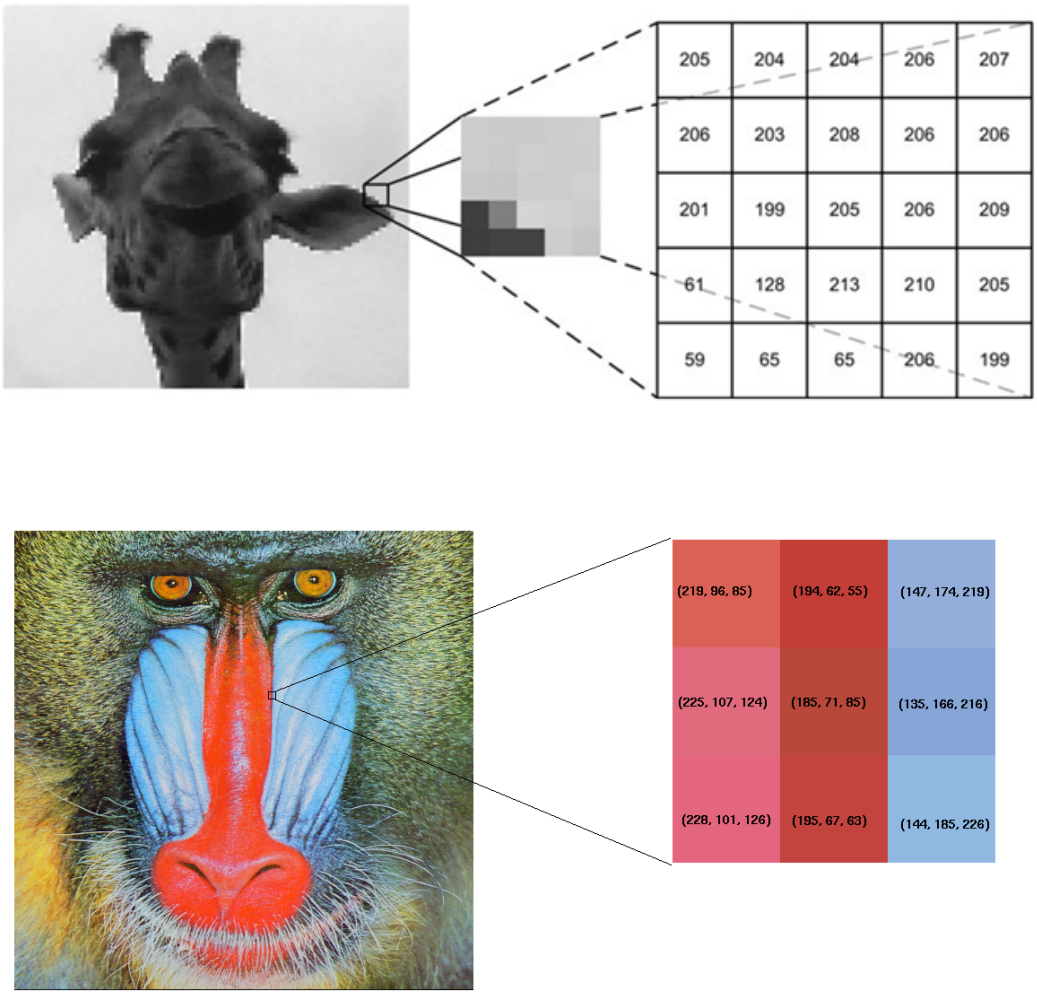


Figure 1.6: Pixel values for a monochromatic and color images.

is called *quantization*. Thus, when x,y and the amplitude values of f are all finite, discrete quantities, we call the image a *digital image*. We have a 8-bit digital image when we use 2^8 distincts integer values to represent the gray values, from 0 (black) to 255 (white) in general. This is the most common type of digital images, but we may also encounter 12-bits or 16-bits images as well. In Figure 1.6 are shown a 8-bit monochromatic (gray level) image and a 8-bit per channel color image.

The result of sampling and quantization is a matrix of integer numbers:

$$u\colon \Omega \to \mathbb{R}, \qquad u = \begin{pmatrix} f(1,1) & f(1,2) & \cdots & f(1,M) \\ f(2,1) & f(2,2) & \cdots & f(2,M) \\ \vdots & \vdots & & \vdots \\ f(N,1) & f(N,2) & \cdots & f(N,M) \end{pmatrix}$$

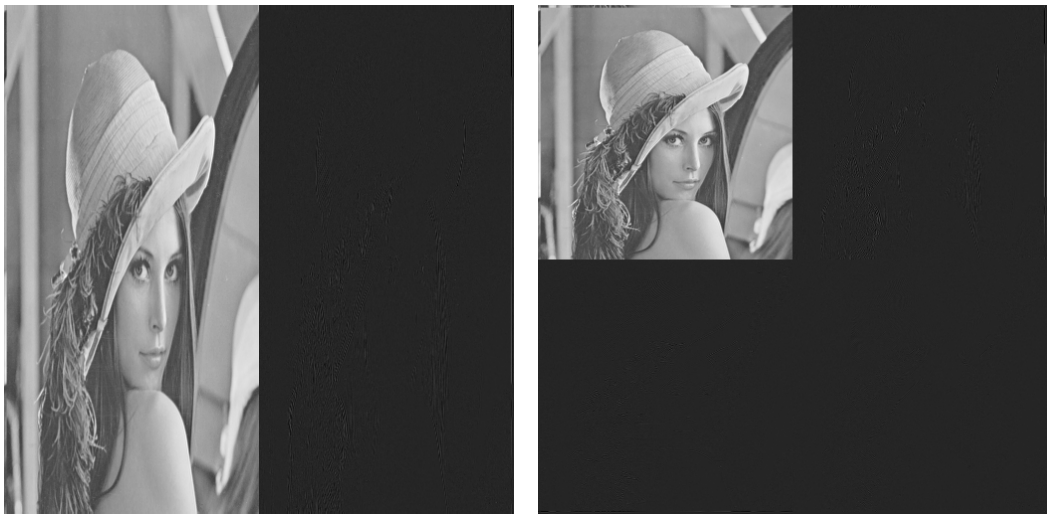


Figure 1.7: The wavelet decomposition of (1.11) is applied in Lena: first one-level decomposition applied to the rows (left) and then to the columns to obtain a one-level wavelet decomposition of the whole image (right).

where $\Omega = [1, \ldots, N] \times [1, \ldots, M]$. We refer to u as an $N \times M$ image, and to simplify notation, in what follows we denote by $x = (i, j)$ a generic pixel, where $x \in \Omega$.

Wavelet image analysis

We can apply the decomposition scheme of (1.11) presented in the previous section in a separable fashion: first we apply a one-level decomposition to each row of the image and then in the columns, obtaining a *1-level wavelet decomposition of the image*. This result consists in four *subbands* that we call *summary* (LL) and *detail* (LH, HL, HH) subbands. This procedure is shown in Figure 1.7. If we repeat the decomposition to the LL subband, we get a *multi-level wavelet decomposition* (see Figures 1.8 and 1.9).

1.2.2 Denoising

In any digital image, the measurement of the color value at each pixel is subject to some perturbations (called *noise*). These perturbations are due to the random nature of the photon counting process at each sensor. The noise can be amplified by digital corrections of the camera or by any image processing software. For example, tools removing blur from images or increasing the contrast enhance the noise. This observation motivates solving the problem of removing the noise present in a digital image, problem that is called *image denoising*.

Noise in digital images can originate from a variety of sources:

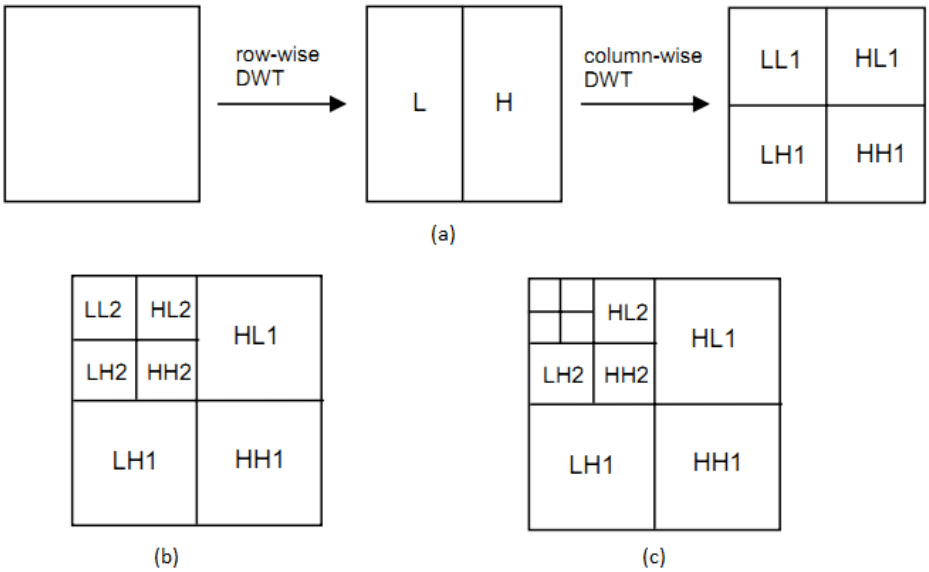


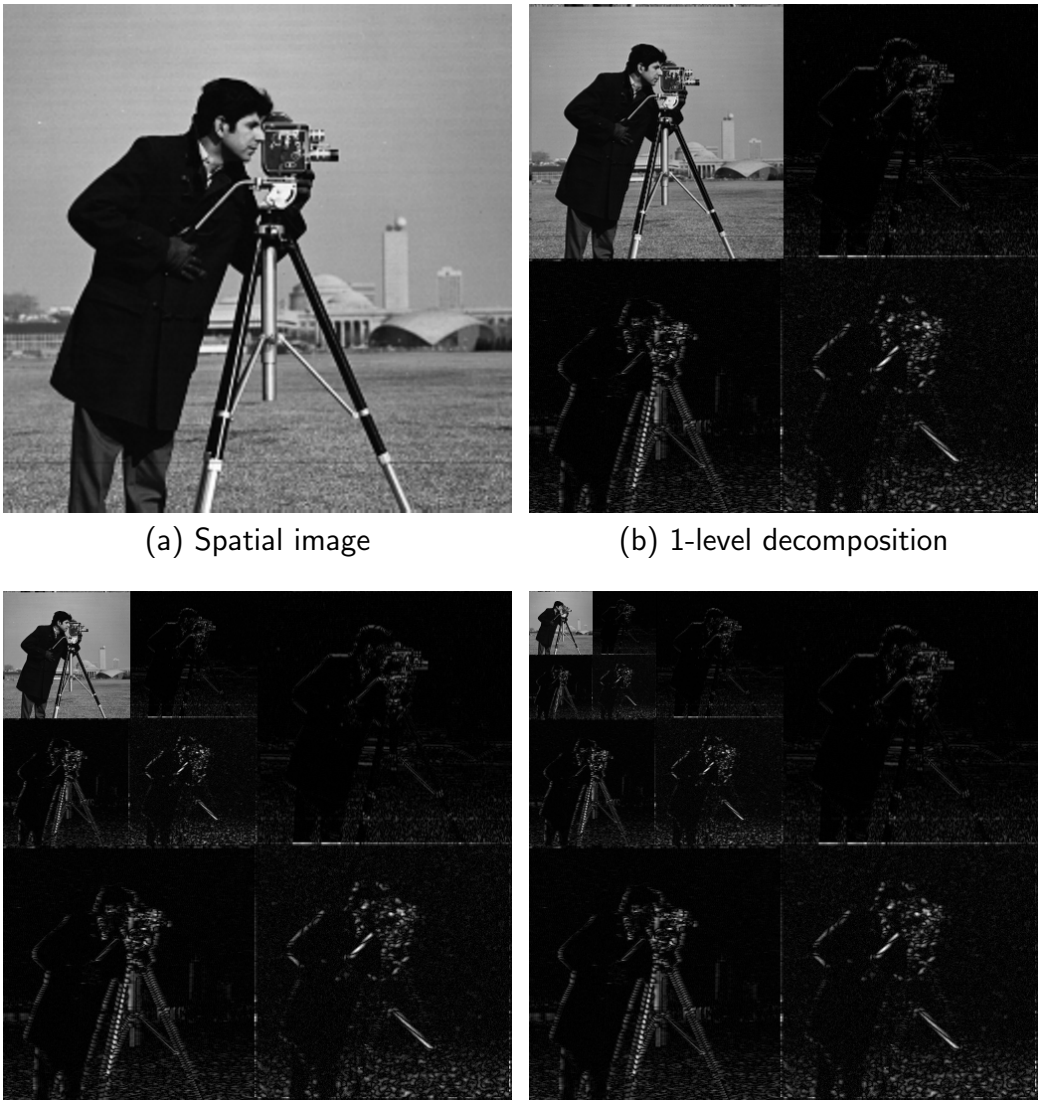
Figure 1.8: (a) Wavelet decomposition is first applied to the image rows and then to the columns to obtain a 1-level decomposition. (b) On a second step, the same procedure is applied to the LL1 subband to obtain a two-level decomposition. (c) We can repeat this procedure to obtain a multi-level decomposition of the image.

- *Capture noise* can be the result of variations in lighting, sensor temperature, electrical sensor noise, sensor nonuniformity, dust in the environment, vibration, lens distortion, focus limitations, sensor saturation (too much light), underexposure (too little light).

- *Sampling noise.* As discussed previously, limitations in sampling and intensity quantization are a source of noise in the representation of the image, as the sampled digital image in not a true representation of the analog (continuous) image.

- *Processing noise.* Limitations in numerical precision (floating-point numbers), potential integer overflow and mathematical approximations (e.g. $\pi = 3.1416$) are all potential sources of noise in the processing itself.

- *Image-encoding noise.* Many modern image compression techniques (to be seen later in this thesis) are *lossy* compression techniques. By lossy we mean that they compress the image by removing visual information that represents detail not generallya perceivable to the human viewer. The problem is that this loss of information due to compression undermines image-processing techniques that rely on this information. This loss of detail is often referred to by the appearance of *compression artefacts* in the image.



(a) Spatial image (b) 1-level decomposition

(c) 2-level decomposition (d) 3-level decomposition

Figure 1.9: Multi-level wavelet decomposition of an image (detail coefficients have been enhanced for better visualization).

(a) Original Lena image (b) Zoom of the original image

(c) Noisy image with $\sigma = 10$ (d) Zoom of noisy image

Figure 1.10: An example of the visual artifacts seen in a noisy image. The noise is white Gaussian with zero mean and variance σ^2.

(a) Noisy image with $\sigma = 20$

(b) Zoom of noisy image

(c) Noisy image with $\sigma = 40$

(d) Zoom of noisy image

Figure 1.11: An example of the visual artifacts seen in a noisy image. The noise is white Gaussian with zero mean and variance σ^2.

In practice, image denoising consists in recovering an image that has been degraded by using a priori knowledge of the degradation phenomenon. Thus, restoration techniques are oriented toward modeling the degradation and applying the inverse process in order to recover the original image, and usually involve formulating a criterion of goodness that yields an optimal estimate of the desired result.

Noise model

Let

$$u\colon \Omega \to \mathbb{R}, \qquad \Omega = [1,\dots,N]\times[1,\dots,M]$$

be an image, and

$$u_n = u + n, \qquad n(x) \sim N(0,\sigma^2)\ i.i.d.$$

the corrupted image with an additive white Gaussian noise. The *image denoising problem* consist of estimating u given u_n and σ. See Figures 1.10 and 1.11 to see the effect of adding Gaussian noise to an image.

Although other types of noise can be considered (instead of Gaussian), they are somehow related to this one (by means of the Central Limit Theorem) or reduced to it in certain circumstances by applying a *Variance Stabilizing Transform* like the *Anscombe transform* for Poisson noise [1], [35].

Measures of image quality

In order to test and compare denoising methods, there exists a number of quantitative measures of image quality. Basically, we add noise to an image, denoise it by a method, and measure the difference (error) between the restored image with the original one. The following measures are commonly used in image processing:

Let u and $\hat{u}$ be the original and restored image respectively (both with domain Ω):

- *Mean squared error (MSE)*: defined by

$$\mathrm{MSE}(u,\hat{u}) = \frac{1}{|\Omega|}\sum_{x\in\Omega}(u(x)-\hat{u}(x))^2$$

- *Peak signal-to-noise ratio (PSNR)*: measures the ratio between the maximum possible energy of a signal and the energy of corrupting noise that affects the fidelity of its representation:

$$\mathrm{PSNR}(u,\hat{u}) = 20\log_{10}\left(\frac{\mathrm{MAX}_I}{\sqrt{\mathrm{MSE}(u,\hat{u})}}\right),$$

  where MAX_I is the maximum possible pixel value of the image.

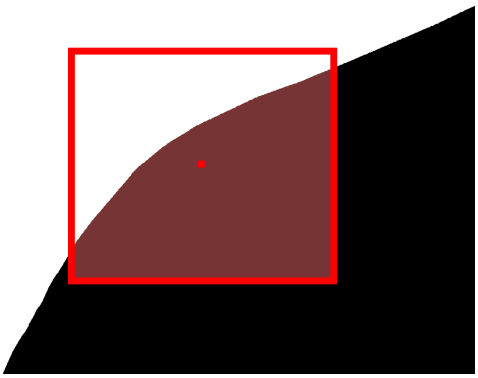


Figure 1.12: When denoising a pixel (in red) near an edge, in neighborhood filters only the closer pixels having similar gray values are averaged (showed in dark red inside the window).

The most commonly used measure is PSNR. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not. One has to be extremely careful with the range of validity of this metric, and it is common that it doesn't imply a higher visual quality in the reconstruction. Hence, subjective ratings may also be used as ground truth to develop new algorithms.

Some classical denoising methods

Here, we review some classical denoising methods, previous to the appearance of the so called *Non-local methods*. The later methods will be studied in Chapter 2. For the discussions below, we work on the continuous image u instead of the digital one, as denoising methods are more easily expressed in that domain, but in practice they must be discretized.

1) Kernel filtering

The idea of this type of method is simple: to denoise a pixel x, we average (using a kernel) the spatially closest pixels in the image, as closer pixels should have similar gray level value. They can be expressed by

$$M_K u(x) = K \star u(x) = \int_{\mathbb{R}^2} K(x-y)u(y)\,dy.$$

For example, if the kernel K is the normalized indicator function on a ball $B_h(x)$ of center x and radius h, then

$$M_h u(x) = \frac{1}{\pi h^2} \int_{B_h(x)} u(y)\,dy$$

is the uniform average of u within that ball, and h is a parameter that controls the denoising amount of the method. A common choice is the *Gaussian filtering*

$$M_{G_h}u(x) = \frac{1}{C_h}\int_{\mathbb{R}^2} e^{-\frac{\|x-y\|^2}{h^2}}u(y)\,dy,$$

where $C_h = \int_{\mathbb{R}^2} e^{-\frac{\|x-y\|^2}{h^2}}\,dy$ is the normalizing constant.

The assumption that closer pixels should have similar gray values is only valid for homogeneous regions and therefor edges and textures are blurred (see Figure 1.13 (c)).

2) Neighborhood filtering

We can better select the pixels to be averaged if we also take into account the gray values when comparing closer pixels. Thus, in an edge, only pixels that belong to the same level line of x will be averaged (see Figure 1.12). This can be accomplished by the *Yaroslavsky neighborhood filter*:

$$\mathrm{YNF}_{h,\rho}u(x) = \frac{1}{C(x)}\int_{B_\rho(x)} e^{-\frac{|u(y)-u(x)|^2}{h^2}}u(y)\,dy$$

where $C(x)$ is a normalizing factor, $B_\rho(x)$ is a ball of center x and radius ρ and h is the filtering parameter. The result is shown in Figure 1.13 (d).

3) Wavelet shrinkage denoising

This method is very important in this study because it is the most related denoising method to our restoration problem, as it will explained in short.

Let $\{w_i\colon i \in J\}$ be a wavelet (analysis) basis and $\{\tilde{w}_i\colon i \in J\}$ its dual (synthesis) basis. That is, for an image u we have the following decomposition:

$$u = \sum_{i\in J}\langle u, w_i\rangle\tilde{w}_i.$$

Following the wavelet shrinkage method of Donoho and Johnstone [14], we have a denoised version u_d of u setting to zero all wavelet coefficients with respect to a reasonable threshold:

$$u_d = \sum_{i\in J}\tau_\lambda(\langle u, w_i\rangle)\tilde{w}_i \quad \text{where } \tau_\lambda(x) = \begin{cases} 0 & \text{if } |x| \le \lambda \\ x & \text{otherwise} \end{cases}.$$

The above shrunk wavelet representation involves both:

- Large coefficients corresponding to noise (*outliers*) that are kept and generate artifacts with the shape of the functions $\tilde{w}_i$.

- Some coefficients, erroneously set to zero, leading to *pseudo-Gibbs oscillations* and *erased microtextures* related to the Gibbs phenomena of Fourier series truncation (see Figures 1.14 and 1.15).

In Figure 1.13 (e) we can see this described artifacts. We will talk more about outliers in Section 1.3.

State-of-the-art image denoising methods are based in restoration using image *patches*, which are little portions of the image. At the end of this chapter we will present the description of our main problem and in Chapter 2 we will study the *Non-local Bayes* method which is nowadays one of the best image denoising algorithm based on this patch-based approach. A result of this denoising method is shown un Figure 1.13 (f).

It is worth to mention the Image Processing On Line journal (IPOL)[1]. IPOL is a research journal of image processing and image analysis which emphasizes the role of mathematics as a source for algorithm design and the reproducibility of the research. Each article contains a text on an algorithm and its source code, with an online demonstration facility and an archive of experiments. Text and source code are peer-reviewed and the demonstration is controlled. IPOL is an Open Science and Reproducible Research journal, in which we can find the implementations of some of the best image denoising algorithms up to date, as well as algorithm for other image processing tasks.

1.2.3 Compression

Various space agencies from around the world, including the European Space Agency (ESA), the National Aeronautics and Space Administration (NASA), the Canadian Space Agency (CSA), and the Japan Aerospace Exploration Agency (JAXA), are collaborating on a program to monitor global change that will generate half a terabyte of data per day when it is fully operational. New sequencing technology is resulting in ever-increasing database sizes containing genomic information while new medical scanning technologies could result in the generation of petabytes of data. Although new storage and transmission technologies are constantly developed, the data volume growth is in general faster.

In a daily life example, most common digital images uses 1 byte (8-bits) per color channel per pixel, that is, 3 bytes per pixel. This allows to have $2^3 = 256$ differents tones per channel, totaling $256^3 > 16$ millons of differents colors that we can use to represent a digital image. Thus, an 8 Mpx (*megapixels*) image, taken by a common smartphone camera to date, will occupy $8 \times 10^6 \times 3$ bytes $\simeq$ 22 megabytes (22 MB) if we save all the pixels information of the image. But in practice, we can see that our smartphone

[1] `http://www.ipol.im/`

(a) Original image

(b) Noisy ($\sigma = 10$) - $PSNR = 28.14$

(c) Gaussian filtering - $PSNR = 23.67$

(d) Yaroslavsky filter - $PSNR = 30.49$

(e) Wavelet shrinkage - $PSNR = 29.58$

(f) Non-local Bayes - $PSNR = 34.75$

Figure 1.13: Comparision between four image denoising methods.

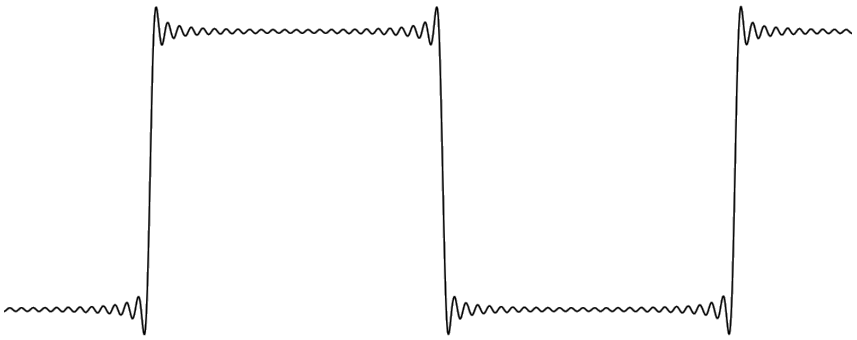


Figure 1.14: Gibbs phenomena near discontinuities due to Fourier series truncation

saves pictures of this size using about 10% to 15% of this space.

This can be achieved by applying a *compression* technique, that consists in storing the same image (or a very similar one) in much less space than it would take if we saved the data as it is, which will lead to less space for data storage and faster transmission. There are two types of compression:

- *Lossless*: the compression is done in a way that we can recover the *exact* original data (no information is lost).
- *Lossy*: we keep only the most important part of the data (in some sense) such that the reconstructed data looks like the original, hence we do not recover the exact image but an *approximation* of it.

Lossless compression

Lossless compression is a class of data compression algorithms that allows the original data to be perfectly reconstructed from the compressed data. By contrast, lossy compression permits reconstruction only of an approximation of the original data, though this usually improves compression rates (and therefore reduces file sizes). Lossless compression is used in cases where it is important that the original and the decompressed data be identical, or where deviations from the original data could be deleterious. Typical examples are executable programs, text documents, and source code. Some image file formats, like PNG or GIF, use only lossless compression, while others like TIFF and JPEG may use either lossless or lossy methods.

One of the main types of lossless compression creates and assigns a unique prefix-free code (there is no whole code word in the system that is a prefix (initial segment) of any other code word) to each unique symbol that occurs in

(a) Original image and the result of thresholding

(b) Zoom of the above images

Figure 1.15: Artifacts introduced by setting to zero small wavelet coefficients (thresholding). Note the presence of pseudo-Gibbs oscillations near sharp edges, erased microtextures and outliers. In the zoom images the contrast was adjusted in order to enhance the artifacts for better visualization.

the input. These entropy encoders then compress data by replacing each fixed-length input symbol with the corresponding variable-length prefix-free output codeword. The length of each codeword is approximately proportional to the negative logarithm of the probability or frequency of occurrence. Therefore, the most common symbols use the shortest codes. A well known prefix code is the Huffman coding scheme [19]. As lossless compression is a reversible process, we do not care about it in this thesis.

Lossy compression

Lossy compression is based on an irreversible process called *quantization*. We can model a generic quantizer with a function that maps the set of values that a coefficient can take (in most cases, the real line $\mathbb{R}$), in a *finite* set of values:

$$Q\colon \mathbb{R} \to C_Q, \qquad C_Q = \{c_1, c_2, \ldots, c_m\}. \tag{1.12}$$

In general, the quantizer maps a whole interval into a unique integer value that represents this interval (see Figure 1.16). Althrough in practice one always has a *floating point* representation ("float") of the coefficients that is already quantized (that is, we do not really have continuous but discrete numerical values), these values need 32 or 64 bits to be represented, and one wants to represent them with fewer bits. For example, the *8-bit floor operator* that maps a real (in practice, float) value x to a 8-bit approximation can be defined as

$$Q(x) = \max(0, \min(255, \lfloor x \rfloor)) = \begin{cases} 0 & \text{if } x \in (-\infty, 1) \\ n & \text{if } x \in [n, n+1) \text{ for } n = 1, 2, \ldots, 254 \\ 255 & \text{if } x \in [255, +\infty) \end{cases} \tag{1.13}$$

where $\lfloor \cdot \rfloor$ stands for the classical *floor* operator (that is, $\lfloor x \rfloor$ is the largest integer less than or equal to x). This quantizer is very suitable for quantizing images before we can store them in a *8-bits per pixel (bpp)* format.

The quantization is an irreversible process because it maps different values x to the same value $Q(x)$, so given $Q(x)$ is not possible to recover x exactly. For this reason, the quantization is a lossy compression scheme. Ideally, we want to discard only the less relevant information of the signal. For example, in a acoustic signal, there are frequencies less audible to the human ear that we do not want to keep, or in an image, small variations in homogeneous areas are less important that sharp edges. Also, if we can do this using only few possible values c_i, and set most of the signal coefficients to the same value (for example, c_0) then we can get high compression ratios by means of a lossless compression scheme such as Huffman coding.

Most commonly used image compression algorithms uses schemes similar to the one shown in Figure 1.17. The best known image compression format is

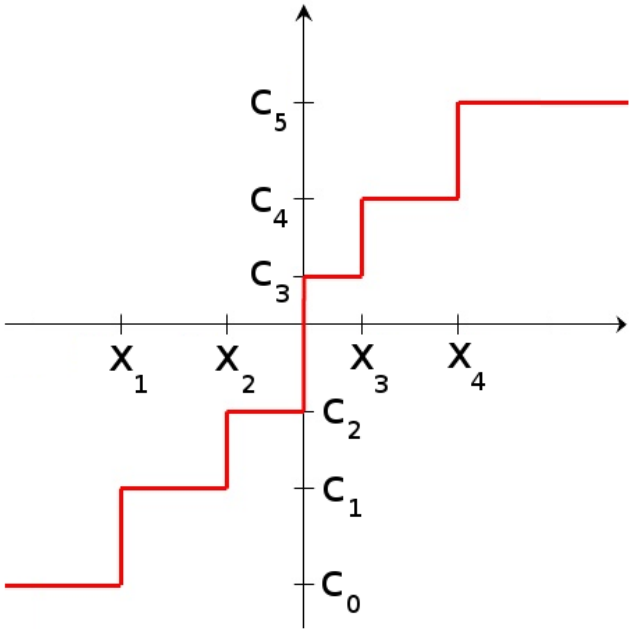


Figure 1.16: Graph of a generic quantizer of the form (1.12).

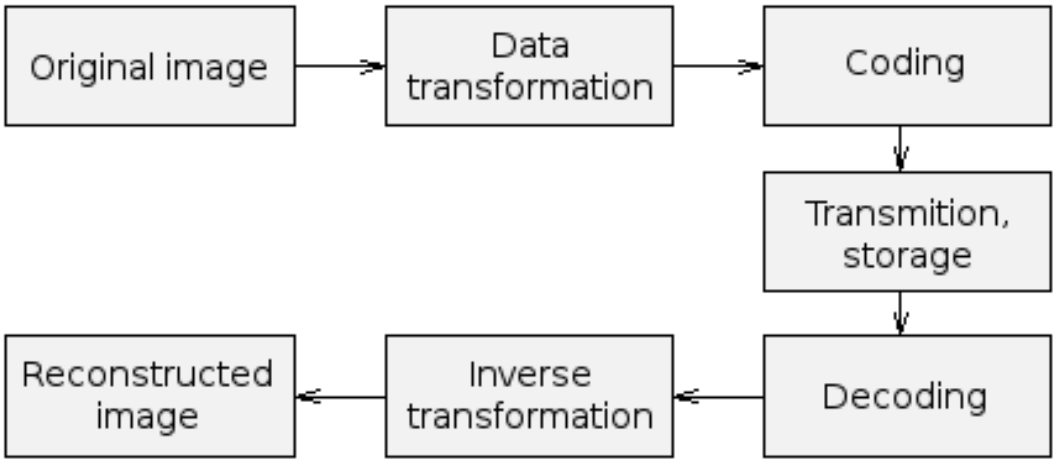


Figure 1.17: Compression/Decompression scheme.

the *JPEG* format[2]. JPEG uses a lossy form of compression based on the discrete cosine transform (DCT), which is related to the Fourier transform. This transform converts each image from the spatial domain into the frequency domain. A perceptual model based loosely on the human psychovisual system discards high-frequency information, i.e. sharp transitions in intensity, and color hue. Most of the high-frequency coefficients, which contribute less to the overall picture than other coefficients, are characteristically small-values that are quantized to zero, achieving high compression ratios.

JPEG2000 is another standard of image compression. JPEG 2000 is an image coding system that uses state-of-the-art compression techniques based on wavelet technology. Its architecture lends itself to a wide range of uses from portable digital cameras through to advanced pre-press, medical imaging and other key sectors. It is similar to the above JPEG standard in the sense that it transforms the image using a transform (this time, a wavelet transform) and quantizes the small coefficients to zero, in general in the high frequency subbands. We will describe this compression scheme in more detail in Chapter 3.

For a complete treatment of compression schemes, see [33], [34].

1.3 Problem statement

To conclude this introduction, we set our main problem, that consists in restoring compressed high-resolution satellital images, which have artifacts caused by the conjunction between adquisition noise and quantization.

CCSDS Recommendation

The *Consultative Committe for Space Data Systems (CCSDS)*[3] was founded in 1982 by the major space agencies of the world. It is a multi-national forum for the development of communications and data systems standards for spaceflight. Today, leading space communications experts from 26 nations collaborate in developing the most well-engineered space communications and data handling standards in the world. The goal is to enhance governmental and commercial interoperability and cross-support, while also reducing risk, development time and project costs. More than 800 space missions have chosen to fly with CCSDS-developed standards, and the number continues to grow.

The CCSDS has issued various recommended standards for image data compression [16], which have been approved for publication by the Manage-

[2] "JPEG" stands for Joint Photographic Experts Group, the name of the committee that created the JPEG standard and also other still picture coding standards. `https://jpeg.org/`

[3] `https://public.ccsds.org/`

ment Council of the CCSDS and represents the consensus technical agreement of the participating CCSDS member agencies. Some of these agencies are:

- Agenzia Spaziale Italiana (ASI)/Italy
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (Roskosmos)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.

It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard. We will review later more details of this standard in Chapter 3, but in order to show some artifacts that arises when using this recommendation, for the moment we present the basic setting of the compression scheme that is considered in this standard.

Let u be an image of size $N \times M$ that we want to capture. Basically, the CCSDS standard compression scheme is based on the quantization of wavelets coefficients. That is, the image that is given to us is

$$u_{qn} = W^{-1} \underbrace{Q(W(u+n))}_{w_{qn}} \tag{1.14}$$

where $n(x) \sim N(0, \sigma^2)$ is a Gaussian noise independent in each pixel, W is a wavelet (invertible) transform and $Q = (Q_k)_{k\in\Omega}$ is the quantizer that gives, for each number x in the range of possible kth-wavelet coefficients, a reference coefficient $Q_k(x)$ of the interval $[a, b]$ in which x lies. In other words, if each noisy coefficient $w_n(k)$ lies in an interval $[a_k, b_k]$ defined by the quantizer Q_k, then the associated interval of $w_{qn}(k)$ is

$$Q_k^{-1}(w_{qn}(k)) = [a_k, b_k]$$

(note that the quantizer Q_k may be different for each $k \in \Omega$). In the following, we denote by a and b the vectors of $\mathbb{R}^\Omega$ of lower and upper bounds of the

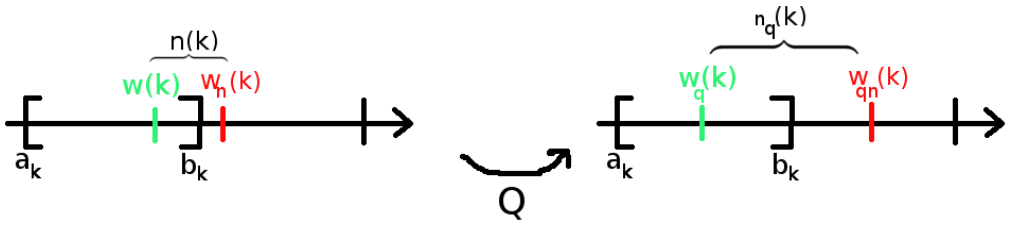


Figure 1.18: When the noise $n(k)$ added to a coefficient $w(k)$ make it belongs to another quantization interval of the original one, we call it an *outlier*.

intervals associated to the data w_{qn}.

A high compression ratio can be achieved by setting a large number of the smallest wavelet coefficients to zero. The coefficients erroneously treated by the compressor causes basically two types of artifacts:

1. Small coefficients which correspond to edges and other details (microtextures) in the image that are difficult to distinguish from the noise, and generate pseudo-Gibbs oscillations and erased microtextures.

2. Coefficients which are highly contaminated by noise (*outliers*) which modify the correct interval of quantization and generate an artifact with the shape of a wavelet basis function.

The first problem was already discussed when we review the wavelet shrinkage denoising method. As in Fourier series truncation, in our case there are coefficients that are needed to represent correctly the behavior of sharp edges, and to set them to zero causes Gibbs-phenomena related artifacts.

To understand the artifacts caused by the second problem, consider the situation shown in Figure 1.18. If we add noise to a wavelet coefficient

$$w_n(k) = w(k) + n(k)$$

and this noise causes that $w_n(k)$ belongs to a different quantization interval than $w(k)$, we call $w_n(k)$ an *outlier*. The result is that the quantization of these two coefficients will be different:

$$w_{qn}(k) = Q(w_n(k)) = Q(w(k)) + n_q(k) = w_q(k) + n_q(k)$$

where $w_q(k)$ is the right quantization of the noiseless coefficient $w(k)$ and $n_q(k)$ is the noise introduced by the outlier. The effect that this error causes when we apply the inverse wavelet transform is

$$W^{-1}(w_{qn}(k)) = W^{-1}(w_q(k)) + W^{-1}(n_q(k)).$$

The right side in this last equation consists of the right transformation of $w_q(k)$ to the spatial domain plus another term that will appear in the image with

the shape of the wavelet element corresponding to coefficient k. In Figure 1.19 we show an example of this type of artifacts.

Note that the quantization is done in the wavelet domain. The errors introduced by the quantization are almost decorrelated in wavelet domain but highly correlated in spatial domain because the outliers. So it is more natural to work in wavelet domain because it is there when we can formulate a adequate noise model for later denoise and decompress the image, as we will do it in the next chapters of this thesis.

Manuscript organization

This thesis is organized as follows: In Chapter 2 we review some of the state-of-the-arts image denoising algorithms (Non-Local Means and Non-Local Bayes), which are patch-based methods, to then adapt these techniques to do denoising in the wavelet domain. In Chapter 3 we present the details of the CCSDS Recommendation standard for image compression (and also for the JPEG2000 format) and then we propose a method to restore the quantized coefficients integrating this degradation process to the denoising algorithm, that is by *jointly* denoising/decompressing the image. In Chapter 4 we present several examples that illustrate the results obtained with our new algorithm and we compare with other methods to show the improvement of our approach in restore some details in the compressed image. Finally, in Chapter 5 we present some conclusions of our work and discuss future work that will be adressed shortly.

Figure 1.19: Original image (left) and a noisy/quantized image (right) with $\sigma = 10$ and a 0.30 bpp compresion bit rate. Note the presence of pseudo-Gibbs effects, the loss of microtextures and some outliers.

Chapter 2

Image denoising

The latest methods in image denoising have attempted to take advantage of the self-similar structures present in most images in the spatial domain, and are based in processing groups of patches (square regions of the image). In this chapter we review the principal methods of this type, the state-of-the-art in image denoising.

2.1 Patch-based denoising

First, we begin defining a few concepts that we will use hereafter:

In this thesis we will work with patches. A *patch* P of the image

$$u\colon \Omega \to \mathbb{R}, \qquad \Omega = [1,\dots,N]\times[1,\dots,M]$$

is a region of the image u with support Ω_P:

$$P\colon \Omega_P \to \mathbb{R}, \qquad \Omega_P \subseteq \Omega.$$

To fix ideas let us consider *square patches* of size $t\times t$ by setting

$$\Omega_P = [i,\dots,i+t-1]\times[j,\dots,j+t-1]$$

(but further on we shall consider other possible shapes for patches). This is a square region which upper-left corner has coordinates (i,j). We can define the $t\times t$*-patch space* of u as

$$\mathcal{P}_t(u) = \{P\colon P \text{ is a square patch of } u \text{ of size } t\times t\} \subseteq \mathbb{R}^{t\times t}.$$

A measure of the similarity between two patches $P_1, P_2 \in \mathcal{P}_t(u)$ is their euclidean distance: viewing them as matrices $P_1, P_2 \in \mathbb{R}^{t\times t}$ we can set

$$d(P_1,P_2) = \sqrt{\sum_{x\in S_t}(P_1(x)-P_2(x))^2} = \|P_1-P_2\|_F \tag{2.1}$$

Figure 2.1: Self-similarities in a natural image (from Buades' course slides)

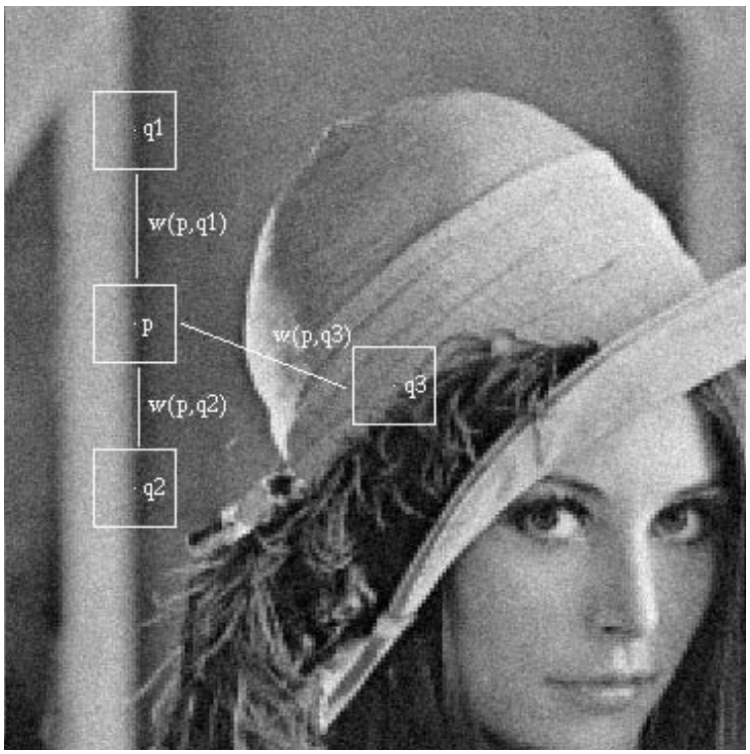


Figure 2.2: Similarities through distances in the patch space (from [7]).

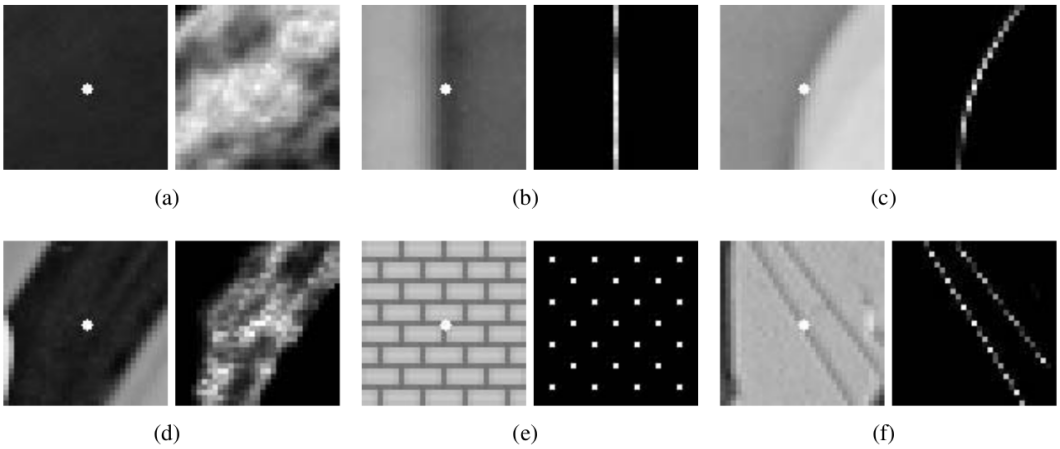


Figure 2.3: Distribution of the similar patches for the central one (from [7]).

where $S_t = [1, \dots, t]^2$ and $\|A\|_F$ is the Frobenius norm of a matrix $A \in \mathbb{R}^{t \times t}$. For example, in Figure 2.2 we can see that, although the centers have similar gray values, the patches Q_1 and Q_2 are very similar to P but Q_3 isn't.

In order to better estimate the noiseless image u from u_n, the modern techniques are based in computing estimators of the original patches P of u instead of single pixel estimators, because patches are more robust to noise that singletons.

We present the main ideas of this family of methods and later adapt one of them (the Non-Local Bayes approach) to our restoration problem.

2.1.1 Non-Local Means

Let P be a noiseless patch with dimension $t \times t$ of an image u, and $\tilde{P}$ the corresponding patch of the noisy image u_n, corrupted with additive white Gaussian noise of variance σ^2 so that

$$\tilde{P} = P + n.$$

The *Non-Local Means* method (Buades et al. [6], [8]) seems to be one of the first denosing methods exploiting self-similarities in an image, and it is inspired in Efros and Leung texture synthesis [15]. It proposes to compute, given a reference patch to be denoised, a weighted average of patches using the distance of them to the reference patch as weights.

A problem when working with u_n (the noisy image) instead of u in all patch-based algoritms is that the distance of two noisy patches $\tilde{P}, \tilde{Q} \in \mathcal{P}_t(u_n)$ is biased by the noise. Indeed, the expectation of this distance is

$$\mathbb{E}[d^2(\tilde{P}, \tilde{Q})] = \mathbb{E}[\|(P + n_P) - (Q + n_Q)\|^2] = \mathbb{E}[d^2(P, Q)] + 2\sigma^2.$$

We can define a new metric that takes into account this bias, by setting

$$d_\sigma^2(\tilde{P},\tilde{Q}) = \max\{d^2(\tilde{P},\tilde{Q}) - 2\sigma^2, 0\}.$$

This is not a distance in the strict sense, but it measures similarities in the noisy patch space up to noise. Hence, if two noisy patches satisfy $d^2(\tilde{P},\tilde{Q}) < 2\sigma^2$, then we can consider that their differences are due to noise. Thus, if $\tilde{P}$ is a noisy patch of size $t \times t$ to be restored, we can set

$$\hat{P} = \frac{1}{C(\tilde{P})} \sum_{\tilde{Q}\in R_P} p(\tilde{P},\tilde{Q})\tilde{Q} \tag{2.2}$$

as the estimated noise-free patch P of u, where

- $R_{\tilde{P}}$ is a *search region* of the image that depends on $\tilde{P}$
- $p(\tilde{P},\tilde{Q}) = e^{-d_\sigma^2(\tilde{P},\tilde{Q})/h^2}$ is the weight associated to a noisy patch $\tilde{Q} \in R_{\tilde{P}}$
- h is a *filtering parameter* that controls the denoising power of the algorithm
- $C(\tilde{P})$ is a normalization constant: $C(\tilde{P}) = \sum_{\tilde{Q}\in R_P} p(\tilde{P},\tilde{Q})$

A priori, we can use $R_{\tilde{P}} = \mathcal{P}_t(u)$ (the whole patch space, that is, the whole image) as the search region (which justifies the name *non-local*) but in practice a smaller region is used for efficiency reasons (a square region centered in $\tilde{P}$ of size $T\times T$ with $T > t$). We will discuss later more about the search region.

For patches that are common enough in the picture (as homogeneous areas like the sky in Figure 2.1 or even like the patches on the edge of the roof), sufficient patches $\tilde{Q}_i$ very similar to $\tilde{P}$ can be found, for which the mainly differences between them are the noise, that is $d_\sigma^2(\tilde{P},\tilde{Q}_i) \simeq 0$ and then $p(\tilde{P},\tilde{Q}_i) \simeq 1$. In other words, we can think of $\tilde{Q}_i$ as samples of a random Gaussian vector of mean P, and hence (2.2) becomes almost a mean computation:

$$\hat{P} = \frac{1}{k}\sum_{i=1}^{k} \tilde{Q}_i, \tag{2.3}$$

which is a good estimator of P. Note that, in addition, the weights used in (2.2) do not give too much priority to the reference patch $\tilde{P}$ which is the only one that satisfies $d^2(\tilde{P},\tilde{Q}) = 0$ with the Euclidean distance.

For rare patches (for example a patch that contains the face of the sculpture in Figure 2.1), there are no similar patches and hence the estimation (2.2) leads to a poor restoration of $\tilde{P}$. See Figures 2.4 and 2.5 for visual comparison.

Figure 2.4: Results of NLMeans. From top to bottom: Original image, Noisy with $\sigma = 10$, Denoised with NLMeans. Left: Full image, Right: Zoom of the image on the left.

Figure 2.5: Results of NLMeans. From top to bottom: Original image, Noisy with $\sigma = 20$, Denoised with NLMeans. Left: Full image, Right: Zoom of the image on the left.

2.1.2 Non-Local Bayes

A more recent NLMeans variant is *Non-Local Bayes* (Lebrun et al. [21]) that replaces the mean of similar patches by a Bayesian method to estimate the original (noiseless) patch. Because our proposed method is mainly based on NLBayes, in this section we review the main ideas behind NLBayes denoising, which consists of a *maximum-a-posteriori* (MAP) estimation.

First step

The conditional probability density function (pdf) is

$$\mathbb{P}(\tilde{P} \mid P) = \frac{1}{\sqrt{(2\pi\sigma^2)^N}} \exp\left(-\frac{\|P-\tilde{P}\|^2}{2\sigma^2}\right)$$

which is the product of the pdf's of the iid normal random variables $n_i \sim N(0,\sigma^2)$ of the noise. Here, $\|P\|$ is the Euclidean norm of P viewed as a column vector of length $N = t^2$. In practice, we know $\tilde{P}$, so our goal is to infer P by maximizing its posterior probability $\mathbb{P}(P \mid \tilde{P})$. Using Bayes' rule, we can express this conditional probability as

$$\mathbb{P}(P \mid \tilde{P}) = \frac{\mathbb{P}(\tilde{P} \mid P)\mathbb{P}(P)}{\mathbb{P}(\tilde{P})}. \tag{2.4}$$

As $\tilde{P}$ is being observed, this formula can be used to infer the original patch P by maximizing the right term, viewed as a function of P.

In order to do so, assume that we form clusters of similar patches of the image (like the colored ones shown in Figure 2.1), and that each element in the cluster that contains P is a random Gaussian sample of mean $\overline{P}$ and covariance matrix Σ_P. That is, each patch Q similar to P verifies

$$\mathbb{P}(Q) = \frac{1}{\sqrt{(2\pi)^N |\det \Sigma_P|}} \exp\left(-\frac{(Q-\overline{P})^T \Sigma_P^{-1} (Q-\overline{P})}{2}\right). \tag{2.5}$$

Noting that the denominator in (2.4) does not depend on P, we have

$$\begin{aligned}\arg\max_P \mathbb{P}(P \mid \tilde{P}) &= \arg\max_P \ \mathbb{P}(\tilde{P} \mid P)\mathbb{P}(P) \\ &= \arg\max_P \ C_1 \exp\left(-\frac{\|P-\tilde{P}\|^2}{2\sigma^2}\right) C_2 \exp\left(-\frac{(P-\overline{P})^T \Sigma_p^{-1} (P-\overline{P})}{2}\right)\end{aligned}$$

where

$$C_1 = \frac{1}{\sqrt{(2\pi\sigma^2)^N}}, \qquad C_2 = \frac{1}{\sqrt{(2\pi)^N |\det \Sigma_P|}} \tag{2.6}$$

are normalization constants that do not depend on the minimization variable. We can remove them for now, but C_2 will be useful later on when we combine

estimations belonging to differents clusters of patches.

Then, taking the $-\log$ version of the above problem, we get the noise-free patch estimation

$$\hat{P} = \arg\min_X \frac{\|X - \tilde{P}\|^2}{\sigma^2} + (X - \overline{P})^T \Sigma_P^{-1} (X - \overline{P}). \tag{2.7}$$

The function to be minimized in (2.7) can be interpreted as follows:

- The first term $\frac{\|X-\tilde{P}\|^2}{\sigma^2}$ is called the *data fitting term*, because it requires the restored patch to be similar to the noisy one. Its weight $1/\sigma^2$ compensates the bias introduced by the noise in the computation of the distance between patches: if the noise variance σ^2 is large, this term is not too heavy and allows to modify the noisy patch a lot, but if σ^2 is small, this term penalizes large modifications of $\tilde{P}$ and only allows small variations in the restoration.

- The second term $(X-\overline{P})^T\Sigma_P^{-1}(X-\overline{P})$ is a *regularization term*: it adjusts the restored patch to the Gaussian model associated to the cluster of P. Indeed, this term is the squared Mahalanobis distance of X to the mean $\overline{P}$ (which will be similar to the result of NLMeans) and takes into account the variability of the Gaussian model of the cluster, as it is captured by the covariance matrix Σ_P.

Parameters estimations

In practice, we do not know the mean and covariance matrix of the real cluster, hence we need to estimate them from the noisy patches $\tilde{Q}$ which are similar to $\tilde{P}$ (supposing that they are the noisy versions of the patches of the cluster of P). Thus we use the estimates

$$\overline{\tilde{P}} = \frac{1}{k}\sum_{i=1}^{k} \tilde{P}_i, \qquad \Sigma_{\tilde{P}} = \frac{1}{k-1}\sum_{i=1}^{k} (\tilde{P}_i - \overline{\tilde{P}})(\tilde{P}_i - \overline{\tilde{P}})^T. \tag{2.8}$$

Taking into account that the noise is white with variance σ^2, the above estimators verify:

$$\mathbb{E}[\overline{\tilde{P}}] = \overline{P}, \qquad \mathbb{E}[\Sigma_{\tilde{P}}] = \Sigma_P + \sigma^2 I \quad \Rightarrow \quad \mathbb{E}[\Sigma_{\tilde{P}} - \sigma^2 I] = \Sigma_P.$$

Replacing these estimates in (2.7) we get:

$$\hat{P} = \arg\min_X \frac{\|X - \tilde{P}\|^2}{\sigma^2} + (X - \overline{\tilde{P}})^T (\Sigma_{\tilde{P}} - \sigma^2 I)^{-1} (X - \overline{\tilde{P}}). \tag{2.9}$$

This is a quadratic problem that can be resolved analytically:

Proposition 2.1. The patch that minimizes the problem (2.9) is

$$\hat{P} = \overline{\tilde{P}} + (\Sigma_{\tilde{P}} - \sigma^2 I)\Sigma_{\tilde{P}}^{-1}(\tilde{P} - \overline{\tilde{P}}). \tag{2.10}$$

Proof. Differentiating the quadratic function (2.9) with respect to X and equating to zero yields

$$X - \tilde{P} + \sigma^2(\Sigma_{\tilde{P}} - \sigma^2 I)^{-1}(X - \overline{\tilde{P}}) = 0.$$

Taking into account that $I + \sigma^2(\Sigma_{\tilde{P}} - \sigma^2 I)^{-1} = (\Sigma_{\tilde{P}} - \sigma^2 I)^{-1}\Sigma_{\tilde{P}}$ yields

$$(\Sigma_{\tilde{P}} - \sigma^2 I)^{-1}\Sigma_{\tilde{P}} X = \tilde{P} + \sigma^2(\Sigma_{\tilde{P}} - \sigma^2 I)^{-1}\overline{\tilde{P}}$$

and therefore

$$\begin{aligned} X &= \Sigma_{\tilde{P}}^{-1}(\Sigma_{\tilde{P}} - \sigma^2 I)\tilde{P} + \sigma^2\Sigma_{\tilde{P}}^{-1}\overline{\tilde{P}} \\ &= \overline{\tilde{P}} + (\Sigma_{\tilde{P}} - \sigma^2 I)\Sigma_{\tilde{P}}^{-1}(\tilde{P} - \overline{\tilde{P}}). \end{aligned}$$

□

The only problem is that $\Sigma_{\tilde{P}}$ must be invertible, but in practice the estimation of Σ_P can result in a singular (or near singular) matrix (for example, if a small number of neighbors are used for the estimation). When this is the case, the algorithm does not modify the patch (see the NLBayes implementation in IPOL[1]).

Flat area trick

The authors in [21] realized that, in homogeneous areas, the estimated Gaussian model of the cluster "learns" part of the noise present in the image. That is, in those cases the variability of the model (captured in Σ_P) is mainly due to the noise, so it is not worthy to be constructed. For those patches, the algorithm sets $\hat{P} = \overline{\tilde{P}}$. This is named as *flat area trick*.

Second pass

Once we have $\hat{u}$, a denoised version of the image u_n, it can be used in a second pass of the algorithm to improve the estimates of the Gaussian model of the clusters, $\overline{P}$ and Σ_P, and then of P.

For a noisy patch $\tilde{P}$ of u_n to be denoised, suppose that we find the k nearest neighbors in the oracle image $\hat{u}$ and build the Gaussian model with them, obtaining the mean $\overline{\tilde{P}}^{\hat{u}}$ and the covariance matrix $\Sigma_{\tilde{P}}^{\hat{u}}$. Now, we suppose that this model is "noise-free" and hence it is not necessary to correct the

[1] `http://www.ipol.im/pub/art/2013/16/`

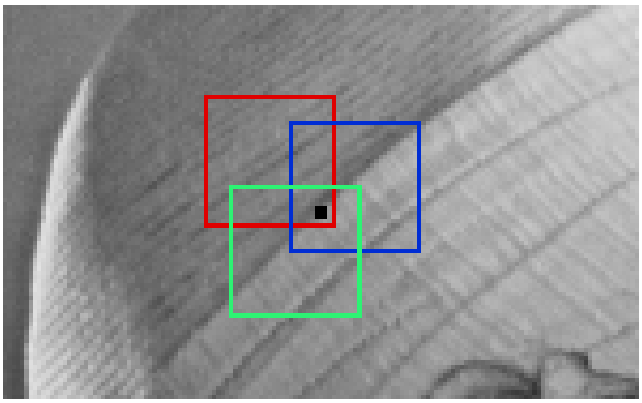


Figure 2.6: Aggregation of estimations of the same pixel belonging to different patches.

covariance matrix as in (2.9). Analogous computations to obtain the final restored patch leads to

$$\hat{P}_2 = \overline{\tilde{P}}^{\hat{u}} + \Sigma^{\hat{u}}_{\tilde{P}}(\Sigma^{\hat{u}}_{\tilde{P}} + \sigma^2 I)^{-1}(\tilde{P} - \overline{\tilde{P}}^{\hat{u}}). \tag{2.11}$$

As noted by the autors, applying more than two steps doesn't give us better results.

2.1.3 Implementation details

There are other details to be considered in orden to obtain a working algorithm, that are discussed in what follows.

Aggregation

The above discussions considers patch restoration. Because there are overlapping patches, that is, a pixel in the image belongs to multiple patches (each of which is restored in a different manner), we need to integrate all these pixel estimations into a single one (see Figure 2.6). This process is called *aggregation*. In this subsection, we describe some of the common ways to do it.

The simplest way to combine all the estimations of a single pixel is to *average* them. That is, if x is a pixel of the image and $\hat{P}_1, \ldots, \hat{P}_m$ are restored patches that contains x in their supports, we can set

$$\hat{u}(x) = \frac{1}{m}\sum_{i=1}^{m} \hat{P}_i(x). \tag{2.12}$$

This is the choice in NLBayes, but we will discuss other ways to do the aggregation in the next sections.

t	$0 \leq \sigma < 20$	$20 \leq \sigma < 50$	$50 \leq \sigma < 70$	$70 \leq \sigma$
1st step	3	5	7	7
2nd step	3	3	5	7

Table 2.1: Patch size t in each step of NLBayes (from [21]).

Patch size

The size of the patches $t \times t$ used in the algorithms of NLMeans and NLBayes influences the quality of the restorations. The choice of the patch size for NL-Bayes in connection with the noise σ is discussed in [21] (see Table 2.1).

In general, large noise values of σ demand larger patch sizes. This may be caused because, when σ is high, the finest details (high frequencies) are completely lost and the best we can do is to recover the lowest frequencies, which are better captured using larger patches.

Number of neighbors

Another parameter that influences the quality of the restoration is the number k of neighbors used to build the Gaussian model (2.5) of the cluster of patches to which the reference patch $\tilde{P}$ to be restored belongs. If only a few of them are considered, the model may not capture the texture behind the noise properly. Moreover, the covariance matrix (2.8) may not even be invertible. This minimum number depends on the patch size t (because t^2 is the dimension of the Gaussian model).

On the other hand, if we keep a large number of neighbors, we could be using patches that are not so similar to the reference one, specially on rare textures. In those cases, the model is not very useful and it will blur the texture to be restored.

So, an optimal number of neighbors needs to be found. Empirically, in [21] the value of k is choosen as a function of the patch size t (see Table 2.2). In the second step, considering that we are using an oracle image for searching the similar patches and thus the patch space is "noise-free", then we only keep the closest patches to the patch to be restored (those belonging to a ball centered at the reference patch), hence the number of k nearest neighbors to be used can be smaller that the maximum showed in Table 2.2 (this number will be maximum mainly in flat areas, and small in rare textures).

Search region of similar patches

In [21], it is mentioned that the NLBayes algorithm uses a *local* search window for computational purposes, which means that it chooses a search window of

k	$t = 3$	$t = 5$	$t = 7$
1st step	30	60	90
2nd step	≤ 30	≤ 60	≤ 90

Table 2.2: Number of neighbors k for the Gaussian model in each step of NLBayes (from [21]).

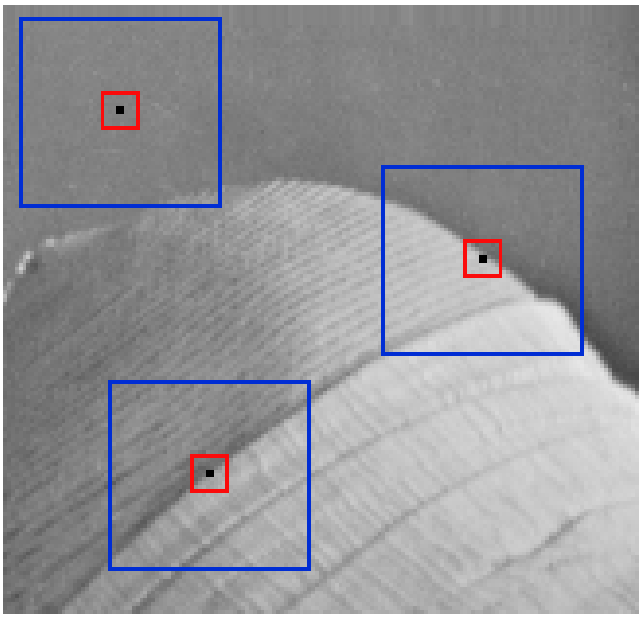


Figure 2.7: Search window of size $T \times T$ for the search of similar patches of the central one of size $t \times t$.

size $T \times T$ with $T > t$ ($t \times t$ being the size of the patches) centered at the reference patch to be denoised (see Figure 2.7). It is well known that the k nearest neighbors search can be the bottleneck of any algorithm if we do not care about it, even more in spaces with dimension greater than 10.

In the next sections, we discuss the adaptations made to NLBayes and provide with another justification of this local approach based on theorical and empirical considerations.

We summarize the complete algorithm of NLBayes in Algorithm 1.

NLB: Non-Local Bayes Restoration Algorithm

input : Noisy and quantised image u_n
input : Total noise affecting the image σ
output: A restored image $\hat{u}$

# First pass:
for *each patch* $\tilde{P} \in u_n$ **do**
  $N_P = \text{obtain\_nearest\_neighbors}(\tilde{P}, u_n)$
  $(\overline{\tilde{P}}, \Sigma_{\tilde{P}}) = \text{obtain\_Gaussian\_model}(\tilde{P}, N_P)$ (Subsection 2.1.2)
  $\hat{P}^{(1)} = \overline{\tilde{P}} + (\Sigma_{\tilde{P}} - \sigma^2 I)\Sigma_{\tilde{P}}^{-1}(\tilde{P} - \overline{\tilde{P}})$ (Subsection 2.1.2)
end
for $x \in \Omega$ **do**
  $\hat{u_0}(x) = \text{aggregate\_patches}(x)$ (Subsection 2.1.3)
end

# Second pass: (Subsection 2.1.2)
for *each patch* $\tilde{P} \in u_n$ **do**
  $N_P^{\hat{u}} = \text{obtain\_nearest\_neighbors}(\tilde{P}, \hat{u}_0)$
  $(\overline{\tilde{P}}^{\hat{u}}, \Sigma_{\tilde{P}}^{\hat{u}}) = \text{obtain\_Gaussian\_model}(\tilde{P}, N_P^{\hat{u}})$
  $\hat{P}^{(2)} = \overline{\tilde{P}}^{\hat{u}} + \Sigma_{\tilde{P}}^{\hat{u}}(\Sigma_{\tilde{P}}^{\hat{u}} + \sigma^2 I)^{-1}(\tilde{P} - \overline{\tilde{P}}^{\hat{u}})$
end
for $x \in \Omega$ **do**
  $\hat{u}(x) = \text{aggregate\_patches}(x)$
end

Algorithm 1: Non-Local Bayes Restoration Algorithm

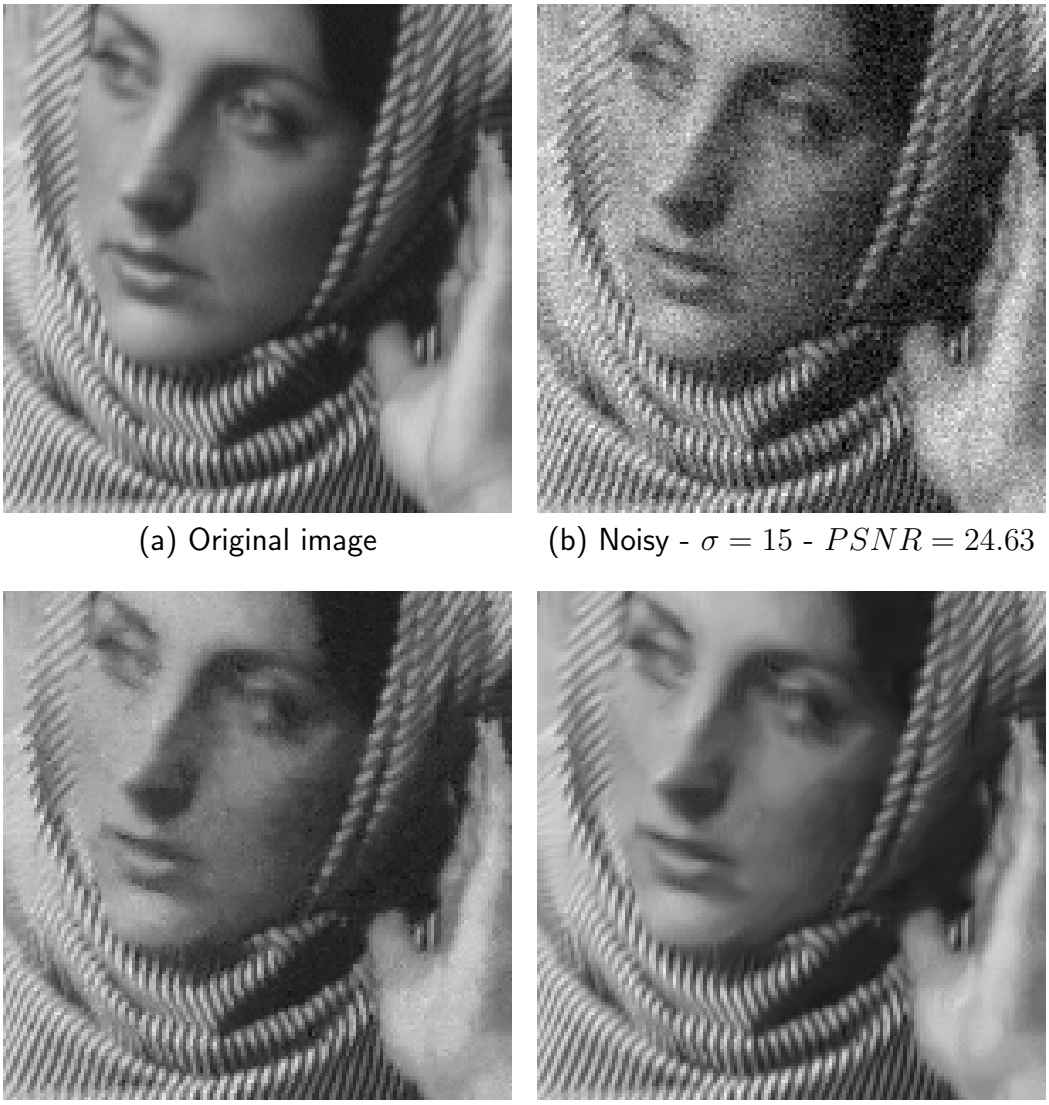
2.1.4 Examples

In Figures 2.8 to 2.10 we can see the differences of NLMeans and NLBayes restoration. Note that NLBayes restores textures better than NLMeans. In

(a) Original image (b) Noisy - $\sigma = 15$ - $PSNR = 24.63$

(c) NLMeans result - $PSNR = 31.56$ (d) NLBayes result - $PSNR = 33.64$

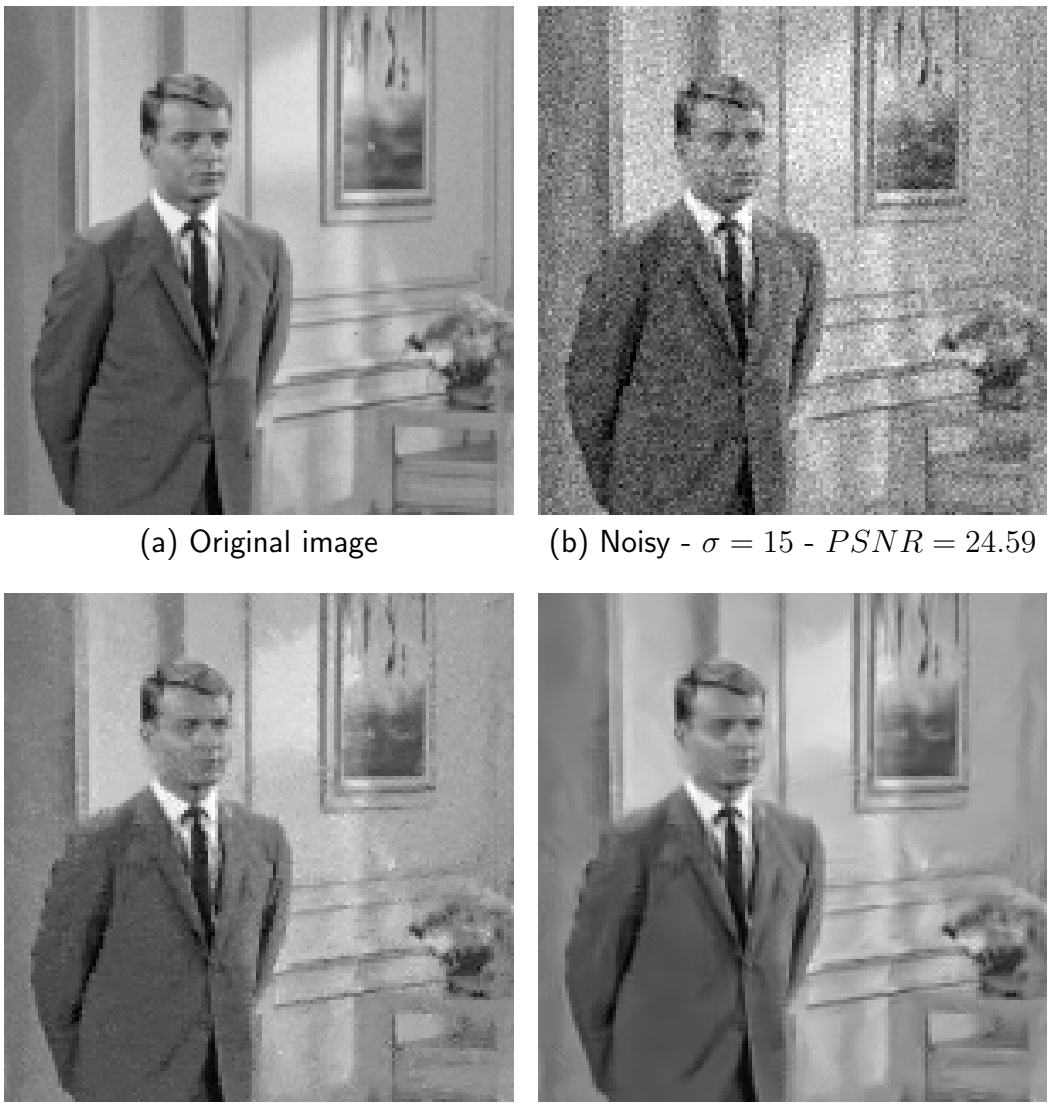
Figure 2.8: Comparison between NLMeans and NLBayes restoration in *Lena*.



(a) Original image (b) Noisy - $\sigma = 15$ - $PSNR = 24.63$

(c) NLMeans result - $PSNR = 29.93$ (d) NLBayes result - $PSNR = 31.82$

Figure 2.9: Comparison between NLMeans and NLBayes restoration in *Barbara*.



(a) Original image (b) Noisy - $\sigma = 15$ - $PSNR = 24.59$

(c) NLMeans result - $PSNR = 30.48$ (d) NLBayes result - $PSNR = 32.11$

Figure 2.10: Comparison between NLMeans and NLMeans restoration in *Living Room*.

this example and in the rest of this thesis, the NLMeans and NLBayes algorithms were taken from the *Image Processing On Line* journal[2].

2.2 Denoising in the wavelet domain

In order to apply this denoising method to our problem, we must introduce two main changes: wavelet (instead of spatial) domain, and quantization (due to the compression scheme). In the following section we proceed with the former problem, and the next chapter is dedicated to the latter.

In the above sections, we discuss the denoising method in spatial domain. This domain is the natural choice in a general context in which the noise is due to the camera sensors capturing light information. In our restoration problem, we have access only to noisy wavelet coefficients that have been quantized; this leads to a highly correlated noise (with the shape of the wavelet basis elements) in the spatial domain. Hence, given the noisy wavelet coefficients, we will try to recover the original coefficients by restoring patches in the wavelet domain.

2.2.1 Simpler approach

In order to understand the behavior of the algorithm in the wavelet domain, and before considering the whole denoising problem as presented in section 1.3, we begin considering the following simplifications:

1. Orthogonal wavelet transform (for example, Daubechies `db8`)
2. A 1-level wavelet decomposition (summary + three detail subbands)
3. No lossy compression (the coefficients are not quantized)

Noise model

First of all, an important observation is that the noise in the wavelet coefficients, when translated from the spatial to the wavelet domain through an orthogonal transform W, is still white noise:

Proposition 2.2. Let W be an $m \times m$ orthogonal matrix and a signal or image $\tilde{u} = u + n$ of length m that is corrupted with white Gaussian noise with zero mean and variance σ^2. Then the transformed residual $W\tilde{u} - Wu$ is also white Gaussian with variance σ^2.

[2] `http://www.ipol.im/`

Proof. If there is no quantization:

$$W\tilde{u}(k) = Wu(k) + Wn(k)$$

and hence the noise in the kth wavelet coefficient,

$$Wn(k) = \sum_x \psi_k(x)n(x),$$

is a *finite* linear combination (due to the compact support of the wavelet basis elements) of normal random variables, which is again normal.

Recall that if $X \in \mathbb{R}^d$ is a random vector and $Y = AX$ (where A is a $d \times d$ matrix), their covariance matrices verify

$$\Sigma_Y = A\Sigma_X A'.$$

In our case, we have $X = n$ (the spatial noise) and $Y = Wn = AX$ the wavelet noise, where A is the orthogonal matrix ($AA' = I$) associated to the orthogonal wavelet transform W. Then the covariance matrix of the wavelet noise is

$$\Sigma_{Wn} = A\Sigma_n A' = A(\sigma^2 I)A' = \sigma^2(AA') = \sigma^2 I.$$

This shows that the wavelet noise is decorrelated, and then it is independent in each coefficient (because the white Gaussian distribution assumption of n) with $Wn(k) \sim N(0, \sigma^2)$. □

Hence, we can use the same white noise model in the wavelet domain, as in the previous sections.

Translated samples

The disadvantage of working directly with wavelet coefficients is that the signal suffers a subsampling process. In other words, in the wavelet domain we do not have access to a patch centered on each pixel of the original image, but only to a quarter of it, because the wavelet transform downsamples the image to half the size in each dimension (see Figure 2.11). Then, for a patch to be restored, there are fewer similar patches to build the Gaussian model than in the spatial domain.

This difficulty is due to the lack of translation invariance of the wavelet basis: for example, if we translate the image one pixel to the right, the coefficients in the wavelet basis will be different (not a translation of the old coefficients). Coifman and Donoho [11] attribute some visual artifacts of wavelet denoising, like the pseudo-Gibbs phenomena in the neighborhood of discontinuities to this lack of invariance.

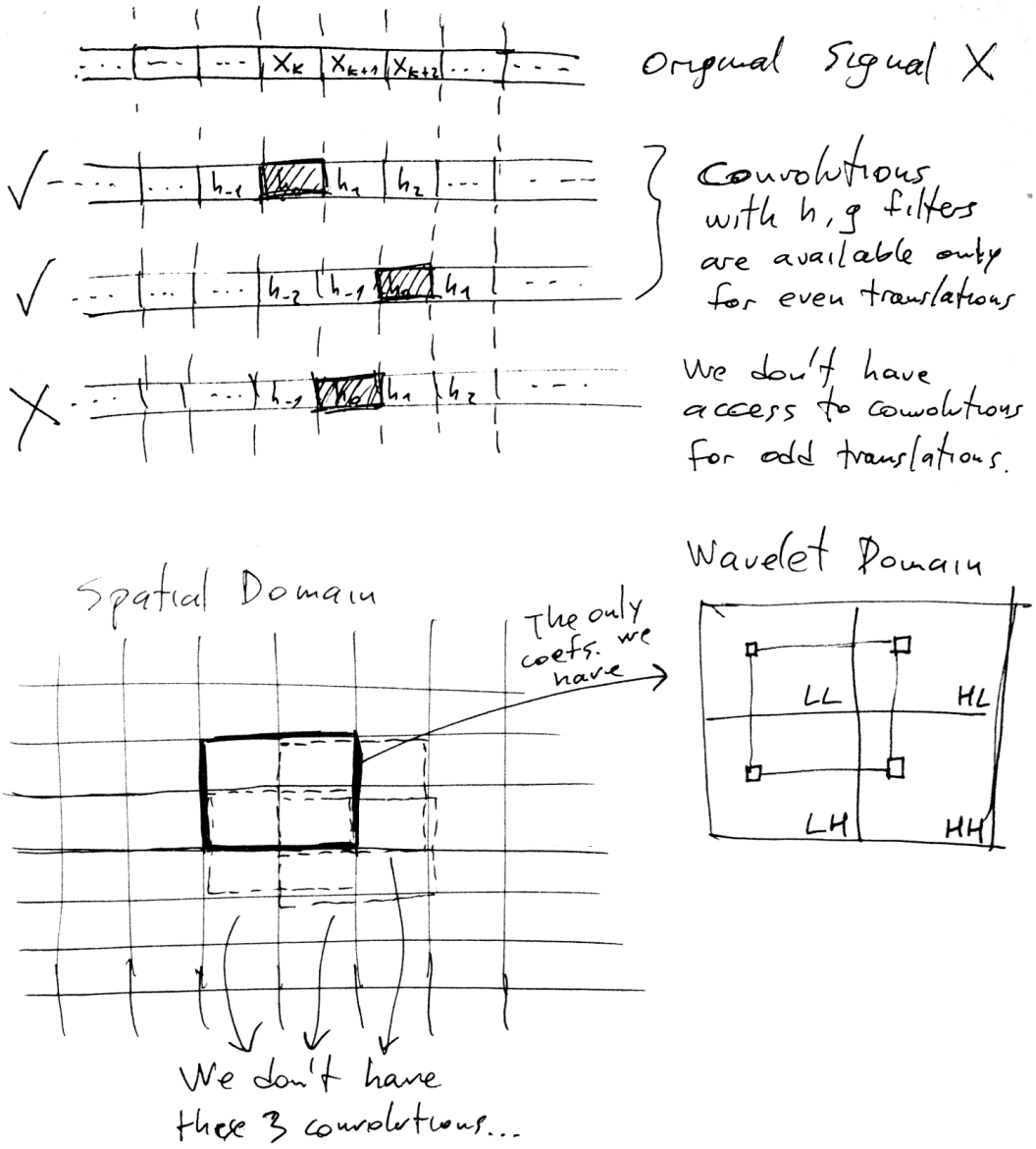


Figure 2.11: Lack of samples in wavelet domain due to downsampling of the wavelet transform.

To compensate this issue, we generate the coefficients of one pixel translation of the noisy spatial image in the horizontal, vertical and diagonal directions. We use these new coefficients only to find similar patches for the first ones (without translation) and to build the Gaussian model for that cluster, but not to denoise them.

Subband soft-thresholding

The values of the wavelet coefficients in the summary are in general larger than the original ones (about 80% larger) but the noise variance is the same (σ^2). Then, the relative noise in the summary will be smaller than in the original image. On the other hand, the detail coefficients on the high-frequency subbands are in general smaller than the original ones and have the same noise variance, which means that the bands are relatively more contaminated with noise than the summary.

This observation motivates a different treatment of the summary and high-frequency subbands:

We need not to worry too much about the summary because it comes to us slightly denoised by the low-pass wavelet filter (see Figure 2.12 (a)). So, we can apply almost any filter to the summary (even none if σ is small enough). But we have a different situation in the bands: the noise-free detail coefficients are, in general, very close to zero, even more in flat areas (they take large values only near sharp edges). Hence, there are a lot of coefficients in the noisy wavelet image bands that are essentialy noise (they should be almost zero). See Figure 2.12 (b) for visual comparison.

A difficulty that appears in this context is that, when finding the k most similar patches to a reference one $\tilde{P}$ in the noisy wavelet image (see below for a description of the used patches), the computations of the distances in the bands are highly corrupted by the noise. Thus, the nearest neighbors computed this way are highly biased by the noise.

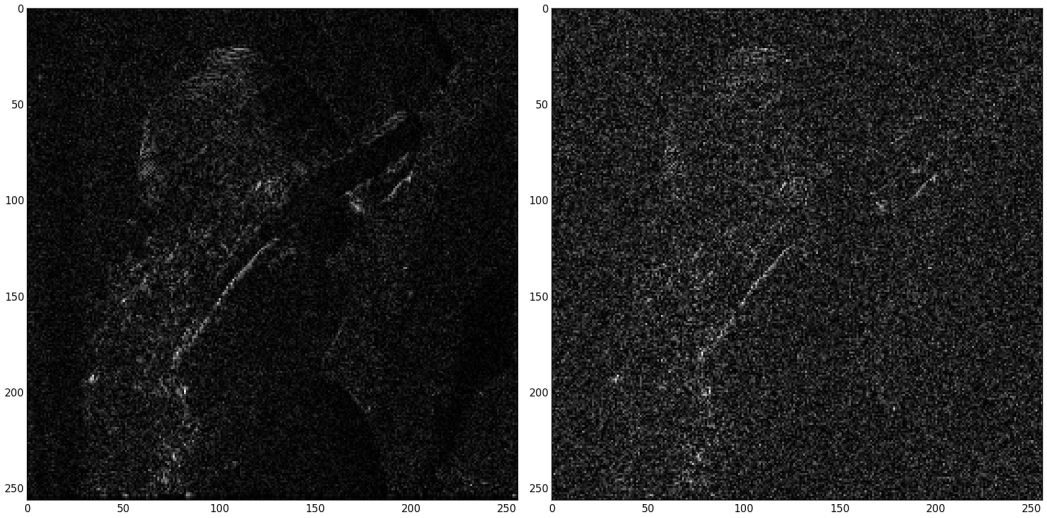
One way to deal with this problem is to apply a thresholding of the detail coefficients, using a threshold that depends on σ, for the purpose of retaining only the significant detail coefficients. A common choice is *soft-thresholding*:

$$\tau_\lambda(x) = \begin{cases} x - sign(x)\lambda & \text{, if } |x| > \lambda \\ 0 & \text{, if } |x| \leq \lambda \end{cases}$$

where $\lambda > 0$ is a threshold parameter. In our case, if $w_n(k)$ is a detail coefficient (in subband LH, HL or HH) then we first set

$$w_{ST}(k) = \tau(w_n(k))$$

(a) Relative noise in summary



(b) Relative noise in diagonal band

Figure 2.12: Relative noise for 1-level `db8` (orthogonal) wavelet decomposition, with spatial noise $\sigma = 20$. Left: Original (noise-free) image, Right: Noisy image

and then we use these new coefficients *only to find the similar patches of* $\tilde{P}$. Note that this procedure consist of a wavelet shrinkage like in Donoho and Johnstone [14].

As an observation, even when we can preprocess the detail coefficients in different ways, the above choice is consistent with the compression scheme to be studied in next chapter, in which the coefficients are naturally thresholded by the quantization process of the compressor.

Patch shape

A key assumption in NLBayes is that there exists self-similarities between patches in the image. This is also true in the wavelet domain, not only in the summary (which is a kind of low resolution version of the spatial image), but also between scales. For example, if there is a sharp edge at a pixel x in the summary, then some of the associated coefficients in the other bands will also be large. Also, in an flat area of the spatial image, both the summary and bands coefficients will be almost constant in the corresponding patches (see Figure 2.13). Hence, the information contained in a pixel (in spatial domain) is then distributed over the four bands of the wavelet domain. So we must consider a patch shape other than a square, which takes into account these correlations between the four bands.

We choose patches of size $t \times t \times 4$, which is equivalent to consider four channel images or three dimensional patches: for each pixel x in a patch P of size $t \times t$ in the summary we have three more values associated to it (one in each of the three bands). An illustration of this type of patches is shown in Figure 2.13. If we take a $t \times t \times 4$ patch of the wavelet domain and we reproject it onto the spatial domain (filling with zeros the rest of the wavelet domain and applying the inverse wavelet transform), we get the result shown in Figure 2.14.

Thus when we compare two patches to determine if they are similar, we are comparing the distances in the summary and separately in the detail subbands, that is, the low and high frequency components of them respectively. This approach is equivalent to comparing distances in the spatial domain if there is no quantization (due to the orthogonality of the wavelet transform, which is energy preserving). But when we quantize the wavelet coefficients, the introduced errors are highly correlated in the spatial domain, appearing in the shape of wavelet basis elements. Hence, the natural way to do denoising is to work in the wavelet domain in which this noise is practically decorrelated (because the quantization is applied to each coefficient independently of the others).

Figure 2.13: Example of a $t \times t \times 4$ patch in wavelet domain. Note the correlations between coefficients of the summary and the other subbands in corresponding positions.

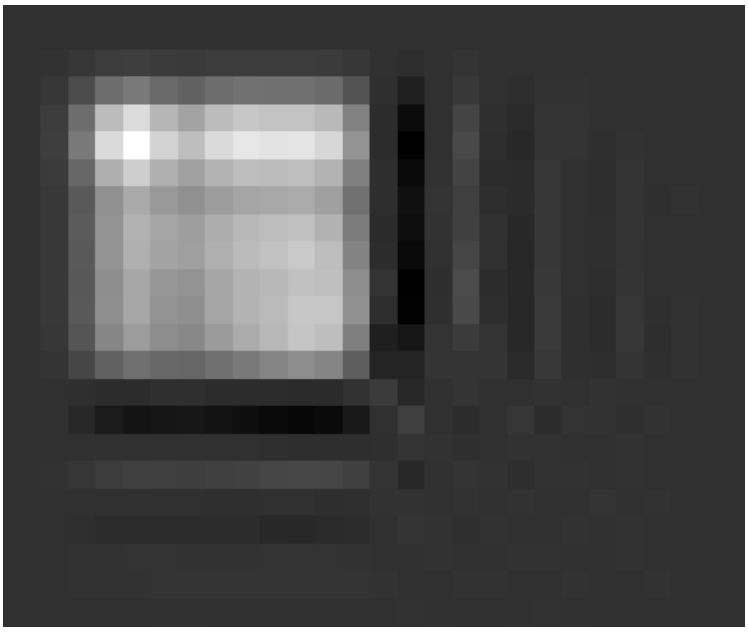


Figure 2.14: Reprojection of a $5 \times 5 \times 4$ patch of the wavelet domain onto the spatial domain. Note that it consists loosely of a 10×10 patch plus some short oscillations with the shapes of the wavelet basis functions in the horizontal, vertical and diagonal directions.

Search window in similar patches search

In Subsection 2.1.3 we mentioned that, despite the usage of the term *non-local* for the reviewed algorithms, the search of similar patches is done on *local search windows*. This is done to reduce computations in the k nearest neighbors search (which are, in general, expensive) to avoiding prohibitive running times.

There are algorithms for similar patches search that exploit the image intrinsic structure. The *Generalized PatchMatch algorithm* [3] is a fast algorithm that tries to propagate the similarities found between two patches to the rest of the patches near the reference one: if a patch centered at pixel x is similar to a patch centered at x', it is highly probably that patch at $x + l$ is similar that the patch at $x' + l$ (at least for small values of l). This permits to compute *global* searches on images and videos faster than using euclidean distances (without considering the spatial corrections), even taking into account various scales and rotations of the patches.

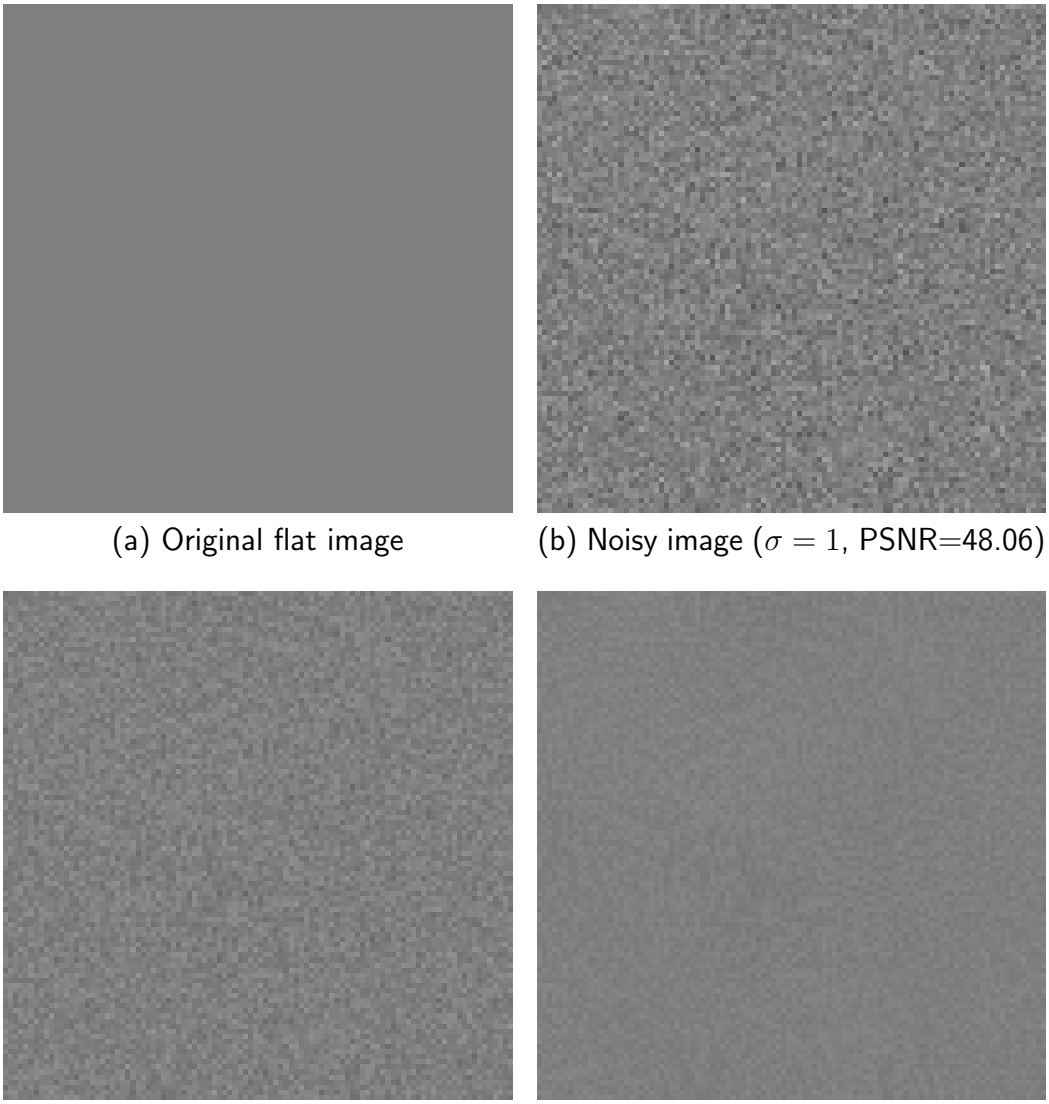
Also in [3] it is pointed out that this type of *global search* of similar patches (that is, the search region is the whole image) is not always better than local search. Intuitively, the larger the search window is, the more similar patches to the reference one can be obtained and thus the denoising will be better. But, in uniform regions, a reference patch will be mainly noise (there is no texture present in it) and the nearest neighbors will be those that best match the noise. In other words, the Gaussian model "learns" part of the noise present in the flat areas because it is the only significant content there. This is related with the *overfitting* problem in supervised machine learning. Figure 2.15 shows the results of an experiment with local and global searches.

So, using *local* instead of *global* patch search for similarities serves two purposes: speed and avoiding overfitting. Another approach that can overcome these difficulties is to use an approximate search like FLANN [26]. This fast algoritm sacrifices some precision in order to gain speed. It can be shown that in the search of image patches in which there are correlations between dimensions, the acceleration can be high without sacrifying too much precision, which would be the case if the points were random samples (see [26]).

We have found experimentally that using FLANN it can be reduced the effects of noise learn with faster times than exact global search, and then to obtain better denoising results. But even in this case, the results are not better than to use a local search approach. So we adopt the local search window strategy as in the original NLBayes algorithm.

Covariance estimation

As was pointed out in NLBayes algorithm, the estimation of the covariance matrix $\Sigma_{\tilde{P}}$ can result in a singular matrix instead of a positive definite one.



(a) Original flat image

(b) Noisy image ($\sigma = 1$, PSNR=48.06)

(c) NLBayes using *global* search (PSNR=52.17)

(d) NLBayes using *local* search (PSNR=58.49)

Figure 2.15: Comparision between global and local search. The original image is a flat one with size 100×100 and gray value equal to 100 (of 255). In both searches, 3×3 patches were used. In (d) a 21×21 local search window centered in each reference patch was used. The dynamic range showed consists of the gray values between 90 and 110 for better visualization.

NLBayes maintains the noisy patch as it is, with the hope that, in the aggregation process, the noise will be mitigated.

In our approach, to avoid this problem we set a threshold $\epsilon > 0$ and every eigenvalue of $\Sigma_{\tilde{P}}$ lower than ϵ is set equal to it. That is, if

$$M = \Gamma \Lambda \Gamma^{-1}$$

is the spectral decomposition of M, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_r)$ where λ_i's are the eigenvalues of M, and Γ is the matrix of the corresponding basis of eigenvectors, then we define $\tilde{\lambda}_i = \max(\lambda_i, \epsilon)$ and

$$\Pi(M) = \Gamma \tilde{\Lambda} \Gamma^{-1}$$

where $\tilde{\Lambda} = \operatorname{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_r)$. Hence, we get that $\Pi(\Sigma_{\tilde{P}})$ is always positive definite and we use it as our covariance matrix estimation. Also, when we need to invert $\Sigma_{\tilde{P}} - \sigma^2 I$, we first apply this operator Γ when the inversion fails.

Aggregation

As explained before, the restored patches overlap each other and need to be combined in order to produce an entire image (in this case, a wavelet domain image). The simplest thing we can do is to combine the estimations of each coefficient contained in all the patches to which it belongs as an average (weighted or not), as in (2.12).

The average of estimated pixel values of different patches does not consider the fidelity of the restorations. Let us denote by

$$F(X, \tilde{P}) = \frac{\|X - \overline{\tilde{P}}\|^2}{\sigma^2} + (X - \overline{\tilde{P}})^T (\Sigma_{\tilde{P}} - \sigma^2 I)^{-1} (X - \overline{\tilde{P}}) - \log(C_2)$$

the minimized functional in (2.9) used to estimate the restored patches $\hat{P}_i$, where

$$C_2 = \frac{1}{\sqrt{(2\pi)^N |\det \Sigma_P|}}$$

of (2.6). Then the restored patch is

$$\hat{P}_i = \arg\min_X \ F(X, \tilde{P}_i).$$

We can use the values $e^{-F(\hat{P}_i, \tilde{P}_i)}$ as a measure of confidence in the restoration of $\tilde{P}_i$. Hence, we can set

$$\hat{u}(x) = \frac{\sum_{i=1}^m e^{-F(\hat{P}_i, \tilde{P}_i)} \hat{P}_i(x)}{\sum_{i=1}^m e^{-F(\hat{P}_i, \tilde{P}_i)}} \tag{2.13}$$

This type of *weighted aggregation* was first proposed in [20]. Other aggregation methods are studied in [32].

Recalling that we are working in the wavelet domain, the $t \times t \times 4$ restored patches contains coefficients of differents frequency subbands. It can be asked if it is better to do the aggregation in the spatial domain, that is, to transform the restored patches and *then* aggregate the different estimations of the *pixels* instead of doing the same with the *coefficients*. But it can be proved that in this case the two procedures are equivalent due to the linearity of the wavelet transform.

Indeed, suppose that $x \in \Omega$ is a pixel in the spatial domain to be restored, and that $P_1^S, \dots, P_l^S$ are the projected patches from the wavelet domain that contain the pixel x in its support (they are finite because the wavelet basis has compact support, see Figure 2.14). Let us denote by $P_1^W, \dots, P_l^W$ the corresponding patches in the wavelet domain. We have:

$$P_i^S = W^{-1} P_i^W \quad i = q, \dots, l.$$

Suppose we fill with zeros outside the support of the above patches in the spatial and wavelet domain respectively, and now we can use x as an absolute coordinate in the spatial domain for all the patches P_i^S. Thus for a given linear aggregation procedure applied we have

$$\left(\sum_{i=1}^{l} \alpha_i P_i^S \right)(x) = \left(\sum_{i=1}^{l} \alpha_i W^{-1} P_i^W \right)(x) = W^{-1} \left(\sum_{i=1}^{l} \alpha_i P_i^W \right)(x).$$

This shows that, for a pixel x, doing the aggregation in the spatial domain is the same that doing it in the wavelet domain (with the patches associated to x) and later project the result back to the spatial domain. We choose to do the aggregation procedure in the wavelet domain for simplicity.

Second pass

Once we have a restored wavelet image $\hat{w}$, we can use it as an oracle for a better estimation of the Gaussian model of each cluster of similar patches in the image. That is, we use the oracle image $\hat{w}$ to better estimate the mean and covariance of the Gaussian models of the clusters, and then to obtain a cleaner estimation of the restored patches, as in the NLBayes algorithm.

Algorithm

In Algorithm 2 we summarize the NLBayes algorithm adapted to the wavelet domain, which we call *Wavelet Non-Local Bayes* (WNLBayes). Here we present the simplest version that uses orthogonal wavelet transform, 1-level wavelet

decomposition and no quantization. In the following we present the other modifications that remain to obtain an algorithm able to denoise the CCSDS standard compressed satellite images.

WNLB: Wavelet Non-Local Bayes Restoration Algorithm (1-level)

input : 1-level wavelet coefficients w_n of noisy image u_n
input : Total noise affecting the image σ
output: A restored image $\hat{u}$

# First step:
for *each patch* $\tilde{P}_k \in w_n$ *(of size* $t \times t \times 4$*)* **do**
  $N_k = \text{obtain\_nearest\_neighbors}(\tilde{P}_k, w_n)$ (on thresholded patches)
  $(\overline{\tilde{P}}_k, \Sigma_{\tilde{P}_k}) = \text{obtain\_Gaussian\_model}(\tilde{P}_k, N_k)$
  $\hat{P}_k^{(1)} = \overline{\tilde{P}}_k + (\Sigma_{\tilde{P}_k} - \sigma^2 I)\Sigma_{\tilde{P}_k}^{-1}(\tilde{P}_k - \overline{\tilde{P}}_k)$ (see (2.10))
end
for $k \in \Omega_w$ **do**
  $\hat{w_0}(k) = \text{aggregate\_patches}(k)$
end

# Second step:
for *each patch* $\tilde{P}_k \in w_n$ **do**
  $N_k^{\hat{w}} = \text{obtain\_nearest\_neighbors}(\tilde{P}_k, \hat{w}_0)$
  $(\overline{\tilde{P}}_k^{\hat{w}}, \Sigma_{\tilde{P}_k}^{\hat{w}}) = \text{obtain\_Gaussian\_model}(\tilde{P}_k, N_k^{\hat{w}})$
  $\hat{P}_k^{(2)} = \overline{\tilde{P}}_k^{\hat{w}} + \Sigma_{\tilde{P}_k}^{\hat{w}}(\Sigma_{\tilde{P}_k}^{\hat{w}} + \sigma^2 I)^{-1}(\tilde{P}_k - \overline{\tilde{P}}_k^{\hat{w}})$
end
for $k \in \Omega_w$ **do**
  $\hat{w}(k) = \text{aggregate\_patches}(k)$
end
$\hat{u} = \text{wavelet\_synthesis}(\hat{w})$

Algorithm 2: 1-level Wavelet Non-Local Bayes Restoration Algorithm

2.2.2 Multi-scale denoising

In the previous section, for simplicity, we assumed that the image was only decomposed in one-level of the wavelet transform. This is not the case in the problem to be studied, because the CCSDS standard imposes to work with 3-level wavelet transform. Next we describe how we can apply the above algorithm to an image with this type of decomposition.

Figure 2.16: Decomposition of an image into one and three wavelet levels.

The more natural way to denoise a 3-level wavelet image is to work on each level separately. That is, we first apply the basic WNLB algorithm described above to the 3rd level (containing the lowest frequencies), thus obtaining a restored level that we next use as the summary for the 2nd level, and so on until we restore each of the three levels.

Also, we must consider *variable sizes* of the patches at each level, because the size of the domain and the signal-to-noise ratio vary in each scale. Thus in each level we must adjust the optimal patch width t in order to obtain better denoising results. In Table A.2 (Appendix A.2) we summarize the parameters used to generate the examples in this thesis that work well for several noise levels.

We must note that the summary for the second level, which is the result of the one-level algorithm of the third level, is already "denoised", so in the second level we only use this summary for neighbors search and build the Gaussian model, but in this level we only estimate the details coefficients on the bands and mantain unchanged the summary. The same procedure is applied to denoise the first level (highest frequencies). See Figure 2.17.

For the second step, like NLBayes, we take the result of the first step of WNLBayes to better estimate the Gaussian model of the algorithm (as an oracle). It may happen that the parameters must be adjusted in order to optimize the results. The adjustment of the optimal parameters in each step of the WNLBayes algorithm has not yet been made and will be considered in future work (see Chapter 5).

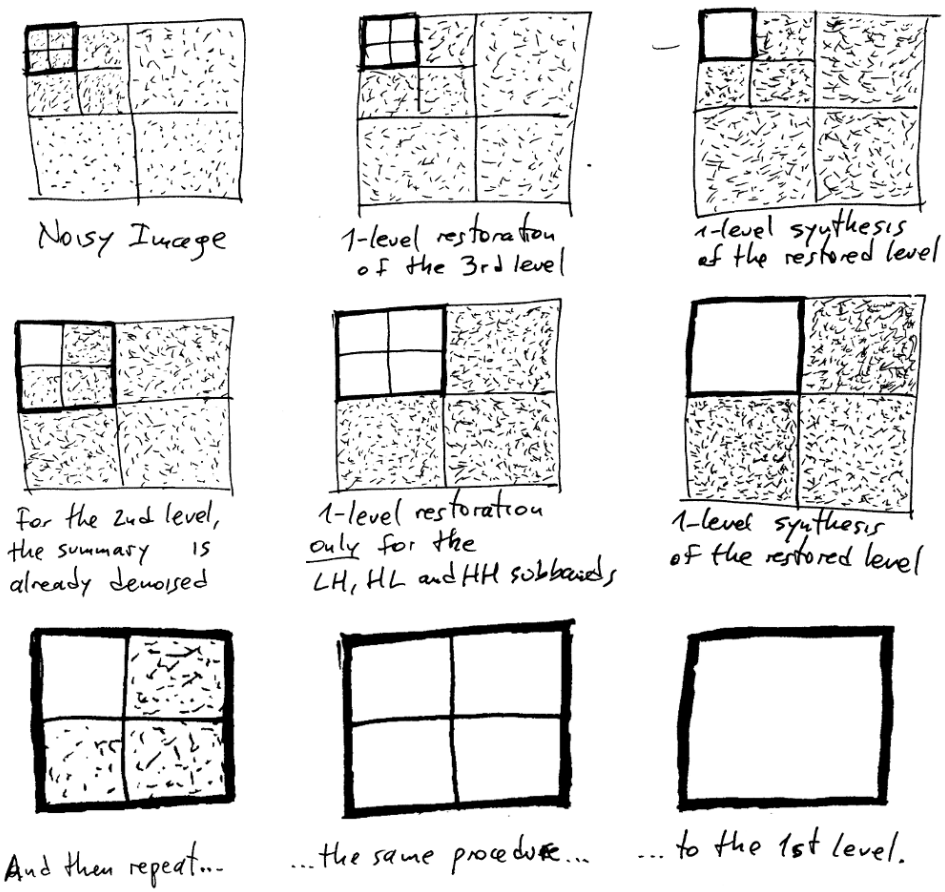


Figure 2.17: Scheme of WNLBayes multi-scale algorithm.

An issue to be noted is about the size of the target images: in a 3-level wavelet decomposition, the summary (lowest frequency subband, see Figure 2.16) is eight times smaller than the original (spatial) image. For example, the most common test images used in image processing are 512×512-sized images, and thus the summary of the 3rd level will have size 64×64. In order to obtain good results, we must apply the multi-scale version of the algorithm to larger images, because visual artifacts can be observed in 512×512 images.

2.2.3 Biorthogonal transform

In order to design an algorithm able to deal with images compressed with the CCSDS of JPEG2000 standard, we must introduce another change to our simple algorithm: *biorthogonal* instead orthogonal wavelet transform. As was explained in the introduction chapter, although the biorthogonal transform is not energy preserving (a nice property of the orthogonal transform) due to the lack of orthogonality, the CDF 9/7 wavelet transform is *almost* orthogonal in the sense presented in [36], that is,

$$WW^* \simeq I, \tag{2.14}$$

where W is the matrix associated to the wavelet transform (see Subsection 1.1.3). Thus, in the CDF 9/7 case, the covariance matrix of the wavelet noise is

$$\Sigma_{Wn} = W\Sigma_n W^* = W(\sigma^2 I)W^* = \sigma^2 WW^* \simeq \sigma^2 I$$

in the sense of (2.14). This shows that we consider the wavelet noise as nearly decorrelated, because the correlations between coefficients corresponding to different basis positions are very close to zero. Hence, in the wavelet domain we can use the same white noise model of the previous sections with negligible error.

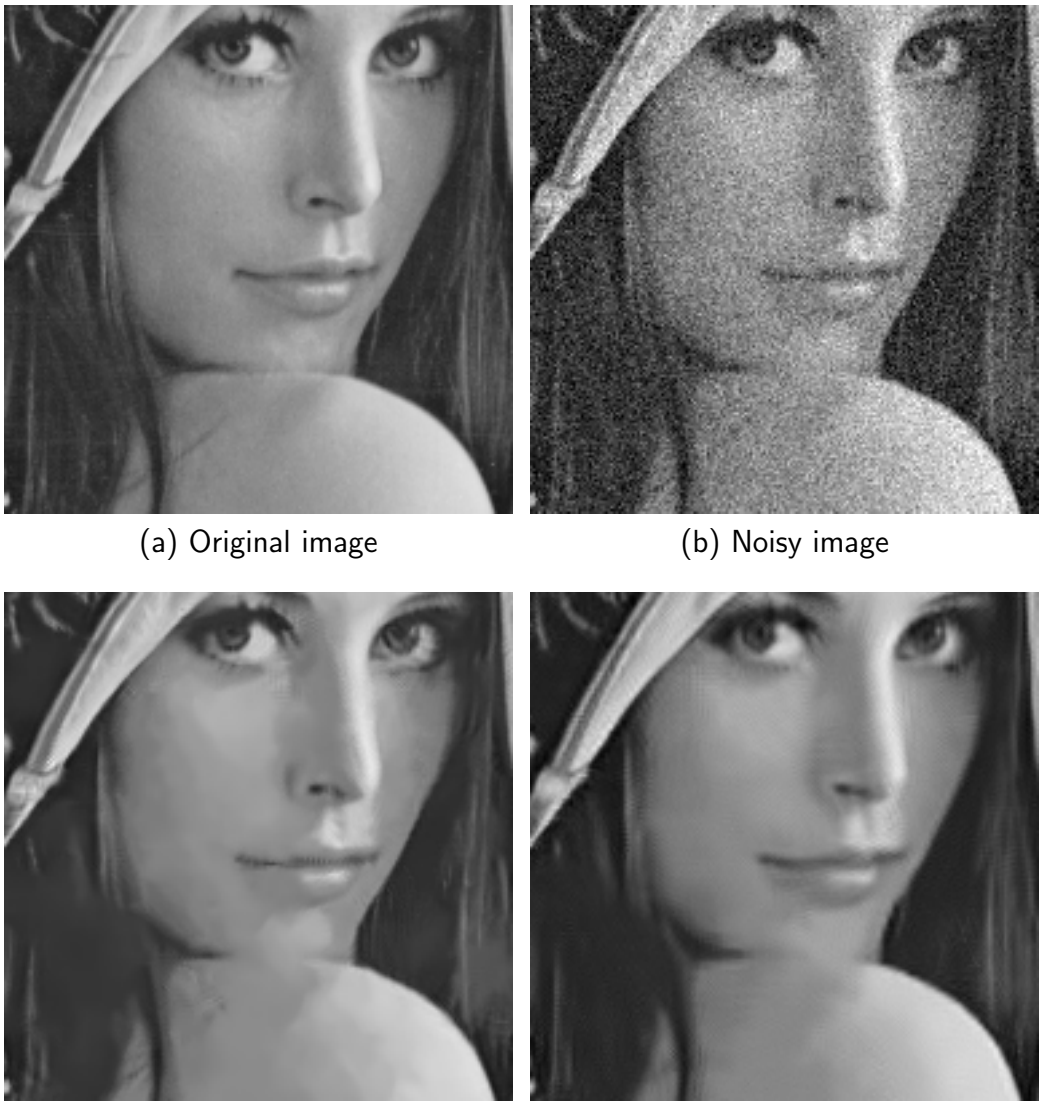
2.2.4 Comparison with NLBayes

Here there are some examples of the results of our algorithm used for denoising, although it is not the final purpose of it. We present a restored image for the 3-level version, and the comparison with the result of NLBayes. Note that WNLBayes restores better some textures (like Lena's hair, shoulder or lines in the background) than others (like the eyes). In Chapter 4 we present several results of WNLBayes restoration in other images.

(a) Original image (b) Noisy image - $PSNR = 22.15$

(c) NLBayes - $PSNR = 32.91$ (d) WNLB (3-level) - $PSNR = 32.00$

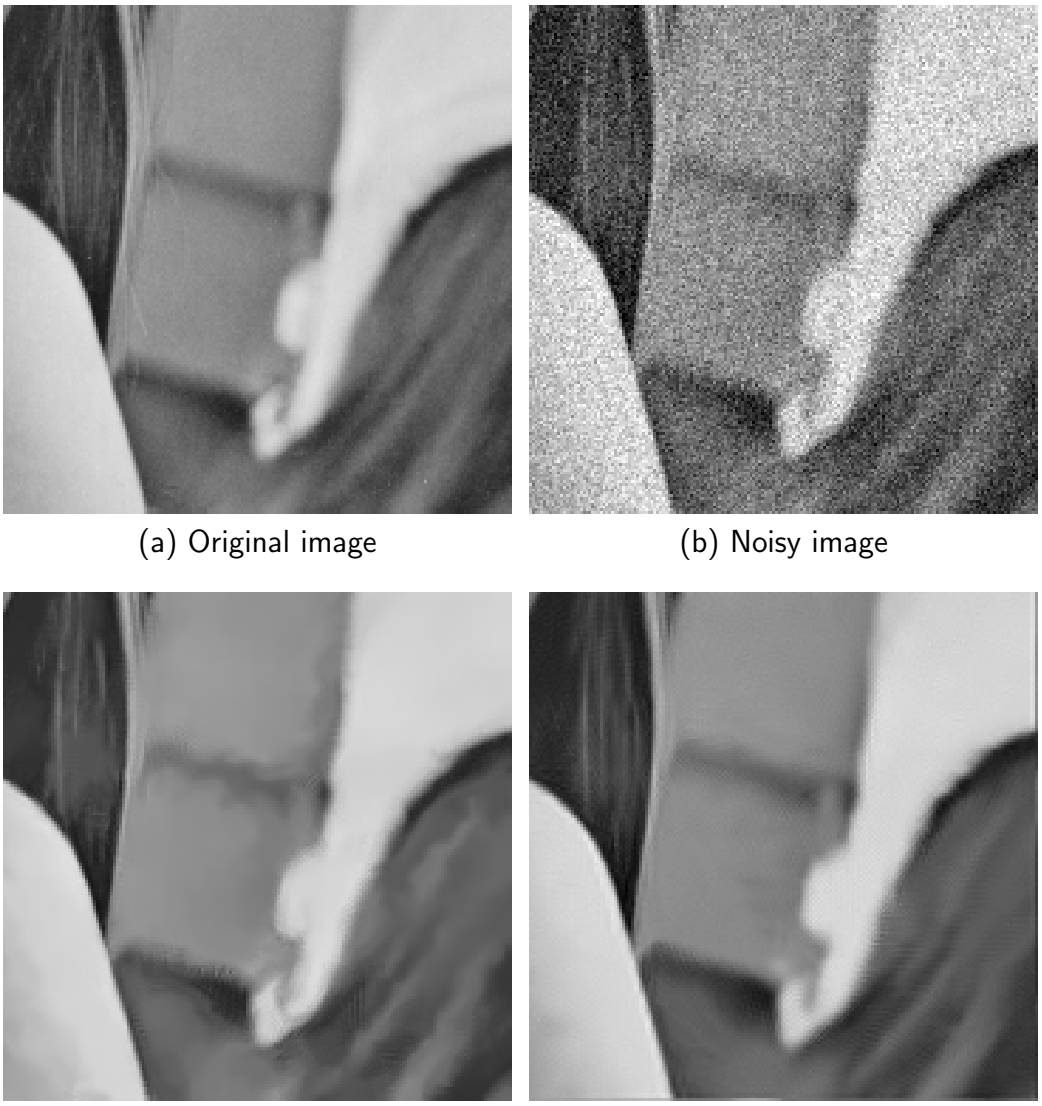
Figure 2.18: Comparision of NLBayes and WNLB (3-level) denoising on Lena with noise $\sigma = 20$.



(a) Original image (b) Noisy image

(c) NLBayes result (d) WNLB (1-level) result

Figure 2.19: Zoom 1 of Figure 2.18.



(a) Original image (b) Noisy image

(c) NLBayes result (d) WNLB (1-level) result

Figure 2.20: Zoom 2 of Figure 2.18.

Chapter 3

Decompression

Compression techniques are necessary when storage capacity and transmission velocity are considered for the design of image or video systems. If we can represent the same (or almost the same) image using half or less of its bits, it will let us to store more than the double of information in the same hard disk or cloud storage, or transmit the same image in the half of the time that it would take if no compression procedure was used (which is very important in real-time applications). Also, we need high compression ratios specially when the image is too large to be transmited over a narrow band channel, as is the case of high-resolution satellite imagery.

A wavelet transform, such as the biorthogonal CDF 9/7 used in Chapter 2, expresses the image in another basis with the same number of coefficients than before, but using *float* values (requiring 32 or 64 bits each) instead of *integer* (as is common in spatial images of 8 or 12 bits gray values). Thus, storing the wavelet domain image as it is will lead to enlargement the original size of the image, which is obviously undesired, as we want to *compress* the data. This problem is called *Dynamic Range Expansion* in [16] (Green book). A way in which we can represent the high precision coefficients into a fewer bits representation of them is called *quantization*.

In this chapter, we give a more detailed discussion of the process of *scalar quantization* of coefficients, which is a specific rounding operation as was described in Chapter 1, and an approach to deal with the loss of information due to the quantization procedure. First, we present our method for joint denoising/dequantizing in spatial domain for simplicity, later we give some details of the CCSDS (and also JPEG2000) compression scheme as our principal target for our denoising algorithm, observing that NLBayes does not have optimal results in this context, and finally we present a complete algorithm to deal with noisy/quantized images for estimating the original coefficients in the wavelet domain that at the same time performs denoising.

3.1 Dequantizing in spatial domain

For the sake of clarity, we first present our dequantizing/denoising method in the spatial domain. In this section, we will work with the quantization of noisy images:

$$u_{qn} = Q(u + n), \qquad n(x) \sim N(0, \sigma^2)\, i.i.d. \tag{3.1}$$

The function Q is a *uniform q-step quantizer* of the form

$$Q_q(x) = \left\lfloor \frac{x}{q} + 0.5 \right\rfloor q \tag{3.2}$$

where $q \in \mathbb{Z}^+$. The 0.5 in the above definition is to give the center of each quantization interval as the quantized coefficients:

Proposition 3.1. $Q_q(x) = \tilde{x}$ if and only if $x \in \left[\tilde{x} - \frac{q}{2}, \tilde{x} + \frac{q}{2}\right)$.

Proof.

$$\begin{aligned}
\left\lfloor \frac{x}{q} + 0.5 \right\rfloor q = \tilde{x} \quad &\Leftrightarrow \quad \left\lfloor \frac{x}{q} + 0.5 \right\rfloor = \frac{\tilde{x}}{q} \quad &&\Leftrightarrow \quad \frac{x}{q} + 0.5 \in \left[\frac{\tilde{x}}{q}, \frac{\tilde{x}}{q} + 1\right) \\
&\Leftrightarrow \quad \frac{x}{q} \in \left[\frac{\tilde{x}}{q} - 0.5, \frac{\tilde{x}}{q} + 0.5\right) \quad &&\underset{q>0}{\Leftrightarrow} \quad x \in \left[\tilde{x} - \frac{q}{2}, \tilde{x} + \frac{q}{2}\right).
\end{aligned}$$

□

Basically, for each $k \in \mathbb{Z}$ this quantizer maps the whole interval $[kq, (k+1)q)$ onto the value $(k+1/2)q$ (the center of the quantization interval). In an image, the quantizer Q_q is applied to each pixel value.

We see that the presence of noise can help in the dequantization process. A parameter that indicates if the noise can help in dequantizing is q/σ, that is, the relation between quantization and noise:

- If q/σ is too large (say $q/\sigma > 7$) then the noise variance is very small compared with the quantization step, and then outliers are very rare (only appear when the signal is very close to a quantization threshold).

- If q/σ is too small (say $q/\sigma < 3$) then σ is much greater than q and thus most of the coefficients will be outliers and then the quantization doesn't affect too much to the noisy image (we see almost the same noise after quantization).

- In the middle (say $3 \leq q/\sigma \leq 7$) the noise may help to distinguish where a coefficient is in the center of the quantified interval, when is close to the left border or when is close to the right border.

A one-dimensional example that illustrates this fact is shown in Figure 3.1.

In Figure 3.2 we can see the loss of image quality of quantization in spatial domain on a 8-bit gray image. If no quantization is applied, they are 256 gray values availables to represent the image. Quantizing with step $q = 10$ reduce to 26 the number of gray values, and with $q = 30$ they are only 9 distinct gray values.

For example, in a noise-free homogeneous region whose pixel values belongs to the same quantization interval, all its pixels will be quantized to the same gray value (see that effect on the cheeks of Lena in Figure 3.2 (e)). If we add noise to that region, now some pixels will fall in a different quantization interval depending on how far of the center they are (we call to this type of quantified coefficient as *outlier*) and thus we can better estimate the texture of the region before quantization (as we see in the same example region in Figure 3.2 (f)). This is related to the *dithering* technique, and the reason for observing a staircase effect in (e) but not in (f).

The above discussions motivates the development of an algorithm that performs *jointly* denoising and dequantizing.

3.1.1 Candès' method

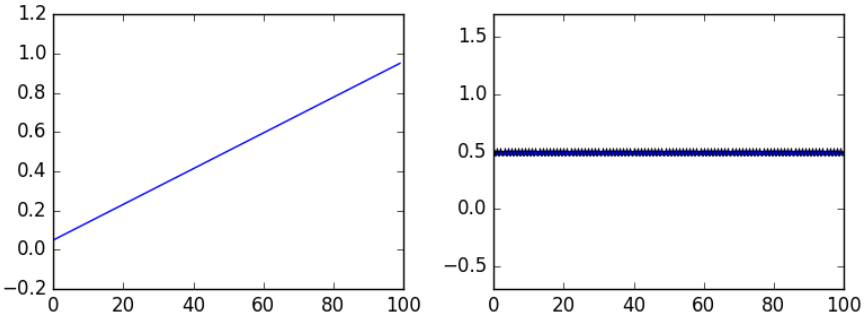
To restore original coefficients from quantized ones (*dequantizing*), we follow a method inspired by Candès et al [38] and using again the Gaussian model of Lebrun et al. [21]: Let's denote by $\tilde{P}$ a patch of the quantified noisy image u_{qn} and P the corresponding original patch to be estimated, both with support Ω_P. To simplify notation, we denote by P_x the pixel x of a patch P and by $[a_x, b_x]$ the quantization interval corresponding to the coefficient $\tilde{P}_x$.

The conditional pdf of $\tilde{P}$ given P is

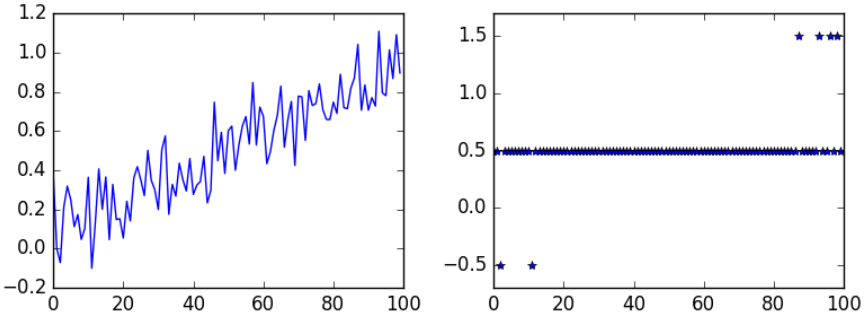
$$\begin{aligned}\mathbb{P}(\tilde{P} \,|\, P) &= \prod_{x\in\Omega_P} \mathbb{P}(\tilde{P}_x \,|\, P_x) = \prod_{x\in\Omega_P} \mathbb{P}(Q(P_x + n_x) = \tilde{P}_x) \\ &= \prod_{x\in\Omega_P} \mathbb{P}(P_x + n_x \in Q^{-1}(\tilde{P}_x) = [a_x, b_x]) \\ &= \prod_{x\in\Omega_P} \mathbb{P}\left(\frac{n_x}{\sigma} \in \left[\frac{a_x - P_x}{\sigma}, \frac{b_x - P_x}{\sigma}\right]\right).\end{aligned}$$

But $n_x/\sigma \sim N(0,1)$, then the above is the probability that a normal random variable falls into an interval. So

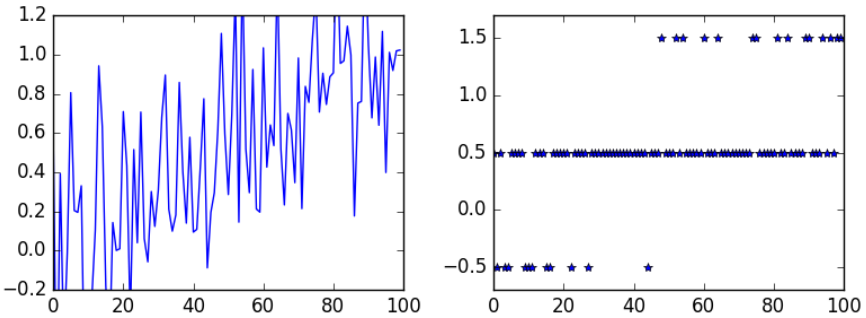
$$\mathbb{P}(\tilde{P} \,|\, P) = \prod_{x\in\Omega_P} \left[\Phi\left(\frac{b_x - P_x}{\sigma}\right) - \Phi\left(\frac{a_x - P_x}{\sigma}\right)\right]$$



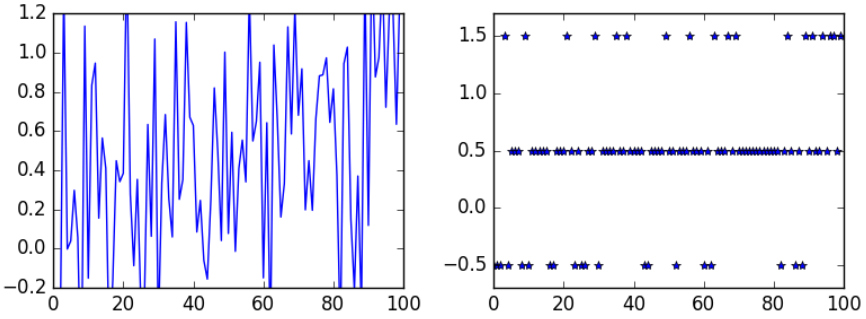
(a) Original signal - $q/\sigma = +\infty$



(b) Noisy signal - $q/\sigma = 7$

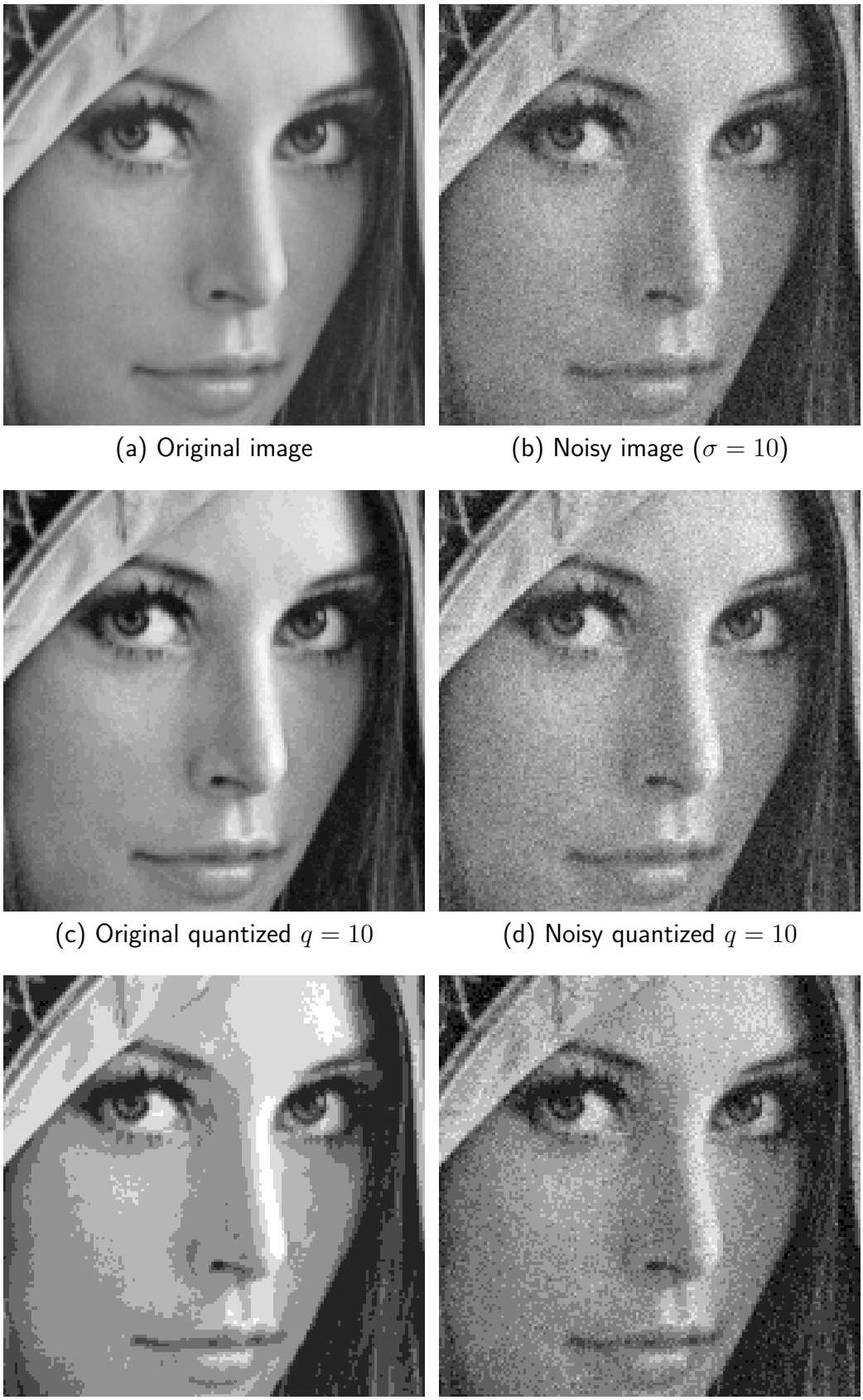


(c) Noisy signal - $q/\sigma = 3$



(d) Noisy signal - $q/\sigma = 2$

Figure 3.1: One-dimensional example in which the noise can help in dequantizing the original signal. First row: Original signal and its quantization. Second to bottom rows: If we add noise to the above signal and then quantize, the outliers can give information about the original signal.



(a) Original image (b) Noisy image ($\sigma = 10$)

(c) Original quantized $q = 10$ (d) Noisy quantized $q = 10$

(e) Original quantized $q = 30$ (f) Noisy quantized $q = 30$

Figure 3.2: Effects caused by quantization in spatial domain on a 8-bit gray image.

where

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2}\, dt$$

is the *normal distribution function* (ndf).

If we suppose again that P follows a Gaussian model with mean $\overline{P}$ and covariance matrix Σ_P, then

$$\mathbb{P}(P) = c. \exp\left(-\frac{(P-\overline{P})\Sigma_P^{-1}(P-\overline{P})}{2}\right).$$

Varying P of domain Ω_P, we obtain

$$\begin{aligned}\arg\max_P \ \mathbb{P}(P \mid \tilde{P}) &= \arg\max_P \ \mathbb{P}(\tilde{P} \mid P)\mathbb{P}(P) \\ &= \arg\min_P \ -\log \mathbb{P}(\tilde{P} \mid P) - \log \mathbb{P}(P).\end{aligned}$$

Thus, our main minimization problem is

$$\min_P D(P) + R(P) \tag{3.3}$$

where

$$D(P) = \sum_{x \in \Omega_P} -\log\left[\Phi\left(\frac{b_x - P_x}{\sigma}\right) - \Phi\left(\frac{a_x - P_x}{\sigma}\right)\right] \tag{3.4}$$

is a convex differentiable *data fit* function and

$$R(P) = \frac{1}{2}(P-\overline{P})\Sigma_P^{-1}(P-\overline{P}) \tag{3.5}$$

is a cuadratic *regularization* term.

Proposition 3.2. The data fitting functional D in (3.4) is convex.

Proof. Consider

$$h(z) = \Phi(\alpha - z) - \Phi(\beta - z) = \frac{1}{\sqrt{2\pi}} \int_{\alpha - z}^{\beta - z} e^{-t^2}\, dt.$$

This function is the convolution of two log-concave functions (the Gaussian density and the indicator function in $[\alpha, \beta]$) which is log-concave too (see [5], section 3.5.2), hence D is convex (is a sum of convex functions of the form $-\log h(z/\sigma)$). □

Note that the functional (3.3) to be minimized is very similar to (2.7) of NLBayes denoising algorithm. The regularization term (3.5) is the same (because is based in the same Gaussian model for the cluster) and the data fitting term (3.4) takes into account the quantization of the coefficients. But now, this problem is not quadratic and we no longer have the possibility to obtain an analytic solution like (2.10) in NLBayes. Hence, in order to obtain the restored patches, we have to apply an optimization algorithm to the convex problem (3.3).

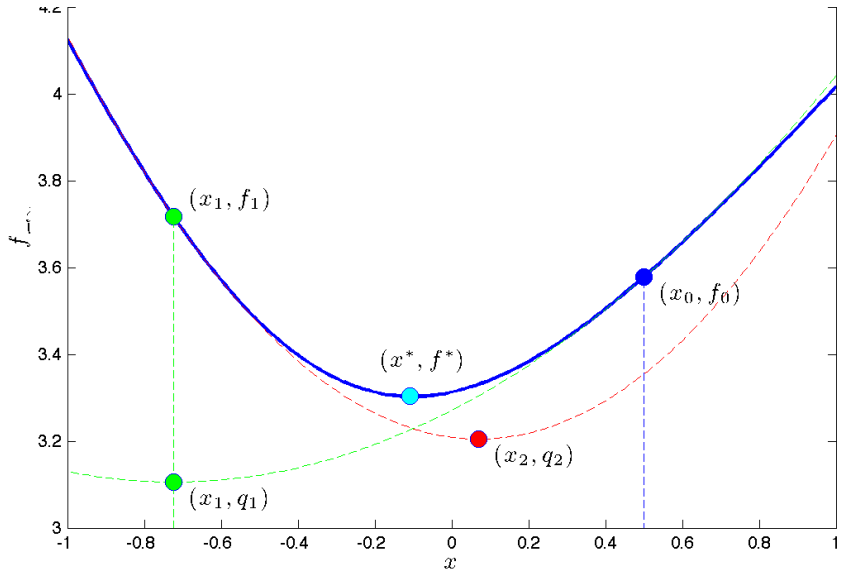


Figure 3.3: Newton's minimization algorithm: In each step, it minimizes the quadratic approximation of the objetive function at the current point (represented by dashed lines).

3.1.2 Minimization algorithms

As was noted in the previous section, for restoring noisy patches we need to solve the convex differentiable problem of (3.3). This is not as good as having an analytical solution like in NLBayes algorithm, but at least we are in a auspicious situation: we can use the fast and widely studied *Newton's minimization algorithm*. Here we will describe briefly the Newton's method for minimization (for more details, see for example [4], [5]).

Newton's algorithm

Suppose we have the following minimization problem:

$$\min_{x \in X} f(x) \tag{3.6}$$

where $f\colon X \subseteq \mathbb{R}^n \to \mathbb{R}$ is a twice differentiable *strictly convex* function, that is, $\nabla^2 f(x)$ positive definite for all $x \in X$. Starting from an initial guess $x_0 \in X$ of the solution, the geometric idea of Newton's Method is to minimize the quadratic approximation $\hat{f}$ of f around x_0 to get a better estimation of the solution (see Figure 3.3). From the Taylor expansion:

$$f(x) \simeq \hat{f}(x) := f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0).$$

From the convexity of $\hat{f}$ (because $\nabla^2 \hat{f}(x) = \nabla^2 f(x_0) \succ 0 \ \forall x \in X$), the minimum is the solution of

$$\nabla \hat{f}(x) = \nabla f(x_0) + \nabla^2 f(x_0)(x - x_0) = 0 \quad \Rightarrow \quad x = x_0 - \nabla^2 f(x_0)^{-1} \nabla f(x_0).$$

The *pure* Newton's minimization method consists in iterating this procedure:

$$\begin{cases} \text{Start with } x_0 \in X \\ x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \end{cases} . \tag{3.7}$$

The quadratic approximation $\hat{f}$ is a good one if we use it in a small region centered in the point in which it was computed. Thus, there are several well known results for the local behavior of Newton's algorithm that ensures quadratic convergence *near* the minimum. This rate of convergence proves that the Newton algorithm is very fast if we start *near the optimum point* x^*:

Proposition 3.3. (Local convergence)
Suppose that f is twice differentiable and the Hessian $\nabla^2 f(x)$ is Lipschitz continuous, that is, it satisfies

$$\exists L > 0 : \ \|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|$$

in a neighborhoodof a solution x^* at which $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive-definite. Then the iteration (3.7) satisfies

- if the starting point x_0 is sufficiently close to x^*, the sequence of iterates converges to x^*,
- the rate of convergence of $\{x_k\}$ is quadratic, and
- the sequence of gradient norms $\{\|\nabla f(x_k)\|\}$ converges Quadratically to zero.

Proof. See [28]. □

The problem is that, in general, it is difficult to predict when we are near the optimum. It can be easily shown that for each $k \geq 0$ the vector

$$h_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

which is the step of the above algorithm is always a descent direction. Indeed:

$$\nabla f(x_k)^T h_k = -\nabla f(x_k)^T \nabla^2 f(x_k)^{-1} \nabla f(x_k) < 0$$

because $\nabla^2 f(x_k)$ (and then $\nabla^2 f(x_k)^{-1}$) is positive-definite, unless $\nabla f(x_k) = 0$ in which case x_k is the global solution of 3.6 (due to strictly convex objetive funtional f). Thus, in the direction of h_k we can decrease the functional f but it may happen that $x_k + h_k$ (the minimum of the quadratic approximation $\hat{f}$) is not better than x_k for f (compare x_0 and x_1 in Figure 3.3). There are several examples that show that Newton's algorithm may fail to converge.

In order to ensure decrease in each iterate of the algorithm, once we have a descent direction h_k we perform a *line search* to find a value $t_k > 0$, called *step length* such that the step $t_k h_k$ satisfies

$$f(x_k + t_k h_k) < f(x_k).$$

A common procedure is to apply a *backtracking line search*, that can help to approach the optimum when we are far from it (see [5] for more detail on the line search procedure). If we impose a stronger regularity in f, we can prove a *global convergence* result that shows that in the last iterations, the line search chooses $t_k = 1$ (full Newton step length) and then the convergence is quadratic when we approach the optimum. In what follows, recall that $A \preceq B$ means that $B - A$ is a positive-semidefinite matrix.

Proposition 3.4. (Global convergence)
If f is twice continuously differentiable and *strongly convex* on a subset S, that is

$$\exists m > 0 : \ mI \preceq \nabla^2 f(x) \quad \forall x \in S \tag{3.8}$$

and the Hessian of f is Lipschitz continuous on S, that is

$$\exists L > 0 : \ \|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\| \quad \forall x, y \in S \tag{3.9}$$

then the Newton algorithm

$$\begin{cases} \text{Start with } x_0 \in X \\ x_{k+1} = x_k - t_k \nabla^2 f(x_k)^{-1} \nabla f(x_k) \end{cases} \tag{3.10}$$

converges for *any* initial guess $x_0 \in S$. Also, there exists numbers η, γ with $0 < \eta \leq m^2/L$ and $\gamma > 0$ such that

- *Damped Newton phase*: If $\|\nabla f(x_k)\| \geq \eta$, then
$$f(x_k) - f(x_{k+1}) \geq \gamma$$
- *Pure Newton phase*: If $\|\nabla f(x_{k_0})\| < \eta$, then the backtracking line search selects $t_k = 1$ for $k \geq k_0$ and the quadratic convergence of Proposition 3.3 is achieved.

Proof. See [5]. □

We also comment that, in order to accelerate the quadratic minimization algorithm (that will be called many times), we can use variants of the Newton's algorithm that approximate the inverse to the Hessian, whose computation is the most costly of the algorithm. So, we have rates of convergence to the optimum similar to the Newton's algorithm with faster computational times. Methods of this type are called *quasi-Newton methods*. Also, the system that computes the Newton step can be solved by applying a Conjugate Gradient method which is also very fast. For a broader study of convex minimization algorithms, see for example [4], [5], [28].

Minimizing QNLBayes functional

Recall that, to restore each patch of the image, we want to solve:

$$\min_P \underbrace{D(P) + R(P)}_{E(P)} \tag{3.11}$$

where

$$\begin{aligned} D(P) &= -\sum_{x\in\Omega_P} \log\left[\Phi\left(\frac{b_x - P_x}{\sigma}\right) - \Phi\left(\frac{a_x - P_x}{\sigma}\right)\right] \\ R(P) &= \frac{1}{2}(P-\overline{P})\Sigma_P^{-1}(P-\overline{P}). \end{aligned} \tag{3.12}$$

If we denote $E(P) = D(P) + R(P)$ and $\Omega_P = \{x_1, \ldots, x_n\}$, direct calculations lead to

$$\begin{aligned} \nabla E(P) &= \left(\frac{e^{-B_i^2/2} - e^{-A_i^2/2}}{\sigma \int_{A_i}^{B_i} e^{-t^2/2}\,dt}\right)_{i=1,\ldots,n} + \Sigma_P^{-1}(P-\overline{P}) \\ \nabla^2 E(P) &= diag(C) + \Sigma_P^{-1} \end{aligned}$$

where

$$\begin{aligned} A_i &= \frac{a_{x_i} - P_{x_i}}{\sigma}, \qquad B_i = \frac{b_{x_i} - P_{x_i}}{\sigma}, \\ C_i &= \frac{(B_i e^{-B_i^2/2} - A_i e^{-A_i^2/2}) \int_{A_i}^{B_i} e^{-t^2/2}\,dt + (e^{-B_i^2/2} - e^{-A_i^2/2})^2}{\left(\sigma \int_{A_i}^{B_i} e^{-t^2/2}\,dt\right)^2} \end{aligned}$$

for $i = 1, \ldots, n$.

Proposition 3.5. The problem (3.11) above has only one minimum that is reached for every initial guess in the segment between $\tilde{P}$ and $\overline{\tilde{P}}$ (in particular, for $(\tilde{P} + \overline{\tilde{P}})/2$), and the Newton method is quadraticaly convergent in a neighborhood of the minimum, as it was established by Proposition 3.4.

Proof. The functional D is positive and convex, and R is a positive-definite (convex) quadratic function, so E is coercive, that is

$$\lim_{\|P\|\to+\infty} E(P) = +\infty$$

which implies in particular that the sublevel set

$$S = \{P \in \mathbb{R}^{t\times t} \colon E(P) \le E_0\}$$

is a compact convex subset, where $E_0 = \max(E(\tilde{P}), E(\overline{\tilde{P}}))$. Thus, the whole segment between $\tilde{P}$ and $\overline{\tilde{P}}$ belong to S. As E is infinitely differentiable, the

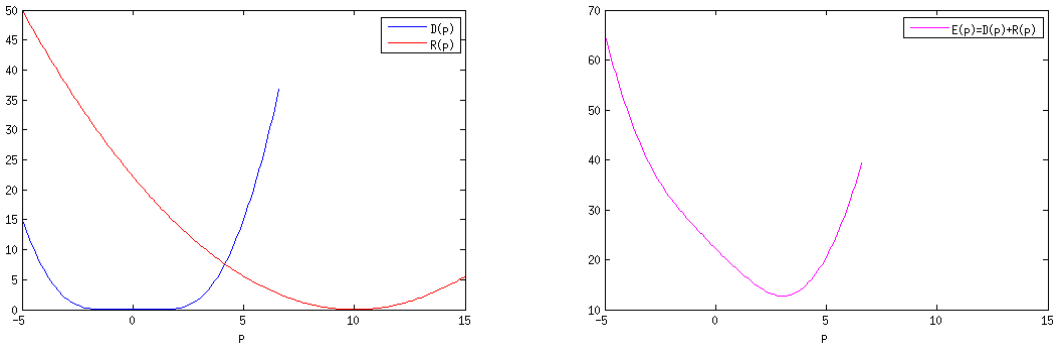


Figure 3.4: Plots of $D(p)$, $R(p)$ y $E(p)$ for $q = 5$, $\tilde{p} = 0$, $[a, b] = [-q/2, q/2]$, $\sigma = 0.5$, $\overline{p} = 2q$, $s = 1.5$.

Lipschitz continuity of (3.9) is also satisfied in S.

Now, if we denote by λ_P the largest eigenvalue of Σ_P, it follows that

$$0 \prec \Sigma_P \preceq \lambda_P I \quad \Rightarrow \quad 0 \prec \frac{1}{\lambda_P} I \preceq \Sigma_P^{-1},$$

that is, R is a strongly convex functional, and hence E is strongly convex too. The proof concludes by applying Proposition 3.4. □

In the next subsection we will show how the QNLBayes algorithm works in a one-dimensional example.

3.1.3 Example in one dimension

Suppose that the patch to be restored consists of only one pixel ($P = p \in \mathbb{R}$), $n \sim N(0, \sigma^2)$ is the associated noise and $[a, b]$ is the quantization interval such that $p + n \in [a, b]$. Thus

$$P(p_{qn} \,|\, p) = P(p + n \in [a, b]) = \Phi\left(\frac{b - p}{\sigma}\right) - \Phi\left(\frac{a - p}{\sigma}\right)$$

and hence

$$D(p) = -\log\left[\Phi\left(\frac{b - p}{\sigma}\right) - \Phi\left(\frac{a - p}{\sigma}\right)\right].$$

For the regularization, we assume mean $\overline{p}$ and variance s^2:

$$R(p) = \frac{(p - \overline{p})^2}{2s^2}.$$

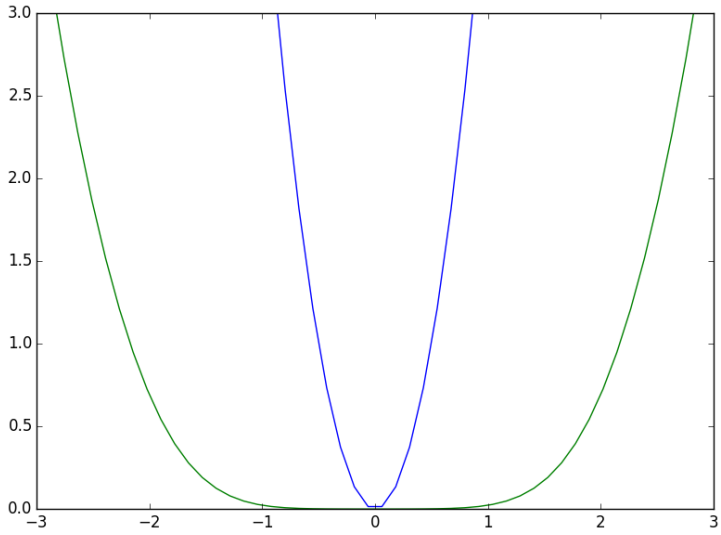


Figure 3.5: Comparision between the data fitting in NLBayes (blue) and QNLBayes (green) for $\tilde{p} = 0$, $[a, b] = [-2, 2]$, $\sigma = 0.5$

Defining $A = \frac{a-p}{\sigma}$ y $B = \frac{b-p}{\sigma}$, the first and second order derivatives of the functional $E(p) = D(p) + R(p)$ are:

$$E'(p) = \frac{e^{-B^2/2} - e^{-A^2/2}}{\sigma \int_A^B e^{-t^2/2}\, dt} + \frac{p - \overline{p}}{s^2}$$

$$E''(p) = \frac{(Be^{-B^2/2} - Ae^{-A^2/2}) \int_A^B e^{-t^2/2}\, dt + (e^{-B^2/2} - e^{-A^2/2})^2}{\left(\sigma \int_A^B e^{-t^2/2}\, dt\right)^2} + \frac{1}{s^2}.$$

In Figure 3.4 are shown the plots of the above functions in the case when $\overline{p}$ doesn't belong to the quantization interval.

The data fitting $D(p)$ of QNLBayes measures the $-\log$ probability that the quantized coefficient $p+n$ belongs to the quantization interval $[a, b]$. So, while $p \in [a, b]$ this functional will be nearly zero (in this case the most probable fact is that $p + n \in [a, b]$, so the probability is near 1), but if $p \notin [a, b]$ the probability of an outlier occurs is small (even more for small values of σ) and then $D(p)$ will be large.

Thus, $D(p)$ permits to choose p almost freely in $[a, b]$ but starts to penalize a lot as we move away from the limits of the quantization interval. In contrast, the data fitting in NLBayes is $\frac{\|P-\tilde{P}\|^2}{\sigma^2}$ and then only considers the noise (not the quantization). A comparison of the two data fitting functionals is shown in Figure 3.5. As we can see, the QNLBayes data fitting functional is more flexible to coefficients belonging to the quantization interval than the one in NLBayes. If we reduce the value of σ, the graph of $D(p)$ in $[a, b]$ becomes

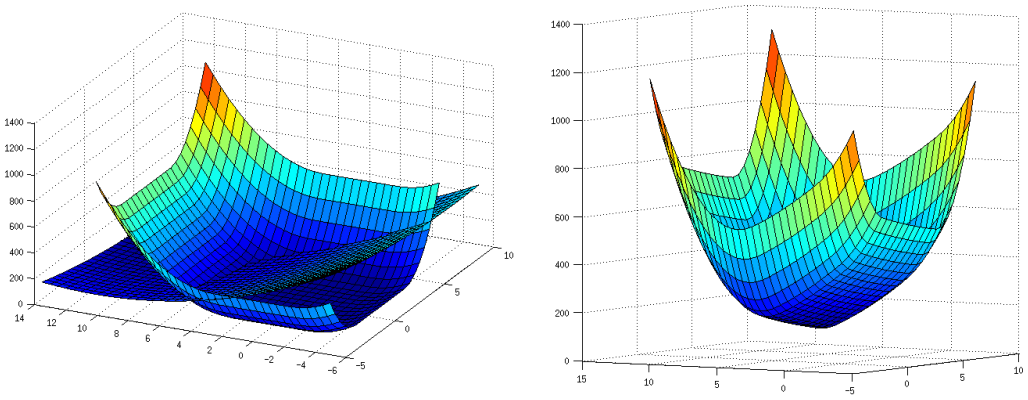


Figure 3.6: Functionals of QNLBayes of a two-dimensional patch. Left: $D(p)$ and $R(p)$. Right: $E(p) = D(p) + R(p)$.

more flat and tends to $+\infty$ more quickly outside $[a, b]$. This is coherent with the fact that, for small values of σ, an occurrence of an outlier is very rare and then the data fitting requires the coefficient to belong to the quantization interval. In Figure 3.6 is shown a two-dimensional example of the QNLBayes functionals.

3.1.4 Comparision with NLBayes

For a fair comparison, we take into account the noise added by the quantization step in the NLBayes algorithm. Given the image formation and compression model described previously, we can model the conditional distribution of the noisy-quantization error by

$$\tilde{P}_x = P_x + n_x$$

where the noise n_x of pixel x follows a distribution $N + U$ with

$$N \sim N(0, \sigma^2), \qquad U \sim U\left[-\frac{q}{2}, \frac{q}{2}\right]$$

which is the sum of a normal and an uniform independent random variable respectively. If q/σ is not too large (which means that the quantization error does not influence too much in the overall error), this distribution can be approximated by a Gaussian of zero mean and variance

$$\sigma_{qn}^2 = \sigma^2 + \frac{q^2}{12} \tag{3.13}$$

(that is, the sum of the variances of N and U). Thus, for the results shown in this subsection, we apply NLBayes with the parameter σ_{qn} given by (3.13).

In Figure 3.7 and 3.8 we can see that QNLBayes is better than NLBayes dealing with quantized images for large values of q/σ. This is not surprising because NLBayes is not designed to denoise quantized images for large values

(a) Original image

(b) Noisy quantized image - PSNR = 28.02 ($q = 5$, $\sigma = 10$)

(c) NLBayes - PSNR = 34.21

(d) QNLBayes - PSNR = 34.09

Figure 3.7: Comparision between NLBayes and QNLBayes restoring noisy quantized Lena with $q = 5$ and $\sigma = 10$ in spatial ($q/\sigma = 0.5$).

(a) Original image

(b) Noisy quantized image - PSNR = 27.95 ($q = 30$, $\sigma = 5$)

(c) NLBayes - PSNR = 32.37

(d) QNLBayes - PSNR = 33.14

Figure 3.8: Comparision between NLBayes and QNLBayes restoring noisy quantized Lena with $q = 30$ and $\sigma = 5$ (in the spatial domain) such that $q/\sigma = 6$.

of q/σ since the noise distribution in this case is far from being Gaussian. As was pointed out in the previous subsection, the QNLBayes algorithm uses the same regularization term as NLBayes (that is, the same Gaussian model for the clusters of similar patches to the one to be denoised) but the data fitting term is more flexible to moving inside the quantization interval, which allows for better estimating the unquantized coefficient.

3.2 Dequantizing in wavelet domain

In this last section we will incorporate the dequantization ingredient into the WNLB algorithm in order to obtain the denoising/dequantizing algorithm in the wavelet domain, which is natural considering the image formation model presented in Chapter 1.

3.2.1 Wavelet compression schemes

In Chapter 1 we presented a general definition for a quantizer

$$Q\colon \mathbb{R} \to C_Q, \qquad C_Q = \{c_1, c_2, \ldots, c_m\}$$

that maps a high-precision coefficient $x \in \mathbb{R}$ into a representation coefficient $Q(x) \in C_Q$ using fewer bits. Moreover, if most of the inputs symbols are mapped to a small number of c_i, then we can use a variable-length representation of the elements of C_Q in order to encode the most frequent c_i with fewer bits and the other (less frequents) with larger code words, so that the overall length will be smaller than using a fixed codelength words for all the c_i coefficients. This procedure is called *entropy coding*.

Also, we see in Chapter 1 that the wavelet transform has the property of concentrating the energy of the image in a small number of coefficients (in general in the LL subband and in the edge positions in the other subbands), setting the rest of the coefficient to very small values that will be quantized to zero. Thus, we are in the context described above: we will have a large amount of wavelet coefficients quantized to a small subset of C_Q (in this case, to zero) and we will produce a very compressed version of the wavelet domain image by means of a entropy coding procedure. The entropy coding is a lossless compresion procedure and a detailed explanation of this process is out of the scope of this thesis. In what follows we will concentrate in the lossy compresion procedure (the quantization) that is where belongs all the loss of image quality.

Dynamic range expansion

If we take for example an 8-bit (integer) image and perform a decimated wavelet transform (that is, mantaining the same number of coefficients than pixels in the original image), the wavelet coefficients in general will be no

longer integer values, and thus we must represent them (in memory) by floating point values (32 or 64 bits each). This difficulty of having to increase the needed bits for representing the *same* image is called *dynamic range expansion*.

Although we do not want to compress the wavelet coefficients, we must reduce the number of bits used to store them in order to avoid excessively increasing the size of the data (by a factor of 4 to 8 for a 8-bit image). The amount of dynamic range expansion allowed by the CCSDS Recomendation, for an integer image of b_I bits per sample varying between 6 and 16, is

$$b_W = b_I + 5$$

where b_W is the number of bits per coefficient in the wavelet domain. This recommendation is for data *storage* and not for memory computations (unless it is needed due to onboard hardware limitations, for example).

Coefficient quantization

Once we follow a convention to atenuate the dynamic range expansion, we address the problem of *quantization* to represent each coefficient in the wavelet domain with fewer bits than the above discussion. The CCSDS uses the concept of *significant bits* to reduce the size of the representation of the wavelet coefficients. If z is an integer and we denote by

$$z = (b_1 b_2 \dots b_n)_2$$

its binary representation, setting its m least significant bits to zero (where $0 \leq m \leq n$) means that we have a quantized version z_m of z by

$$z_m = (b_1 \dots b_{n-m} \underbrace{0 \dots 0}_{m \text{ bits}})_2$$

represented with m fewer bits than z. This may be done by means of the quantizer

$$Q_m(z) = \lfloor z/2^m \rfloor 2^m.$$

For example, if we want to represent $z = 213$ which has binary representation $213 = (11010101)_2$ with 3 bits less, then

$$Q_3(213) = \lfloor 213/2^3 \rfloor 2^3 = 208 = (11010000)_2$$

that corresponds to null the last 3 (the least significants) bits.

Bit allocation

Using an orthogonal wavelet transform W, by the Parseval identity

$$\|u_n\|_2 = \|Wu_n\|_2 = \|w_n\|_2.$$

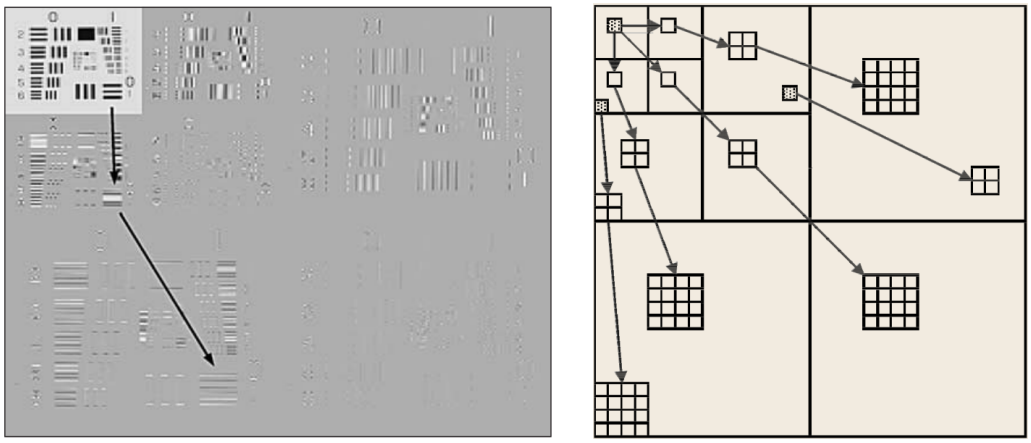


Figure 3.9: Embedded Zero-Tree Wavelet coding. Left: propagation of significant coefficients across frequency bands. Right: example zero-trees that can be defined on the wavelet transform. The roots of the three trees (indicated by shading) originate in the LL_3, LH_3 and HL_2 subbands. (From [36])

That means that the error introduced by the quantization procedure in the spatial domain is the same as that in the wavelet domain. Thus, we can design a quantizer based on considerations made only in wavelet domain. As a result, the coefficients in each subband can be quantized independently of each other, based on the statistics of the subbands, because each subband requires a different amount of bit resources to obtain best coding performance. For example, as has been pointed out, the LL subband contains the majority of the energy of the signal, and hence it is logical to assign more bits to represent that subband than the others.

For a compressed image, the *bit rate* achieved by the compressor, measured in *bits/pixel (bpp)*, is defined as the number of bits used in the compressed representation of the image divided by the number of pixels in the image. Thus, a compression of 0.5 bpp means that we can store an image of n pixels in a compressed image using $n/2$ bits overall (in contrast with $8n$ bits needed to store the gray image in an 8-bit representation). The only issue to be resolved is the number of bits to be assigned to each individual subband to give the best performance. We can think that we have a bit budget for the whole image (given by the compression bit rate) that we must distribute over all the coefficients of the wavelet domain. This is refered as *bit allocation* in the literature.

For example, JPEG2000 standard is based on a method called *Embedded Zero-Tree Wavelet coding* (EZW). As was previously noted, the energy of the signal is concentrated in a small number of wavelet coefficients, and thus coding only these ones with the available bits and setting the rest to zero will lead to a good compression rate with minimum quality distortion. But in this

case we must also code the positions of the non-zero coefficients. Depending on the method used, the amount of resources required to code the position information can be a significant fraction of the total, negating much of the benefit of the energy compaction.

It was found experimentally that zeros could be predicted very accurately across different scales in the wavelet transform. Defining a wavelet coefficient as insignificant with respect to a threshold T if $|x| < T$, the EZW algorithm hypothesized that *"if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then all wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant with respect to T"* [36].

Recognizing that coefficients of the same spatial location and frequency orientation in the wavelet decomposition can be compactly described using tree structures, the EZW called the set of insignificant coefficients, or coefficients that are quantized to zero using threshold T, *zero-trees*. Figure 3.9 shows an example of the correlation between positions of the significant coefficients across scales, and the tree structures generated by the EZW coding. Actually, the JPEG2000 compression scheme is flexible in the choice of the wavelet transform, the specific compression algorithm, and other parameters to be used. For more details about the JPEG2000 compression scheme, we suggest to consult [36].

The CCSDS standard for image compression is a little more specific and its lossy version suggest the usage of the CDF 9/7 transform. In order to minimize the effects of information loss, the image is coded in separated *blocks* that consists in differents regions of the image. To each block is assigned a bit budget to guarantee the overall compression rate. It proposes to code in differents ways the summary and detail coefficients taking into account the nature of each of these subbands. A detailed explanation of the CCSDS standard compressor is out of the scope of this thesis, and we refer the reader to the Green Book of [16].

Some examples

In order to generate compressed noisy images to be restored with the discussed algorithms, here we implement a simple quantizer that was inspired in the above compressors (EZW and CCSDS). It doesn't consider the entropy coding because the latter is a lossless (invertible) process, but uses the notion of assigning more bits to significant coefficients and to null the less significant ones, in a three-level significance scheme. Basically, for a number x, a threshold $T > 0$ and a number of bits b, it does

- If $|x| < T$, then x is considered as insignificant and set to zero.

- If $T \leq |x| < 2T$, then x is considered as less significant and the b bits less significant are set to zero.
- If $|x| \geq 2T$, then x is considered as very significant and it is only rounded to the closest integer.

This quantizer is described in Algorithm 3.

Quantizer with three significance levels

input : Wavelet coefficients w of an image u
input : Threshold T, number of bits b
output: A quantized image $Q(w) = w_q$
output: The corresponding quantization intervals $[a_k, b_k]$ for each $w_q(k)$

for *each* $k \in \Omega_w$ *(wavelet coefficient)* **do**

Quantized coefficient:

$$w_q(k) = \begin{cases} 0 & \text{if } |w(k)| < T \\ \left(\left\lfloor \frac{w(k)}{2^b} \right\rfloor + 0.5\right) 2^b & \text{if } T \leq |w(k)| < 2T \\ \lfloor w(k) \rfloor + 0.5 & \text{if } |w(k)| \geq 2T \end{cases}$$

Quantization interval:

$$[a_k, b_k] = \begin{cases} (-T, T) & \text{if } |w(k)| < T \\ \left[w_q(k) - 2^{b-1}, w_q(k) + 2^{b-1}\right) & \text{if } T \leq |w(k)| < 2T \\ [w_q(k) - 0.5, w_q(k) + 0.5) & \text{if } |w(k)| \geq 2T \end{cases}$$

end

Algorithm 3: Basic quantizer used to test discussed algorithms.

Figure 3.10 shows an example of the compression (using the above algorithm) of a noisy image. Also, as we can see, not all the outliers are removed by NLBayes, in part because there are some of them of similar shapes near each other, so they are considered as part of the texture of the image. These outliers aren't Gaussian noise in spatial domain, so the NLBayes algoritm doesn't work well as its noise model is not satisfied.

3.2.2 Final WNLB algorithm

Recall that now we do not have the noisy coefficients (which is the requeriment for NLBayes to work properly) but instead we have the quantified ones, that is, only a reference of an interval in which this coefficient belongs. No matter what bit allocation procedure was applied in the quantization step of the compression scheme, in what follows we assume that we have access, for each quantized coefficient $w_{qn}(k)$, to the interval $[a_k, b_k]$ to which the unquantized

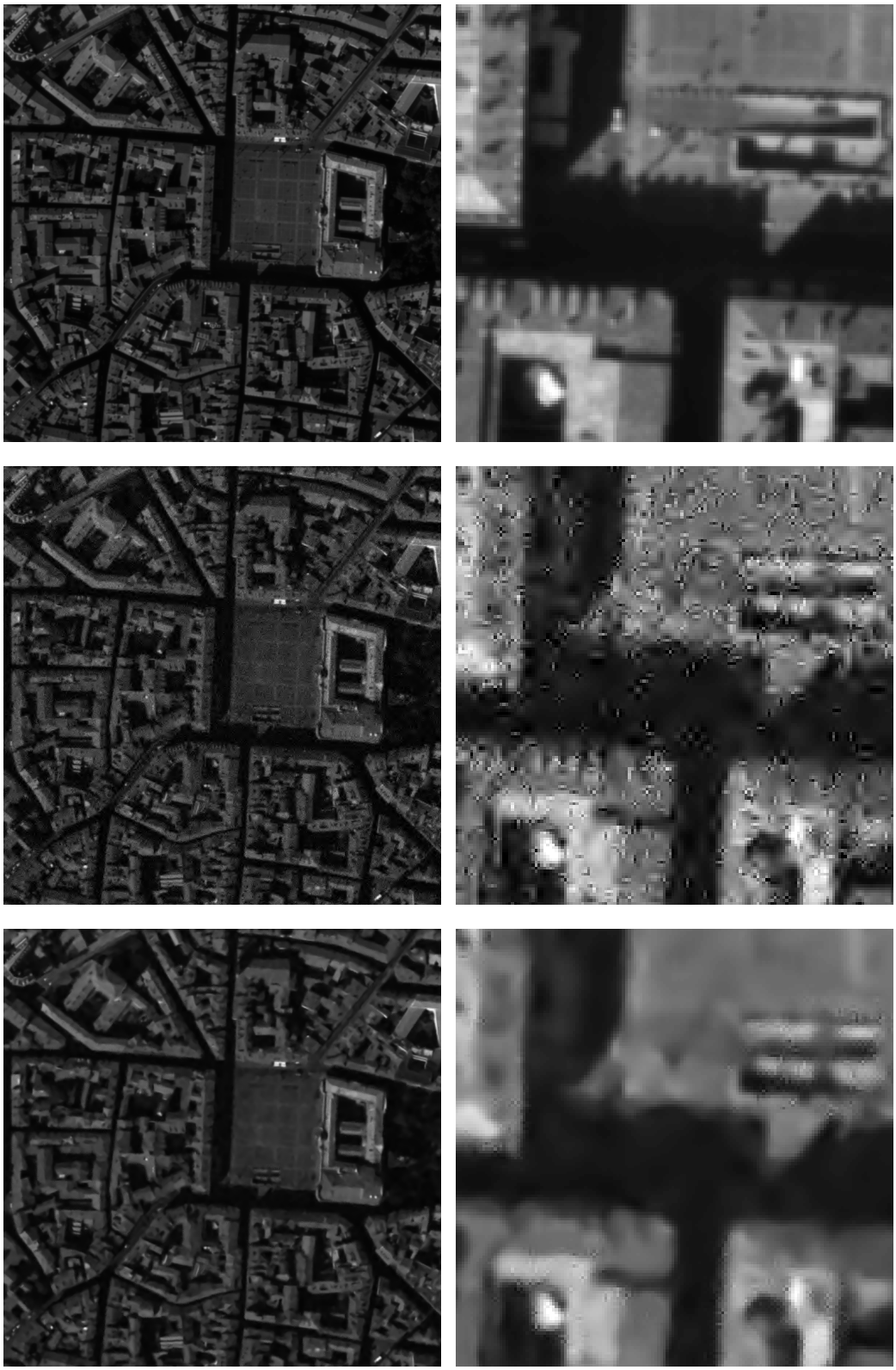


Figure 3.10: From top to bottom: Original satellite *StMichel* image, Compressed noisy image with Algorithm 3 ($\sigma = 15$, $T = 30$, $b = 4$), and NLBayes result (on the left, contrast has been enhanced in order to see in the shadows).

noisy coefficient $w_n(k)$ belongs, as was done in 3.1.1. That general setting permits to apply the resulting algorithm to any coding scheme that uses quantization as part of the compression procedure.

The adaptation of the QNLBayes algorithm presented in the previous subsection to the wavelet domain is straightforward: we can make the changes in a similar manner as was done between NLBayes and WNLBayes in Chapter 2. That is, we take the previous multi-scale WNLBayes algorithm and we replace the way in which we restore each patch (that now consists in quantized noisy *wavelet coefficients* instead of *pixels*).

There are a few modifications to be adressed:

Subbands soft-thresholding

In the quantization process, the small coefficients were set to zero. Thus now it is not necessary to apply a softthresholding to the detail subbands previous to the search of similar patches.

Translated samples

No longer works the generation of traslated samples in the search of similar patches in the cluster estimation, because we do not have access to the uncompressed spatial image by means of the inverse wavelet transform (now the coefficients are quantized). In order to compensate the lack of wavelet patches centered in all pixels due to the subsampling procedure of the wavelet transform, we can use a similar approach to the *Cycle-Spinning* of Coifman and Donoho [11]. This has not yet been done and will be considered in future work (see Chapter 5).

Weighted distances

When we compute distances between patches, we must take into account the noise introduced by the quantization (that now is variable and depends in each coefficient of the patch). Thus, for patches $P, Q \in \mathbb{R}^{4t^2}$ extracted of the wavelet domain, we propose to *standardize* the Euclidean distance, which means that we compute them with

$$d_{qn}(P,Q) = \sum_{k \in \Omega_P} \left[\frac{P(k) - Q(k)}{\sigma_P(k) + \sigma_Q(k)} \right]^2 \tag{3.14}$$

where

$$\sigma_P^2(k) = \sigma^2 + \frac{(b_P(k) - a_P(k))^2}{12} \tag{3.15}$$

is the approximated variance of the noise in the k-th coefficient of P that combines the Gaussian and the quantization noise (here $[a_P, b_P]$ is the quantization interval associated to the patch P) and analogous for $\sigma_Q^2(k)$.

Covariance matrix unbiased estimation

In the correction of the bias introduced by the noise in the covariance matrix, we must add the contribution of the quantization noise. If we assume that the noise (3.15) in each coefficient k of P is independent then the covariance estimate is biased by

$$\Sigma_{\tilde{P}} = \Sigma_P + S_P^2$$

where now the bias is

$$S_P^2 = diag((s_P(k))_k), \qquad s_P(k) = \frac{1}{|N_P|} \sum_{Q \in N_P} \sigma_Q^2(k)$$

(recall that N_P is the set of nearest neighbors of P used for the Gaussian model estimation).

WNLB: Wavelet Non-Local Bayes Restoration Algorithm

input : 3-level quantized wavelet coefficients w_{qn} of noisy image u_n
input : Total noise affecting the image σ, quantization intervals $[a_k, b_k]$ for each $k \in \Omega_w$
output: A restored image $\hat{u}$

\# First step:
for *each patch* $\tilde{P} \in w_{qn}$ *(of size* $t \times t \times 4$*)* **do**
  $N_P = \text{obtain\_nearest\_neighbors}(\tilde{P}, w_{qn})$ (using distances (3.14))
  $(\overline{\tilde{P}}, \Sigma_{\tilde{P}}) = \text{obtain\_Gaussian\_model}(\tilde{P}, N_P)$
  $\hat{P}^{(1)} = \arg\min_P E(P)$ using $\overline{\tilde{P}}, \Sigma_{\tilde{P}}$ and $[a_k, b_k]$
end
for $k \in \Omega_w$ **do**
  $\hat{w_0}(k) = \text{aggregate\_patches}(k)$
end

\# Second step:
for *each patch* $\tilde{P} \in w_{qn}$ **do**
  $N_P^{\hat{w}} = \text{obtain\_nearest\_neighbors}(\tilde{P}, \hat{w_0})$
  $(\overline{\tilde{P}}^{\hat{w}}, \Sigma_{\tilde{P}}^{\hat{w}}) = \text{obtain\_Gaussian\_model}(\tilde{P}, N_P^{\hat{w}})$
  $\hat{P}^{(2)} = \arg\min_P E(P)$ using $\overline{\tilde{P}}^{\hat{w}}, \Sigma_{\tilde{P}}^{\hat{w}}$ and $[a_k, b_k]$
end
for $k \in \Omega_w$ **do**
  $\hat{w}(k) = \text{aggregate\_patches}(k)$
end
$\hat{u} = \text{wavelet\_synthesis}(\hat{w})$

Algorithm 4: Wavelet Non-Local Bayes Restoration Algorithm for compressed noisy images.

We present the final version of WNLB algorithm in Algorithm 4. The results of WNLB in several compressed noisy images are shown in Chapter 5, and comparisons with NLBayes algorithm is also performed.

Chapter 4

Experimental results

In this Chapter we present the results of our denoising/decompression algorithm WNLB as was discussed in the previous chapters. We perform various experiments in order to evaluate the performance of WNLB restoration algorithm in different contexts, and to compare our results with other approaches, we also show the results of NLBayes to the same images. First, we will show the results in a set of experiments using the noise model and compressors studied in this thesis, and later we present some adaptations that can be made in order to test our method on more realistic images.

There exists very little research on noisy decompression algorithms. For this reason we can only compare our algorithm to one of the best available image denoising algorithms like NLBayes, even if it was not designed for compression noise. For compressed noisy images we do not have a noise model in the spatial domain (which was the main reason for considering denoising in the wavelet domain), so the comparisons will be made with the known variance of the noise before quantization.

4.1 Synthetic noise and simplified quantizer

We will perform five different experiments that contemplate the following situations:

- **Experiment 1**: *Barbara* 512×512 with noise $\sigma = 10$, no compression.
- **Experiment 2**: *Lena* 1024×1024 with noise $\sigma = 20$, no compression.
- **Experiment 3**: *Lena* 1024×1024 with noise $\sigma = 20$, compression using Algorithm 3 with $T = 40$ and $b = 3$.
- **Experiment 4**: *St Michel* with noise $\sigma = 15$, compression using Algorithm 3 with $T = 30$ and $b = 4$.
- **Experiment 5**: *St Michel* with noise $\sigma = 8$, compression using uniform quantization with $q = 40$ (thus $q/\sigma = 5$ constant).

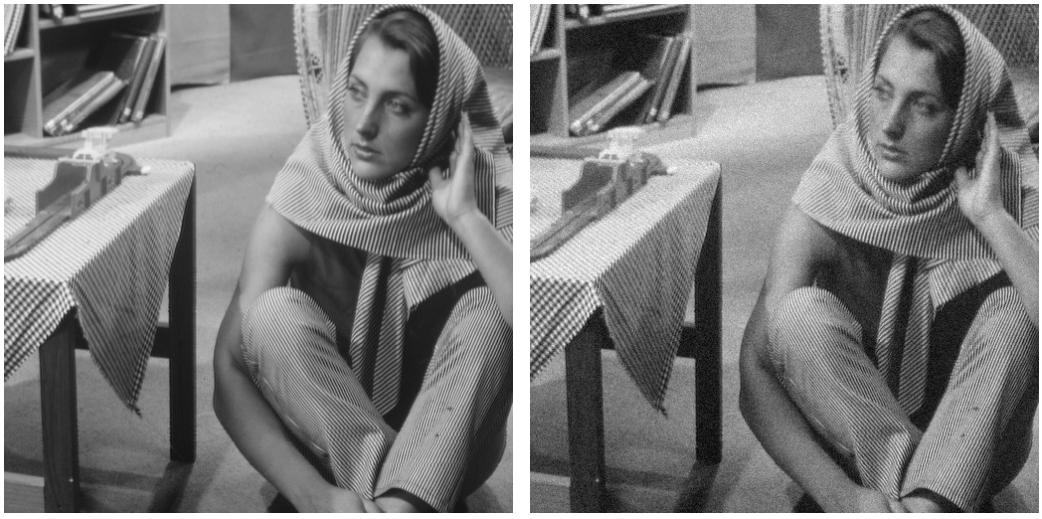
In the first two experiments there is no compression, and in Figures 4.1 to 4.8 the results of NLBayes and WNLBayes denoising in the above experiments are shown. We see that the method it is not so bad in restoring this type of images. In Experiment 1 there is low noise variance and NLBayes is better suited to restore this image. This may happen because the size of the patch in WNLBayes in low-frequency levels corresponds to large patch sizes in spatial domain, which is not the better option as was pointed out in NLBayes [21]. In Experiment 2 we see that, although NLBayes gives better PSNR, WNLBayes can restore certain textures better than NLBayes (see for example the hat in Figure 4.4).

In the last three experiments there are outliers due to the compression procedure. In Experiment 3 and 4 we perform compression with Algorithm 3 that is similar to CCSDS and JPEG2000 compressors. We can see that NLBayes algorithm preserves some of the outliers because they are learnt as part of the Gaussian model prior (mainly when they are mixed with high frequency textures or rare patches). In contrast, WNLBayes is able to remove them, as each outlier corresponds to the noise in only one coefficient in the wavelet domain which is in general decorrelated from the neighboring coefficients (and thus more easily removed).

Finally, in Experiment 5 we compress the satellite image with an uniform quantizer in the wavelet domain of the form (3.2):

$$Q_q(w) = \left\lfloor \frac{w}{q} + 0.5 \right\rfloor q$$

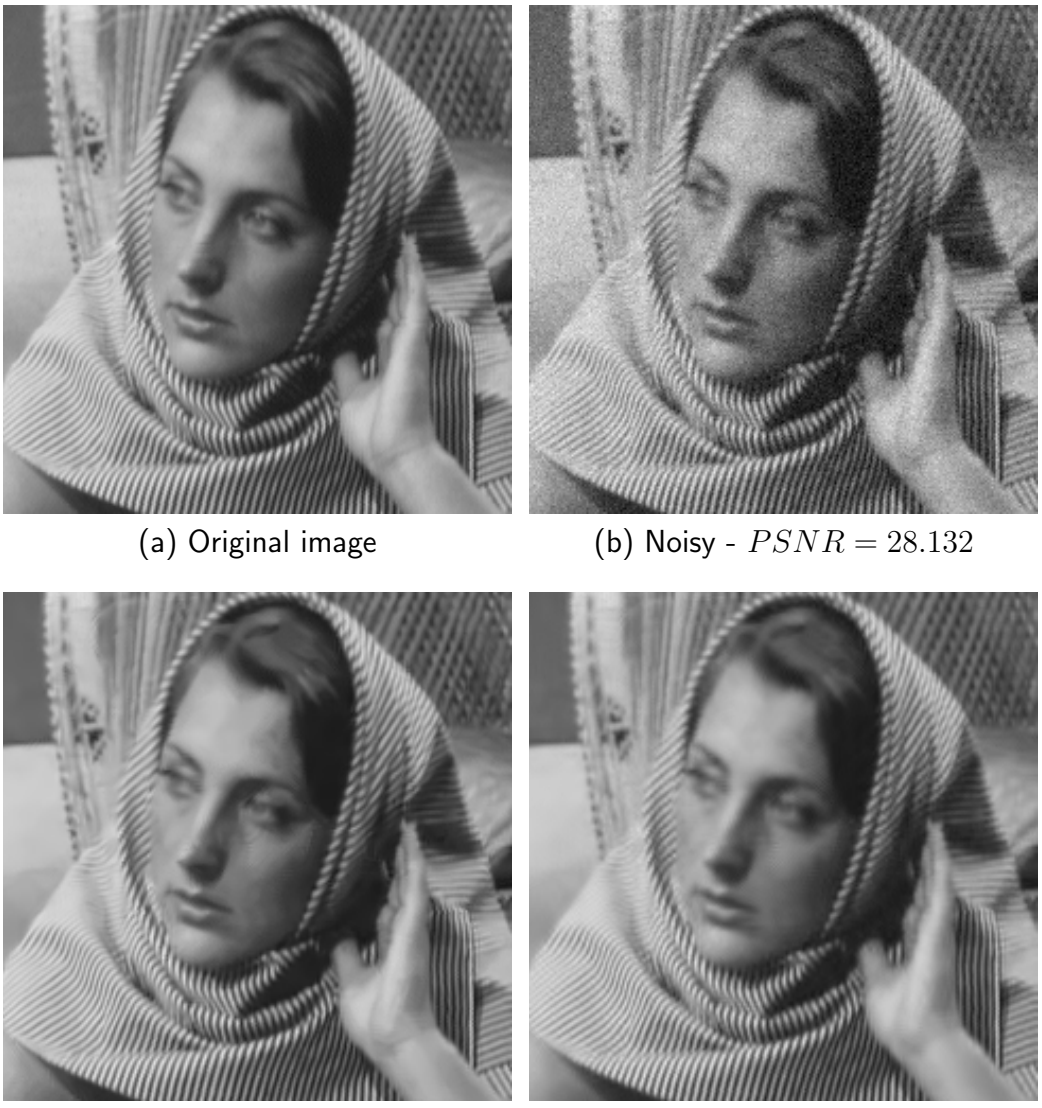
This quantizer is the same to the one used in the spatial domain, and for each $k \in \mathbb{Z}$ it maps the whole interval $[(k-1/2)q, (k+1/2)q)$ to kq. This was done in order to stabilize the value q/σ that now is constant, and thus we can choose a large value of q/σ to show the advantage of our method in these situations, as was pointed out in Subsection 3.1.1. As we can see in this case, although NLBayes is able to remove the outliers, WNLBayes has better visual results as it can better estimate the underlying textures, and also gives better PSNR result.



(a) Original image (b) Noisy - $PSNR = 28.132$

(c) NLBayes - $PSNR = 34.924$ (d) WNLBayes - $PSNR = 34.335$

Figure 4.1: Experiment 1: *Barbara* 512×512 with noise $\sigma = 10$, no compression.



(a) Original image (b) Noisy - $PSNR = 28.132$

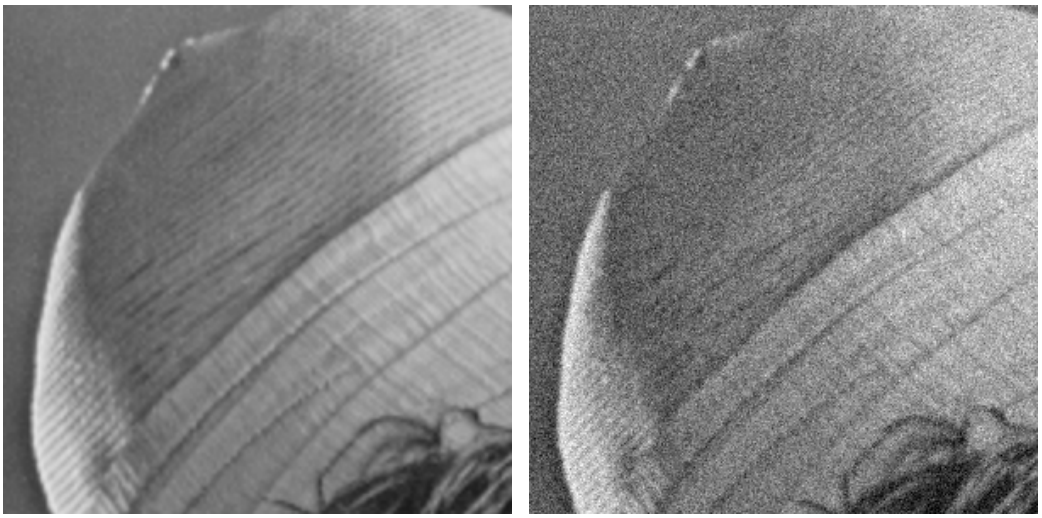
(c) NLBayes - $PSNR = 34.924$ (d) WNLBayes - $PSNR = 34.335$

Figure 4.2: A zoom of the previous image.

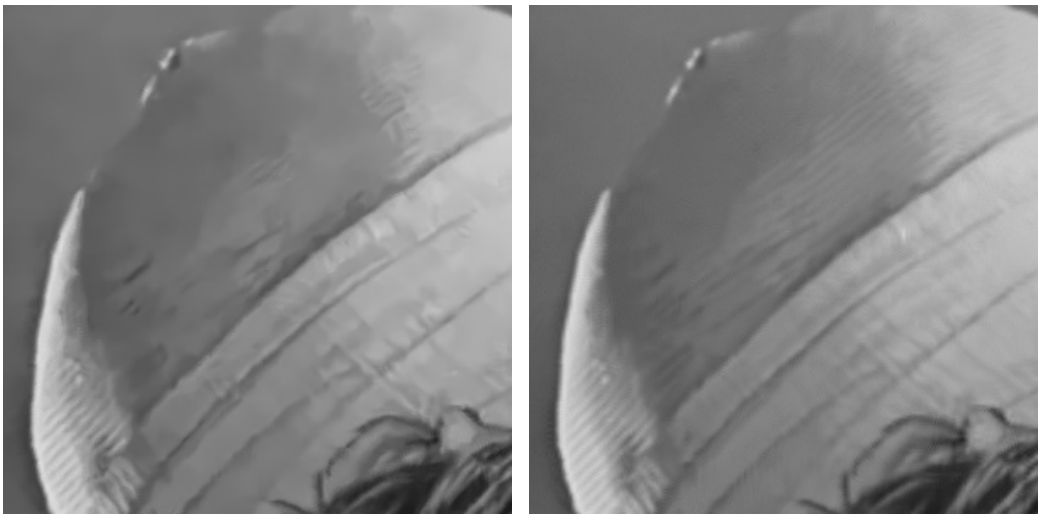
(a) Original image (b) Noisy - $PSNR = 22.144$

(c) NLBayes - $PSNR = 35.406$ (d) WNLBayes - $PSNR = 34.737$

Figure 4.3: Experiment 2: *Lena* 1024×1024 with noise $\sigma = 20$, no compression.



(a) Original image (b) Noisy - $PSNR = 22.144$



(c) NLBayes - $PSNR = 35.406$ (d) WNLBayes - $PSNR = 34.737$

Figure 4.4: A zoom of the previous image.

(a) Original image

(b) Compressed noisy - $PSNR = 26.707$

(c) NLBayes - $PSNR = 34.057$

(d) WNLBayes - $PSNR = 33.950$

Figure 4.5: Experiment 3: *Lena* 1024×1024 with noise $\sigma = 20$, compression using Algorithm 3 with $T = 40$ and $b = 3$.

(a) Original image

(b) Compressed noisy - $PSNR = 26.707$

(c) NLBayes - $PSNR = 34.057$

(d) WNLBayes - $PSNR = 33.950$

Figure 4.6: A zoom of the previous image.

(a) Original image

(b) Compressed noisy - $PSNR = 29.033$

(c) NLBayes - $PSNR = 32.449$

(d) WNLBayes - $PSNR = 32.286$

Figure 4.7: Experiment 4: *St Michel* with noise $\sigma = 15$, compression using Algorithm 3 with $T = 30$ and $b = 4$.

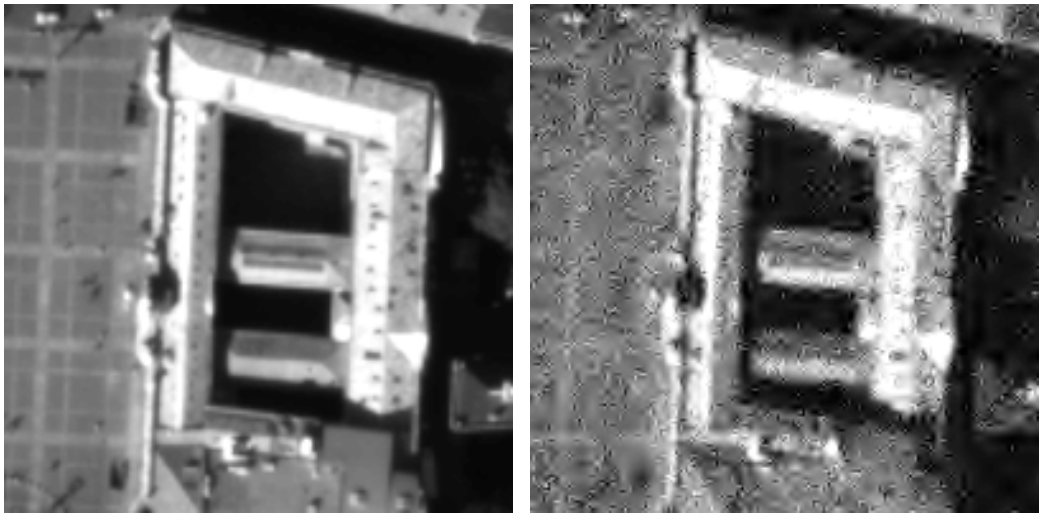
(a) Original image

(b) Compressed noisy - $PSNR = 29.033$

(c) NLBayes - $PSNR = 32.449$

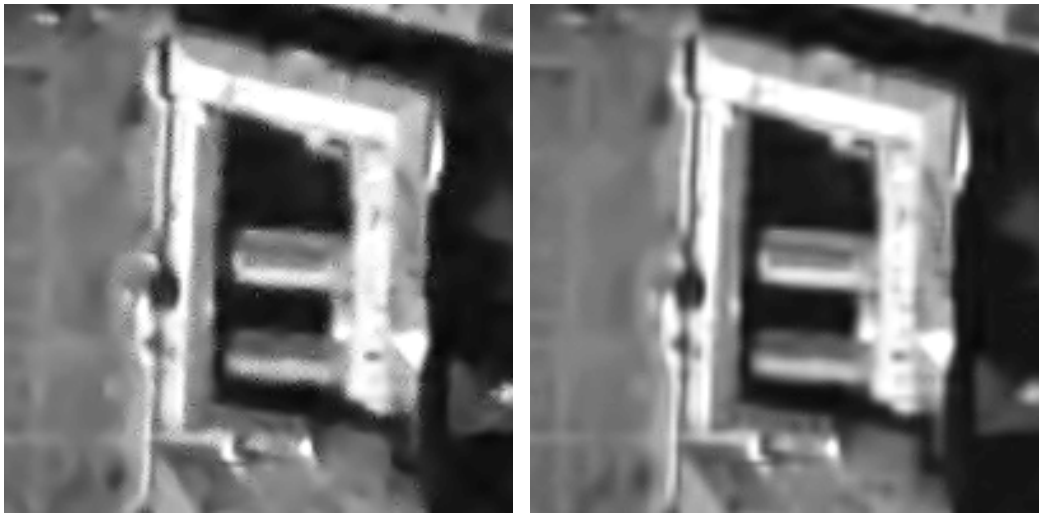
(d) WNLBayes - $PSNR = 32.286$

Figure 4.8: A zoom of the previous image.



(a) Original image

(b) Compressed noisy - $PSNR = 29.033$



(c) NLBayes - $PSNR = 32.449$

(d) WNLBayes - $PSNR = 32.286$

Figure 4.9: Another zoom of the previous image.

(a) Original image

(b) Compressed noisy - $PSNR = 31.444$

(c) NLBayes - $PSNR = 33.911$

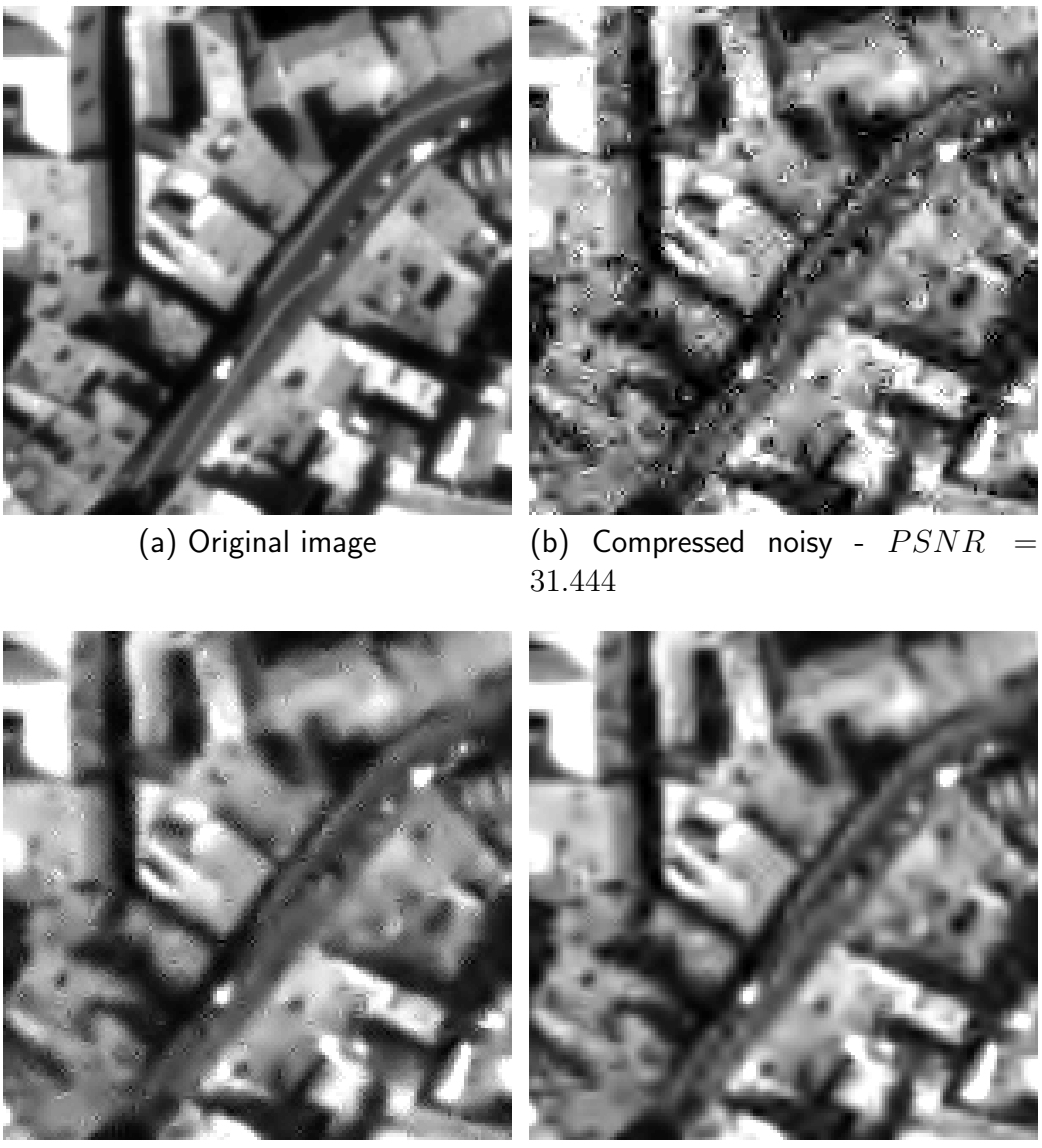
(d) WNLBayes - $PSNR = 35.014$

Figure 4.10: Experiment 5: *St Michel* with noise $\sigma = 8$, compression using uniform quantization with $q = 40$, so $q/\sigma = 5$ is constant.

(a) Original image (b) Compressed noisy - $PSNR = 31.444$

(c) NLBayes - $PSNR = 33.911$ (d) WNLBayes - $PSNR = 35.014$

Figure 4.11: A zoom of the previous image.



(a) Original image

(b) Compressed noisy - $PSNR = 31.444$

(c) NLBayes - $PSNR = 33.911$

(d) WNLBayes - $PSNR = 35.014$

Figure 4.12: Another zoom of the previous image.

4.2 Realistic noise and CCSDS quantizer

In a more realistic application, we must consider another noise model and compression scheme. Instead of a uniform noise, the actual noise model is Gaussian independent in each pixel x of the image with zero mean and variance

$$\sigma^2(x) = c_1^2 + c_2 \times u(x)$$

that depends on the pixel luminance and is therefore spatially variant. Our model performs well in this situation only changing σ by $\sigma(x)$ in the data fitting function (3.12) and in the covariance estimation correction.

The CCSDS compressor, as was discussed in Chapter 3, quantizes the more significant wavelet coefficients using more bits than for less significant coefficients. Given a number of bit-planes $B(k)$, the quantization operator Q applied to a coefficient $w(k)$, returns the same coefficient with all its $B(k)$ least significant bits set to 0, that is

$$Q(w(k)) = \lfloor w(k)/2^{B(k)} \rfloor 2^{B(k)},$$

Note also that the quantization step $q(k) := 2^{B(k)}$ depends on the coefficient itself: to each coefficient corresponds a possibly different quantization step, and this is done in order to ensure a fixed compression ratio. We refer the reader to the Green Book of the CCSDS standard [16].

In order to make a fair comparison, as NLBayes assumes a signal independent noise, we must decorrelate the noise from the signal. The *Variance Stabilization Transform (VST)* is a contrast change transform that is applied to the image in order to make the noise (approximately) stationary. The aim behind the choice of a VST is to find a simple function g to apply to values x in a data set to create new values $y = f(x)$ such that the variability of the values y is not related to their mean value. In our case, if the quantisation effects are not taken into account, the realistic image formation model is such that the acquired image is given by

$$\tilde{u}(x) = u(x) + \tilde{n}(x),$$

where $\tilde{n}$ is a non-uniform Gaussian noise with zero mean and variance $\sigma(x)^2$. Equivalently

$$\tilde{u}(x) = u(x) + \sigma(x)n(x),$$

with $n \sim \mathcal{N}(0, 1)$. It is therefore possible to apply a contrast change g to the acquired image such that the noise variance is nearly constant after contrast change:

$$\text{Var } g(\tilde{u}) - g(u) \approx \text{const.}$$

Using a first order approximation

$$g(\tilde{u}) \approx g(u) + g'(u)\sigma(u)n$$

the noise variance after contrast change can be estimated as

$$\sqrt{\text{Var } g(\tilde{u}) - g(u)} \approx c = g'(u)\sigma(u).$$

Integrating, we obtain:

$$g(u) = \frac{2c}{b}\sqrt{c_1^2 + c_2 u}$$

which gives an image $g(u)$ corrupted with noise of nearly uniform variance c^2.

In Figure 4.13 we can see the results of WNLBayes adapted to realistic noise and compressed with the CCSDS compressor of the above experiment. Also, in Tables 4.1 and 4.2 are shown the quantitative results of NLBayes (with VST) and WNLBayes. As in the previous experiments, we can see that WNL-Bayes is better suited to remove outliers and to restore underlying textures present in the image.

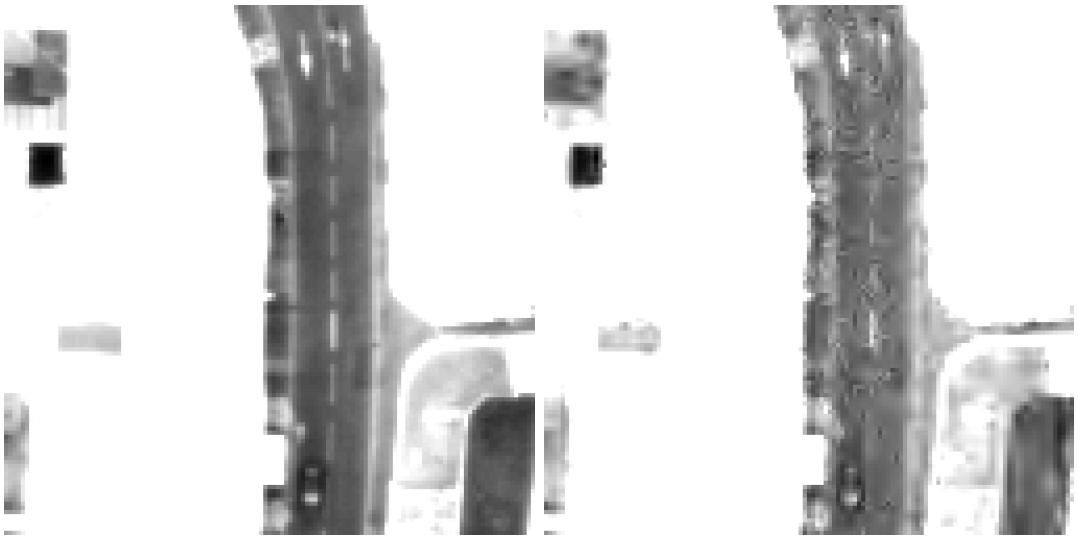
To conclude with this thesis, we mention that all a manuscript was submitted to IEEE Transactions on Computational Imaging, Special Issue: Computational Imaging for Earth Sciences, and it is actually in review process.

Original vs	Noisy	NLBayes	NLBayes/VST	WNLB
Cannes (bpp = 2.0)	13.79	49.66	49.69	**49.83**
St. Mitchel (bpp = 2.77)	50.21	50.43	50.53	**50.78**

Table 4.1: A quantitative analysis of the proposed method for different noise levels and compression ratio. Results are expressed in PSNR. A comparison with NLBayes and NLBayes + VST is also presented.

q/σ range	0.66 - 2.28	2.28 - 2.65	2.65 - 3.19	3.19 - 4.13	4.13 - 9.47
Noisy	50.33	50.20	50.18	50.22	50.13
NLBayes	50.55	50.40	50.40	50.44	50.33
WNLB	50.64	50.52	50.51	50.54	50.46
Increment	0.093	0.11	0.11	0.10	0.13

Table 4.2: Analysis of the performance of the presented method for different q/σ ratios. The table shows the PSNR grouped by quartile. The values correspond to St. Michel Image compressed at 2.7 bpp.



(a) Original Image (b) Noisy and quantised image

(c) Restored image using NLBayes (d) Restored image using WNLB

Figure 4.13: Detail of the results obtained for an image with a noise with local variance and compression ratio of 2 bpp. Image range is saturated in order to view on the dark regions (in this case, the street). Note that NLBayes still presents several outliers (clear on the detail of the street)

Chapter 5

Conclusions and Future Work

Let us summarize the work presented in this thesis:

- In Chapter 1 we presented a brief introduction in image processing tasks as denoising and compression, as well as an introduction to the modern wavelet theory, and then we gave a description of the problem that we want to address.
- In Chapter 2 we reviewed some of the state-of-the-art image denoising algorithm such as NLBayes, and then we discussed how to adapt it to do denoising in the wavelet domain, obtaining, in this way, a denoising algorithm that works in several decomposition levels (frequency subbands) of the wavelet domain.
- In Chapter 3 we gave some details about the CCSDS Recomendation for image compression and the JPEG2000 format, both having very similar compression schemes. Then, we presented an adaptation of Candès' method for dequantization of wavelet coefficients which leads to a convex minimization problem for each patch of the image that can be efficiently solved.
- In Chapter 4 we show several examples that demonstrate that WNLBayes algorithm reaches similar denoising results to the state-of-the-art image methods, which is not surprising because is highly based in one of them: the NLBayes algorithm. We could see that WNLBayes algorithm is better suited for restore compressed noisy images such as highly compressed high-resolution satellite images, as was seen in the performed experiments.

Future work

As was pointed out before, there are several tasks that must be adressed in order to improve the WNLBayes algorithm. Some of these tasks are:

- To exploit the problem structure in the minimization algorithm. For example, some minimization schemes like [27] may be applied, as the funcional $E(P)$ introduced in Section 3.1.1 is of type $E(P) = D(P) + R(P)$ where D is a differentiable convex function and R is *simple*.

- It may be worth to apply the denoising of several translations of the image and to average them to reduce ringing artifacts near edges (pseudo-Gibbs artifacts) as in [11]. The quantization prevents to apply the inverse wavelet transform on the compressed image, so a *frame inversion* must be considered.

- In order to improve the denoising results, an optimal set of the parameters involved in the WNLBayes algorithm must be obtained.

- There still are little pseudo-Gibbs effects that remain due to the fact that the high-frequency coefficients try to compensate the mistakes made on previous levels. Another improvement can be achieved if we take into account intra-scale information in each level denoising.

- The actual implementation of the algorithm was done in Python+Numpy for simplicity. But it is quite slow as the image size grows. So this issue must be resolved before the others.

Appendix A

Appendix

A.1 Wavelet analysis

A.1.1 Orthogonal and biorthogonal wavelet basis

In this subsection we briefly describe how we can construct basis of $L^2(\mathbb{R})$ generated by dilatations and translations of a single function ψ called *wavelet*. There two important types of such a basis: *orthogonal* and *biorthogonal*. Although orthogonality ensures an energy preserving wavelet transform, we must show that sometimes we want other regularity properties which forces us to consider more relaxed assumptions as biorthogonality.

Multiresolution analysis

Based in ideas of Burt and Adelson [9], Mallat [22] and Meyer [24] introduced a concept of *multiresolution analysis* spaces that are very useful for analyzing a signal in a cascade of lower resolution approximations, and also for obtaining other orthonormal basis for $L^2(\mathbb{R})$ with better regularity than Haar basis:

Definition A.1. *Multiresolution Analysis (MRA)*
A sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspaces of $L^2(\mathbb{R})$ is a multiresolution analysis if the following six properties are satisfied:

$$\forall j,k \in \mathbb{Z}, \quad f(t) \in V_j \Leftrightarrow f(t-2^j k) \in V_j \tag{A.1}$$

$$\forall j \in \mathbb{Z}, \quad V_{j+1} \subset V_j \tag{A.2}$$

$$\forall j \in \mathbb{Z}, \quad f(t) \in V_j \Leftrightarrow f(t/2) \in V_{j+1} \tag{A.3}$$

$$\lim_{j\to+\infty} V_j = \cap_{j\in\mathbb{Z}} V_j = \{0\} \tag{A.4}$$

$$\lim_{j\to-\infty} V_j = Closure\,(\cup_{j\in\mathbb{Z}} V_j) = L^2(\mathbb{R}) \tag{A.5}$$

and there exists $\theta \in L^2(\mathbb{R})$ such that $\{\theta(t-k)\}_{k\in\mathbb{Z}}$ is a Riesz basis of V_0.

Let us give an intuitive explanation of these mathematical properties, that will be useful to understand the wavelet tools for later analyzing images. The

scale parameter j is inversely proportional to the resolution level 2^{-j}. (A.1) means that V_j is invariant by any translation proportional to the scale j. The inclusion (A.2) is a property that proves that an approximation at a scale j contains all the necessary information to compute an approximation at a coarser scale $j+1$ (of less resolution). Dilating functions in V_j by 2 enlarges the details by 2 and (A.3) guarantees that it defines an approximation at a coarser scale $j+1$. When the resolution goes to 0, (A.4) implies that we lose all the details of f and

$$\lim_{j \to +\infty} \|P_{V_j} f\| = 0$$

where $P_{V_j} f$ is the projection of f onto the subspace V_j. On the other hand, when the resolution 2^{-j} goes to $+\infty$, property (A.5) imposes that the signal approximation converges to the original signal:

$$\lim_{j \to -\infty} \|f - P_{V_j} f\| = 0$$

Analysis/synthesis basis - Biorthogonal basis

The idea of *Riesz basis* is to loosen up the orthogonality requirement of a basis. The Riesz basis requeriment means that $\{\theta(t-k)\}_{k\in\mathbb{Z}}$ is a family of linearly independent functions such that there exist $0 < A \leq B$ which satisfy

$$\forall f \in V_0, \quad A\|f\|^2 \leq \sum_{k\in\mathbb{Z}} |\langle f, \theta(t-k)\rangle|^2 \leq B\|f\|^2.$$

This energy equivalence guarantees that signal expansions over $\{\theta(t-k)\}_{k\in\mathbb{Z}}$ are numerically stable. Indeed, if two functions f_1, f_2 have very close coefficients measured by $\ell^2(\mathbb{Z})$ metric, which is equivalent to

$$\sum_{k\in\mathbb{Z}} |\langle f_1 - f_2, \theta(t-k)\rangle|^2 \simeq 0$$

then the first inequality of the Riesz basis requeriment implies $\|f_1 - f_2\|^2 \simeq 0$ and then f_1 and f_2 are very close in the $L^2(\mathbb{R})$ metric sense. Also, one may verify that the family

$$\{\theta_{j,k}\}_{j,k\in\mathbb{Z}} = \{2^{-j/2}\theta(2^{-j}t - k)\}_{j,k\in\mathbb{Z}}$$

is a Riesz basis of V_j with the same Riesz bounds A and B at all scales j.

In order to reconstruct a signal $f \in V_j$ through its associated coefficients in a Riesz basis $\{\theta_{j,k}\}_{j,k\in\mathbb{Z}}$, it can be proved (see Appendix **??**) that there exists a *dual basis* $\{\tilde{\theta}_{j,k}\}_{j,k\in\mathbb{Z}}$ such that

$$f = \sum_{k\in\mathbb{Z}} \langle f, \theta_{j,k}\rangle \tilde{\theta}_{j,k} = \sum_{k\in\mathbb{Z}} \langle f, \tilde{\theta}_{j,k}\rangle \theta_{j,k}. \tag{A.6}$$

We call $\{\theta_{j,k}\}_{j,k\in\mathbb{Z}}$ the *analysis basis* and $\{\tilde{\theta}_{j,k}\}_{j,k\in\mathbb{Z}}$ the *synthesis basis*.

Inserting $f = \theta_{j,n}$ in (A.6) we obtain

$$\theta_{j,n} = \sum_{k\in\mathbb{Z}} \langle \theta_{j,n}, \tilde{\theta}_{j,k} \rangle \theta_{j,k}$$

and the linear independence implies that

$$\langle \theta_{j,n}, \tilde{\theta}_{j,k} \rangle = \delta_{kn}.$$

This is known in the literature as a *biorthogonal basis* of V_j.

Orthonormal basis

It can be shown [23] that we can orthogonalize a Riesz basis to produce a function $\phi \in L^2(\mathbb{R})$ such that

$$\{\phi_{j,k}\}_{j,k\in\mathbb{Z}} = \{2^{-j/2}\phi(2^{-j}t - k)\}_{j,k\in\mathbb{Z}}$$

is a orthogonal basis of V_j for all $j \in \mathbb{Z}$. The function ϕ is called the *scaling function* of the MRA. In this case, we have

$$\forall f \in V_j, \quad f = \sum_{k\in\mathbb{Z}} \langle f, \phi_{j,k} \rangle \phi_{j,k}$$

and thus the analysis and synthesis basis are the same. We continue the present discussion using this scaling function for simplicity, but in the next subsection we will justify the importance of having a Riesz basis and its associated dual basis instead of an orthonormal (self-dual) basis.

Examples of multiresolution analysis includes:

- **Piecewise constant approximations.** Space V_j is the set of all $g \in L^2(\mathbb{R})$ such that $g(t)$ is constant for $t \in [k2^j, (k+1)2^j)$ for $k \in \mathbb{Z}$. The approximation at scale j of an $f \in L^2(\mathbb{R})$ is the closest piecewise constant function on intervals of size 2^j. The scaling function can be chosen to be the box window $\phi = 1_{[0,1)}$. The verification of the multiresolution properties is straightforward.

- **Shannon approximations.** Space V_j is defined as the set of functions with a Fourier transform support incluided in $[-2^{-j}\pi, 2^{-j}\pi]$. An orthonormal basis $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ of V_0 can be generated by the *sinc function*:

  $$\phi(t) = \frac{\sin \pi t}{\pi t}.$$

  A plot of the sinc function is shown in Figure A.1. The proof of this fact is related to the Shannon-Whittaker Theorem.

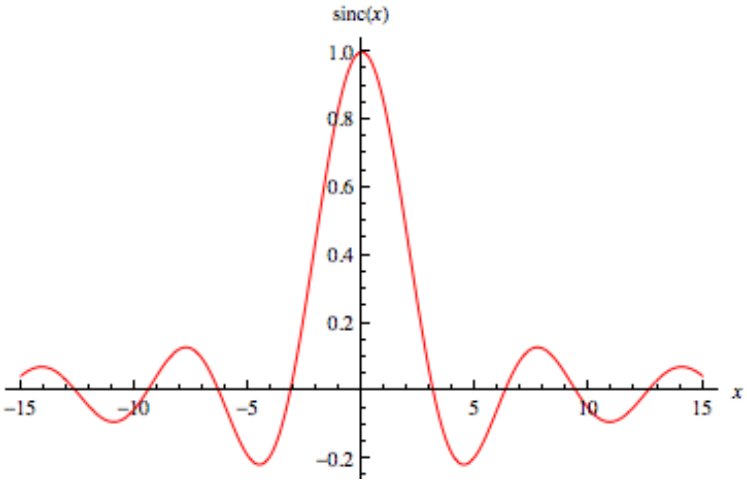


Figure A.1: Sinc funcion $\phi(t) = \frac{\sin \pi t}{\pi t}$. Its integer translations generate an orthonormal basis of the set of functions with frequency support contained in $[-\pi, \pi]$ due to the Shannon-Whittaker Theorem.

Wavelets as high-pass filters

In a MRA, the approximation of f at scales j and $j-1$ are equal to their orthogonal projections on V_j and V_{j-1} respectively, and we know that V_j is included in V_{j-1}. Let W_j be the orthogonal complement of V_j in V_{j-1}:

$$V_{j-1} = V_j \oplus W_j.$$

The orthogonal projection of f on V_{j-1} can be decomposed as the sum of orthogonal projections on V_j and W_j:

$$P_{V_{j-1}} f = P_{V_j} f + P_{W_j} f.$$

Thus, the complement $P_{W_j} f$ provides the "details" of f that appear at the scale $j-1$ but that dissapear at the coarser scale j. In Figure A.2 we can see two level decompositions of a signal using the Haar basis.

Mallat and Meyer [23] proved that we can construct a function that generates each W_j as well as ϕ generates V_j:

Proposition A.2. Given a MRA $\{V_j\}_{j\in\mathbb{Z}}$ with scaling function $\phi \in L^2(\mathbb{R})$, we can construct a function $\psi \in L^2(\mathbb{R})$ such that

$$\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}, \qquad \psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k)$$

is an orthonormal basis of W_j.

The function ψ is called *mother wavelet*. It's easily shown that

$$L^2(\mathbb{R}) = \oplus_{j\in\mathbb{Z}} W_j \quad \Rightarrow \quad \{\psi_{j,k}\}_{j,k\in\mathbb{Z}} \text{ is an orthonormal basis of } L^2(\mathbb{R})$$

Then, using MRA's we can construct new orthonormal basis of $L^2(\mathbb{R})$.

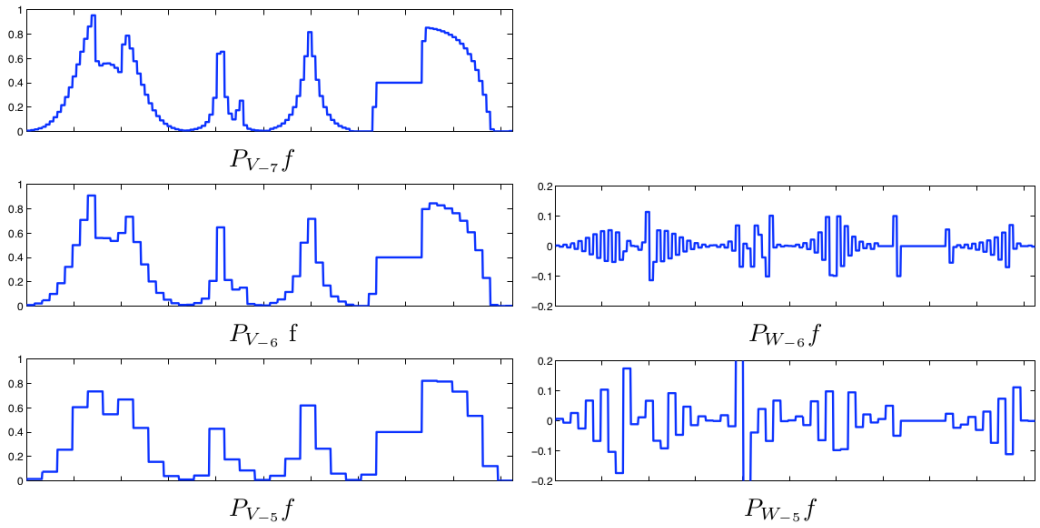


Figure A.2: Cascade approximations of a function f using projections on lower resolution spaces V_j and detail spaces W_j (from [29]).

Daubechies compactly supported wavelets

A important family of orthonormal wavelet basis are due to Daubechies [13]. Daubechies wavelets have a compact support of minimum size for any given number p of vanishing moments, that is

$$\int_{\mathbb{R}} t^k \psi(t)\, dt = 0 \quad \text{for } 0 \leq k < p.$$

This means that ψ is orthogonal to any polynomial of degree $p-1$. The above property implies that, if f is regular, say locally C^k, then f is well approximated in a small interval by a Taylor polynomial of degree k. If $k < p$, then wavelets are orthogonal to this Taylor polynomial and thus produce small-amplitude coefficients $|\langle f, \psi_{j,k}\rangle|$ at fine scales j. In Figure A.3 we can see some of the Daubechies wavelets for differents values of p. Later, we will show that Daubechies wavelets are well suited for analyze images (as well as other regular functions).

A.1.2 Filter banks

We conclude this section presenting the concept of *filter banks* that permits to decompose a finite-length signal with a fast algorithm using discrete convolutions.

Since $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$ and $\{\psi_{j,k}\}_{k\in\mathbb{Z}}$ are orthonormal bases of V_j and W_j respectively, the projection of $f \in L^2(\mathbb{R})$ on these spaces is characterized by

$$a_j(k) = \langle f, \phi_{j,k}\rangle \quad \text{and} \quad d_j(k) = \langle f, \psi_{j,k}\rangle.$$

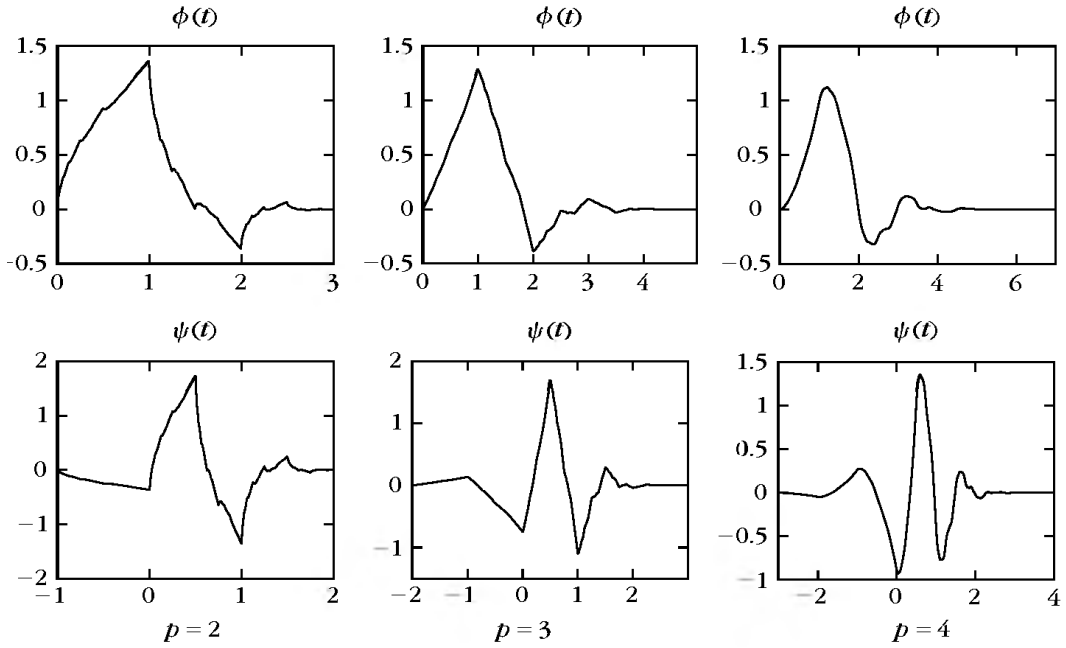


Figure A.3: Daubechies scaling funcion ϕ and wavelet ψ with p vanishing moments.

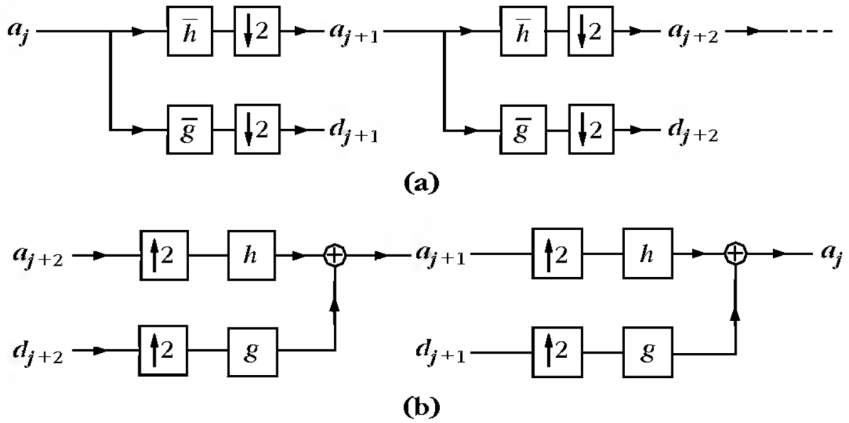


Figure A.4: **(a)** A fast wavelet transform is computed with a cascade of filterings with $\bar{h}$ and $\bar{g}$ followed by a factor 2 subsampling. **(b)** A fast inverse wavelet transform recontructs progressively each a_j by inserting zeroes between samples of a_{j+1} and d_{j+1}, filtering and adding the output.

The next proposition shows that these coefficients are calculated with a cascade of discrete convolutions and subsamplings. We denote $\bar{x}(n) = x(-n)$ and

$$\check{x}(n) = \begin{cases} x(p) & \text{if } n = 2p \\ 0 & \text{if } n = 2p+1 \end{cases}$$

Proposition A.3. *Mallat* [22]
There exists discrete filters $h, g \in \ell^2(\mathbb{R})$ that satisfy:

Decomposition (from V_j to lower resolution V_{j+1}):

$$a_{j+1}(n) = \sum_{k\in\mathbb{Z}} h(k-2n)a_j(n) = a_j \star \bar{h}(2n)$$
$$d_{j+1}(n) = \sum_{k\in\mathbb{Z}} g(k-2n)a_j(n) = a_j \star \bar{g}(2n)$$

Reconstruction (from V_{j+1} to higher resolution V_j):

$$a_j(n) = \sum_{k\in\mathbb{Z}} h(n-2k)a_{j+1}(k) + \sum_{k\in\mathbb{Z}} g(n-2k)d_{j+1}(k) = \check{a}_j \star \overline{h}(n) + \check{d}_j \star \overline{g}(n) \tag{A.7}$$

This means that h is used to project the signal in a lower resolution space at scale $j+1$ and g computes the detail coefficients that are eliminated in this process (and needed for the reconstruction). Thus, the filters h and g act as a low-pass filter and a high-pass filter respectively. The orthonormality of the wavelet basis is reflected in the above filters as

$$\begin{aligned} \sum_{n\in\mathbb{Z}} h(n-2i)h(n-2j) &= \delta_{ij} \\ \sum_{n\in\mathbb{Z}} g(n-2i)g(n-2j) &= \delta_{ij} \\ \sum_{n\in\mathbb{Z}} h(n-2i)g(n-2j) &= 0 \end{aligned} \tag{A.8}$$

To perform decompositions and reconstructions by applying the above convolutions is known as the *Fast Wavelet Transform (FWT)* and the algorithm is $O(N)$, that is, faster than the Fast Fourier Transform which is $O(N \log N)$, where N is the length of the signal [23].

Cohen-Daubechies-Feauveau 9/7 wavelet

In the orthogonal case, we also have filters $\tilde{h}$ and $\tilde{g}$ related to the dual functions $\tilde{\phi}$ and $\tilde{\psi}$ such that the reconstruction in (A.7) is done with these filters instead of h, g. The *Cohen-Daubechies-Feauveau* (CDF) wavelets [10] are the historically first family of biorthogonal wavelets, and the CDF 9/7 is experimentally

Filter Index	h_i	g_i
0	0.852699	0.788486
-1,1	0.377402	0.418092
-2,2	-0.110624	-0.040689
-3,3	-0.023849	-0.064539
-4,4	0.037828	

Table A.1: Cohen-Daubechies-Feauveau (CDF) 9/7 wavelet filters (from [36]).

Parameter	First step	Second step
λ (subbands soft-threshold)	σ	0
t (patch size)	2,3,3	2,3,3
k (# k-NN)	90	90
Search window side	$7t$	$7t$

Table A.2: Parameters used for WNLBayes algorithm (3-level) in each level

the better suited for image analysis [37]. Although this is not a orthogonal wavelet, it is *almost orthogonal* in the sense that the relations (A.8) must be approximately satisfied (see [36]).

A.2 Parameters for WNLBayes algorithm

We summarize in Table A.2 the parameters used in WNLBayes Algorithm 2 for generating the results in this thesis. Although this setting works well for several noise values σ, an optimal set of values must be found in order to optimize the performance of the resulting algorithm.

Bibliography

[1] Francis J Anscombe. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948.

[2] Roger Balian. Un principe d'incertitude fort en théorie du signal ou en mécanique quantique. *CR Acad. Sci. Paris*, 292(2):1357–1361, 1981.

[3] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized PatchMatch correspondence algorithm. In *European Conference on Computer Vision*, September 2010.

[4] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[6] Antoni Buades and Bartomeu Coll. A non-local algorithm for image denoising. In *In CVPR*, pages 60–65, 2005.

[7] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[8] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-Local Means Denoising. *Image Processing On Line*, 1, 2011.

[9] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983.

[10] Albert Cohen, Ingrid Daubechies, and J-C Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45(5):485–560, 1992.

[11] Ronald R Coifman and David L Donoho. *Translation-invariant denoising*. Springer, 1995.

[12] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7):909–996, 1988.

[13] Ingrid Daubechies et al. *Ten lectures on wavelets*, volume 61. SIAM, 1992.

[14] David L Donoho, Iain M Johnstone, Gérard Kerkyacharian, and Dominique Picard. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 301–369, 1995.

[15] Alexei Efros and Thomas Leung. Texture synthesis by non-parametric sampling. In *In International Conference on Computer Vision*, pages 1033–1038, 1999.

[16] The Consultative Committee for Space Data Systems (CCSDS). Technical report, image data compression. 122.0-b-1, 2005. `https://public.ccsds.org/Publications/`.

[17] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of*, 93(26):429–441, 1946.

[18] Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.

[19] David A Huffman et al. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[20] Charles Kervrann. Pewa: Patch-based exponentially weighted aggregation for image denoising. In *Advances in Neural Information Processing Systems*, pages 2150–2158, 2014.

[21] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

[22] Stephane Mallat. Multiresolution approximations and wavelet orthonormal bases of $l^2(r)$. *Transactions of the American mathematical society*, 315(1):69–87, 1989.

[23] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[24] Yves Meyer. *Wavelets and operators*, volume 1. Cambridge university press, 1995.

[25] Yves Meyer. Le traitement du signal et l'analyse mathématique. In *Annales de l'institut Fourier*, volume 50, pages 593–632, 2000.

[26] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.

[27] Yurii Nesterov et al. Gradient methods for minimizing composite objective function. Technical report, UCL, 2007.

[28] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[29] Gabriel Peyré. The numerical tours of signal processing. *Computing in Science & Engineering*, 13(4):94–97, 2011.

[30] Mark A Pinsky. *Introduction to Fourier analysis and wavelets*, volume 102. American Mathematical Soc., 2002.

[31] E. Prestini. *The Evolution of Applied Harmonic Analysis: Models of the Real World*. Applied and Numerical Harmonic Analysis. Springer New York, 2016.

[32] Joseph Salmon and Yann Strozecki. Patch reprojections for non-local methods. *Signal Processing*, 92(2):477 – 489, 2012.

[33] David Salomon. *Data compression: the complete reference*. Springer Science & Business Media, 2004.

[34] Khalid Sayood. *Introduction to data compression*. Newnes, 2012.

[35] Jean-Luc Starck, Fionn D Murtagh, and Albert Bijaoui. *Image processing and data analysis: the multiscale approach*. Cambridge University Press, 1998.

[36] Bryan E. Usevitch. A tutorial on modern lossy wavelet image compression: Foundations of jpeg 2000, 2000.

[37] John D Villasenor, Benjamin Belzer, and Judy Liao. Wavelet filter evaluation for image compression. *IEEE Transactions on image processing*, 4(8):1053–1060, 1995.

[38] Argyrios Zymnis, Stephen Boyd, and Emmanuel Candes. Compressed sensing with quantized measurements. *IEEE Signal Processing Letters*, 17(2):149–152, 2010.