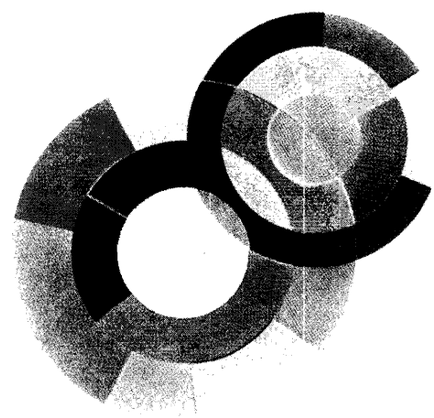
DAPNIA

DAPNIA-02-138

DEC 9 2002



DAPNIA-02-138 11/2002

Image Processing by the Curvelet Tranform

J.L. Starck

Rapport interne

Département d'**A**strophysique, de Physique des **P**articules, de Physique **N**ucléaire et de l'**I**nstrumentation **A**ssociée

DSM/DAPNIA, CEA/Saclay F - 91191 Gif-sur-Yvette Cédex
Tél : (1) 69 08 24 02 Fax : (1) 69 08 99 89
http : //www-dapnia.cea.fr

Couverture : Sylvie Cabrit et Marc Sauvage

Image Processing by the Curvelet Transform

Technical Report

J-L. Starck

DSM/DAPNIA/SEDI-SAP, CEA-Saclay, 91191 Gif-sur-Yvette, France

September 2002 - Version 1.0

Contents

1 Introduction **4**

2 Continuous Ridgelet Transform **6**
2.1 The Radon Transform 7
2.2 Ridgelet Pyramids 7

3 Digital Ridgelet Transform **8**
3.1 The RectoPolar Ridgelet transform 8
3.1.1 A Polar Sampling Scheme for Digital Data 9
3.1.2 Interpolation to rectopolar Grid 9
3.1.3 Exact Reconstruction and Stability 11
3.1.4 One-dimensional Wavelet Transform 12
3.1.5 Combining the Pieces 13
3.2 The Orthonormal Finite Ridgelet Transform 13
3.3 The Slant Stack Ridgelet Transform 14

4 Local Ridgelet Transforms **15**

5 Digital Curvelet Transform **16**
5.1 Discrete Curvelet Transform of Continuum Functions 16
5.2 Digital Realization 18
5.3 Algorithm 18

6 Filtering **21**
6.1 Poisson Observations 21
6.2 Filtering Experiments 21
6.2.1 Lenna Image 21
6.2.2 Recovery of Linear Features 25
6.2.3 Recovery of Curves 25
6.2.4 Denoising of a Color Image 26
6.2.5 Saturn Rings 26
6.2.6 Supernova with Poisson noise 26

7 The Combined Filtering Method **33**
7.1 Introduction 33
7.2 The Combined Filtering Principle 34
7.3 The Minimization Method 34
7.4 Experiments 35
7.5 Discussion 36

8 The Combined Deconvolution Method **40**

9 Contrast Enhancement **42**
9.1 Introduction 42
9.2 Contrast Enhancement using the Curvelet Transform 43
9.3 Discussion 45

10 Morphological Component Analysis **50**
10.1 Introduction 50
10.2 The Combined Transformation 51
10.3 Soft-MGA 51
10.4 Hard-MGA 53
10.5 Experiments 53
10.5.1 Experiment 1: Infrared Gemini Data 53
10.5.2 Experiment 2: A370 56

11 Program **58**
11.1 Ridgelet transform 58
11.1.1 im_radon 58
11.1.2 rid_trans 59
11.1.3 rid_stat 60
11.1.4 rid_filter 61
11.2 Curvelet transform 62
11.2.1 cur_trans 62

11.2.2 cur_stat 63
11.2.3 cur_filter 64
11.2.4 cur_colfilter 65
11.2.5 cur_contrast 66
11.2.6 cur_colcontrast 67
11.3 Combined Filtering 68
11.3.1 cb_filter 68
11.4 Morphological Component Analysis 69
11.4.1 cb_mca 69

Appendix A: Wavelet transform using the Fourier Transform **76**

Appendix B: The " À Trous" Wavelet Transform Algorithm **79**

1 Introduction

Wavelets and related multiscale representations pervade all areas of signal processing. The recent inclusion of wavelet algorithms in JPEG 2000, the new still-picture compression standard, testifies to this lasting and significant impact. The most used wavelet transform algorithm is certainly the decimated bi-orthogonal wavelet transform (OWT). Using the OWT, a signal s can be decomposed by:

$$s(l) = \sum_k c_{J,k}\phi_{J,l}(k) + \sum_k \sum_{j=1}^{J} \psi_{j,l}(k) w_{j,k} \tag{1}$$

with $\phi_{j,l}(x) = 2^{-j}\phi(2^{-j}x - l)$ and $\psi_{j,l}(x) = 2^{-j}\psi(2^{-j}x - l)$, where ϕ and ψ are respectively the scaling function and the wavelet function. J is the number of resolutions used in the decomposition, w_j the wavelet (or details) coefficients at scale j, and c_J is a coarse or smooth version of the original signal s. Thus, the algorithm outputs $J + 1$ subband arrays. The indexing is such that, here, $j = 1$ corresponds to the finest scale (high frequencies).

The application of the OWT to image compression, using the 7-9 filters [2] and the zerotree coding [45, 44] has lead to impressive results, compared to previous methods like JPEG.

A series of recent papers [7, 6], however, argued that wavelets and related classical multiresolution ideas are playing with a limited dictionary made up of roughly isotropic elements occurring at all scales and locations. We view as a limitation the facts that those dictionaries do not exhibit highly anisotropic elements and that there is only a fixed number of directional elements, independent of scale. Despite the success of the classical wavelet viewpoint, there are objects, e.g. images that do not exhibit isotropic scaling and thus call for other kinds of multiscale representation. In short, the theme of this line of research is to show that classical multiresolution ideas only address a portion of the whole range of interesting multiscale phenomena and that there is an opportunity to develop a whole new range of multiscale transforms.

Following on this theme, Candès and Donoho introduced new multiscale systems like curvelets [6] and ridgelets [5] which are very different from wavelet-like systems. Curvelets and ridgelets take the form of basis elements which exhibit very high directional sensitivity and are highly anisotropic. In two-dimensions, for instance, curvelets are localized along curves, in three dimensions along sheets, etc. Continuing at this informal level of discussion we will rely on an example to illustrate the fundamental difference between the wavelet and ridgelet approaches –postponing the mathematical description of these new systems.

We investigate in this technical report the best way to implement the ridgelet and the curvelet transform for the purpose of image restoration. Second and third sections describe respectively the ridgelet transform and the curvelet transform. Section four shows how a the ridgelet and the wavelet coefficients can be thresholded in order to filter an image. Comparisons with other methods are presented. Instruction for using the programs are given in the last section.

Consider an image which contains a vertical band embedded in white noise with relatively large amplitude. Figure 1 (top left) represents such an image. The parameters are as follows: the pixel width of the band is 20 and the SNR is set to be 0.1. Note that it is not possible to distinguish the band by eye. The wavelet transform (undecimated wavelet transform) is also incapable of detecting the presence of this object; roughly speaking, wavelet coefficients correspond to averages over approximately isotropic neighborhoods (at different scales) and

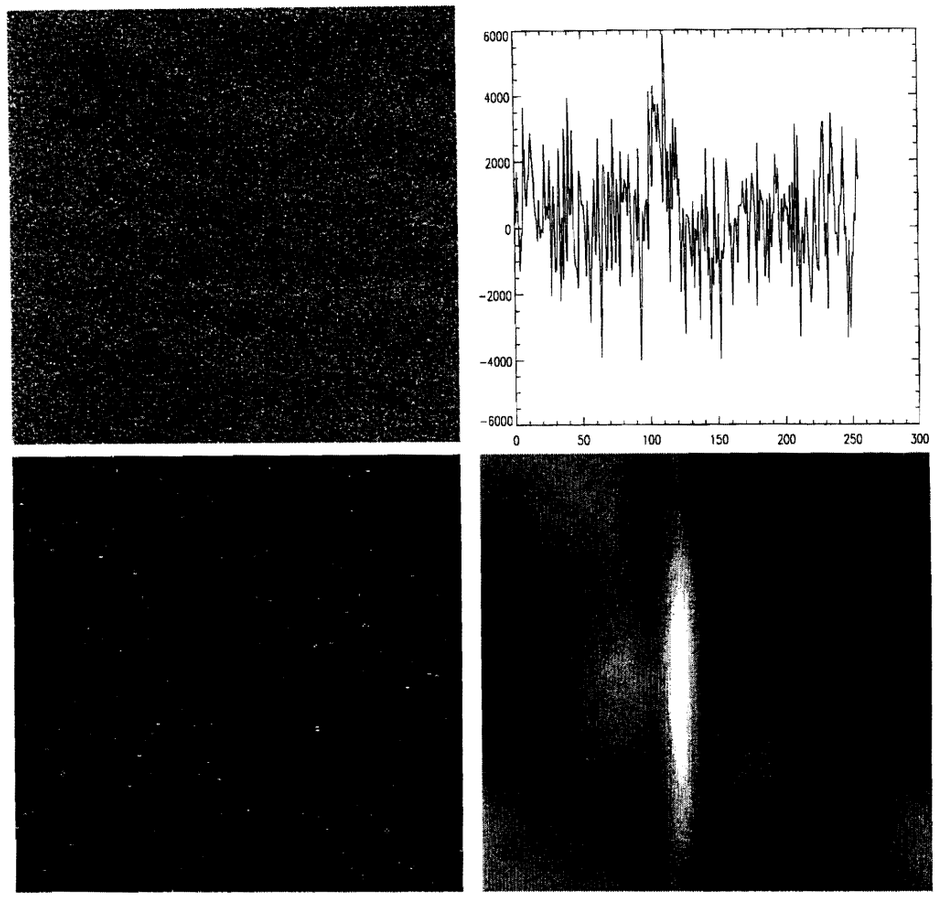


Figure 1: Top, original image containing lines and gaussians. Bottom left, reconstructed image for the à trous wavelet coefficient, bottom right, reconstructed image from the ridgelet coefficients.

those wavelets clearly do not correlate very well with the the very elongated structure (pattern) of the object to be detected.

We now turn our attention towards procedures of a very different nature which are based on line measurements. To be more specific, consider an *ideal* procedure which consists in integrating the image intensity over columns; that is, along the orientation of our object. We use the adjective "ideal" to emphasize the important fact that this method of integration requires a priori knowledge about the structure of our object. This method of analysis gives of course an improved signal to noise ratio for our linear functional better correlate the object in question, see the top right panel of Figure 1.

This example will make our point. Unlike wavelet transforms, the ridgelet transform processes data by first computing integrals over lines with all kinds of orientations and locations. We will explain in the next section how the ridgelet transform further processes those line integrals. For now, we apply naive thresholding of the ridgelet coefficients and "invert" the ridgelet transform; the bottom right panel of Figure 1 shows the reconstructed image. The

qualitative difference with the wavelet approach is striking. We observe that this method allows the detection of our object even in situations where the noise level (standard deviation of the white noise) is five times superior to the object intensity.

2 Continuous Ridgelet Transform

The two-dimensional continuous ridgelet transform in $\mathbf{R}^2$ can be defined as follows [5]. We pick a smooth univariate function $\psi : \mathbf{R} \to \mathbf{R}$ with sufficient decay and satisfying the admissibility condition

$$\int |\hat{\psi}(\xi)|^2/|\xi|^2 \, d\xi < \infty, \tag{2}$$

which holds if, say, ψ has a vanishing mean $\int \psi(t)dt = 0$. We will suppose that ψ is normalized so that $\int |\hat{\psi}(\xi)|^2 \xi^{-2} d\xi = 1$.

For each $a > 0$, each $b \in \mathbf{R}$ and each $\theta \in [0, 2\pi)$, we define the bivariate *ridgelet* $\psi_{a,b,\theta} : \mathbf{R}^2 \to \mathbf{R}^2$ by

$$\psi_{a,b,\theta}(x) = a^{-1/2} \cdot \psi((x_1 \cos\theta + x_2 \sin\theta - b)/a); \tag{3}$$

this function is constant along lines $x_1 \cos\theta + x_2 \sin\theta = const$. Transverse to these ridges it is a wavelet.

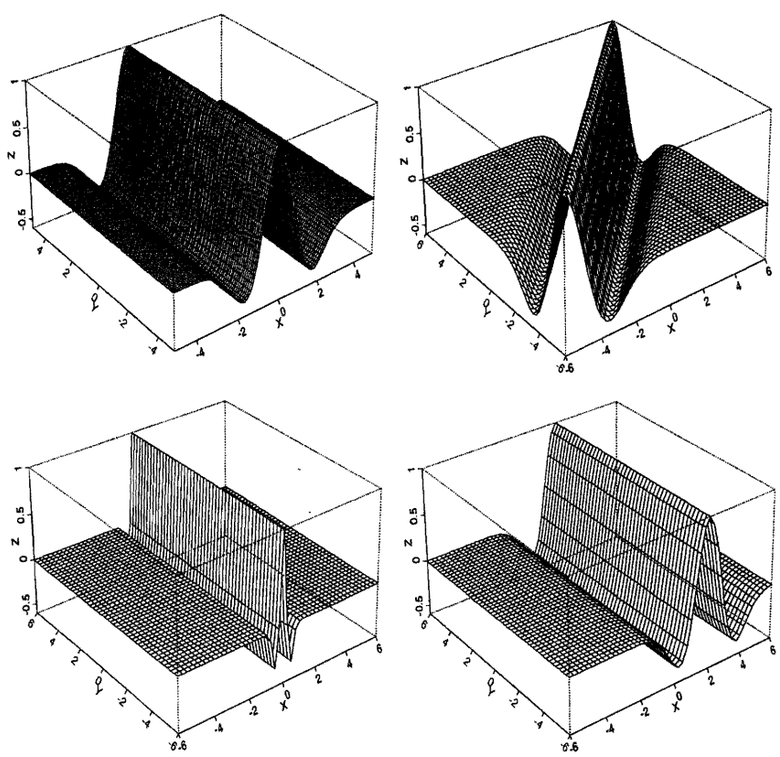


Figure 2: A Few Ridgelets

Figure 2 graphs a few ridgelets with different parameter values. The top right, bottom left

and right panels are obtained after simple geometric manipulations of the upper left ridgelet, namely rotation, rescaling, and shifting.

Given an integrable bivariate function $f(x)$, we define its ridgelet coefficients by

$$\mathcal{R}_f(a,b,\theta) = \int \psi_{a,b,\theta}(x) f(x) dx.$$

We have the exact reconstruction formula

$$f(x) = \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{\infty} \mathcal{R}_f(a,b,\theta) \psi_{a,b,\theta}(x) \frac{da}{a^3} db \frac{d\theta}{4\pi} \tag{4}$$

valid a.e. for functions which are both integrable and square integrable. Furthermore, this formula is stable as one has a Parseval relation

$$\int |f(x)|^2 dx = \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{\infty} |\mathcal{R}_f(a,b,\theta)|^2 \frac{da}{a^3} db \frac{d\theta}{4\pi}. \tag{5}$$

Hence, much like the wavelet or Fourier transforms, the identity (4) expresses the fact that one can represent any arbitrary function as a continuous superposition of ridgelets. Discrete analogs of (4)-(5) exist, see [5], or [26] for a slightly different approach.

2.1 The Radon Transform

A basic tool for calculating ridgelet coefficients is to view ridgelet analysis as a form of wavelet analysis in the Radon domain. We recall that the Radon transform of an object f is the collection of line integrals indexed by $(\theta, t) \in [0, 2\pi) \times \mathbf{R}$ given by

$$Rf(\theta, t) = \int f(x_1, x_2) \delta(x_1 \cos\theta + x_2 \sin\theta - t)\, dx_1 dx_2, \tag{6}$$

where δ is the Dirac distribution. The ridgelet coefficients $\mathcal{R}_f(a,b,\theta)$ of an object f are given by analysis of the Radon transform via

$$\mathcal{R}_f(a,b,\theta) = \int Rf(\theta,t) a^{-1/2} \psi((t-b)/a)\, dt.$$

Hence the ridgelet transform is precisely the application of a 1-dimensional wavelet transform to the slices of the Radon transform where the angular variable θ is constant and t is varying.

2.2 Ridgelet Pyramids

Let Q denote a dyadic square $Q = [k_1/2^s, (k_1+1)/2^s) \times [k_2/2^s, (k_2+1)/2^s)$ and let $\mathcal{Q}$ be the collection of all such dyadic squares. We write $\mathcal{Q}_s$ for the collection of all dyadic squares of scale s. Associated to the squares $Q \in \mathcal{Q}_s$ we construct a partition of energy as follows. With w a nice smooth window obeying $\sum_{k_1,k_2} w^2(x_1 - k_1, x_2 - k_2) = 1$, we dilate and transport w to all squares Q at scale s, producing a collection of windows (w_Q) such that the w_Q^2's, $Q \in \mathcal{Q}_s$, make up a partition of unity. We also let T_Q denote the transport operator acting on functions g via

$$(T_Q g)(x_1, x_2) = 2^s g(2^s x_1 - k_1, 2^s x_2 - k_2).$$

With these notations, it is not hard to see that

$$fw_Q = \int \langle f, w_Q T_Q \psi_{a,b,\theta} \rangle T_Q \psi_{a,b,\theta} \frac{da}{a^3} db \frac{d\theta}{4\pi}$$

and, therefore, summing the above equality across squares at a given scale gives

$$f = \sum_{Q \in \mathcal{Q}_s} f w_Q^2 = \sum_Q \int \langle f, w_Q T_Q \psi_{a,b,\theta} \rangle w_Q T_Q \psi_{a,b,\theta} \frac{da}{a^3} db \frac{d\theta}{4\pi}. \tag{7}$$

The identity (7) expresses the fact that one can represent any function as a superposition of elements of the form $w_Q T_Q \psi_{a,b,\theta}$; that is, of ridgelet elements localized near the squares Q. For the function $T_Q \psi_{a,b,\theta}$ is the ridgelet ψ_{a_Q,θ_Q,b_Q} (3) with parameters obeying

$$a_Q = 2^{-s} a, \qquad \theta_Q = \theta, \qquad b_Q = b + k_1 2^{-s} \cos\theta + k_2 2^{-s} \sin\theta$$

and thus $w_Q T_Q \psi_{a,b,\theta}$ is a windowed ridgelet, supported near the square Q, hence the name *local ridgelet transform.*

The previous paragraph discussed the construction of local ridgelets of fixed length, roughly 2^{-s} (s fixed). Letting the scale s vary defines the multiscale ridgelet dictionary $\{\psi^Q_{a,b,\theta} : s \geq s_0, Q \in \mathcal{Q}_s, a > 0, b \in \mathbf{R}, \theta \in [0, 2\pi)\}$ by

$$\psi^Q_{a,b,\theta} = w_Q \, T_Q \psi_{a,b,\theta};$$

that is, a whole pyramid of local ridgelets at various lengths and locations. This is, of course, a massively overcomplete representation system and no formula like (7) is available for this multiscale ridgelet pyramid, because it is highly overcomplete.

3 Digital Ridgelet Transform

So a basic strategy for calculating the continuous ridgelet transform is first to compute the Radon transform $Rf(t, \theta)$ and second, to apply a one-dimensional wavelet transform to the slices $Rf(\cdot, \theta)$.

Several digital ridgelet transform have been proposed, and we will described three of them in this section, based on different implementations of the Radon transform.

3.1 The RectoPolar Ridgelet transform

In this section we develop a digital procedure which is inspired by this viewpoint, and is realizable on n by n numerical arrays.

A fundamental fact about the Radon transform is the projection-slice formula [16]:

$$\hat{f}(\lambda \cos\theta, \lambda \sin\theta) = \int Rf(t, \theta) e^{-i\lambda t} dt.$$

This says that the Radon transform can be obtained by applying the one-dimensional inverse Fourier transform to the two-dimensional Fourier transform restricted to radial lines going through the origin.

This of course suggests that approximate Radon transforms for digital data can be based on discrete fast Fourier transforms. This is a widely used approach, in the literature of medical

imaging and synthetic aperture radar imaging, for which the key approximation errors and artifacts have been widely discussed. In outline, one simply does the following, for gridded data $(f(i_1, i_2))$, $0 \leq i_1, i_2 < n-1$.

1. *2D-FFT.* Compute the two-dimensional FFT of f giving the array $(\hat{f}(k_1, k_2))$, $-n/2 \leq k_1, k_2 \leq n/2 - 1$.

2. *Cartesian to Polar Conversion.* Using an interpolation scheme, substitute the sampled values of the Fourier transform obtained on the square lattice with sampled values of $\hat{f}$ on a polar lattice: that is, on a lattice where the points fall on lines going through the origin.

3. *1D-IFFT.* Compute the one-dimensional IFFT on each line, i.e. for each value of the angular parameter.

The use of this strategy in connection with ridgelet transforms has been discussed in the articles [23, 22, 54].

3.1.1 A Polar Sampling Scheme for Digital Data

For our implementation of the Cartesian-to-polar conversion, we have used a pseudo-polar grid, in which the pseudo-radial variable has level sets which are squares rather than circles. Starting with Oppenheim and Mersereau [40] this grid has often been called the *concentric squares* grid in the signal processing literature; in the medical tomography literature it is associated with the *linogram* [28, 29], while in [3] it is called the rectopolar grid; see this last reference for a complete bibliographic treatment. The geometry of the rectopolar grid is illustrated on Figure 3. We select $2n$ radial lines in the frequency plane obtained by connecting the origin to the vertices (k_1, k_2) lying on the boundary of the array (k_1, k_2) , i.e. such that k_1 or $k_2 \in \{-n/2, n/2\}$. The polar grid $\xi_{\ell,m}$ (ℓ serves to index a given radial line while the position of the point on that line is indexed by m) that we shall use is the intersection between the set of radial lines and that of Cartesian lines parallel to the axes. To be more specific, the sample points along a radial line $\mathcal{L}$ whose angle with the vertical axis is less or equal to $\pi/4$ are obtained by intersecting $\mathcal{L}$ with the set of horizontal lines $\{x_2 = k_2,\ k_2 = -n/2, -n/2+1, \ldots, n/2\}$. Similarly, the intersection with the vertical lines $\{x_1 = k_1,\ k_1 = -n/2, -n/2+1, \ldots, n/2\}$ defines our sample points whenever the angle between $\mathcal{L}$ and the horizontal axis is less or equal to $\pi/4$. The cardinality of the rectopolar grid is equal to $2n^2$ as there are $2n$ radial lines and n sampled values on each of these lines. As a result, data structures associated with this grid will have a rectangular format. We observe that this choice corresponds to irregularly spaced values of the angular variable θ.

3.1.2 Interpolation to rectopolar Grid

To obtain samples on the rectopolar grid, we should, in general, interpolate from nearby samples at the Cartesian grid. In principle, compare [3, 20], the interpolation of Fourier transforms is a very delicate matter because of the well-known fact that the Fourier transform of an image is highly oscillatory, and the phase contains crucial information about the image. In our approach, however, we use a crude interpolation method: we simply impute for $\hat{f}(\xi_{\ell,m})$ the value of the Fourier transform taken at the point on the Cartesian grid nearest to $\xi_{\ell,m}$.

There are, of course, more sophisticated ways to realize the Cartesian-to-polar conversion; even simple bilinear interpolation would offer better theoretical accuracy. A very high

of the OFRT is nothing like a ridge function.

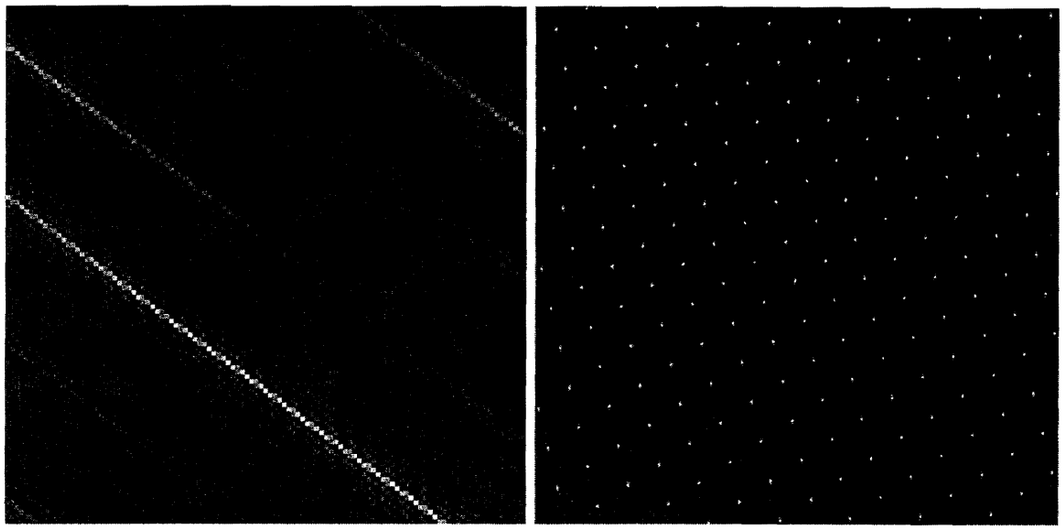


Figure 5: Left, backprojection of a ridgelet coefficient by the FFT-based ridgelet transform, and right, backprojection of a finite ridgelet coefficient.

Because of this specific definition of a line, the thresholding of the OFRT coefficients produces strong artefacts. Figure 6 left shows a part of the original standard Lena image, and Figure 6 right shows the reconstruction after the hard thresholding of the OFRT. A kind of noise has been added to the noise-free image! Finally, the OFRT presents another limitation: the image size must be a prime number. This last point is however not too restrictive, because we generally use a partitionning when denoising the data, and a prime number block size can be used. The OFTR is interesting from the conceptual point of view, but will certainly be of no help for real applications.

3.3 The Slant Stack Ridgelet Transform

The Fast Slant Stack [3] is geometrically more accurate than the previously decribed methods. The backprojection of a point in Radon space is exactly a ridge function in the spatial domain (see figure 7). The transformation of a $n \times n$ image is a $2n \times 2n$ image. n line integrals with angle between $[-\frac{\pi}{4}, \frac{\pi}{4}]$ are calculated from the zero padded image on the y-axis, and n line integrals with angle between $[\frac{\pi}{4}, \frac{3\pi}{4}]$ It is computed by zero padding the image on the x-axis. For a given angle inside $[-\frac{\pi}{4}, \frac{\pi}{4}]$, 2n line integrals are calculated by first shearing the zero-padded image, and then integrating the pixels values along all horizontal lines (resp. vertical lines for angles in $[\frac{\pi}{4}, \frac{3\pi}{4}]$). The shearing is performed column per column (resp. line per line) by using the 1D FFT. Figure 8 shows an example of the image shearing step with two different angles ($5\frac{\pi}{4}$ and $-\frac{\pi}{4}$). A ridgelet transform based on the Fast Slant Stack transform has been proposed in [20]. Connection between the Fast Slant Stack and the linogram has been investigated in [3], and a Fast Slant Stack is proposed, based on the 2D Fourier Transform.

Figure 6: Left, part of Lena image, and right, reconstruction after finite ridgelet coefficient thresholding.

4 Local Ridgelet Transforms

A digital version of the ideas presented in section 2.2 decomposes the original n by n image into smoothly overlapping blocks of sidelength b pixels in such a way that the overlap between two vertically adjacent blocks is a rectangular array of size b by $b/2$; we use overlap to avoid blocking artifacts. For an n by n image, we count $2n/b$ such blocks in each direction.

The partitioning introduces redundancy, as a pixel belongs to 4 neighboring blocks. We present two competing strategies to perform the analysis and synthesis:

1. The block values are weighted (analysis) in such a way that the co-addition of all blocks reproduce exactly the original pixel value (synthesis).

2. The block values are those of the image pixel values (analysis) but are weighted when the image is reconstructed (synthesis).

Of course, there are intermediate strategies and one could apply smooth windowing at both the analysis and synthesis stage as discussed in Section 2.2, for example. In the first approach, the data are smoothly windowed and this presents the advantage to limit the analysis artifacts traditionally associated with boundaries. The drawback, however, is a loss of sensitivity. Indeed, suppose for sake of simplicity that a vertical line with intensity level L intersects a given block of size b. Without loss of generality assume that the noise standard deviation is equal to 1. When the angular parameter of the Radon transform coincides with that of the line, we obtain a measurement with a signal intensity equal to bL while the noise standard deviation is equal to $\sqrt{b}$ (in this case, the value of the Signal to Noise Ratio (SNR) is $\sqrt{b}L$). If weights are applied at the analysis stage, the SNR is roughly equal to $L \sum_{i=1}^{b} w_i / \sqrt{\sum_{i=1}^{b} w_i^2} < \sqrt{b}L$. Experiments have shown that this sensitivity loss may have substantial effects in filtering applications and, therefore, the second approach seems more appropriate since our goal is image restoration.

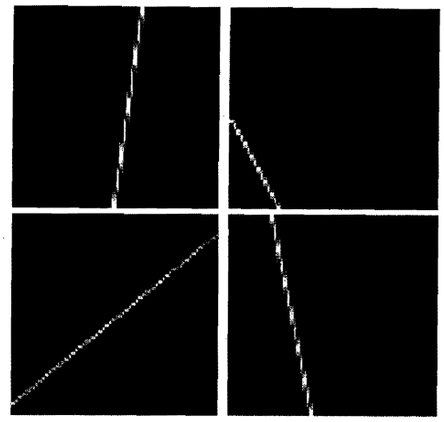


Figure 7: Backproject of a point at four different locations in the Radon space.

We calculate a pixel value, $f(i,j)$ from its four corresponding block values of half-size $\ell = b/2$, namely, $B_1(i_1,j_1)$, $B_2(i_2,j_1)$, $B_3(i_1,j_2)$ and $B_4(i_2,j_2)$ with $i_1, j_1 > b/2$ and $i_2 = i_1 - \ell, j_2 = j_1 - \ell$, in the following way:

$$\begin{aligned} f_1 &= w(i_2/\ell)B_1(i_1,j_1) + w(1-i_2/\ell)B_2(i_2,j_1) \\ f_2 &= w(i_2/\ell)B_3(i_1,j_2) + w(1-i_2/\ell)B_4(i_2,j_2) \\ f(i,j) &= w(j_2/\ell)f_1 + w(1-j_2/\ell)f_2 \end{aligned} \tag{9}$$

with $w(x) = \cos^2(\pi x/2)$. Of course, one might select any other smooth, nonincreasing function satisfying, $w(0) = 1$, $w(1) = 0$, $w'(0) = 0$ and obeying the symmetry property $w(x) + w(1-x) = 1$.

It is worth mentioning that the spatial partitioning introduces a redundancy factor equal to 4.

Finally, we note that in order to be in better agreement with the theory one should of course introduce a normalizing factor depending upon the block-size. However, since we are concerned about de-noising and the thresholding of individual coefficients, the normalization is a non-issue. Renormalizing coefficients automatically renormalizes corresponding thresholds in the exact same way, see section 6.

5 Digital Curvelet Transform

5.1 Discrete Curvelet Transform of Continuum Functions

We now briefly return to the continuum viewpoint of Section 2.2. Suppose we set an initial goal to produce a decomposition using the multiscale ridgelet pyramid. The hope is that this would allow us to use thin 'brushstrokes' to reconstruct the image, with all lengths and widths available to us. In particular, this would seem allow us to trace sharp edges precisely using a few elongated elements with very narrow widths.

As mentioned in Section 2.2, the full multiscale ridgelet pyramid is highly overcomplete. As a consequence, convenient algorithms like simple thresholding will not find sparse decompositions when such good decompositions exist. An important ingredient of the curvelet

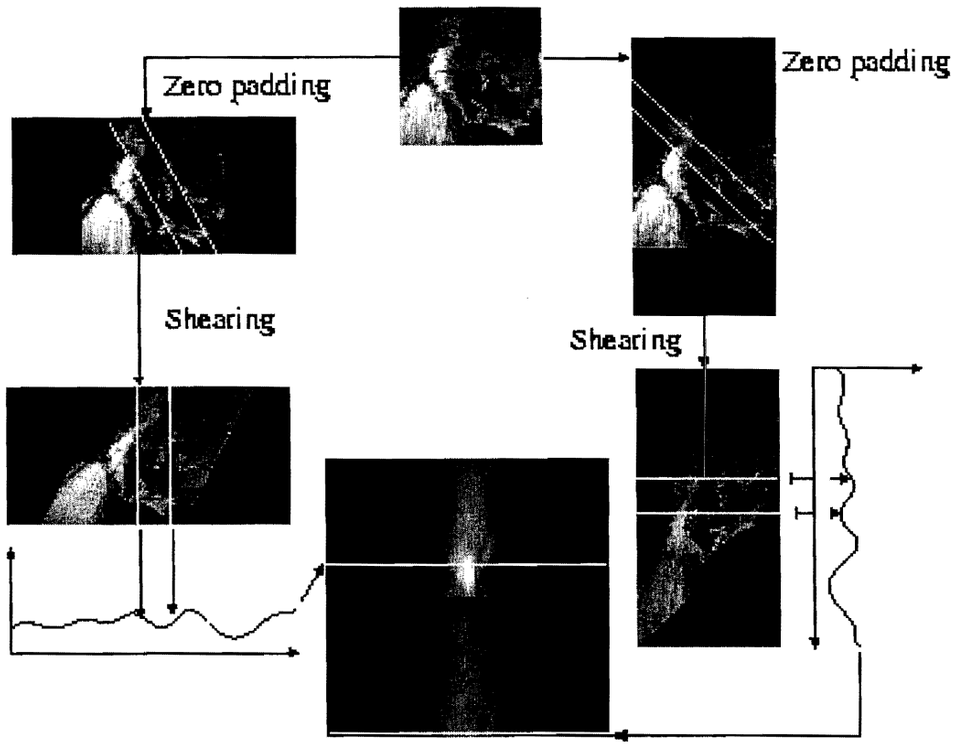


Figure 8: Slant Stack Transform of an image.

transform is to restore sparsity by reducing redundancy across scales. In detail, one introduces interscale orthogonality by means of subband filtering. Roughly speaking, different levels of the multiscale ridgelet pyramid are used to represent different subbands of a filter bank output. At the same time, this subband decomposition imposes a relationship between the width and length of the important frame elements so that they are anisotropic and obey $width = length^2$.

The discrete curvelet transform of a continuum function $f(x_1, x_2)$ makes use of a dyadic sequence of scales, and a bank of filters $(P_0 f, \Delta_1 f, \Delta_2 f, \ldots)$ with the property that the passband filter Δ_s is concentrated near the frequencies $[2^{2s}, 2^{2s+2}]$, e.g.

$$\Delta_s = \Psi_{2s} * f, \quad \widehat{\Psi_{2s}}(\xi) = \widehat{\Psi}(2^{-2s}\xi).$$

In wavelet theory, one uses a decomposition into dyadic subbands $[2^s, 2^{s+1}]$. In contrast, the subbands used in the discrete curvelet transform of continuum functions have the nonstandard form $[2^{2s}, 2^{2s+2}]$. This is nonstandard feature of the discrete curvelet transform well worth remembering.

With the notations of section 2.2, the curvelet decomposition is the sequence of the following steps:

- *Subband Decomposition.* The object f is decomposed into subbands:

$$f \mapsto (P_0 f, \Delta_1 f, \Delta_2 f, \ldots).$$

- *Smooth Partitioning.* Each subband is smoothly windowed into "squares" of an appropriate scale (of sidelength $\sim 2^{-s}$):

$$\Delta_s f \mapsto (w_Q \Delta_s f)_{Q \in \mathcal{Q}_s}.$$

- *Renormalization.* Each resulting square is renormalized to unit scale

$$g_Q = (T_Q)^{-1}(w_Q \Delta_s f), \qquad Q \in \mathcal{Q}_s. \tag{10}$$

- *Ridgelet Analysis.* Each square is analyzed via the discrete ridgelet transform.

In this definition, the two dyadic subbands $[2^{2s}, 2^{2s+1}]$ and $[2^{2s+1}, 2^{2s+2}]$ are merged before applying the ridgelet transform.

5.2 Digital Realization

In developing a transform for digital n by n data which is analogous to the discrete curvelet transform of a continuous function $f(x_1, x_2)$, we replace each of the continuum concepts with the appropriate digital concept mentioned in sections above. In general, the translation is rather obvious and direct. However, experience shows that one modification is essential; we found that, rather than merging the two the two dyadic subbands $[2^{2s}, 2^{2s+1}]$ and $[2^{2s+1}, 2^{2s+2}]$ as in the theoretical work, in the digital application, leaving these subbands separate, applying spatial partitioning to each subband and applying the ridgelet transform on each subband separately led to improved visual and numerical results.

We believe that the "à trous" subband filtering algorithm is especially well-adapted to the needs of the digital curvelet transform. The algorithm decomposes an n by n image I as a superposition of the form

$$I(x, y) = c_J(x, y) + \sum_{j=1}^{J} w_j(x, y),$$

where c_J is a coarse or smooth version of the original image I and w_j represents 'the details of I' at scale 2^{-j}, see [57] for more information. Thus, the algorithm outputs $J + 1$ subband arrays of size $n \times n$. (The indexing is such that, here, $j = 1$ corresponds to the finest scale (high frequencies).)

5.3 Algorithm

We now present a sketch of the discrete curvelet transform algorithm:

1. apply the à trous algorithm with J scales,
2. set $B_1 = B_{min}$,
3. for $j = 1, \ldots, J$ do,
   (a) partition the subband w_j with a block size B_j and apply the digital ridgelet transform to each block,

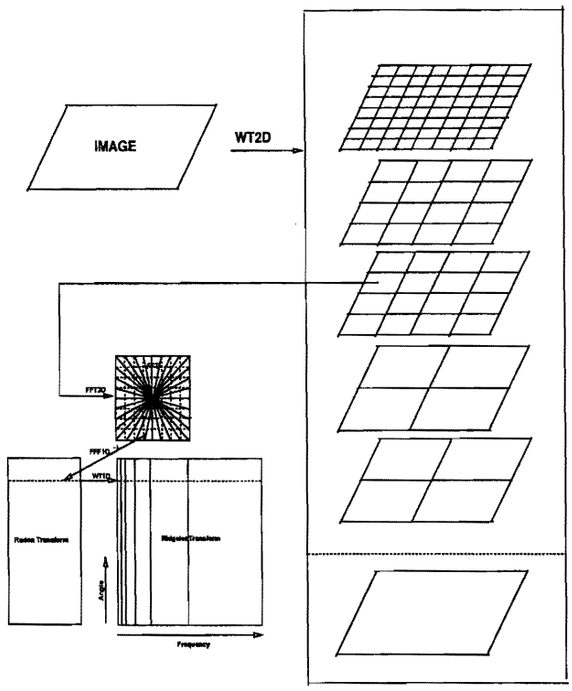


Figure 9: Curvelet transform flowgraph. The figure illustrates the decomposition of the original image into subbands followed by the spatial partitioning of each subband. The ridgelet transform is then applied to each block.

(b) if j modulo $2 = 1$ then $B_{j+1} = 2B_j$,

(c) else $B_{j+1} = B_j$.

The sidelength of the localizing windows is doubled *at every other* dyadic subband, hence maintaining the fundamental property of the curvelet transform which says that elements of length about $2^{-j/2}$ serve for the analysis and synthesis of the j-th subband $[2^j, 2^{j+1}]$. Note also that the coarse description of the image c_J is not processed. Finally, Figure 9 gives an overview of the organization of the algorithm.

This implementation of the curvelet transform is also redundant. The redundancy factor is equal to $16J + 1$ whenever J scales are employed. Finally, the method enjoys exact reconstruction and stability, because this invertibility holds for each element of the processing chain.

Figure 10 shows a few curvelets at different scales, orientations and locations.

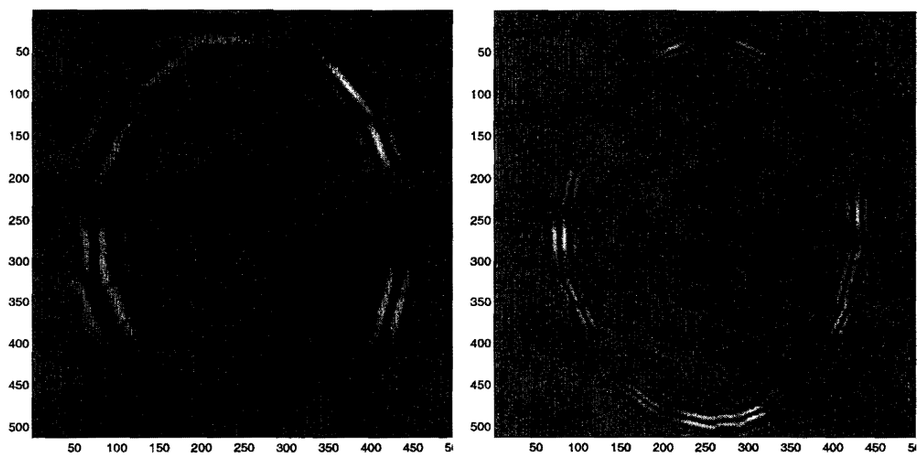


Figure 10: A few curvelets.

6 Filtering

We now apply our digital transforms for removing noise from image data. The methodology is standard and is outlined mainly for the sake of clarity and self-containedness.

Suppose that one is given noisy data of the form

$$x_{i,j} = f(i,j) + \sigma z_{i,j},$$

where f is the image to be recovered and z is white noise, i.e. $z_{i,j} \overset{i.i.d.}{\sim} N(0,1)$. Unlike FFT's or FWT's, our discrete ridgelet (resp. curvelet) transform is not norm-preserving and, therefore, the variance of the noisy ridgelet (resp. curvelet) coefficients will depend on the ridgelet (resp. curvelet) index λ. For instance, letting F denote the discrete curvelet transform matrix, we have $Fz \overset{i.i.d.}{\sim} N(0, FF^T)$. Because the computation of FF^T is prohibitively expensive, we calculated an approximate value $\tilde{\sigma}_\lambda^2$ of the individual variances using Monte-Carlo simulations where the diagonal elements of FF^T are simply estimated by evaluating the curvelet transforms of a few standard white noise images.

Let y_λ be the noisy curvelet coefficients ($y = Fx$). We use the following hard-thresholding rule for estimating the unknown curvelet coefficients:

$$\hat{y}_\lambda = y_\lambda \quad \text{if} \quad |y_\lambda|/\sigma \geq k\tilde{\sigma}_\lambda \tag{11}$$

$$\hat{y}_\lambda = 0 \quad \text{if} \quad |y_\lambda|/\sigma < k\tilde{\sigma}_\lambda. \tag{12}$$

In our experiments, we actually chose a scale-dependent value for k; we have $k = 4$ for the first scale ($j = 1$) while $k = 3$ for the others ($j > 1$).

6.1 Poisson Observations

Assume now that we have Poisson data $x_{i,j}$ with unknown mean $f(i,j)$. The Anscombe transformation [1]

$$\tilde{x} = 2\sqrt{x + \frac{3}{8}} \tag{13}$$

stabilizes the variance and we have $\tilde{x} = 2\sqrt{f} + \epsilon$ where ϵ is a vector with independent and approximately standard normal components. In practice, this is a good approximation whenever the number of counts is large enough, greater than 30 per pixel, say.

For small number of counts, a possibility is to compute the Radon transform of the image, and then to apply the Anscombe transformation to the Radon data. The rationale being that, roughly speaking, the Radon transform corresponds to a summation of pixel values over lines and that the sum of independent Poisson random variables is a Poisson random variable with intensity equal to the sum of the individual intensities. Hence, the intensity of the sum may be quite large (hence validating the Gaussian approximation) even though the individual intensities may be small. This might be viewed as an interesting feature as unlike wavelet transforms, the ridgelet and curvelet transforms tend to average data over elongated and rather large neighborhoods.

6.2 Filtering Experiments

6.2.1 Lenna Image

In our first example, a Gaussian noise with a standard deviation equal to 20 was added to the classical **Lenna** image (512 by 512). Several methods were used to filter the noisy image:

1. Thresholding of Monoscale ridgelet transforms with scale (= block size) (8, 16, 32 and 64),
2. Thresholding of Curvelet transform, and
3. Wavelet de-noising methods in the following four families:
   (a) Bi-orthogonal wavelet transform using the Dauchechies-Antonini 7/9 filters (FWT-7/9) and hard thresholding.
   (b) Undecimated bi-orthogonal wavelet transform (UWT-7/9) with hard thresholding; we used $k = 4$ for the finest scale, and 3 for the others.
   (c) Multiscale entropy processing using the undecimated wavelet transform. This method is discussed in [59, 56].
   (d) Wavelet-domain Hidden Markov Models (WHMM) using Daubechies orthonormal wavelets of length 8. This method [13] attempts to model the joint probability density of the wavelet coefficients and derives the filtered coefficients using an empirical Bayesian approach. We used this rather than a competing method of Simoncelli [46] owing to availability of a convenient software implementation.

We use the PSNR as an 'objective' measure of performance. In addition, we used our own visual capabilities to identify artifacts whose effects may not be well-quantified by the PSNR value. The sort of artifacts we are particularly concerned about may be seen on display in the upper right panel of Figure 11, which displays a wavelet reconstruction. This image has a number of problems near edges. In reconstructing some edges which should follow smooth curves one gets edges which are poorly defined and very choppy in reconstruction (for example in the crown of the hat); also some edges which are accurately reconstructed exhibit oscillatory structure along the edge which is not present in the underlying image (for example in the shoulder and the hat brim). We refer to all such effects as artifacts.

Method	PSNR	Comments
Noisy image	22.13	
FWT7-9 + Universal Hard thresh.	28.35	many artifacts
UWT7-9 + ksigma Hard thresh.	31.94	very few artifact
UWT7-9 + Multiscale entropy	32.10	very few artifact
WHMM	30.80	some noise remains
Local ridgelets ($B = 8$)	29.99	artifacts
Local ridgelets ($B = 16$)	30.87	few artifacts
Local ridgelets ($B = 32$)	30.97	few artifacts
Local ridgelets ($B = 64$)	30.79	few artifacts
Curvelets ($B = 16$)	31.95	very few artifact

Table 1: Table of PSNR values after filtering the noisy image (Lenna + Gaussian white noise (sigma = 20)).

Our experiments are reported on Figures 11 and 12. The latter figure represents a detail of the original image and helps the reader observe the qualitative differences between the different methods. We observe that:

- The curvelet transform enjoys superior performance over local ridgelet transforms, regardless of the the block size, and
- The undecimated wavelet transform approach outputs a PSNR comparable to that obtained via the curvelet transform (the PSNR is slightly better for the multiscale entropy method).
- The curvelet reconstruction does not contain the quantity of disturbing artifacts along edges that one sees in wavelet reconstructions. An examination of the details of the restored images (Figure 12) is instructive. One notices that the decimated wavelet transform exhibits distortions of the boundaries and suffers substantial loss of important detail. The undecimated wavelet transform gives better boundaries, but completely omits to reconstruct certain ridges in the hatband. In addition, it exhibits numerous small-scale embedded blemishes; setting higher thresholds to avoid these blemishes would cause even more of the intrinsic structure to be missed.
- The curvelet reconstructions display higher sensitivity than the wavelet-based reconstructions. In fact both wavelet reconstructions obscure structure in the hatband which was visually detectable in the noisy panel at upper left. In comparison, every structure in the image which is visually detectable in the noisy image is clearly displayed in the curvelet reconstruction.

These observations are not limited to the particular experiment displayed here. We have observed similar characteristics in many other experiments; see Figure 17 for another example.

To study the dependency of the curvelet denoising procedure on the noise level, we generated a set of noisy images (the noise standard deviation varies from 5 to 100) from both `Lenna` and `Barbara`. We then compared the three different filtering procedures based respectively on the curvelet transform and on the undecimated/decimated wavelet transforms. This series of experiments is summarized in Figure 13 which displays the PSNR versus the noise standard deviation. These experimental results show that the curvelet transform outperforms wavelets for removing noise from those images, as the curvelet PSNR is systematically higher than the wavelet PSNR's – and this, across a broad range of noise levels. Other experiments with color images led to similar results.

6.2.2 Recovery of Linear Features

The next experiment (Figure 14) consists of an artificial image containing a few bars, lines and a square. The intensity is constant along each individual bar; from left to right, the intensities of the ten vertical bars (these are in fact thin rectangles which are 4 pixels wide and 170 pixels long) are equal to $\frac{32}{2^i}$, $i = 0, \ldots 9$. The intensity along all the other lines is equal to 1, and the noise standard deviation is 1/2. Displayed images have been log-transformed in order to better see the results at low signal to noise ratio.

The curvelet reconstruction of the nonvertical lines is obviously sharper than that obtained using wavelets. The curvelet transform also seems to go one step further as far as the reconstruction of the vertical lines is concerned. Roughly speaking, for those templates, the wavelet transforms stops detecting signal at a SNR equal to 1 (we defined here the SNR as the intensity level of the pixels on the line, divided by the noise standard deviation of the noise) while the cut-off value equals 0.5 for the curvelet approach. It is important to note that the horizontal and vertical lines correspond to privileged directions for the wavelet transform,

because the underlying basis functions are direct products of functions varying solely in the horizontal and vertical directions. Wavelet methods will given even poorer results on lines of the same intensity but tilting substantially away from the Cartesian axes. Compare the reconstructions of the faint diagonal lines in the image.

6.2.3 Recovery of Curves

In this experiment (Figure 15), we have added a Gaussian noise to "War and Peace," a drawing from Picasso which contains many curved features. Fig.15 bottom left and right shows respectively the restored images by the undecimated wavelet transform and the curvelet transform. Curves are more sharply recovered with the curvelet transform.

The authors are working on new methods (some of which will be based on the curvelet transform) to extract and recover curves from noisy data with greater accuracy and, therefore, this example is merely to be taken for illustrative purposes.

6.2.4 Denoising of a Color Image

In a wavelet based denoising scenario, color RGB images are generally mapped into the YUV space, and each YUV band is then filtered independently from the others. The goal here to see whether the curvelet transform would give improved results. We used four of the classical color images, namely `Lenna`, `Peppers`, `Baboon`, and `Barbara` (all images except perhaps `Barbara` are available from the USC-SIPI Image Database [14]. We performed the series of experiments described in section 6.2.1 and summarized our findings on Figure 16 which again displays the PSNR versus the noise standard deviation for the four images.

In all cases, the curvelet transform outperforms the wavelet transforms in terms of PSNR – at least for moderate and large values of the noise level. In addition, the curvelet transform outputs images that are visually more pleasant. Figure 17 illustrates this last point.

6.2.5 Saturn Rings

A Gaussian white noise with a standard deviation fixed to 20 was added to the `Saturn` image. We employed several methods to filter the noisy image:

1. Thresholding of the Curvelet transform.
2. Bi-orthogonal undecimated wavelet de-noising methods using the Dauchechies-Antonini 7/9 filters (FWT-7/9) and hard thresholding.
3. à trous wavelet transform algorithm and hard thresholding.

Our experiments are reported on Figure 18. The curvelet reconstruction does not contain the quantity of disturbing artifacts along edges that one sees in wavelet reconstructions. An examination of the details of the restored images is instructive. One notices that the decimated wavelet transform exhibits distortions of the boundaries and suffers substantial loss of important detail. The à trous wavelet transform gives better boundaries, but completely omits to reconstruct certain ridges. In addition, it exhibits numerous small-scale embedded blemishes; setting higher thresholds to avoid these blemishes would cause even more of the intrinsic structure to be missed.

Figure 11: Noisy image (top left) and filtered images using the decimated wavelet transform (top right), the undecimated wavelet transform (bottom left) and the curvelet transform (bottom right).

6.2.6 Supernova with Poisson noise

Figure 19 shows an example of an X-ray image filtering by the ridgelet transform using such an approach. Figure 19 left and right shows respectively the XMM/Newton image of the Kepler SN1604 supernova and the ridgelet filtered image (using a five sigma hard thresholding).

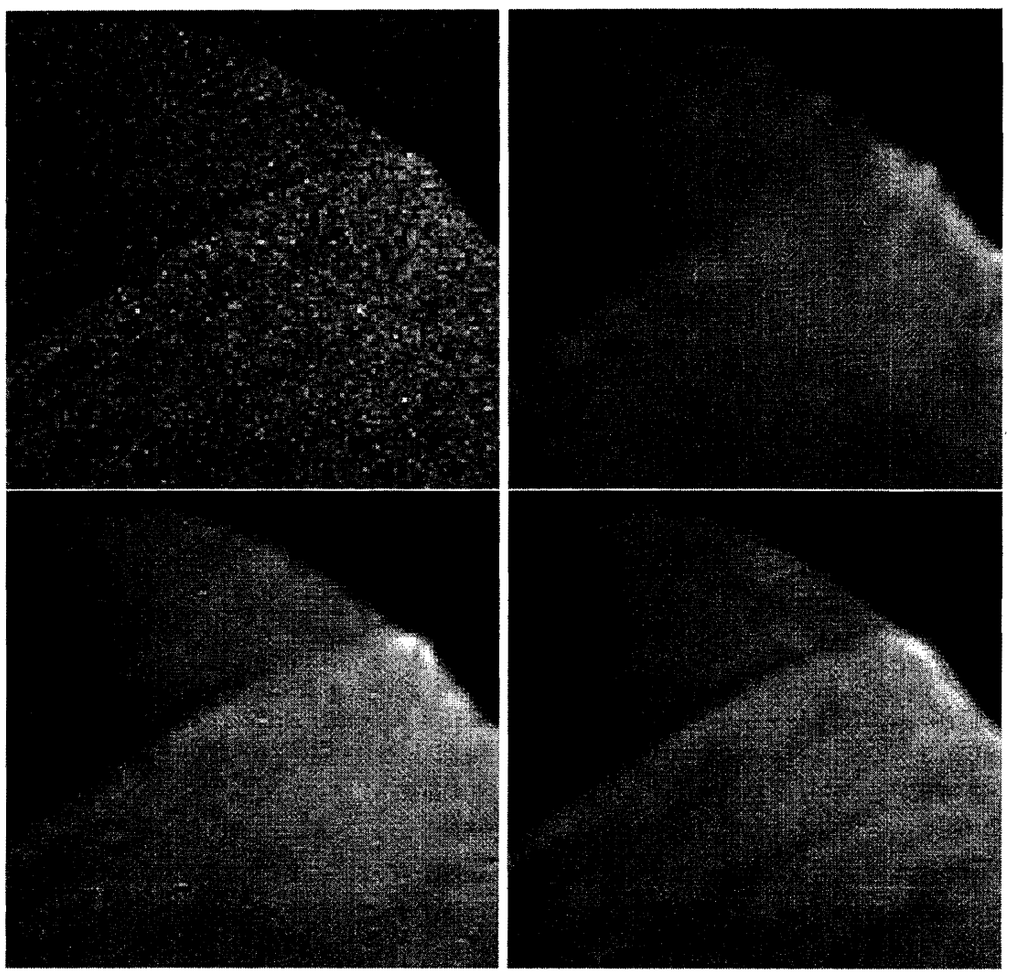


Figure 12: The figure displays the noisy image (top left), and the restored images after denoising by means of the DWT (top right), UWT (bottom left), and the curvelet transform (bottom right). The diagonal lines of the hat have been recovered with much greater fidelity in the curvelet approach.

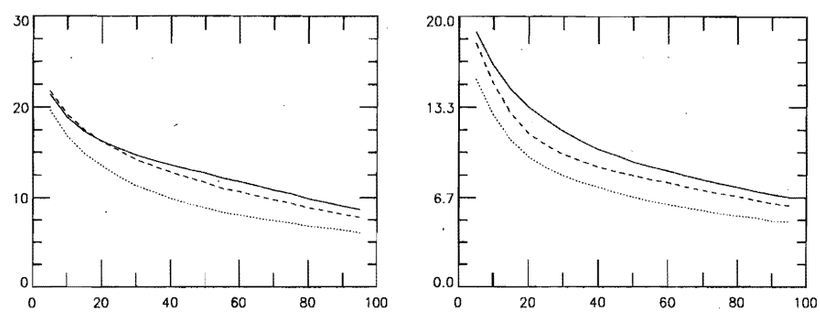


Figure 13: PSNR versus noise standard deviation for different denoising methods. The three methods based on the curvelet, undecimated and decimated wavelet transforms are represented with a continuous, dashed, and dotted line respectively. The left panel corresponds to **Lenna**, and the right to **Barbara**.

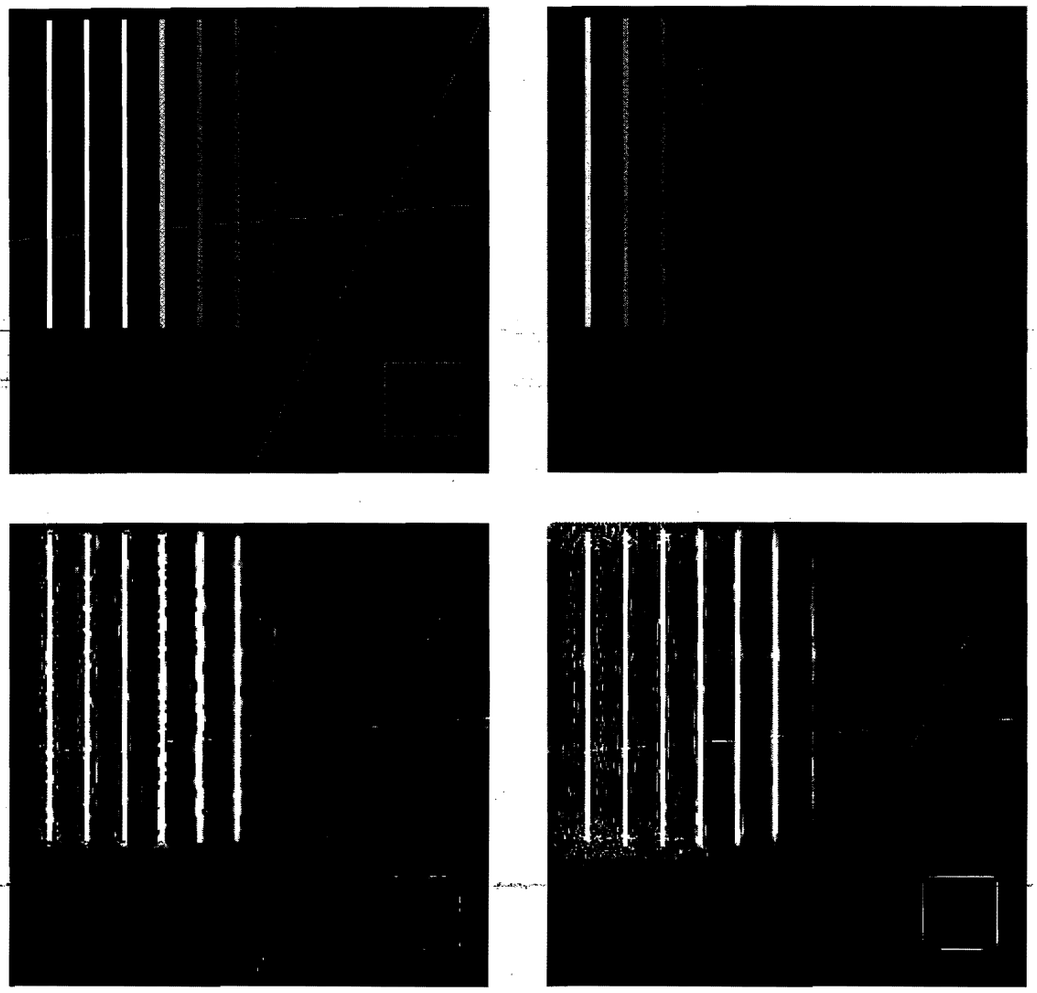


Figure 14: The top panels display a geometric image and that same image contaminated with a Gaussian white noise. The bottom left and right panels display the restored images using the undecimated wavelet transform and the curvelet transform respectively.

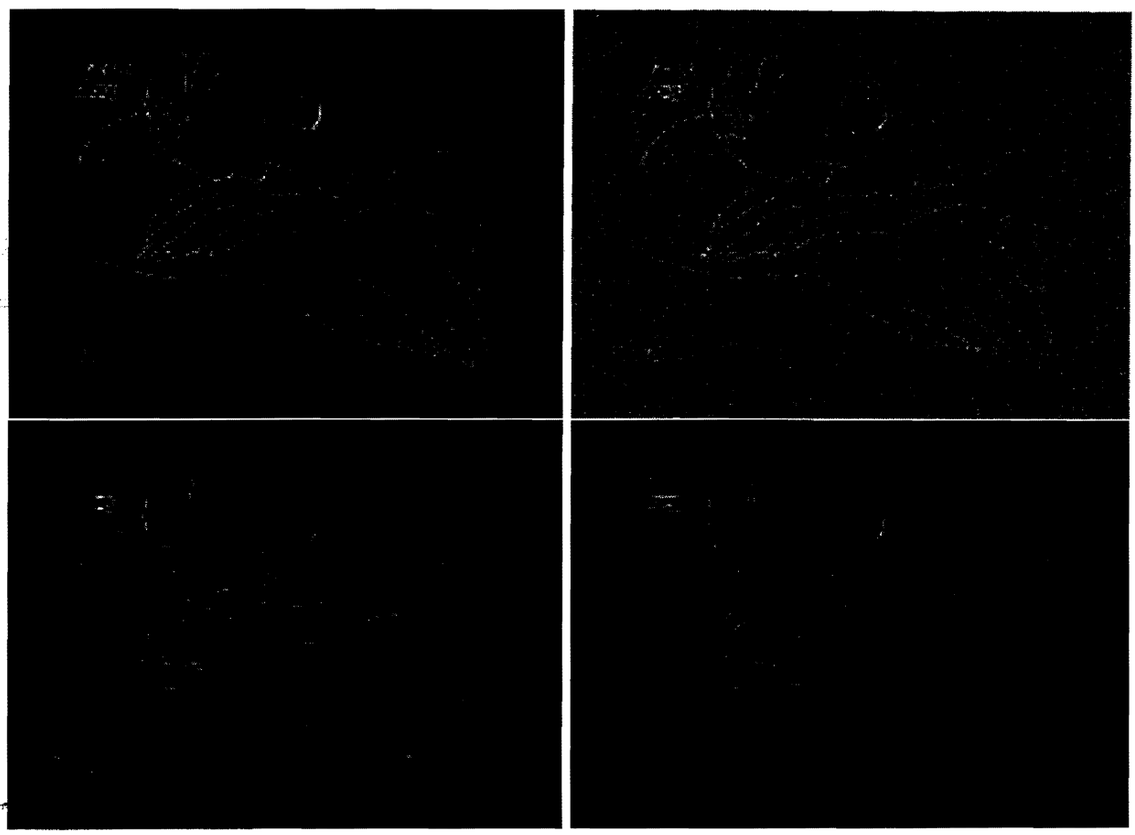


Figure 15: The top panels display a Picasso picture (War and Peace) and that same image contaminated with a Gaussian white noise. The bottom left and right panels display the restored images using the undecimated wavelet transform and the curvelet transform respectively.

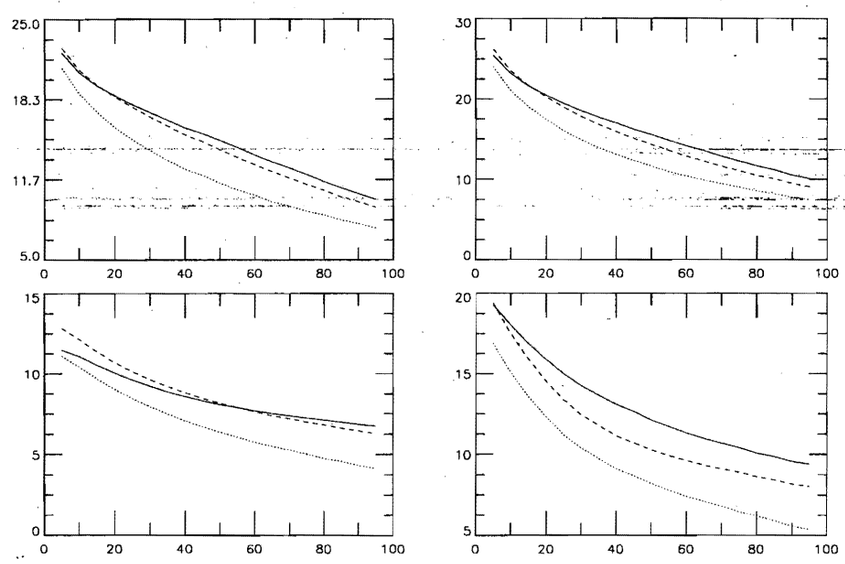


Figure 16: PSNR versus noise standard deviation using different filtering methods. YUV and curvelet, YUV and undecimated wavelet, and YUV and decimated wavelet transforms are represented respectively with a continuous, dashed, and dotted line. The upper left panel corresponds to Lenna (RGB), the upper right to pepper (RGB), the bottom left to Baboon (RGB), and the bottom right to Barbara (RGB).

Figure 17: Upper left: noisy **Barbara** image. Upper right: restored image after applying the curvelet transform. Details of the restored images are shown on the bottom left panel (undecimated wavelet transform) and right (curvelet transform) panel.

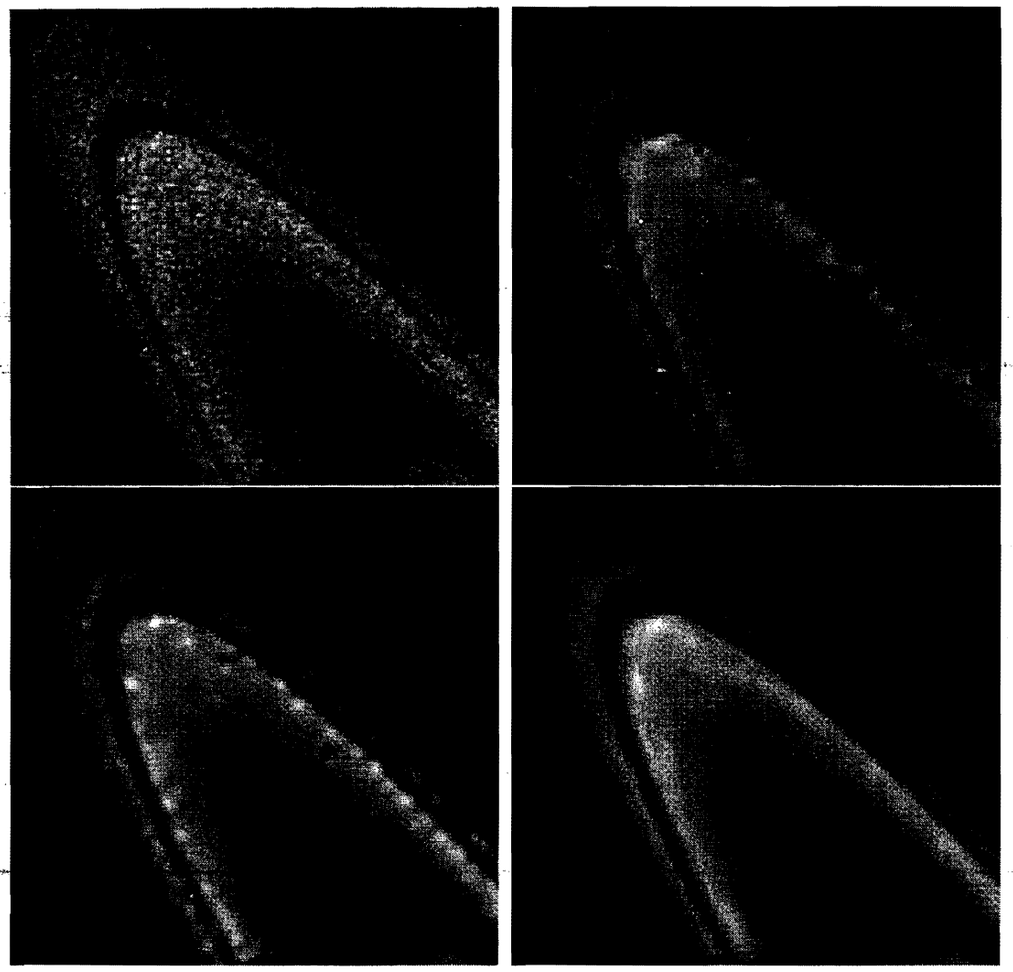


Figure 18: Top left, part of Saturn image with a Gaussian noise. Top right, filtered image using the undecimated bi-orthogonal wavelet transform. Bottom left and right, filtered image by the à trous wavelet transform algorithm and the curvelet transform.

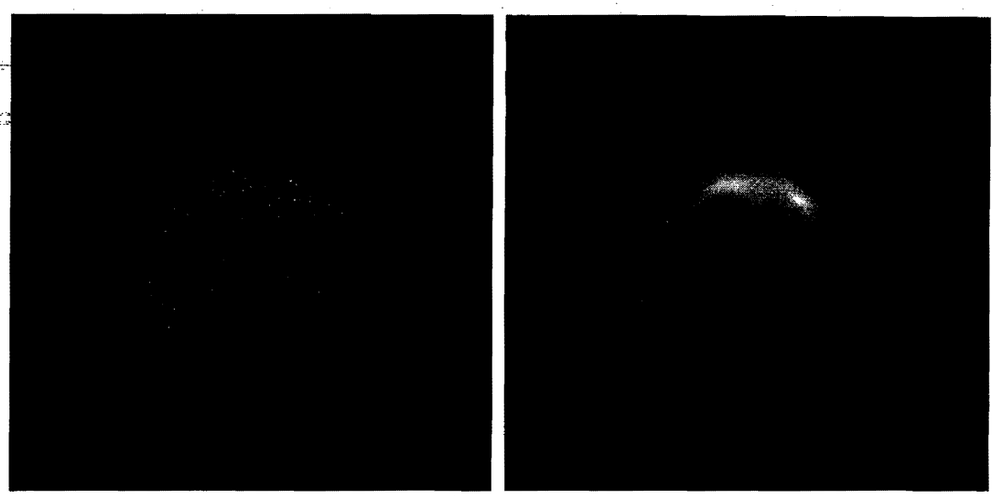


Figure 19: Left, XMM/Newton image of the Kepler SN1604 supernova. Right, ridgelet filtered image.

7 The Combined Filtering Method

7.1 Introduction

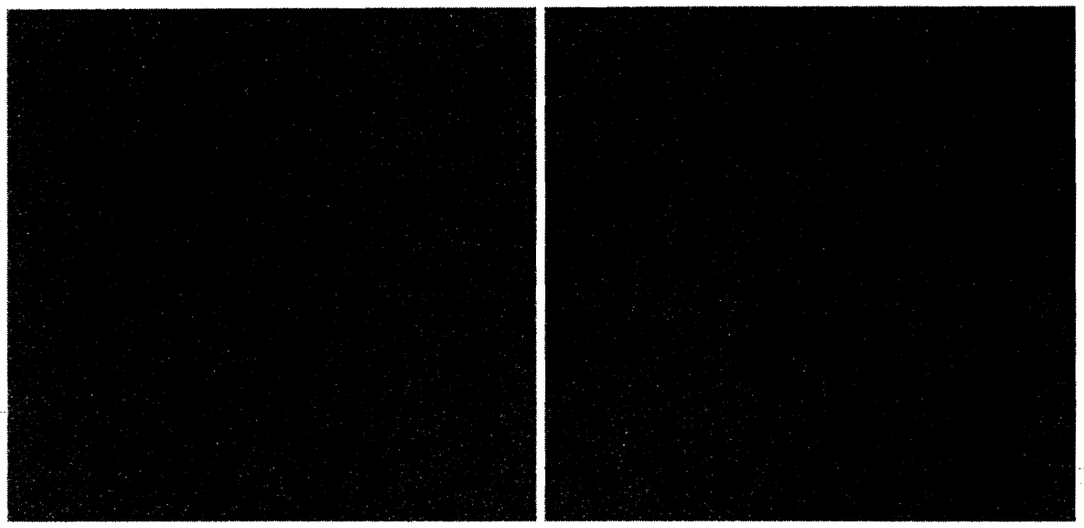


Figure 20: Residual for thresholding of the undecimated wavelet transform and thresholding of the curvelet transform.

Although the results obtained by simply thresholding curvelet expansion are encouraging, there is of course ample room for further improvement. A quick inspection of the residual images for both the wavelet and curvelet transforms shown on figure 20 reveals the existence of very different features. For instance, wavelets do not restore long edges with high fidelity while curvelets are challenged with small features such as Lenna's eyes. Loosely speaking, each transform has its own area of expertise and this complementarity may be of great potential. This section will develop a denoising strategy based on the idea of combining both transforms.

In general, suppose that we are given K linear transforms $T_1, \ldots, T_K$ and let α_k be the coefficient sequence of an object x after applying the transform T_k, i.e. $\alpha_k = T_k x$. We will suppose that for each transform T_k we have available a reconstruction rule that we will denote by T_k^{-1} although this is clearly an abuse of notations. Finally, T will denote the block diagonal matrix with the T_k's as building blocks and α the amalgamation of the α_k's.

A hard thresholding rule associated with the transform T_k synthesizes an estimate $\tilde{s}_k$ via the formula

$$\tilde{s}_k = T_k^{-1}\delta(\alpha_k) \tag{14}$$

where δ is a rule that sets to zero all the coordinates of α_k whose absolute value falls below given a sequence of thresholds (such coordinates are said to be nonsignificant).

In practice, a widely used approach is to compute the average of the $\tilde{s}_k$'s giving a reconstruction of the form

$$\tilde{s} = \sum_k \tilde{s}_k / K. \tag{15}$$

For instance, in the literature of image processing it is common to average reconstructions obtained after thresholding the wavelet coefficients of translated versions of the original dataset

(cycle-spinning), i.e. the T_k's are obtained by composing translations and the wavelet transform. In our setup, we do not find this solution very appealing as this creates the opportunity to average high-quality and low-quality reconstructions.

7.2 The Combined Filtering Principle

Given data y of the form $y = s + \sigma z$, where s is the image we wish to recover and z is standard white noise, we propose solving the following optimization problem [55]:

$$\min \|T\tilde{s}\|_{\ell_1}, \quad \text{subject to} \quad s \in C, \tag{16}$$

where C is the set of vectors $\tilde{s}$ which obey the linear constraints

$$\begin{cases} \tilde{s} \geq 0, \\ |T\tilde{s} - Ty| \leq e; \end{cases} \tag{17}$$

here, the second inequality constraint only concerns the set of significant coefficients, i.e. those indices μ such that $\alpha_\mu = (Ty)_\mu$ exceeds (in absolute value) a threshold t_μ. Given a vector of tolerance (e_μ), we seek a solution whose coefficients $(T\tilde{s})_\mu$ are within e_μ of the noisy empirical α_μ's. Think of α_μ as being given by

$$y = \langle y, \varphi_\mu \rangle,$$

so that α_μ is normally distributed with mean $\langle f, \varphi_\mu \rangle$ and variance $\sigma_\mu^2 = \sigma^2 \|\varphi_\mu\|_2^2$. In practice, the threshold values range typically between three and four times the noise level σ_μ and in our experiments we will put $e_\mu = \sigma_\mu/2$. In short, our constraints guarantees that the reconstruction will take into account any pattern which is detected as significant by a any of the K transforms.

We use an ℓ_1 penalty on the coefficient sequence because we are interested in *low complexity* reconstructions. There are other possible choices of complexity penalties; for instance, an alternative to (16) would be

$$\min \|\tilde{s}\|_{TV}, \quad \text{subject to} \quad s \in C.$$

where $\|\cdot\|_{TV}$ is the Total Variation norm, i.e. the discrete equivalent of the integral of the euclidian norm of the gradient.

7.3 The Minimization Method

We propose solving (16) using the method of hybrid steepest descent (HSD) [62]. HSD consists in building the sequence

$$s^{n+1} = P(s^n) - \lambda_{n+1} \nabla_J (P(s^n)); \tag{18}$$

here, P is the ℓ_2 projection operator onto the feasible set C, ∇_J is the gradient of equation 16, and $(\lambda_n)_{n\geq 1}$ is a sequence obeying $(\lambda_n)_{n\geq 1} \in [0,1]$ and $\lim_{n\to+\infty} \lambda_n = 0$.

Unfortunately, the projection operator P is not easily determined and in practice we will use the following proxy; compute $T\tilde{s}$ and replace those coefficients which do not obey the constraints $|T\tilde{s} - Ty| \leq e$ (those which fall outside of the prescribed interval) by those of y; apply the inverse transform.

The combined filtering algorithm (CFA) is:

1. Initialize $L_{\max} = 1$, the number of iterations N_i, and $\delta_\lambda = \frac{L_{\max}}{N_i}$.
2. Estimate the noise standard deviation σ, and set $e_k = \frac{\sigma}{2}$.
3. for k = 1, .., K calculate the transform: $\alpha_k^{(s)} = T_k s$.
4. Set $\lambda = L_{\max}$, $n = 0$, and $\tilde{s}^n$ to 0.
5. While $\lambda >= 0$ do
   - $u = \tilde{s}^n$.
   - for k = 1, .., K do
     - Calculate the transform $\alpha_k = T_k u$.
     - For all coefficients $\alpha_{k,l}$ do
       * Calculate the residual $r_{k,l} = \alpha_{k,l}^{(s)} - \alpha_{k,l}$
       * if $\alpha_{k,l}^{(s)}$ is significant and $\mid r_{k,l} \mid > e_{k,l}$ then $\alpha_{k,l} = \alpha_{k,l}^{(s)}$
       * $\alpha_{k,l} = sgn(\alpha_{k,l})(\mid \alpha_{k,l} \mid -\lambda)_+$.
     - $u = T_k^{-1}\alpha_k$
   - Threshold negative values in u and $\tilde{s}^{n+1} = u$.
   - $n = n + 1$, $\lambda = \lambda - \delta_\lambda$, and goto 5.

7.4 Experiments

Method	PSNR	Comments
Noisy image	22.13	
OWT7-9 + ksigma Hard thresh.	28.35	many artifact
UWT7-9 + ksigma Hard thresh.	31.94	very few artifact
Curvelet (B=16)	31.95	no artifact
Combined filtering	32.72	no artifact

Table 2: PSNR after filtering the simulated image (Lena + Gaussian noise (sigma=20)). In the combined filtering, a curvelet and an undecimated wavelet transform have been used.

The noisy **Lenna** image (the noise standard deviation being equal 20) has been filtered by the undecimated wavelet transform, the curvelet transform, and by our combined transform approach (curvelet and undecimated wavelet transforms). The results are reported in Table 2. Figure 21 displays the noisy image (top left), and the restored image after denoising by the combined transforms (bottom right). Details are displayed on Figure 21 bottom left. Figure 21 bottom right shows the full residual image, and can be compared to the residual images shown in Figure 20. The residual is much better when the combined filtering is applied, and no feature can anymore be detected by eye. This was not the case for both the wavelet and the curvelet filtering.

Figure 22 displays the PSNR of the solution versus the number of iterations. On this example, the algorithm is shown to converge rapidly. From a practical viewpoint only four or five iterations are truly needed. Note that the result obtained after a single iteration is

Method	PSNR (coeff. l_1 norm minim.)	PSNR (TV minim.)
Undecimated Wavelet only	32.00	32.43
Curvelet only	32.03	32.40
Wavelet + Curvelet (Combined Filtering)	32.72	32.77

Table 3: PSNR after filtering the simulated image (Lena + Gaussian noise (sigma=20)). In the combined filtering, a curvelet and an undecimated wavelet transform have been used.

already superior than those available using methods based on the thresholding of wavelet or curvelet coefficients alone.

Several papers have been recently published, based on the concept of minimizing the total variation under constraints in the wavelet domain [27, 17, 36] or in the curvelet domain [9]. Our combined approach can be seen as a generalization of these methods. We made a set of experiments in order to estimate (i) if the total variation is better than the l_1 norm of the multiscale coefficients, and (ii) if the combined approach improves the results compared to a single transform based method.

In our example, a Gaussian noise with a standard deviation equal to 20 was added to the classical Lenna image (512 by 512). Several methods were used to filter the noisy image:

1. TV + constraint in the Wavelet Domain.
2. TV + constraint in the Curvelet Domain.
3. Wavelet l_1 norm minimization + wavelet constraints.
4. Curvelet l_1 norm minimization + curvelet constraints.
5. Combined filtering method using multiscale coefficients l_1 norm minimization.
6. Combined filtering method using TV minimization.

We use the PSNR as an 'objective' measure of performance. The noisy image PSNR is 22.13. PSNR results from the different tested methods are reported in Table 3.

We observe that the combined filtering leads to a significant improvement when compared to a single transform based method. The TV penalization gives better results when a single transform is used, while it seems not to have too much importance for the combined filtering approach. We will see in the following that the latter is not true for the deconvolution problem.

7.5 Discussion

We believe that the denoising experiments presented in this paper are of very high quality;

1. The combined filtering leads to a real improvement both in terms of PSNR and visual appearance.

2. The combined approach arguably challenges the eye to distinguish structure/features from residual images on real image data (at least for the range of noise levels that was considered here). Single transforms cannot manage such a feat.

We also note that the combined reconstruction may tend to be free of major artifacts which is very much unlike typical thresholding rules. Although the ease of implementation is clear we did not address the computational issues associated with our method. In a nutshell, the algorithms we described requires calculating each transform and its inverse only a limited number of times.

In our examples, we constructed a combined transform from linear transforms (wavelets, ridgelets and curvelets) but our paradigm extends to any kind of nonlinear transform such as the Pyramidal Median Transform [57] or morphological multiscale transforms [30].

Figure 21: Noisy image (top left), and filtered image the combined transforms (top right). Bottom left panel shows a detail of the filtered image. The full residual image is displayed on the bottom right.

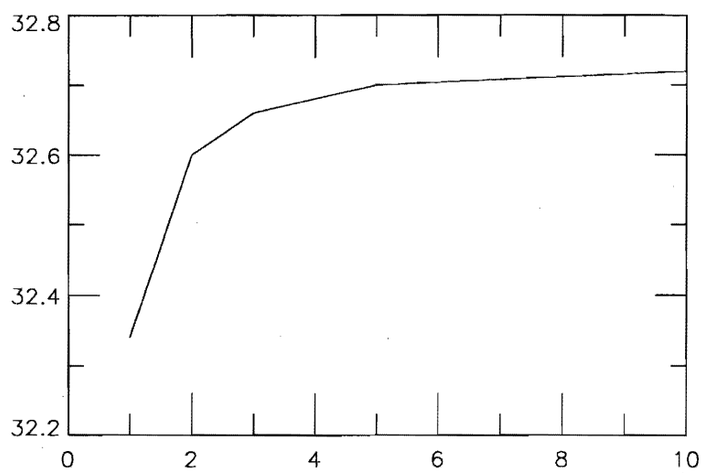


Figure 22: PSNR versus the number of iterations.

8 Deconvolution

8.1 Wavelet and Deconvolution

Consider an image characterized by its intensity distribution 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: $I(x,y) = (P * O)(x,y) + N(x,y)$, where P is the point spread function (PSF) of the imaging system, and N is an additive noise. 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 [4]).

The wavelet based non-iterative algorithm, the Wavelet-Vaguelette decomposition [25], consists in first applying an inverse filtering ($F = P^{-1} * I + P^{-1} * N = O + Z$ where $\hat{P}^{-1}(\nu) = \frac{1}{\hat{P}(\nu)}$). The noise $Z = P^{-1} * N$ is not white but remains Gaussian. It is amplified when the deconvolution problem is unstable. Then, a wavelet transform is applied on F, the wavelet coefficients are soft or hard thresholded [24], and the inverse wavelet transform furnishes the solution. The method has been refined by adapting the wavelet basis to the frequency response of the inverse of P [34]. It leads a special basis, this *Mirror Wavelet Basis*, which has a time-frequency tiling structure different from conventional wavelets one, and isolates the frequency ν_s where $\hat{P}$ is close to zero, because a singularity in $\hat{P}^{-1}(\nu_s)$ influences the noise variance in the wavelet scale corresponding the frequency band which includes ν_s. Because it may not be possible to isolate all singularities, Neelamani [42] has advocated an hybrid approach, and propose to still use the Fourier domain to restrict excessive noise amplification. These approaches are fast and competitive compared to linear methods, and the wavelet thresholding removes the Gibbs oscillations. It presents however several drawbacks: (i) the first step (division in the Fourier space by the Point Spread Function) cannot always be done properly, (ii) the positivity a priori is not used, (iii) it is not trivial to consider non Gaussian noise. As an alternative, several wavelet based iterative algorithms has been proposed [57], especially in the astronomical domain where the positivity a priori is known to improve significantly the result. The simplest method consists in first estimating the multiresolution support M (i,e. $M(j,x,y) = 1$ if the wavelet transform of the data presents a significant coefficient at band j and at pixel position (x,y), and 0 otherwise), and to apply the following iterative scheme:

$$O^{n+1} = O^n + P^* * \mathcal{W}^{-1}[M.\mathcal{W}(I - P * O^n)] \tag{19}$$

where $\mathcal{W}$ is the wavelet transform operator. At each iteration, information is extracted from the residual only at scales and positions defined by the multiresolution support. M is estimated from the input data and the correct noise modeling can easily be considered.

9 The Combined Deconvolution Method

As for the filtering, we expect that the combination of different transforms improve the quality of the result. The combined approach for the deconvolution leads to two different methods. Indeed, if the noise is Gaussian and if the division by the PSF in the Fourier space can be done properly, then the deconvolution problem becomes a filtering problem where the noise is still Gaussian, but not white. The Combined Filtering Algorithm can then be applied using the curvelet transform and the wavelet transform, but by estimating first the correct thresholds

in the different bands of both transforms. As the mirror wavelet basis is known to produce better results than the wavelet basis, it is recommended to use it instead of the standard undecimated wavelet transform.

An iterative deconvolution method is more general and can always be applied. Furthermore, the correct noise modeling can much more easily be taken into account. The method consists in detecting first all the significant coefficients with all used multiscale transforms. If we use K transforms $T_1, \ldots, T_K$, we derive K multiresolution supports $M_1, \ldots, M_K$ from the input image I using a noise modeling.

For instance, in case of Poisson noise, we apply the Anscombe transform to the data (i.e. $\mathcal{A}(I) = 2\sqrt{I + \frac{3}{8}}$). Then we detect the significant coefficients with the kth transform T_k, assuming a Gaussian noise with standard deviation equals to 1, in $T_k\mathcal{A}(I)$ instead of $T_k I$. $M_k(j, x, y) = 1$ if a coefficient in band j at pixel position (x, y) is detected , and $M_k(j, x, y) = 0$ otherwise. For the band J which corresponds to the smooth array in transforms such as the wavelet or the curvelet transform, we force $M_k(J, x, y) = 1$ for all (x, y).

Then, we propose to solve the following optimization problem:

$$\min \mathcal{S}(\tilde{O}), \quad \text{subject to} \quad \tilde{O} \in C, \tag{20}$$

where $\mathcal{S}$ is an edge preservation penalization term defined by:

$$\mathcal{S}(\tilde{O}) = \int \| \nabla \tilde{O} \|_p,$$

with $p = 1.1$. C is the set of images $\tilde{O}$ which obey the two constraints:

1. $\tilde{O} \geq 0$ (positivity).
2. $M_k T_k I = M_k T_k [P * \tilde{O}]$, for all k.

The second constraint impose the fidelity to the data, or more exactly, to the data significant coefficients, obtained by the different transforms. Non-significant (i.e. noisy) coefficients are not taken into account, preventing any noise amplification in the final algorithm.

The solution is computed by using the projected Landweber method [4]:

$$\tilde{O}^{n+1} = \mathcal{P}_c \left[\tilde{O}^n + \alpha (P^* * \bar{R}^n - \lambda \frac{\partial \mathcal{S}(\tilde{O})}{\partial O}) \right], \tag{21}$$

where $\mathcal{P}_c$ is the projection operator which enforces the positivity (i.e. set to 0 all negative values), $\bar{R}^n$ is the significant residual which is obtained using the following algorithm:

- Set $I_0^n = I^n = P * \tilde{O}^n$.
- For $k = 1, \ldots, K$ do $I_k^n = I_{k-1}^n + T_k^{-1} \left[M_k (T_k I - T_k I_{k-1}^n) \right]$
- The significant residual $\bar{R}^n$ is obtained by: $\bar{R}^n = I_K^n - I^n$.

α is a convergence parameter and λ is the regularization hyperparameter. As the noise is controlled by the multiscale transforms, the regularization parameter does not have the same importance as in standard deconvolution methods. A much lower value is enough to remove the artifacts relative to the use of the wavelets and the curvelets. The positivity constraint can be applied at each iteration.

Figure 23 top shows the Logan-Shepp Phantom and the simulated data, i.e. original image convolved by a Gaussian point spread function (FWHM=3.2) and Poisson noise. Figure 23 bottom shows the deconvolution with (left) a pure wavelet deconvolution method (no penalization term) and (right) the combined deconvolution method (parameter $\lambda = 0.4$).

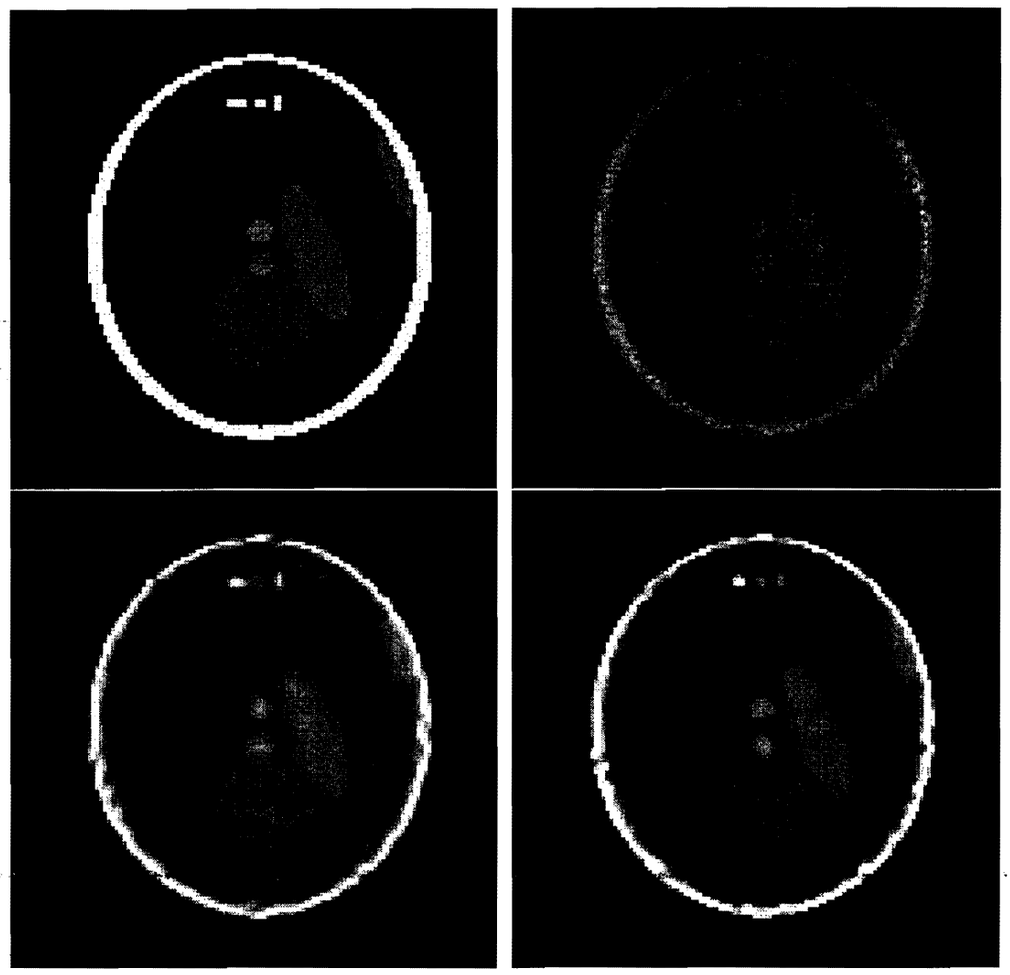


Figure 23: Top, original image (Phantom) and simulated data (i.e. convolved image plus Poisson noise). Bottom, deconvolved image by the wavelet based method and the combined approach.

10 Contrast Enhancement

10.1 Introduction

Because some features are hardly detectable by eye in an image, we often transform it before display. Histogram equalization is one the most well-known methods for contrast enhancement. Such an approach is generally useful for images with a poor intensity distribution. Since edges play a fundamental role in image understanding, a way to enhance the contrast is to enhance the edges. For example, we can add to the original image its Laplacian ($I' = I + \gamma \Delta I$, where γ is a parameter). Only features at the finest scale are enhanced (linearly). For a high γ value, only the high frequencies are visible. Multiscale edge enhancement [61] can be seen as a generalization of this approach to all resolution levels.

In color images, objects can exhibit variations in color saturation with little or no correspondence in luminance variation. Several methods have been proposed in the past for color image enhancement [60]. The retinex concept was introduced by Land [35] as a model for human color constitancy. The single scale retinex (SSR) method [33] consists of applying the following transform to each band i of the color image:

$$R_i(x,y) = \log(I_i(x,y)) - \log(F(x,y) * I_i(x,y)) \tag{22}$$

where $R_i(x,y)$ is the retinex output, $I_i(x,y)$ is the image distribution in the ith spectral band, and F is a Gaussian function. A gain/offset is applied to the retinex output which clips the highest and lowest signal excursions. This can be done by a k-sigma clipping. The retinex method is efficient for dynamic range compression, but does not provide good tonal rendition [43]. The Multiscale Retinex (MSR) combines several SSR outputs to produce a single output image which has both good dynamic range compression and color constancy, and good tonal rendition [32]. The MSR can be defined by:

$$R_{MSR_i} = \sum_{j=1}^{N} w_j R_{i,j} \tag{23}$$

with

$$R_{i,j}(x,y) = \log(I_i(x,y)) - \log(F_j(x,y) * I_i(x,y)) \tag{24}$$

N is the number of scales, $R_{i,j}$ is the ith spectral component of the MSR output, and w_j is the weight associated with the scale j. The Gaussian F_j is given by:

$$F_j(x,y) = K \exp -\frac{r^2}{c_j^2} \tag{25}$$

c_j defines the width of the Gaussian. In [32], three scales were recommended with c_j values equal respectively to 15,80,250, and all weights w_j fixed to $\frac{1}{N}$. The Multiscale Retinex introduces the concept of multiresolution for contrast enhancement. Velde [61] has explicitly introduced the wavelet transform and has proposed an algorithm which modifies the wavelet coefficients in order to amplify faint features. The idea is to first transform the image using the dyadic wavelet transform (two directions per scale). The gradient $G_{j,k}$ at scale j and at pixel location k is calculated at each scale j from the wavelet coefficients $w_{j,k}^{(h)}$ and $w_{j,k}^{(v)}$ relative to the horizontal and vertical wavelet bands: $G_{j,k} = \sqrt{(w_{j,k}^{(h)})^2 + (w_{j,k}^{(v)})^2}$. Then the two

wavelet coefficients at scale j and at position k are multiplied by $y(G_{j,k})$, where y is defined by:

$$\begin{aligned} y(x) &= (\frac{m}{c})^p \text{ if } \mid x \mid < c \\ y(x) &= (\frac{m}{\mid x \mid})^p \text{ if } c \leq \mid x \mid < m \\ y(x) &= 1 \text{ if } \mid x \mid \geq m \end{aligned} \tag{26}$$

Three parameters are needed: p, m and c. p determines the degree of non-linearity in the

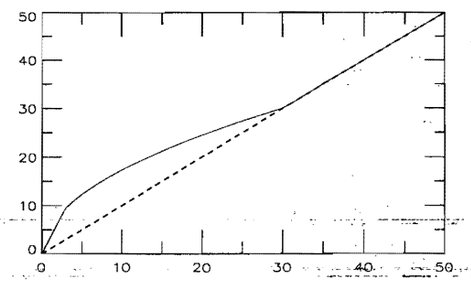


Figure 24: Enhanced coefficients versus original coefficients. Parameters are m=30,c=3 and p=0.5.

nonlinear rescaling of the luminance, and must be in $]0, 1[$. Coefficients larger than m are not modified by the algorithm. The c parameter corresponds to the noise level. Figure 24 shows the modified wavelet coefficients versus the original wavelet coefficients for a given set of parameters ($m = 30$, $c = 3$ and $p = 0.5$). Finally, the enhanced image is obtained by the inverse wavelet transform from the modified wavelet coefficients. For color images, a similar method can be used, but by calculating the multiscale gradient $\Gamma_{j,k}$ from the multiscale gradient of the three L, u, v components: $\Gamma_j(i) = \sqrt{\| G^L_{j,k} \|^2 + \| G^u_{j,k} \|^2 + \| G^v_{j,k} \|^2}$. All wavelet coefficients at scale j and at position k are multiplied by $y(\Gamma_{j,k})$, the enhanced $\tilde{L}$, $\tilde{u}$, $\tilde{v}$ components are reconstructed from the modified wavelet coefficients, and the $(\tilde{L},\tilde{u},\tilde{v})$ image is transformed into an RGB image. More details can be found in [61].

Wavelet bases present some limitations, because they are not adapted to the detection of highly anisotropic elements, such as alignments in an image, or sheets in a cube. Recently, other multiscale systems like ridgelets [5] and curvelets [6, 54] which are very different from wavelet-like systems have been developed. Curvelets and ridgelets take the form of basis elements which exhibit very high directional sensitivity and are highly anisotropic. The curvelet transform uses the ridgelet transform in its digital implementation. We first describe the ridgelet and the curvelet transform, then we show how contrast enhancement can be obtained from the curvelet coefficients.

10.2 Contrast Enhancement using the Curvelet Transform

Since the curvelet transform is well-adapted to represent images containing edges, it is a good candidate for edge enhancement [49, 58]. Curvelet coefficients can be modified in order to enhance edges in an image. A function y_c must be defined which modifies the values of the curvelet coefficients. It could be a function similar to the one defined for the wavelet

coefficients [61] (see equation 26). This function presents however the drawback to amplify the noise (linearly) as well as the signal of interest. We introduce explicitly the noise standard deviation σ in the equation:

$$\begin{aligned} y_c(x,\sigma) &= 1 \text{ if } x < c\sigma \\ y_c(x,\sigma) &= \frac{x - c\sigma}{c\sigma}(\frac{m}{c\sigma})^p + \frac{2c\sigma - x}{c\sigma} \text{ if } x < 2c\sigma \\ y_c(x,\sigma) &= (\frac{m}{x})^p \text{ if } 2c\sigma \leq x < m \\ y_c(x,\sigma) &= (\frac{m}{x})^s \text{ if } x \geq m \end{aligned} \tag{27}$$

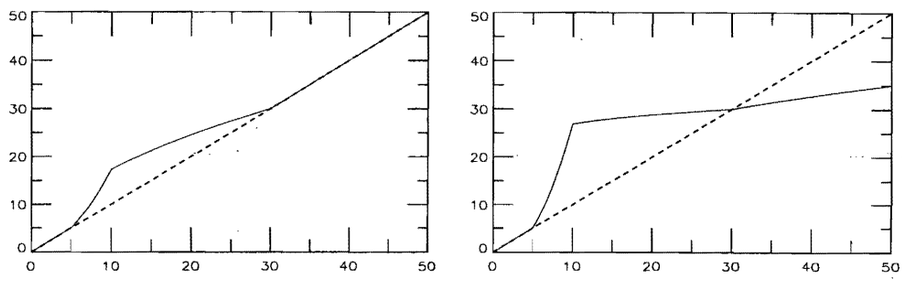


Figure 25: Enhanced coefficients versus original coefficients. Left, parameters are m=30,c=0.5,s=0, and p=0.5. Right, parameters are m=30,c=0.5,s=0.7,p=0.9.

We have fixed $m = c = p = 0.5$ and $s = 0$ in all our experiments. p determines the degree of non-linearity and s introduces a saturation. c becomes a normalized parameter, and a c value larger than 3 guaranties that the noise will not be amplified. The m parameter can be defined either from the noise standard deviation ($m = K_m\sigma$) or from the maximum curvelet coefficient M_c of the relative band ($m = lM_c$, with $l < 1$). The first choice allows the user to define the coefficients to amplify as a function of their signal to noise ratio, while the second one gives an easy and general way to fix the m parameter independently of the range of the pixel values. Figure 25 shows the curve representing the enhanced coefficients versus the original coefficients for two sets of parameters. In the second case, a saturation is added. The curvelet enhancement method for grayscale images consists of the following steps:

1. Estimate the noise standard deviation σ in the input image I.
2. Calculate the curvelet transform of the input image. We get a set of bands w_j, each band w_j contains N_j coefficients and corresponds to a given resolution level.
3. Calculate the noise standard deviation σ_j for each band j of the curvelet transform (see [54] more details on this step).
4. For each band j do
   - Calculate the maximum M_j of the band.
   - Multiply each curvelet coefficient $w_{j,k}$ by $y_c(\mid w_{j,k} \mid, \sigma_j)$.
5. Reconstruct the enhanced image from the modified curvelet coefficients.

For color images, we apply first the curvelet transform on the three components L, u, v. For each curvelet coefficient, we calculate $e = \sqrt{c_L^2 + c_u^2 + c_v^2}$, where (c_L, c_u, c_v) are respectively the curvelet coefficients of the three components, and the modified coefficients are obtained by: $(\tilde{c}_L, \tilde{c}_u, \tilde{c}_v) = (y_c(e,\sigma)c_L, y_c(e,\sigma)c_u, y_c(e,\sigma)c_v)$.

Values in the enhanced components can be larger than the authorized upper limit (in general 255), and we found it necessary to add a final step to our method, which is a sigma-clipping saturation.

Examples

Saturn Image

Figure 26 shows respectively from left to right and from top to bottom the Saturn image, the histogram equalized image, the wavelet multiscale edge enhanced image and the curvelet multiscale edge enhanced image (parameters were $s = 0$, $p = 0.5$, $c = 3$, and $l = 0.5$). The curvelet multiscale edge enhanced image shows clearly better the rings and edges of Saturn.

Satellite Image

Figure 27 shows the results for the enhancement of a grayscale satellite image, and Figure 28 shows the results for the enhancement of a color image (Kodak image of the day 14/05/01) by the retinex, the multiscale retinex and the curvelet multiscale edge enhancement methods. Figures 29 shows the results for the enhancement of a color image (Kodak image of the day 11/12/01).

10.3 Discussion

A number of properties, respected by the curvelet filtering described here, are important for contrast stretching:

1. Noise must not be amplified in enhancing edges.
2. Colors should not be unduly modified. In multiscale retinex, for example, a tendance towards increased grayness is seen. This is not the case using curvelet.
3. It is very advantageous if block effects do not occur. Block overlapping is usually not necessary in curvelet-based contrast enhancement, unlike in the case of noise filtering.

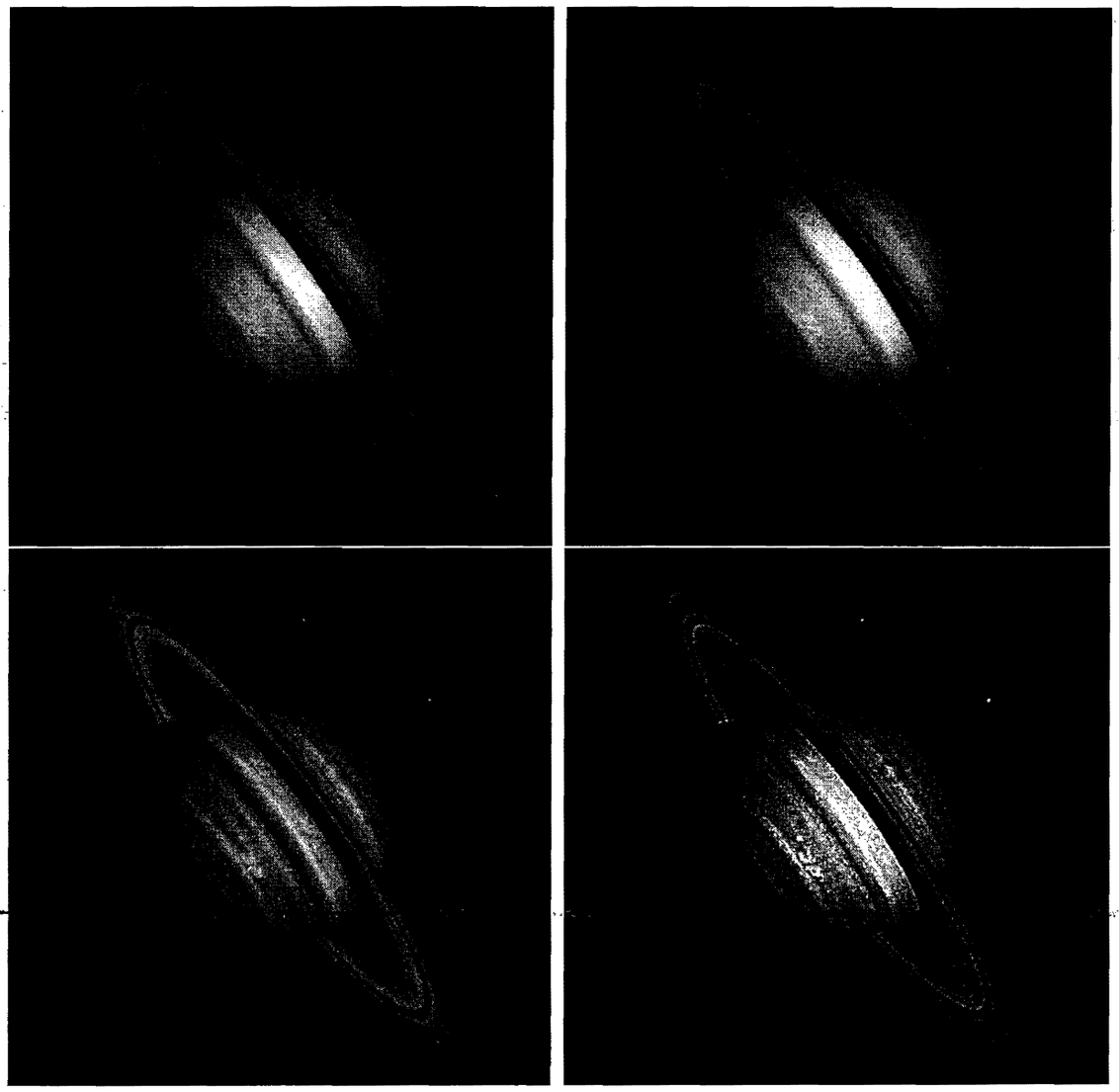


Figure 26: Top, Saturn image and its histogram equalization. Bottom, enhancement image by the wavelet transform and the curvelet transform.

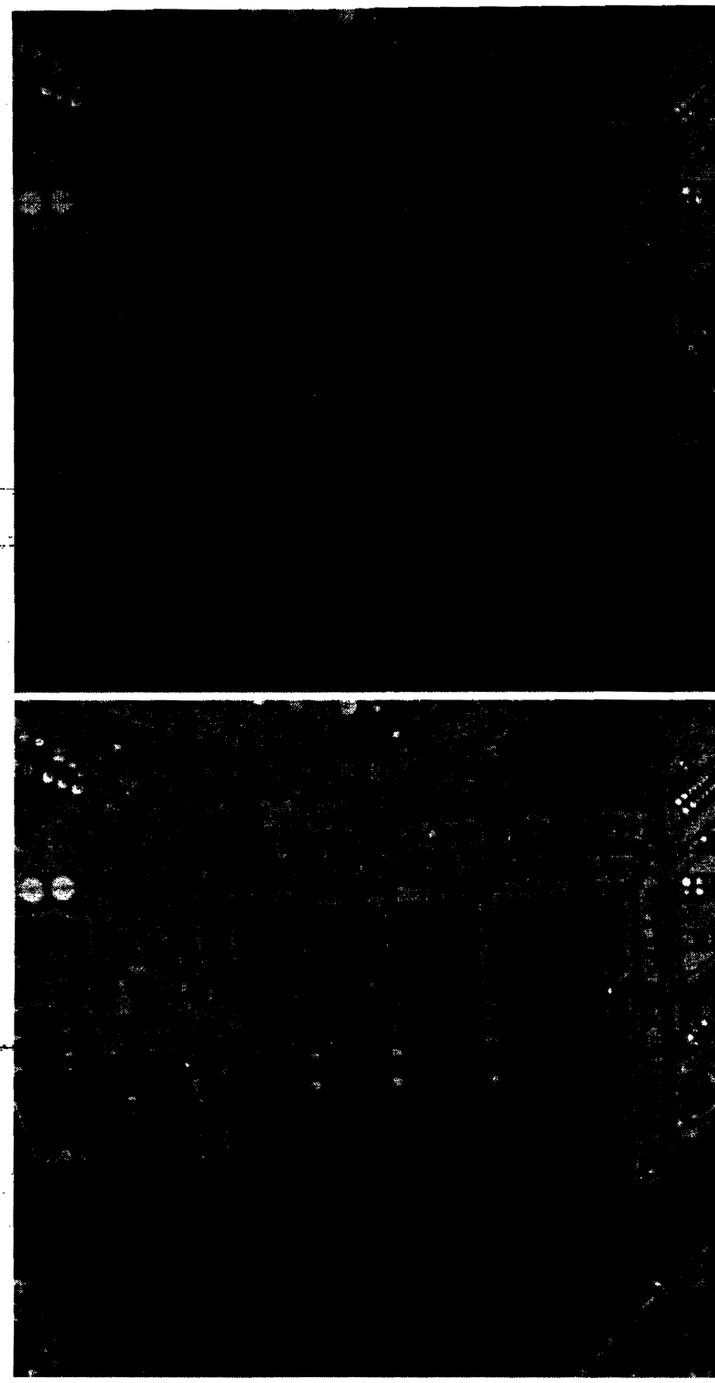


Figure 27: Top, grayscale image, and bottom, curvelet enhanced image.

Figure 28: Top, color image (Kodak picture of the day 14/05/02) and retinex method. Bottom, multiscale retinex method and multiscale edge enhancement.

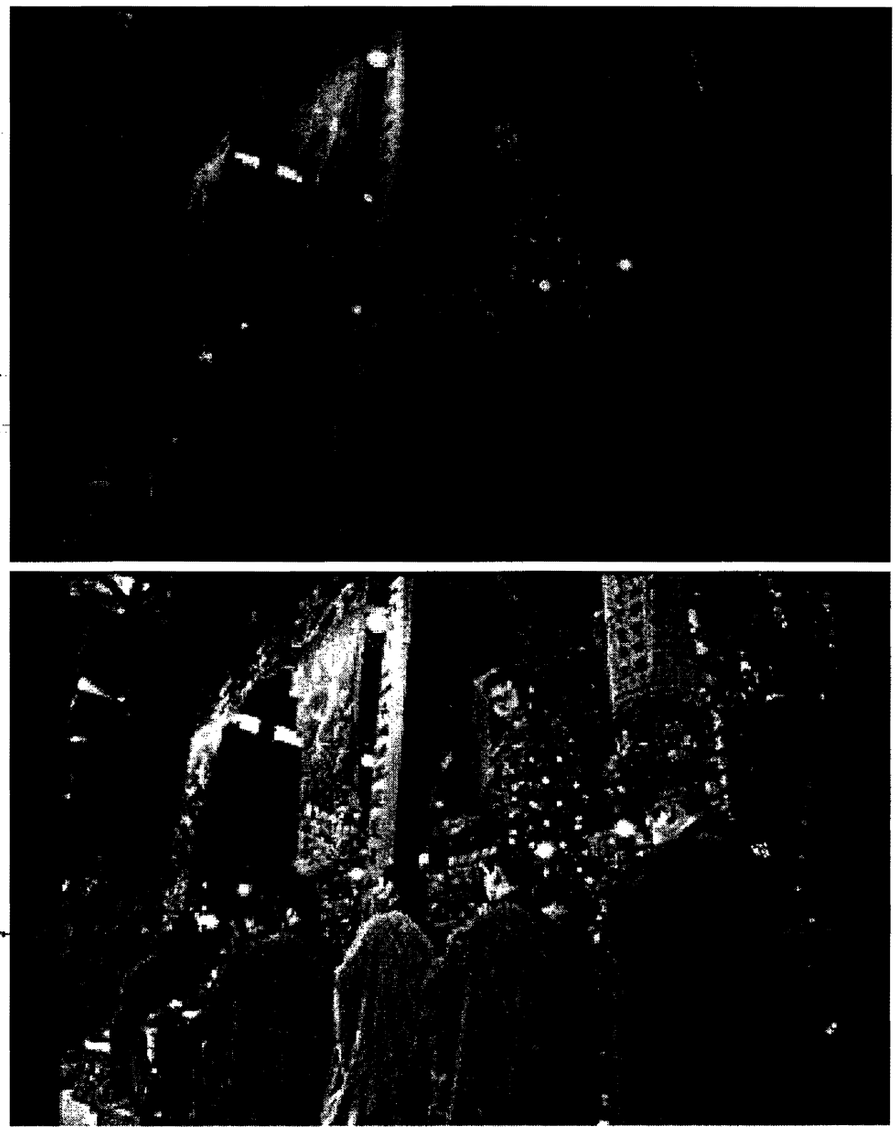


Figure 29: Left, color image (Kodak picture of the day 11/12/01), and right, curvelet enhanced image.

11 Morphological Component Analysis

11.1 Introduction

The content of an image is often complex, and there is not a single transform which is optimal to represent all the contained features. For example, the Fourier transform better represents some textures, while the wavelet transform better represents singularities. Even if we limit our class of transforms to the wavelet one, decision have to be taken between an isotropic wavelet transform which produce good results for isotropic objects (such stars and galaxies in astronomical images, cells in biological images, etc), or an orthogonal wavelet transform, which is better for images with edges. This has motivated the development of different methods [11, 41, 31], and the two most frequently discussed approaches are the Matching Pursuit (MP) [38] and the Basis pursuit (BP) [11]. A dictionary $\mathcal{D}$ being defined as a collection of waveforms $(\varphi_\gamma)_{\gamma\in\Gamma}$, the general principe consists in representing a signal s as a "sparse" linear combination of a small number of basis such that:

$$s = \sum_\gamma a_\gamma \varphi_\gamma \tag{28}$$

or an approximate decomposition

$$s = \sum_{i=1}^{m} a_{\gamma_i} \varphi_{\gamma_i} + R^{(m)}. \tag{29}$$

Matching pursuit [38, 37] method (MP) uses a greedy algorithm which adaptively refines the signal approximation with an iterative procedure:

- Set $s^0 = 0$ and $R^0 = 0$.
- Find the element $\alpha_k \varphi_{\gamma_k}$ which best correlates with the residual.
- Update s and R:

$$\begin{aligned} s^{k+1} &= s^k + \alpha_k \varphi_{\gamma_k} \\ R^{k+1} &= s - s^k. \end{aligned} \tag{30}$$

In case of non orthogonal dictionaries, it has been shown [11] that MP may spend most of the time correcting mistakes made in the first few terms, and therefore is suboptimal in term of sparsity.

Basis pursuit method [11] (BP) is a global procedure which synthesizes an approximation $\tilde{s}$ to s by minimizing a functional of the type

$$\|s - \tilde{s}\|_{\ell_2}^2 + \lambda \cdot \|\alpha\|_{\ell_1}, \quad \tilde{s} = \Phi\alpha. \tag{31}$$

Between all possible solutions, the chosen one has the minimum l^1 norm. This choice of l_1 norm is very important. A l_2 norm, as used in the method of frames [15], does not preserve the sparsity [11].

In many cases, BP or MP synthesis algorithms are computationally very expensive. We present in the following an alternative approach, that we call *Morphological Component Analysis* (MCA), which combines the different available transforms in order to benefit of the advantages of each of them.

11.2 The Combined Transformation

Depending on the content of the data, several transforms can be combined in order to get an optimal representation of all features contained in our data set. In addition to the ridgelet and the curvelet transform, we may want to use the à trous algorithm which is very well suited to astronomical data, or the undecimated wavelet transform which is commonly used in the signal processing domain.

Other transform such wavelet packets, the Fourier transform, the Pyramidal median transform [57], or other multiscale morphological transforms, could also be considered. However, we found that in practice, these four transforms (i.e. curvelet, ridgelet, à trous algorithm, and undecimated wavelet transform) furnishes a very large panel of waveforms which is generally large enough to well represents all features contained in the data.

In general, suppose that we are given K linear transforms $T_1, \ldots, T_K$ and let α_k be the coefficient sequence of an object x after applying the transform T_k, i.e. $\alpha_k = T_k x$. We will suppose that for each transform T_k we have available a reconstruction rule that we will denote by T_k^{-1} although this is clearly an abuse of notations.

Therefore, we search a vector $\alpha = \alpha_1, \ldots, \alpha_K$ such that

$$s = \Phi\alpha \tag{32}$$

where $\Phi\alpha = \sum_{k=1}^{K} T_k^{-1}\alpha_k$. As our dictionary is overcomplete, there is an infinity of vectors verifing this condition, and we need to solve the following optimization problem:

$$min \parallel s - \phi\alpha \parallel^2 + \mathcal{C}(\alpha) \tag{33}$$

where $\mathcal{C}$ is a penalty term. We easily see that chosing $\mathcal{C}(\alpha) = \parallel \alpha \parallel_{l_1}$ leads to the BP method, where the dictionary $\mathcal{D}$ is only composed of the basis elements of the chosen transforms.

Two iterative methods, *soft-MCA* and *hard-MCA*, allowing us to realize such a combined transform, are described in this section.

11.3 Soft-MCA

Noting $T_1, ..., T_K$ the K transform operators, a solution α is obtained by minimizing a functional of the form:

$$J(\alpha) = \parallel s - \sum_{k=1}^{K} T_k^{-1}\alpha_k \parallel_2^2 + \lambda \sum_k \parallel \alpha_k \parallel_1 \tag{34}$$

where s is the original signal, and α_k are the coefficients obtained with the transform T_k.

An simple algorithm to achieve such an solution is [48, 53]:

1. Initialize $L_{\max}$, the number of iterations N_i, $\lambda = L_{\max}$, and $\delta_\lambda = \frac{L_{\max}}{N_i}$.
2. While $\lambda >= 0$ do
3. For k = 1, .., K do
   - Calculate the residual $R = s - \sum_k T_k^{-1}\alpha_k$.
   - Calculate the transform T_k of the residual: $r_k = T_k R$.
   - Add the residual to α_k: $\alpha_k = \alpha_k + r_k$.

- Soft threshold the coefficient α_k with the λ threshold.

4. $\lambda = \lambda - \delta$, and goto 2.

Figure 30 illustrates the result in the case where the input image contains only lines and Gaussians. In this experiment, we have initialized $L_{\max}$ to 20, and δ to 2 (10 iterations). Two transform operators were used, the à trous wavelet transform and the ridgelet transform. The first is well adapted to the detection of Gaussian due to the isotropy of the wavelet function [57], while the second is optimal to represent lines [8]. Figure 30 top, bottom left, and bottom right represents respectively the original image, the reconstructed image from the à trous wavelet coefficient, and the reconstructed image from the ridgelet coefficient. The addition of both reconstructed images reproduces the original one.

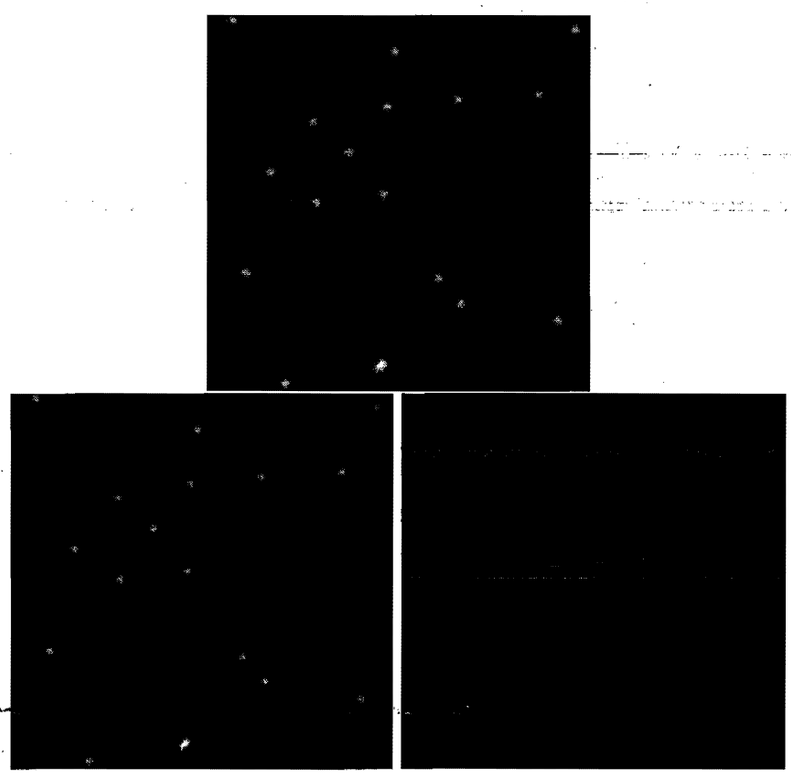


Figure 30: Top, original image containing lines and gaussians. Botton left, reconstructed image for the à trous wavelet coefficient, bottom right, reconstructed image from the ridgelet coefficients.

In some specific cases where the data are sparse in all bases, it has been shown [31, 21] that the solution is identical to the solution when using a $\| \cdot \|_0$ penalty term. This is however generally not the case. The problem we met in image restoration applications, when minimizing equation 34, is that both the signal and noise are split into the bases. The way the noise is distributed in the coefficients α_k is not known, and leads to the problem that we do not know at which level we should threshold the coefficients. Using the threshold we would have used with a single transform makes a strong over-filtering of the data. Using the

l^1 optimization for data restoration implies to first study how the noise is distributed in the coefficients. The hard-MCA method does not present this drawback.

11.4 Hard-MCA

The following algorithm consists in hard thresholding the residual successively on the different bases [48, 53].

1. For noise filtering, estimate the noise standard deviation σ, and set $L_{\min} = k_\sigma$. Otherwise, set $\sigma = 1$ and $L_{\min} = 0$.
2. Initialize $L_{\max}$, the number of iterations N_i, $\lambda = L_{\max}$ and $\delta_\lambda = \frac{L_{\max} - L_{\min}}{N_i}$.
3. Set all coefficients α_k to 0.
4. While $\lambda >= L_{\min}$ do
5. for k = 1, .., K do
   - Calculate the residual $R = s - \sum_k T_k^{-1} \alpha_k$.
   - Calculate the transform T_k of the residual: $r_k = T_k R$.
   - For all coefficients $\alpha_{k,i}$ do
     - Update the coefficients: if $\alpha_{k,i} \neq 0$ or $\mid r_{k,i} \mid > \lambda\sigma$ then $\alpha_{k,i} = \alpha_{k,i} + r_{k,i}$.
6. $\lambda = \lambda - \delta_\lambda$, and goto 5.

For an exact representation of the data, k_σ must be set to 0. Choosing $k_\sigma > 0$ introduces a filtering. If a single transform is used, it corresponds to the standard k-sigma hard thresholding.

It seems that starting with a high enough $L_{\max}$ and a high number of iterations would lead to the l^0 optimization solution, but this remains to be proved.

11.5 Experiments

11.5.1 Experiment 1: Infrared Gemini Data

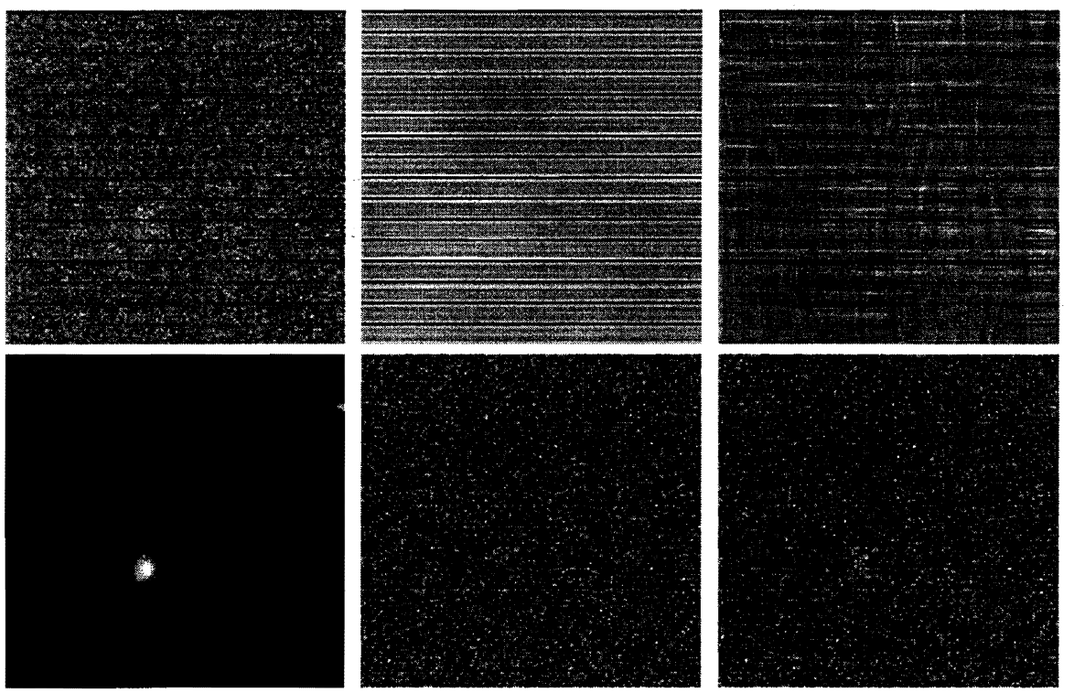


Figure 31: Upper left, galaxy SBS 0335-052 (10 μm), upper middle, upper middle, and bottom left, reconstruction respectively from the ridgelet, the curvelet and wavelet coefficients. Bottom middle, residual image. Bottom right, artifact free image.

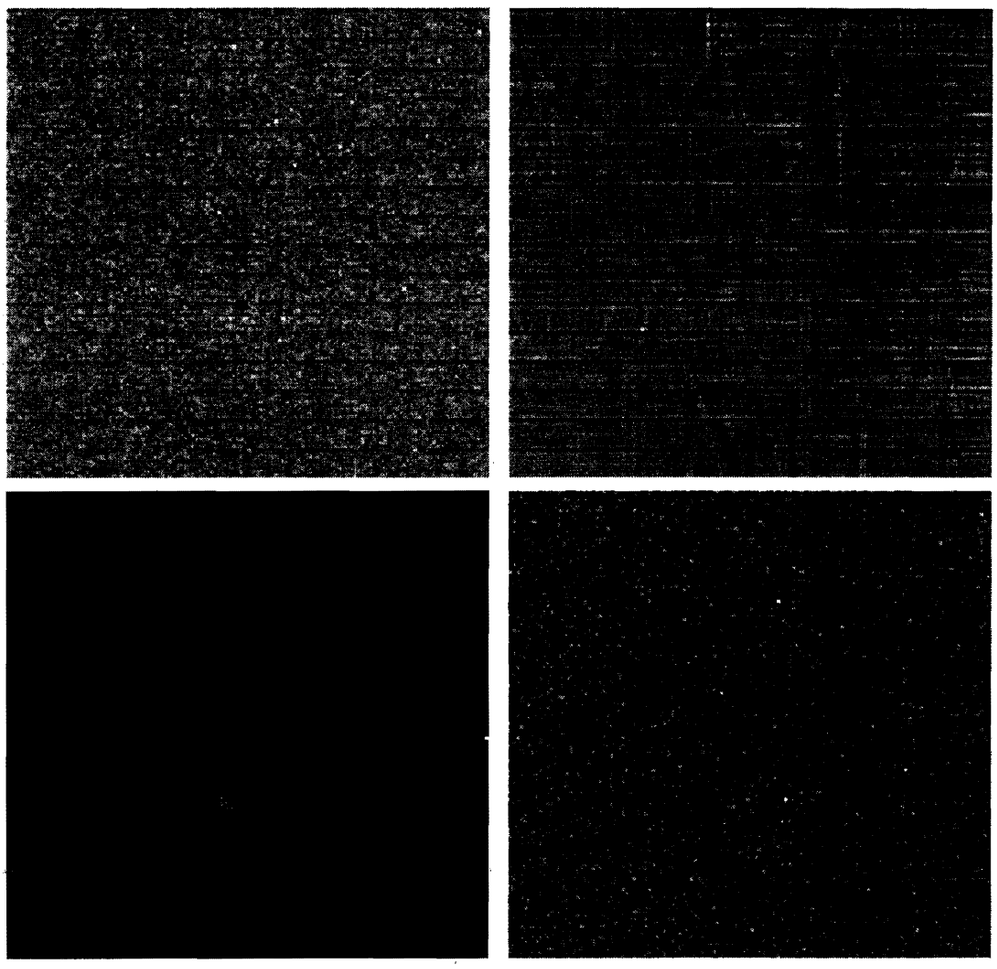


Figure 32: Upper left, galaxy SBS 0335-052 (20 μm), upper right, addition of the reconstructed images from both the ridgelet and the curvelet coefficients, bottom left, reconstruction from the wavelet coefficients, and bottom right, residual image.

Fig. 31 upper left shows a compact blue galaxy located at 53 Mpc. The data have been obtained on ground with the GEMINI-OSCIR instrument at 10 μm. The pixel field of view is $0.089''$/pix, and the source was observed during 1500s. The data are contaminated by a noise and a stripping artifact due to the instrument electronic. The same kind of artifact pattern were observed with the ISOCAM instrument [50].

This image, noted D_{10}, has been decomposed using wavelets, ridgelets, and curvelets. Fig. 31 upper middle, upper right, and bottom left show the three images R_{10}, C_{10}, W_{10} reconstructed respectively from the ridgelets, the curvelets, and the wavelets. Image in Fig. 31 bottom middle shows the residual, i.e. $e_{10} = D_{10} - (R_{10} + C_{10} + W_{10})$. Another interesting image is the artifact free one, obtained by subtracting R_{10} and C_{10} from the input data (see Fig. 31 bottom right). The galaxy has well been detected in the wavelet space, while all stripping artifact have been capted by the ridgelets and curvelets.

Fig. 32 upper left shows the same galaxy, but at 20 μm. We have applied the same decomposition on D_{20}. Fig. 32 upper right shows the coadded image $R_{20} + C_{20}$, and we can see bottom left and right the wavelet reconstruction W_{20} and the residudal $e_{20} = D_{20} - (R_{20} + C_{20} + W_{20})$.

11.5.2 Experiment 2: A370

Figure 33 upper left shows the HST A370 image. It contains many anisotropic features such the gravitationnal arc, and the arclets. The image has been decomposed using three transforms: the ridgelet transform, the curvelet transform, and the à trous wavelet transform. Three images have then been reconstructed from the coefficients of the three basis. Figure 33 upper right shows the coaddition of the ridgelet and curvelet reconstructed images. The à trous reconstructed image is displayed in Figure 33 lower left, and the coaddition of the three images can be seen in Figure 33 lower right. The gravitational arc and the arclets are all represented in the ridgelet and the curvelet basis, while all isotropic features are better represented in the wavelet basis.

We can see that this Morphological Component Analysis (MCA) allows us to separate automatically features in an image which have different morphological aspects. It is very different from other techniques such as Principal Component Analysis or Independent Component Analysis [10] where the separation is performed via statistical properties.

Figure 33: Top left, HST image of A370, top right coadded image from the reconstructions from the ridgelet and the curvelet coefficients, bottom left reconstruction from the à trous wavelet coefficients, and bottom right addition of the three reconstructed images.

12 Program

12.1 Ridgelet transform

12.1.1 im_radon

Program *im_radon* makes a (inverse-) Radon transform of a square $n \times n$ image. The output file which contains the transformation has a suffix, .rad. If the output file name given by the user does not contain this suffix, it is automatically added. The ".rad" file is a FITS format file, and can be manipulated by any package dealing with FITS format, or can be converted to another format using the *im_convert* program. For the two first Radon transform methods, the user can change the number of directions and the resolution. For other methods, the number of directions and the resolution are fixed, and the x and y options are not valid. Option f, w and s allow the user to perform a filtering-backprojection. They are valid only when the selected Radon method is the second one. "w" fixes the width of the filter and the "s" is the sigma parameter of the Gaussian filter.

USAGE: im_radon options image_in trans_out

where options are:

- **[-m type_of_radon_method]**

  1. Radon transform (resp. backprojection) in spatial domain.
     By default, the output image is $2n \times n$ image.
  2. Radon projection in spatial domain and reconstruction in Fourier domain. By default, the output image is $2n \times n$ image. The reconstruction is available only for image with a size n being a power of 2.
  3. Radon transformation and reconstruction in Fourier space (i.e. Linogram).
     The output is $2n \times n$ image. The number of lines is multiplied by two.
  4. Finite Radon Transform.
     The output image is a $(n+1) \times n$ image. The input image size n must be a prime number.
  5. Slant Stack Radon transform.
     The output image has twice the number of lines and twice the same number of columns of the input image. The output is a $2n \times 2n$ image. The reconstruction is not available with this transform.

  Default is Radon transformation and reconstruction in Fourier space.

- **[-y OutputLineNumber]**
  For the RADON transform, OutputLineNumber = number of projection, and default is twice the input image lines number. Only valid for Radon methods 1 and 2.

- **[-x OutputColumnNumber]**
  For the RADON transform, OutputLineNumber = number of pixels per projection, and default is the input image column number. Only valid for Radon methods 1 and 2.

- **[-r]**
  Inverse RADON transform.

- **[-f]**
  Filter each scan of the Radon transform. Only valid for Radon method 2.

- **[-w FilterWidth]**
  Filter width. Only valid for Radon method 2. Default is 100.
- **[-s SigmaParameter**
  Sigma parameter for the filtering. Only valid for Radon method 2. Default is 10.
- **[-v]**
  Verbose. Default is no

Examples:

- im_radon image.fits trans
  Apply the Radon transform to an image.
- im_info -r trans.rad rec
  Reconstruct an image from its Radon transform.

12.1.2 rid_trans

Program *rid_trans* makes the ridgelet transform (and the inverse when -r option is set). The output file which contains the transformation has a suffix, .rid. If the output file name given by the user does not contain this suffix, it is automatically added. The ".rid" file is a FITS format file, and can be manipulated by any package dealing with FITS format, or can be converted to another format using the *im_convert* program. The default transform is the second one (RectoPolar Ridgelets using a FFT based WT). The two first transform are based on the RectoPolar (i.e. linogram) radon transform, but the first applies a standard bi-orthogonal wavelet transform (WT) on the Radon image lines, while the second one use the Fourier based WT which introduces a redundancy of 2. For an $n \times n$, the output has $2n$ lines and n column with the first transform, and is a $2n \times 2n$ image for the second. When the overlapping is set, the size is doubled in each direction.

USAGE: rid_trans options image_in trans_out

where options are:

- **[-t type_of_ridgelet]**
  1. RectoPolar Ridgelet Transform using a standard bi-orthogonal WT.
  2. RectoPolar Ridgelet Transform using a FFT based Pyramidal WT.
  3. Finite ridgelet transform.

  Default is 2.
- **[-n number_of_scale]**
  Number of scales used in the wavelet transform. Default is automatically calculated.
- **[-b BlockSize]**
  Block Size. Default is image size.
- **[-i]**
  Print statistical information about each band. Default is no.
- **[-O]**
  No block overlapping. Default is no. When this option is set, the number of lines and columns is multiplied by two.

- **[-r]**
  Inverse RIDGELET transform.

- **[-x]**
  Write all bands separately as images in the FITS format with prefix 'band_j' (j being the band number).

- **[-v]**
  Verbose. Default is no

Examples:

- rid_transform image.fits trans
  Apply the Ridgelet transform to an image.

- rid_transform -r trans.rid rec
  Reconstruct an image from its Ridgelet transform.

12.1.3 rid_stat

Program *rid_stat* makes the ridgelet transform, and gives statistical information on the ridgelet coefficients. At each scale, it caculates the standard deviation, the skewness, the kurtosis, the minimum, and the maximum. The output file is a fits file containing a two dimensional array $T[J-1,5]$ (J being the number of scales), with the following the syntax:

- $T[j,0]$ = standard deviation of the jth ridgelet band.
- $T[j,1]$ = skewness of the jth ridgelet band.
- $T[j,2]$ = kurtosis of the jth ridgelet band.
- $T[j,3]$ = minimum of the jth ridgelet band.
- $T[j,4]$ = maximum of the jth ridgelet band.

The last ridgelet scale is not used. If the "-A" option is set, these statistics are calculated only the ridgelet coefficients relative the specified angle.

USAGE: rid_stat options image_in trans_out

where options are:

- **[-t type_of_ridgelet]**

  1. RectoPolar Ridgelet Transform using a standard bi-orthogonal WT.
  2. RectoPolar Ridgelet Transform using a FFT based Pyramidal WT.
  3. Finite ridgelet transform.

  Default is 2.

- **[-n number_of_scale]**
  Number of scales used in the wavelet transform. Default is automatically calculated.

- **[-b BlockSize]**
  Block Size. Default is 16.

- **[-O]**
  Use overlapping block. Default is no.
- **[-A Angle]**
  Statistic for a given angle. The value must be given in degrees. Default is no, statistics are calculated from all coefficients.
- **[-v]**
  Verbose. Default is no

Example:

- rid_stat -v image.fits tabstat

12.1.4 rid_filter

Program *rid_filter* filters an image using the ridgelet transform.

USAGE: rid_filter options image_in imag_out

where options are

- **[-t type_of_ridgelet]**
  1. RectoPolar Ridgelet Transform using a standard bi-orthogonal WT.
  2. RectoPolar Ridgelet Transform using a FFT based Pyramidal WT.
  3. Finite ridgelet transform.

  Default is 2.
- **[-n number_of_scale]**
  Number of scales used in the wavelet transform. Default is automatically calculated.
- **[-h]**
  Apply the ridgelet transform only on the high frequencies. Default is no.
- **[-b BlockSize]**
  Block Size. Default is image size.
- **[-F FirstDetectionScale]**
  First detection scale. Default is 1.
- **[-s Nsigma]**
  False detection rate. The false detection rate for a detection is given

$$\epsilon = \mathrm{erfc}(NSigma/\sqrt{2}) \tag{35}$$

  Nsigma parameter allows us to express the false detection rate even if it is not Gaussian noise.
  Default is 3.
- **[-g sigma]**
  Gaussian noise: sigma = noise standard deviation.
  Default is automatically estimated.
- **[-p]**
  Poisson noise.

- [-O]
  Do not apply block overlapping. By default, block overlapping is used.
- [-v]
  Verbose.

Examples:

- rid_filter image.fits fima
  Filter an image using all default options.
- rid_filter -h -s5 image.fits fima
  Five sigma filtering, filtering only the high frequencies.

12.2 Curvelet transform

12.2.1 cur_trans

Program *cur_trans* makes the curvelet transform (and the inverse when -r option is set). The output file which contains the transformation has a suffix, .cur. If the output file name given by the user does not contain this suffix, it is automatically added. The ".cur" file is a 3D FITS format file, and can be manipulated by any package dealing with FITS format. The curvelet transform uses the ridgelet transform, and the default ridgelet transform is the RectoPolar one with a FFT based Pyramidal WT.

USAGE: cur_trans options image_in trans_out

where options are:

- [-t **type_of_ridgelet**]
  1. RectoPolar Ridgelet Transform using a standard bi-orthogonal WT.
  2. RectoPolar Ridgelet Transform using a FFT based Pyramidal WT.
  3. Finite ridgelet transform.

  Default is 2.
- [-n **number_of_scale**]
  Number of scales used in the 2D wavelet transform. Default is 4.
- [-N **number_of_scale**]
  Number of scales used in the ridgelet transform. Default is automatically calculated.
- [-b **BlockSize**]
  Block Size. Default is 16.
- [-r]
  Inverse CURVELET transform.
- [-i]
  Print statistical information about each band. Default is no.
- [-O]
  Block overlapping. Default is no.

- **[-x]**
  Write all bands separately as images in the FITS format with prefix 'band_j' (j being the band number).

- **[-v]**
  Verbose. Default is no.

Examples:

- cur_trans -i image.fits trans
  Curvelet transform of an image.

- cur_trans -r trans.cur rec
  Image reconstruction from its curvelet transform.

12.2.2 cur_stat

Program *cur_stat* makes the curvelet transform, and gives statistical information on the curvelet coefficients. At each scale, it caculates the standard deviation, the skewness, the kurtosis, the minimum, and the maximum. The output file is a fits file containing a two dimensional array $T[J-1,5]$ (J being the number of bands), with the following the syntax:

- $T[j,0]$ = standard deviation of the jth ridgelet band.
- $T[j,1]$ = skewness of the jth ridgelet band.
- $T[j,2]$ = kurtosis of the jth ridgelet band.
- $T[j,3]$ = minimum of the jth ridgelet band.
- $T[j,4]$ = maximum of the jth ridgelet band.

USAGE: cur_stat options image_in trans_out

where options are:

- **[-n number_of_scale]**
  Number of scales used in the wavelet transform. Default is automatically calculated.

- **[-b BlockSize]**
  Block Size. Default is 16.

- **[-O]**
  Use overlapping block. Default is no.

- **[-v]**
  Verbose. Default is no

Example:

- cur_stat -v image.fits tabstat

12.2.3 cur_filter

Program *cur_filter* filters an image using the curvelet transform.

USAGE: cur_filter options image_in imag_out

where options are

- **[-t type_of_ridgelet]**
  1. RectoPolar Ridgelet Transform using a standard bi-orthogonal WT.
  2. RectoPolar Ridgelet Transform using a FFT based Pyramidal WT.
  3. Finite ridgelet transform.

  Default is 2.
- **[-n number_of_scale]**
  Number of scales used in the 2D wavelet transform. Default is 4.
- **[-N number_of_scale]**
  Number of scales used in the ridgelet transform. Default is automatically calculated.
- **[-b BlockSize]**
  Block Size. Default is 16.
- **[-g sigma]**
  Gaussian noise: sigma = noise standard deviation.
  Default is automatically estimated.
- **[-s Nsigma]**
  False detection rate.
  Default is 3.
- **[-O]**
  Do not apply block overlapping. By default, block overlapping is used.
- **[-P]**
  Supress the positivity constraint. Default is no.
- **[-I NoiseFileName]**
  If the noise is stationary, the program can estimated the correct thresholds from a noise realization.
- **[-v]**
  Verbose

Examples:

- cur_filter image.fits sol
  Curvelet filtering of an image.
- cur_filter -n 5 -s4 image.fits sol
  Curvelet filtering of an image, using five resolution levels, and a 4sigma detection.

12.2.4 cur_colfilter

Program *cur_colfilter* filters an color image using the curvelet transform.

USAGE: cur_colfilter options image_in imag_out

where options are

- **[-n number_of_scale]**
  Number of scales used in the 2D wavelet transform. Default is 4.
- **[-N number_of_scale]**
  Number of scales used in the ridgelet transform. Default is automatically calculated.
- **[-b BlockSize]**
  Block Size. Default is 16.
- **[-g sigma]**
  Gaussian noise: sigma = noise standard deviation.
  Default is automatically estimated.
- **[-s Nsigma]**
  False detection rate.
  Default is 3.
- **[-O]**
  Do not apply block overlapping. By default, block overlapping is used.
- **[-v]**
  Verbose

Example:

- cur_colfilter image.fits sol
  Curvelet transform of a color image.

12.2.5 cur_contrast

Program *cur_contrast* filters an color image using the curvelet transform.

USAGE: cur_contrast options image_in imag_out

where options are

- **[-n number_of_scales]**
  Number of scales used in the wavelet transform. Default is 4.
- **[-N number_of_scales]**
  Number of scales used in the ridgelet transform. Default is automatically calculated.
- **[-b BlockSize]**
  Block size used by the curvelet transform. Default is 16.
- **[-O]**
  Use overlapping block. Default is no.
- **[-g sigma]**
  Noise standard deviation. Only used when is filtering is performed. Default is automatically estimated.

- **[-s NSigmalLow]**
  Coefficient < NSigmalLow*SigmaNoise are not modified. default is 5.
- **[-S NSigmalUp]**
  Coefficient > NSigmalUp*SigmaNoise are not modified. default is 20.
- **[-M MaxCoeff]**
  If MaxBandCoef is the maximum coefficient in a given curvelet band, Coefficient > MaxBandCoef*MaxCoeff are not modified. default is 0.5.
- **[-P P_parameter]**
  Determine the degree on non-linearity. P must be in]0,1[. Default is 0.5.
- **[-T P_parameter]**
  Curvelet coefficent saturation parameter. T must be in [0,1]. Default is 0.
- **[-c]**
  By default a sigma clipping is performed. When this option is set, no sigma clipping is performed.
- **[-K ClippingValue]**
  Clipping value. Default is 3.
- **[-L Saturation]**
  Saturate the reconstructed image. Coeff larger than Saturation*MaxData are set to Saturation*MaxData. Default is 1. If L is set to 0, then no saturation is applyied.

Examples:

- cur_contrast image.fits image_out.fits
  Enhance the contrast using the curvelet transform.
- cur_contrast -O image.fits image_out.fits
  Enhance the contrast using the curvelet transform abd block overlapping.
- cur_contrast -M 0.8 image.fits image_out.fits
  Enhance more the contrast.

12.2.6 cur_colcontrast

The program *col_colcontrast* enhances the contrast of a color image using the curvelet transform. The command line is:

USAGE: cur_colcontrast option in_image out_image

where options are:

- **[-n number_of_scales]**
  Number of scales used in the wavelet transform. Default is 4.
- **[-N number_of_scales]**
  Number of scales used in the ridgelet transform. Default is automatically calculated.
- **[-b BlockSize]**
  Block size used by the curvelet transform. Default is 16.
- **[-O]**
  Use overlapping block. Default is no.

- **[-g sigma]**
  Noise standard deviation. Default is automatically estimated.
- **[-s NSigmalLow]**
  Coefficient < NSigmalLow*SigmaNoise are not modified. default is 5.
- **[-S NSigmalUp]**
  Coefficient > NSigmalUp*SigmaNoise are not modified. default is 20.
- **[-M MaxCoeff]**
  If MaxBandCoef is the maximum coefficient in a given curvelet band, Coefficient > MaxBandCoef*MaxCoeff are not modified. default is 0.5.
- **[-P P_parameter]**
  Determine the degree on non-linearity. P must be in]0,1[. Default is 0.5.
- **[-T P_parameter]**
  Curvelet coefficent saturation parameter. T must be in [0,1]. Default is 0.
- **[-c]**
  By default a sigma clipping is performed. When this option is set, no sigma clipping is performed.
- **[-K ClippingValue]**
  Clipping value. Default is 3.
- **[-L Luminance_Saturation]**
  Value in the luminance map which are larger Saturation*MaxData are set to Saturation*MaxData. Default is 1.

Examples:

- cur_colcontrast image.tiff image_out.tiff
  Enhance the contrast using the curvelet transform.
- cur_colcontrast -n 5 image.tiff image_out.tiff
  Ditto, but use five scales instead of four.
- cur_colcontrast -M 0.9 image.tiff image_out.tiff
  Enhance more the contrast.

12.3 Combined Filtering

12.3.1 cb_filter

Program *cb_filter* filters an image corrupted by Gaussian noise by the combined filtering method. By default, two undecimated bi-orthogonal WT and the curvelet transform are used. The number of iterations is defaulted 10. In general, the algorithm converges with less than six iterations. If the "-T" option is set, the total variation is minimized instead of the l_1 norm of the multiscale coefficients. A deconvolution can also be performed using the "-P" option. In this case, a division is first done in Fourier space between the input image and the Point Spread Function. All Fourier component with a norm lower than ϵ (default value is 10^{-3}) are set zero. Then the deconvolved image is filtered by the combined filtering method using the new noise properties (still Gaussian, but not white).

USAGE: cb_filter options image_in trans_out

where options are:

- **[-t TransformSelection]**
  1. A trous algorithm
  2. bi-orthogonal WT with 7/9 filters
  3. Ridgelet transform
  4. Curvelet transform
  5. Mirror Basis WT
- **[-n number_of_scales]**
  Number of scales used in the WT, the a trous and the curvelet transform. Default is 4.
- **[-b BlockSize]**
  Block Size in the ridgelet transform. Default is image size.
- **[-i NbrIter]**
  Number of iteration. Default is 10.
- **[-F FirstDetectionScale]**
  First detection scale in the ridgelet transform. Default is 1.
- **[-k]**
  Kill the last scale in ridgelet. Default no.
- **[-K]**
  Kill last scale in the atrous algorithm and the curvelet. Default no.
- **[-L FirstSoftThreshold]**
  First soft thresholding value.Default is 0.5.
- **[-l LastSoftThreshold]**
  Last soft thresholding value.Default is 0.5.
- **[-u]**
  Number of undecimated scales in the WT. Default is 1.
- **[-s Nsigma]**
  False detection rate. Default is 4.
- **[-g sigma]**
  Gaussian noise: sigma = noise standard deviation.
  Default is automatically estimated.
- **[-O]**
  No block overlapping. Default is no.
- **[-T]**
  Minimize the Total Variation instead of the L1 norm.
- **[-P PsfFile]**
  Apply a deconvolution using the PSF in the file PsfFile.
- **[-e Eps]**
  Remove frequencies with $|PP^*| < \epsilon$. Default is 10^{-3}.
- **[-C TolCoef]**
  Default is 0.5.
- **[-v]**
  Verbose. Default is no.

Example:

- cb_filter image.fits sol
  Image filtering by the combined filtering method, using both the curvelet and wavelet transform.

12.4 Morphological Component Analysis

12.4.1 cb_mca

Program *cb_mca* decomposes an image on a set of basis. The output image is the coaddition of all reconstructed images from the different bases. As ech transform may have a lot of parameters, we have decided to fixe many of them, and to let the user select those we consider as the most important. Some options are effective for the several transforms.

USAGE: cb_mca options image_in imag_out

where options are

- **[-t TransformSelection]**
  1. A trous algorithm
  2. Ridgelet transform
  3. Multiscale Ridgelet
  4. bi-orthogonal WT with 7/9 filters
  5. Cosinus transform
  6. Curvelet transform
  7. Pyramidal Median transform
- **[-n number_of_scales]**
  Number of scales used in the WT, the a trous, the PMT and the curvelet transform. Default is 4. Number of ridgelet in the multi-ridgelet transform.
- **[-b BlockSize]**
  Block Size in the ridgelet transform. Default is image size. Starting Block Size in the multi-ridgelet transform. Default is 8.
- **[-i NbrIter]**
  Number of iteration. Default is 10.
- **[-F FirstDetectionScale]**
  First detection scale in the (multi-) ridgelet transform. Default is 1.
- **[-k]**
  Kill last scale in ridgelet, and multiscale ridgelet transform. Default no.
- **[-K]**
  Kill last scale in the atrous algorithm, the PMT, and the curvelet. Default no.
- **[-L FirstSoftThreshold]**
  First soft thresholding value.Default is 0.5.

- **[-L LastSoftThreshold]**
  Last soft thresholding value.Default is 0.5.

- **[-f]**
  Apply the noise modeling (filtering). Default is no.

- **[-u]**
  Number of undecimated scales in the WT. Default is 1.

- **[-w]**
  Write to the disk the reconstructed image from each decomposition. File names are:

  - "xx_atrou.fits": for the à trous algorithm
  - "xx_rid.fits": for the ridgelet transform
  - "xx_mrid.fits": for the multiscale ridgelet transform, and "xx_mrid_rj.fits" for one of its ridgelet transform, where j is the transform number.
  - "xx_cur.fits": for the curvelet transform
  - "xx_cos.fits": for the cosinus transform
  - "xx_wt.fits": for the orthogonal wavelet transform
  - "xx_pmt.fits": for the pyramidal median transform

- **[-s Nsigma]**
  False detection rate.
  Default is 3.

- **[-g sigma]**
  Gaussian noise: sigma = noise standard deviation.
  Default is automatically estimated.

- **[-p]**
  Poisson Noise. Default is no (Gaussian).

- **[-O]**
  No block overlapping. Default is no.

- **[-P]**
  Supress the positivity constraint. Default is no.

- **[-v]**
  Verbose. Default is no.

References

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

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

[3] A. Averbuch, R.R. Coifman, D.L. Donoho, M. Israeli, and J. Waldén. Fast Slant Stack: A notion of Radon transform for data in a cartesian grid which is rapidly computible, algebraically exact, geometrically faithful and invertible. *SIAM J. Sci. Comput.*, 2001. To appear.

[4] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging.* Institute of Physics, 1998.

[5] E. J. Candès. Harmonic analysis of neural netwoks. *Applied and Computational Harmonic Analysis*, 6:197–218, 1999.

[6] E. J. Candès and D. L. Donoho. Curvelets – a surprisingly effective nonadaptive representation for objects with edges. In A. Cohen, C. Rabut, and L.L. Schumaker, editors, *Curve and Surface Fitting: Saint-Malo 1999*, Nashville, TN, 1999. Vanderbilt University Press.

[7] E. J. Candès and D. L. Donoho. Ridgelets: the Key to Higher-dimensional Intermittency? *Philosophical Transactions of the Royal Society of London A*, 357:2495–2509, 1999.

[8] E.J. Candès and D. Donoho. Ridgelets: the key to high dimensional intermittency? *Philosophical Transactions of the Royal Society of London A*, 357:2495–2509, 1999.

[9] E.J. Candès and F. Guo. New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction. *Signal Processing*, 82(11):1519–1543, 2002.

[10] J.F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 86:2009–2025, October 1998.

[11] S.S. Chen, D.L. Donoho, and M.A. Saunder. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[12] R.R. Coifman and D.L. Donoho. Translation invariant de-noising. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, pages 125–150. Springer-Verlag, 1995.

[13] 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.

[14] USC-SIPI Image Database. `http://sipi.usc.edu/services/database/-Database.html`.

[15] I. Daubechies. Time-frequency localization operators: A geometric phase space approach. *IEEE Transactions on Information Theory*, 34:605–612, 1988.

[16] S.R. Deans. *The Radon Transform and Some of Its Applications.* Wiley, 1983.

[17] P. Dhérété, S. Durand, J. Froment, and B. Rougé. A best wavelet packet basis for joint image deblurring-denoising and compression. In *SPIE's 47th Annual Meeting, Seattle, July 7-11*, 2002.

[18] M. N. Do and M. Vetterli. Image denoising using the orthonormal finite ridgelet transform. In *SPIE conference on Signal and Image Processing: Wavelet Applications in Signal and Image Processing VIII*, August 2000.

[19] M. N. Do and M. Vetterli. Orthonormal finite ridgelet transform for image compression. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, September 2000.

[20] D. Donoho and A.G. Flesia. Digital Ridgelet Transform Based on True Ridge Functions. In J. Schmeidler and G.V. Welland, editors, *Beyond Wavelets*. Academic Press, 2002.

[21] D. Donoho and X. Huo. Uncertainty Principles and Ideal Atomic Decomposition. *IEEE Transactions on Information Theory*, 47(7), 2001.

[22] D. L. Donoho. Fast ridgelet transforms in dimension 2. Technical report, Stanford University, Department of Statistics, Stanford CA 94305–4065, 1997.

[23] D. L. Donoho. Digital ridgelet transform via rectopolar coordinate transform. Technical report, Stanford University, 1998.

[24] D.L. Donoho. Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data. In American Mathematical Society, editor, *Proceedings of Symposia in Applied Mathematics*, volume 47, pages 173–205, 1993.

[25] D.L. Donoho. Nonlinear solution of inverse problems by wavelet-vaguelette decomposition. *Applied and Computational Harmonic Analysis*, 2:101–126, 1995.

[26] D.L Donoho. Orthonormal ridgelets and linear singularities. *Siam J. Math Anal.*, 31(5):1062–1099, 2000.

[27] S. Durand and J. Froment. Reconstruction of wavelet coefficients using total variation minimization. Technical Report 2001-18, CMLA, November 2001.

[28] P. Edholm and G.T. Herman. Linograms and image reconstruction from projections. *IEEE Transactions on Medical Imaging*, 6(4):301–307, 1987.

[29] P. Edholm and G.T. Herman. Image reconstruction from linograms: Implementation and evaluation. *IEEE Transactions on Medical Imaging*, 7(3):239–246, 1988.

[30] H.J.A.M. Heijmans and J. Goutsias. Multiresolution signal decomposition schemes. Part 2: Morphological wavelets. *IEEE Transactions on Image Processing*, 9:1897–1913, 2000.

[31] Xiaoming Huo. *Sparse Image Representation via Combined Transforms*. PhD thesis, Stanford Univesity, August 1999.

[32] D. J. Jobson, Z. Rahman, and G. A. Woodell. A multi-scale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing*, 6(7):965–976, 1997.

[33] D. J. Jobson, Z. Rahman, and G. A. Woodell. Properties and performance of a center/surround retinex. *IEEE Transactions on Image Processing*, 6(3):451–462, 1997.

[34] J. Kalifa. *Restauration minimax et déconvolution dans une base d'ondelettes miroir.* PhD thesis, Ecole Polytechnique, 5 May 1999.

[35] E. Land. Recent advances in retinex theory. *Vision Research*, 26(1):7–21, 1986.

[36] F. Malgouyres. Mathematical analysis of a model which combines total variation and wavelet for image restoration. *Journal of information processes*, 2(1):1–10, 2002.

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

[38] S.G. Mallat and Z. Zhang. Atomic decomposition by basis pursuit. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.

[39] F. Matus and J. Flusser. Image representations via a finite Radon transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):996–1006, 1993.

[40] R.M. Mersereau and A.V. Oppenheim. Digital reconstruction of multidimensional signals from their projections. *Proc. IEEE*, 62(10):1319–1338, 1974.

[41] F.G. Meyer, A. Averbuch, J-O Stromberg, and R.R. Coifman. Multi-layered image representation: Application to image compression. In *International Conference on Image Processing, ICIP'98, Chicago*, 1998.

[42] R. Neelamani. Wavelet-based deconvolution for ill-conditionned systems. MS thesis, Deptment of ECE, Rice University, 1999.

[43] Z. Rahman, D. J. Jobson, and G. A. Woodell. Multi-scale retinex for color image enhancement. In IEEE, editor, *IEEE international conference on Image Processing*, 1996.

[44] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchival trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250, 1996.

[45] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41:3445–3462, 1993.

[46] E. P. Simoncelli. Bayesian denoising of visual images in the wavelet domain. In P Muller and B. Vidakovic, editors, *Bayesian Inference in Wavelet Based Models.* Springer-Verlag, 1999.

[47] E.P. Simoncelli, W.T Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multi-scale transforms [or "what's wrong with orthonormal wavelets"]. *IEEE Trans. Information Theory*, 1992.

[48] J.-L. Starck. Detection of anisotropic features by the curvelet transform. In J.-L. Starck and F. Murtagh, editors, *SPIE conference on Signal and Image Processing: Astronomical Data Analysis, San Diego, 1-4 August*, volume 4477, pages 131–141. SPIE, 2001.

[49] J.-L. Starck. Image processing by the curvelet transform. In R. Leonardi, editor, *2002 Tyrrhenian International Workshop on Digital Communications, Advanced Methods for Multimedia Signal Processing*. SPIE, 2002.

[50] J.-L. Starck, A. Abergel, H. Aussel, M. Sauvage, R. Gastaud, A. Claret, X. Desert, C. Delattre, and E. Pantin. ISOCAM data processing. *Astronomy and Astrophysics, Supplement Series*, 134:135–148, 1999.

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

[52] J.-L. Starck, A. Bijaoui, B. Lopez, and C. Perrier. Image reconstruction by the wavelet transform applied to aperture synthesis. *Astronomy and Astrophysics*, 283:349–360, 1994.

[53] J.-L. Starck, E. Candes, and D.L. Donoho. Astronomical image representation by the curvelet tansform. *Astronomy and Astrophysics*, 2002. submitted.

[54] J.-L. Starck, E. Candès, and D.L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):131–141, 2002.

[55] J.-L. Starck, D.L. Donoho, and E. Candès. Very high quality image restoration. In A. Laine, M.A. Unser, and A. Aldroubi, editors, *SPIE conference on Signal and Image Processing: Wavelet Applications in Signal and Image Processing IX, San Diego, 1-4 August*. SPIE, 2001.

[56] J.-L. Starck and F. Murtagh. Multiscale entropy filtering. *Signal Processing*, 76:147–165, 1999.

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

[58] J.-L. Starck, F. Murtagh, E. Candes, and D.L. Donoho. Gray and color image contrast enhancement by the curvelet transform. *IEEE Transactions on Image Processing*, 2002. submitted.

[59] J.-L. Starck, F. Murtagh, and R. Gastaud. A new entropy measure based on the wavelet transform and noise modeling. *IEEE Transactions on Circuits and Systems II*, 45:1118–1124, 1998.

[60] A. Toet. Multiscale color image enhancement. In SPIE, editor, *SPIE international conference on Image Processing and its Applications*, pages 583–585, 1992.

[61] K.V. Velde. Multi-scale color image enhancement. In ICIP, editor, *Proceedings of the International Conference on Image Processing*, volume 3, pages 584–587, 1999.

[62] Isao Yamada. The hybrid steepest descent method for the variational inequality problem over the intersection of fixed point sets of nonexpansive mappings. In D. Butnariu, Y. Censor, and S. Reich, editors, *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*. Elsevier, 2001.

Annex A: Wavelet transform using the Fourier Transform

We start with the set of scalar products $c_0(k) =< f(x), \phi(x-k) >$. If $\phi(x)$ has a cut-off frequency $\nu_c \leq \frac{1}{2}$ ([52, 51, 57]), the data are correctly sampled. The data at resolution $j = 1$ are:

$$c_1(k) =< f(x), \frac{1}{2}\phi(\frac{x}{2} - k) > \tag{36}$$

and we can compute the set $c_1(k)$ from $c_0(k)$ with a discrete filter $\hat{h}(\nu)$:

$$\hat{h}(\nu) = \begin{cases} \frac{\hat{\phi}(2\nu)}{\hat{\phi}(\nu)} & \text{if } \mid \nu \mid < \nu_c \\ 0 & \text{if } \nu_c \leq \mid \nu \mid < \frac{1}{2} \end{cases} \tag{37}$$

and

$$\forall \nu, \forall n \quad \hat{h}(\nu + n) = \hat{h}(\nu) \tag{38}$$

where n is an integer. So:

$$\hat{c}_{j+1}(\nu) = \hat{c}_j(\nu)\hat{h}(2^j\nu) \tag{39}$$

The cut-off frequency is reduced by a factor 2 at each step, allowing a reduction of the number of samples by this factor.

The wavelet coefficients at scale $j + 1$ are:

$$w_{j+1}(k) =< f(x), 2^{-(j+1)}\psi(2^{-(j+1)}x - k) > \tag{40}$$

and they can be computed directly from $c_j(k)$ by:

$$\hat{w}_{j+1}(\nu) = \hat{c}_j(\nu)\hat{g}(2^j\nu) \tag{41}$$

where g is the following discrete filter:

$$\hat{g}(\nu) = \begin{cases} \frac{\hat{\psi}(2\nu)}{\hat{\phi}(\nu)} & \text{if } \mid \nu \mid < \nu_c \\ 1 & \text{if } \nu_c \leq \mid \nu \mid < \frac{1}{2} \end{cases} \tag{42}$$

and

$$\forall \nu, \forall n \quad \hat{g}(\nu + n) = \hat{g}(\nu) \tag{43}$$

The frequency band is also reduced by a factor 2 at each step. Applying the sampling theorem, we can build a pyramid of $N + \frac{N}{2} + \ldots + 1 = 2N$ elements. For an image analysis the number of elements is $\frac{4}{3}N^2$. The overdetermination is not very high.

The B-spline functions are compact in direct space. They correspond to the autoconvolution of a square function. In Fourier space we have:

$$\hat{B}_l(\nu) = (\frac{\sin \pi\nu}{\pi\nu})^{l+1} \tag{44}$$

$B_3(x)$ is a set of 4 polynomials of degree 3. We choose the scaling function $\phi(\nu)$ which has a $B_3(x)$ profile in Fourier space:

$$\hat{\phi}(\nu) = \frac{3}{2}B_3(4\nu) \tag{45}$$

In direct space we get:

$$\phi(x) = \frac{3}{8}[\frac{\sin\frac{\pi x}{4}}{\frac{\pi x}{4}}]^4 \tag{46}$$

This function is quite similar to a Gaussian and converges rapidly to 0. For 2-dimensions the scaling function is defined by $\hat{\phi}(u,v) = \frac{3}{2}B_3(4r)$, with $r = \sqrt{(u^2+v^2)}$. This is an isotropic function.

The wavelet transform algorithm with n_p scales is the following:

1. Start with a B_3-spline scaling function and derive ψ, h and g numerically.
2. Compute the corresponding FFT image. Name the resulting complex array T_0.
3. Set j to 0. Iterate:
4. Multiply T_j by $\hat{g}(2^j u, 2^j v)$. We get the complex array W_{j+1}. The inverse FFT gives the wavelet coefficients at scale 2^j;
5. Multiply T_j by $\hat{h}(2^j u, 2^j v)$. We get the array T_{j+1}. Its inverse FFT gives the image at scale 2^{j+1}. The frequency band is reduced by a factor 2.
6. Increment j.
7. If $j \leq n_p$, go back to 4.
8. The set $\{w_1, w_2, \ldots, w_{n_p}, c_{n_p}\}$ describes the wavelet transform.

If the wavelet is the difference between two resolutions, i.e.

$$\hat{\psi}(2\nu) = \hat{\phi}(\nu) - \hat{\phi}(2\nu) \tag{47}$$

and:

$$\hat{g}(\nu) = 1 - \hat{h}(\nu) \tag{48}$$

then the wavelet coefficients $\hat{w}_j(\nu)$ can be computed by $\hat{c}_{j-1}(\nu) - \hat{c}_j(\nu)$.

Reconstruction.

If the wavelet is the difference between two resolutions, an evident reconstruction for a wavelet transform $\mathcal{W} = \{w_1, \ldots, w_{n_p}, c_{n_p}\}$ is:

$$\hat{c}_0(\nu) = \hat{c}_{n_p}(\nu) + \sum_j \hat{w}_j(\nu) \tag{49}$$

But this is a particular case, and other alternative wavelet functions can be chosen. The reconstruction can be made step-by-step, starting from the lowest resolution. At each scale, we have the relations:

$$\hat{c}_{j+1} = \hat{h}(2^j\nu)\hat{c}_j(\nu) \tag{50}$$
$$\hat{w}_{j+1} = \hat{g}(2^j\nu)\hat{c}_j(\nu) \tag{51}$$

We look for c_j knowing c_{j+1}, w_{j+1}, h and g. We restore $\hat{c}_j(\nu)$ based on a least mean square estimator:

$$\begin{aligned} &\hat{p}_h(2^j\nu) \mid \hat{c}_{j+1}(\nu) - \hat{h}(2^j\nu)\hat{c}_j(\nu) \mid^2 + \\ &\hat{p}_g(2^j\nu) \mid \hat{w}_{j+1}(\nu) - \hat{g}(2^j\nu)\hat{c}_j(\nu) \mid^2 \end{aligned} \tag{52}$$

is to be minimum. $\hat{p}_h(\nu)$ and $\hat{p}_g(\nu)$ are weight functions which permit a general solution to the restoration of $\hat{c}_j(\nu)$. From the derivation of $\hat{c}_j(\nu)$ we get:

$$\hat{c}_j(\nu) = \hat{c}_{j+1}(\nu)\hat{\tilde{h}}(2^j\nu) + \hat{w}_{j+1}(\nu)\hat{\tilde{g}}(2^j\nu) \tag{53}$$

where the conjugate filters have the expression:

$$\hat{\tilde{h}}(\nu) = \frac{\hat{p}_h(\nu)\hat{h}^*(\nu)}{\hat{p}_h(\nu)|\hat{h}(\nu)|^2 + \hat{p}_g(\nu)|\hat{g}(\nu)|^2} \tag{54}$$

$$\hat{\tilde{g}}(\nu) = \frac{\hat{p}_g(\nu)\hat{g}^*(\nu)}{\hat{p}_h(\nu)|\hat{h}(\nu)|^2 + \hat{p}_g(\nu)|\hat{g}(\nu)|^2} \tag{55}$$

In this analysis, the Shannon sampling condition is always respected and no aliasing exists. The denominator is reduced if we choose:

$$\hat{g}(\nu) = \sqrt{1 - \mid \hat{h}(\nu) \mid^2}$$

This corresponds to the case where the wavelet is the difference between the square of two resolutions:

$$\mid \hat{\psi}(2\nu) \mid^2 = \mid \hat{\phi}(\nu) \mid^2 - \mid \hat{\phi}(2\nu) \mid^2 \tag{56}$$

The reconstruction algorithm is:

1. Compute the FFT of the image at the low resolution.
2. Set j to n_p. Iterate:
3. Compute the FFT of the wavelet coefficients at scale j.
4. Multiply the wavelet coefficients $\hat{w}_j$ by $\hat{\tilde{g}}$.
5. Multiply the image at the lower resolution $\hat{c}_j$ by $\hat{\tilde{h}}$.
6. The inverse Fourier transform of the addition of $\hat{w}_j\hat{\tilde{g}}$ and $\hat{c}_j\hat{\tilde{h}}$ gives the image c_{j-1}.
7. Set $j = j - 1$ and return to 3.

The use of a scaling function with a cut-off frequency allows a reduction of sampling at each scale, and limits the computing time and the memory size.

Appendix B: The " À Trous" Wavelet Transform Algorithm

In a wavelet transform, a series of transformations of an image is generated, providing a resolution-related set of "views" of the image. The properties satisfied by a wavelet transform, and in particular the *à trous* wavelet transform ("with holes", so called because of the interlaced convolution used in successive levels: see step 2 of the algorithm below) are further discussed in [57].

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

$$c_0(k) =< f(x), \phi(x-k) > \tag{57}$$

The scaling function is chosen to satisfy the dilation equation:

$$\frac{1}{2}\phi(\frac{x}{2}) = \sum_k h(k)\phi(x-k) \tag{58}$$

h is a discrete low pass filter associated with the scaling function ϕ. This means that a low-pass filtering of the image is, by definition, closely linked to another resolution level of the image.

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

$$c_j(k) = \frac{1}{2^j} < f(x), \phi(\frac{x-k}{2^j}) > \tag{59}$$

This is consequently obtained by the convolution:

$$c_j(k) = \sum_l h(l) \;\; c_{j-1}(k + 2^{j-1}l) \tag{60}$$

The signal difference w_j between two consecutive resolutions is:

$$w_j(k) = c_{j-1}(k) - c_j(k) \tag{61}$$

or:

$$w_j(k) = \frac{1}{2^j} < f(x), \psi(\frac{x-k}{2^j}) > \tag{62}$$

Here, the wavelet function ψ is defined by:

$$\frac{1}{2}\psi(\frac{x}{2}) = \phi(x) - \frac{1}{2}\phi(\frac{x}{2}) \tag{63}$$

For the scaling function, $\phi(x)$, the B-spline of degree 3 was used in our calculations. 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 we carry out a discrete convolution of the data $c_{j-1}(k)$ using the filter h. The distance between the central pixel 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 go to step 2.

5. The set $\mathcal{W} = \{w_1, ..., w_p, c_p\}$ represents the wavelet transform of the data.

The most general way to handle the boundaries is to consider that $c(k+N) = c(N-k)$. But other methods can be used such as periodicity ($c(k+N) = c(k)$), or continuity ($c(k+N) = c(N)$). Choosing one of these methods has little influence on our general restoration strategy. We used continuity.

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:

$$c_0(k) = c_p(k) + \sum_{j=1}^{p} w_j(k) \tag{64}$$

This equation provides a reconstruction formula for the original signal.

At each scale j, we obtain a set $\{w_j\}$. This has the same number of pixels as the input signal.

The above *à trous* algorithm has been discussed in terms of a single index, x, but is easily extendable to two-dimensional space. The use of the B_3 spline leads to a convolution with a mask of 5×5:

$$\begin{pmatrix} \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{3}{128} & \frac{3}{32} & \frac{9}{64} & \frac{3}{32} & \frac{3}{128} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \end{pmatrix}$$

In one dimension, this mask is: $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$.

To facilitate computation, a simplification of this wavelet is to assume separability in the 2-dimensional case. In the case of the B_3 spline, this leads to a row by row convolution with $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$; followed by column by column convolution.

As for the one dimensional case, an exact reconstruction is obtained by adding all the scales and the final smoothed array:

$$c_0(x,y) = c_p(x,y) + \sum_{j=1}^{p} w_j(x,y) \tag{65}$$