Image Enhancement and Image Restoration by Multiscale Methods

Jean-Luc Starck

Service d'Astrophysique, CEA-Saclay, France.
jstarck@cea.fr, http://jstarck.free.fr

Collaborators:

D.L. Donoho, *Department of Statistics, Stanford*
E. Candès, *Department of Applied Mathematics, California Institute of Technology*
A. Bijaoui, *Nice Observatory*
F. Murtagh, *Queen's University, Belfast*

I Multiscale Transforms

II Image Restoration

1 - Image Enhancement - Dynamic Range Compression
2 - Denoising

III Combination of Several Multiscale Transforms

IV Astronomical Data Analysis

Multiscale Transforms

Critical Sampling	Redundant Transforms
	Pyramidal decomposition (Burt and Adelson)
(bi-) Orthogonal WT	**Undecimated Wavelet Transform**
Lifting scheme construction	**Isotropic Undecimated Wavelet Transform**
Wavelet Packets	Complex Wavelet Transform
Mirror Basis	Steerable Wavelet Transform
	Dyadic Wavelet Transform
	Nonlinear Pyramidal decomposition (Median)

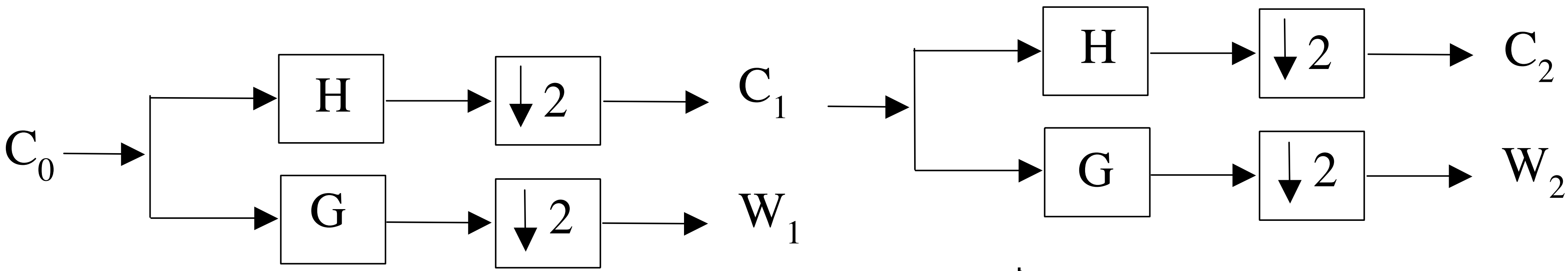
New Multiscale Construction

Contourlet	**Ridgelet**
Bandelet	**Curvelet** (Several implementations)
Finite Ridgelet Transform	
Platelet	
(W-)Edgelet	
Adaptive Wavelet	

The Orthogonal Wavelet Transform (OWT)

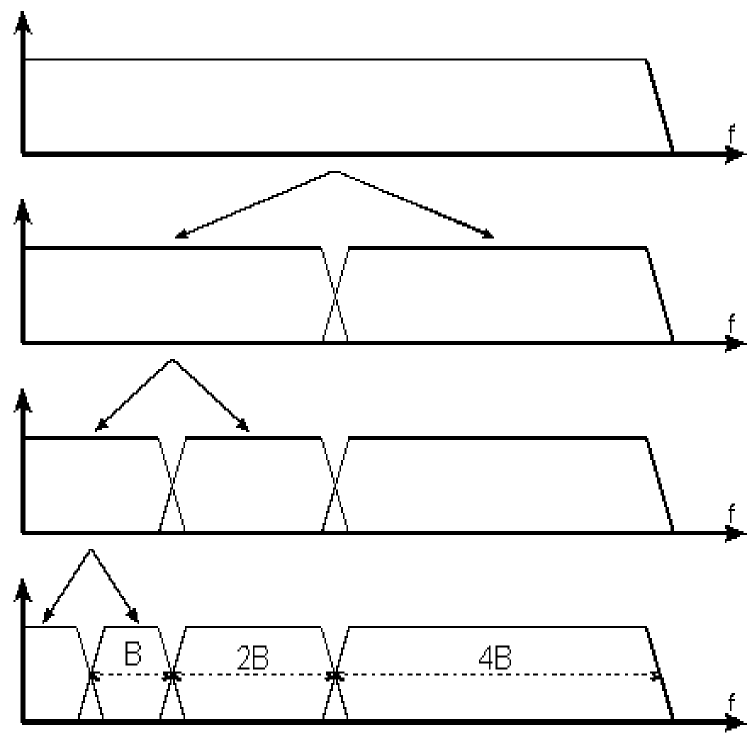
$$s_l = \sum_k c_{J,k} \phi_{J,l}(k) + \sum_k \sum_{j=1}^{J} \psi_{j,l}(k) w_{j,k}$$

Transformation



$$c_{j+1,l} = \sum_h h_{k-2l} c_{j,k} = (\bar{h} * c_j)_{2l}$$

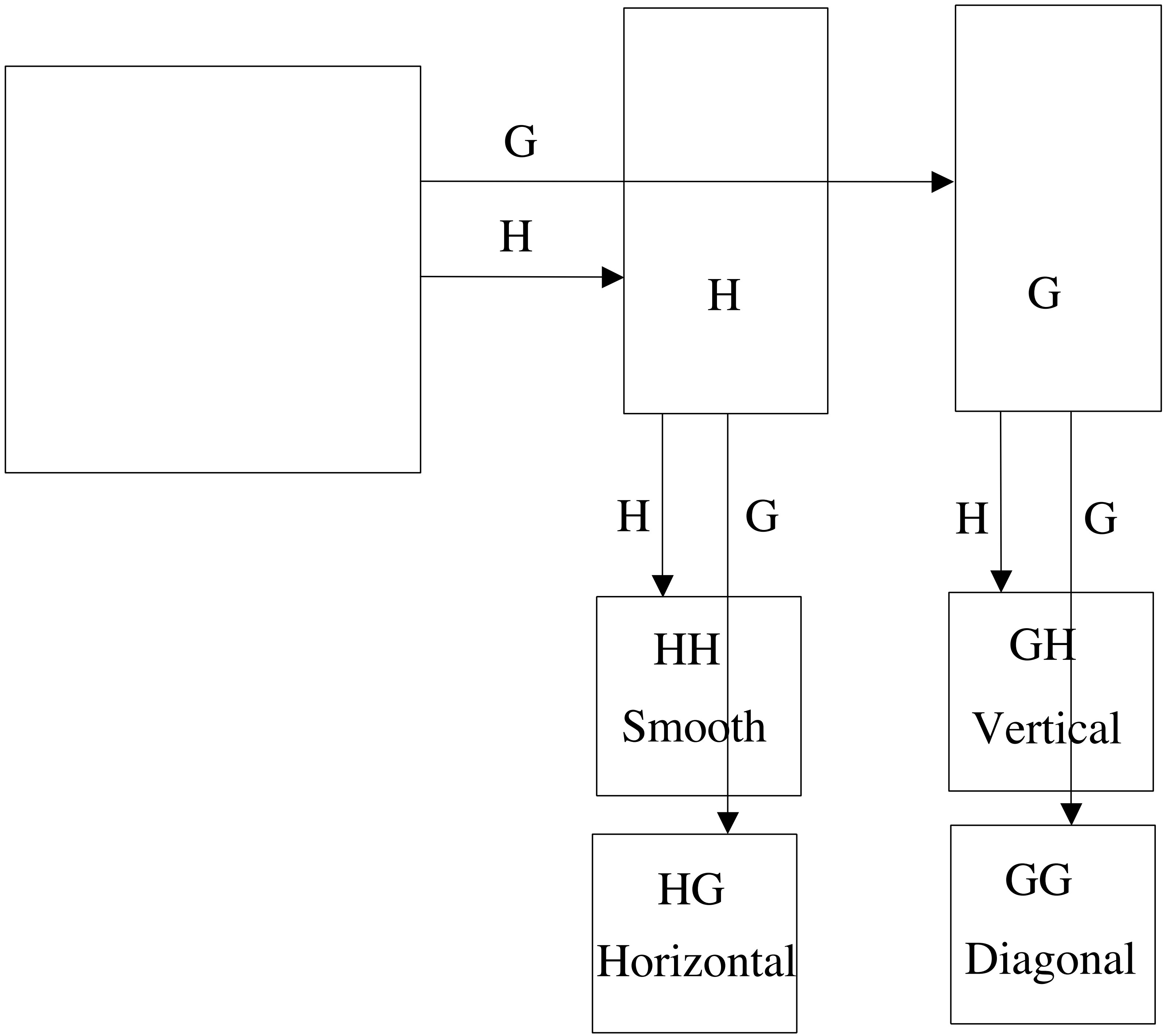
$$w_{j+1,l} = \sum_h g_{k-2l} c_{j,k} = (\bar{g} * c_j)_{2l}$$



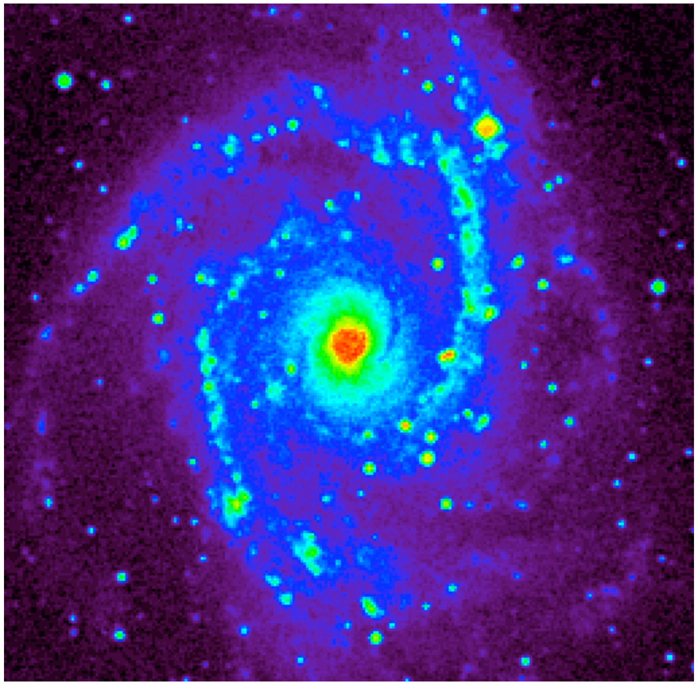
Reconstruction:

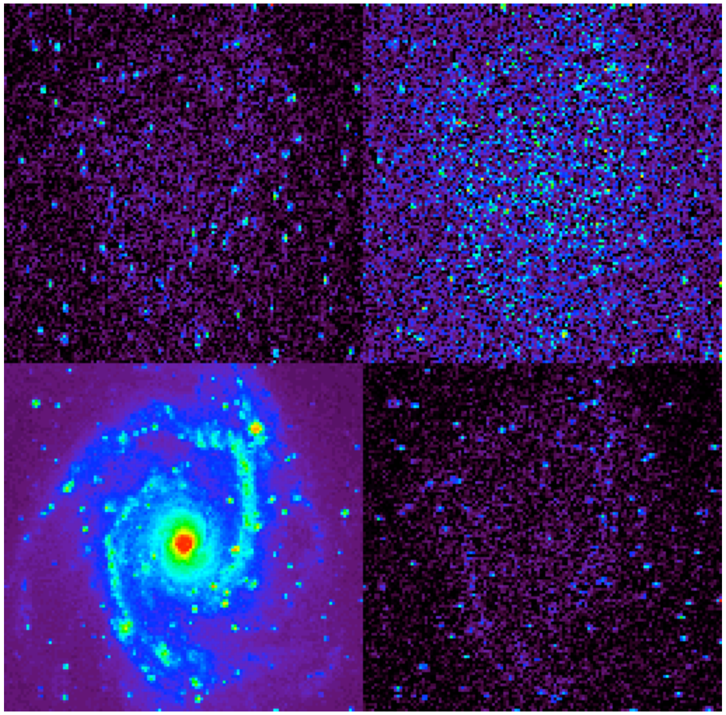
$$c_{j,l} = \sum_k \tilde{h}_{k+2l} c_{j+1,k} + \tilde{g}_{k+2l} w_{j+1,k} = \tilde{h} * \breve{c}_{j+1} + \tilde{g} * \breve{w}_{j+1}$$

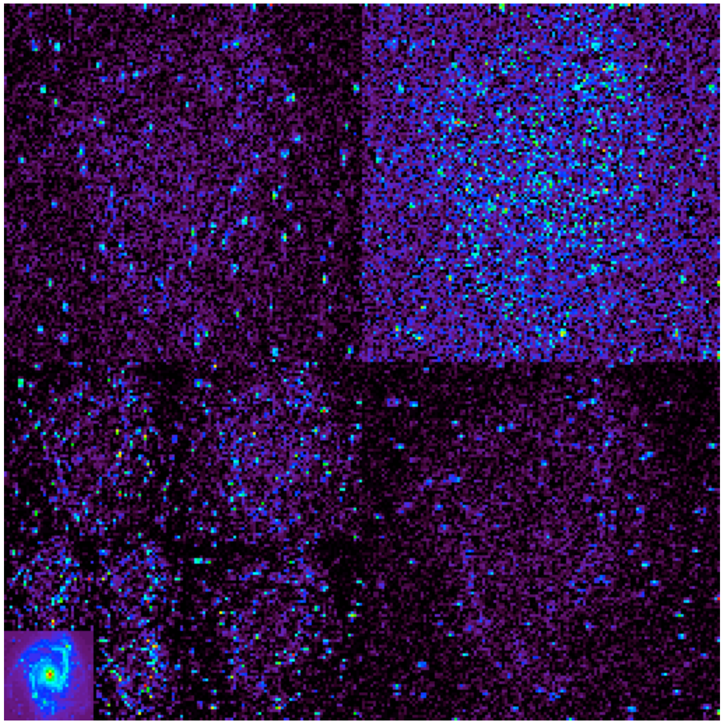
$$\breve{x} = (x_1, 0, x_2, 0, x_3, \ldots, 0, x_j, 0, \ldots, x_{n-1}, 0, x_n)$$



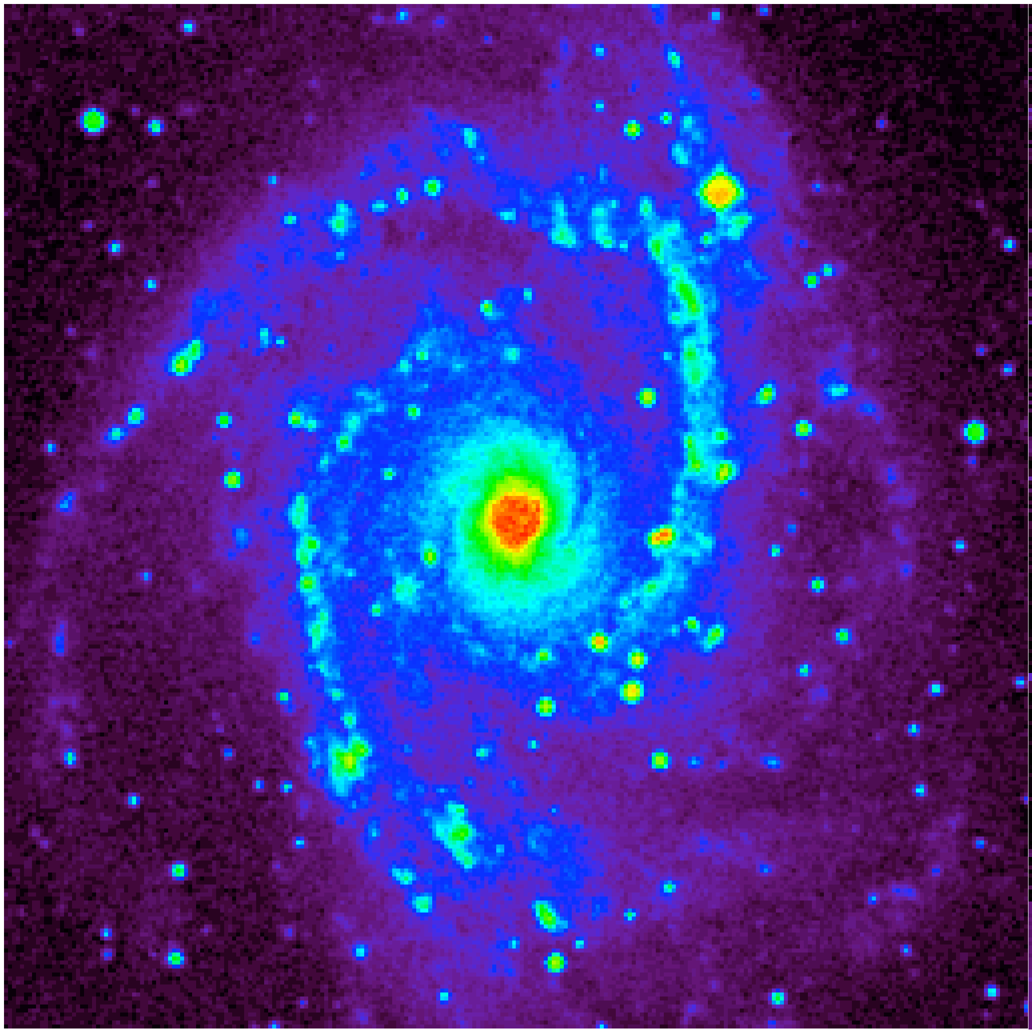
G
H
H
G
H
G
H
G
HH
Smooth
GH
Vertical
HG
Horizontal
GG
Diagonal



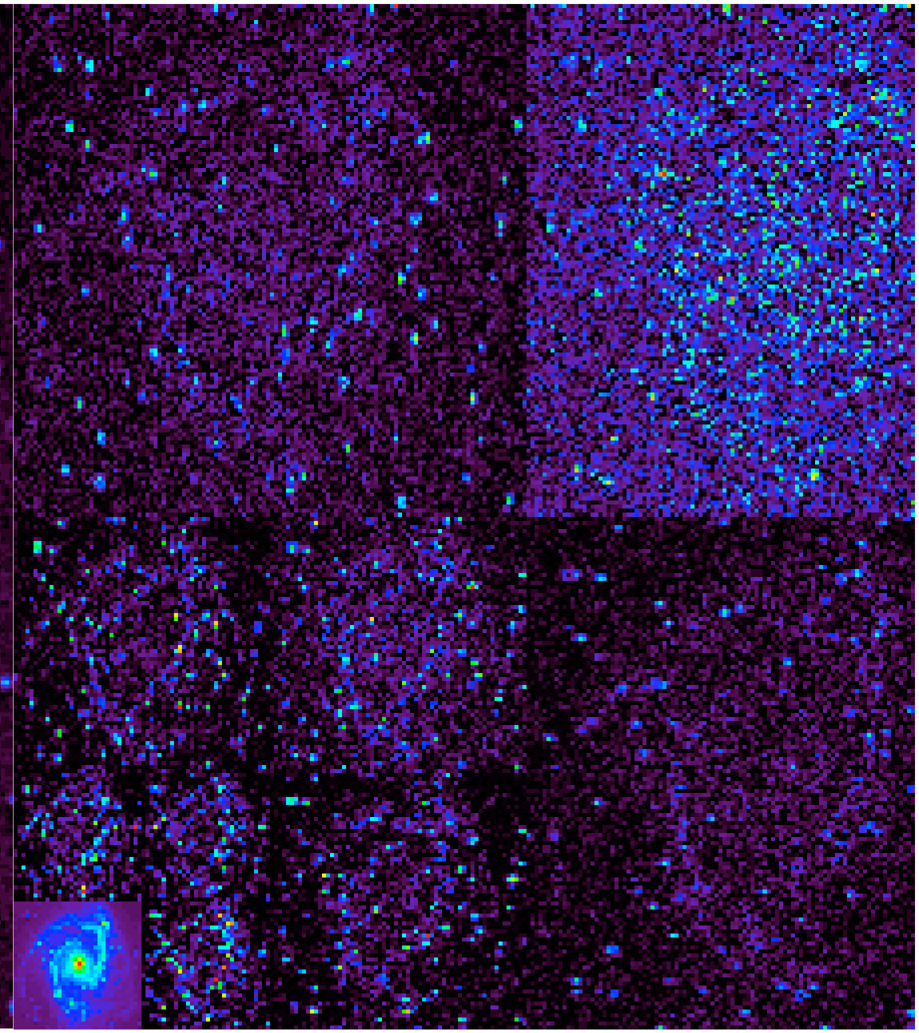




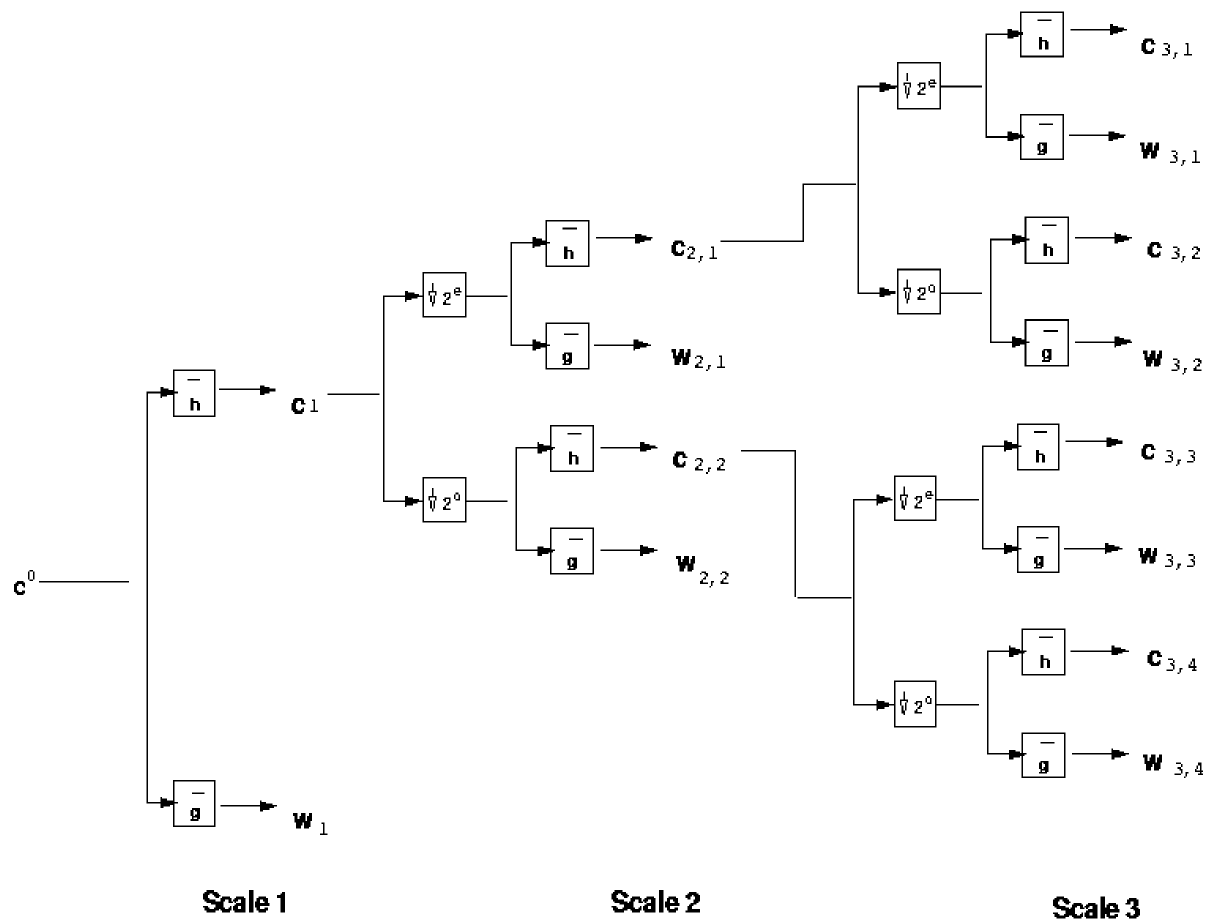
NGC2997



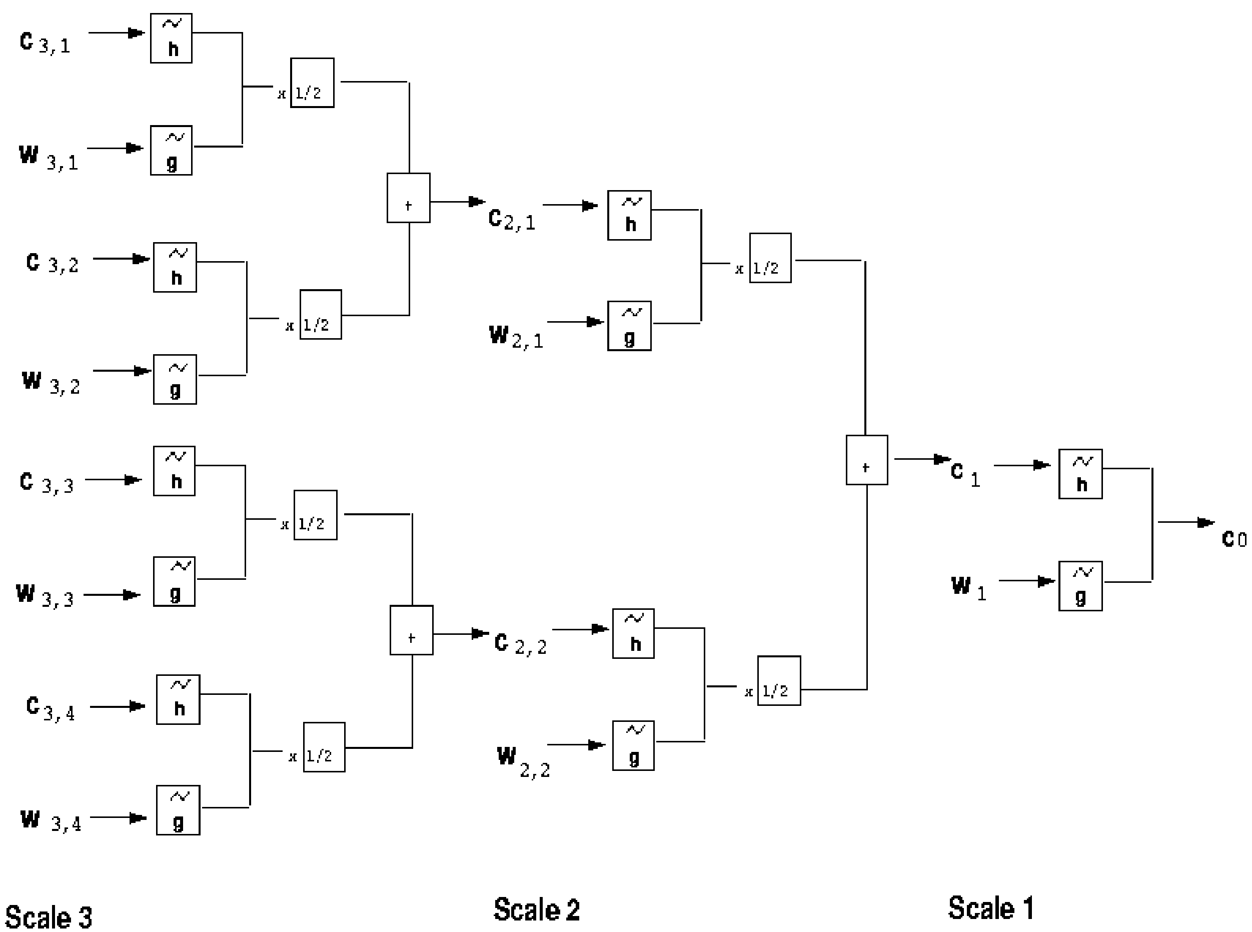
NGC2997 WT



1D undecimated wavelet transform



1D undecimated wavelet reconstruction



The à trous Algorithm

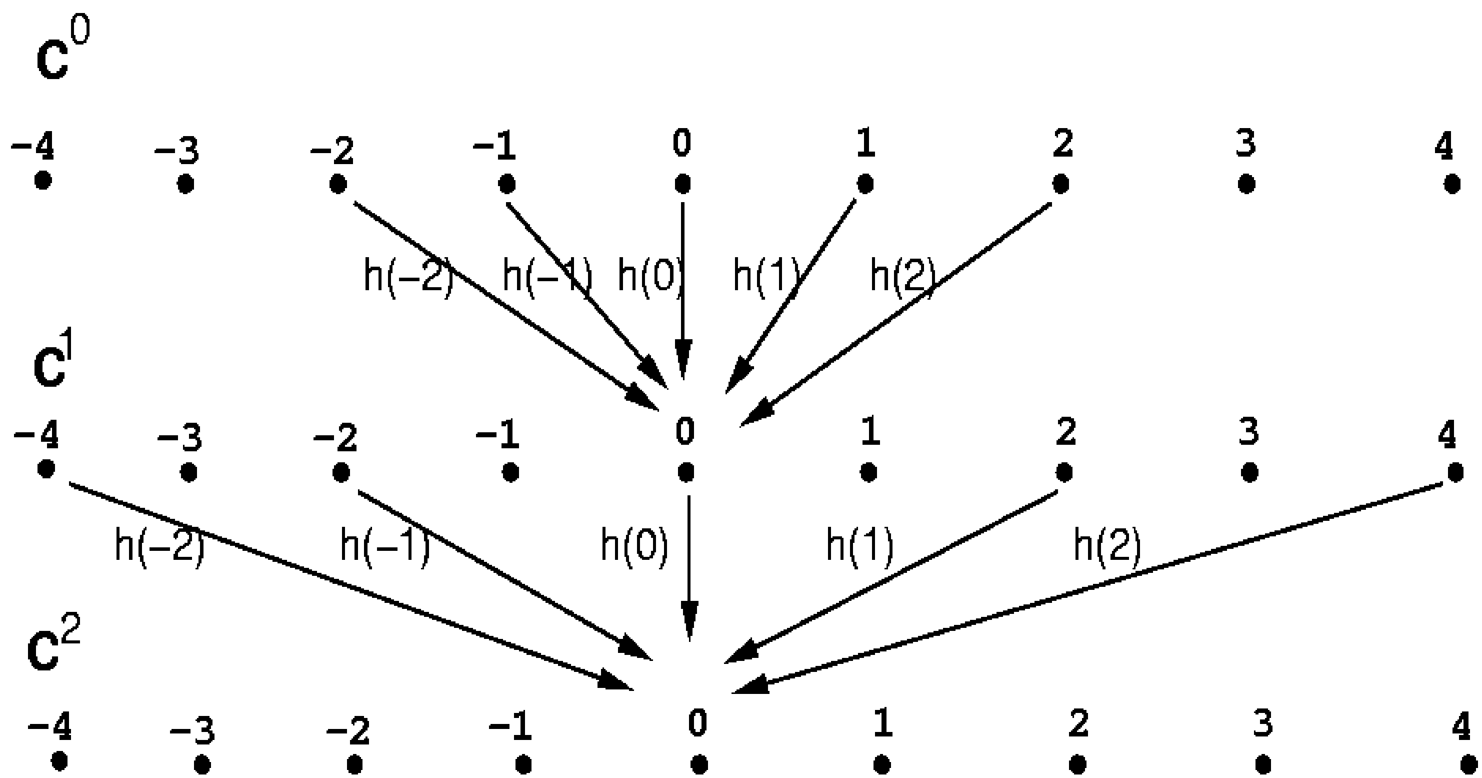
It exists however a very efficient way to implement it. The "à trous" algorithm consists in considering the filter $h^{(j)}$ instead of h where $h^{(j)}_l = h_l$ if $l/2^j$ is an integer and 0 otherwise. For example, we have $h^{(1)} = (\ldots, h_{-2}, 0, h_{-1}, 0, h_0, 0, h_1, 0, h_2, \ldots)$. Then $c_{j+1,l}$ and $w_{j+1,l}$ can be expressed by:

$$
\begin{aligned}
c_{j+1,l} &= (\bar{h}^{(j)} * c_j)_l = \sum_k h_k c_{j,l+2^j k} \\
w_{j+1,l} &= (\bar{g}^{(j)} * c_j)_l = \sum_k g_k c_{j,l+2^j k}
\end{aligned}
$$

The reconstruction is obtained by:

$$
c_j = \frac{1}{2}(\tilde{h}^{(j)} * c_{j+1} + \tilde{g}^{(j)} w_{j+1})
$$

Passage from c_0 to c_1, and from c_1 to c_2

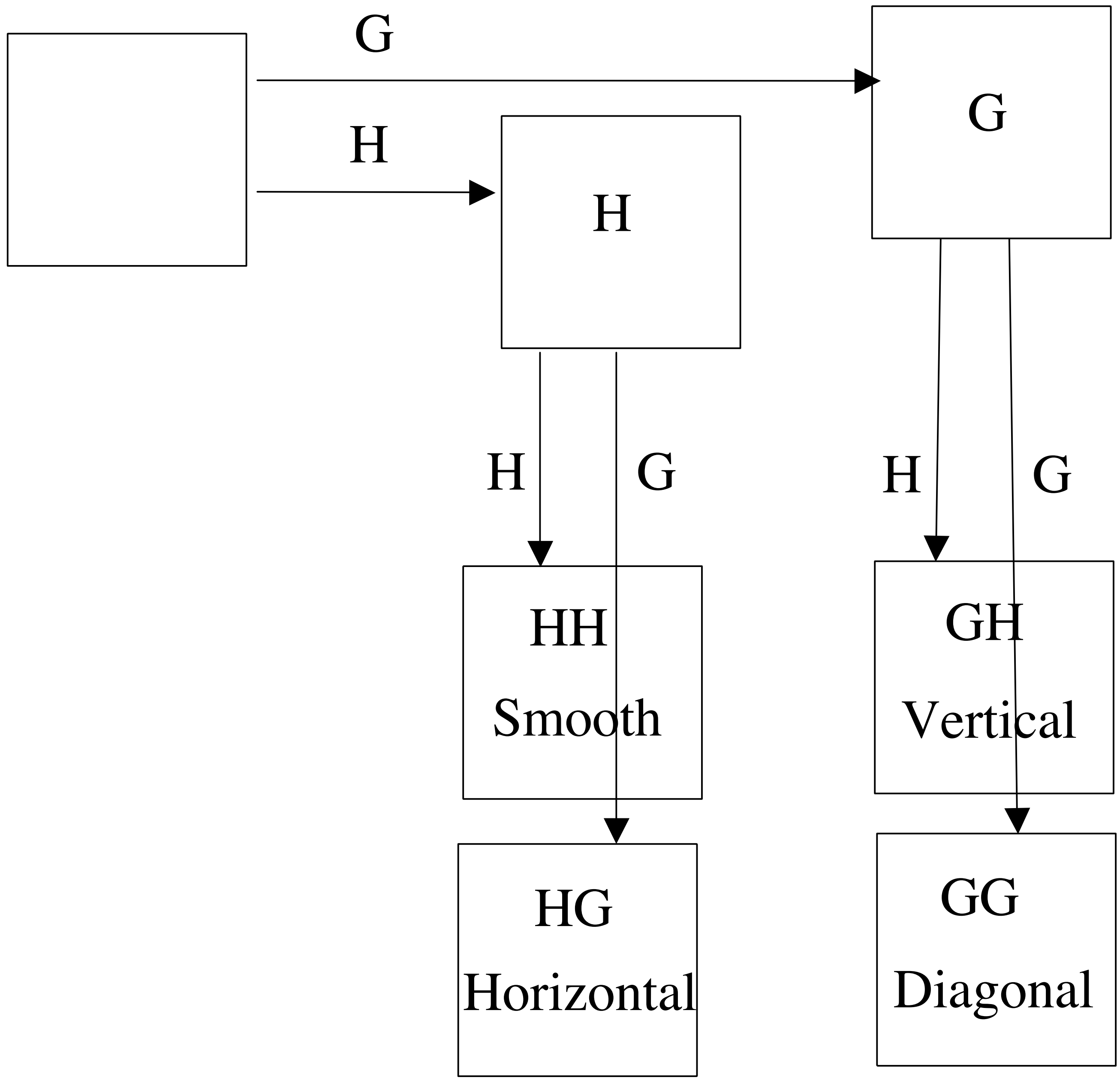


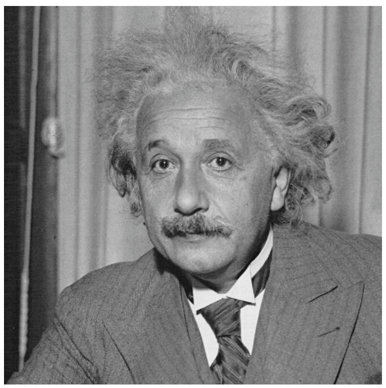
2D Undecimated Wavelet Transform

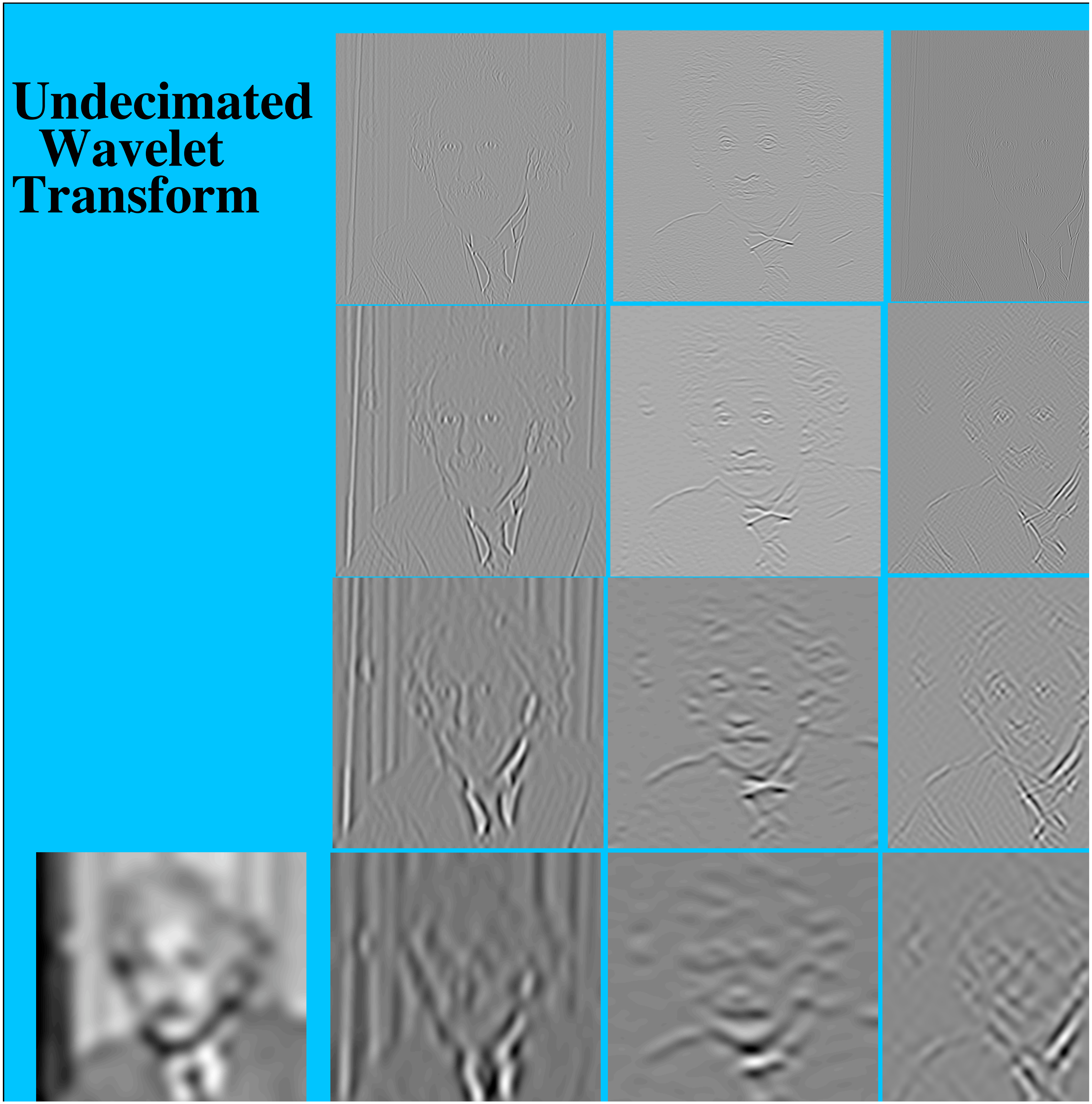
The à trous algorithm can be extended to 2D:

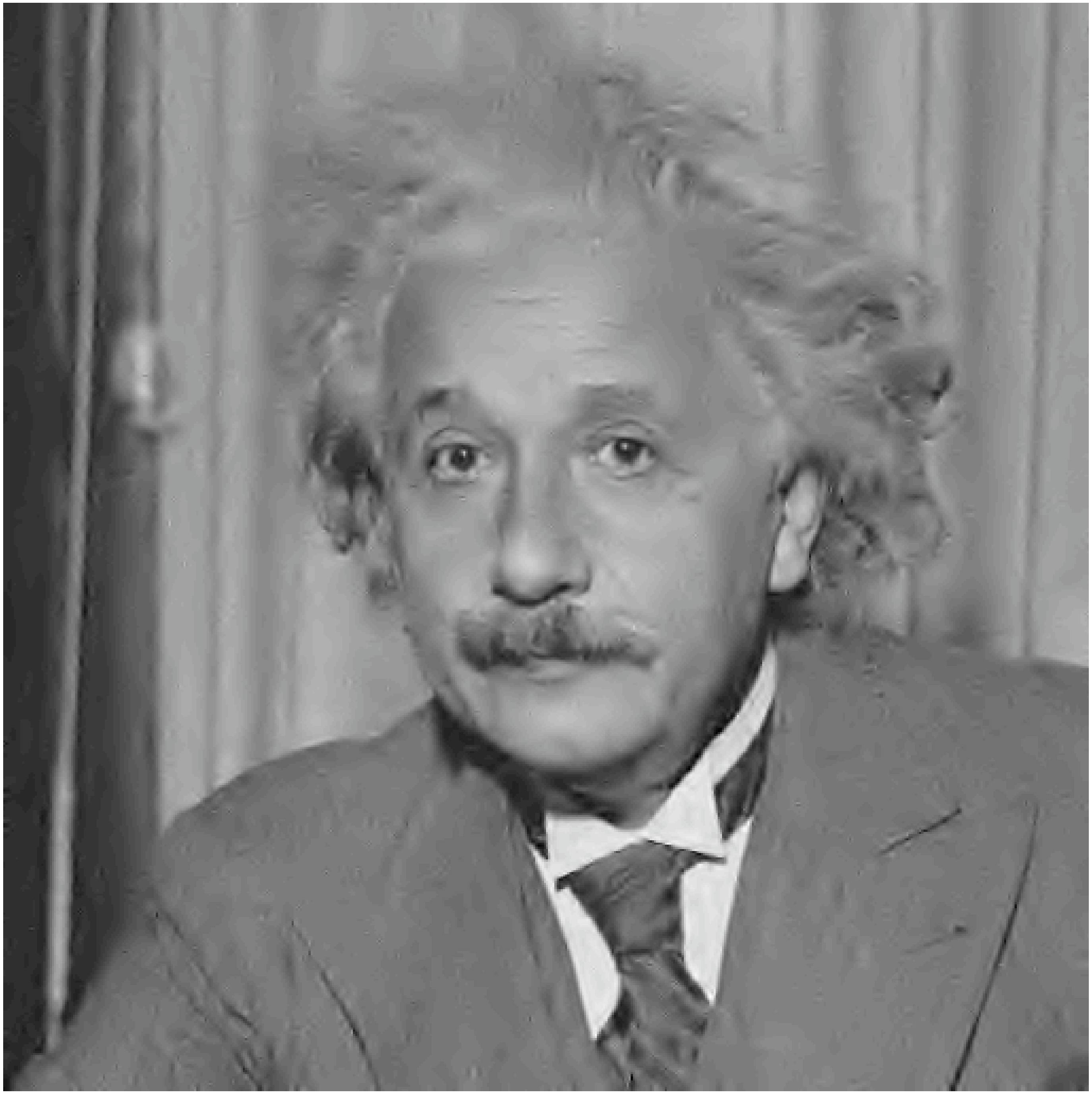
$$
\begin{aligned}
c_{j+1,k,l} &= (\bar{h}^{(j)}\bar{h}^{(j)} * c_j)_{k,l} \\
w_{j+1,1,k,l} &= (\bar{g}^{(j)}\bar{h}^{(j)} * c_j)_{k,l} \\
w_{j+1,2,k,l} &= (\bar{h}^{(j)}\bar{g}^{(j)} * c_j)_{k,l} \\
w_{j+1,3,k,l} &= (\bar{g}^{(j)}\bar{g}^{(j)} * c_j)_{k,l}
\end{aligned}
$$

where $hg * c$ is the convolution of c by the separable filter hg (i.e convolution first along the columns per h and then convolution along the lines per g).


G
H
G
H
H
G
H
G
HH
Smooth
GH
Vertical
HG
Horizontal
GG
Diagonal




Undecimated
Wavelet
Transform

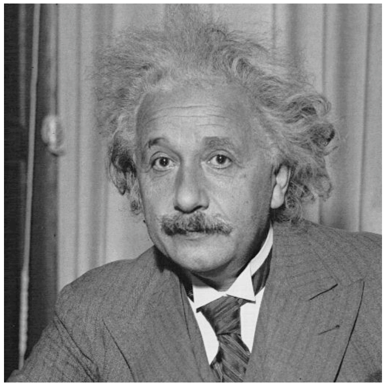


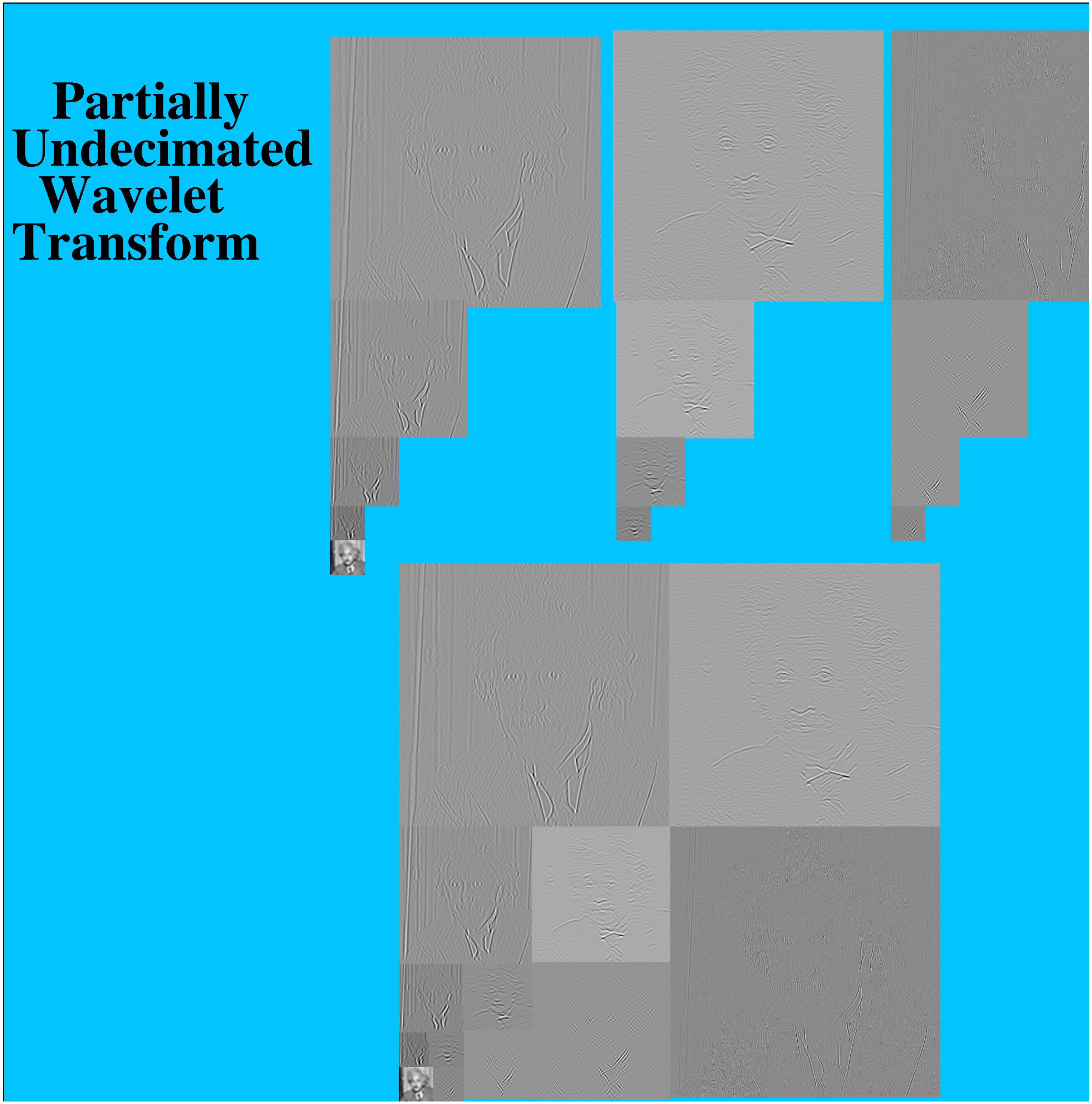
Partially Decimated Wavelet Transform

The Undecimated wavelet transform (UWT) is highly redundant. A compromise can be found by not decimating only one or two scales. We note $PWT^{(u)}$ the wavelet transform where the first u scales are undecimated. $PWT^{(0)}$ corresponds to the bi-orthogonal WT and $PWT^{(J)}$ corresponds to the UWT. For a $PWT^{(1)}$ the redundancy factor is 4. For the passage from a resolution j to the next one, it will the same operations as for the UWT when $j \leq u$. Noting $j' = MIN(j, u)$, we have:

$$
\begin{aligned}
c_{j+1,k,l} &= (\bar{h}^{(j')}\bar{h}^{(j')} * c_j)_{k,l} \\
w_{j+1,1,k,l} &= (\bar{g}^{(j')}\bar{h}^{(j')} * c_j)_{k,l} \\
w_{j+1,2,k,l} &= (\bar{h}^{(j')}\bar{g}^{(j')} * c_j)_{k,l} \\
w_{j+1,3,k,l} &= (\bar{g}^{(j')}\bar{g}^{(j')} * c_j)_{k,l}
\end{aligned}
$$

After the uth scale, the number of holes in the filters $\bar{h}$ and $\bar{g}$ remains unchanged.

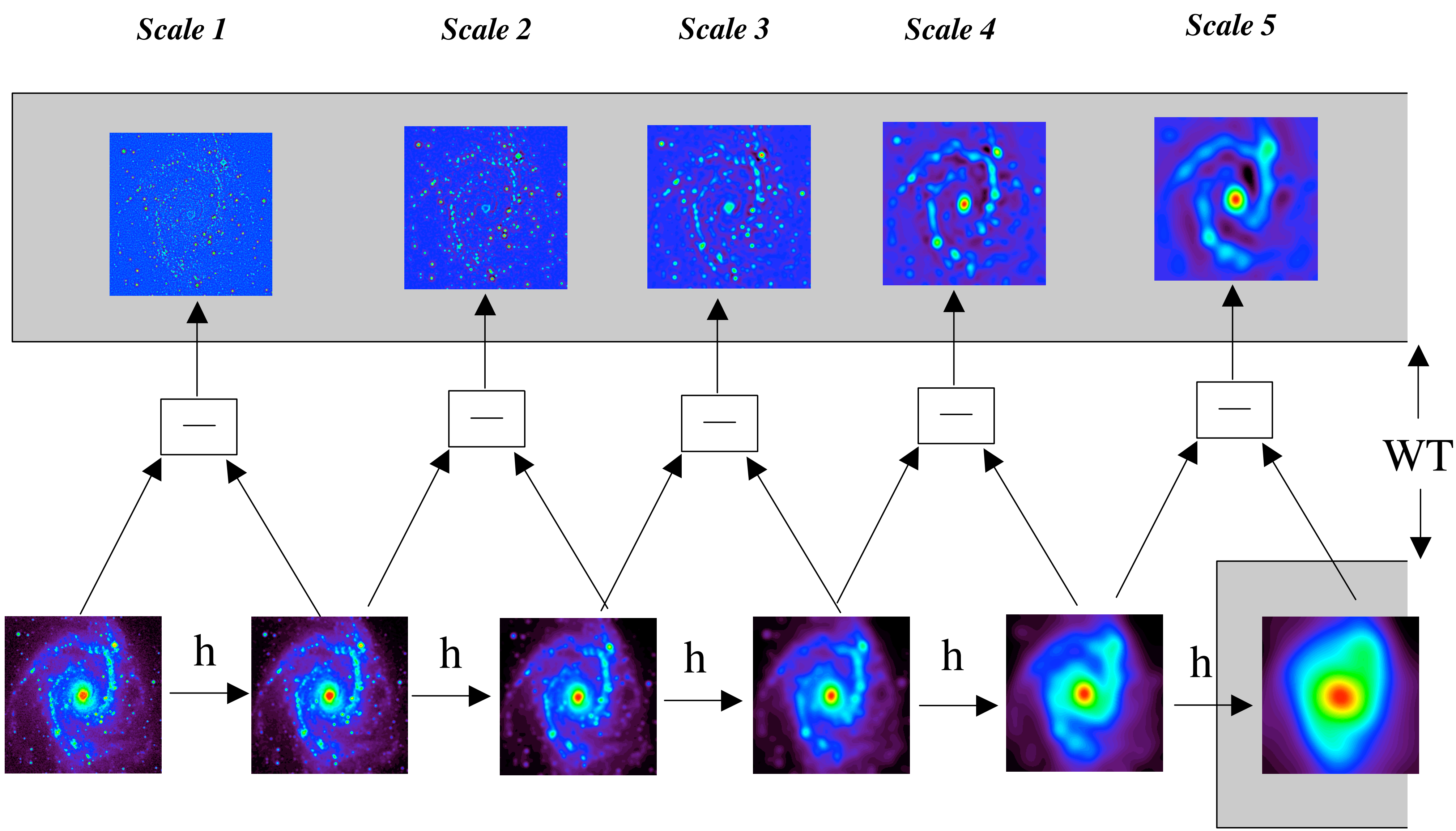



Partially
Undecimated
Wavelet
Transform

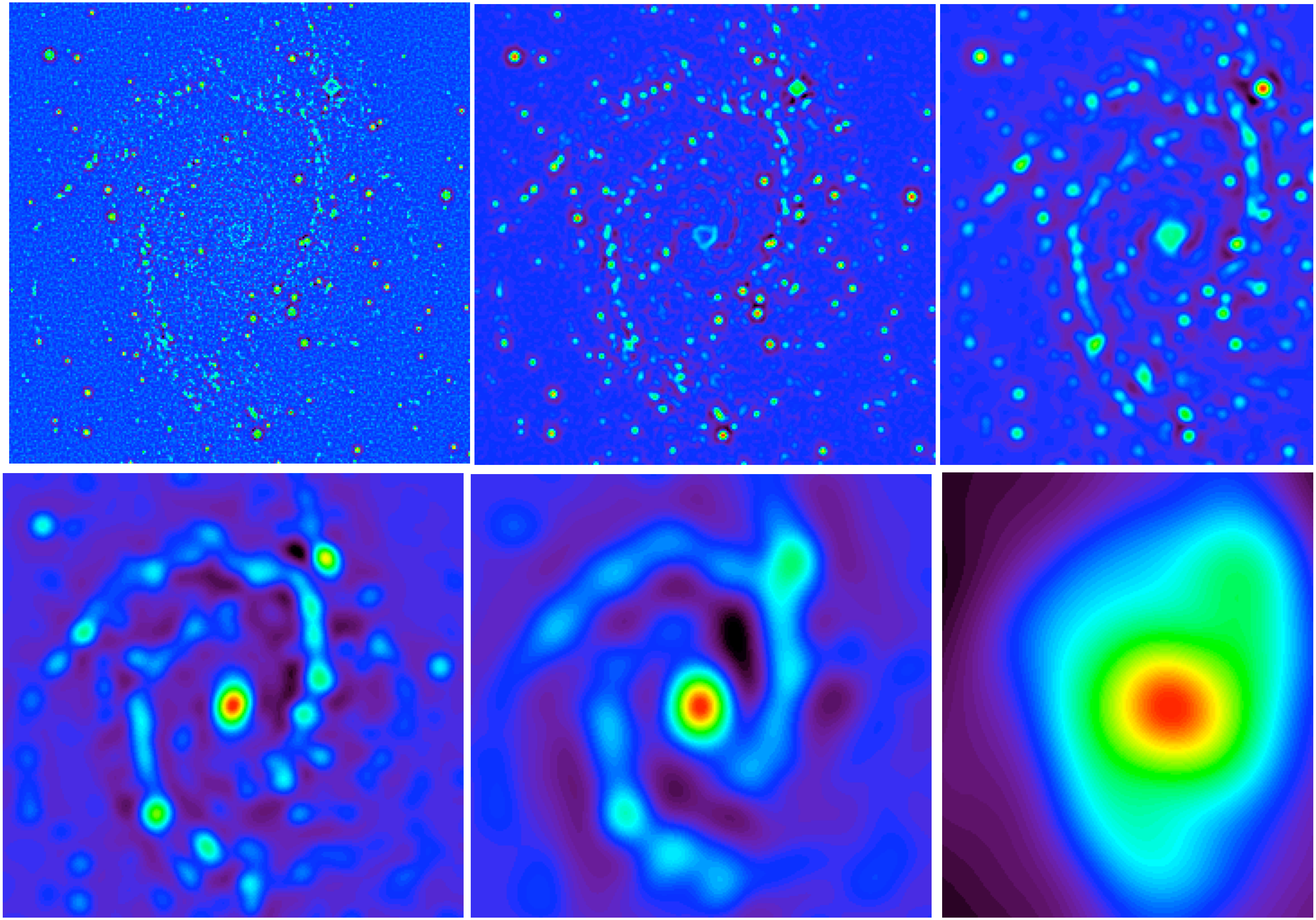
OWT UWT

Redundancy	1	4	7	10	13
PSNR	29.34	30.36	31.35	31.67	31.77

ISOTROPIC UNDECIMATED WAVELET TRANSFORM



Isotropic Undec. WT: $I(k,l) = c_{J,k,l} + \sum_{j=1}^{J} w_{j,k,l}$



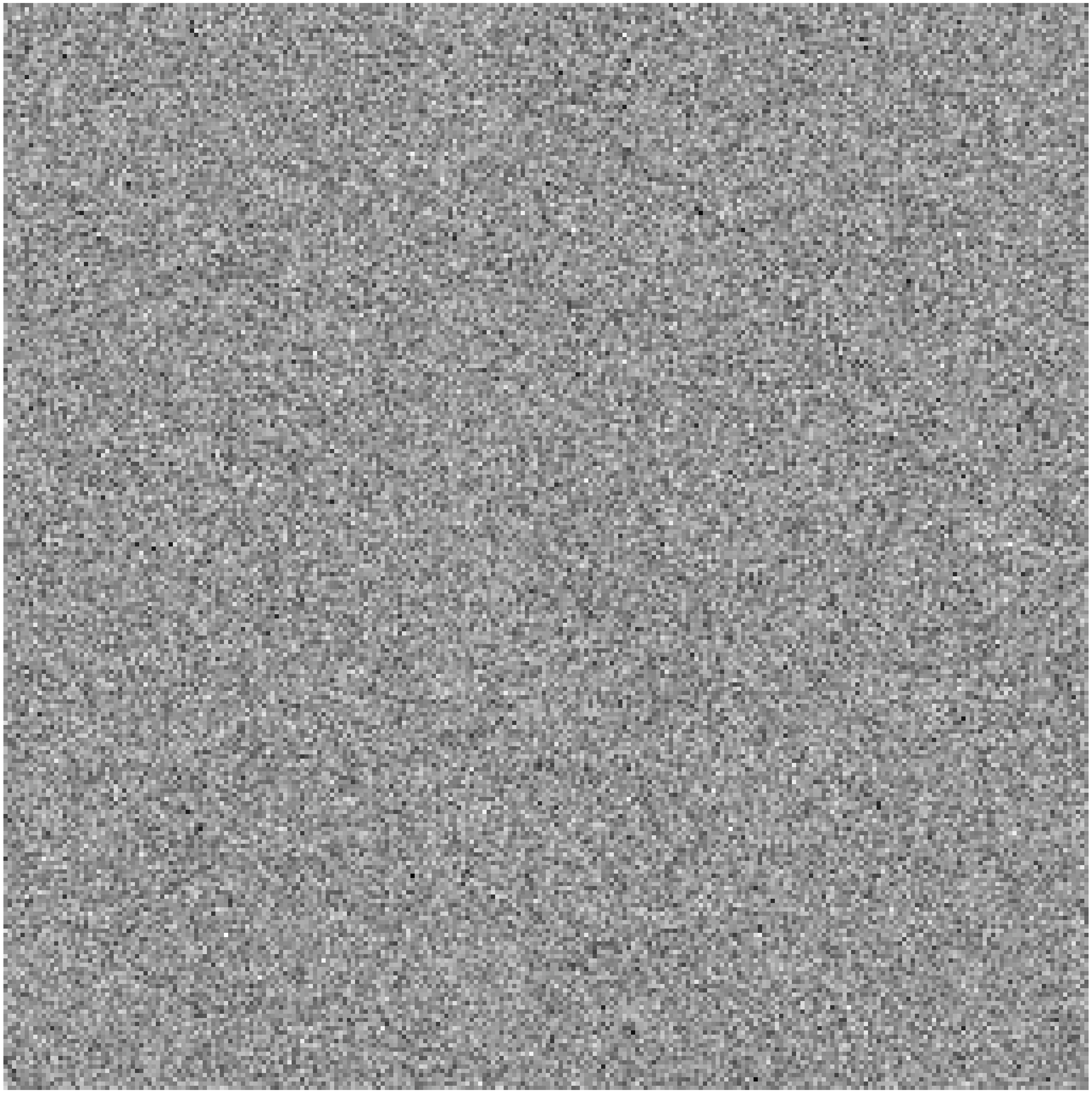
Problems related to the WT

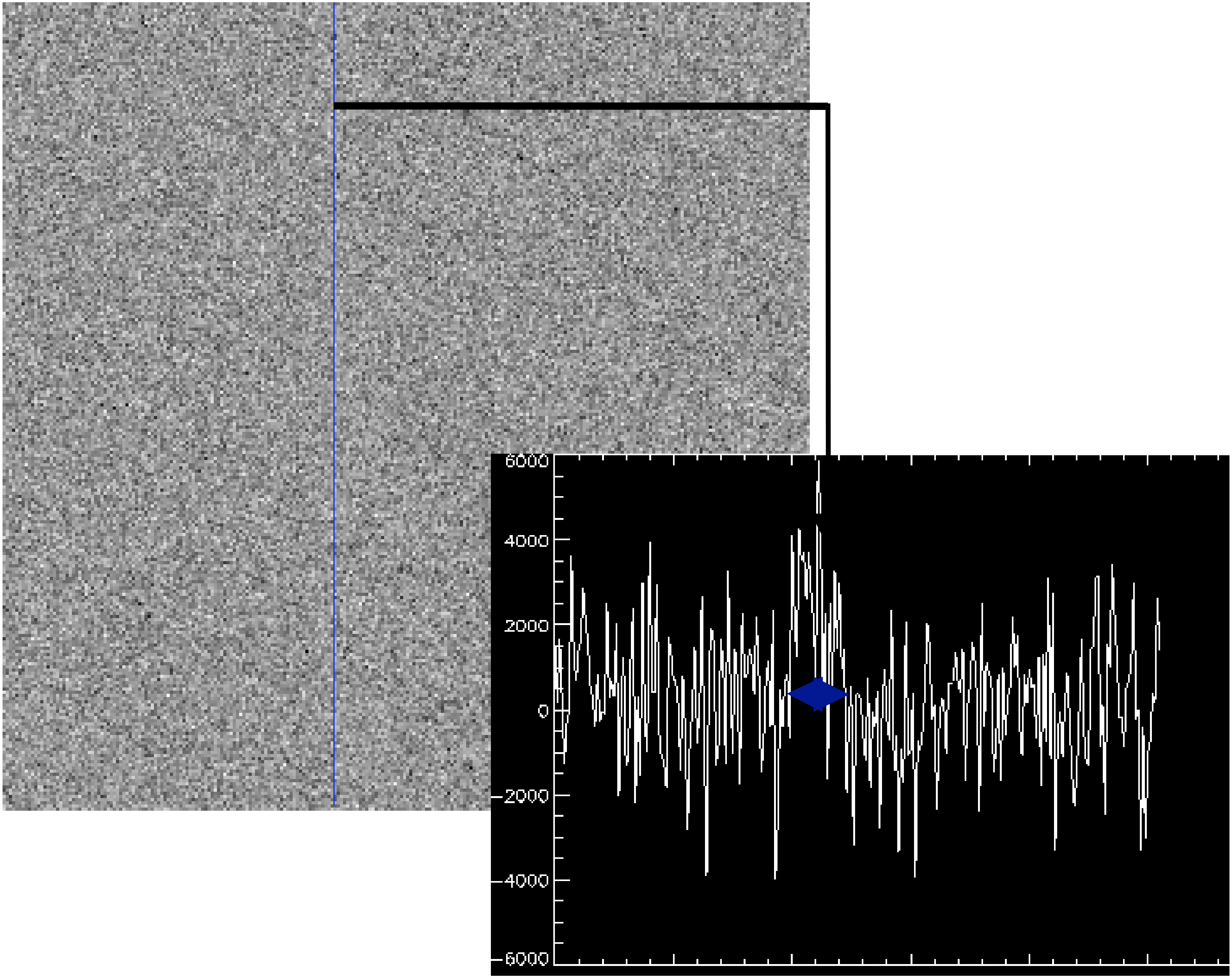
1) Edges representation:
if the WT performs better than the FFT to
represent edges in an image, it is still not optimal.

2) There is only a fixed number of directional elements
independent of scales.

3) Limitation of existing scale concepts:
there is no highly anisotropic elements.

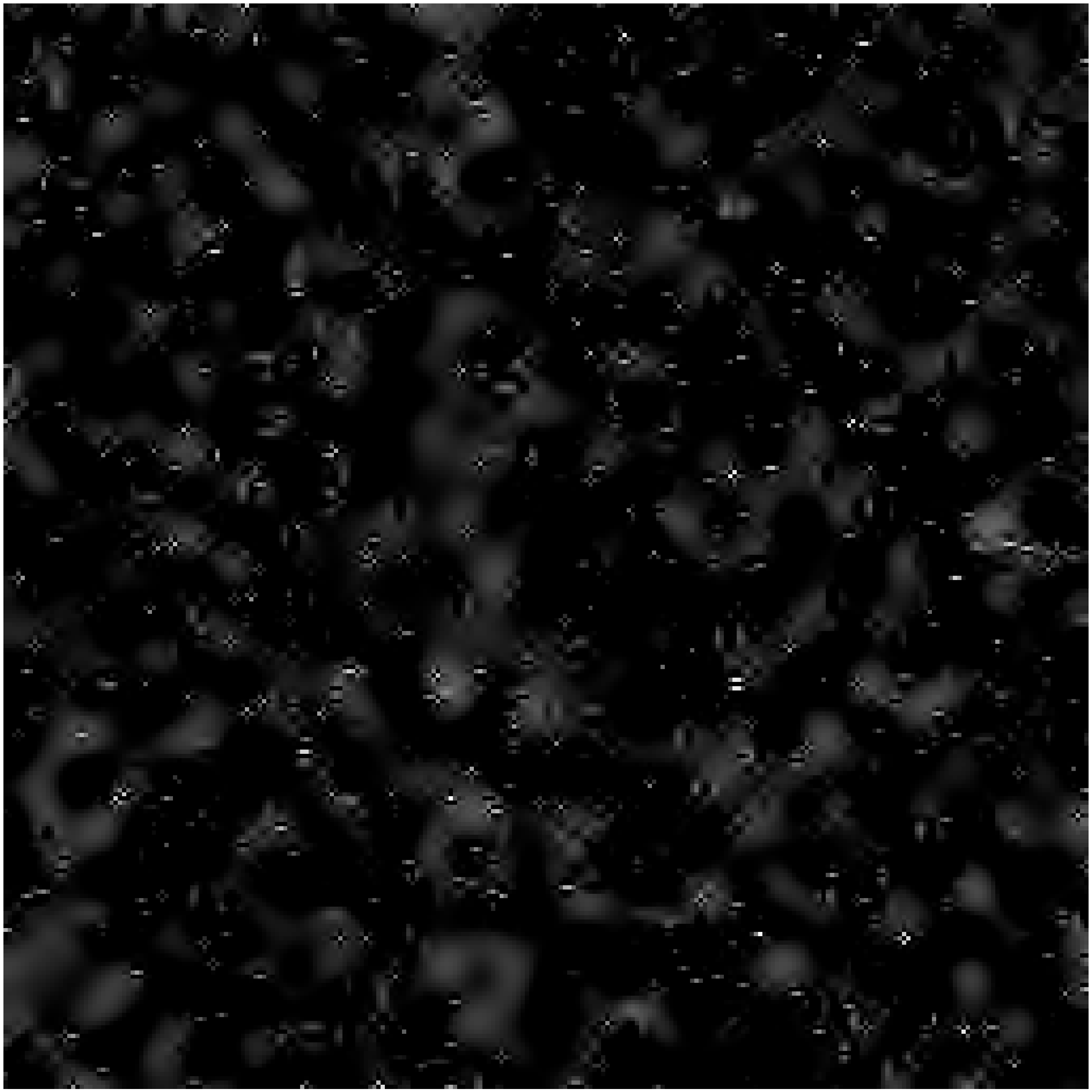
SNR = 0.1



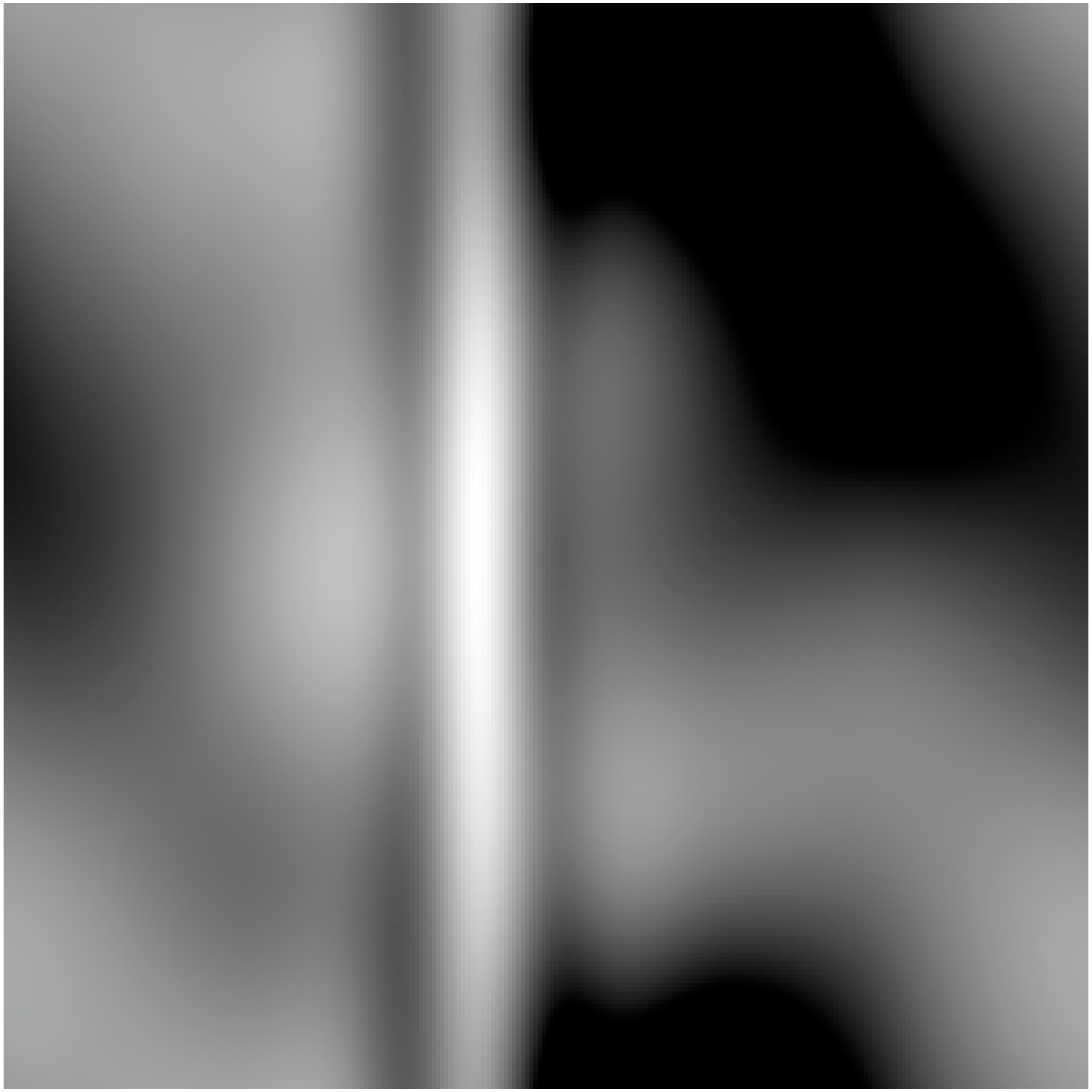


6000
4000
2000
0
-2000
-4000
-6000

Undecimated Wavelet Filtering (3 sigma)

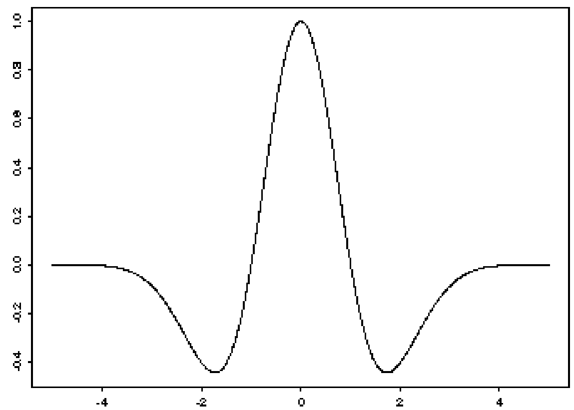


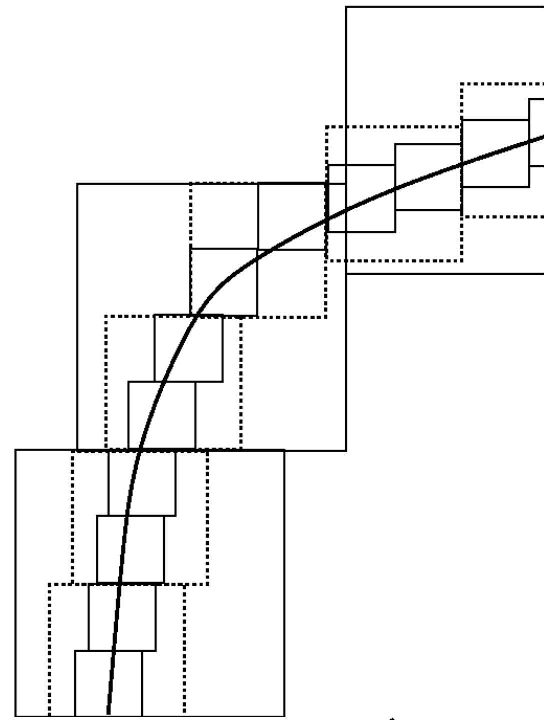
Ridgelet Filtering (5sigma)



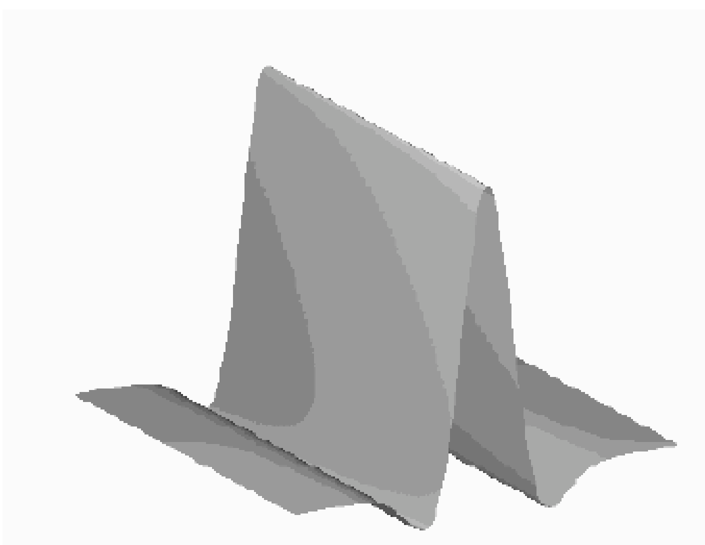
The Curvelet Transform

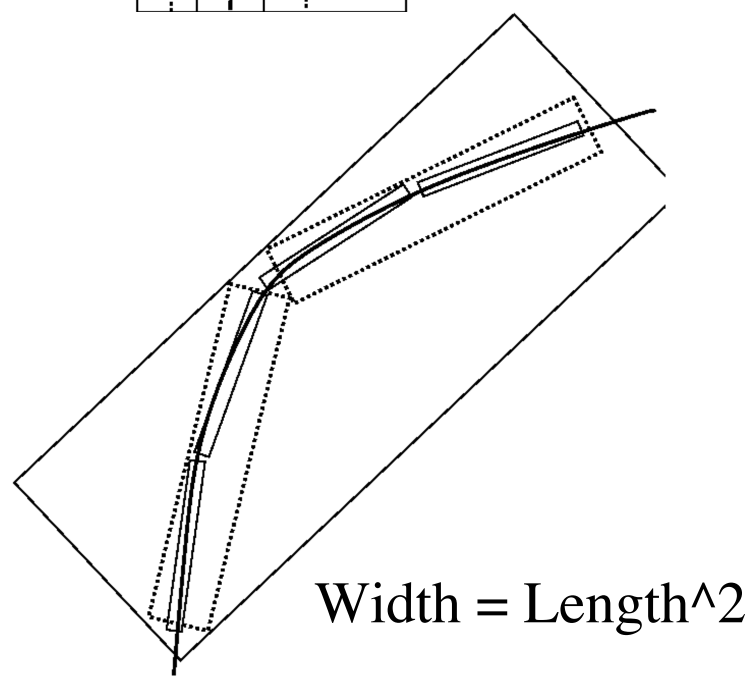
Wavelet

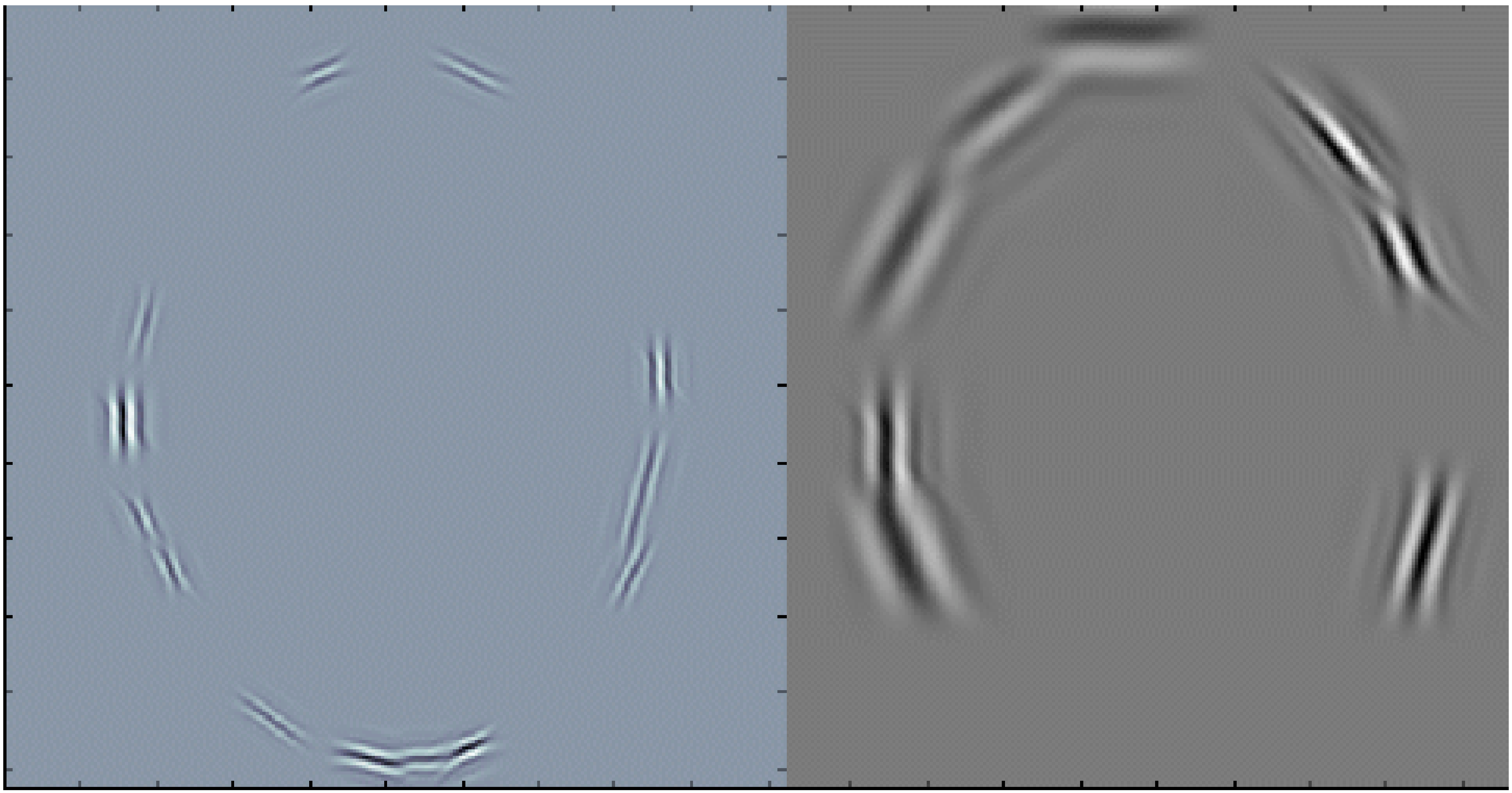




Curvelet







The Curvelet Transform for Image Denoising, *IEEE Transaction on Image Processing, 11, 6, 2002,*

- 2D Wavelet Tranforfm

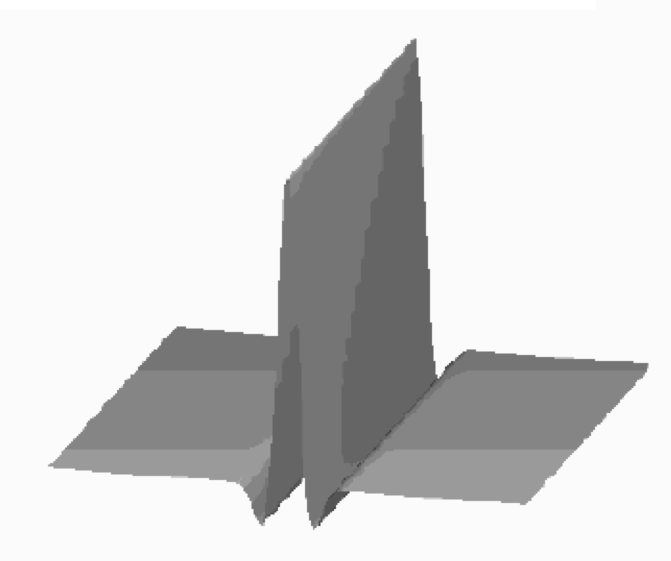
- Local Ridgelet Transform

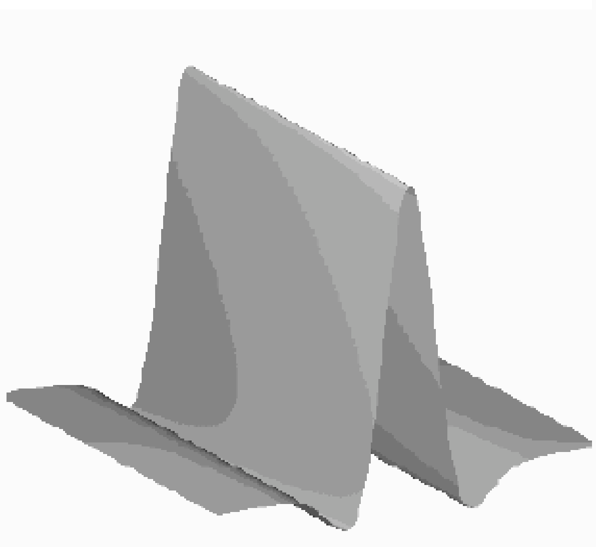
Continuous Ridgelet Transform

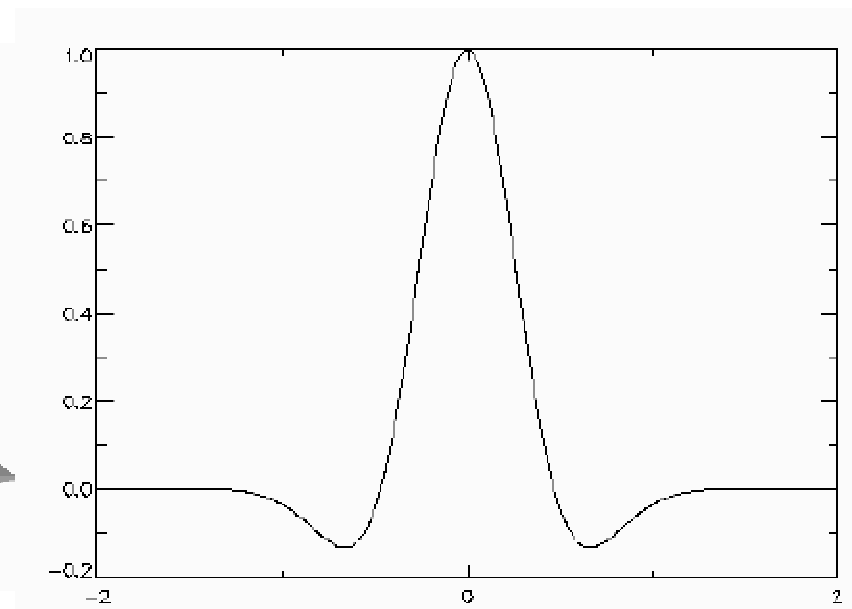
Ridgelet Transform (Candes, 1998): $R_f(a,b,\theta) = \int \psi_{a,b,\theta}(x) f(x)dx$

Ridgelet function: $\psi_{a,b,\theta}(x) = a^{\frac{1}{2}}\psi\left(\frac{x_1\cos(\theta) + x_2\sin(\theta) - b}{a}\right)$

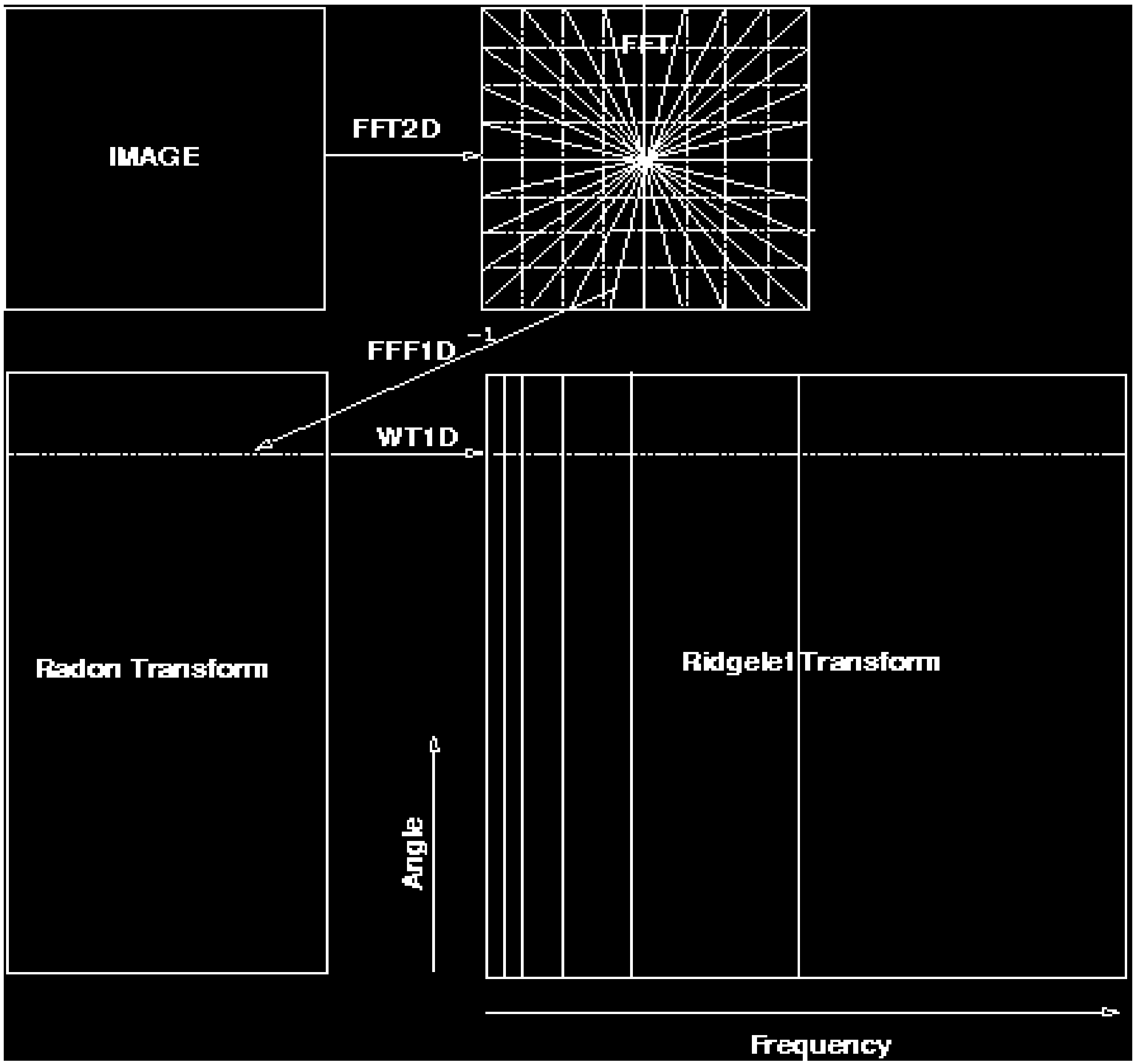
The function is constant along lines. Transverse to these ridges, it is a wavelet.





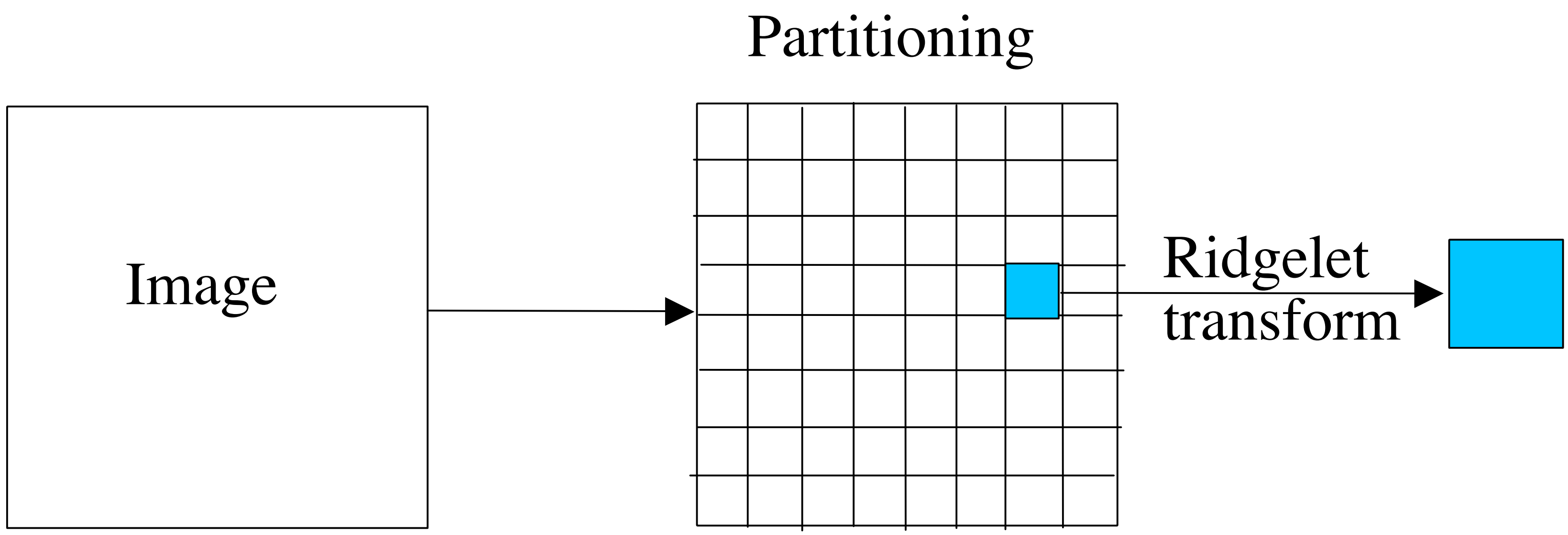


The ridgelet coefficients of an object f are given by analysis of the Radon transform via: $R_f(a,b,\theta) = \int Rf(\theta,t)\psi(\frac{t-b}{a})dt$

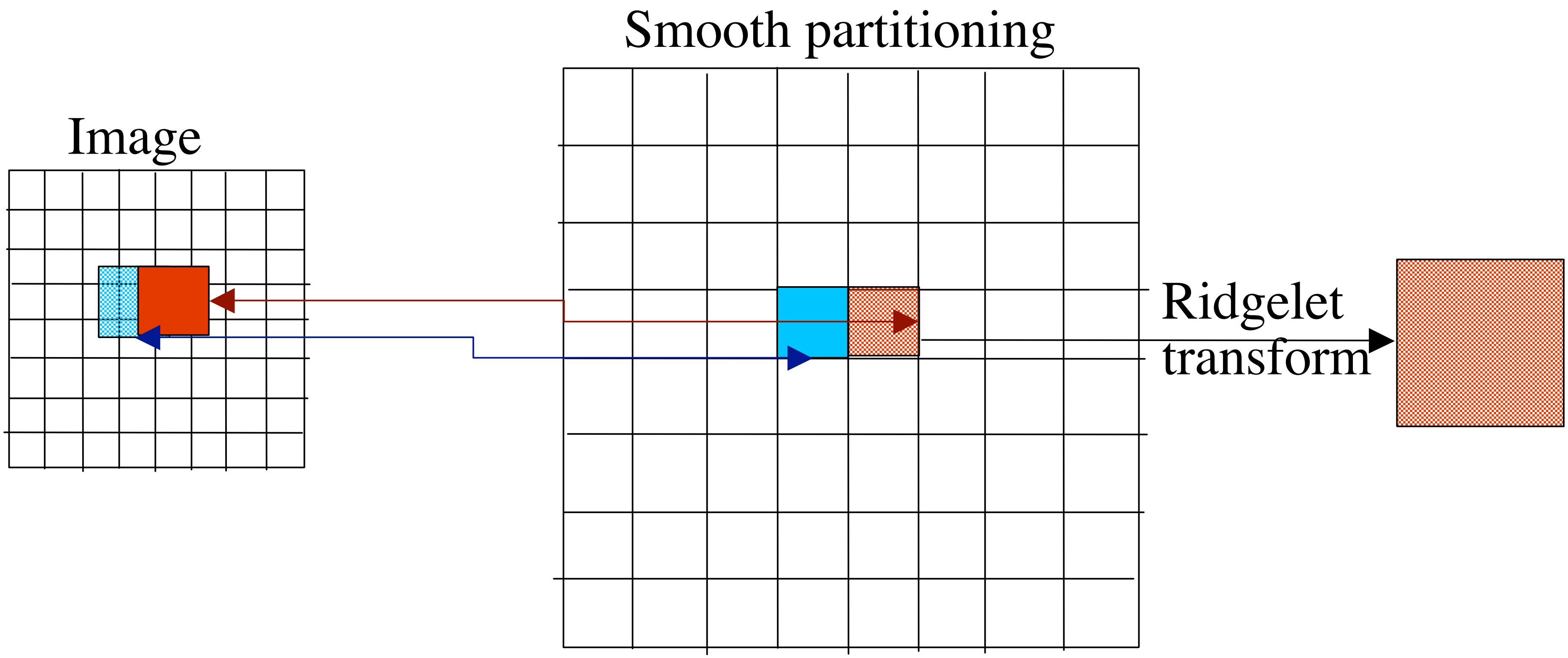


Local Ridgelet Transform

The ridgelet transform is optimal to find only lines of the size of the image. To detect line segments, a partitioning must be introduced. The image is decomposed into blocks, and the ridgelet transform is applied on each block.

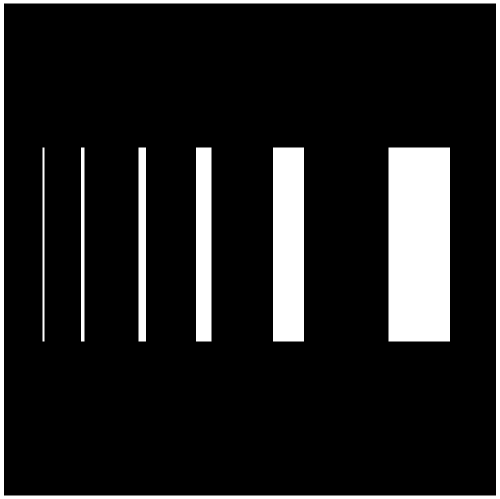


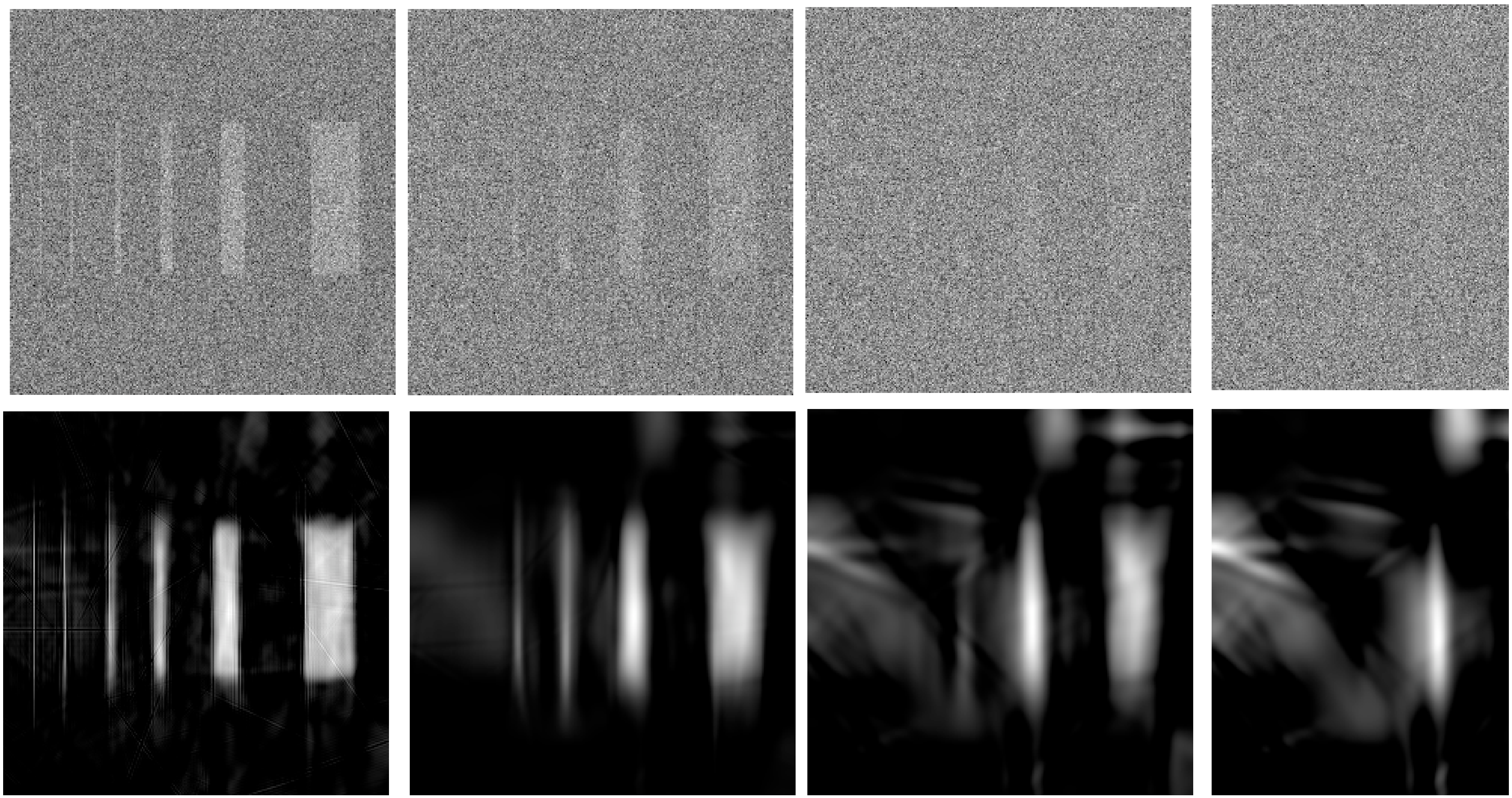
In practice, we use overlap to avoid blocking artifacts.



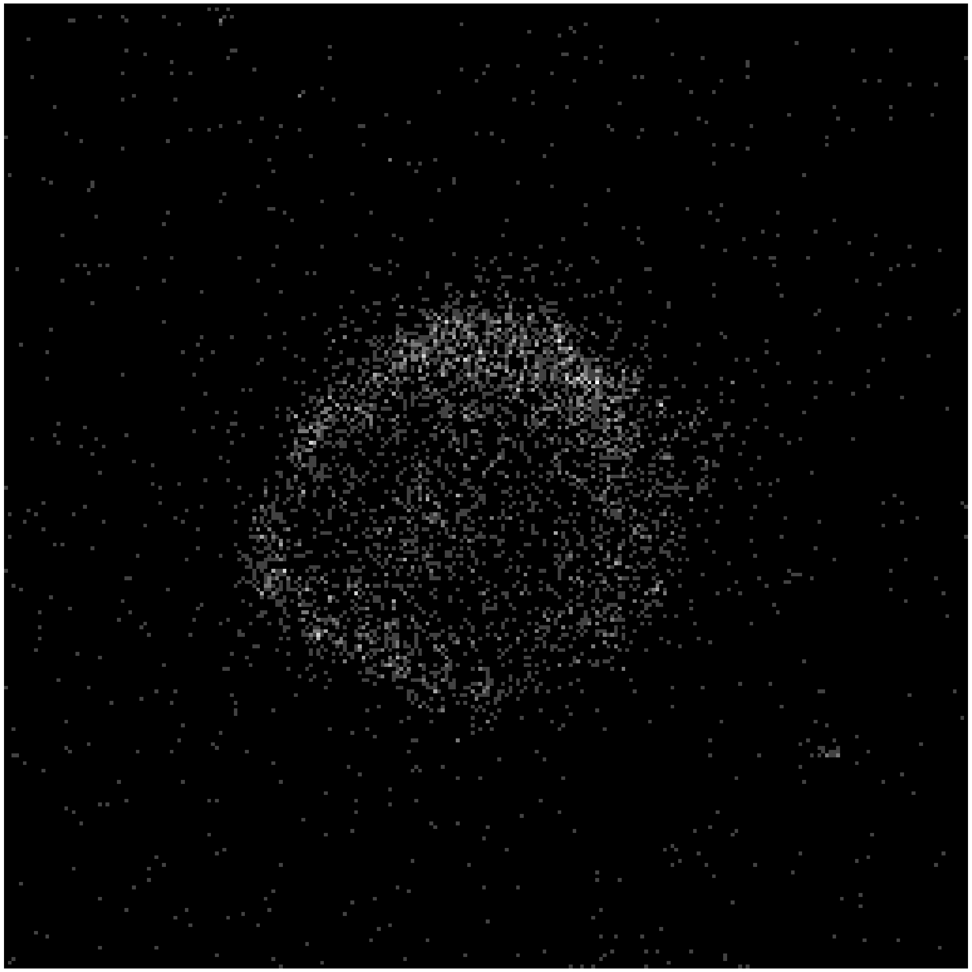
The partitioning introduces a redundancy, as a pixel belongs to 4 neighbouring blocks.

Line detection by the ridgelet transform

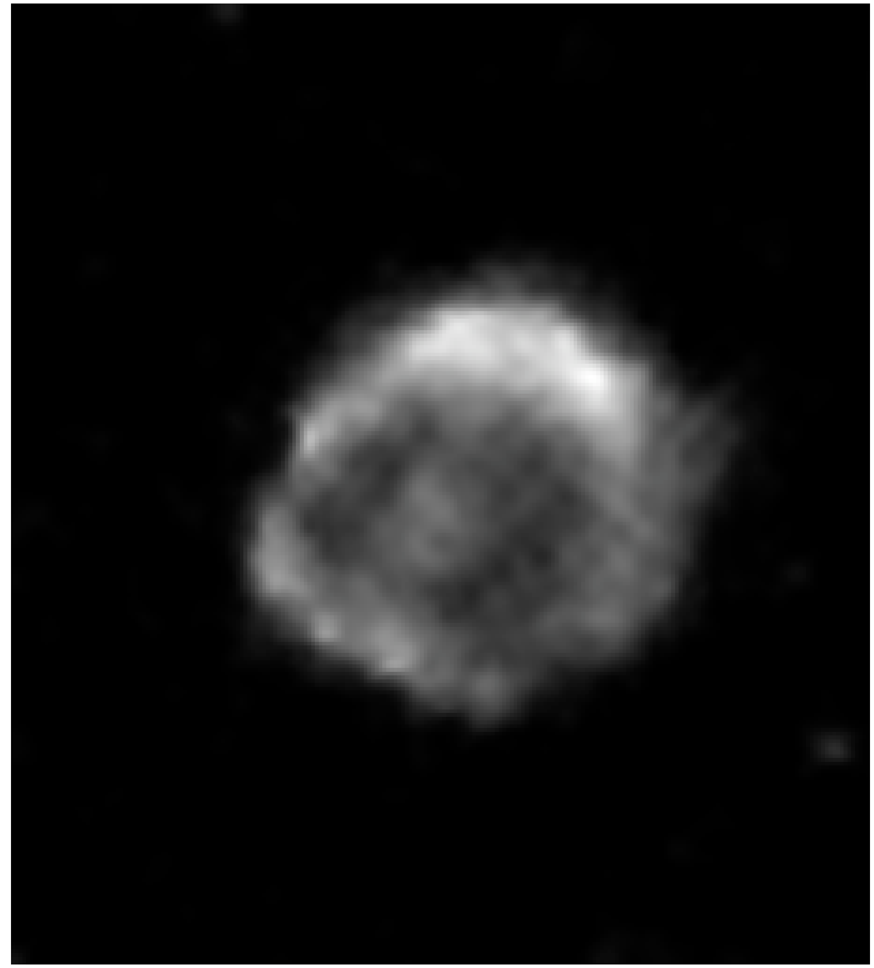




NEWTON/XMM Image of the supernovae SN1604



Ridgelet Filtering

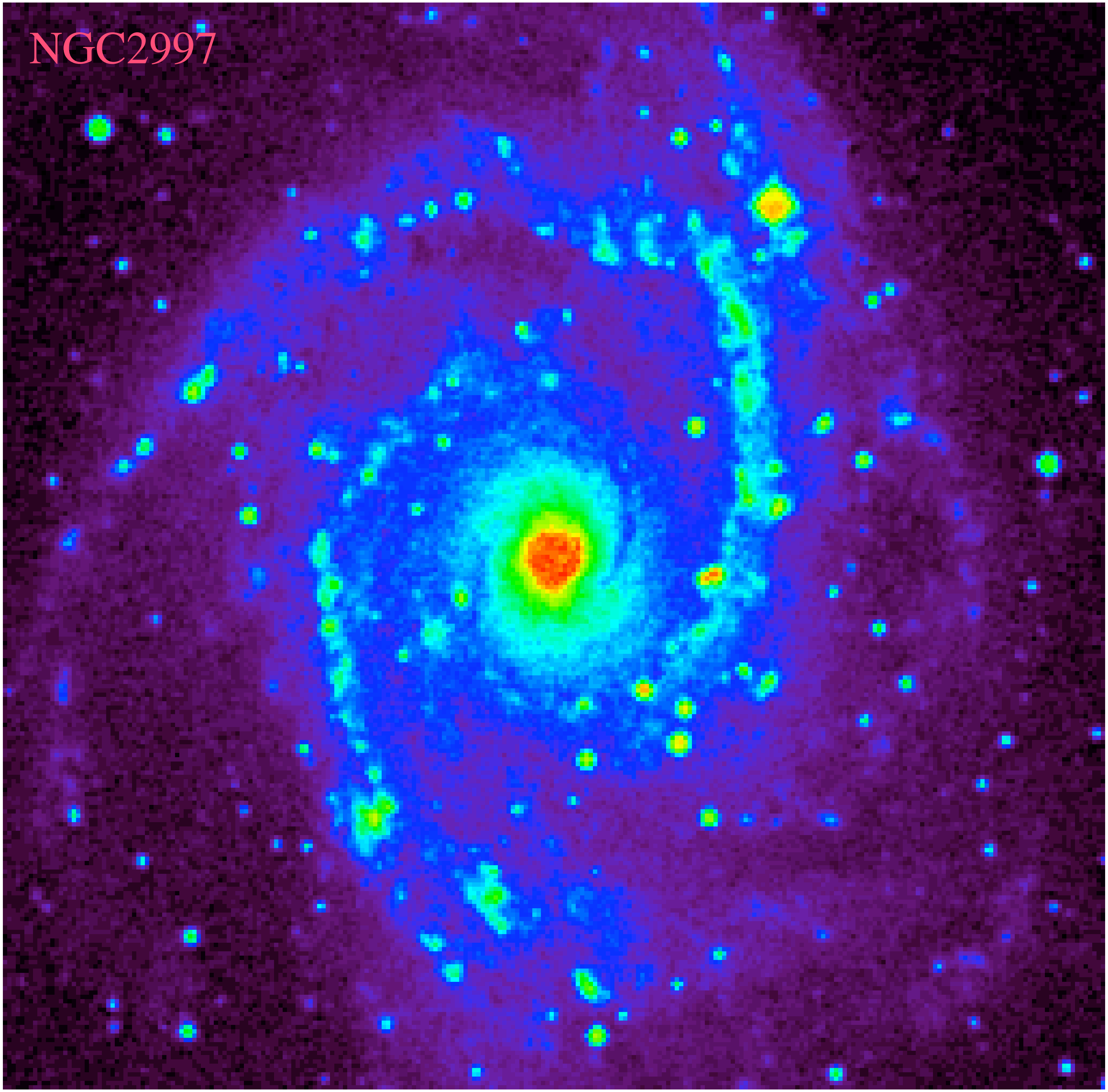


The Curvelet Transform

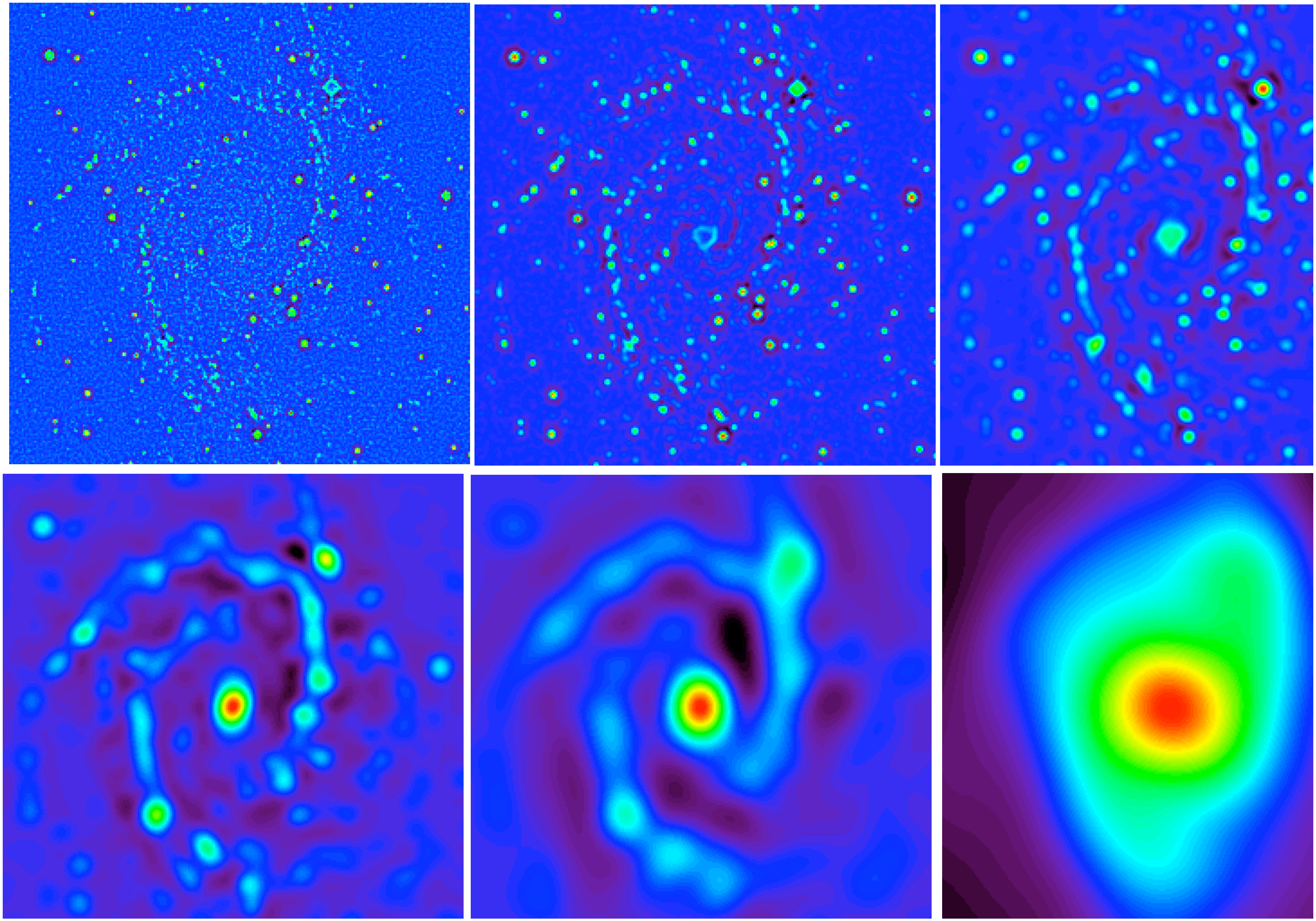
The curvelet transform opens us the possibility to analyse an image with different block sizes, but with a single transform.

The idea is to first decompose the image into a set of wavelet bands, and to analyze each band by a ridgelet transform. The block size can be changed at each scale level.

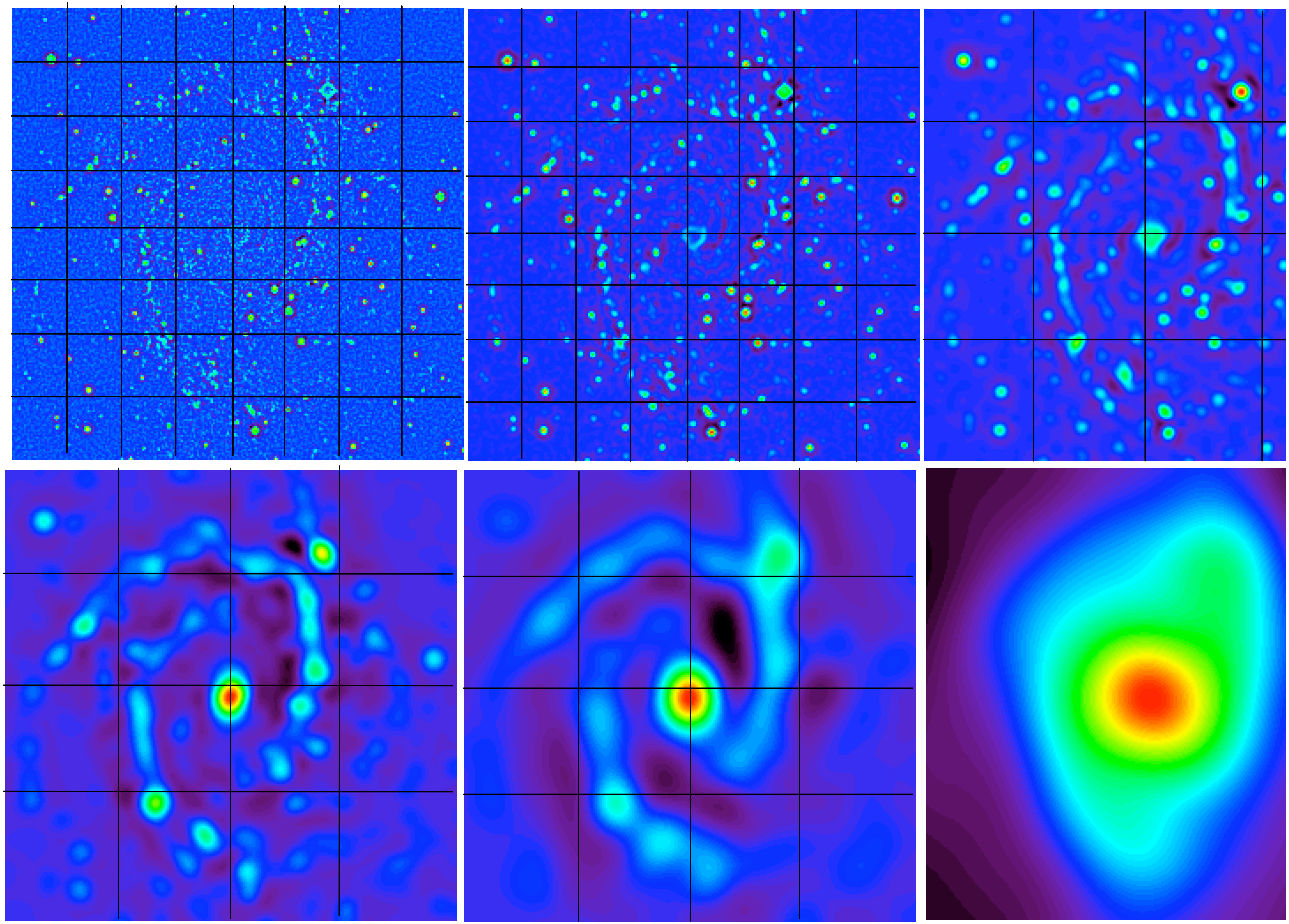
- à trous wavelet transform
- Partitionning
- ridgelet transform
    - . Radon Transform
    - . 1D Wavelet transform


NGC2997

A trous algorithm: $I(k,l) = c_{J,k,l} + \sum_{j=1}^{J} w_{j,k,l}$



PARTITIONING



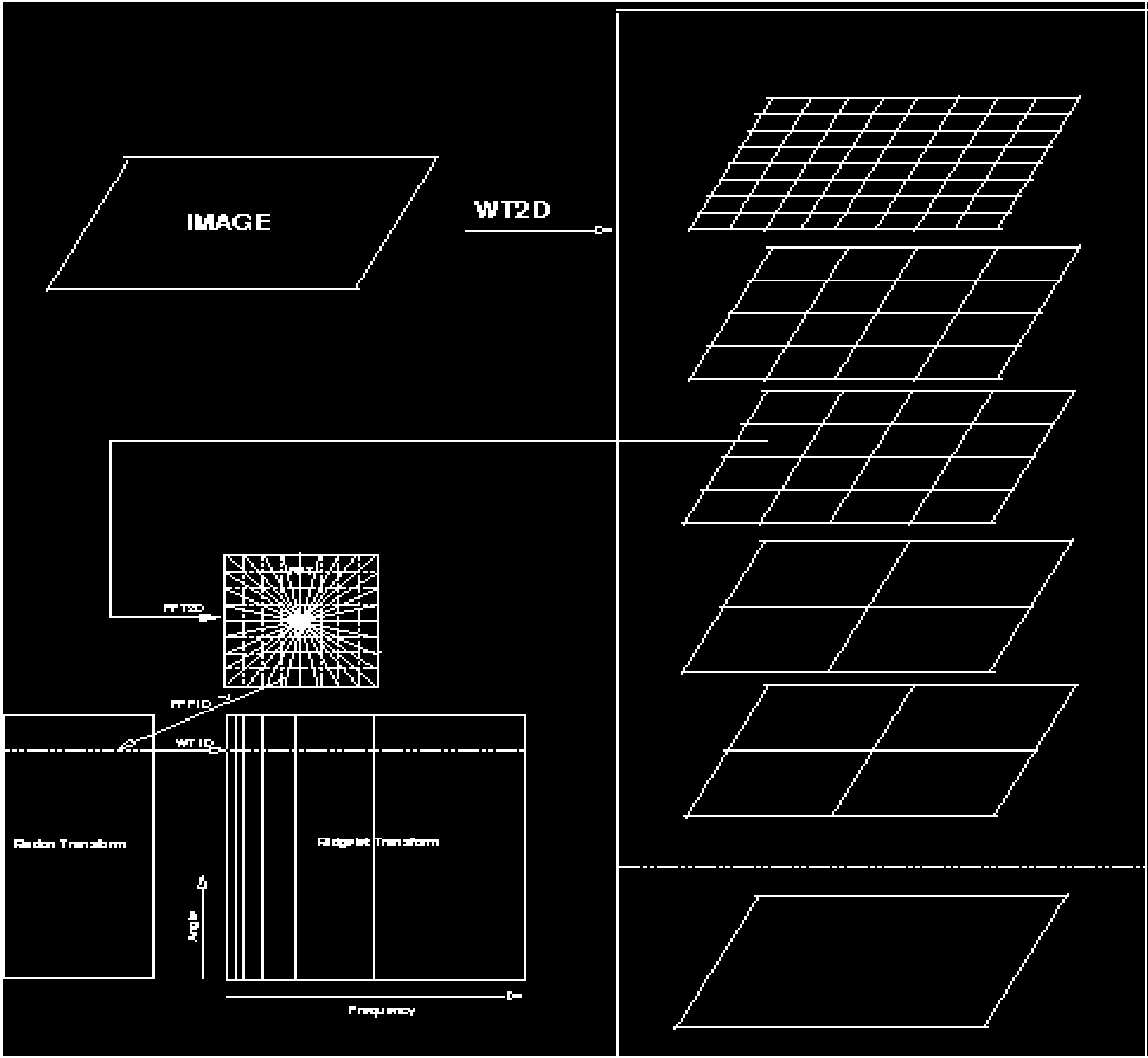


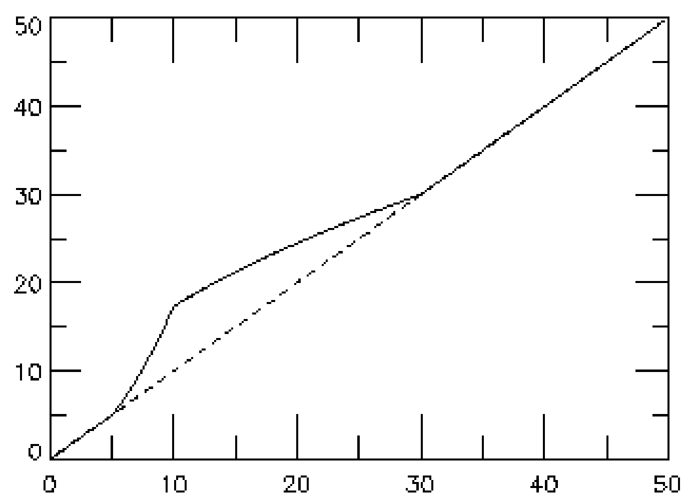
IMAGE
WT2D
FFT2D
FFT1D
WT1D
Radon Transform
Ridgelet Transform
Angle
Frequency

CONTRAST ENHANCEMENT

$$\tilde{I} = C_R\left(y_c\left(C_T I\right)\right)$$

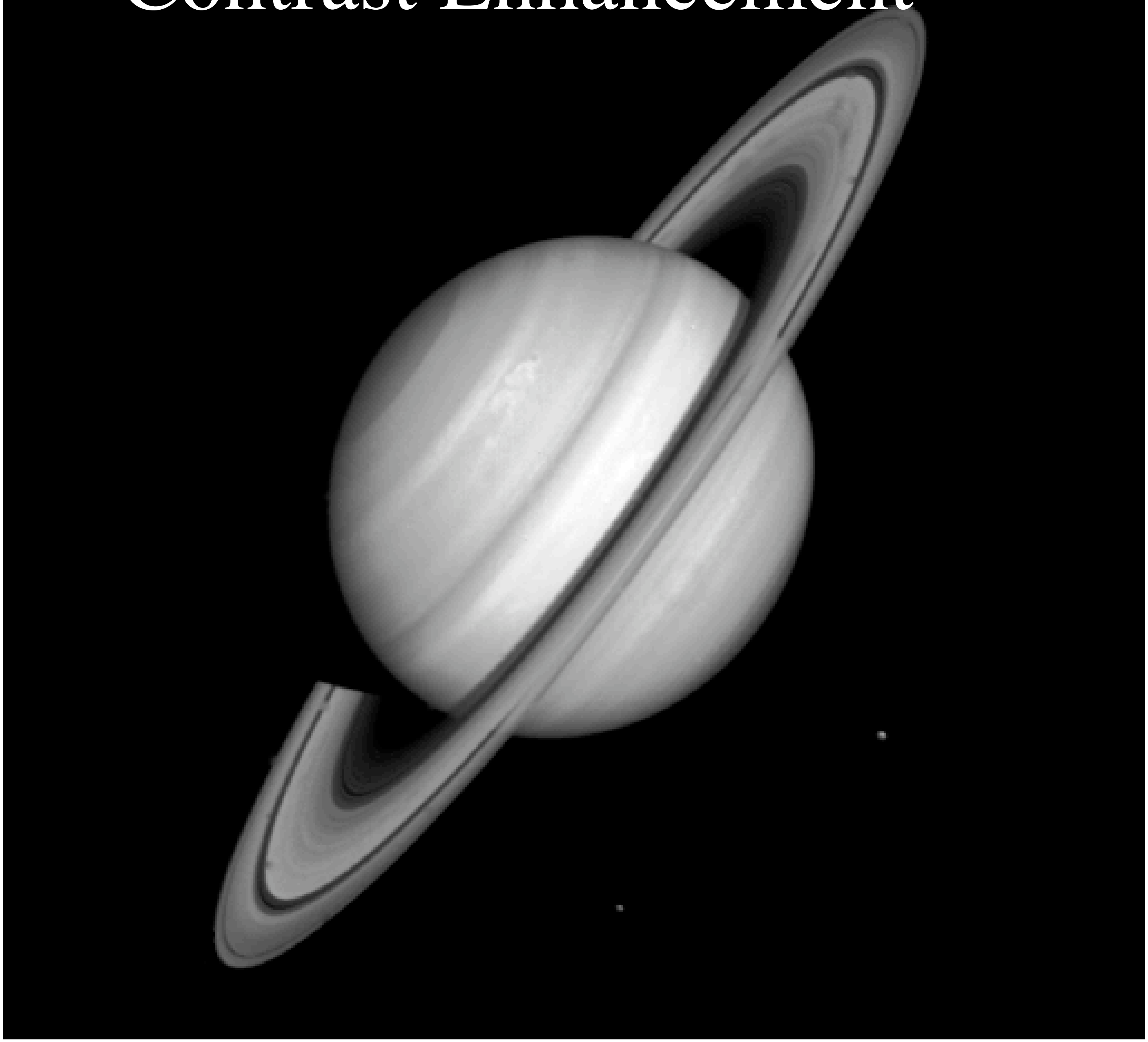
$$\begin{cases} y_c(x,\sigma) = 1 & \text{if} \quad x < c\sigma \\ y_c(x,\sigma) = \dfrac{x - c\sigma}{c\sigma}\left(\dfrac{m}{c\sigma}\right)^p + \dfrac{2c\sigma - x}{c\sigma} & \text{if} \quad x < 2c\sigma \\ y_c(x,\sigma) = \left(\dfrac{m}{x}\right)^p & \text{if} \quad 2c\sigma \le x < m \\ y_c(x,\sigma) = \left(\dfrac{m}{x}\right)^s & \text{if} \quad x > m \end{cases}$$

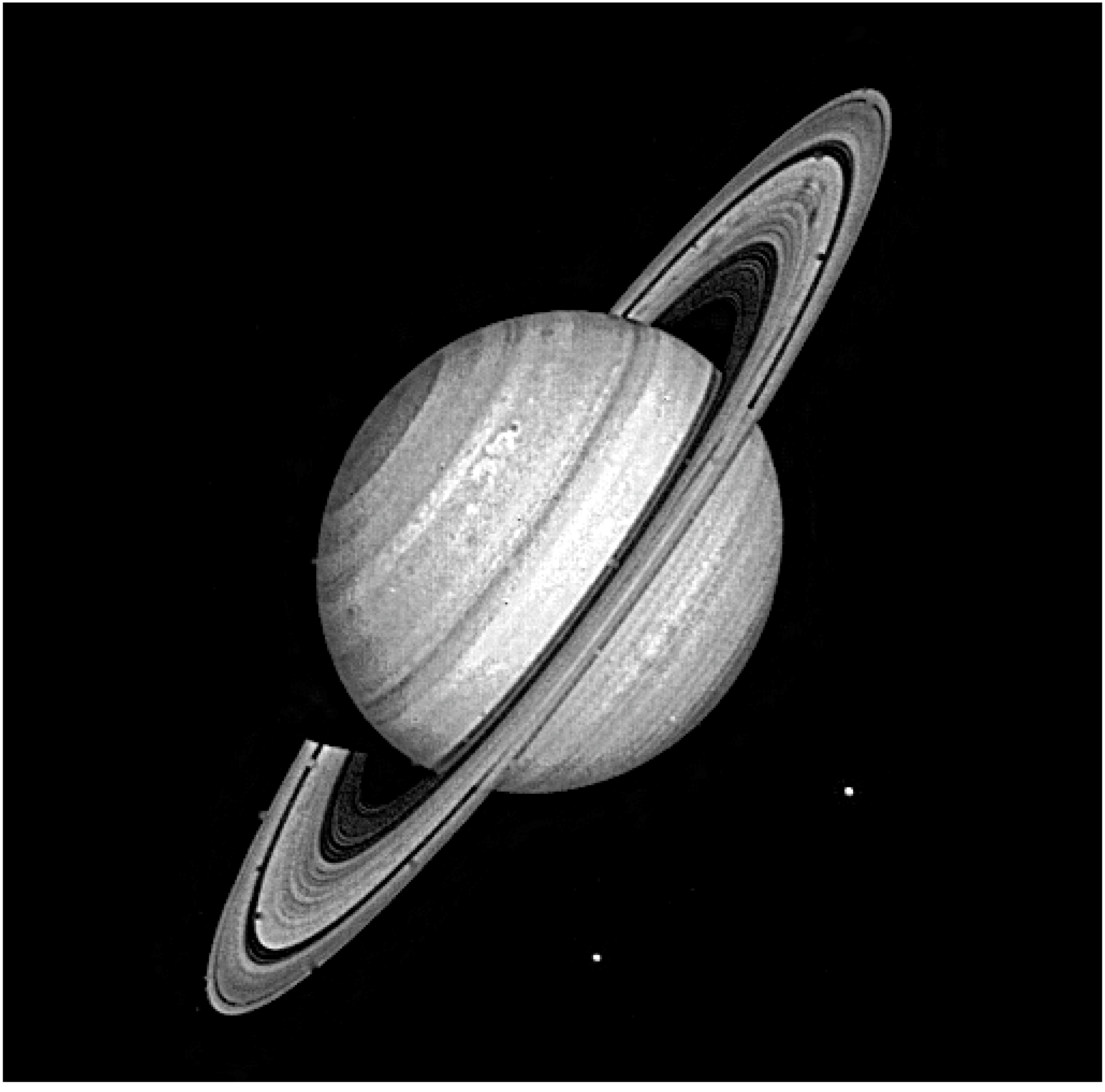
Modified curvelet coefficient

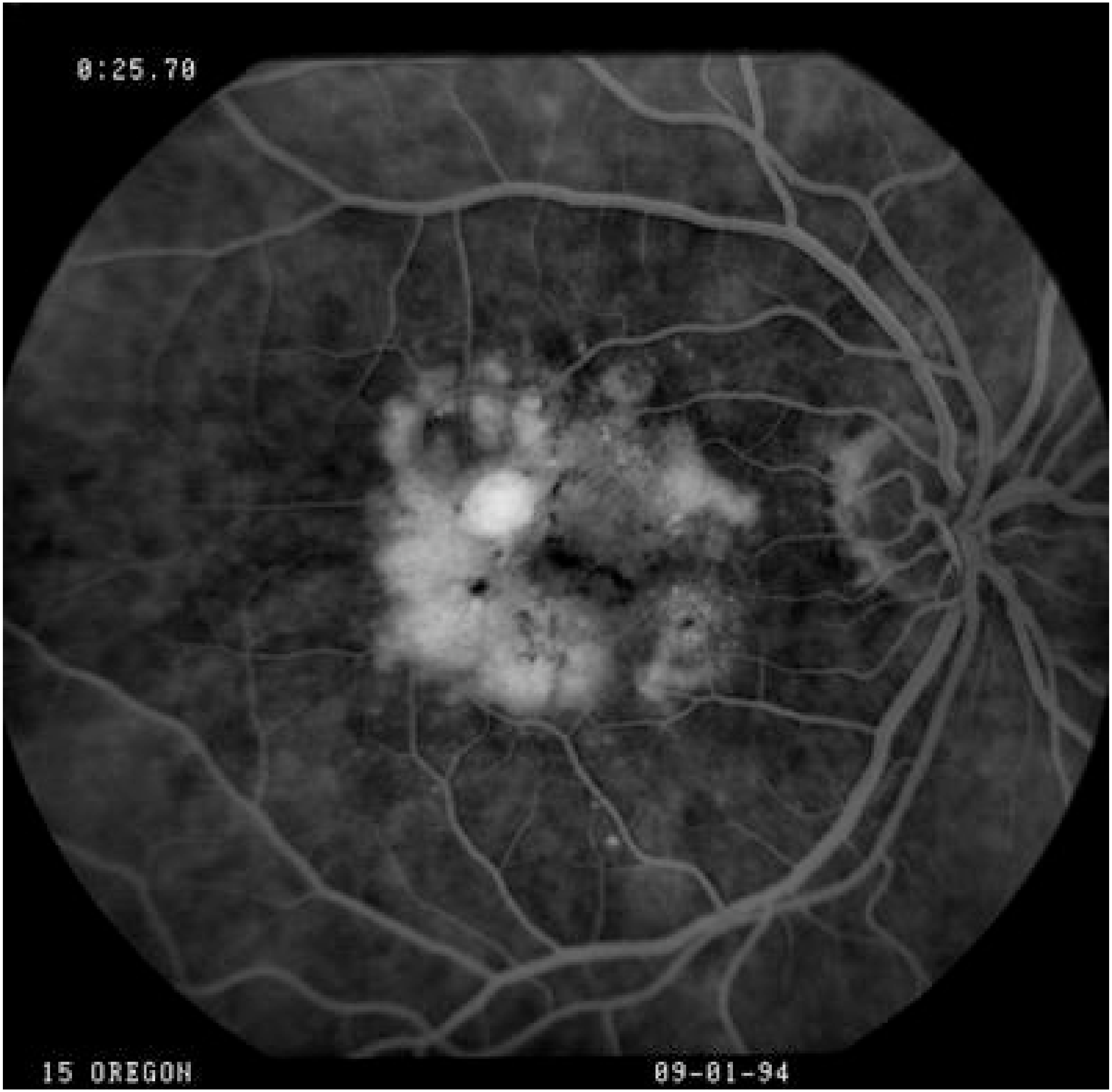


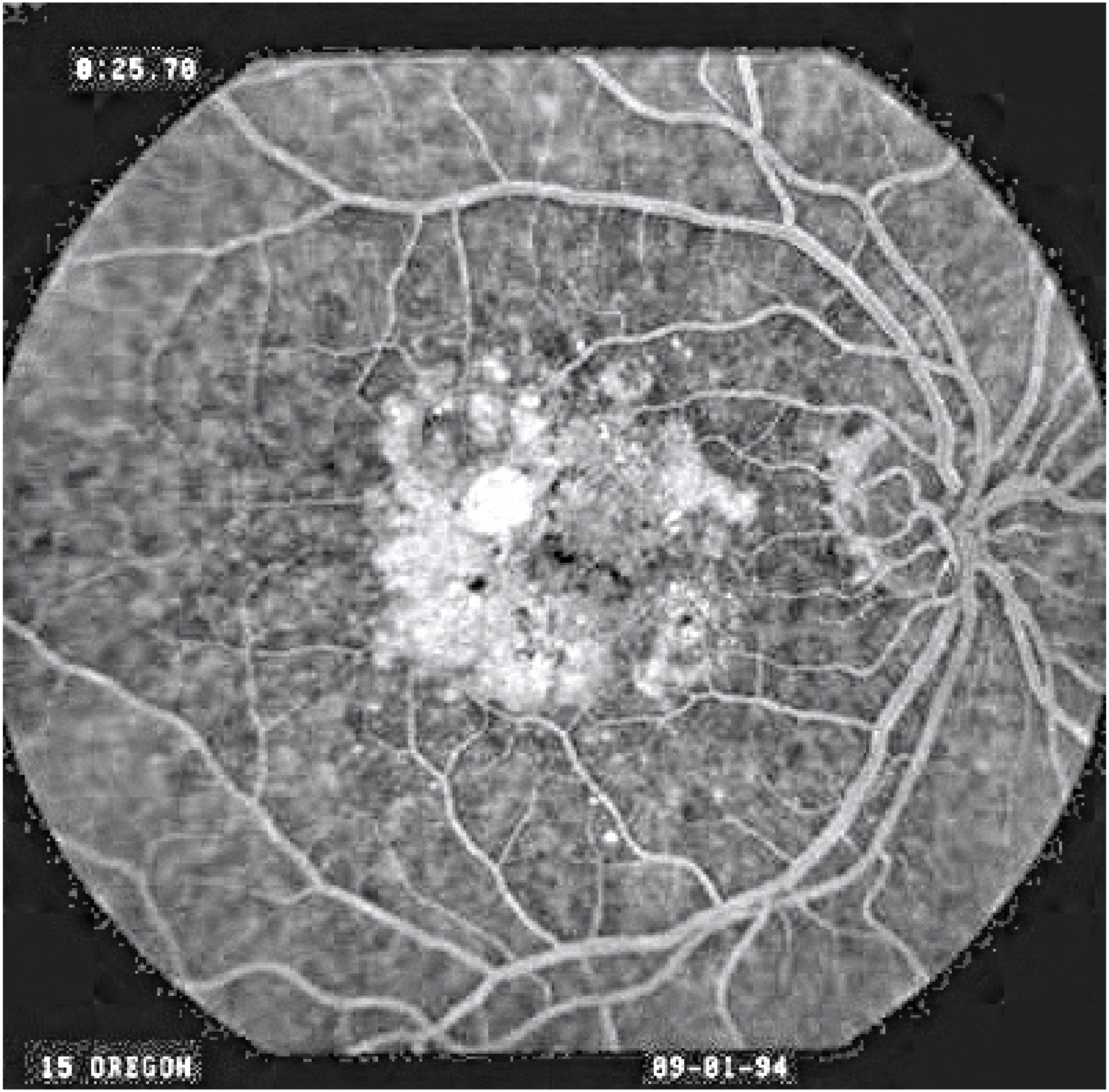
Curvelet coefficient

Contrast Enhancement






0:25.70
15 OREGON
09-01-94


0:25.70
15 OREGON
09-01-94

Color Images

- R,G,B ===> L, U, V

- We apply the curvelet transform to the three components

- At each scale and at each position, we calculate:

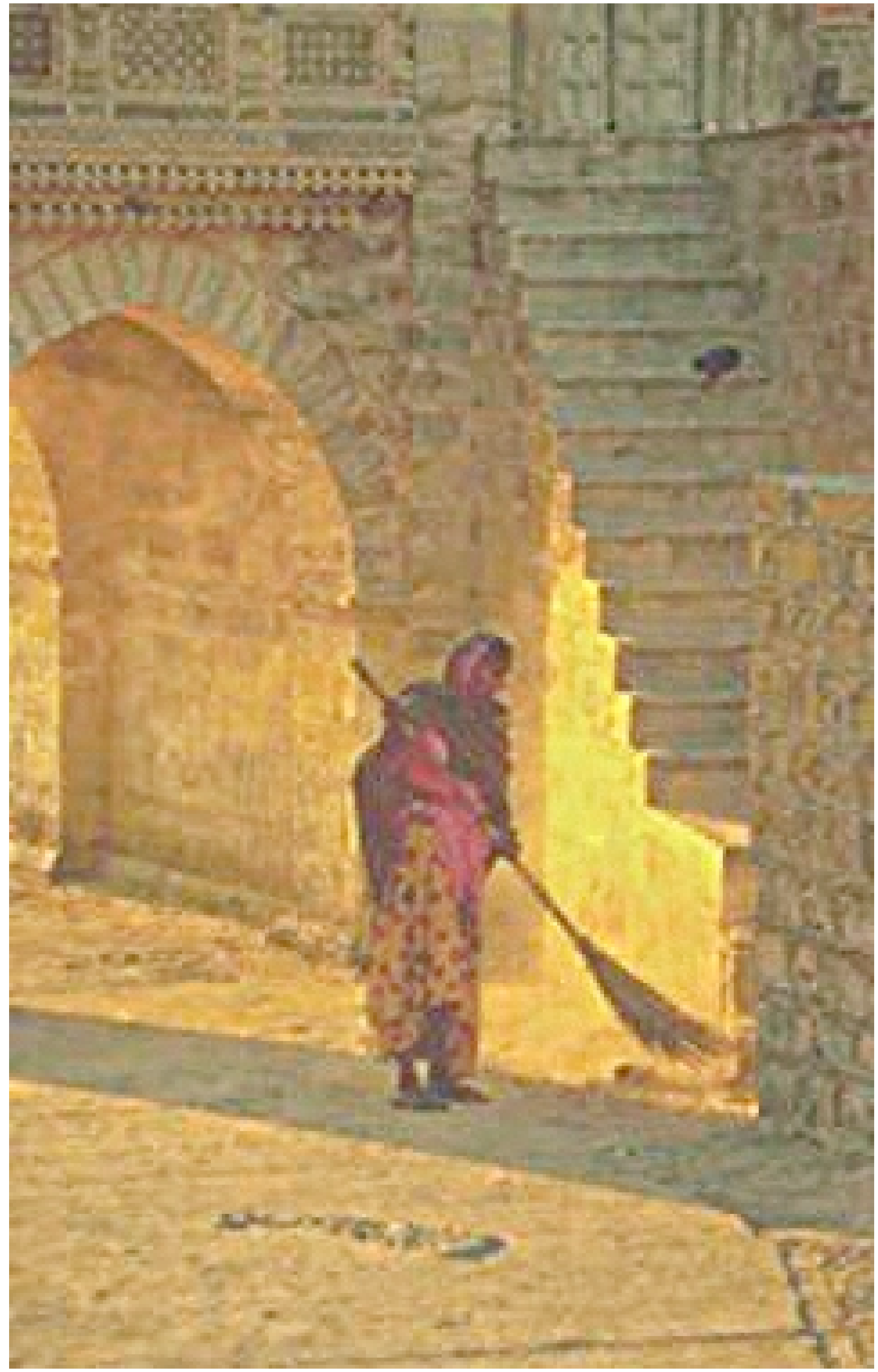
$$e = \sqrt{c_L^2 + c_u^2 + c_v^2}$$

- Coefficients correction

$$\left(\tilde{c}_L, \tilde{c}_U, \tilde{c}_V\right) = \left(y_c(e,\sigma)c_L, y_c(e,\sigma)c_U, y_c(e,\sigma)c_V\right.$$

- Inverse curvelet transform

- L,UV to RGB

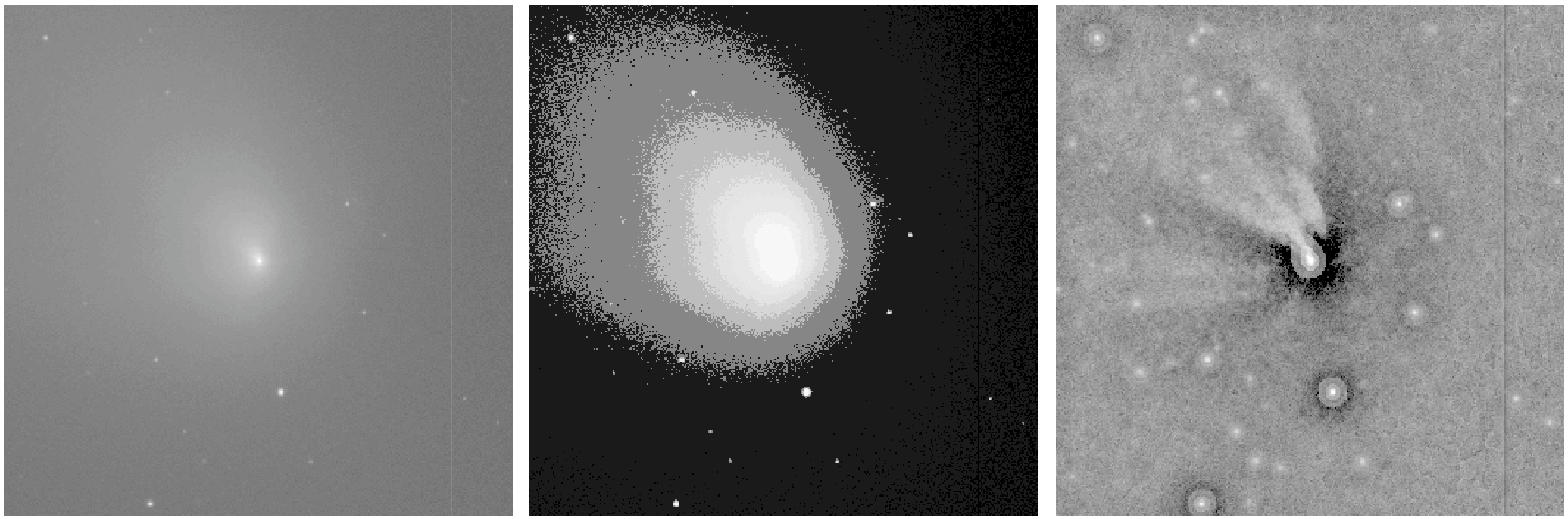



Dynamic Range Compression

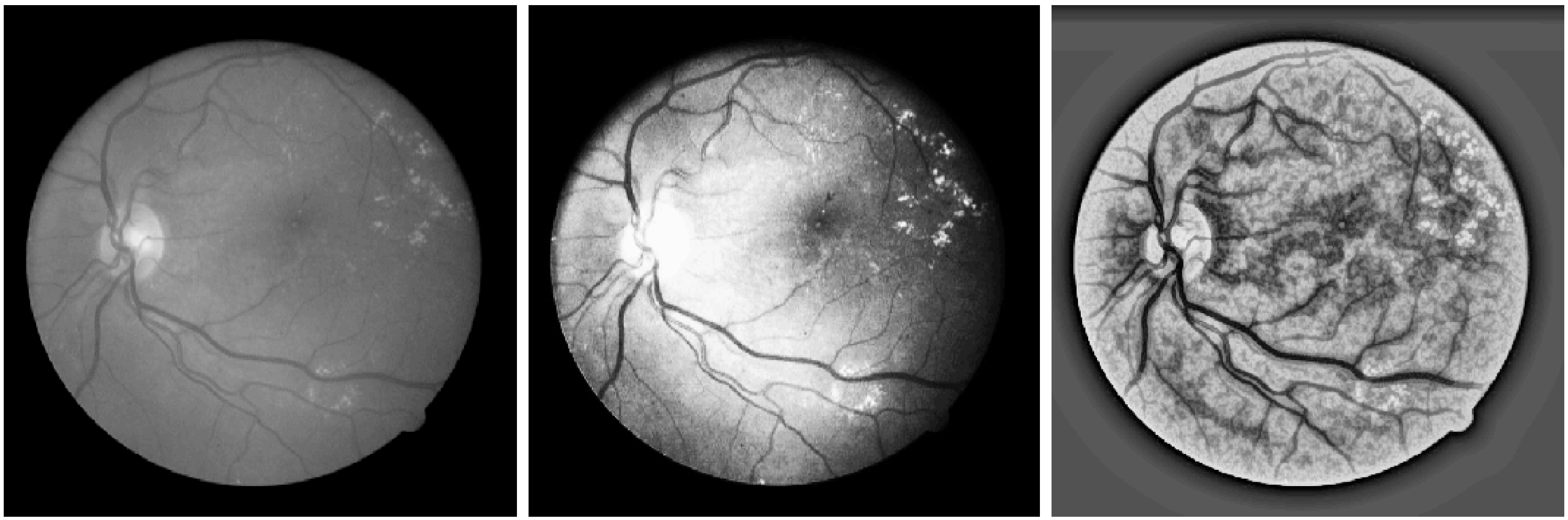
Images with a high dynamic range are also difficult to analyze. For example, astronomers generally visualize their images using a logarithmic look-up-table conversion.

Wavelet can also be used to compress the dynamic range at all scales, and therefore allows us to clearly see some very faint features. For instance, the wavelet-log representations consists in replacing $w_{j,k,l}$ by $\log(|w_{j,k,l}|)$, leading to the alternative image

$$I_{k,l} = \log(c_{J,k,l}) + \sum_{j=1}^{J} \text{sgn}(w_{j,k,l}) \log(\mid w_{j,k,l} \mid + \epsilon)$$



Left - Hale-Bopp Comet image. Middle - histogram equalization results, Right - wavelet-log representations.



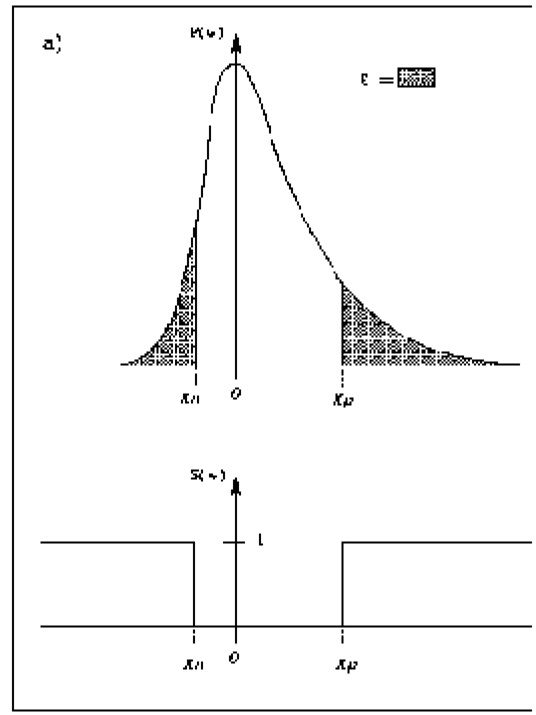
Left, opthalmic medical image. Middle - histogram equalization results, Bottom - wavelet-log representations.

WAVELET FILTERING

NOISE MODELING

For a positive coefficient: $P = \Pr ob(w > w_{j,x,y})$

For a negative coefficient: $P = \Pr ob(w < w_{j,x,y})$



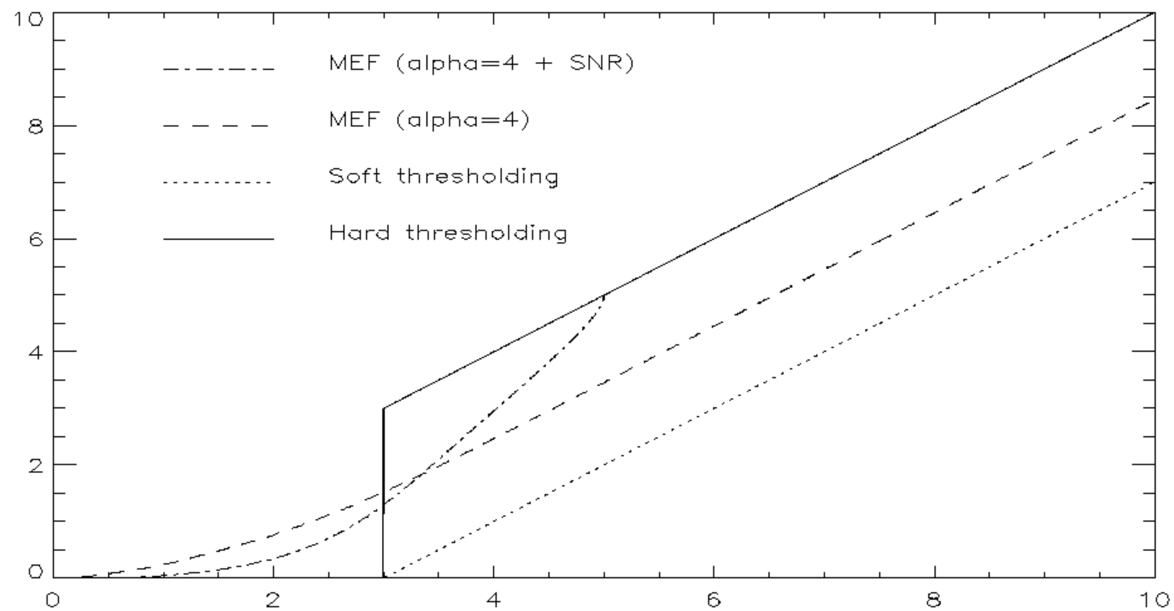
Given a threshold t:
if P > t, the coefficient could be due to the noise.
if P < t, the coefficient cannot be due to the noise, and a **significant coefficient** is detected.

Hard Thresholding:
$$\delta(c) = c \quad if\ |c| \geq t$$
$$= 0 \quad if\ |c| < t$$

Soft Thresholding:
$$\delta(c) = \operatorname{sgn}(c)\left(|c| - t\right)_+$$

$$\tilde{y} = W_R[\delta(W_T y)]$$

Filtered wavelet coefficients versus wavelet coefficients



Noise Modeling in the wavelet space

The noise in the data follows a distribution law which can be:

- a White Gaussian Noise
- Correlated Noise
- a Poisson Noise
- a Poisson + Gaussian distribution (noise in the CCD)
- Poisson noise with few events (Galaxies counting, X ray images, ...)
- Speckle noise
- Root Mean Square map: we have a noise standard deviation of each data value.

Noise modeling with less constraint on the noise behavior

If the noise doesn't follows any of these laws, we can derive a noise modeling from any of the following assumptions:

- it is additive, and non-stationary.
- it is multiplicative and stationary.
- it is multiplicative, but non-stationary.
- it is undefined but stationary.

Gaussian Noise

$$p(w_{j,l}) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-w_{j,l}^2/2\sigma_j^2}$$

Rejection of hypothesis $\mathcal{H}'$ depends (for a positive coefficient value) on:

$$P = Prob(w_{j,l} > W) = \frac{1}{\sqrt{2\pi}\sigma_j} \int_{w_{j,l}}^{+\infty} e^{-W^2/2\sigma_j^2} dW$$

and if the coefficient value is negative, it depends on

$$P = Prob(w_{j,l} < W) = \frac{1}{\sqrt{2\pi}\sigma_j} \int_{-\infty}^{w_{j,l}} e^{-W^2/2\sigma_j^2} dW$$

Given stationary Gaussian noise, it suffices to compare $w_{j,l}$ to $k\sigma_j$.

if $\mid w_j \mid \geq \ k\sigma_j$ then w_j is significant

if $\mid w_j \mid < \ k\sigma_j$ then w_j is not significant

Hard Thresholding

$$\tilde{w}_j = w_j \text{ if } \mid w_j \mid \geq T_j$$
$$= 0 \text{ otherwise}$$

with $T_j = k\sigma_j$, with k between 3 and 5. For a energy normalized wavelet transform algorithm, we have $\sigma_j = \sigma$ for all j.

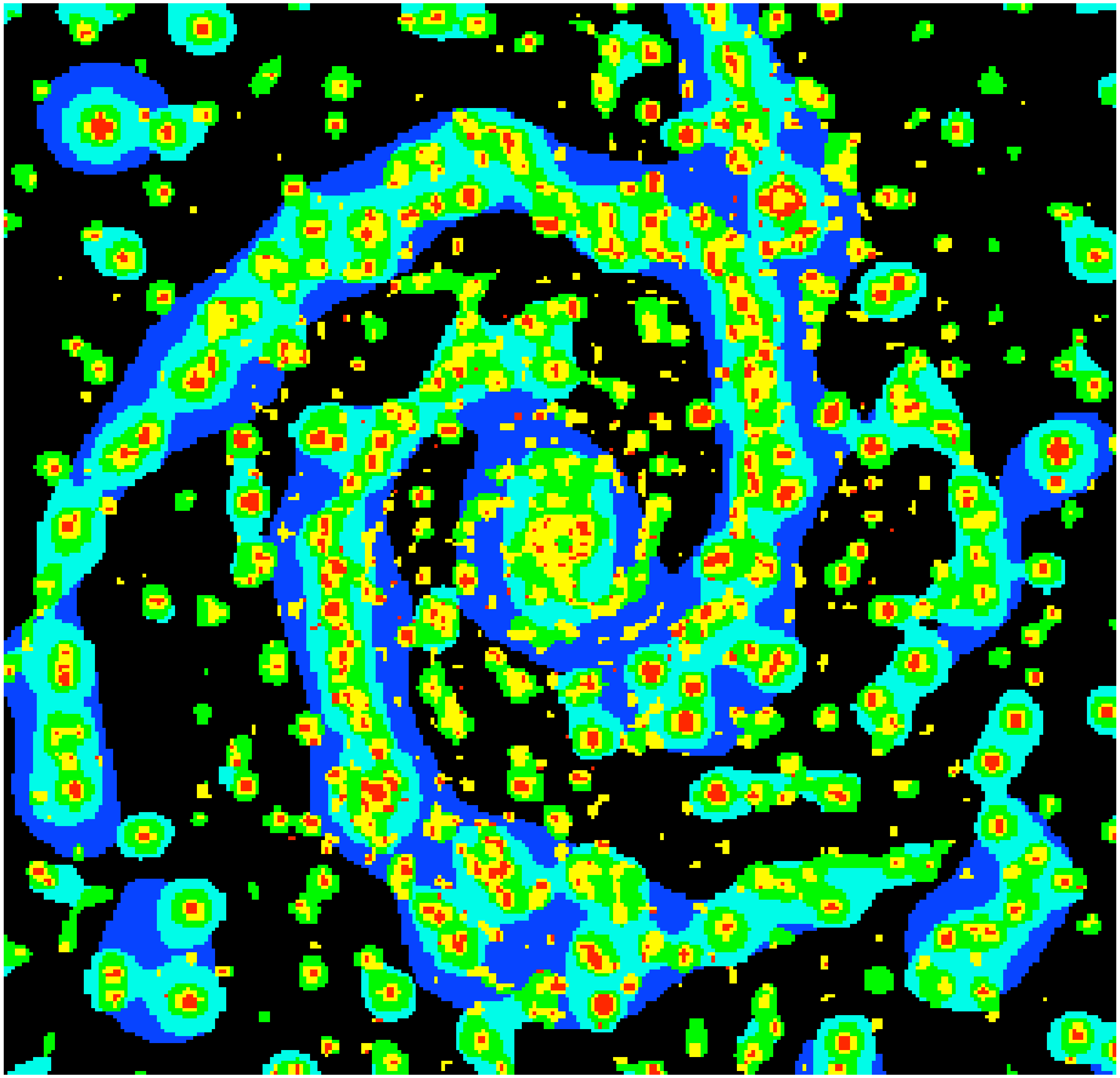
ASTRONOMICAL DATA FILTERING

WE NEED TO PRESERVE THE FLUX: $(Ws)_l = (W\tilde{s})_l$

MULTIRESOLUTION SUPPORT

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

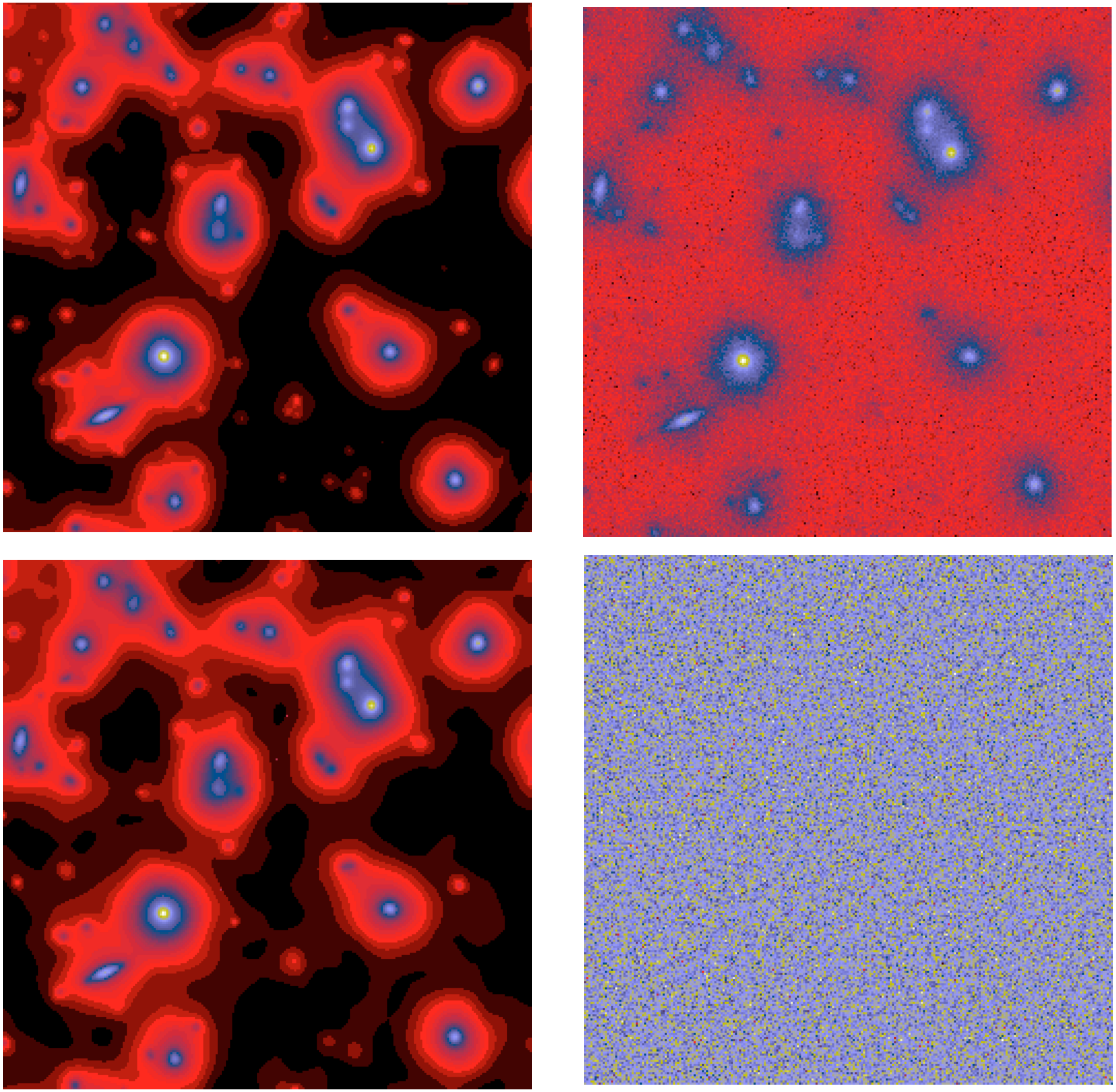
NGC2997 MULTIRESOLUTION SUPPORT

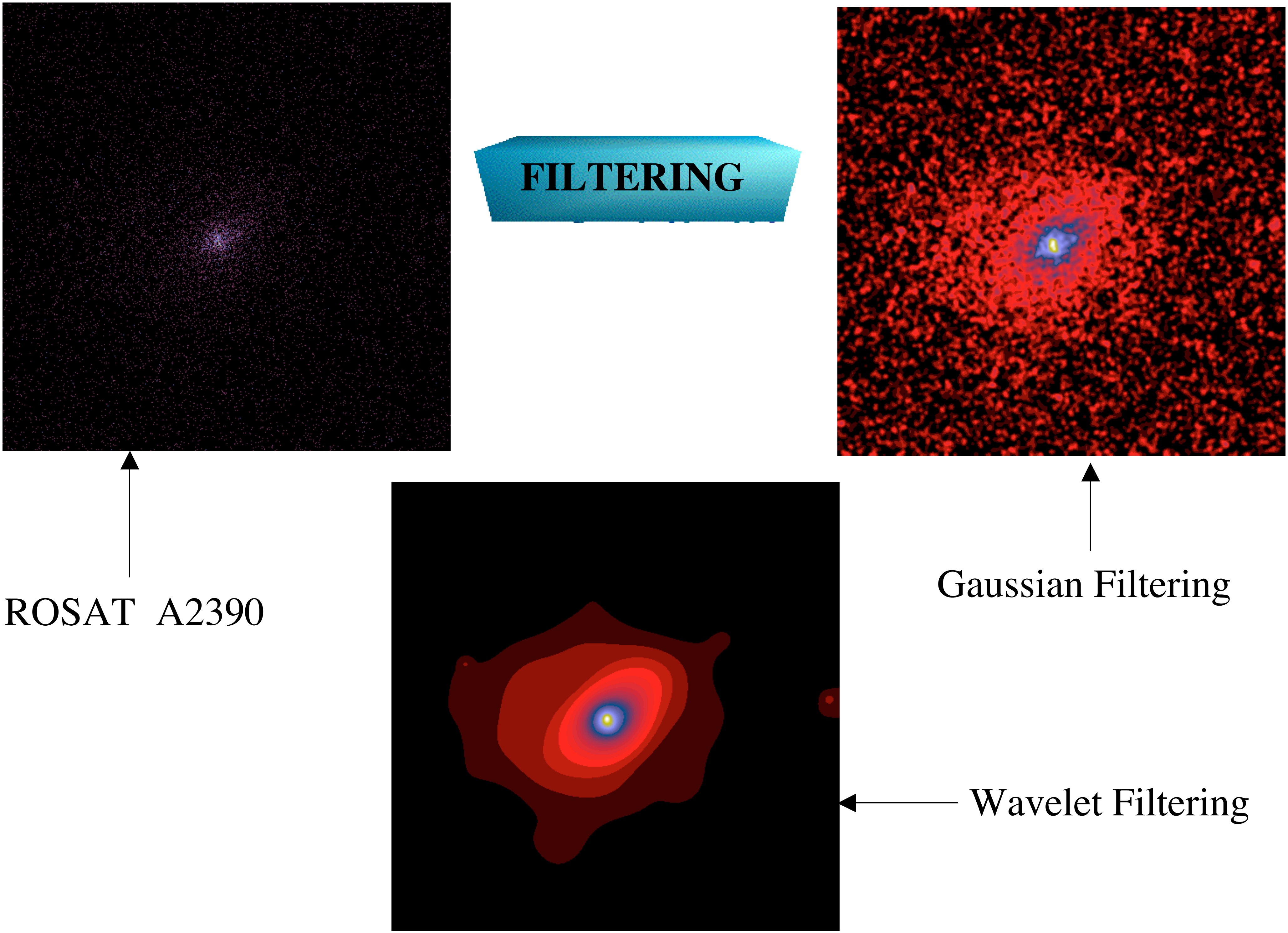


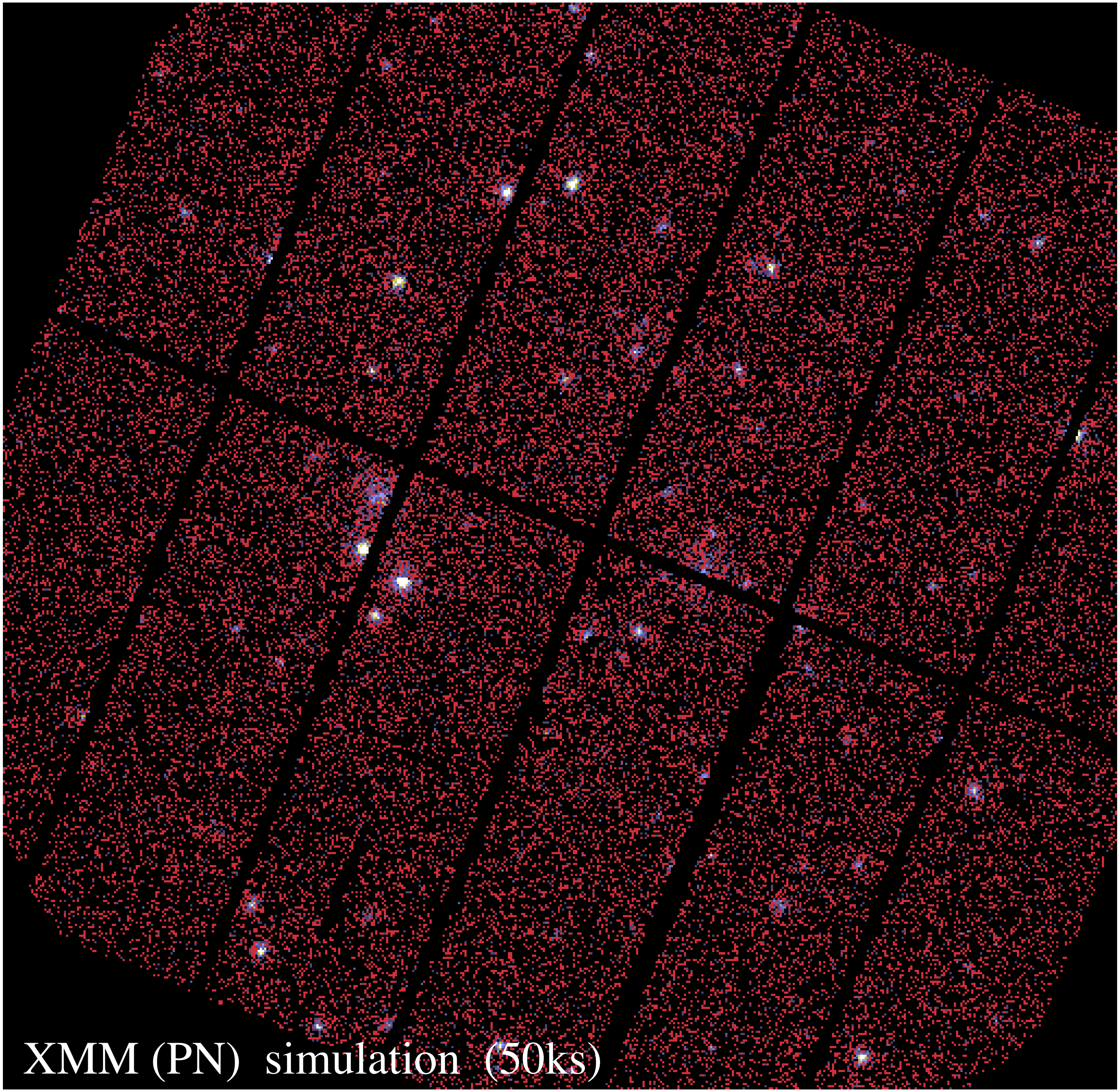
Iterative Filtering

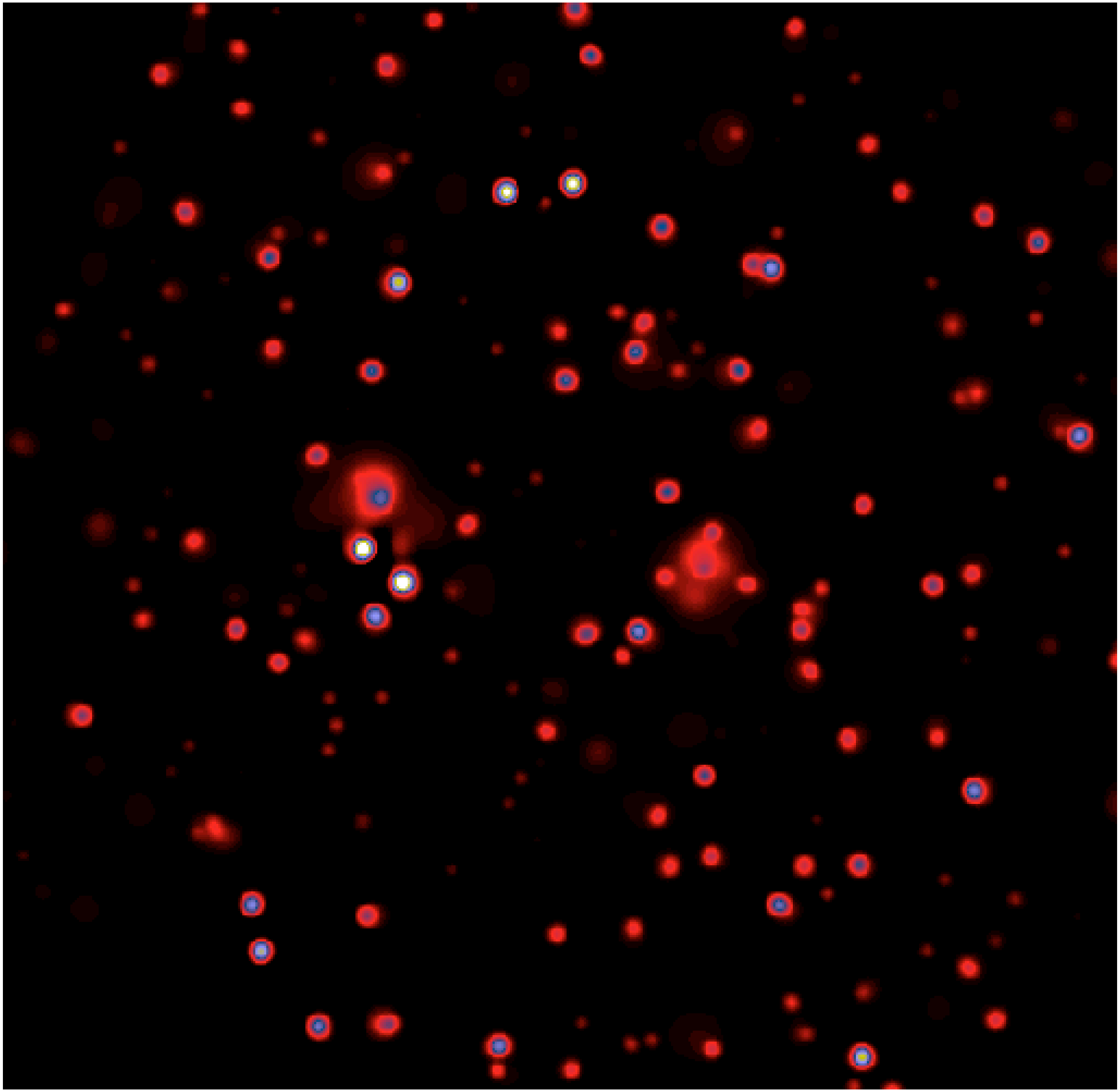
The filtering can be seen as an inversed problem. Indeed, we want to reconstruct an image from the detected wavelet coefficient. The problem of reconstruction consists in searching a signal $\tilde{s}$ such that its wavelet coefficients are the same as those of the detected structure. The solution is found by minimization of

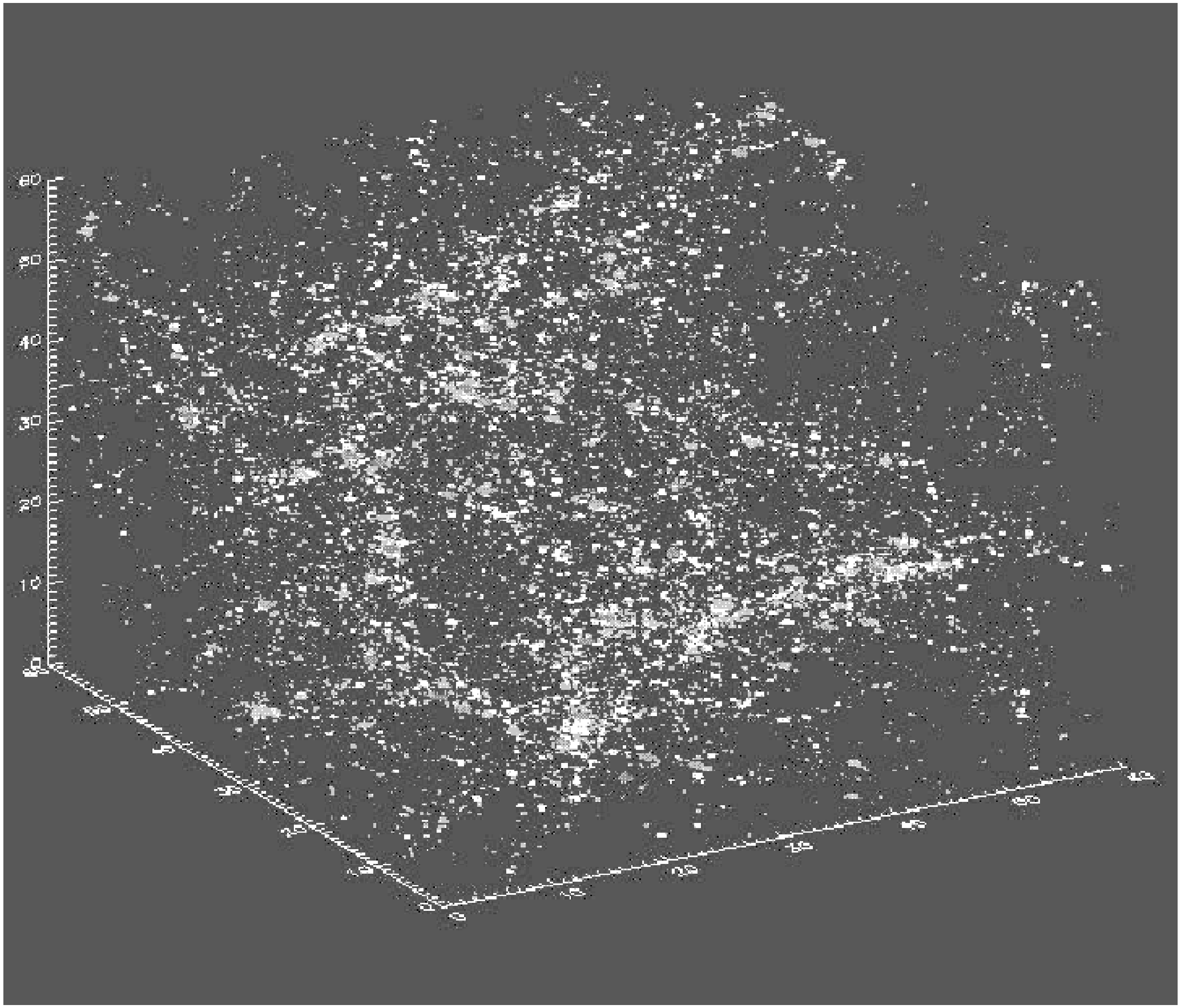
$$J(\tilde{s}) = \parallel M[\mathcal{W}s - \mathcal{W}\tilde{s}] \parallel^2 + \mathcal{C}(\tilde{s})$$

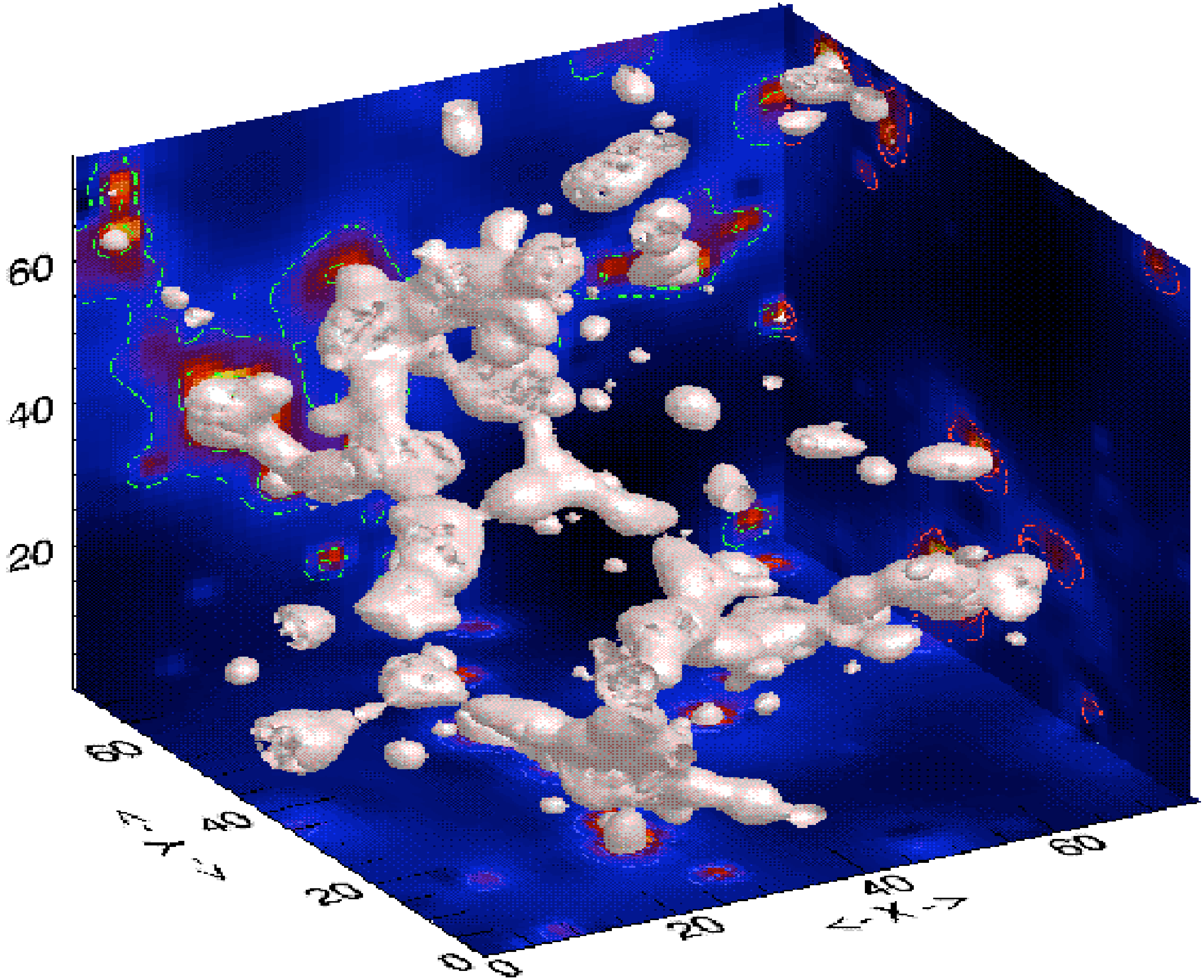



FILTERING
ROSAT A2390
Gaussian Filtering
Wavelet Filtering


XMM (PN) simulation (50ks)







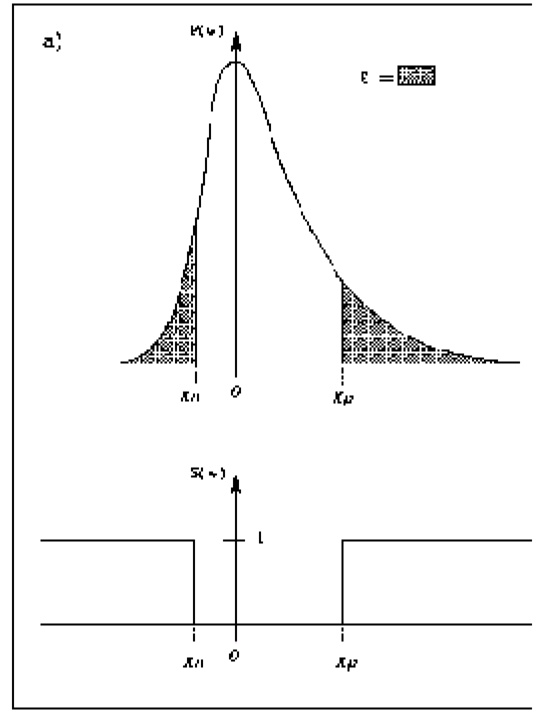
60
40
20
60
40
20
0
0
20
40
60

CURVELET FILTERING

NOISE MODELING

For a positive coefficient: $P = Prob\{W \geq w\}$

For a negative coefficient $P = Prob\{W \leq w\}$

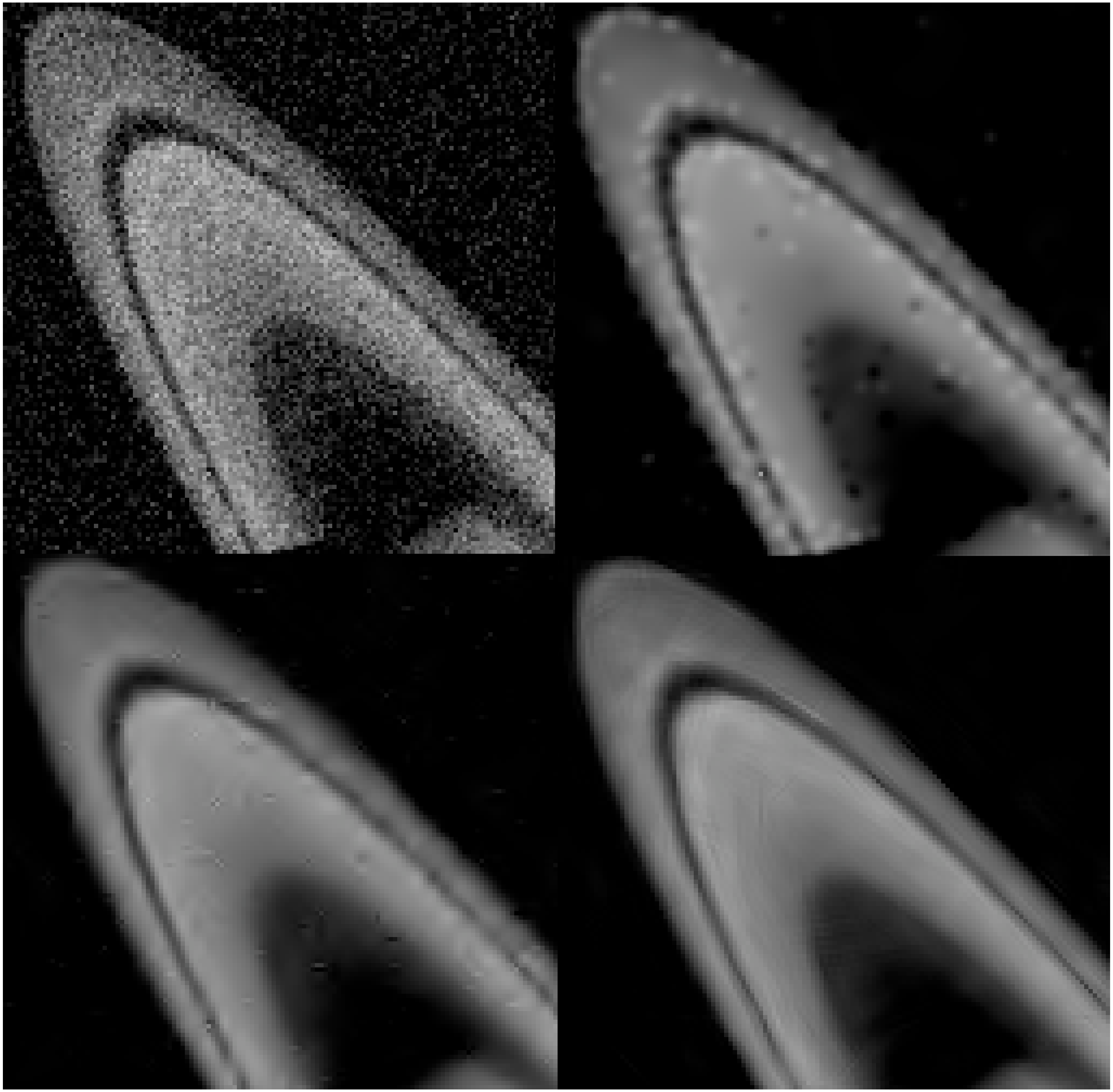


Given a threshold t:
if P > t, the coefficient could be due to the noise.
if P < t, the coefficient cannot be due to the noise, and a **significant coefficient** is detected.

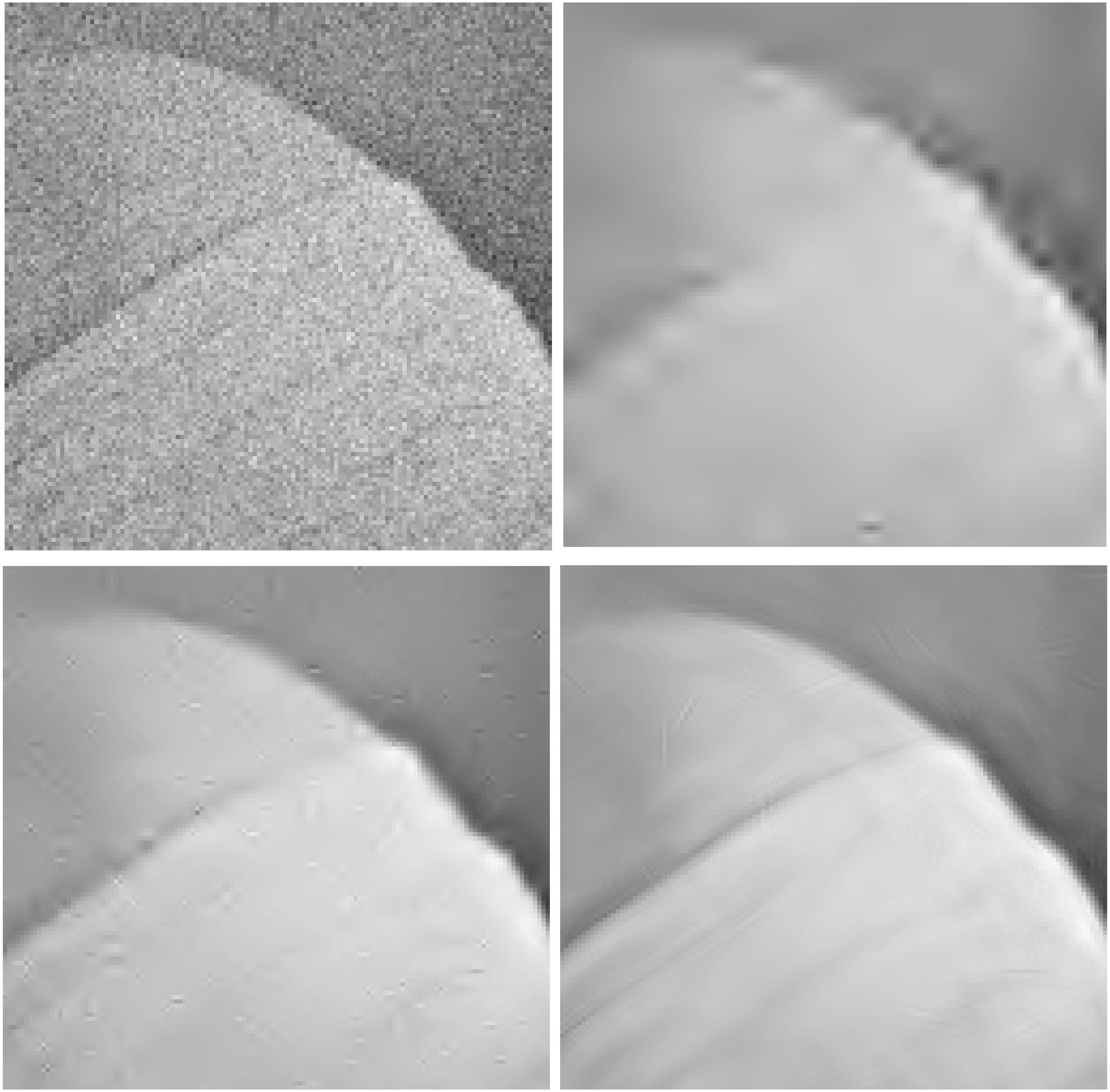
$$\tilde{y} = C_R[\delta(C_T y)]$$

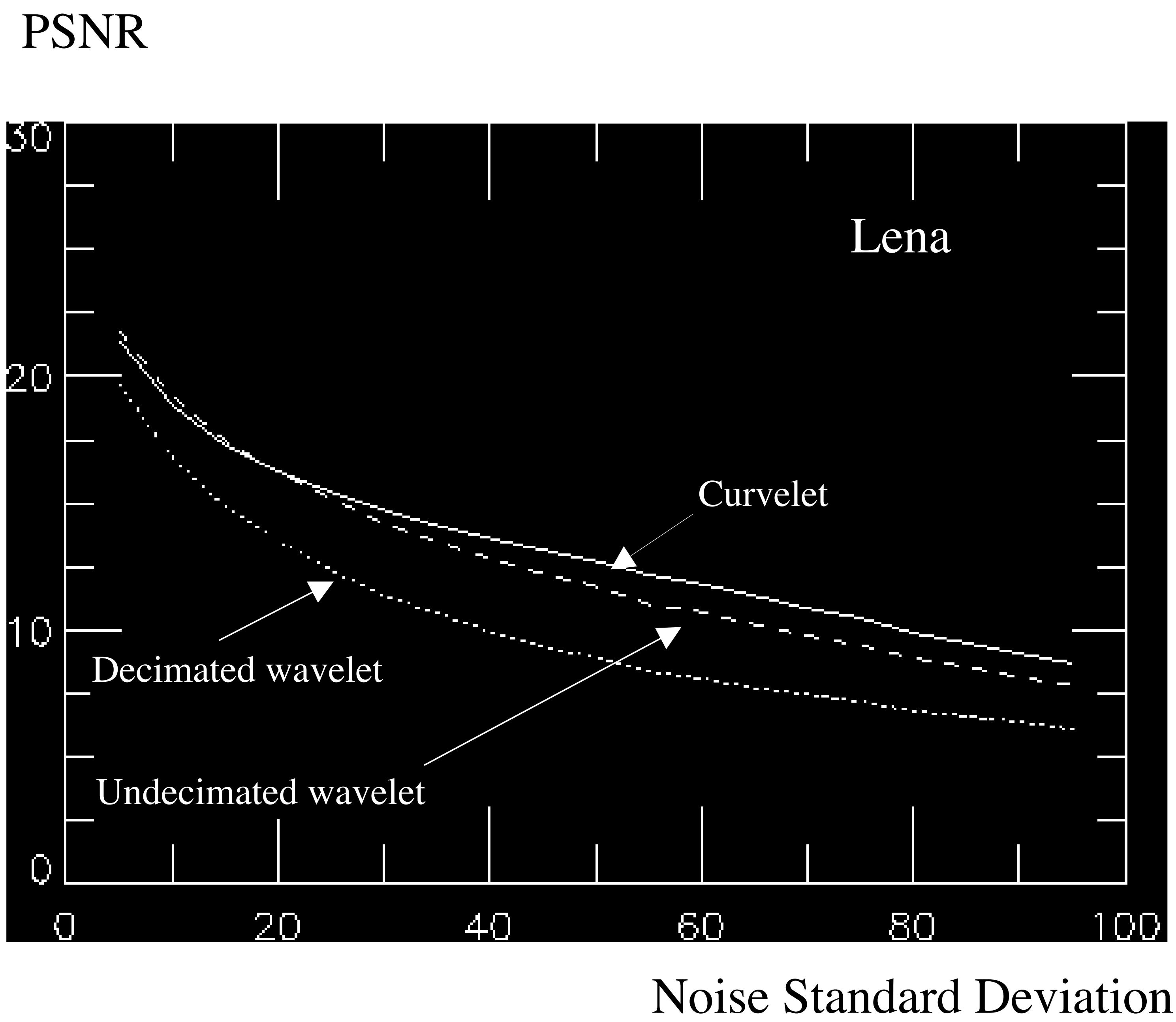
Hard Thresholding:

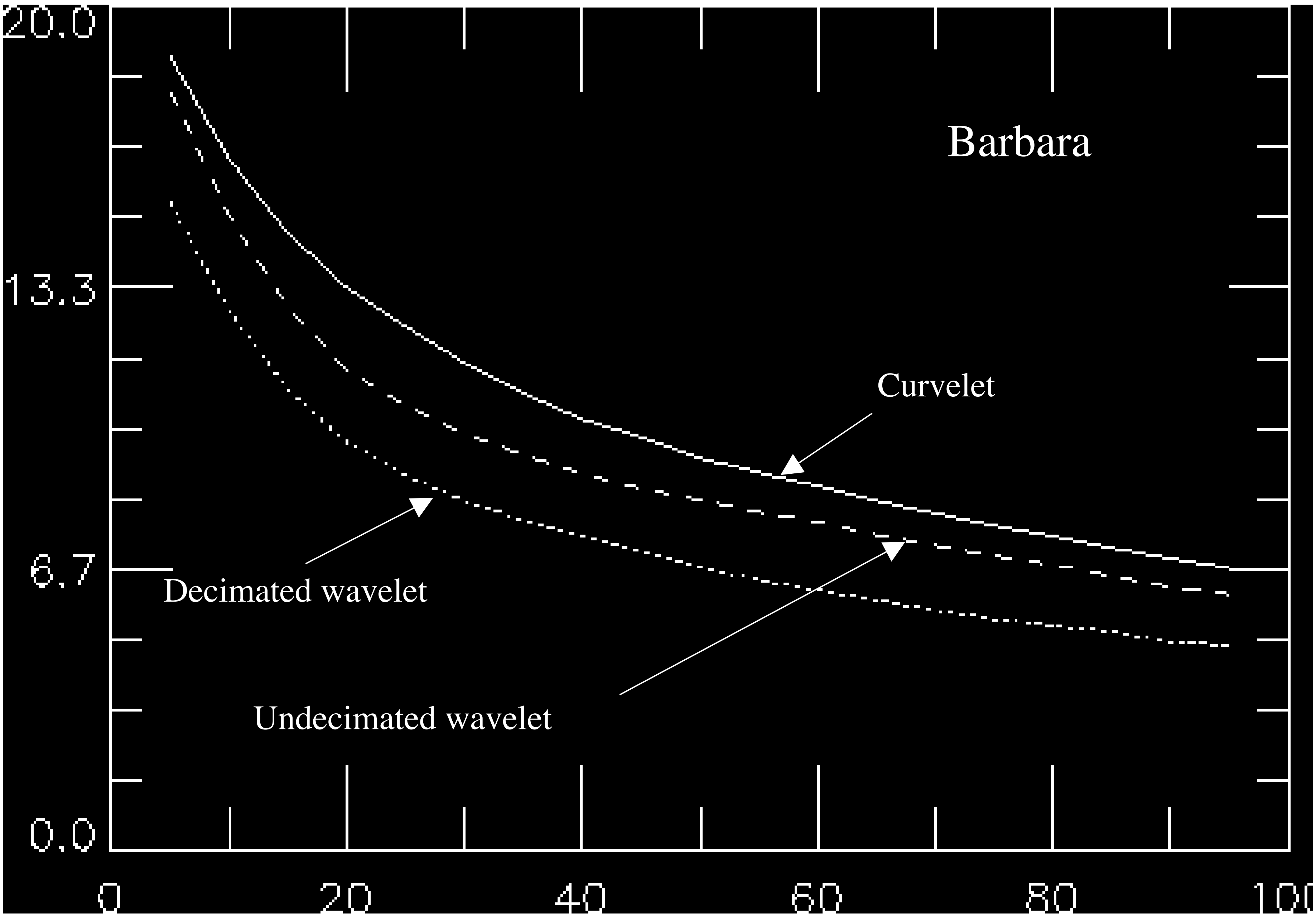
$$\delta(c) = c \qquad if\ |c| \geq t$$
$$= 0 \qquad if\ |c| < t$$




Curvelet




PSNR
30
20
10
0
Lena
Curvelet
Decimated wavelet
Undecimated wavelet
0
20
40
60
80
100
Noise Standard Deviation

20.0
13.3
6.7
0.0
0
20
40
60
80
100
Barbara
Curvelet
Decimated wavelet
Undecimated wavelet