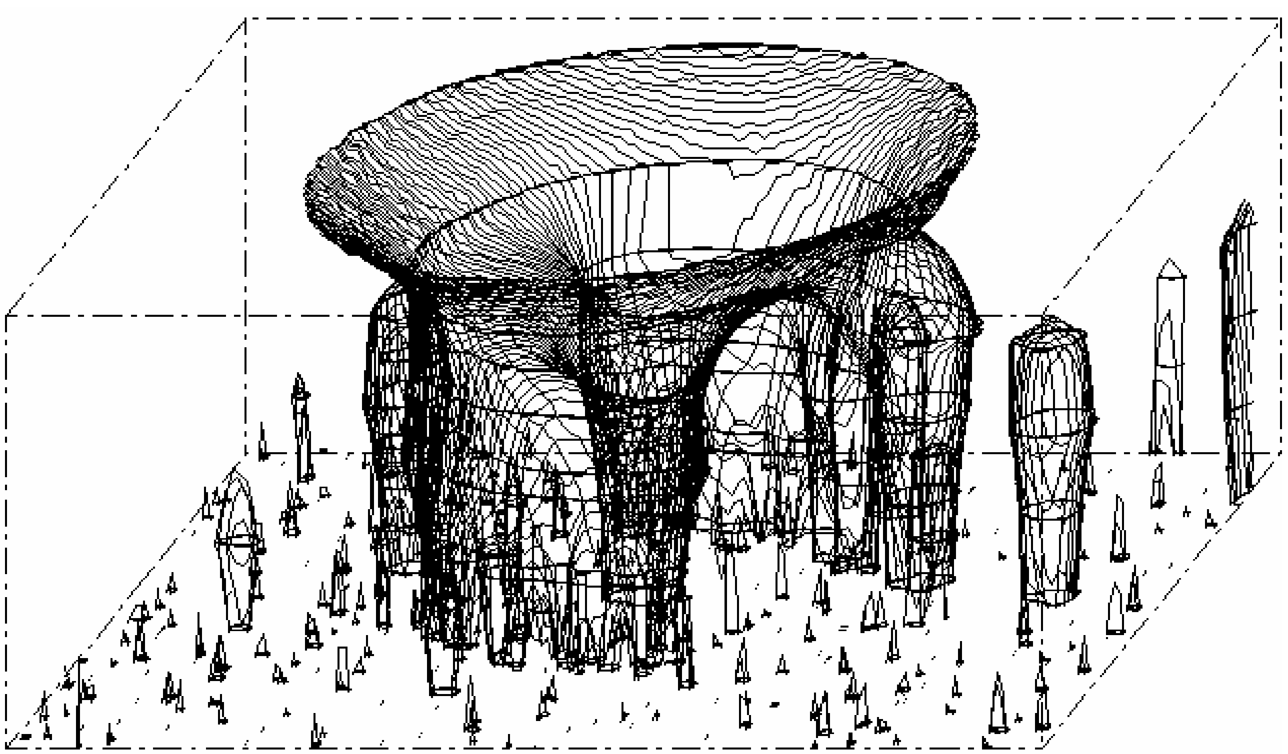
Multiscale Entropy:
Definition and Applications

J.-L. Starck

DAPNIA/SEI-SAP, CEA/Saclay,
91191 Gif sur Yvette, France
jstarck@cea.fr

29 November 1999



Contents

Chapter 1

Multiscale Entropy Theory

1.1 Entropy and image restoration

1.1.1 Introduction

The term "entropy" is due to Clausius (1865), and the concept of entropy was introduced by Boltzmann into statistical mechanics in order to measure the number of microscopic ways that a given macroscopic state can be realized. Shannon (1948) [72] founded the mathematical theory of communication when he suggested that the information gained in a measurement depends on the number of possible outcomes out of which one is realized. Shannon also suggested that the entropy can be used for maximization of the bit transfer rate under a quality constraint. Jaynes (1957) [44] proposed to use the entropy measure for radio interferometric image deconvolution, in order to select between a set of possible solutions that which contains the minimum of information or, following his entropy definition, that which has maximum entropy. In principle, the solution verifying such a condition should be the most reliable. A great deal of work has been carried out in the last 30 years on the use of the entropy for the general problem of data filtering and deconvolution [1, 13, 15, 32, 39, 64, 68, 76, 92, 57, 58].

Traditionally information and entropy are determined from events and the probability of their occurrence. Signal and noise are basic building-blocks of signal and data analysis in the physical sciences. Instead of the probability of an event, we are led to consider the probabilities of our data being either signal or noise.

Observed data Y in the physical sciences are generally corrupted by noise, which is often additive and which follows in many cases a Gaussian distribution, a Poisson distribution, or a combination of both. Other noise models may also be considered. Using Bayes' theorem to evaluate the probability distribution of the realization of the original signal X, knowing the data Y, we have

$$\mathrm{p}(X|Y) = \frac{\mathrm{p}(Y|X).\mathrm{p}(X)}{\mathrm{p}(Y)} \tag{1.1}$$

$\mathrm{p}(Y|X)$ is the conditional probability distribution of getting the data Y given an original signal X, i.e. it represents the distribution of the noise. It is given, in the case of uncorrelated Gaussian noise with variance σ^2, by:

$$\mathrm{p}(Y|X) = \exp\left\{ - \sum_{pixels} \frac{(Y-X)^2}{2\sigma^2} \right\} \tag{1.2}$$

The denominator in equation 1.1 is independent of X and is considered as a constant (stationary noise). $\mathrm{p}(X)$ is the a priori distribution of the solution X. In the absence of any information on the solution X except its positivity, a possible course of action is to derive the probability of X from its entropy, which is defined from information theory.

The main idea of information theory [72] is to establish a relation between the received information and the probability of the observed event [8]. If we note $\mathcal{I}(E)$ the information related to the event E, and p the probability of this event happening, then we consider that

$$\mathcal{I}(E) = f(p) \tag{1.3}$$

Then we assume the two following principles:

- The information is a decreasing function of the probability. This implies that the more information we have, the less will be the probability associated with one event.

- Additivity of the information. If we have two independent events E_1 and E_2, the information $\mathcal{I}(E)$ associated with the happening of both is equal to the addition of the information of each of them.

$$\mathcal{I}(E) = \mathcal{I}(E_1) + \mathcal{I}(E_2) \tag{1.4}$$

Since E_1 (of probability p_1) and E_2 (of probability p_2) are independent, then the probability of both happening is equal to the product of p_1 and p_2. Hence

$$f(p_1 p_2) = f(p_1) + f(p_2) \tag{1.5}$$

Then we can say that the information measure is

$$\mathcal{I}(E) = k \ln(p) \tag{1.6}$$

where k is a constant. Information must be positive, and k is generally fixed at -1.

Another interesting measure is the mean information which is denoted

$$H = -\sum_i p_i \ln(p_i) \tag{1.7}$$

This quantity is called the entropy of the system and was established by Shannon in 1948 [72].

This measure has several properties:

- It is maximal when all events have the same probability $p_i = 1/N_e$ (N_e being the number of events), and is equal to $\ln(N_e)$. It is in this configuration that the system is the most undefined.
- It is minimal when one event is sure. In this case, the system is perfectly known, and no information can be added.
- The entropy is a positive, continuous, and symmetric function.

If we know the entropy H of the solution (the next section describes different ways to calculate it), we derive its probability by

$$\mathrm{p}(X) = \exp(-\alpha H(X)) \tag{1.8}$$

Given the data, the most probable image is obtained by maximizing $\mathrm{p}(X|Y)$. Taking the logarithm of equation 1.1, we thus need to maximize

$$\ln(\mathrm{p}(X|Y)) = -\alpha H(X) + \ln(\mathrm{p}(Y|X)) - \ln(\mathrm{p}(Y)) \tag{1.9}$$

The last term is a constant and can be omitted. Then, in the case of Gaussian noise, the solution is found by minimizing

$$J(X) = \sum_{pixels} \frac{(Y-X)^2}{2\sigma^2} + \alpha H(X) = \frac{\chi^2}{2} + \alpha H(X) \tag{1.10}$$

which is a linear combination of two terms: the entropy of the signal, and a quantity corresponding to χ^2 in statistics measuring the discrepancy between the data and the predictions of the model. α is a parameter that can be viewed alternatively as a Lagrangian parameter or a value fixing the relative weight between the goodness-of-fit and the entropy H.

For the deconvolution problem, the object-data relation is given by the convolution

$$Y = P * X \tag{1.11}$$

where P is the point spread function, and the solution is found (in the case of Gaussian noise) by minimizing

$$J(X) = \sum_{pixels} \frac{(Y - P * X)^2}{2\sigma^2} + \alpha H(X) \qquad (1.12)$$

The way the entropy is defined is fundamental, because from its definition will depend the solution. The next section discusses the different approaches which have been proposed in the past.

1.1.2 The concept of entropy

We wish to estimate an unknown probability density $p(X)$ of the data. Shannon [72], in the framework of the information theory, has defined the entropy of an image X by

$$H_s(X) = -\sum_{k=1}^{N_b} p_k \log p_k \tag{1.13}$$

where $X = \{X_1, ..X_N\}$ is an image containing integer values, N_b is number of possible values which can take a given pixel X_k (256 for a 8 bits image), and p_k values are derived the histogram of X:

$$p_k = \frac{\#X_j = k}{N} \tag{1.14}$$

$\#X_j = k$ giving the number of pixels $X_j = k$.

If the image contains floatting values, it is possible to to build up the histogram L of values L_i, using a suitable interval Δ, counting up how many times m_k each interval $(L_k, L_k + \Delta)$ occurs among the N occurrences. Then the probability that a data value belongs to an interval k is $p_k = \frac{m_k}{N}$, and each data value has a probability p_k.

The entropy is minimum and equal to zero when the signal is flat, and increases when we have some fluctuations. Using this entropy in equation 1.10 leads to minimize:

$$J(X) = \frac{\chi^2}{2} + \alpha H_s(X) \tag{1.15}$$

It is a minimum entropy restoration method.

The trouble with this approach is that, because the number of occurrences is finite, the estimate p_k will be in error by an amount proportional to $m_k^{-\frac{1}{2}}$ [33]. The error becomes significant when m_k is small. Furthermore this kind of entropy definition is not easy to use for signal restoration, because the gradient of equation 1.15 is not easy to compute. For these reasons, other entropy functions are generally used. The main ones are:

- Burg [15]:

$$H_b(X) = -\sum_{k=1}^{N} \ln(X_k) \tag{1.16}$$

- Frieden [32]:

$$H_f(X) = -\sum_{k=1}^{N} X_k \ln(X_k) \tag{1.17}$$

- Gull and Skilling [39]:

$$H_g(X) = \sum_{k=1}^{N} X_k - M_k - X_k \ln(\frac{X_k}{M_k}) \quad (1.18)$$

  where M is a given model, usually taken as a flat image

where N is the number of pixels, and k represents an index pixel.

Each of these entropies can be used, and they correspond to different probability distributions that one can associate with an image [64]. (See [32, 76] for descriptions). The last definition of the entropy has the advantage of having a zero maximum when X equals the model M. All of these entropy measures are negative (if $X_k > 1$), and maximum when the image is flat. They are negative because an offset term is omitted which has no importance for the minimization of the functional. The fact that we consider that a signal has maximum information value when it is flat is evidently a curious way to measure information. A consequence is that we must now maximize the entropy if we want a smooth solution, and the probability of X must be redefined by:

$$\mathrm{p}(X) = \exp(\alpha H(X)) \quad (1.19)$$

The sign has been inverted (see equation 1.8), which is natural if we want the best solution to be the smoothest. These three entropies, above, lead to the Maximum Entropy Method method (MEM), for which the solution is found by minimizing (for Gaussian noise)

$$J(X) = \sum_{k=1}^{N} \frac{(Y_k - X_k)^2}{2\sigma^2} - \alpha H(X) \quad (1.20)$$

These different entropy functions which have been proposed for image restoration have the property of being maximal when the image is flat, and of decreasing when we introduce some information. So minimizing the information is equivalent to maximizing the entropy, and this has led to the well known Maximum Entropy Method (MEM). For the Shannon entropy (which is obtained from the histogram of the data), this is the opposite. The entropy is null for a flat image, and increases when the data contains some information. So, if the Shannon entropy were used for restoration, this would lead to a Minimum Entropy Method.

In 1986, Narayan and Nityanda [64] compared several entropy functions, and finally concluded by saying that all were comparable if they have good properties, i.e. they enforce positivity, and they have a negative second derivative which discourages ripple. They showed also that results varied strongly with the background level, and that these entropy functions produced poor results for negative structures, i.e. structures under the background level (absorption area in an image, absorption band in a spectrum, etc.), and compact structures in the signal. The Gull and Skilling entropy gives rise to the difficulty of estimating a model. Furthermore it has been shown [13] that the solution is dependent on this choice.

The determination of the α parameter is also not an easy task and in fact it is a very serious problem facing the maximum entropy method. In the historic MAXENT algorithm of Skilling and Gull, the choice of α is such that it must satisfy the ad hoc constraint $\chi^2 = N$ when the deconvolution is achieved, N being the number of degrees of freedom of the system i.e. the number of pixels in image deconvolution problems. But this choice systematically leads to an under-fitting of the data [89] which is clearly apparent for imaging problems with little blurring. In reality, the χ^2 statistic is expected to vary in the range $N \pm \sqrt{2N}$ from one data realization to another. In the Quantified Maximum Entropy point of view [76], the optimum value of α is determined by including its probability $\mathrm{P}(\alpha)$ in Bayes' equation and then by maximizing the marginal probability of having α, knowing the data and the model m. In practice, a value of α which is too large gives a resulting image which is too regularized with a large loss of resolution. A value which is too small leads to a poorly regularized solution showing unacceptable artifacts. Taking a flat model of the prior image softens the discontinuities which may appear unacceptable for astronomical images often containing stars and other point-like objects. Therefore the basic maximum entropy method appears to be not very appropriate for this kind of image which contains high and low spatial frequencies at the same time. Another point to be noted is a ringing effect of the maximum entropy method algorithm, producing artifacts around bright sources.

To solve these problems while still using the maximum entropy concept, some enhancements of the maximum entropy method have been proposed. Noticing that neighboring pixels of reconstructed images with MAXENT could have values differing a lot in expected flat regions [18], Gull and Skilling introduced the concepts of hidden image S and intrinsic correlation function C (Gaussian or cubic spline-like) in the Preblur MAXENT algorithm.

The ICF describes a minimum scale length of correlation in the desired image O which is achieved by assuming that

$$O = C * S \tag{1.21}$$

This corresponds to imposing a minimum resolution on the solution O. Since the hidden space image S is not spatially correlated, this can be regularized by the entropy

$$H_g(h) = \sum_{k=1}^{N} S_k - M_k - S_k \ln(\frac{S_k}{M_k}) \tag{1.22}$$

Since in astronomical images many scale lengths are present, the *Multi-channel Maximum Entropy Method*, developed by Weir [91, 92], uses a set of ICFs having different scale lengths, each defining a channel. The visible-space image is now formed by a weighted sum of the visible-space image channels O_j:

$$O = \sum_{j=1}^{N_c} p_j O_j \tag{1.23}$$

where N_c is the number of channels. Like in Preblur MAXENT, each solution O_j is supposed to be the result of the convolution between a hidden image S_j with a low-pass filter (ICF) C_j:

$$O_j = C_j * S_j \tag{1.24}$$

But such a method has several drawbacks:

1. The solution depends on the width of the ICFs [13].
2. There is no rigorous way to fix the weights p_j [13].
3. The computation time increases linearly with the number of pixels.
4. The solution obtained depends on the choice of the models M_j ($j = 1 \ldots N_c$) which were chosen independently of the channel.

In 1993, Bontekoe et al. [13] used a special application of this method which they called Pyramid Maximum Entropy on infrared image data. The pyramidal approach allows the user to have constant ICF width, and the computation time is reduced. It is demonstrated [13] that all weights can be fixed ($p_j = 1$ for each channel).

This method eliminates the first three drawbacks, and gives better reconstruction of the sharp and smooth structures. But in addition to the two last drawbacks, a new one is added: as the images O_j have different sizes (due to the pyramidal approach), the solution O is built by duplicating the pixels of the subimages O_j of each channel. This procedure is known to produce artifacts due to the appearance of high frequencies which are incompatible with the real spectrum of the true image $\hat{O}$.

However this problem can be easily overcome by duplicating the pixels before convolving with the ICF, or expanding the channels using linear interpolation. Thus the introduction of the "pyramid of resolution" has solved some problems and brought lots of improvements to the classic maximum entropy method, but has also raised other questions. In order to derive the model from a physical value, Pantin and Starck [68] introduced the wavelet transform, and defined entropy as follows:

$$H(O) = \frac{1}{\sigma_I^2} \sum_{j=1}^{l} \sum_{k=1}^{N_j} \sigma_j (w_{j,k} - M_{j,k} - |w_{j,k}| \ln \frac{|w_{j,k}|}{M_{j,k}}) \tag{1.25}$$

where σ_I is the noise standard deviation in the data, l is the number of scales, and N_j is the number of samples in the band j ($N_j = N$ for the à trous algorithm). The multiscale entropy is the sum of the entropy at each scale.

The coefficients $w_{j,k}$ are wavelet coefficients, and we take the absolute value of $w_{j,k}$ in this definition because the values of $w_{j,k}$ can be positive or negative, and a negative signal contains also some information in the wavelet transform.

The advantage of such a definition of entropy is the fact we can use previous work concerning the wavelet transform and image restoration [63, 82, 80]. The noise behavior has already been studied in the wavelet transform and we can estimate the standard deviation of the noise σ_j at scale j. These estimates can be naturally introduced in our models m_j

$$M_{j,k} = k_m \sigma_j \tag{1.26}$$

The model M_j at scale j represents the value taken by a wavelet coefficient in the absence of any relevant signal and, in practice, it must be a small value compared to any significant signal value. Following the Gull and Skilling procedure, we take M_j as a fraction of the noise because the value of σ_j can be considered as a sort of physical limit under which a signal cannot be distinguished from the noise ($k_m = \frac{1}{100}$).

1.1.3 Conclusion

As described above, many studies have been carried out in order to improve the functional to be minimized. But the question which should be raised is: what is a good entropy for signal restoration?

Trying to answer, this corresponds to asking what is the information in the signal. We first assume that a signal X can be decomposed in several components:

$$X = S + B + N \tag{1.27}$$

where S is the signal of interest, B lis the background, and N the noise.

The entropy should verify the following criteria [85]:

1. **The information in a flat signal is zero ($S = 0$, $N = 0$ et $B = \mathrm{Cst}$).**

2. **The amount of information in a signal is independent of the background ($H(X)$ is independant of B).**

3. **The amount of information is dependent on the noise ($H(X)$ is dependant of N). A given signal X doesn't furnish the same information if the noise N is high or small.**

4. **The entropy must work in the same way for a pixel which has a value $B + \epsilon$, and for a pixel which has a value $B - \epsilon$. $H(X)$ must be a function of the absolute value of S instead of S.**

5. **The amount of information is dependent on the correlation in the signal. If the signal S presents large features above the noise, it contains a lot of information. By generating a new set of data from S, by randomly taking the pixel values in S, the large features will evidently disappear, and this new signal will contain less information. But the pixel values will be the same as in S.**

Fig. 1.1 illustates the last point perfectly. The second image is obtained by distributing randomly the Saturn image pixel values, and the standard entropy definitions produce the same information measurement for both images. The concept of information becomes really subjective, or at least it depends on the application domain. Indeed, for someone who is not involved in image processing, the second image contains *less* information than the first one. For someone working on image transmission, it is clear that the second image will require more bits for lossless transmission, and from this point of view, he/she will consider that the second image contains *more* information. Finally, for data restoration, all fluctuations due to noise are not of interest, and do not contain relevant information. From this physical point of view, the standard definition of entropy seems badly adapted to information measurement in signal restoration.

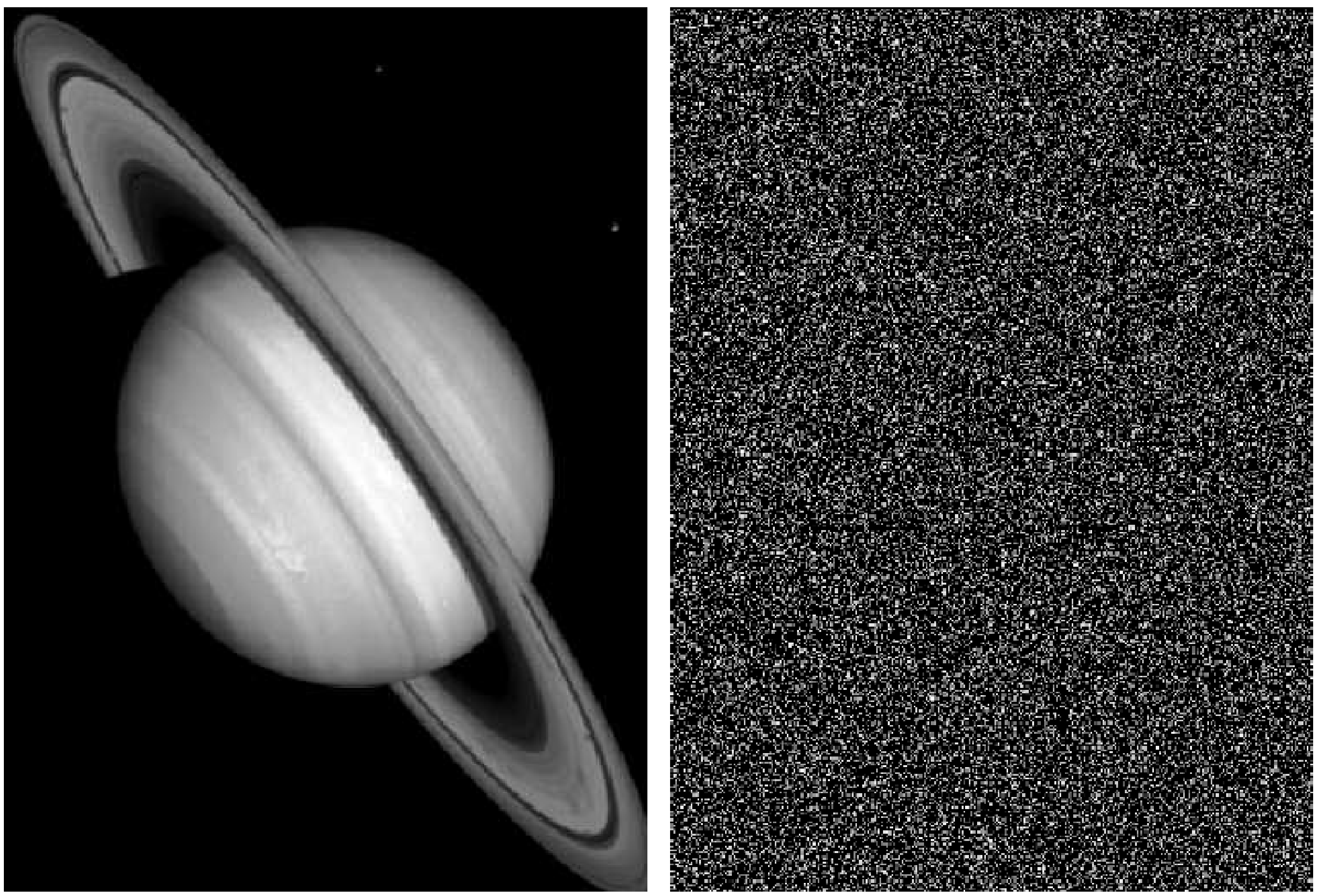


Figure 1.1: Saturn image (left) and the same data distributed differently (right). These two images have the same entropy, using any of the standard entropy definitions.

1.2 Entropy from noise modeling

1.2.1 Information and wavelet coefficient

In the case of signal restoration, the noise is the main problem. This means that we should not consider the probability of appearance of a pixel value in an image, but rather its probability of being due to the signal (or to the noise). If we consider a variable x which follows a probability distribution $p(x)$, we can define the information in x by $-\ln(p(x))$, and a signal S can be considered as a set of individual variables x_k (pixels), each of which follows the same probability distribution. Then the information contained in the data can be measured by $-\sum_{k=1}^{N}\ln(p(x_k))$. If X follows a Gaussian distribution with zero mean, we have

$$H(X) = \sum_{k=1}^{N} \frac{x_k^2}{2\sigma^2} + \mathrm{Cst} \tag{1.28}$$

The energy gives a good measurement of information. But many of the required criteria are not fulfilled by using such an entropy (correlation between pixels, background-independence, etc.). It seems difficult to derive a good probability distribution from the pixel values which fulfill the entropy requirements.

This is not so for transformed data, especially when using the wavelet transform. This has already been done, in fact, for finding threshold levels in filtering methods by means of wavelet coefficient thresholding [81, 60, 30, 46, 19, 3]. Thus we must introduce the concept of multiresolution into our entropy. We will now consider that the information contained in some dataset is the sum of the information at different resolution levels j. The wavelet transform W of a signal by a fast algorithm contains a set of coefficients $w_{j,k}$ (j being the scale index), and a set of coefficients c_k representing the signal at a very low resolution (see [54, 84] for more information about the wavelet transform). If the number of scales is enough large, we can assume that the coefficients c_k furnish information only about the background, and not on the signal of interest. The entropy of X must be measured only from the wavelet coefficients $w_{j,k}$.

Due to the properties of the wavelet transform, the set w_j for all j has a zero mean. From noise modeling, we can derive the probability distribution in the wavelet space of a wavelet coefficient, assuming it is due to the noise. The entropy becomes

$$H(X) = \sum_{j=1}^{l} \sum_{k=1}^{N_j} h(w_{j,k}) \tag{1.29}$$

with $h(w_{j,k}) = -\ln p(w_{j,k})$. We will note in the following h for the entropy (or information) relative to a wavelet coefficient, and H for the multiscale entropy (MSE) of a signal or an image. For Gaussian noise, we get

$$H(X) = \sum_{j=1}^{l} \sum_{k=1}^{N_j} \frac{w_{j,k}^2}{2\sigma_j^2} + \mathrm{Cst} \tag{1.30}$$

where σ_j is the noise at scale j. We see that the information is proportional to the energy of the wavelet coefficients. The higher a wavelet coefficient, then the lower will be the probability, and the higher will be the information furnished by this wavelet coefficient. As the constant has no effect in the solution calculation in restoration problems, we take the liberty to remove it in the following.

We can see easily that this entropy fulfills all the requirements of section 1.1.2. If we consider two signals S_1, S_2, derived from a third one S_0 by adding noise:

$$\begin{aligned} S_1 &= S_0 + N_1(\sigma_1) \\ S_2 &= S_0 + N_2(\sigma_2) \end{aligned} \tag{1.31}$$

then we have:

$$\text{if } \sigma_1 < \sigma_2 \text{ then } H(S_1) > H(S_2) \tag{1.32}$$

and a flat image has zero entropy.

Our entropy definition is completely dependent on the noise modeling. If we consider a signal S, and we assume that the noise is Gaussian, with a standard deviation equal to σ, we won't measure the same information compared to the case when we consider that the noise has another standard deviation value, or if the noise follows another distribution. As for the Shannon entropy, the information increases with the entropy, and using such an entropy leads to a Minimum Entropy Method.

Figure 1.2 shows the information measure at each scale for both the saturn image and its scrambled version. The global information is the addition of the information at each scale. We see that for the scrambled image (dashed curve), the information-versus-scale curve is flat, while for the unscrambled saturn image, it increases with the scale.

1.2.2 Signal information and noise information

In the previous section, we have seen how it was possible to measure the information H related to a wavelet coefficient. Assuming the signal

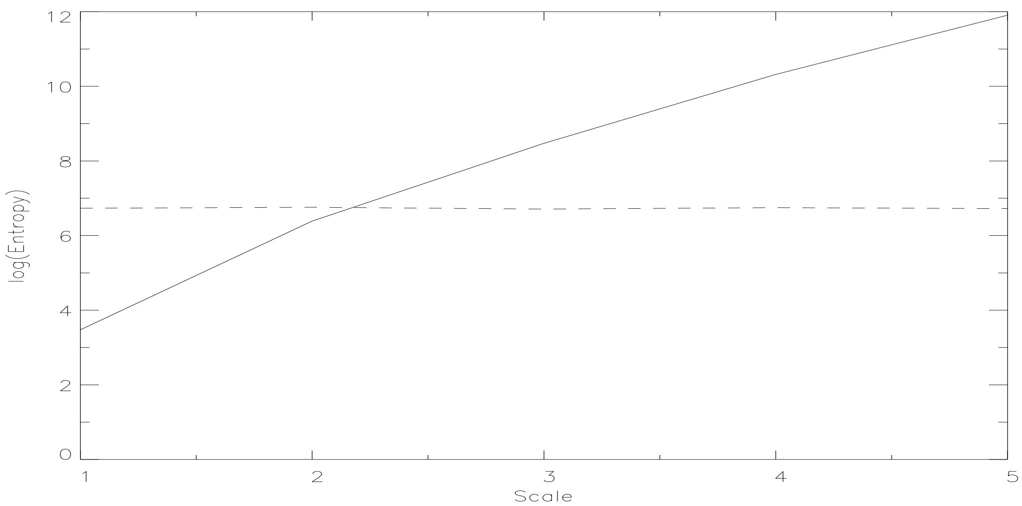


Figure 1.2: Multiscale entropy of the saturn image (continuous curve), and multiscale entropy of the scrambled image (dashed curve).

X is still composed of the three elements S,B,N (X=$S+F+B$, signal of interest, background, and noise), H is independant of B. But some of this information is spurious and undesirable and has been introduced via the noise N. To get the useful information, we must subtract out this spurious portion. Trying to decompose our information measure into two components, one (H_S) corresponding to the non-corrupted part, and another (H_N) to the corrupted part, we have

$$H(X) = H_S(X) + H_N(X) \tag{1.33}$$

We will define in the following H_S as the signal information, and H_N as the noise information. It must be clear that noise does not contain any information, and what we call noise information is a quantity which is measured as information by the multiscale entropy, and which is probably not informative to us.

If a wavelet coefficient is small, its value can be due to noise, and the information h relative to this single wavelet coefficient should be assigned to H_N. If the wavelet coefficient is high, compared to the noise standard deviation, its value cannot be due to the noise, and h should be assigned to H_S. h can be distributed as H_N or H_S based on the probability $P_n(w_{j,k})$ that the wavelet coefficient is due to noise, or the probability $P_s(w_{j,k})$ that it is due to signal. We have $P_s(w_{j,k}) = 1 - P_n(w_{j,k})$. For the Gaussian noise case, we estimate $P_n(w_{j,k})$ that a wavelet coefficient is due to the noise by

$$\begin{aligned} P_n(w_{j,k}) = \text{Prob}(W > \mid w_{j,k} \mid) &= \frac{2}{\sqrt{2\pi}\sigma_j} \int_{\mid w_{j,k} \mid}^{+\infty} \exp(-W^2/2\sigma_j^2) dW \\ &= \text{erfc}(\frac{\mid w_{j,k} \mid}{\sqrt{2}\sigma_j}) \end{aligned}$$

For each wavelet coefficient $w_{j,k}$, we have to estimate now the fractions h_n and h_s of h which should be assigned to H_n and H_s. Hence signal information and noise information are defined by

$$\begin{aligned} H_s(X) &= \sum_{j=1}^{l} \sum_{k=1}^{N_j} h_s(w_{j,k}) \\ H_n(X) &= \sum_{j=1}^{l} \sum_{k=1}^{N_j} h_n(w_{j,k}) \end{aligned} \tag{1.34}$$

Note that $H_s(X) + H_n(X)$ is always equal to $H(X)$.

N1-MSE

A first approach for deriving h_s and h_n from P_s and P_n is to just consider P_s and P_n as weights on the information h. Then we have:

$$\begin{aligned} h_s(w_{j,k}) &= P_s(w_{j,k}) h(w_{j,k}) \\ h_n(w_{j,k}) &= P_n(w_{j,k}) h(w_{j,k}) \end{aligned} \tag{1.35}$$

and the noise and signal information in a signal are

$$\begin{aligned} H_s(X) &= \sum_{j=1}^{l} \sum_{k=1}^{N_j} h_s(w_{j,k}) \\ H_n(X) &= \sum_{j=1}^{l} \sum_{k=1}^{N_j} h_n(w_{j,k}) \end{aligned} \tag{1.36}$$

which leads for a Gaussian noise to:

$$\begin{aligned} H_s(X) &= \sum_{j=1}^{l} \sum_{k=1}^{N_j} \frac{w_{j,k}^2}{2\sigma_j^2} \text{erf}(\frac{\mid w_{j,k} \mid}{\sqrt{2}\sigma_j}) \\ H_n(X) &= \sum_{j=1}^{l} \sum_{k=1}^{N_j} \frac{w_{j,k}^2}{2\sigma_j^2} \text{erfc}(\frac{\mid w_{j,k} \mid}{\sqrt{2}\sigma_j}) \end{aligned} \tag{1.37}$$

We will refer to these functions by the name N1-MSE in the following.

N2-MSE

By the previous entropy measure, information relative to high wavelet coefficients is completely assigned to the signal. For a restoration, this allows us also to exclude wavelet coefficients with high signal-to-noise ratio (SNR) from the regularization. It leads to perfect fit of the solution with the data at scales and positions with high SNR. If we want to consider the information due to noise, even for significant wavelet coefficients, the noise information relative to a wavelet coefficient must be estimated differently. The idea for deriving h_s and h_n is the following: we imagine that the information h relative to a wavelet coefficient is a sum of small information components dh, each of them having a probability of being noise information, or signal information [85]. For example, for two coefficients u and w ($w > u$), with a Gaussian noise ($\sigma = 1$), the information relative to w is $h(w) = w^2$, and by varying u from 0 to w with a step du, the information $h(u)$ increases until it is equal to $h(w)$. When u becomes closer to w, the difference $w-u$ can be due to the noise, and the added information $dh(u) = h(u+du)-h(u)$ is contaminated by the noise. The idea is to weight $dh(u)$ with the probability that $w-u$ is due to the noise. Hence, h_n and h_n are calculated by:

$$h_n(w_{j,k}) = \int_0^{|w_{j,k}|} P_n(\mid w_{j,k} \mid -u)(\frac{\partial h(x)}{\partial x})_{x=u} du \qquad (1.38)$$

is the noise information relative to a single wavelet coefficient, and

$$h_s(w_{j,k}) = \int_0^{|w_{j,k}|} P_s(\mid w_{j,k} \mid -u)(\frac{\partial h(x)}{\partial x})_{x=u} du \qquad (1.39)$$

is the signal information relative to a single wavelet coefficient. For Gaussian noise, we have

$$\begin{aligned} h_n(w_{j,k}) &= \frac{1}{\sigma_j^2} \int_0^{|w_{j,k}|} u \text{ erfc}(\frac{\mid w_{j,k} \mid -u}{\sqrt{2}\sigma_j}) du \\ h_s(w_{j,k}) &= \frac{1}{\sigma_j^2} \int_0^{|w_{j,k}|} u \text{ erf}(\frac{\mid w_{j,k} \mid -u}{\sqrt{2}\sigma_j}) \end{aligned} \qquad (1.40)$$

and the noise and signal information in a signal are

$$\begin{aligned} H_s(X) &= \sum_{j=1}^{l} \sum_{k=1}^{N_j} h_s(w_{j,k}) \\ H_n(X) &= \sum_{j=1}^{l} \sum_{k=1}^{N_j} h_n(w_{j,k}) \end{aligned} \qquad (1.41)$$

We will refer to these functions by the name N2-MSE in the following.

Equations 1.35 and 1.34 lead to two different ways to regularize a signal. The first requires that we use all the information which is furnished in high wavelet coefficients, and leads to an exact preservation of the flux in a structure. If the signal presents high discontinuities, artifacts can appear in the solution due to the fact that the wavelet coefficients located at the discontinuities are not noisy, but have been modified like noise. The second equation doesn't have this drawback, but a part of the flux of a structure (compatible with noise amplitude) can be lost in the restoration process. It is however not as effective as in the standard maximum entropy methods.

LOG-MSE

The multiscale entropy function used in [68] (we call it LOG-MSE in the following) can be considered in our framework if h is defined by:

$$h(w_{j,k}) = \frac{\sigma_j}{\sigma_X^2}[w_{j,k} - M_{j,k} - \mid w_{j,k} \mid \log(\frac{\mid w_{j,k} \mid}{K_m \sigma_j})] \tag{1.42}$$

where σ_X is the noise standard deviation in the data. And h_n is defined by:

$$h_n(w_{j,k}) = A(P_n(w_{j,k}))h(w_{j,k}) \tag{1.43}$$

where A is a function which takes the values 0 or 1 depending on $P_n(w_{j,k})$:

$$A(P_n(w_{j,k})) = \begin{cases} 1 & \text{if } P_n(w_{j,k}) > \epsilon \\ 0 & \text{if } P_n(w_{j,k}) \leq \epsilon \end{cases} \tag{1.44}$$

Wavelet coefficients which are significant will impose $A(P_n(w_{j,k}))$ to be equal to 0 (because their probabilities of being due to noise is very small), and do not contribute to H_n. This means that using H_n in a regularization process will have an effect only on scales and positions where no significant wavelet coefficient is detected.

1.2.3 Conclusion

In practice we prefer N1-MSE and N2-MSE for several reasons. First the way the model is used in equation 1.42 is a bit artificial, and there is an undetermination when the wavelet coefficient is equal to 0. Furthermore, LOG-MSE seems difficult to generalize to other classes of noise, which is not the case for N1-MSE and N2-MSE. N2-MSE has the advantage of estimating the corrupted part in the measured information

h, even for large wavelet coefficients. Concerning the five points, cited in section 1.1.3:

1. *The information in a flat signal is zero*:
   It is true for N1-MSE and N2-MSE. IN the case of LOG-MSE, it is true if we take $\sigma = 0$.

2. *The amount of information in a signal is independent of the background*:
   It is always true because the last scale of the wavelet transform is never taken into account in the entropy calculation.

3. *The amount of information is dependent on the noise*:
   Whatever the method, it is normalized coefficients which are used, so it is always true.

4. *The entropy must work in the same way for a pixel which has a value $B + \epsilon$, and for a pixel which has a value $B - \epsilon$*:
   For N1-MSE and N2-MSE, it is true because the entropy is calculated from the absolute values or the square of the wavelet coefficients. For LOG-MSE, this point is not verified because a $w_{j,k}$ appears. But it has no effect on the solution, because it is the derivated of the entropy which is used in the calculations, and this term becomes constant.

5. *The amount of information is dependent on the correlation in the signal. If the signal S presents large features above the noise, it contains a lot of information*:
   As the entropy is calculated from the wavelet coefficients, this point is always verified.

Chapter 2

Multiscale Entropy Applied to Filtering

2.1 Introduction

The wavelet transform (WT) has been widely used in recent times and furnishes a new approach for describing and modeling data. Using wavelets, a signal can be decomposed into components of different scales. There are many 2D WT algorithms [84]. The most well-known are perhaps the orthogonal wavelet transform proposed by Mallat [55], and its biorthogonal version [22]. These methods are based on the principle of reducing the redundancy of the information in the transformed data. Other WT algorithms exist, such as the Feauveau algorithm [31] (which is an orthogonal transform using an isotropic wavelet), or the à trous algorithm which is non-orthogonal and furnishes a very redundant dataset [41]. All these methods have advantages and drawbacks. Following the content of the data, and the nature of the noise, each can be considered as optimal.

Once the vision model is chosen, the second fundamental point is to estimate the noise behavior in the transformed data. Linear transforms have in this case the advantage of allowing robust estimation of noise variance. But again, different strategies can be employed, which include soft or hard thresholding [30, 28], and in these latter cases threshold level estimation [80, 60, 30, 46, 19, 3, 65, 66, 66].

We review in the second section the algorithms which can be used for a multiresolution decomposition (we call these vision models in the sequel), and which strategies can be used for treating the noise, once the data have been transformed. Then we introduce in section 3 the Multiscale Entropy Filtering method (MEF), and present a large number of examples. Results of a set of simulations are presented and discussed in section 4 in order to compare the MEF method to other standard wavelet-based methods. Finally, we discuss why a single vision model is often not sufficient to describe the data, and how the combination of several vision models can improve the result. This leads to the concept of a multiple vision model.

2.2 Multiresolution and filtering

This section reviews different strategies available for wavelet coefficient filtering. A range of important and widely-used transform and filtering approaches are used.

2.2.1 The choice of the multiresolution transform

- The (bi-) orthogonal wavelet transform.

  This wavelet transform [55], often referred to as the Fast Wavelet Transform (FWT), is certainly the most widely used among available discrete wavelet transform algorithms. It is a non-redundant representation of the information. An introduction to this type of transform can be found in [87, 27].

  An example is the Haar wavelet transform which consists of using a wavelet defined by

$$\begin{array}{ll} \psi(x) = 1 & \text{if } 0 \leq x < \frac{1}{2} \\ \psi(x) = -1 & \text{if } \frac{1}{2} \leq x < 1 \\ \psi(x) = 0 & \text{otherwise} \end{array}$$

  A large class of orthogonal wavelet functions are available.

- The Feauveau wavelet transform.

  Feauveau [31] introduced quincunx analysis based on Adelson's work [2]. This analysis is not dyadic and allows an image decomposition with a resolution factor equal to $\sqrt{2}$. By this method, we have only one wavelet image at each scale, and not three as in the previous method.

- The à trous algorithm [41].

  The wavelet transform of an image by this algorithm produces, at each scale j, a set $\{w_j\}$. This has the same number of pixels as the image. Furthermore, using a wavelet defined as the difference between the scaling functions of two successive scales ($\frac{1}{2}\psi(\frac{x}{2}) =$

$\phi(x) - \phi(\frac{x}{2})$), the original image c_0 can be expressed as the sum of all the wavelet scales and the smoothed array c_l:

$$c_0 = c_l + \sum_{j=1}^{l} w_j \tag{2.1}$$

and a pixel at position x, y can be expressed also as the sum of all the wavelet coefficients at this position, plus the smoothed array:

$$c_{0,k} = c_{l,k} + \sum_{j=1}^{l} w_{j,k} \tag{2.2}$$

- The multiresolution median transform. The median transform is nonlinear, and offers advantages for robust smoothing (i.e. the effects of outlier pixel values are mitigated). The multiresolution median transform [86] (which is not a wavelet transform) consists of a series (c_1, ..., c_p) of smoothings of the input image, with successively broader kernels. Each resolution scale w_j is constructed from differencing two successive smoothed images ($w_j = c_{j-1} - c_j$). For integer input image values, this transform can be carried out in integer arithmetic only which may lead to computational savings. As in the case of the à trous algorithm, the original image can be expressed as a sum of the scales and the smoothed array.

2.2.2 Filtering in wavelet space

We review in this section some important strategies for treating the noise, once the data have been transformed.

Non-Gaussian noise

If the noise in the data I is Poisson, the transformation [5]

$$t(I) = 2\sqrt{I + \frac{3}{8}} \tag{2.3}$$

acts as if the data arose from a Gaussian white noise model, with $\sigma = 1$, under the assumption that the mean value of I is sufficiently large. The arrival of photons, and their expression by electron counts, on CCD detectors may be modeled by a Poisson distribution. In addition, there is additive Gaussian read-out noise. The Anscombe transformation has been extended to take this combined noise into account. The generalization of the variance stabilizing Anscombe formula is derived as

[63]:

$$t(I) = \frac{2}{g}\sqrt{gI + \frac{3}{8}g^2 + \sigma^2 - gm} \tag{2.4}$$

where g is the electronic gain of the detector, σ and m the standard deviation and the mean of the read-out noise.

This implies that for the filtering of an image with Poisson noise or a mixture of Poisson and Gaussian noise, we will first pre-transform the image I into another one $t(I)$ with Gaussian noise. Then $t(I)$ will be filtered, and the filtered image will be inverse-transformed.

For other kinds of noise, modeling must be performed in order to define the noise probability distribution of the wavelet coefficients [84]. In the following, we will consider only stationary Gaussian noise.

Hard thresholding

This consists of setting to 0 all wavelet coefficients which have an absolute value lower than a threshold T_j ($T_j = K\sigma_j$, where j is the scale of the wavelet coefficient, σ_j is the noise standard deviation at the scale j, and K is a constant generally chosen equal to 3). For an energy-normalized wavelet transform algorithm, we have $\sigma_j = \sigma$ for all j.

The appropriate value of σ_j in the succession of wavelet scales is assessed from the standard deviation of the noise σ in the original signal and from study of the noise in the wavelet space. This study consists of simulating a signal containing Gaussian noise with a standard deviation equal to 1, and taking the wavelet transform of this signal. Then we compute the standard deviation σ_j^e at each scale. We get a curve σ_j^e as a function of j, giving the behavior of the noise in the wavelet space. Due to the properties of the wavelet transform, we have $\sigma_j = \sigma\sigma_j^e$ (see [83] for a description of how σ can be automatically calculated directly from the data).

Soft thresholding

Soft thresholding consists of replacing each wavelet coefficient $w_{j,k}$ (j being the scale index, and k the position index) by the value $\tilde{w}_{j,k}$ where

$$\begin{aligned} \tilde{w}_{j,k} &= sgn(w_{j,k})(\mid w_{j,k} \mid -T_j) \text{ if } \mid w_{j,k} \mid \geq T_j & (2.5) \\ &= 0 \text{ otherwise} & (2.6) \end{aligned}$$

Donoho universal approach

Donoho [30, 28] has suggested to take $T_j = \sqrt{2\log(n)}\sigma_j$ (where n is the number of pixels) instead of the standard $K\sigma$ value. This leads to a new soft and hard thresholding approach.

Other threshold-based approaches are available. SURE, Stein unbiased risk estimator ([29]) is adaptive in that it is resolution-dependent. The SURE estimator can break down when the wavelet coefficients are mostly around zero. In contrast, the Donoho *universal* hard and soft thresholding approach may overly smooth the data, which is potentially rectified by the minimax criterion proposed in [30]. Note also that Chipman et al. [19] found that SURE create high frequency artifacts.

Multiresolution Wiener filtering

Multiresolution Wiener filtering [80] consists of multiplying all coefficients $w_{j,k}$ of a given scale j by

$$\alpha_j = \frac{S_j}{S_j + N_j} \tag{2.7}$$

where S_j and N_j are respectively the variance of the signal and of the noise at the scale j ($N_j = \sigma_j^2$). In the absence of any information about the signal, we take S_j equal to the difference between the variance of the data w_j and the variance of the noise N_j.

Hierarchical Wiener filtering

Hierarchical Wiener filtering [80] tries to introduce a prediction $w^h_{j,k}$ into the estimation of $\tilde{w}_{j,k}$.

$$\tilde{w}_{j,k} = \frac{H_j}{N_j + H_j + Q_j} w_{j,k} + \frac{N_j}{N_j + H_j + Q_j} w^h_{j,k} \tag{2.8}$$

with:

$$Q_j = \frac{H_j N_j}{S_j} \tag{2.9}$$

where H_j is the variance of the image D obtained by taking the difference of the scale j and the following one $j+1$ ($D = w_j - w_{j+1}$, and $H_j = \frac{1}{N}\sum_k (D_k - m_D)^2$, where N is the number of pixels and m_D the mean of D). If a pyramidal transform is used, the scale w_{j+1} must be first interpolated to the size of the scale of w_j.

This prediction $w^h_{j,k}$ is obtained from the coefficient at the same position but at the following scale. In the case of the à trous algorithm $w^h_{j,k} = w_{j+1,k}$, while for a pyramidal transform, $w^h_{j,k} = w_{j+1,\frac{k}{2}}$.

Hierarchical hard thresholding

The threshold used here, T_h [80], is equal to $T_j = K\sigma_j$ if $\mid w_{j,k} \mid \geq T_j$, and $T_h = T_j f(\mid \frac{w^h_{j,k}}{\sigma_{j+1}} \mid)$ otherwise. The function $f(a)$ must return a value between 0 and 1. A possible function for f is:

- $f(a) = 0$ if $a \geq k$
- $f(a) = 1 - \frac{1}{K}a$ if $a < K$

If the predicted wavelet coefficient has a high signal to noise ratio (SNR) (this means that there is certainly some information at this position), the threshold level becomes null, and the wavelet coefficient will not be thresholded, even if its value is small. The threshold level becomes adaptive.

Conclusion

In a soft thresholding, we consider that all coefficients must be corrected because there are all noisy. Hard thresholding principle is that wavelet coefficient with high signal-to-noise ratio should not be corrected because we may lose some significative information. Depending on the quality criterion on the solution, we may prefer one or the other thresholding approach. If the visual aspect quality is the main criterion, the soft thresholding is generelly better, because there is less visual artefact. For astronomical images, a soft thresholding should never be used because it leads to a photometry lose of all objects, which can easily be verified by looking to the residual map (i.e. data - filtered data). Concerning the theshold level, Donoho one corresponds to a minimum risk. Larger is the number of pixels, larger is the risk, and it is normal that the threshold depends on the number of pixels. The $k\sigma$ threshold corresponds to a false detection probability, the probability to detect a coefficient as significant when it is due to the noise. The 3σ value corresponds to 0.27 % of false detection. The Multiresolution Wiener filter is derived from the hypothesis that the signal and the noise follow a Gaussian distribution. This hypothesis is in general not true for the signal. The Hierarchical Wiener filtering introduces the concept of interscale dependance between the wavelet coefficients. Indeed, if we find a wavelet coefficient with a high signal-to-noise ratio at a given scale, we will certainly also find one at the following scale. This interscale dependance is actually used by the best image compression methods [73, 71]. The hierarchical threshold level is varying with this interscale dependance. In the following, these methods will be evaluated using two different images.

2.3 Multiscale entropy filtering

2.3.1 Filtering

The problem of filtering or restoring data D can be expressed by the following: We search for a solution $\tilde{D}$ such that the difference between D and $\tilde{D}$ minimizes the information due to the signal, and such that $\tilde{D}$ minimizes the information due to the noise.

$$J(\tilde{D}) = H_s(D - \tilde{D}) + H_n(\tilde{D}) \tag{2.10}$$

Furthermore, the smoothness of the solution can be controlled by adding a parameter:

$$J(\tilde{D}) = H_s(D - \tilde{D}) + \alpha H_n(\tilde{D}) \tag{2.11}$$

In practice [17], we minimize for each wavelet coefficient $w_{j,k}$:

$$j(\tilde{w}_{j,k}) = h_s(w_{j,k} - \tilde{w}_{j,k}) + \alpha h_n(\tilde{w}_{j,k}) \tag{2.12}$$

$j(\tilde{w}_{j,k})$ can be obtained be any minimization routine. In our examples, we have used a simple dichotomy.

Figure 2.1 shows the result when minimizing the functional j with different α values, and a noise standard deviation equal to 1. The corrected wavelet coefficient is plotted versus the wavelet coefficient. From the top curve to the bottom one, α is respectively equal to 0, 0.1, 0.5, 1, 2, 5, 10. The higher the value of α, the more the corrected wavelet coefficient is reduced. When α is equal to 0, there is no regularization and the data are unchanged.

2.3.2 The regularization parameter

The α parameter can be used in different ways:

- It can be fixed to a given value (user parameter): $\alpha = \alpha_u$. This method leads to very fast filtering using the optimization proposed in the following.
- It can be calculated under the constraint that the residual should have some specific characteristic. For instance, in the case of

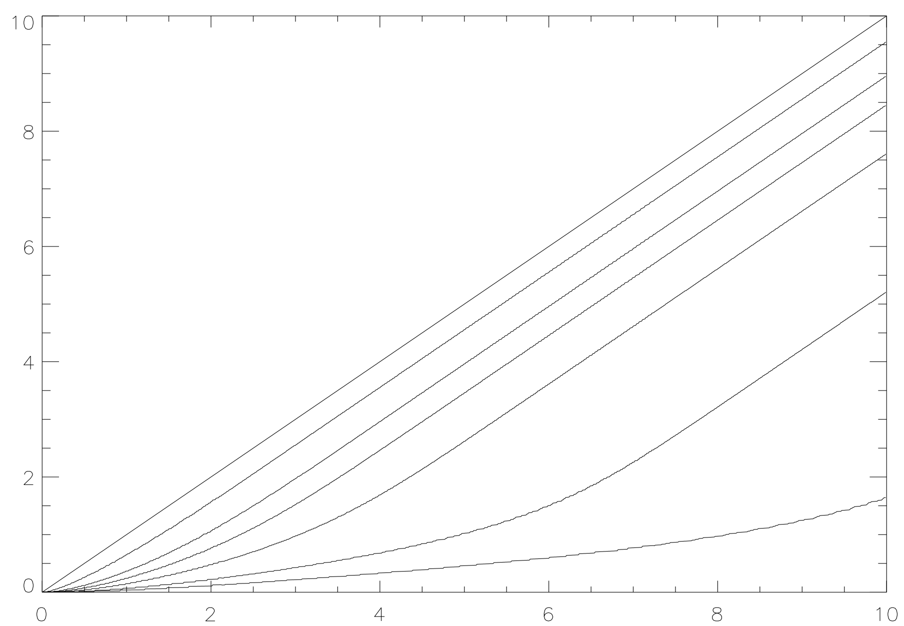


Figure 2.1: Corrected wavelet coefficient versus the wavelet coefficient with different α values (from the top curve to the bottom one, α is respectively equal to 0,0.1,0.5, 1, 2, 5,10).

Gaussian noise, we expect a residual with a standard deviation equal to the noise standard deviation. In this case, $\alpha = \alpha_c \alpha_u$. The parameter finally used is taken as the product of a user parameter (defaulted to 1) and the calculated value α_c. This allows the user to keep open the possibility of introducing an under-smoothing, or an over-smoothing. It is clear that such an algorithm is iterative, and will always take more time than a simple hard thresholding approach.

- We can permit more constraints on α by using the fact that we expect a residual with a given standard deviation at each scale j equal to the noise standard deviation σ_j at the same scale. Then rather than a single α we have an α_j per scale.

A more sophisticated way to fix the α value is to introduce a distribution (or a priori knowledge) of how the regularization should work. For instance, in astronomical image restoration, the analyst generally prefers that the flux (total intensity) contained in a star or in a galaxy is not modified by the restoration process. This means that the residual at positions of astronomical objects will approximately be equal

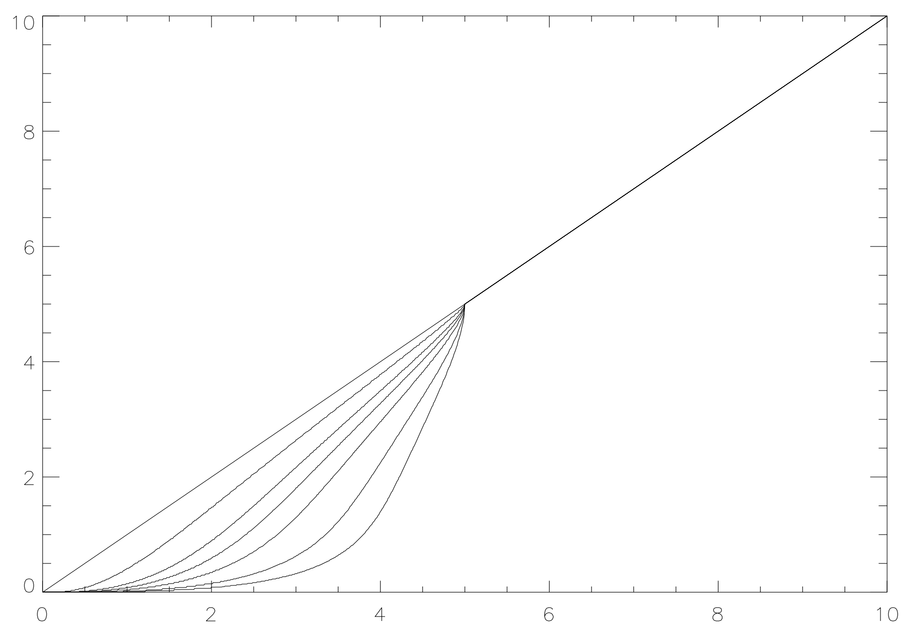


Figure 2.2: Corrected wavelet coefficient versus the wavelet coefficient with different α values.

to zero. All zero areas in the residual map obviously do not relate to realistic noise behavior, but from the user's point of view they are equally important. For the user, all visible objects in the filtered map contain the same flux as in the raw data. In order to obtain this kind of regularization, the α parameter is no longer a constant value, but depends on the raw data. Hence we have one α per wavelet coefficient, which will be denoted $\alpha_s(w_{j,k})$, and it can be derived by

$$\alpha_s(w_{j,k}) = \alpha_j \frac{1 - L(w_{j,k})}{L(w_{j,k})} \tag{2.13}$$

with $L(w_{j,k}) = MIN(1, \frac{|w_{j,k}|}{k_s \sigma_j})$, where k_s is a user parameter (typically defaulted to 3).

When $L(w_{j,k})$ is close to 1, $\alpha_s(w_{j,k})$ becomes equal to zero, and there is no regularization anymore, and the obvious solution is $\tilde{w}_{j,k} = w_{j,k}$. Hence, the wavelet coefficient is preserved from any regularization. If $L(w_{j,k})$ is close to 0, $\alpha_s(w_{j,k})$ tends toward infinity, then the first term in equation (2.12) is negligible, and the solution will be $\tilde{w}_{j,k} = 0$. In practice, this means that all coefficients higher than $k_s\sigma_j$ are untouched as in the hard thresholding approach. We also notice that by considering a distribution $L(w_{j,k})$ equal to 0 or 1 (1 when $\mid w \mid > K\sigma$ for instance), the solution is then the same as a hard thresholding solution.

2.3.3 The use of a model

Using a model in wavelet space has been successfully applied for denoising (see for example [19, 25, 42]). If we have a model D_m for the data, this can also naturally be inserted into the filtering equation:

$$J_m(\tilde{D}) = H_s(D - \tilde{D}) + \alpha H_n(\tilde{D} - D_m) \tag{2.14}$$

or, for each wavelet coefficient $w_{j,k}$:

$$j_m(\tilde{w}_{j,k}) = h_s(w_{j,k} - \tilde{w}_{j,k}) + \alpha h_n(\tilde{w}_{j,k} - w^m_{j,k}) \tag{2.15}$$

where $w^m_{j,k}$ is the corresponding wavelet coefficient of D_m.

The model can be of quite different types. It can be an image, and in this case, the coefficients $w^m_{j,k}$ are obtained by a simple wavelet transform of the model image. It can also be expressed by a distribution or a given function which furnishes a model wavelet coefficient w^m from the data. For instance, the case where we want to keep intact high wavelet coefficients (see equation 2.13) can also be treated by the use of a model, just by calculating $w^m_{j,k}$ by

$$w^m_{j,k} = P_s(w_{j,k}) w_{j,k} \tag{2.16}$$

when $w_{j,k}$ has a high signal to noise ratio, $P_s(w_{j,k})$ is close to 1, and $w^m_{j,k}$ is equal to $w_{j,k}$. Then $\alpha h_n(\tilde{w}_{j,k} - w^m_{j,k})$ is equal to zero and $\tilde{w}_{j,k} = w_{j,k}$, i.e. no regularization is carried out on $w_{j,k}$.

Other models may also be considered. When the image contains contours, it may be interesting to derive the model from the detected edge. Zero-crossing wavelet coefficients indicate where the edges are [56]. By averaging three wavelet coefficients in the direction of the detected edge, we get a value w_a, from which we derive the SNR S_e of the edge ($S_e = 0$ if there is no detected edge). The model value w^m is set to w_a if a contour is detected, and 0 otherwise. This approach has the advantage to filter the wavelet coefficient, and even if an edge is clearly detected the smoothing operates in the direction of the edge.

There is naturally no restriction on the model. When we have a priori information of the content of an image, we should use it in order to improve the quality of the filtering. It is clear that the way we use the knowledge of the presence of edges in an image is not a closed question. The model in the entropy function is an interesting direction to investigate in the future.

2.3.4 The multiscale entropy filtering algorithm

The Multiscale Entropy Filtering algorithm (MEF) consists of minimizing for each wavelet coefficient $w_{j,k}$ at scale j

$$j_m(\tilde{w}_{j,k}) = h_s(w_{j,k} - \tilde{w}_{j,k}) + \alpha_j h_n(\tilde{w}_{j,k} - w^m_{j,k}) \tag{2.17}$$

or

$$j_{ms}(\tilde{w}_{j,k}) = h_s(w_{j,k} - \tilde{w}_{j,k}) + \alpha_j \alpha_s(w_{j,k}) h_n(\tilde{w}_{j,k} - w^m_{j,k}) \tag{2.18}$$

if the SNR is used. By default the model $w^m_{j,k}$ is set to 0. There is no user parameter because the α_j are calculated automatically in order to verify the noise properties. If an over-smoothing (or an under-smoothing) is desired, a user parameter must be introduced. We propose in this case to calculate the α_j in the standard way, and then to multiply the calculated values by a user value α_u defaulted to 1. Increasing α_u will lead to an over-smoothing, while decreasing α_u implies an under-smoothing.

Using a simple dichotomy, the algorithm becomes:

1. Estimate the noise in the data σ (see [67, 83]).
2. Wavelet transform of the data.
3. Calculate from σ the noise standard deviation σ_j at each scale j.
4. Set $\alpha_j^{min} = 0$, $\alpha_j^{max} = 200$.
5. For each scale j do
   (a) Set $\alpha_j = \frac{\alpha_j^{min} + \alpha_j^{max}}{2}$
   (b) For each wavelet coefficient $w_{j,k}$ of scale j, find $\tilde{w}_{j,k}$ by minimizing $j_m(\tilde{w}_{j,k})$ or $j_{ms}(\tilde{w}_{j,k})$
   (c) Calculate the standard deviation of the residual: $\sigma_j^r = \sqrt{\frac{1}{N_j} \sum_{k=1}^{N_j} (w_{j,k} - \tilde{w}_{j,k})^2}$
   (d) If $\sigma_j^r > \sigma_j$ then the regularization is too strong, and we set α_j^{max} to α_j, otherwise we set α_j^{min} to α_j (σ_j is derived from the method described in section 2.2.2).
6. If $\alpha_j^{max} - \alpha_j^{min} > \epsilon$ then go to 5.
7. Multiply all α_j by the constant α_u.
8. For each scale j and for each wavelet coefficient w find $\tilde{w}_{j,k}$ by minimizing $j_m(\tilde{w}_{j,k})$ or $j_{ms}(\tilde{w}_{j,k})$.

9. Reconstruct the filtered image from $\tilde{w}_{j,k}$ by the inverse wavelet transform.

The minimization of j_m or j_{ms} (step 5b) can be done by any method. For instance, a simple dichotomy can be used in order to find $\tilde{w}$ such that

$$\nabla(j_m(\tilde{w}_{j,k})) = \frac{\partial h_s(w_{j,k} - \tilde{w}_{j,k})}{\partial \tilde{w}_{j,k}} + \alpha_j \frac{\partial h_n(\tilde{w}_{j,k})}{\partial \tilde{w}_{j,k}} = 0 \quad (2.19)$$

In the case of Gaussian noise, and by using N2-MSE approach, we derive from equations 2.41 and 2.46 described in the next section that $\nabla(j_m(\tilde{w}_{j,k}))$ is defined by:

$$\begin{aligned} \nabla(j(\tilde{w}_{j,k})) = & -\frac{w_{j,k} - \tilde{w}_{j,k}}{\sigma_j^2} \mathrm{erf}\left(\frac{w_{j,k} - \tilde{w}_{j,k}}{\sqrt{2}\sigma_j}\right) + \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_j}\left[1 - e^{-\frac{(w_{j,k} - \tilde{w}_{j,k})^2}{2\sigma_j^2}}\right] + \\ & \alpha_j \left(\frac{\tilde{w}_{j,k}}{\sigma_j^2} \mathrm{erfc}\left(\frac{\tilde{w}_{j,k}}{\sqrt{2}\sigma_j}\right) + \frac{1}{\sigma_j}\sqrt{\frac{2}{\pi}}\left[1 - e^{-\frac{\tilde{w}_{j,k}^2}{2\sigma_j^2}}\right]\right) \quad (2.20) \end{aligned}$$

The idea to treat the wavelet coefficients such that the residual respects some constraint has also been used in [65, 66, 4] using cross-validation. However, cross validation appears to overfit the data [87].

2.3.5 Derivative calculation of h_s and h_n for N2-MSE

The erf function

The erf and erc functions are defined by

$$\begin{aligned} \mathrm{erf(x)} &= \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \\ \mathrm{erfc}(x) &= 1 - \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \quad (2.21) \end{aligned}$$

These functions have the following symetries

erf(0) = 0	erf(∞) = 1
erfc(0) = 1	erfc(∞) = 0
erf(-x) = erf(x)	erfc(-x) = 2-erfc(x)

(2.22)

Gaussian case

We compute the contribution of the wavelet coefficient x to the noise information:

$$h_n(x) = \frac{1}{\sigma^2} \int_0^x t \,\mathrm{erfc}\left(\frac{x - t}{\sqrt{2}\sigma}\right) dt \quad (2.23)$$

$$\begin{aligned}
\frac{dh_n(x)}{dx} &= h_n(x+dx) - h_n(x) \\
&= \frac{1}{\sigma^2}\int_0^{x+dx} t \text{ erfc}\left(\frac{x+dx-t}{\sqrt{2}\sigma}\right) dt - \frac{1}{\sigma^2}\int_0^{x} t \text{ erfc}\left(\frac{x-t}{\sqrt{2}\sigma}\right) dt \\
&= \frac{1}{\sigma^2}\int_0^{x} \left[t \text{ erfc}\left(\frac{x+dx-t}{\sqrt{2}\sigma}\right) - t \text{ erfc}\left(\frac{x-t}{\sqrt{2}\sigma}\right)\right] dt + \frac{1}{\sigma^2}\int_x^{x+dx} t \text{ erfc}\left(\frac{x+dx-t}{\sqrt{2}\sigma}\right) dt \\
&= \frac{1}{\sigma^2}\int_0^{x} t\frac{\partial \text{ erfc}\left(\frac{x-t}{\sqrt{2}\sigma}\right)}{\partial x} + \frac{1}{\sigma^2}\int_x^{x+dx} t \text{ erfc}\left(\frac{x+dx-t}{\sqrt{2}\sigma}\right) dt \\
&= \frac{1}{\sigma^2}\int_0^{x} t\frac{\partial \ \text{ erfc}(\frac{x-t}{\sqrt{2}\sigma})}{\partial x}dt + \frac{x}{\sigma^2} \ \text{erfc}(0) \qquad (2.24)
\end{aligned}$$

Now, because $\text{erfc}(0) = 1$ we have:

$$\frac{dh_n(x)}{dx} = \frac{x}{\sigma^2} + \frac{1}{\sigma^2}\int_0^x t\frac{\partial \ \text{erfc}(\frac{x-t}{\sqrt{2}\sigma})}{\partial x}dt \qquad (2.25)$$

We derive the function erfc:

$$\frac{\partial \, \text{erfc}(x)}{\partial x} = -\frac{2}{\sqrt{\pi}}e^{-x^2} \qquad (2.26)$$

$$\frac{\partial \, \text{erfc}(\frac{(x-t)}{\sqrt{2}\sigma})}{\partial x} = -\frac{2}{\sqrt{\pi}}\frac{1}{\sqrt{2}\sigma}e^{-\frac{(x-t)^2}{2\sigma^2}} = -\sqrt{\frac{2}{\pi}}\frac{1}{\sigma}e^{-\frac{(x-t)^2}{2\sigma^2}} \qquad (2.27)$$

Now we deduce for the derivative of h_n:

$$\frac{dh_n(x)}{dx} = \frac{x}{\sigma^2} + \frac{1}{\sigma^2}\int_0^x -\sqrt{\frac{2}{\pi}}\frac{1}{\sigma}t \, e^{-\frac{(x-t)^2}{2\sigma^2}} dt \qquad (2.28)$$

$$\frac{dh_n(x)}{dx} = \frac{x}{\sigma^2} + \frac{1}{\sigma^3}\sqrt{\frac{2}{\pi}}\int_0^x t \, e^{-\frac{(x-t)^2}{2\sigma^2}} dt \qquad (2.29)$$

We create the variable J

$$J = \int_0^x t \, e^{-\frac{(x-t)^2}{2\sigma^2}} dt \qquad (2.30)$$

We create the variable u

$$\begin{array}{rl}
u = \frac{t-x}{\sqrt{2}\sigma} & t = x + u\sqrt{2}\sigma \\
 & dt = \sqrt{2}\sigma du \\
t = 0 \Rightarrow u = \frac{-x}{\sqrt{2}\sigma} & t = x \Rightarrow u = 0
\end{array} \qquad (2.31)$$

The variable J can be written with u

$$J = \int_{\frac{-x}{\sqrt{2}\sigma}}^{0} (x + u\sqrt{2}\sigma) e^{-u^2} \sqrt{2}\sigma du \tag{2.32}$$

$$J = \sqrt{2}\sigma x \int_{\frac{-x}{\sqrt{2}\sigma}}^{0} e^{-u^2} du + 2\sigma^2 \int_{\frac{-x}{\sqrt{2}\sigma}}^{0} u\, e^{-u^2} du \tag{2.33}$$

The first part of J can be rewritten as:

$$J_0 = \sqrt{2}\sigma x \int_{0}^{\frac{x}{\sqrt{2}\sigma}} e^{-u^2} du \tag{2.34}$$

J_0 can be expressed with the error function.

$$J_0 = \sqrt{2}\sigma \frac{\sqrt{\pi}}{2} x \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) = \sigma \sqrt{\frac{\pi}{2}}\, x \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) \tag{2.35}$$

Now the second part of J is obvious

$$J_1 = 2\sigma^2 \int_{\frac{-x}{\sqrt{2}\sigma}}^{0} u\, e^{-u^2} du \tag{2.36}$$

or

$$\frac{de^{-u^2}}{du} = -2ue^{-u^2} \tag{2.37}$$

We replace

$$J_1 = -\sigma^2 \int_{\frac{-x}{\sqrt{2}\sigma}}^{0} d(e^{-u^2}) \tag{2.38}$$

$$J_1 = \sigma^2 [e^{-\frac{x^2}{2\sigma^2}} - 1] \tag{2.39}$$

Now we can write J

$$J = J_0 + J_1 = \sigma \sqrt{\frac{\pi}{2}}\, x \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) + \sigma^2 \left[e^{-\frac{x^2}{2\sigma^2}} - 1\right] \tag{2.40}$$

We can write the derivative of h_n

$$\begin{aligned}
\frac{dh_n(x)}{dx} &= \frac{x}{\sigma^2} - \frac{1}{\sigma^3}\sqrt{\frac{2}{\pi}}\, J \\
&= \frac{x}{\sigma^2} - \frac{x}{\sigma^2}\, x \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) + \frac{1}{\sigma}\sqrt{\frac{2}{\pi}} \left[1 - e^{-\frac{x2}{2\sigma2}}\right] \\
&= \frac{x}{\sigma^2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}\sigma}\right) + \frac{1}{\sigma}\sqrt{\frac{2}{\pi}} \left[1 - e^{-\frac{x^2}{2\sigma^2}}\right] \qquad (2.41)
\end{aligned}$$

In order to minimize the functional (2.10), we may want to calculate the derivative of $h_s(y-x)$, where $h_s(y-x)$ measures the amount of information contained in the residual (y being the data).

$$h_s(y-x) = \frac{1}{\sigma^2}\int_0^{y-x} t \ \mathrm{erf}\left(\frac{y-x-t}{\sqrt{2}\sigma}\right) dt \tag{2.42}$$

Denoting $z = y - x$, we have

$$\begin{aligned} h_s(z) &= \frac{1}{\sigma^2}\int_0^{z} t \ \mathrm{erf}\left(\frac{z-t}{\sqrt{2}\sigma}\right) dt \\ &= \frac{1}{\sigma^2}\int_0^{z} t dt - \frac{1}{\sigma^2}\int_0^{z} t \ \mathrm{erfc}\left(\frac{z-t}{\sqrt{2}\sigma}\right) dt \end{aligned} \tag{2.43}$$

and

$$\frac{dh_s(x)}{dx} = \frac{dh_s(z)}{dz}\frac{dz}{dx} \tag{2.44}$$

$$\frac{dh_s(z)}{dz} = \frac{z}{\sigma^2} - \frac{dh_n(z)}{dz} \tag{2.45}$$

then

$$\begin{aligned} \frac{dh_s(y-x)}{dx} &= -\frac{y-x}{\sigma^2} + \frac{y-x}{\sigma^2}\mathrm{erfc}\left(\frac{y-x}{\sqrt{2}\sigma}\right) + \sqrt{\frac{2}{\pi}}\frac{1}{\sigma}\left[1 - e^{-\frac{(y-x)^2}{2\sigma^2}}\right] \\ &= -\frac{y-x}{\sigma^2}\mathrm{erf}\left(\frac{y-x}{\sqrt{2}\sigma}\right) + \sqrt{\frac{2}{\pi}}\frac{1}{\sigma}\left[1 - e^{-\frac{(y-x)^2}{2\sigma^2}}\right] \end{aligned} \tag{2.46}$$

2.3.6 General Case

The contribution of the wavelet coefficient x to the noise and signal information in the general case is

$$\begin{aligned} h_n(x) &= \int_0^{|x|} P_n(x-u)\left(\frac{\partial h(x)}{\partial x}\right)_{x=u} du \\ h_s(x) &= \int_0^{|x|} P_s(x-u)\left(\frac{\partial h(x)}{\partial x}\right)_{x=u} du \end{aligned} \tag{2.47}$$

Assuming $h(x) = \frac{1}{2}x^2$, we have

$$\begin{aligned} h_n(x) &= \int_0^{|x|} P_n(x-u) u du \\ h_s(x) &= \int_0^{|x|} P_s(x-u) u du \end{aligned} \tag{2.48}$$

$$\frac{dh_s(x)}{dx} = \int_0^x \left(\frac{\partial P_s(x-u)}{\partial x}\right)_{x=u} udu + \frac{1}{dx}\int_x^{x+dx} P_s(x-u)udu \quad (2.49)$$

Since $P_s(0) = 0$, the second term tends to zero.
Denoting $\frac{\partial P_s(x-u)}{\partial x} = -\frac{\partial P_s(x-u)}{\partial u}$, we have

$$\begin{aligned}\frac{dh_s(x)}{dx} &= -\int_0^x \frac{\partial P_s(x-u)}{\partial u} udu \\ &= -([uP_s(x-u)]_0^x - \int_0^x P_s(x-u)du) \\ &= \int_0^x P_s(x-u)du \\ &= \int_0^x P_s(u)du \end{aligned} \quad (2.50)$$

and from $h_n = h - h_s$ we get

$$\frac{dh_n(x)}{dx} = x - \int_0^x P_s(u)du \quad (2.51)$$

and

$$\frac{dh_s(y-x)}{dx} = -\int_0^{y-x} P_s(u)du \quad (2.52)$$

It is easy to verify that replacing $P_s(x) = \mathrm{erf}(x)$, and $P_n(x) = \mathrm{erfc}(x)$ (case of Gaussian noise) we find the same equation as in the Gaussian case.

2.3.7 Optimization

In the case of Gaussian noise, the calculation of *erf* and *erfc* functions could lead to a considerable time computation, when compared to a simple filtering method. This can be easily avoided by precomputing tables, which is possible due to the specific properties of $\frac{\partial h_s}{\partial \tilde{w}}$ and $\frac{\partial h_n}{\partial \tilde{w}}$. h_s and h_n are functions of the standard deviation of the noise, and we denote the reduced functions by h_s^r and h_n^r, i.e. h_s and h_n for noise standard deviation equal to 1. It is easy to verify that

$$\frac{\partial h_s(w_{j,k})}{\partial \tilde{w}} = \sigma_j \frac{\partial h_s^r(\frac{w_{j,k}}{\sigma_j})}{\partial \tilde{w}} \quad (2.53)$$

$$\frac{\partial h_n(w_{j,k})}{\partial \tilde{w}} = \sigma_j \frac{\partial h_n^r(\frac{w_{j,k}}{\sigma_j})}{\partial \tilde{w}} \quad (2.54)$$

Furthermore, $\frac{\partial h_n^r}{\partial \tilde{w}}$ and $\frac{\partial h_s^r}{\partial \tilde{w}}$ are symmetric functions, $\frac{\partial h_n^r}{\partial \tilde{w}}$ converges to a constant value C (C=0.798), and $\frac{\partial h_s^r}{\partial \tilde{w}}$ tends to $C - w$ when w is large enough (> 5). In our implementation, we precomputed the tables using a step-size of 0.01 from 0 to 5. If no model is introduced and if the SNR is not used, the filtered wavelet of coefficients is a function of α and $\frac{w_j}{\sigma_j}$, and a second level of optimization can be performed by precomputed tables of solutions for different values of α.

2.4 Examples

2.4.1 1D data filtering

Figures 2.3, 2.4 and 2.5 show the results of the multiscale entropy method on simulated data (2048 pixels). From top to bottom, each figure shows simulated data, the noisy data, the filtered data, and both noisy and filtered data overplotted. For the two first filterings, all default parameters were taken (noise standard deviation and α_j automatically calculated, $\alpha_u = 1$, and the chosen wavelet transform algorithm is the à trous one). For the block signal (Fig. 2.3), default parameters were also used, but the multiresolution transform we used is the multiresolution median transform.

Figure 2.6 shows the result after applying the MEF method to a real spectrum (512 pixels). The last plot shows the difference between the original and the filtered spectrum. As we can see, the residual contains only noise. In this case, we used also default parameters, but we introduce the SNR in the calculation of α.

2.4.2 Image filtering

A simulated 256×256 image containing stars and galaxies is shown in Fig. 2.7 (top left). The simulated noisy image, the filtered image and the residual image are respectively shown in Fig. 2.7 top right, bottom left, and bottom right. We can see that there is no structure in the residual image.

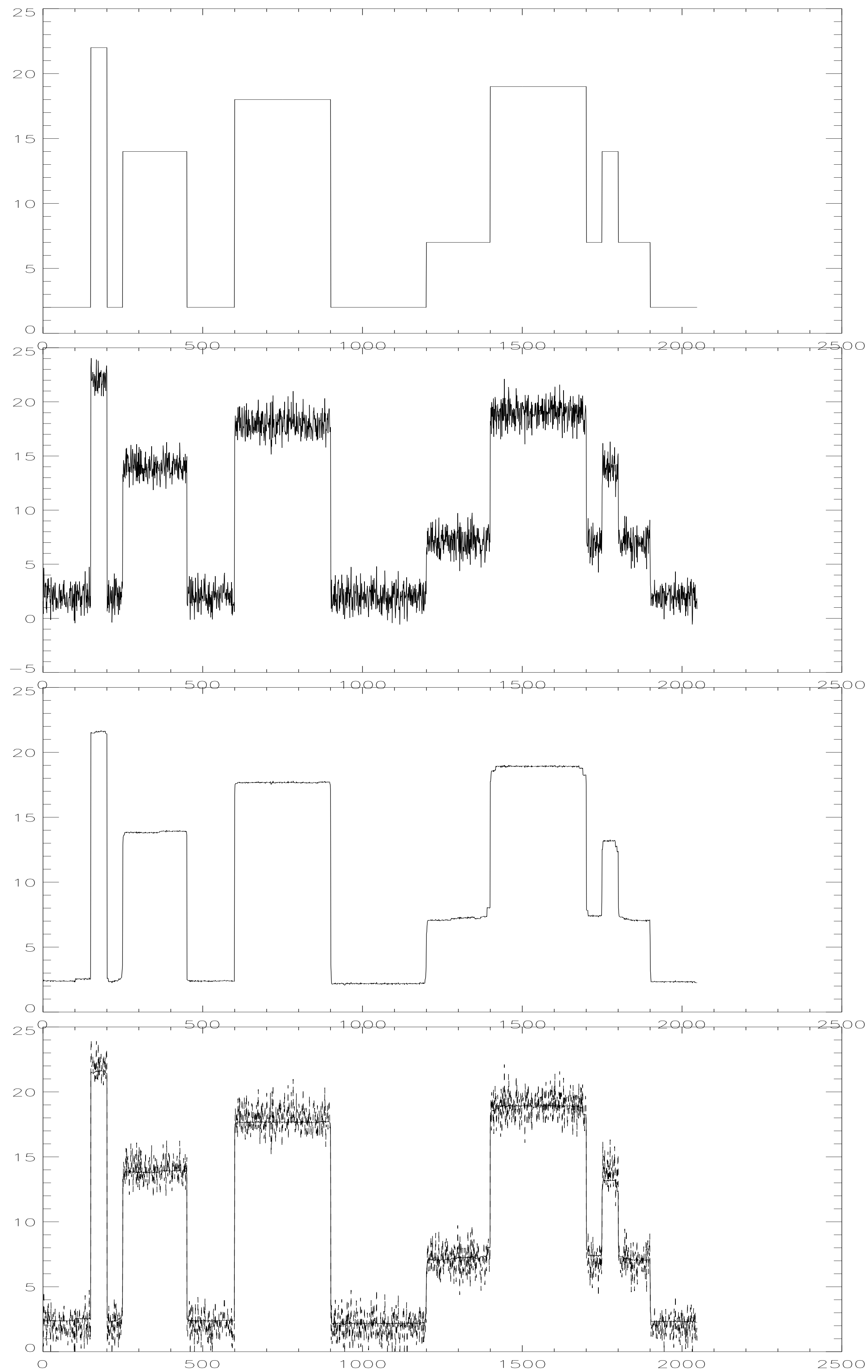


Figure 2.3: From top to bottom, simulated block data, noise blocks, filtered blocks, and both noisy and filtered blocks overplotted.

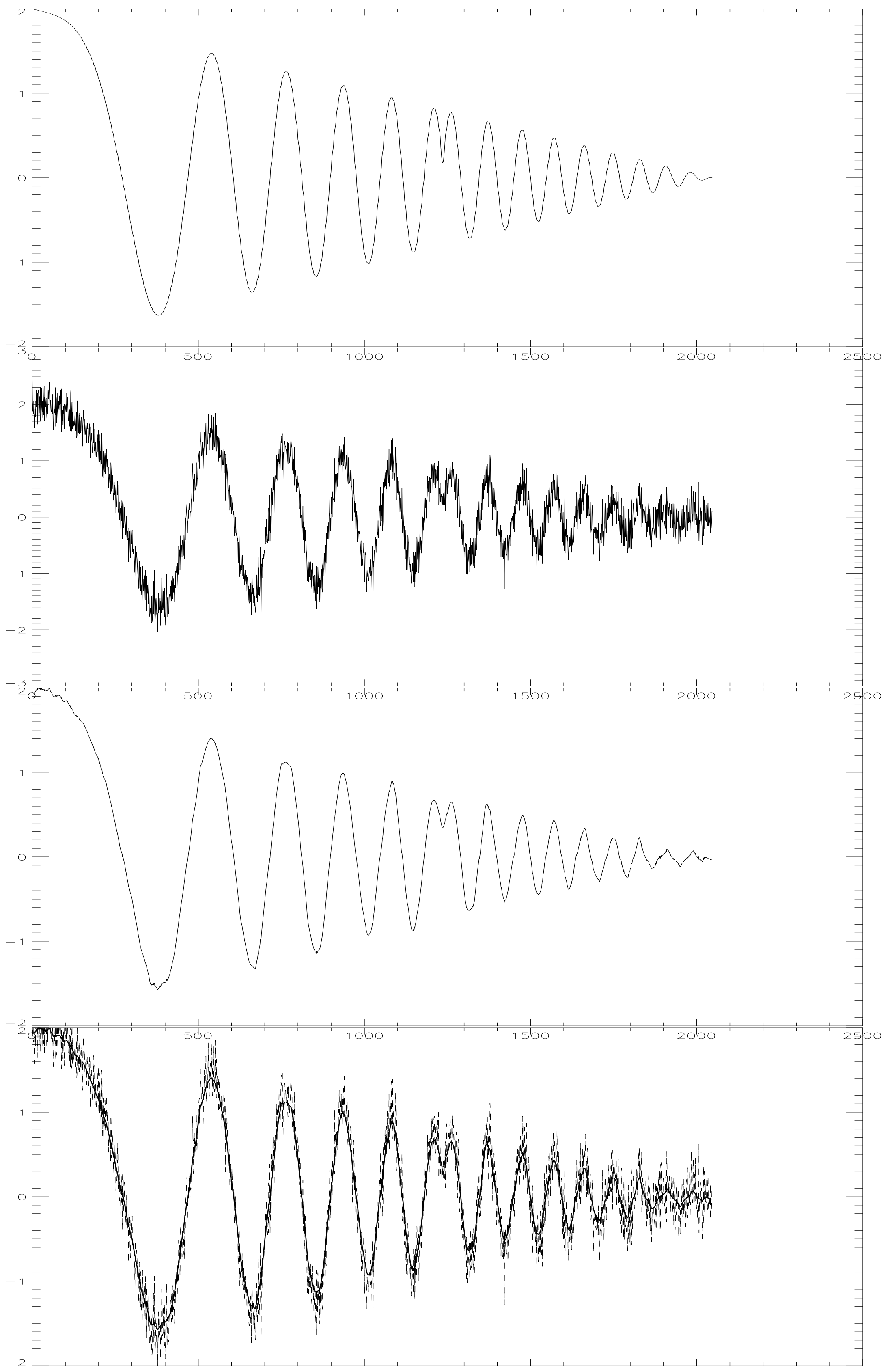


Figure 2.4: From top to bottom, simulated data, noisy data, filtered data, and both noisy and filtered data overplotted.

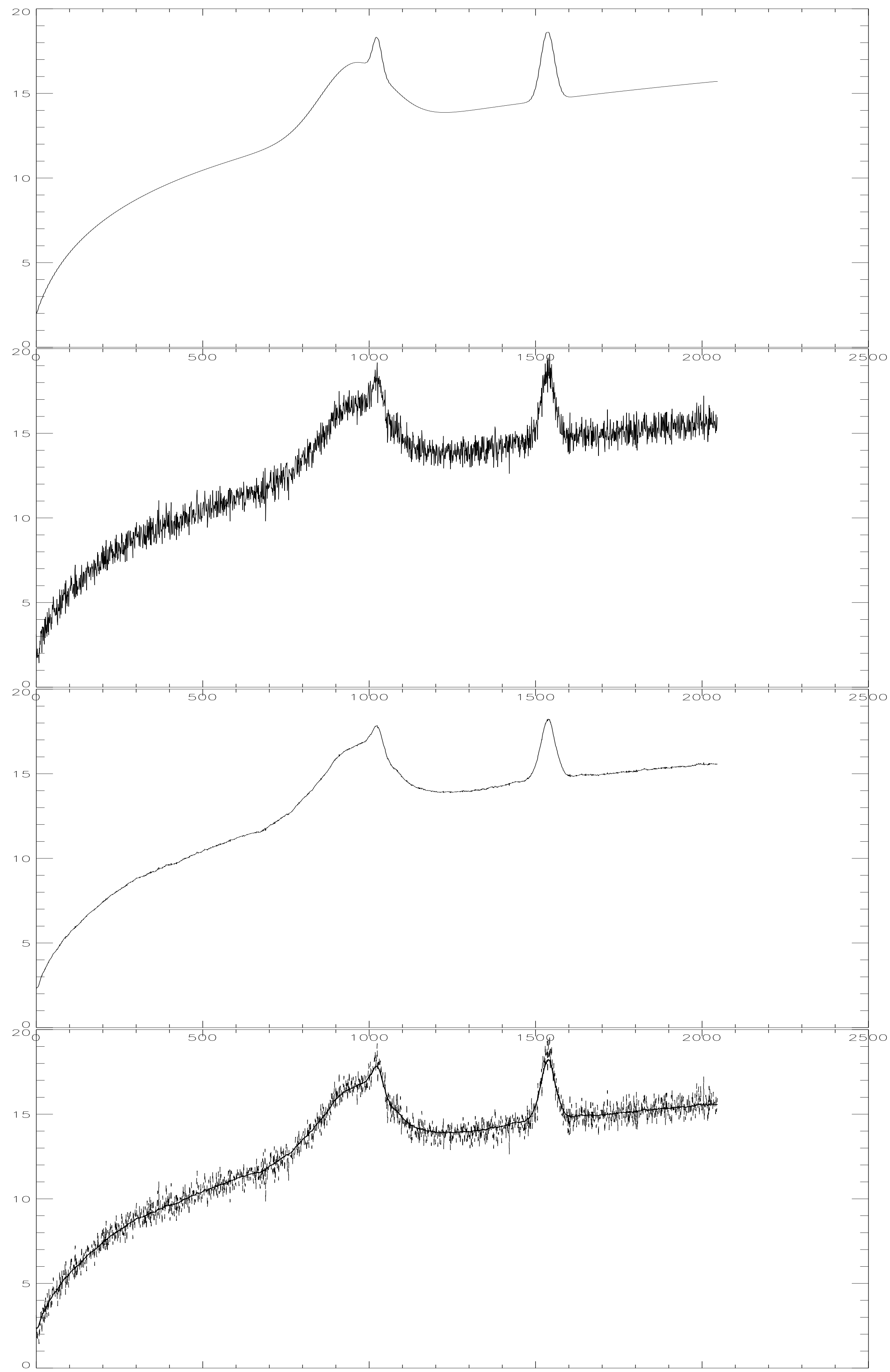


Figure 2.5: From top to bottom, simulated data, noisy data, filtered data, and both noisy and filtered data overplotted.

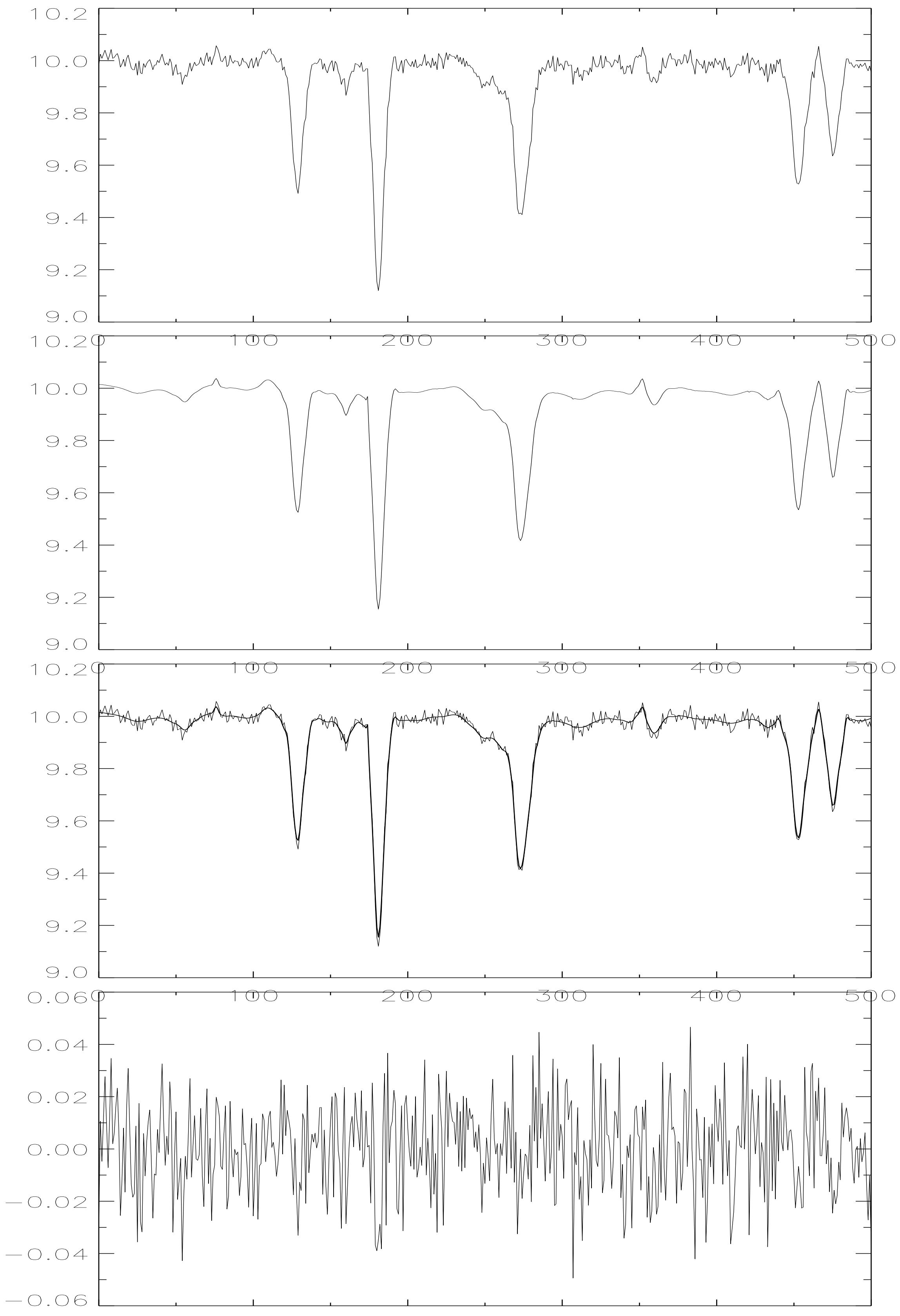


Figure 2.6: From top to bottom, real spectrum, filtered spectrum, both noisy and filtered spectrum overplotted, and difference between the spectrum and the filtered data. As we can see, the residual contains only noise.

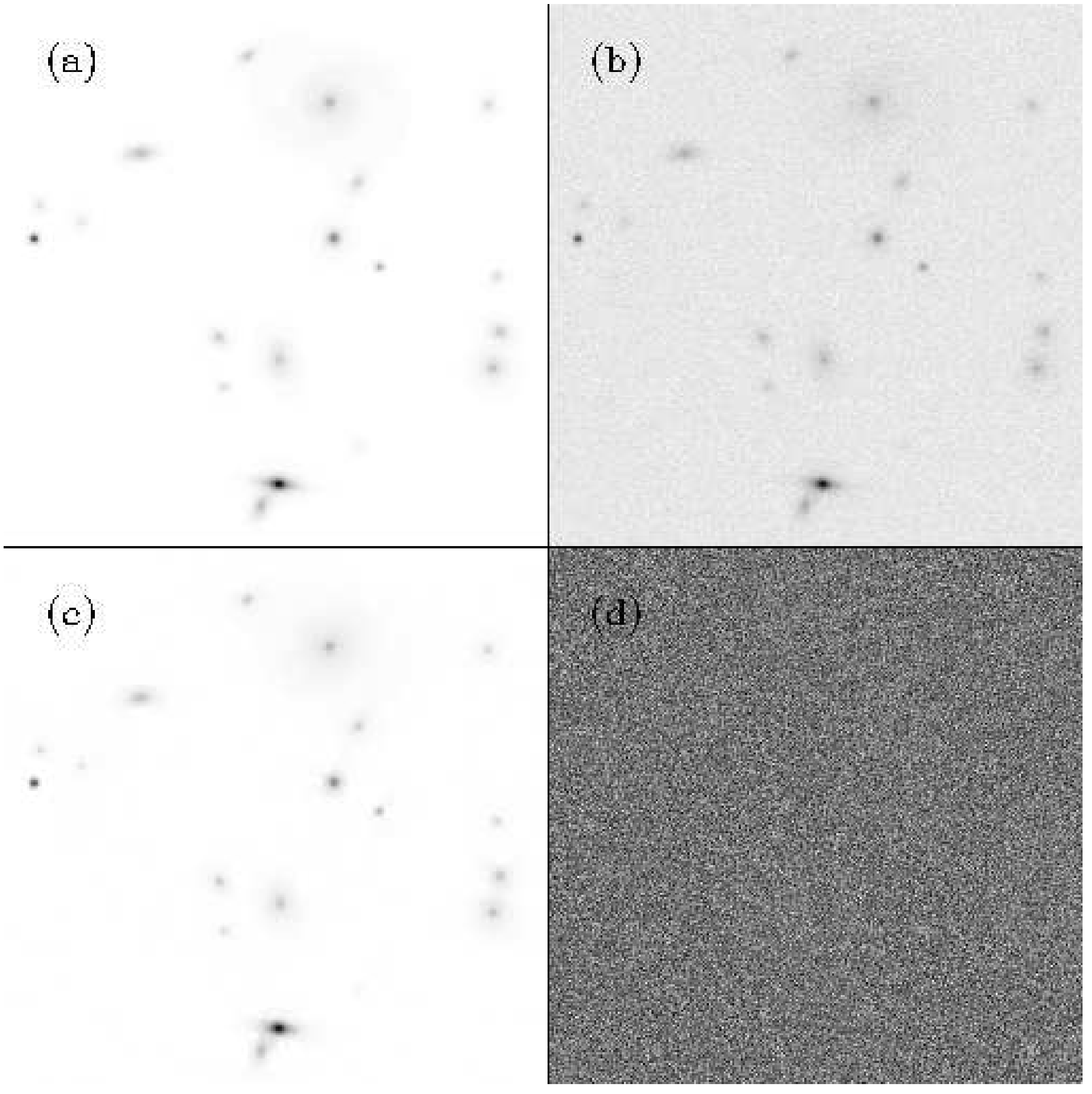


Figure 2.7: (a) Simulated image, (b) simulated image and Gaussian noise, (c) filtered image, and (d) residual image.

2.5 Comparison with other methods from simulations

2.5.1 Simulation descriptions

A set of simulations were carried out based on two images: the classical Lena 512×512 image, and a 512×512 landscape image. From each image, three images were created by adding Gaussian noise with standard deviations of 5,10,30. These six images were filtered using different multiresolution methods and different noise treatment methods. The multiresolution methods were:

1. Haar wavelet transform (FWT-Haar).
2. Mallat-Daubechies biorthogonal wavelet transforms using the Dauchechies-Antonini 7/9 filters [6] (FWT-7/9).
3. Feauveau wavelet transform.
4. À trous algorithm using a B-spline scaling function (see [81, 84] for more details).
5. Multiresolution median transform (MMT) [86].

The first two belong to the class of fast wavelet transforms. The third is also a non-redundant transform, but compared to the FWT, the wavelet function is isotropic. The à trous algorithm is redundant and symmetric, and finally the MMT is not a wavelet transform, but does allow a multiresolution representation.

Using these five transforms, we used eight different strategies for correcting the multiresolution coefficients from the noise:

- k-sigma hard and soft thresholding
- Donoho hard and soft thresholding
- Multiscale entropy method
- Hierarchical hard thresholding
- Multiresolution Wiener filtering
- Hierarchical Wiener filtering

The last three strategies have up to now only been used with redundant transforms (à trous algorithm and MMT in our case).

Finally, close to two hundred filtered images were created. Four resolution scales were used for the filtering, and the constant k for the hard thresholding was always taken as equal to 4 for the first scale, and 3 for the others. For the multiscale entropy method, the parameter α was determined by the program in order to get a standard deviation of the residual (i.e. image minus filtered image) of the same order as the noise standard deviation.

For each filtered image, the PSNR (peak signal-to-noise) ratio between the original image I and the filtered image F was calculated as:

$$PSNR_{dB} = 10 \log_{10} \frac{255}{NRMSE^2} \tag{2.55}$$

where NRMSE is the normalized root mean square error:

$$NRMSE^2 = \frac{\sum_{pix}(I - F)^2}{\sum_{pix} I^2} \tag{2.56}$$

We also calculated the correlation factor, but we found that this does not furnish more information than the PSNR. If the PSNR is an objective measure, it is however not sufficient, because it does not allow us to control whether artifacts are present or not. Images were therefore also visually assessed, in order to decide if artifacts are visible.

Results of the simulations are presented in Tables 2.5.1,2.5.1.

2.6 Simulation Analysis

Multiresolution algorithm

Filtering using the Haar transform always produces artifacts, even at low noise levels. When using other filters, artifacts appear only beyond a given noise level. Improving the filter set improves the filtered image quality, which is a well-known result. When the noise increases, artifacts appear, even with a good filter set such as the Antonini 7/9 one.

- **Feauveau WT.** The standard orthogonal WT is always better than the Feauveau method for filtering.
- **À trous algorithm.** This does not create artifacts when thresholding, and results are significantly better (from the visual point of view) at high noise levels, compared to orthogonal WT approaches. As opposed to the standard WT method, this transform is symmetric and performs better on isotropic structures compared to faint contours. This is the reason for its success on

astronomical images where objects are diffuse and more or less isotropic (stars, galaxies, etc.).

- **Multiresolution median transform.** This transform is non-linear, and noise estimation at the different scales cannot be carried out in the same rigorous way as with linear transforms. For pure Gaussian noise, there is clearly no interest in using this transform, even if it respects well the morphology of the objects contained in the image. For some other kinds of noise, the non-linearity can be an advantage, and it can then be considered.

Conclusion

The Feauveau WT and the MMT are not competitive for filtering in the case of Gaussian noise. FWT-7/9 allows better restoration of the edges than the à trous algorithm, but the à trous algorithm is more robust from the visual point of view. The important point to be made is clearly that the way the information is represented is fundamental. At high noise levels, whatever the chosen filter set, we will always have more artifacts using the FWT than with the à trous algorithm.

Noise treatment strategies

- **The optimal method depends on the noise level.**
  At low noise levels, simple thresholding using an orthogonal wavelet transform leads to very good results. When the noise increases, artifacts appear. Non-orthogonal transforms produce better results, and soft thresholding strategies lead to more acceptable image quality.

- **Donoho soft and hard thresholding versus the k-sigma approach.**
  Whatever the multiresolution transform and the noise level, the k-sigma hard (respectively soft) thresholding is always better than the Dohono hard (respectively soft) thresholding. Both PSNR ratio and visual aspect are better using the k-sigma approach. This outcome is not too surprising. Indeed the threshold, in the Donoho approach, is increasing with the number of pixels (justified in order to have a fixed number of "artifacts"). For our 512×512 image, this approach is equivalent to thresholding at 5σ. But then the thresholding level is too high, because the main coefficients between 3σ and 5σ are significant. The larger the image size, the stronger will be the over-smoothing.

- **Hierarchical thresholding.**
  The modification of the thresholding level at a given scale using the information at the following scale improves the result. The PSNR is better, and the visual aspect is similar to the hard thresholding. This procedure could certainly be also introduced into orthogonal transforms.

- **Quality of the multiscale entropy method.**
  The multiscale entropy method proposes a visually good solution whatever the noise level. It is in fact a method which preserves high wavelet coefficients, and corrects other wavelet coefficients in an adaptive, soft, manner.

2.7 Conclusion: Toward Combined Filtering

If a hard or a soft thresholding approach is used, the k-sigma value should be preferred to the universal $\sqrt{(2\log(n))}$ value. Multiresolution Wiener filtering and hierarchical Wiener filtering are not at all competitive.

The multiscale entropy method is an adaptive soft approach which is certainly the best when considering both visual quality and the PSNR criterion. At low noise levels, a FWT can be used, which allows better restoration of edges (assuming the image does contain edges!), and at high noise levels, the à trous algorithm must be chosen since otherwise artifacts related to decimation appear. However, these artifacts are less severe than those produced by poor thresholding.

Figure 2.8 shows how a wavelet coefficient is modified using a hard thresholding, a soft thresholding, MEF method, and MEF method with α as a function of the SNR. As we can see, MEF methods are intermediate between hard and soft thresholding, but do not present any discontinuity as the hard thresholding. This is the reason why good SNR is obtained with the MEF method, while retaining also good visual quality.

As the previous section demonstrates, it is not easy to find an optimal method for all noise amplitudes. At low noise levels, hard thresholding methods produce perfect results, and when the noise increases, the situation changes. Orthogonal transforms produce artifacts related to the decimation, hard thresholding produces visual artifacts which are less severe with soft methods, such as soft thresholding and multiscale entropy. The isotropic wavelet transform allows better extraction of isotropic structures, whereas the biorthogonal transform describes the contours in a better way. The MMT permits also a relatively good

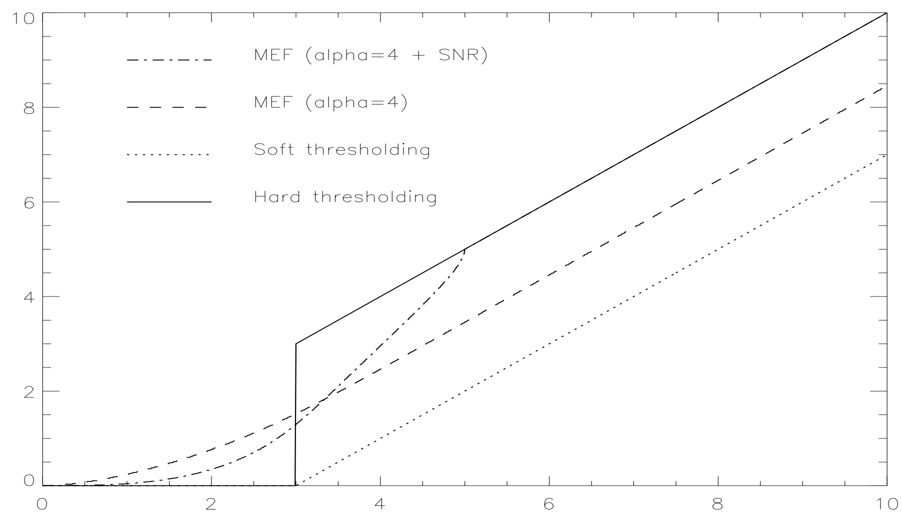


Figure 2.8: Filtered wavelet coefficients versus wavelet coefficients (for a noise standard deviation equal to 1) by four methods: hard thresholding, soft thresholding, multiscale entropy filtering, and multiscale entropy filtering with a non-constant α (SNR-dependent) value.

object shape description. The conclusion is that there is no perfect method. Each method has its advantages and its drawbacks, and can be in some situations better than others. Parameters for choosing one method rather than another include the noise level, the nature of the noise, and the shape of the significant information.

Having a default optimal method seems impossible when using any of the strategies described above. This can also be interpreted in another way: the complexity of an image is so high, that all vision models are too simplistic to fully describe the information, and subsequently to separate the signal from the noise.

We have therefore developed the idea to combine the results obtained from different filtering strategies. Seven filtered images have been simply averaged. These seven images were obtained from the soft thresholding, the hard thresholding, and the multiscale entropy method using both the à trous algorithm and the biorthogonal transform, and the hierarchical thresholding with the à trous algorithm. PSNR is represented in Table 2.7. Comparing Table 2.7 with previous results, we see that PSNR of the combined method is clearly always above that of all other methods. The visual aspect is also always improved. This means that the results obtained from different vision models do not present the same artifacts, which tend to disappear when averaging. The significant information is perhaps lost in some of the combined images, but not in all, and the combined filtered image is always better.

Figures 2.9 and 2.10 show filtered images (Lena and the landscape image) using different methods. The upper left shows the noisy image, after adding Gaussian noise ($\sigma = 30$). The upper right, lower left, and lower right show respectively the filtered image by hard thresholding using FWT-7/9, by the multiscale entropy method, and by the combined method.

The combined filtering has in fact already been used by Donoho [29], when he repeated n times the filtering operations by shifting its 1D signal from one pixel at each operation. The border problem related to the shifting is managed by "wrap-around", – the last pixel precedes the first. All results were then averaged, and this effectively reduces the number of artifacts. This approach however raises two problems: first, it seems difficult to extend it to 2D or 3D signals because the number of filtering operations needed will become very high. Second, the way the boundary is managed can lead to significant artifacts if there is strong discontinuity between the first pixels and the last ones, which is obviously possible in practice (whenever there is a continuum or trend in the data).

The combined filtering proposed here is more general and it can be applied whatever the size or the dimension of the signal. We have used here seven filterings, but this choice does not result from detailed study. We plan in the future to analyze which combinations lead to the highest improvement.

Method	FWT-Haar	FWT-7/9	Feauveau	à trous	MMT
Hard thresh.	34.63	35.95	33.27	35.20	34.82
Soft thresh.	32.35	33.83	30.67	32.30	32.43
Donoho hard thresh.	33.19	34.62	31.05	33.98	33.68
Donoho soft thresh.	30.69	32.09	28.73	30.76	31.19
Hierarchical thresh.	-	-	-	35.26	34.89
Hierarchical Wiener	-	-	-	33.35	31.91
Multiresol. Wiener	-	-	-	33.42	31.93
Multiscale Entropy	35.86	36.76	-	35.82	35.56

Table 2.1: PSNR after filtering the simulated image (Lena + Gaussian noise (sigma=5)).

Method	FWT-Haar	FWT-7/9	Feauveau	à trous	MMT
Hard thresh.	31.31	32.97	29.87	32.63	31.80
Soft thresh.	29.72	31.29	28.05	30.03	30.15
Donoho hard thresh.	29.94	31.55	27.68	31.33	30.88
Donoho soft thresh.	28.18	29.66	26.77	28.49	29.09
Hierarchical thresh.	-	-	-	32.75	31.93
Hierarchical Wiener	-	-	-	31.71	30.33
Multiresol. Wiener	-	-	-	31.68	30.24
Multiscale Entropy	32.12	33.39	-	32.41	31.95

Table 2.2: PSNR after filtering the simulated image (Lena + Gaussian noise (sigma=10)).

Method	FWT-Haar	FWT-7/9	Feauveau	à trous	MMT
Hard thresh.	26.82	27.97	26.00	28.58	28.19
Soft thresh.	26.27	27.67	25.85	26.85	27.27
Donoho hard thresh.	25.99	27.46	25.80	27.03	27.42
Donoho soft thresh.	25.29	26.78	25.80	25.85	26.58
Hierarchical thresh.	-	-	-	28.97	28.42
Hierarchical Wiener	-	-	-	28.08	27.96
Multiresol. Wiener	-	-	-	27.25	26.81
Multiscale Entropy	27.45	28.75	-	28.37	27.96

Table 2.3: PSNR after filtering the simulated image (Lena + Gaussian noise (sigma=30)).

Method	FWT-Haar	FWT-7/9	Feauveau	à trous	MMT
Hard thresh.	32.50	33.02	30.48	32.49	31.79
Soft thresh.	30.35	30.97	28.27	29.87	29.59
Donoho hard thresh.	31.04	31.53	28.38	31.23	30.67
Donoho soft thresh.	28.80	29.40	26.70	28.46	28.45
Hierarchical thresh.	-	-	-	32.51	31.82
Hierarchical Wiener	-	-	-	30.59	30.32
Multiresol. Wiener	-	-	-	30.65	30.35
Multiscale Entropy	34.63	34.94	-	34.30	33.97

Table 2.4: PSNR after filtering the simulated image (Landscape + Gaussian noise (sigma=5)).

Method	FWT-Haar	FWT-7/9	Feauveau	à trous	MMT
Hard thresh.	29.32	30.00	27.38	29.88	28.91
Soft thresh.	27.89	28.66	26.18	27.78	27.45
Donoho hard thresh.	28.05	28.74	25.86	28.58	27.98
Donoho soft thresh.	26.53	27.32	25.32	26.50	26.52
Hierarchical thresh.	-	-	-	29.99	28.99
Hierarchical Wiener	-	-	-	29.59	28.04
Multiresol. Wiener	-	-	-	29.64	28.04
Multiscale Entropy	30.80	31.35	-	30.70	30.16

Table 2.5: PSNR after filtering the simulated image (Landscape + Gaussian noise (sigma=10)).

Method	FWT-Haar	FWT-7/9	Feauveau	à trous	MMT
Hard thresh.	25.44	26.01	24.80	26.55	25.90
Soft thresh.	25.03	25.88	24.75	25.33	25.10
Donoho hard thresh.	24.77	25.60	24.72	25.36	25.19
Donoho soft thresh.	24.26	25.29	24.725	24.61	24.51
Hierarchical thresh.	-	-	-	27.07	26.21
Hierarchical Wiener	-	-	-	26.52	25.83
Multiresol. Wiener	-	-	-	25.89	25.24
Multiscale Entropy	26.33	27.11	-	26.88	26.16

Table 2.6: PSNR after filtering the simulated image (Landscape + Gaussian noise (sigma=30)).

Images	PSNR
Lena+5σ	37.09
Lena+10σ	33.90
Lena+30σ	29.44
Landscape+5σ	34.70
Landscape+10σ	31.39
Landscape+30σ	27.42

Table 2.7: PSNR after averaging of seven filtered images.

Figure 2.9: Upper left, Lena image + noise ($\sigma = 30$). Upper right, filtered image by hard thresholding using FWT-7/9. Lower left, filtered image by the multiscale entropy method, and lower fight, combined filtered image.

Figure 2.10: Upper left, landscape image + noise ($\sigma = 30$). Upper right, filtered image by hard thresholding using FWT-7/9. Lower left, filtered image by the multiscale entropy image, and lower right, combined filtered image.

Chapter 3

Multiscale Entropy Applied to Deconvolution

3.1 Introduction to deconvolution

Consider an image characterized by its intensity distribution (the "data") I, corresponding to the observation of a "real image" O through an optical system. If the imaging system is linear and shift-invariant, the relation between the data and the image in the same coordinate frame is a convolution:

$$I = O * P + N \tag{3.1}$$

P is the point spread function (PSF) of the imaging system, and N is additive noise. In practice $O*P$ is subject to non-stationary noise which one can tackle by simultaneous object estimation and restoration [45]. The issue of more extensive statistical modeling will not be further addressed here (see [49, 50, 59]), beyond noting that multiresolution frequently represents a useful framework, allowing the user to introduce a priori knowledge of objects of interest.

In Fourier space we have:

$$\hat{I} = \hat{O}\hat{P} + \hat{N} \tag{3.2}$$

We want to determine $O(x, y)$ knowing I and P. This inverse problem has led to a large amount of work, the main difficulties being the existence of: (i) a cut-off frequency of the point spread function, and (ii) the additive noise (see for example [23]).

Eqn. 3.1 is usually in practice an ill-posed problem. This means that there is not a unique solution.

3.2 Non-regularized deconvolution methods

Inversed filtering

This method, sometimes called *Fourier-quotient method*, computes the Fourier transform of the deconvolved object $\hat{O}$ by a simple division between the image $\hat{I}$ and the PSF $\hat{P}$

$$\hat{\tilde{O}}_u = \frac{\hat{I}_u}{\hat{P}_u} = \hat{O}_u + \frac{\hat{N}_u}{\hat{P}_u} \quad (3.3)$$

This algorithm is very fast. We only need to do a Fourier transform and an inverse Fourier transform. For frequencies close the frequency cut-off, the noise term becomes important, and the noise is amplified. Then in the presence of noise, this method cannot be used. To reduce the artifact, it is possible to convolve the solution with a smoothing function (Gaussian). Another possibility is to use Wiener filtering which is defined by:

$$\hat{W}_u = \frac{\mid \hat{S}_u \mid^2}{\mid \hat{S}_u \mid^2 + \mid \hat{N}_u \mid^2} \quad (3.4)$$

where $\mid \hat{S}_u \mid^2$ and $\mid \hat{N}_u \mid^2$ are the spectral density of the signal and the noise. The filter is

$$\hat{W}_{du} = \frac{\hat{P}_u^*}{\mid \hat{P}_u \mid^2 + \frac{|\hat{N}_u|^2}{|\hat{O}_u|^2}} \quad (3.5)$$

Wiener filtering has serious drawbacks (artifact creation such as ringing effects), and needs spectral noise estimation. Its advantage is that it is very fast.

Jansson-Van Cittert

Van Cittert [21] restoration is relatively easy to write. We start with $k = 0$ and $O^{(0)} = I$ and we iterate:

$$O^{(n+1)} = O^{(n)} + \alpha(I - P * O^{(n)}) \quad (3.6)$$

where α is a convergence parameter generally taken as 1. When k tends to infinity, we have $O = O + I - p * O$, so $I = P * O$. In Fourier space, the convolution product becomes a product

$$\hat{O}^{(n+1)} = \hat{O}^{(n)} + \alpha(\hat{I} - \hat{P}\hat{O}^{(n)}) \quad (3.7)$$

In this equation, the object distribution is modified by adding a term proportional to the residual. The algorithm converges quickly after only 5 or 6 iterations. But the algorithm generally diverges in the presence of noise. Jansson [43] modified this technique in order to give it more robustness by considering constraints on the solution. If we wish that $A \leq O_k \leq B$, the iteration is

$$O_k^{(n+1)} = O_k^{(k)} + r(k)[I_k - P_k * O_k^{(n)}] \tag{3.8}$$

with:

$$r_k = C[1 - 2(B-A)^{-1} \mid O_k^{(n)} - 2^{-1}(A+B) \mid]$$

C being a constant.

Gradient method

The one-step gradient method is provided by the minimization of the norm $\parallel I - P * O \parallel$ [47] and leads to:

$$O^{(n+1)} = O^{(n)} + \alpha P^* * [I - (P * O^{(n)})] \tag{3.9}$$

where $P^*(x,y) = P(-x,-y)$. P^* is the transpose of the point spread function, and $O^{(n)}$ is the current estimate of the desired "real image". In Fourier space we have:

$$\hat{O}^{(n+1)} = \hat{O}^{(n)} + \alpha \hat{P}^*(\hat{I} - \hat{P}\hat{O}^{(n)}) \tag{3.10}$$

This method is more robust than Van Cittert's. The conjugate gradient method provides a faster way to minimize this norm with a somewhat more complex algorithm.

Richardson-Lucy

The Richardson-Lucy method [69, 52] can be derived from Bayes' theorem on conditional probabilities. Given Poisson noise, Shepp and Vardi [75] showed that a maximum likelihood solution was obtained, by use of an expectation-maximization algorithm. Richardson-Lucy image restoration leads to:

$$\begin{aligned} O^{(n+1)} &= O^{(n)}[(I/I^{(n)}) * P^*] \\ I^{(n)} &= P * O^{(n)} \end{aligned} \tag{3.11}$$

This method is commonly used in astronomy. Flux is preserved and the solution is always positive. The positivity of the solution can be obtained too with Van Cittert's and the one-step gradient methods by thresholding negative values in $O^{(n)}$ at each iteration.

Conclusion

All these methods have a severe drawback: noise amplification, which prevents the detection of weak objects, and leads to false detections. To resolve these problems, some constraints must be added to the solution (positivity is already one such constraint, but it is not enough). The addition of such constraints is called regularization. Several regularization methods exist.

3.3 Tikhonov regularization

Tikhonov regularization [88] consists of minimizing the term:

$$\parallel I(x,y) - (P * O)(x,y) \parallel + \lambda \parallel H * O \parallel \tag{3.12}$$

where H corresponds to a high-pass filter. This criterion contains two terms. The first, $\parallel I(x,y) - P(x,y) * O(x,y) \parallel^2$, expresses fidelity to the data $I(x,y)$, and the second, $\lambda \parallel H * O \parallel^2$, expresses smoothness of the restored image. λ is the regularization parameter and represents the trade-off between fidelity to the data and the smoothness of the restored image. Finding the optimal value λ necessitates use of numerical techniques such as cross-validation [37, 34]. This method works well, but computationally it is relatively lengthy and produces smoothed images. This second point can be a real problem when we seek compact structures such as is the case in astronomical imaging.

Tikhonov regularization and wavelet transform

If $w_j^{(I)}$ are the wavelet coefficients of the image I at the scale j, we have:

$$\begin{aligned} \hat{w}_j^{(I)}(u,v) &= \hat{g}(2^{j-1}u, 2^{j-1}v) \prod_{i=j-2}^{i=0} \hat{h}(2^i u, 2^i v) \hat{I}(u,v) \\ &= \frac{\hat{\psi}(2^j u, 2^j v)}{\hat{\phi}(u,v)} \hat{P}(u,v)\hat{O}(u,v) \\ &= \hat{w}_j^{(P)} \hat{O}(u,v) \end{aligned} \tag{3.13}$$

where $w_j^{(P)}$ are the wavelet coefficients of the PSF at the scale j. The wavelet coefficients of the image I are the product of convolution of object O by the wavelet coefficients of the PSF.

To deconvolve the image, we have to minimize for each scale j:

$$\parallel \frac{\hat{\psi}(2^j u, 2^j v)}{\hat{\phi}(u,v)} \hat{P}(u,v)\hat{O}(u,v) - \hat{w}_j^{(I)}(u,v) \parallel^2 \tag{3.14}$$

and for the plane at the lower resolution:

$$\| \frac{\hat{\phi}(2^{n-1}u, 2^{n-1}v)}{\hat{\phi}(u,v)} \hat{P}(u,v)\hat{O}(u,v) - \hat{c}^{(I)}_{n-1}(u,v) \|^2 \tag{3.15}$$

n being the number of planes of the wavelet transform ($(n-1)$ wavelet coefficient planes and one plane for the image at the lower resolution). The problem does not generally have a unique solution, and we need to do a regularization [88]. At each scale, we add the term:

$$\gamma_j \parallel w_j^{(O)} \parallel^2 \ \min \tag{3.16}$$

This is a smoothness constraint. We want to have the minimum information in the restored object. From equations 3.14, 3.15, 3.16, we find:

$$\hat{D}(u,v)\hat{O}(u,v) = \hat{N}(u,v) \tag{3.17}$$

with:

$$\begin{aligned} \hat{D}(u,v) &= \sum_j \mid \hat{\psi}(2^j u, 2^j v) \mid^2 (\mid \hat{P}(u,v) \mid^2 + \gamma_j) \\ &\quad + \mid \hat{\phi}(2^{n-1}u, 2^{n-1}v)\hat{P}(u,v) \mid^2 \end{aligned}$$

and:

$$\begin{aligned} \hat{N}(u,v) &= \hat{\phi}(u,v)[\sum_j \hat{P}^*(u,v)\hat{\psi}^*(2^j u, 2^j v)\hat{w}_j^{(I)} \\ &\quad + \hat{P}^*(u,v)\hat{\phi}^*(2^{n-1}u, 2^{n-1}v)\hat{c}^{(I)}_{n-1}] \end{aligned}$$

if the equation is well constrained, the object can be computed by a simple division of $\hat{N}$ by $\hat{D}$. An iterative algorithm can be used to do this inversion if we want to add other constraints such as positivity. We have in fact a multiresolution Tikhonov regularization. This method has the advantage to furnish a solution quickly, but optimal regularization parameters γ_j cannot be found directly, and several tests are generally necessary before finding an acceptable solution. However, the method can be useful if we need to deconvolve a large number of images with the same noise characteristics. In this case, parameters have to be determined only the first time. In a more general perspective, we prefer to use one of the following iterative algorithms.

3.4 The CLEAN approach

3.4.1 The CLEAN algorithm

This approach assumes the object is composed of point sources. It tries to decompose the image (called the dirty map), obtained by inverse Fourier transform of the calibrated uv data, into a set of δ-functions. This is done iteratively by finding the point with the largest absolute brightness and subtracting the point spread function (dirty beam) scaled with the product of the loop gain and the intensity at that point. The resulting residual map is then used to repeat the process. The process is stopped when some prespecified limit is reached. The convolution of the δ-functions with an ideal point spread function (clean beam) plus the residual equals the restored image (clean map). This solution is only possible if the image does not contain large-scale structures. The algorithm is:

1. Compute the dirty map $I^{(0)}(x, y)$ and the dirty beam $A(x, y)$.

2. Find the maximum value, and the coordinate $(x_{\mathrm{max}}, y_{\mathrm{max}})$ of the corresponding pixel in $I^{(i)}(x, y)$.

3. Compute $I^{(i+1)}(x, y) = I^{(i)}(x, y) - \gamma I_{\mathrm{max}} A_{\mathrm{m}}(x, y)$ with $A_{\mathrm{m}}(x, y) = A(x - x_{\mathrm{max}}, y - y_{\mathrm{max}})$ and the loop gain γ inside [0,1].

4. If the residual map is at the noise level, then go to step 5.
   Else $i \longleftarrow i + 1$ and go to step 2.

5. The clean map is the convolution of the list of maxima with the clean beam (which is generally a Gaussian).

6. Addition of the clean map and the residual map produces the deconvolved image.

3.4.2 Multiresolution CLEAN

The CLEAN solution is only available if the image does not contain large-scale structures. Wakker and Schwarz (1988) introduced the concept of Multiresolution CLEAN (MRC) in order to alleviate the difficulties occurring in CLEAN for extended sources. The MRC approach consists of building two intermediate images, the first one (called the smooth map) by smoothing the data to a lower resolution with a Gaussian function, and the second one (called the difference map) by subtracting the smoothed image from the original data. Both these images are then processed separately. By using a standard CLEAN algorithm on them, the smoothed clean map and difference clean map are obtained. The recombination of these two maps gives the clean map at the full resolution.

In order to describe how the clean map at the full resolution is obtained from the smoothed and difference clean map, a number of symbols must be defined:

- G = the normalized ($\int G(x)dx = 1$) smoothing function; the width of the function is chosen such that the full-width at half maximum of the smoothed dirty beam is f times larger than the full-width at half maximum of the original dirty beam.
- A = dirty beam
- D = dirty map
- $\delta = \delta$-functions
- R = residual after using CLEAN on the map
- B = clean beam with peak value 1
- C = clean map
- s = the scale factor of the dirty beam needed to rescale the smooth dirty beam back to a peak value 1
- r = the scale factor of the dirty beam needed to rescale the smooth clean beam back to a peak value 1
- A_{s} = normalized smooth dirty beam = $sA * G$
- A_{d} = normalized difference dirty beam = $1/(1-\frac{1}{s})(A - \frac{A_{\mathrm{s}}}{s})$
- B_{s} = normalized smooth clean beam = $rB * G$
- B_{d} = normalized difference clean beam = $1/(1-\frac{1}{r})(B - \frac{B_{\mathrm{s}}}{r})$

From the δ-functions found by the CLEAN algorithm, one can restore the dirty map by convolving with the dirty beam and adding the residuals:

$$D = D_{\mathrm{s}} + D_{\mathrm{d}} = \delta_{\mathrm{s}} * A_{\mathrm{s}} + R_{\mathrm{s}} + \delta_{\mathrm{d}} * A_{\mathrm{d}} + R_{\mathrm{d}} \tag{3.18}$$

which can be written also as:

$$D = \left[s\delta_{\mathrm{s}} * G + \frac{s}{s-1}\delta_{\mathrm{d}} * (1-G) \right] * A + R_{\mathrm{s}} + R_{\mathrm{d}} \tag{3.19}$$

If we replace the dirty beam by the clean beam, we obtain the clean map:

$$C = \frac{s}{r}\delta_{\mathrm{s}} * B_{\mathrm{s}} + \frac{s(r-1)}{r(s-1)}\delta_{\mathrm{d}} * B_{\mathrm{d}} + R_{\mathrm{s}} + R_{\mathrm{d}} \tag{3.20}$$

The MRC algorithm needs three parameters. The first fixes the smoothing function G, and the other two are the loop gain and the extra loop gain which are used by CLEAN respectively on the smooth dirty map and difference dirty map.

This algorithm may be viewed as an artificial recipe, but we have shown [82] that it is linked to multiresolution analysis as defined by Mallat [55]. Mallat's theory provides a new representation where a function is a sum of detail structures obtained with the same pattern, the wavelet, with suitable translations and dilations. Wavelet analysis leads to a generalization of MRC from a set of scales.

Our approach allows MRC algorithms to be harmonized with the classical theory of deconvolution.

3.4.3 CLEAN and wavelets

The wavelet transform chosen.

We have seen that there are many wavelet transforms. For interferometric deconvolution, we choose the wavelet transform based on the FFT for the following reasons:

- The convolution product is kept at each scale.
- The data are already in Fourier space, so this decomposition is natural.
- There is a pyramidal implementation available which does not take much memory.

Hence until the end of this chapter, we will consider the use of the pyramidal transform based on the FFT.

Deconvolution by CLEAN in wavelet space.

If $w_j^{(I)}$ are the wavelet coefficients of the image I at the scale j, we get:

$$\hat{w}_j^{(I)}(u,v) = \hat{w}_j^{(P)}\hat{O}(u,v) \tag{3.21}$$

where $w_j^{(P)}$ are the wavelet coefficients of the point spread function at the scale j. The wavelet coefficients of the image I are the convolution product of the object O by the wavelet coefficients of the point spread function.

At each scale j, the wavelet plane $w_j^{(I)}$ can be decomposed by CLEAN ($w_j^{(I)}$ represents the dirty map and $w_j^{(P)}$ the dirty beam) into a set, denoted δ_j, of weighted δ-functions.

$$\begin{aligned}\delta_j = \{&A_{j,1}\delta(x - x_{j,1}, y - y_{j,1}), A_{j,2}\delta(x - x_{j,2}, y - y_{j,2}), \ldots,\\ &A_{j,n_j}\delta(x - x_{j,n_j}, y - y_{j,n_j})\}\end{aligned} \tag{3.22}$$

where n_j is the number of δ-functions at the scale j and $A_{j,k}$ represents the height of the peak k at the scale j.

By repeating this operation at each scale, we get a set $\mathcal{W}_\delta$ composed of weighted δ-functions found by CLEAN ($\mathcal{W}_\delta = \{\delta_1, \delta_2, \ldots\}$). If B is the ideal point spread function (clean beam), the estimation of the wavelet coefficients of the object at the scale j is given by:

$$\begin{aligned} w_j^{(E)}(x,y) &= \delta_j * w_j^{(B)}(x,y) + w_j^{(R)}(x,y) \\ &= \sum_k A_{j,k} w_j^{(B)}(x - x_{j,k}, y - y_{j,k}) + w_j^{(R)}(x,y)\end{aligned} \tag{3.23}$$

where $w_j^{(R)}$ is the residual map. The clean map at the full resolution is obtained by the reconstruction algorithm. If we take a Gaussian function as the scaling function, and the difference between two resolutions as the wavelet ($\frac{1}{2}\psi(\frac{x}{2}, \frac{y}{2}) = \phi(x,y) - \frac{1}{2}\phi(\frac{x}{2}, \frac{y}{2})$), we find the algorithm proposed by Wakker and Schwarz [90]. The MRC algorithm in the wavelet space is:

1. We compute the wavelet transforms of the dirty map, the dirty beam and the clean beam.
2. For each scale j, we decompose by CLEAN the wavelet coefficients of the dirty map into a list of weighted δ-functions δ_j.
3. For each scale j, we convolve δ_j by the wavelet coefficients of the clean beam and we add the residual map $w_j^{(R)}$ to the result in order to obtain the wavelet coefficients of the clean map.
4. We compute the clean map at the full resolution by using the reconstruction algorithm.

Improvements to multiresolution CLEAN.

We apply CLEAN to each plane of the wavelet transform. This allows us to detect at each scale the significant structure. The reconstructed image gives the estimation $\tilde{O}$ found by MRC of the object. But MRC does not assume that this estimation is compatible with the measured visibilities. We want:

$$\mid \hat{\tilde{O}}(u,v) - V_{\rm m}(u,v) \mid < \ \Delta_{\rm m}(u,v) \tag{3.24}$$

where $\Delta_{\rm m}(u,v)$ is the error associated with the measure $V_{\rm m}$.

To achieve this, we use the position of the peaks determined by the MRC algorithm. We have seen that after the use of CLEAN, we get a list of positions δ_j on each plane j, with approximate heights A_j. In fact, we get a nice description of the significant structures in the wavelet space. The height values are not sufficiently accurate, but CLEAN enhances these structures. So we have to determine heights which reduce the error. We do so using Van Cittert's algorithm [21] which converges, even in the presence of noise, because our system is well regularized. Then, heights of the peaks contained in $\mathcal{W}_\delta$ will be modified by the following iterative algorithm:

1. Set $n = 0$ and $\mathcal{W}_\delta^{(0)} = \mathcal{W}_\delta$.
2. Compute $A_{j,l}^{(n+1)} = A_{j,l}^{(n)} + \mathcal{Q}_{j,l}.\mathcal{W}_\delta^{(n)}$ so that we then have:

   $$\delta_j^{(n+1)} = \{A_{j,1}^{(n+1)}\delta(x - x_{j,1}, y - y_{j,1}),$$

   and:

   $$\mathcal{W}_\delta^{(n+1)} = \{\delta_1^{(n+1)}, \delta_2^{(n+1)}, \ldots\}$$

3. $n = n + 1$ and go to step 1.

$\mathcal{Q}$ is the operator that:

- computes the wavelet coefficients of the clean map $w^{(C)}$ by convolving at each scale $\delta_j^{(n)}$ by the clean beam wavelet $w_j^{(B)}$

  $$w_j^{(C)} = \delta_j^{(n)} * w_j^{(B)}$$

- reconstructs the estimated object $O^{(n)}$ at full resolution from $w^{(C)}$
- thresholds the negative values of $O^{(n)}$
- computes the residual $r^{(n)}$ by:

  $$\hat{r}^{(n)} = p(V - \hat{O}^{(n)})$$

  where p is a weight function which depends on the quality of the measurement V (error bars). A possible choice for p is:

  - $p(u,v) = 0$ if we do not have any information at this frequency (i.e. a frequency hole).
  - $p(u,v) = 1 - 2\frac{\Delta_{\rm m}(u,v)}{V_{\rm m}(0,0)}$ if $\Delta_{\rm m}(u,v)$ is the error associated with the measurement $V_{\rm m}(u,v)$.

- computes the wavelet transform $w^{(r^{(n)})}$ of $r^{(n)}$
- extracts the wavelet coefficient of $w^{(r^{(n)})}$ which is at the position of the peak $A_{j,l}\delta(x - x_l, y - y_l)$.

The final deconvolution algorithm is:

1. Convolution of the dirty map and the dirty beam by the scaling function.
2. Computation of the wavelet transform of the dirty map which yields $w^{(I)}$.
3. Computation of the wavelet transform of the dirty beam which yields $w^{(D)}$.
4. Estimation of the standard deviation of the noise N_0 of the first plane from the histogram of w_0. Since we process oversampled images, the values of the wavelet image corresponding to the first scale ($w_0^{(I)}$) are nearly always due to the noise. The histogram shows a Gaussian peak around 0. We compute the standard deviation of this Gaussian function, with a 3-sigma clipping, rejecting pixels where the signal could be significant.
5. Computation of the wavelet transform of the clean beam. We get $w^{(B)}$. If the clean beam is a Dirac delta, then $\hat{w}_j^{(B)}(u,v) = \frac{\psi(2^j u, 2^j v)}{\phi(u,v)}$.
6. Set j to 0.
7. Estimation of the standard deviation of the noise N_j from N_0. This is done from the study of the variation of the noise between two scales, with the hypothesis of a white Gaussian noise.
8. Detection of significant structures by CLEAN: we get δ_j from $w_j^{(I)}$ and $w_j^{(D)}$. The CLEAN algorithm is very sensitive to the noise. Step 1 of this algorithm offers more robustness. CLEAN can be modified in order to optimize the detection.
9. $j = j + 1$ and go to step 7.

10. Reconstruction of the clean map from $\mathcal{W}_\delta = \{\delta_1, \delta_2, \cdots\}$ by the iterative algorithm using Van Cittert's method.

The limited support constraint is implicit because we put information only at the position of the peaks, and the positivity constraint is introduced in the iterative algorithm. We have made the hypothesis that MRC, by providing the coordinates of the peaks, gives the exact position of the information in the wavelet space and we limited the deconvolution problem by looking for the height of the peaks which give the best results. It is a very strong limited support constraint which allows our problem to be regularized. CLEAN is not used as a deconvolution algorithm, but only as a tool to detect the position of structures.

3.5 Regularization from the multiresolution support

3.5.1 Noise suppression based on the wavelet transform

We have noted how, in using an iterative deconvolution algorithm such as Van Cittert or Richardson-Lucy, we define $R^{(n)}(x,y)$, the residual at iteration n:

$$R^{(n)}(x,y) = I(x,y) - (P * O^{(n)})(x,y) \tag{3.25}$$

By using the à trous wavelet transform algorithm ([10, 78, 79]), $R^{(n)}$ can be defined as the sum of its p wavelet scales and the last smooth array:

$$R^{(n)}(x,y) = c_p(x,y) + \sum_{j=1}^{p} w_j(x,y) \tag{3.26}$$

where the first term on the right is the last smoothed array, and w denotes a wavelet scale.

The wavelet coefficients provide a mechanism to extract only the significant structures from the residuals at each iteration. Normally, a large part of these residuals are statistically non-significant. The significant residual ([62, 82]) is then:

$$\bar{R}^{(n)}(x,y) = c_p(x,y) + \sum_{j=1}^{p} T(w_j(x,y)) w_j(x,y) \tag{3.27}$$

T is a function which is defined by:

$$T(w) = \begin{cases} 1 & \text{if w is significant} \\ 0 & \text{if w is non-significant} \end{cases} \tag{3.28}$$

Assuming that the noise follows a given law, $w_j(x,y)$ is significant if the probability that the wavelet coefficient is due to noise is small.

$$w_j(x,y) \text{ is significant if } \left\{ \begin{array}{ll} P(W > w_j(x,y)) < \epsilon & \text{if } w_j(x,y) \geq 0 \\ P(W < w_j(x,y)) < \epsilon & \text{if } w_j(x,y) < 0 \end{array} \right. \tag{3.29}$$

3.5.2 Noise suppression based on the multiresolution support

The multiresolution support

The multiresolution support [81] of an image describes in a logical or boolean way whether an image I contains information at a given scale j and at a given position (x,y). If $M^{(I)}(j,x,y) = 1$ (or $true$), then I contains information at scale j and at the position (x,y). M depends on several parameters:

- The input image.
- The algorithm used for the multiresolution decomposition.
- The noise.
- All constraints we want the support additionally to satisfy.

Such a support results from the data, the treatment (noise estimation, etc.), and from knowledge on our part of the objects contained in the data (size of objects, alignment, etc.). In the most general case, a priori information is not available to us.

The multiresolution support of an image is computed in several steps:

1. We compute the wavelet transform of the image.
2. We estimate the noise standard deviation at each scale. We deduce the statistically significant level at each scale.
3. The binarization of each scale leads to the multiresolution support.
4. Modification using a priori knowledge (if desired).

Step 4 is optional. A typical use of a priori knowledge is the suppression of isolated pixels in the multiresolution support in the case where the image is obtained with a point spread function (PSF) of more than one pixel. Then we can be sure that isolated pixels are residual noise which has been detected as significant coefficients. If we use a pyramidal algorithm or a nonlinear multiresolution transform, the same method can be used.

The multiresolution support is obtained by detecting at each scale the significant coefficients. The multiresolution support is defined by:

$$M(j,x,y) = \begin{cases} 1 & \text{if } w_j(x,y) \text{ is significant} \\ 0 & \text{if } w_j(x,y) \text{ is not significant} \end{cases} \tag{3.30}$$

Regularization

In the approach presented in the preceding section, a wavelet coefficient is significant if it is above a threshold. Therefore a coefficient which is less than this threshold is not considered, even if a significant coefficient had been found at the same scale as this coefficient, during previous iterations; and consequently we were justified in thinking that we had found signal at this scale, and at this position. Arising out of this approach, it follows that the wavelet coefficients of the residual image could contain signal, above the set threshold, which is ignored.

In order to conserve such signal, we use the notion of multiresolution support. Whenever we find signal at a scale j and at a position (x,y), we will consider that this position in the wavelet space belongs to the multiresolution support of the image.

Eqn. 3.27 becomes:

$$\bar{R}^{(n)}(x,y) = c_p(x,y) + \sum_{j=1}^{p} M(j,x,y)\ w_j(x,y) \tag{3.31}$$

An alternative approach was outlined in [63] and [81]: the support was initialized to zero, and built up at each iteration of the restoration algorithm. Thus in eqn. 3.31 above, $M(j,x,y)$ was additionally indexed by n, the iteration number. In this case, the support was specified in terms of significant pixels at each scale, j; and in addition pixels could become significant as the iterations proceeded, but could not be made non-significant. In practice, we have found both of these strategies to be equally acceptable.

Regularization of Van Cittert's Algorithm

Van Cittert's iteration [21] is:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + \alpha R^{(n)}(x,y) \tag{3.32}$$

with $R^{(n)}(x,y) = I^{(n)}(x,y) - (P * O^{(n)})(x,y)$. Regularization using significant structures leads to:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + \alpha \bar{R}^{(n)}(x,y) \tag{3.33}$$

The basic idea of our method consists of detecting, at each scale, structures of a given size in the residual $R^{(n)}(x,y)$ and putting them in the restored image $O^{(n)}(x,y)$. The process finishes when no more structures are detected. Then, we have separated the image $I(x,y)$ into two images $\tilde{O}(x,y)$ and $R(x,y)$. $\tilde{O}$ is the restored image, which ought not to contain any noise, and $R(x,y)$ is the final residual which ought not to contain any structure. R is our estimate of the noise $N(x,y)$.

Regularization of the One-Step Gradient Method

The one-step gradient iteration is:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + P(-x,-y) * R^{(n)}(x,y) \qquad (3.34)$$

with $R^{(n)}(x,y) = I(x,y) - (P * O^{(n)})(x,y)$. Regularization by significant structures leads to:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y) + P(-x,-y) * \bar{R}^{(n)}(x,y) \qquad (3.35)$$

Regularization of the Richardson-Lucy Algorithm

From eqn. 3.1, we have $I^{(n)}(x,y) = (P * O^{(n)})(x,y)$. Then $R^{(n)}(x,y) = I(x,y) - I^{(n)}(x,y)$, and hence $I(x,y) = I^{(n)}(x,y) + R^{(n)}(x,y)$.
The Richardson-Lucy equation is:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y)\left[\frac{I^{(n)}(x,y) + R^{(n)}(x,y)}{I^{(n)}(x,y)} * P(-x,-y)\right] \qquad (3.36)$$

and regularization leads to:

$$O^{(n+1)}(x,y) = O^{(n)}(x,y)\left[\frac{I^{(n)}(x,y) + \bar{R}^{(n)}(x,y)}{I^{(n)}(x,y)} * P(-x,-y)\right] \qquad (3.37)$$

Convergence

The standard deviation of the residual decreases until no more significant structures are found. Convergence can be estimated from the residual. The algorithm stops when a user-specified threshold is reached:

$$(\sigma_{R^{(n-1)}} - \sigma_{R^{(n)}})/(\sigma_{R^{(n)}}) < \epsilon \qquad (3.38)$$

3.6 Regularization using Markov model

3.6.1 Introduction

Let us briefly review the ingredients of the proposed regularization method. The basic procedure uses a Markov random field as the prior model, the relation (3.1) for image formation consisting typically of blur and superposition of Gaussian noise, and Bayes' formula to obtain the posterior distribution $P(x/y)$.

Random Markov field

A random sequence x is called a Markov random field if for each site s:

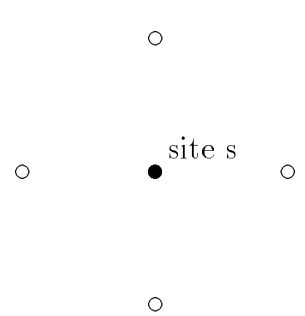
$$P(x) > 0 \qquad \forall x \in \Omega \tag{3.39}$$

$$P(x_s \mid x_t; t \in S - \{s\}) = P(x_s \mid x_t; t \in \mathcal{V}_s) \tag{3.40}$$

where Ω represents all possible configurations and S is the dimension of the field x. At every site s of the field x, the conditional probability depends on a neighborhood $\mathcal{V}_s$ which is defined by the neighbors t of the site s. Note that $\mathcal{V}_s$ increases with the order of the model.

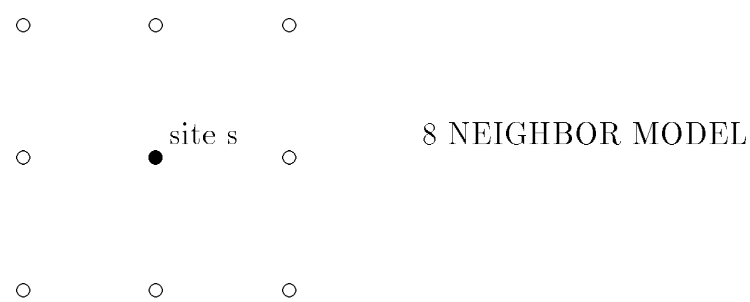
Examples of models

The four-neighbor system is defined as follows:



4 NEIGHBOR MODEL

Because the edges of the image are not only parallel to the axes, there seems to be need to include diagonal neighbor pixels. Then, the four-neighbor model can be improved as follows:



Gibbs distribution

The prior distribution $P(x)$ can be interpreted as a Gibbs energy. Then, x is a Markov random field if $P(x)$ is a Gibbs distribution such that:

$$P(x) = \frac{1}{Z}\exp(-U(x)) \tag{3.41}$$

where Z is a partition function and $U(x)$ a prior energy such that:

$$U(x) = \sum_{c \in \mathcal{C}} V_c(x) \tag{3.42}$$

where $V_c(x)$ are potentials functions and c is a clique which is defined as a pair of adjacent pixels.

Potential function and line process

In order to preserve edges, the use of a *line process* was introduced by Geman and Geman [36]. In practice, a line process is defined via a potential function ϕ which has particular properties (see [35]).

Suppose $\phi(u)$ has the following properties on $[0, +\infty[$:

- $\phi(0) = 0$
- $\phi(\sqrt{u})$ concave
- $\lim_{u \to +\infty} \phi(u) = 1$

Then, there exists a function $\psi(l)$ defined on an interval [0,L] such that:

$$\phi(u) = \inf_{0 \le l \le L}(lu^2 + \psi(l)) \tag{3.43}$$

and such that $\psi(l)$ has the properties:

- $\psi(0) = 1$

- $\psi(L) = 0$
- $\psi(l)$ strictly decreasing.

The line process is defined as the derivative of $\phi(\sqrt{u})$:

$$l = \phi'(\sqrt{u}) \tag{3.44}$$

Posterior modeling

In the context of Bayesian estimation, the posterior distribution $P(x \mid y)$ can be written as:

$$P(x \mid y) \propto P(y \mid x)P(x) \tag{3.45}$$

where

$$P(y \mid x) \propto \exp(-U(y \mid x)) \tag{3.46}$$

and

$$P(x) \propto \exp(-U(x)) \tag{3.47}$$

Then, the posterior $P(x \mid y)$ is also a Gibbs energy such that:

$$U(x \mid y) = U(y \mid x) + U(x) \tag{3.48}$$

The traditional choice in image restoration is:

$$U(y \mid x) = \frac{\|y - Hx\|^2}{2\sigma^2} \tag{3.49}$$

In the first order case, the prior $P(x)$ is defined such that:

$$U(x) = \sum_{cliques} \phi(x_r - x_s) \tag{3.50}$$

where ϕ is the potential function and $(x_s - x_r)$ is the difference between the values x_r and x_s of the two neighbors inside the clique c. The solution is estimated by maximizing the posterior distribution $P(x \mid y)$. Then, the energy $U(x \mid y)$ must be minimized.

$$U(x \mid y) = \frac{\|y - Hx\|^2}{2\sigma^2} + \lambda \sum_{cliques} \phi(x_r - x_s) \tag{3.51}$$

where λ is the regularizing parameter.

3.6.2 Application

Minimization of the prior energy

The difficulty is to minimize the non quadratic energy $U(x)$. From the work of Geman and Reynolds [35], the minimization of $U(x)$ is equivalent to minimizing $U(x \mid l)$ such that:

$$\min_x (U(x)) = \min_{x,l} (U(x,l)) \tag{3.52}$$

where l is the vector of the line variables l_c. According to equation (3.43), it follows that $U(x \mid l)$ can be written as:

$$U(x,l) = \sum_{cliques\ c} l_c (x_r - x_s)^2 + \psi(l_c) \tag{3.53}$$

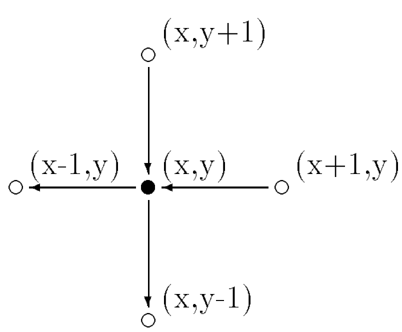
Note that the regularization becomes "half-quadratic" (see [11]) by using equation (3.53):

- With l fixed, $U(x,l)$ is quadratic in x. The minimization in x reduces to the resolution of a linear system.
- With x fixed, the minimum $\hat{l}_c$ is given by the expression $\hat{l}_c = \phi'(u)/2u$ where $u = (x_r - x_s)$.

We suppose that the variables $\hat{l}_c$ do not interact with each other. In fact, these *line variables* map the discontinuities (or the edges) of the image x. $\hat{l}_c$ takes a value near zero at the edges and a value L in homogeneous areas.

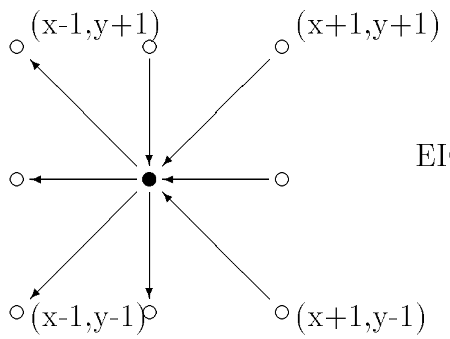
Line variables system

For the first order Markov case, a line variable is labeled as an arrow between two horizontal or vertical and adjacent pixels (x_r, x_s). Then, each arrow is associated with a first order clique.



FOUR-NEIGHBOR SYSTEM

The four neighbor system can be improved by including diagonal adjacencies.



EIGHT-NEIGHBOR SYSTEM

The potential function

Concerning the choice of ϕ, the following convex function proposed in [14] is used:

$$\phi_\delta(u) = |u/\delta| - \ln(1 + |u/\delta|) \tag{3.54}$$

where $u = (x_r - x_s)$ and δ is a scaling parameter. An example is given for $\delta = 500$ in Figure 3.1. It shows that ϕ is quadratic with $u << \delta$ and linear with $u >> \delta$. The derivative is equal to $\phi'_\delta(u) = u/\delta(|u| + \delta)$.

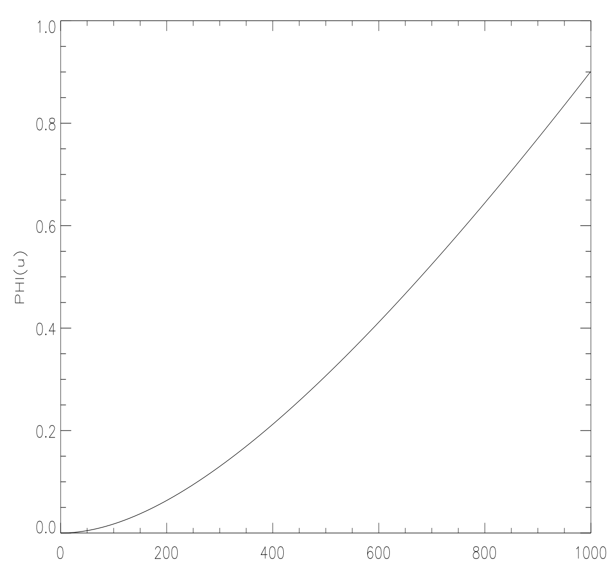


Figure 3.1:

Note that updating the line variables is a simplified operation by using the following formula:

$$\hat{l}_c = \phi'_\delta(u)/2u = \frac{1}{2\delta}/(\delta + |u|) \tag{3.55}$$

The prior energy formula

In the case of the four nearest neighbors, the prior energy is defined as:

$$U(x) = \sum_{vertical\,cliques} \phi_\delta(I_{y,x} - I_{y+1,x}) \tag{3.56}$$

$$+ \sum_{horizontal\,cliques} \phi_\delta(I_{y,x} - I_{y,x+1}) \tag{3.57}$$

where $I_{y,x}$ is the pixel intensity in image I at row y and column x.

By adding the following expressions to the terms in equation (3.57), we obtain the energy $U(x)$ of the eight neighbor system:

$$\sum_{diagonal\,cliques} \phi_\delta(I_{y,x} - I_{y+1,x+1}) \tag{3.58}$$

and

$$\sum_{diagonal\,cliques} \phi_\delta(I_{y,x} - I_{y-1,x+1}) \tag{3.59}$$

Instead of using a stochastic approach, we use a single site update algorithm [14] in order to minimize the half-quadratic criterion (3.51). The solution is computed by visiting the entire set of pixels in a determined fashion (checkerboard) to update each value x_{ij} of the estimated solution x at row i and column j. We know that the energy $U(x)$ is quadratic as a function of x_{ij}. Its minimum value is reached at m_{ij}:

$$m_{ij} = x_{ij} + \frac{[H^t y]_{ij} - [H^t H x]_{ij} - 2\sigma^2 \lambda \sum l_c (x_{ij} - x_c)}{[H^t H]_{ij,ij} + 2\sigma^2 \lambda \sum l_c} \tag{3.60}$$

where the sums extend to the neighborhood of the currently visited pixel x_{ij}. The algorithm is initialized with an image where all pixels are equal to zero.

Parameter estimation

This parameter allows to fix the threshold under which the smoothness of the solution is preserved and above which discontinuities or edges stay in the estimated solution. The value δ is estimated from the observed image y by looking at the evolution of edges between adjacent pixels. In fact, one method of selecting δ would be to determine the global maximal difference between two adjacent pixels for all possible cliques of the observed image. If δ is too high, then the estimated solution will be smooth.

This parameter balances fidelity to the prior constraints and fidelity to the data. In every experiment, λ is empirically tuned to obtain visually good results. In fact, since appropriate values for δ are more or less estimated, the value for λ is not an evident choice. However, if λ is too high, the solution is over-regularized and if λ is too small, the solution is not stabilized.

3.7 Multiscale Entropy Deconvolution

3.7.1 The principle

The standard maximum entropy method (MEM) has been described in the preceding chapter. The most realistic solution is that which minimizes the the amount of information, but remains compatible with the data. By the MEM method, minimizing the information is equivalent to maximizing the entropy and the functional to minimize is

$$J(O) = \sum_{k=1}^{N} \frac{(I_k - (P * O)_k)^2}{2\sigma_I^2} - \alpha H(O) \tag{3.61}$$

where H is either the Frieden or the Gull and Skilling entropy.

Using the Shannon entropy or the multiscale entropy, minimizing the information is equivalent to minimizing the entropy and the functional to minimize is

$$J(O) = \sum_{k=1}^{N} \frac{(I_k - (P * O)_k)^2}{2\sigma_I^2} + \alpha H(O) \tag{3.62}$$

We have seen that in the case of Gaussian noise, H is given by the energy of the wavelet coefficients. We have

$$J(O) = \sum_{k=1}^{N} \frac{(I_k - (P * O)_k)^2}{2\sigma_I^2} + \alpha \sum_{j=1}^{l} \sum_{k=1}^{N_j} \frac{w_{j,k}^2}{2\sigma_j^2} \tag{3.63}$$

where σ_j is the noise at scale j, N_j the number of pixels at the scale j, σ_I the noise standard deviation in the data, and l the number of scales.

Rather than minimizing the amount of information in the solution, we may prefer to minimize the amount of information which can be due to the noise. The function is now:

$$J(O) = \sum_{k=1}^{N} \frac{(I_k - (P * O)_k)^2}{2\sigma_I^2} + \alpha H_n(O) \tag{3.64}$$

Using the N2-MSE approach, H_n is defined by

$$H_n(w_{j,k}) = \sum_{j=1}^{l} \sum_{k=1}^{N_j} \int_0^{|w_{j,k}|} p_n(\mid w_{j,k} \mid - u) (\frac{\partial h(x)}{\partial x})_{x=u} du \tag{3.65}$$

which gives for Gaussian noise

$$H_n(X) = \sum_{j=1}^{l} \sum_{k=1}^{N_j} \frac{1}{\sigma_j^2} \int_0^{|w_{j,k}|} u \ \mathrm{erf}(\frac{\mid w_{j,k} \mid - u}{\sqrt{2}\sigma_j}) \tag{3.66}$$

The solution is found by computing the gradient $\nabla(J(O))$ and performing the following iterative schema:

$$O^{n+1} = O^n - \gamma \nabla(J(O^n)) \tag{3.67}$$

As for the multiscale entropy filtering, there are several ways to use the parameter α. In the same way, we can consider an α_j per scale, and introduce a kind of adaptive regularization, depending on the signal-to-noise ratio of the input data wavelet coefficients.

3.7.2 The parameters

In order to introduce flexibility in the way we restore the data, we introduce two parameters $\beta_{j,k}$ and $\alpha_{j,k}$ which allow us to weight respectively the two terms of the equation to be minimized:

$$J(O) = \frac{1}{2\sigma_I^2} \sum_{k=1}^{N} (c_{l,k}(R) + \sum_{j=1}^{l} \beta_{j,k} w_{j,k}(R))^2 + \sum_{j=1}^{l} \sum_{k=1}^{N_j} \alpha_{j,k} h(w_{j,k}(O))$$

where $R = I - P * O$, and $R = c_{l,k}(R) + \sum_{j=1}^{l} w_{j,k}(R)$ ($w_{j,k}(R)$ are the wavelet coefficients of R using the à trous algorithm, and $w_{j,k}(O)$ are the wavelet coefficients of O).

We consider three approaches for estimating $\beta_{j,k}$

1. no weighting: $\beta_{j,k} = 1$
2. soft weighting: $\beta_{j,k} = p_s(w_{j,k}(I))$
   In this case, $\beta_{j,k}$ is equal to the probability that the input data wavelet coefficient is due to signal (and not to noise).
3. hard weighting: $\beta_{j,k} = 0$ or 1 depending on $p_n(w_{j,k}(I))$ ($p_n(w_{j,k}(I)) = 1. - p_s(w_{j,k}(I))$). This corresponds to using only significant input data wavelet coefficients. If M is the multiresolution support of the input image I, then $\beta_{j,k} = M(j,k)$.

$\alpha_{j,k}$ is the product of three values: $\alpha_{j,k} = \alpha_u \alpha_j \beta'_{j,k}$.

- α_u is a user parameter (defaulted to 1) which allows us to control the smoothness of the solution. Increasing α_u produces a smoother solution.
- α_j is a constant which depends only on the PSF.
- $\beta'_{j,k}$ depends on the input data and can take the following value:

1. no regularization ($\beta'_{j,k} = 0$): only the first term of the functional is minimized.
2. no protection from regularization ($\beta'_{j,k} = 1$): the regularization is applied at all positions and at all the scales.
3. soft protection ($\beta'_{j,k} = p_n(w_{j,k}(I))$): the regularization becomes adaptive, depending on the probability that the input wavelet coefficient is due to noise.
4. hard protection ($\beta'_{j,k} = 0$ or 1 depending on $p_n(w_{j,k}(I))$). If M is the multiresolution support of the input image I, then $\beta_{j,k} = 1 - M(j,k)$.
5. Soft + hard protection: ($\beta'_{j,k} = 0$ or $p_n(w_{j,k}(I))$ depending on $p_n(w_{j,k}(I))$).

We easily see that choosing a hard weighting and no regularization leads to deconvolution from the multiresolution support.

The problem is now to find how to calculate the α_j values. Taking all α_j equal to 1 would mean that we try to restore all bands in the same way. However if the PSF is very large, structures which may appear roughly the size of the pixel are certainly artifacts due to the noise. At the contrary, if the PSF is very extended, there is no reason to not consider as true a compact structure in the solution. This reason leads us to consider that the regularization should not be applied in the same way at all scales, but should depend on the PSF. Since at each iteration the solution is updated by adding a part the residual (which contains the noise) convolved by the PSF, we estimate α_j by the following method:

- we generate a random noise N_r of standard deviation equal to 1.
- we convolve it by the PSF: $C_r = N_r * P$.
- we take the wavelet transform of C_r.
- we take the wavelet transform of N_r.
- we calculate the standard deviation at each scale of C_r: $\sigma^r_j(w_j(C_r))$
- we calculate the standard deviation at each scale of N_r: $\sigma^e_j(w_j(N_r))$
- we calculate the value: $a_j = \frac{\sigma^e_j}{\sigma^r_j}$
- we normalize a_j by $a_j = a_j / max(a_i)$ (for $i = 1..l$).

By this method, the regularization will be stronger at small scales if the PSF is large.

The Intrinsic Correlation Function

In many cases, there is no sense to try to deconvolve an image at the resolution of the pixel (especially when the PSF is very large). The idea to limit the resolution is relatively old, because it is already this concept which is used in the CLEAN algorithm [40]. Indeed the Clean-Beam fixes the resolution in the final solution. This principle was also developed by Lannes [48] under a different form. Finally this concept have been re-invented, first by Gull and Skilling, who have called the Clean-Beam the *Intrinsic Correlation Function* (ICF), and more recently by Magain [53].

The ICF is usually a Gaussian, but in some cases it may be useful to take another function. For example, if we want to compare two images I_1 and I_2 which are obtained with two wavelengths or with two different instruments, their PSFs P_1 and P_2 will certainly be different. The classic way would be convolve I_1 with P_2 and I_2 with P_1, so we are sure that both are at the same resolution. But unfortunately, we lose some resolution. Deconvolving both images is generally not possible because we can never be sure that both solutions O_1 and O_2 will have the same resolution.

A solution would be to deconvolve only the image which has the worse resolution (say I_1), and to limit the deconvolution to the second image resolution (I_2). Then, we just have to take P_2 for the ICF. The deconvolution problem is to find $\tilde{O}$ (hidden solution) such that:

$$I_1 = P_1 * P_2 * \tilde{O} \tag{3.68}$$

and our real solution O_1 at the same resolution as I_2 is just obtained by convolving $\tilde{O}$ by P_2. O_1 and I_2 can then be compared.

Introducing an ICF G in the deconvolution equation leads to just considering a new PSF P' which is the convolution between P and G. The deconvolution is carried out using P', and the solution must be reconvolved by G at the end. By this way, the solution has a limited resolution, but aliasing may occur during the iterative process, and it is not sure that the artefacts will disapear afer the re-convolution with G. Magain [53] has proposed an original alternative to this problem, by assuming that the PSF can be consider as the convolution product of tho terms, the ICF G and an unknow S, $P = G * S$. Using S instead of P in the deconvolution process, and a suffisant large FWHM value for G imply that the Shannon sampling theorem is never violated. But the problem is now to calculate S, knowing P and G, which is again a deconvolution problem. Unfortunately, this delicate point was not discussed in the original paper. Propagation of the error on S estimation in the final solution has also until now never been investigated, but this approach seems promising.

ICF calculation

This section describes how to calculate the Full Width at Half Maximum for a given sampling, in order to not violate the Shannon sampling theorem. Gaussian functions are generally chosen for the ICF. The resolution to achieve is fixed by its standard deviation, or its Full Width at Half Maximum (FWHM=2.34σ). As the Fourier transform of a Gaussian of standard deviation σ is also a Gaussian of standard deviation $\sigma_\nu = \frac{N}{2\pi\sigma}$, ($N$ being the number of pixels), we can estimate the smallest FWHM which does not violate the Shannon sampling theorem. It is impossible in theory, but in practice we can consider that values under a given ϵ have no computation effect. The Shannon theorem is experimentally respected if

$$\exp \frac{\nu^2}{2\sigma_\nu^2} < \epsilon \text{ when } u > \frac{N}{2} \tag{3.69}$$

For $u = \frac{N}{2}$, we have: $\exp \frac{-\pi^2 \sigma^2}{2} < \epsilon$
Then the smallest ICF standard deviation σ_g is given by

$$\sigma_g = \sqrt{\frac{-2 \log \epsilon}{\pi^2}} \tag{3.70}$$

Table 3.1 gives the σ_g values for different values of ϵ. If the resolution to achieve is smaller than σ_g, it means that the solution sampling must be fainter than the data sampling.

Table 3.1: ICF Standard deviation.

ϵ	σ_g	FWHM
10^{-3}	1.18	2.77
10^{-4}	1.37	3.20
10^{-5}	1.53	3.57
10^{-7}	1.81	4.23
10^{-10}	2.16	5.05
10^{-20}	3.05	7.15

Chapter 4

Multiscale Entropy applied to Background Fluctuation Analysis

The mean entropy vector

The multiscale entropy has been defined by:

$$H(X) = \sum_{j=1}^{l} \sum_{k=1}^{N} h(w_j) \tag{4.1}$$

with $h(w_j = \ln(p(w_j(k))))$. In order to study the behavior of the information at a given scale, we prefer to calculate the mean entropy vector E defined by:

$$E(j) = \frac{1}{N} \sum_{k=1}^{N} h(w_j) \tag{4.2}$$

$E(j)$ gives the mean entropy at the scale j. From the mean entropy vector, we have statistical information on each scale separately. Having a noise model, we are able to calculate (generally from simulations) the mean entropy vector $E^{(noise)}(j)$ resulting from a pure noise. Then we define the normalized mean entropy vector by

$$E_n(j) = \frac{E(j)}{E^{(noise)}(j)} \tag{4.3}$$

Figure 4.1 shows the result of a simulation. Five simulated images were created by adding n sources to an 1024×1024 image containing Gaussian noise of standard deviation equal to 1. The n sources are identical, with a maximum equal to 1, and standard deviation equal to 2. Defining the signal to noise ratio (SNR) as the ratio between the standard deviation in the smallest box which contains at least 90% of the flux of the source, and the noise standard deviation, we have a SNR equal to 0.25. The sources are not detectable in the simulated image, nor in its wavelet transform. Figure 4.2 shows a region which contains a source at the center. It is clear there is no way to find this kind of noisy signal. The five images were created using a number of sources respectively equal to 0,50,100,200 and 400, and the simulation was repeated ten times with different noise maps in order to have an error bar on each entropy measurement. For the image which contains 400 sources, the number of pixels affected by a source is less than 2.5%.

When the number of sources increases, the difference beteen the multiscale entropy curves increases. Even if the sources are very faint, the presence of signal can be clearly detected using the mean entropy vector. But it is obvious that the positions of these sources remain unknown.

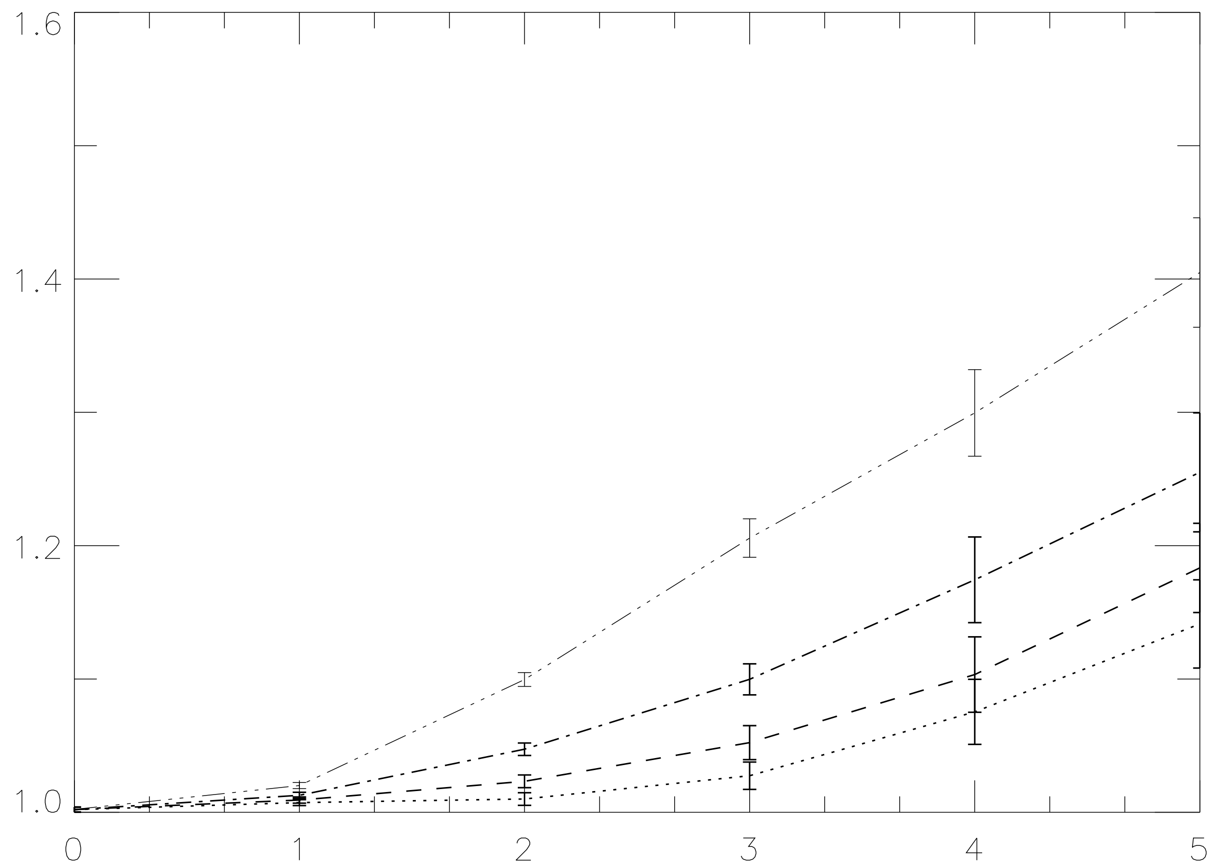


Figure 4.1: Mean entropy versus the scale of 5 simulated images containing undetectable sources and noise. Each curve corresponds to the multiscale transform of one image. From top to bottom, the image contains respectively 400,200,100, 50 and 0 sources.

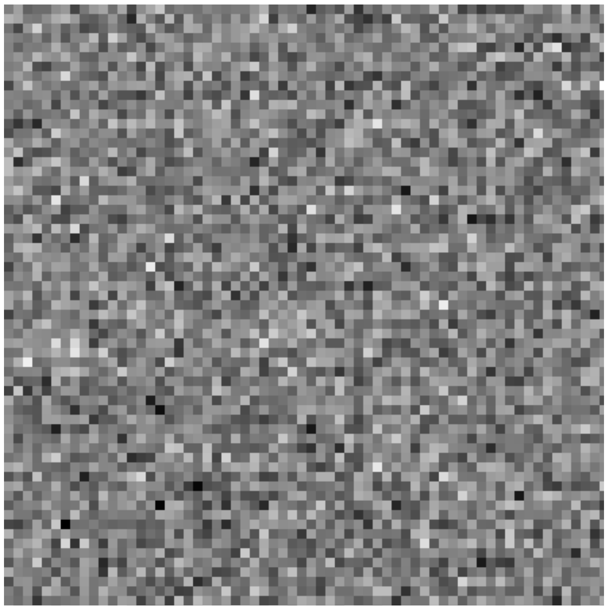


Figure 4.2: Region of a simulated image containing an undetectable source at the center.

Chapter 5

Multiscale Entropy as a Measure of Relevant Information in an Image

Since the multiscale entropy extracts the information from the signal only, it was a challenge to see if the astronomical content of an image was related to its multiscale entropy.

For this purpose, we studied the astronomical content of 200 images of 1024×1024 pixels extracted from scans of 8 different plates carried out by the MAMA facility (Paris, France) [38] and stored at CDS (Strasbourg, France) in the Aladin archive [12]. We estimated the content of these images in three different ways:

1. By counting the number of objects in an astronomical catalog (USNO A2.0 catalog) within the image. The USNO (United States Naval Observatory) catalog was obtained by source extraction from the same survey plates as we used in our study.

2. By counting the number of objects estimated in the image by the Sextractor object detection package [7]. As in the case of USNO these detections are mainly point sources (stars, as opposed to spatially extended objects like galaxies).

3. By counting the number of structures detected at several scales using the MR/1 multiresolution analysis package [61].

Figs. 5.1 show the results of plotting these numbers for each image against the multiscale signal entropy of the image. The best results are obtained using the MR/1 package, followed by Sextractor and then by the number of sources extracted from USNO. Of course the latter two basically miss the content at large scales, which is taken into account by MR/1.

Sextractor and multiresolution methods were also applied to a set of CCD images from CFH UH8K, 2MASS and DENIS near infrared surveys. Results obtained were very similar to what was obtained above. This seems to point to multiscale entropy as being a universal measurement of image content.

Subsequently we looked for the relation between the multiscale entropy and the optimal compression rate of an image which we can obtain by multiresolution techniques [84]. By optimal compression rate we mean a compression rate which allows all the sources to be preserved, and which does not degrade the astrometry and photometry. Louys et al. [51] and Couvidat [24] have estimated this optimal compression rate using the compression program of the MR/1 package [61].

Fig. 5.2 shows the relation obtained between the multiscale entropy and the optimal compression rate for all the images used in our previous tests including CCD ones. The power law relation is obvious thus allowing us to conclude that:

- The compression rate depends strongly on the astronomical content of the image. We can then say that compressibility is also an estimator of the content of the image.

- The multiscale entropy allows us to predict the optimal compression rate of the image.

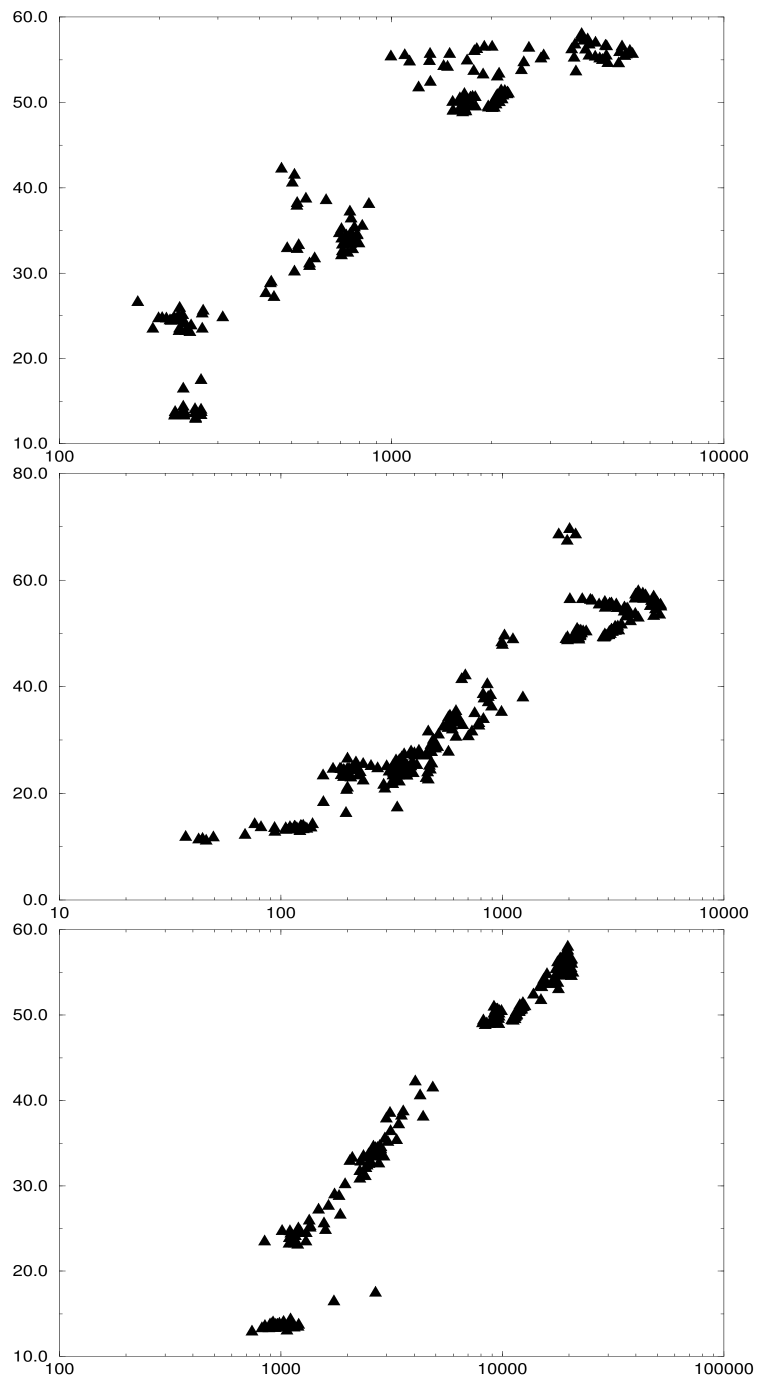


Figure 5.1: Multiscale entropy versus the number of objects: the number of objects is, respectively, obtained from (top) the USNO catalog, (middle) the Sextractor package, and (bottom) the MR/1 package.

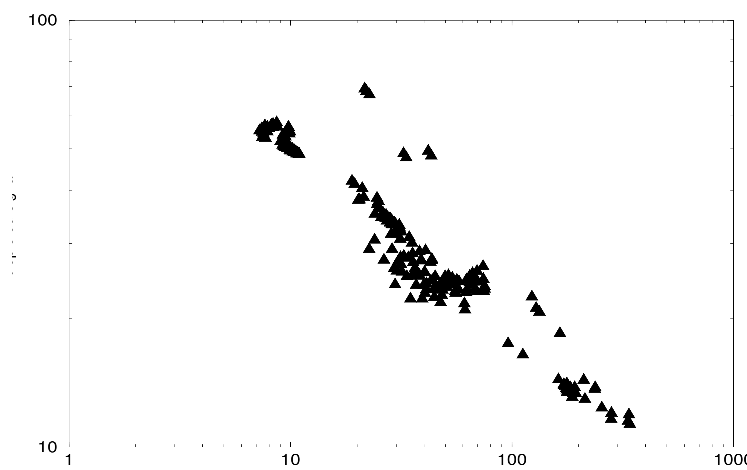


Figure 5.2: Multiscale entropy of astronomical images versus the optimal compression ratio. Images which contain a high number of sources have a small ratio and a high multiscale entropy value. The relation is almost linear.

Chapter 6

Conclusion

We have seen that information must be measured from the transformed data, and not from the data itself. This approach has been used in fact for several years in the domain of image compression. Indeed, modern image compression methods consist firstly of applying a transformation (cosine transform for JPEG, wavelet transform, etc.) to the image, and then coding the coefficients obtained. A good transform for image compression is obviously an orthogonal transform because there is no redundancy, and the number of pixels is the same as in the original image. The exact number of bits necessary to code the coefficients is given by the Shannon entropy. For signal restoration, the problem is not to reduce the number of bits in the representation of the data, and we prefer to use a non-orthogonal wavelet transform, which avoids artifacts in reconstruction due to undersampling.

We could have used the Shannon entropy to measure the information at a given scale, and derive the bins of the histogram from the standard deviation of the noise, but for several reasons we thought it better to directly introduce noise probability into our information measure. Firstly, we have seen that this leads, for Gaussian noise, to a very physical relation between the information and the wavelet coefficients: information is proportional to the energy of the wavelet coefficients normalized by the standard deviation of the noise. Secondly, it works even in the case of images with few photons/events (the histograms in this case present a bias). We have seen that the equations are easy to manipulate. Finally, experiments have confirmed that this approach gives good results. We have seen also that our new information measure leads naturally to a new method for signal restoration. We are now experimenting with this method, and working on generalizations to other classes of noise.

Bibliography

[1] J.G. Ables, 1974.

[2] E.H. Adelson, E. Simoncelli, and R. Hingorani. Optimal image addition using the wavelet transform. *SPIE Visual Communication and Image Processing II*, 845:50–58, 1987.

[3] U. Amato and D.T. Vuza. Besov regularization, thresholding and wavelets for smoothing data. *Numerical Functional Analysis and Optimization*, 18:461–493, 1997.

[4] U. Amato and D.T. Vuza. Wavelet approximation of a function from samples affected by noise. *Revue Roumaine Math. Pures Appl.*, 1998. to appear.

[5] F.J. Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 15:246–254, 1948.

[6] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, 1992.

[7] E. Bertin and S. Arnouts. Sextractor: Software for source extraction. *Astronomy and Astrophysics, Suppl. Ser.*, 117:393–404, June 1996.

[8] A. Bijaoui. *Introduction au Traitement Numérique des Images.* Masson, Paris, 1984.

[9] A. Bijaoui, P. Bury, and E. Slezak. Catalog analysis with multiresolution insights. 1. detection and characterization of significant structures. Technical report, Observatoire de la Côte d'Azur, 1994.

[10] A. Bijaoui, J.L. Starck, and F. Murtagh. Restauration des images multi-échelles par l'algorithme à trous. *Traitement du Signal*, 3:11, 1994.

[11] L. Blanc-Féraud and M. Barlaud. Edge preserving restoration of astrophysical images. *Vistas in Astronomy*, vol. 40(no. 4):pp. 531–538, 1996.

[12] F. Bonnarel, P. Fernique, F. Genova, J. G. Bartlett, O. Bienaymé, D. Egret, J. Florsch, H. Ziaeepour, and M. Louys. ALADIN: A reference tool for identification of astronomical sources. *Astronomical Data Analysis Software and Systems VIII, ASP Conference Series, Vol. 172. Ed. David M. Mehringer, Raymond L. Plante, and Douglas A. Roberts. ISBN: 1-886733-94-5 (1999), p. 229.*, 8:229–232, 1999.

[13] Tj.R. Bontekoe, E. Koper, and D.J.M. Kester. Pyramid maximum entropy images of IRAS survey data. *Astronomy and Astrophysics.*, 294:1037–1053, 1994.

[14] S. Brette and J. Idier. Optimized single site update algorithms for image deblurring. *Proceedings of IEEE ICIP*, pages 65–68, 1996.

[15] J.P. Burg. Annual Meeting International Society Exploratory Geophysics, Reprinted in Modern Spectral Analysis, D.G. Childers, ed., IEEE Press, New York, 34–41, 1978.

[16] P. Bury. *De la distribution de matière à grande échelle à partir des amas d'Abell.* PhD thesis, Université de Nice Sophia Antipolis, Janvier 1995.

[17] A. Chambolle, R.A. DeVore, N. Lee, and B.J. Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Signal Processing*, 7(3):319–335, 1998.

[18] M.K. Charter. Drug absorption in man, and its measurement by MaxEnt. In *Maximum Entropy and Bayesian Methods*, pages 325–339. Kluwer, 1989.

[19] H.A. Chipman, E.D. Kolaczyk, and R.E. McCulloch. Adaptive Bayesian wavelet shrinkage. *Journal of the American Statistical Association*, 92(440):1413–1421, 1997.

[20] C.H. Chui. *Wavelet Analysis and Its Applications.* Academic Press, 1992.

[21] P.H. Van Cittert. Zum Einfluß der Spaltbreite auf die Intensitätsverteilung in Spektrallinien ii. *Zeitschrift für Physik*, 69:298–308, 1931.

[22] A. Cohen, I. Daubechies, and J.C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications in Pure and Applied Mathematics*, 45:485–560, 1992.

[23] T.J. Cornwell. Image restoration. In D.M. Alloin and J.M. Mariotti, editors, *Diffraction-Limited Imaging with Very Large Telescopes*, Dordrecht, 1989. Kluwer.

[24] S. Couvidat. DEA Dissertation, Stasbourg Observatory, 1999.

[25] M. Crouse, R. Nowak, and R. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46:886–902, 1998.

[26] I. Daubechies. Orthogonal bases of compactly supported wavelets. *Communications in Pure and Applied Mathematics*, 41:909–996, 1988.

[27] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1992.

[28] D.L. Donoho. Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data. *Proceedings of Symposia in Applied Mathematics*, 47, 1993.

[29] D.L. Donoho. Translation-invariant de-noising. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*. Springer-Verlag, 1995.

[30] D.L. Donoho and I.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. Technical Report 400, Stanford University, 1993.

[31] J.C. Feauveau. *Analyse multirésolution par ondelettes non-orthogonales et bancs de filtres numériques*. PhD thesis, Université Paris Sud, 1990.

[32] B.R. Frieden. *Image Enhancement and Restoration*. Springer-Verlag, Berlin, 1978.

[33] B.R. Frieden. *Probability, Statistical Optics, and Data Testing: a problem solving approach*. John Wiley & Sons, Berlin, springer-verlag edition, 1978.

[34] N.P. Galatsanos and A.K. Katsaggelos. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. *IEEE Transactions on Image Processing*, 1:322–336, 1992.

[35] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14(no. 3):367–383, 1992.

[36] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6(no. 6):721–741, 1984.

[37] G.H. Golub, M. Heath, , and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

[38] J. Guibert. The MAMA facility - a survey of scientific programmes. In *Digitised Optical Sky Surveys*, pages 103+, 1992.

[39] S.F. Gull and J. Skilling. *MEMSYS5 Quantified Maximum Entropy User's Manual*, 1991.

[40] J.A. Högbom. Aperture synthesis with a non-regular distribution of interferometer baselines. *Astronomy and Astrophysics Supplement Series*, 15:417–426, 1974.

[41] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. *Wavelets: Time-Frequency Methods and Phase-Space*, chapter A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform, pages 286–297. Springer-Verlag, 1989.

[42] M. Jansen and D. Roose. Bayesian correction of wavelet threshold procedures for image de-noising. In *Proceedings of the Joint Statistical Meeting, Bayesian Statistical Science*, 1998. to appear.

[43] P.A. Jansson, R.H. Hunt, and E.K. Peyler. Resolution enhancement of spectra. *Journal of the Optical Society of America*, 60:596–599, 1970.

[44] E.T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:171–190, 1957.

[45] A. K. Katsaggelos. *Digital Image Restoration*. Springer-Verlag, Berlin, 1991.

[46] H. Krim, K. Drouiche, and J. C. Pesquet. Multiscale detection of nonstationary signals. In *Time-Frequency and Time-Scale Analysis*, pages 105–108, Victoria, B.C., Canada, 1992. IEEE Signal Processing Society.

[47] L. Landweber. An iteration formula for Fredholm integral equations of the first kind. *American Journal of Mathematics*, 73:615–624, 1951.

[48] A. Lannes and S. Roques. Resolution and robustness in image processing: a new regularization principle. *Journal of the Optical Society of America*, 4(1), 1987.

[49] J. Llacer and J. Núñez. Iterative maximum likelihood estimator and Bayesian algorithms for image reconstruction in astronomy. In R.L. White and R.J. Allen, editors, *The Restoration of HST Images and Spectra*, Baltimore, 1990. Space Telescope Science Institute.

[50] H. Lorenz and G.M. Richter. Adaptive filtering of HST images: preprocessing for deconvolution. In P. Benvenuti and E. Schreier, editors, *Science with the Hubble Space Telescope*, pages 203–206, Garching, 1993. European Southern Observatory.

[51] M. Louys, J.-L. Starck, S. Mei, F. Bonnarel, and F. Murtagh. Astronomical image compression. *Astronomy and Astrophysics, Suppl. Ser.*, 136:579–590, 1999.

[52] L. B. Lucy. An iteration technique for the rectification of observed distributions. *Astronomical Journal*, 79:745–754, 1974.

[53] P. Magain, F. Courbin, and S. Sohy. Deconvolution with correct sampling. *Astrophysical Journal*, 494:472, 1998.

[54] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[55] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[56] S. G. Mallat. Zero crossings of a wavelet transform. *IEEE Transactions on Information Theory*, 37(4):1019–1033, 1991.

[57] A. Mohammad-Djafari. Maximum entropie et problèmes inverses en imagerie. *Traitement du Signal*, pages 87–116, 1994.

[58] A. Mohammad-Djafari. Entropie en traitement du signal. *Traitement du Signal*, 15(6):545–551, 1998.

[59] R. Molina. On the hierarchical Bayesian approach to image restoration. Applications to astronomical images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1122–1128, 1994.

[60] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized gaussian and complexity priors. *IEEE Transactions on Information Theory*, 45:909–919, 1999.

[61] MR/1. Multiresolution Image and Data Analysis Software Package, Version 2.0, Multi Resolutions Ltd., (http://www.multiresolution.com), 1999.

[62] F. Murtagh and J.L. Starck. Multiresolution image analysis using wavelets: Some recent results and some current directions. *ST-ECF Newsletter*, 21:19–20, 1994.

[63] F. Murtagh, J.L. Starck, and A. Bijaoui. Image restoration with noise suppression using a multiresolution support. *Astronomy and Astrophysics, Suppl. Ser.*, 112:179–189, 1995.

[64] R. Narayan and R. Nityananda. Maximum entropy image restoration in astronomy. *Ann. Rev. Astron. Astrophys.*, 24:127–170, 1986.

[65] G. P. Nason. Wavelet regression by cross-validation. Technical report, Dept. of Math., Univ. of Bristol, University Walk, Bristol, BS8 1TW, U.K., Mar. 24, 1994.

[66] G.P. Nason. Wavelet shrinkage by cross-validation. *Journal of the Royal Statistical Society B*, 58:463–479, 1996.

[67] S.I. Olsen. Estimation of noise in images: An evaluation. *CVGIP*, 55:319–323, 1993.

[68] E. Pantin and J.L. Starck. Deconvolution of astronomical images using the multiscale maximum entropy method. *Astronomy and Astrophysics, Suppl. Ser.*, 315:575–585, 1996.

[69] W.H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62:55–59, 1972.

[70] M.M. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael. *Wavelets and Their Applications*. Jones and Barlett, 1992.

[71] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchival trees. *IEEE Trans. on Circ. and Syst. for Video Tech.*, 6(3):243–250, 1996.

[72] C.E. Shannon. A mathematical theory for communication. *Bell System Tech J.*, 27:379–423, 1948.

[73] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Process.*, 41(12):3445–3462, 1993.

[74] M.J. Shensa. Discrete wavelet transforms: wedding the à trous and mallat algorithms. *IEEE Transactions on Signal Processing*, 1992.

[75] L.A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, MI-1:113–122, 1982.

[76] J. Skilling. Classic maximum entropy. In *Maximum Entropy and Bayesian Methods*, pages 45–52. Kluwer, 1989.

[77] E. Slezak, V. de Lapparent, and A. Bijaoui. Objective detection of voids and high density structures in the first CfA redshift survey slice. *Astrophysical Journal*, 409, 1993.

[78] J.L. Starck. *Analyse en ondelettes et Haute Résolution Angulaire*. PhD thesis, Université de Nice Sophia Antipolis, 4 December 1992.

[79] J.L. Starck. The wavelet context. Technical report, European Southern Observatory – Image Processing Group, Release 93NOV 1993.

[80] J.L. Starck and A. Bijaoui. Filtering and deconvolution by the wavelet transform. *Signal Processing*, 35:195–211, 1994.

[81] J.L. Starck, A. Bijaoui, and F. Murtagh. Multiresolution support applied to image filtering and deconvolution. *CVGIP: Graphical Models and Image Processing*, 57:420–431, 1995.

[82] J.L. Starck and F. Murtagh. Image restoration with noise suppression using the wavelet transform. *Astronomy and Astrophysics*, 288:343–348, 1994.

[83] J.L. Starck and F. Murtagh. Automatic noise estimation from the multiresolution support. *Publications of the Astronomical Society of the Pacific*, 110(744):193–199, 1998.

[84] J.L. Starck, F. Murtagh, and A. Bijaoui. *Image Processing and Data Analysis: The Multiscale Approach*. Cambridge University Press, Cambridge (GB), 1998.

[85] J.L. Starck, F. Murtagh, and R. Gastaud. A new entropy measure based on the wavelet transform and noise modeling. Special Issue on Multirate Systems, Filter Banks, Wavelets, and Applications of *IEEE Transactions on CAS II*, 45(8), 1998.

[86] J.L. Starck, F. Murtagh, B. Pirenne, and M. Albrecht. Astronomical image compression based on noise suppression. *Publications of the Astronomical Society of the Pacific*, 108:446–455, 1996.

[87] G. Strang and T. Nguyen. *Wavelet and Filter Banks*. Wellesley-Cambridge Press, Box 812060, Wellesley MA 02181, fax 617-253-4358, 1996.

[88] A.N. Tikhonov, A.V. Goncharski, V.V. Stepanov, and I.V. Kochikov. Ill-posed image processing problems. *Soviet Physics – Doklady*, 32:456–458, 1987.

[89] D.M. Titterington. General structure of regularization procedures in image reconstruction. *AA*, 144:381–387, 1985.

[90] B.P. Wakker and U.J. Schwarz. The multi-resolution Clean and its application to the short-spacing problem in interferometry. *Ann. Rev. Astron. Astrophys.*, 200:312, 1988.

[91] N. Weir. Application of maximum entropy techniques to hst data. In *3rd ESO/ST-ECF Data Analysis Workshop*, 1991.

[92] N. Weir. A multi-channel method of maximum entropy image restoration. In D.M. Worral, C. Biemesderfer, and J. Barnes, editors, *Astronomical Data Analysis Software and System 1*, pages 186–190. Astronomical Society of the Pacific, 1992.

Appendix A: The "À Trous" Wavelet Transform Algorithm

In a wavelet transform, a series of transformations of a signal is generated, providing a resolution-related set of "views" of the signal. The properties satisfied by a wavelet transform, and in particular by the *à trous* wavelet transform, are further discussed by Bijaoui et al. [10]. Extensive literature exists on the wavelet transform and its applications ([26, 20, 70, 84]). The discrete *à trous* algorithm is described in [41, 74].

We consider spectra, $\{c_0(k)\}$, defined as the scalar product at samples k of the function $f(x)$ with a scaling function $\phi(x)$ which corresponds to a low pass filter:

$$c_0(k) = < f(x), \phi(x-k) > \tag{6.1}$$

The scaling function is chosen to satisfy the dilation equation:

$$\frac{1}{2}\phi(\frac{x}{2}) = \sum_l h(l)\phi(x-l) \tag{6.2}$$

where h is a discrete low-pass filter associated with the scaling function ϕ. This means that a low-pass filtering of the signal is, by definition, closely linked to another resolution level of the signal. The distance between levels increases by a factor 2 from one scale to the next.

The smoothed data $c_j(k)$ at a given resolution j and at a position k is the scalar product

$$c_j(k) = \frac{1}{2^j} < f(x), \phi(\frac{x-k}{2^j}) > \tag{6.3}$$

This is consequently obtained by the convolution:

$$c_j(k) = \sum_l h(l) \quad c_{j-1}(k + 2^{j-1}l) \tag{6.4}$$

The signal difference w_j between two consecutive resolutions is:

$$w_j(k) = c_{j-1}(k) - c_j(k) \tag{6.5}$$

or:

$$w_j(k) = \frac{1}{2^j} < f(x), \psi(\frac{x-k}{2^j}) > \tag{6.6}$$

Here, the wavelet function ψ is defined by:

$$\frac{1}{2}\psi(\frac{x}{2}) = \phi(x) - \frac{1}{2}\phi(\frac{x}{2}) \tag{6.7}$$

Equation 6.6 defines the discrete wavelet transform, for a resolution level j.

For the scaling function, $\phi(x)$, the B-spline of degree 3 was used in our calculations. As a filter we use $h = (\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$. See Starck (1993) for discussion of linear and other scaling functions. Here we have derived a simple algorithm in order to compute the associated wavelet transform:

1. We initialize j to 0 and we start with the data $c_j(k)$.
2. We increment j, and carry out a discrete convolution of the data $c_{j-1}(k)$ using the filter h. The distance between the central sample and the adjacent ones is 2^{j-1}.
3. After this smoothing, we obtain the discrete wavelet transform from the difference $c_{j-1}(k) - c_j(k)$.
4. If j is less than the number p of resolutions we want to compute, then we go to step 2.
5. The set $\mathcal{W} = \{w_1, ..., w_p, c_p\}$ represents the wavelet transform of the data.

A series expansion of the original signal, c_0, in terms of the wavelet coefficients is now given as follows. The final smoothed array $c_p(x)$ is added to all the differences w_j:

$$c_0(k) = c_p + \sum_{j=1}^{p} w_j(k) \tag{6.8}$$

This equation provides a reconstruction formula for the original signal. At each scale j, we obtain a set $\{w_j\}$ which we call a wavelet scale. The wavelet scale has the same number of samples as the signal.

Appendix B: Noise Modeling the Wavelet Space

Gaussian noise

We have the probability density

$$p(w_j(x, y)) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-w_j(x,y)^2/2\sigma_j^2} \tag{6.9}$$

So we need to estimate, in the case of Gaussian noise models, the noise standard deviation at each scale. These standard deviations can be determined analytically in the case of some transforms, including the à trous transform, but the calculations can become complicated.

The appropriate value of σ_j in the succession of wavelet planes is assessed from the standard deviation of the noise σ_I in the original image I, and from study of the noise in the wavelet space. This study consists of simulating an image containing Gaussian noise with a standard deviation equal to 1, and taking the wavelet transform of this image. Then we compute the standard deviation σ_j^e at each scale. We get a curve σ_j^e as a function of j, giving the behavior of the noise in the wavelet space. (Note that if we had used an orthogonal wavelet transform, this curve would be linear.) Due to the properties of the wavelet transform, we have $\sigma_j = \sigma_I \sigma_j^e$. The standard deviation of the noise at a scale j of the image is equal to the standard deviation of the noise of the image multiplied by the standard deviation of the noise of scale j of the wavelet transform.

Poisson noise

If the noise in the data I is Poisson, the Anscombe transform

$$t(I(x, y)) = 2\sqrt{I(x, y) + \frac{3}{8}} \tag{6.10}$$

acts as if the data arose from a Gaussian white noise model (Anscombe, [5]), with $\sigma = 1$, under the assumption that the mean value of I is large.

For Poisson parameter values under about 20, the Anscombe transformation looses control over the bias. In this case, an alternative approach to variance stabilization is needed. An approach for very small numbers of counts, including frequent zero cases, has been described in in [77], [9] and [16], and will be described below. Small numbers of detector counts will most likely be associated with the image background. Note that errors related to small values carry the risk of removing real objects, but not of amplifying noise.

Gaussian and Poisson Noise

The arrival of photons, and their expression by electron counts, on CCD detectors may be modeled by a Poisson distribution. In addition, there is additive Gaussian read-out noise. The Anscombe transformation (eqn. 6.10) has been extended [63] to take this combined noise into account. As an approximation, consider the signal's value, $I(x,y)$, as a sum of a Gaussian variable, γ, of mean g and standard-deviation σ; and a Poisson variable, n, of mean m_0: we set $I(x,y) = \gamma + \alpha n$ where α is the gain.

The generalization of the variance stabilizing Anscombe formula is:

$$t = \frac{2}{\alpha}\sqrt{\alpha I(x,y) + \frac{3}{8}\alpha^2 + \sigma^2 - \alpha g} \qquad (6.11)$$

With appropriate values of α, σ and g, this reduces to Anscombe's transformation (eqn. 6.10).

Poisson Noise with Few Photons or Counts

A wavelet coefficient at a given position and at a given scale j is

$$w_j(x,y) = \sum_{k \in K} n_k \psi\left(\frac{x_k - x}{2^j}, \frac{y_k - y}{2^j}\right) \qquad (6.12)$$

where K is the support of the wavelet function ψ and n_k is the number of events which contribute to the calculation of $w_j(x,y)$ (i.e. the number of photons included in the support of the dilated wavelet centered at (x,y)).

If a wavelet coefficient $w_j(x,y)$ is due to the noise, it can be considered as a realization of the sum $\sum_{k \in K} n_k$ of independent random variables with the same distribution as that of the wavelet function (n_k being the number of photons or events used for the calculation of $w_j(x,y)$). Then we compare the wavelet coefficient of the data to the values which can be taken by the sum of n independent variables.

The distribution of one event in the wavelet space is directly given by the histogram H_1 of the wavelet ψ. Since independent events are considered, the distribution of the random variable W_n (to be associated with a wavelet coefficient) related to n events is given by n autoconvolutions of H_1

$$H_n = H_1 \otimes H_1 \otimes ... \otimes H_1 \tag{6.13}$$

For a large number of events, H_n converges to a Gaussian.

In order to facilitate the comparisons, the variable W_n of distribution H_n is reduced by

$$c = \frac{W_n - E(W_n)}{\sigma(W_n)} \tag{6.14}$$

and the cumulative distribution function is

$$F_n(c) = \int_{-\infty}^{c} H_n(u)du \tag{6.15}$$

From F_n, we derive c_{min} and c_{max} such that $F(c_{min}) = \epsilon$ and $F(c_{max}) = 1 - \epsilon$.

Therefore a reduced wavelet coefficient $w_j^r(x,y)$, calculated from $w_j(x,y)$, and resulting from n photons or counts is significant if:

$$F(w^r) > c_{max} \tag{6.16}$$

or

$$F(w^r) < c_{min} \tag{6.17}$$

and $w_j^r(x,y)$ is obtained by

$$\begin{aligned} w_j^r(x,y) &= \frac{w_j(x,y)}{\sqrt{n}\sigma_{\psi_j}} && (6.18) \\ &= \frac{w_j(x,y)}{\sqrt{n}\sigma_{\psi}} 4^j && (6.19) \end{aligned}$$

where σ_ψ is the standard deviation of the wavelet function, and σ_{ψ_j} is the standard deviation of the dilated wavelet function ($\sigma_{\psi_j} = \sigma_\psi / 4^j$).

Root mean square map

If, associated to the data $I(x,y)$, we have the root mean square map $R_\sigma(x,y)$, the noise in I is non homogeneous. For each wavelet coefficient $w_j(x,y)$ of I, the exact standard deviation $\sigma_j(x,y)$ have to be calculated from R_σ. A wavelet coefficient $w_j(x,y)$ is obtained by the correlation product between the image I and a function g_j:

$$w_j(x,y) = \sum_k \sum_l I(x,y) g_j(x+k, y+l) \tag{6.20}$$

Then we have

$$\sigma_j^2(x,y) = \sum_k \sum_l R_\sigma^2(x,y) g_j^2(x+k, y+l). \tag{6.21}$$

In the case of the à trous algorithm, the coefficients $g_j(x,y)$ are not known exactly, but they can easily be computed by taking the wavelet transform of a Dirac w^δ. The map σ_j^2 is calculated by correlating the square of the wavelet scale j of w^δ by $R_\sigma^2(x,y)$.

Speckle Noise

Speckle occurs in all types of coherent imagery such as synthetic aperture radar (SAR) imagery, acoustic imagery and laser illuminated imagery. The probability density function (pdf) of the modulus of a homogeneous scene is a Rayleigh distribution:

$$p(\rho) = \frac{\rho}{\sigma^2} e^{-\frac{\rho^2}{2\sigma^2}} \quad M_\rho = \sqrt{\frac{\pi}{2}}\sigma \quad \sigma_\rho = \sqrt{\frac{4-\pi}{2}}\sigma$$

The ratio σ_ρ / M_ρ is a constant of value $\sqrt{\frac{4-\pi}{\pi}}$. This means that the speckle is a multiplicative noise. The pdf of the modulus of a log-transformed speckle noise is:

$$p(\ell) = \frac{e^{2\ell}}{\sigma^2} e^{-\frac{e^{2\ell}}{2\sigma^2}} \quad M_\ell = 0.058 + \log(\sigma) \quad \sigma_\ell = \sqrt{\frac{\pi^2}{24}} = 0.641$$

A better estimator for Rayleigh distribution is the energy ($I = \rho^2$) which is known to have an exponential (*i.e.* Laplace) distribution of parameter $a = 2\sigma^2$

$$p(I) = \frac{1}{a} e^{-\frac{I}{a}}$$

Other Types of Noise

For any type of noise, an analogous study can be carried out in order to find the detection level at each scale and at each position. The types of noise considered so far in this chapter correspond to the general cases in astronomical imagery. We now describe briefly methods which can be used for non-uniform and multiplicative noise.

Additive Non-Uniform Noise

If the noise is additive, but non-uniform, we cannot estimate a standard deviation for the whole image. However, we can often assume that the noise is locally Gaussian, and we can compute a local standard deviation of the noise for each pixel. In this way, we obtain a standard deviation map of the noise, $I_\sigma(x,y)$. A given wavelet coefficient $w_j(x,y)$ is calculated from the pixels of the input image I in the range $I(x - l \ldots x+l, y-l \ldots y+l)$ where l is dependent on the wavelet transform algorithm, the wavelet function, and the scale j. An upper limit $u_j(x,y)$ for the noise associated with $w_j(x,y)$ is found by just considering the maximum value in $I_\sigma(x-l \ldots x+l, y-l \ldots y+l)$ and by multiplying this value by the constant σ_j^e (defined in the subsection "Gaussian noise" at the beginning of this Appendix).

$$u_j(x,y) = \max(I_\sigma(x - l \ldots x + l, y - l \ldots y + l))\sigma_j^e \tag{6.22}$$

The detection level is not constant over each scale.

Multiplicative Noise

If the noise is multiplicative, the image can be transformed by taking its logarithm. In the resulting image, the noise is additive, and a hypothesis of Gaussian noise can be used in order to find the detection level at each scale.

Multiplicative Non-Uniform Noise

In this case, we take the logarithm of the image, and the resulting image is treated as for additive non-uniform noise above.

Unknown Noise

If the noise does not follow any known distribution, we can consider as significant only wavelet coefficients which are greater than their local standard deviation multiplied by a constant: $w_j(x,y)$ is significant if

$$\mid w_j(x,y) \mid > \; k\sigma(w_j(x - l \ldots x + l, y - l \ldots y + l)) \tag{6.23}$$