2D and 3D Multiscale Geometric Analysis

J.-L. Starck
Dapnia/SEDI-SAP,
Service d'Astrophysique
CEA-Saclay, France.

jstarck@cea.fr
http://jstarck.free.fr

Part I : New transforms

1 - Introduction: Sparsity

2 - Wavelet: recent results

3 - Multiscale Geometric Analysis (MGA)

New Multiscale Geometric Transforms:

- 2D ridgelet transform
- 2D curvelet transform
- The fast curvelet transform

Part II : Extension to the Sphere and to the third dimension

4 - MGA on the Sphere

5 - 3D Multiscale Geometric Analysis (MGA)

- 3D Ridgelet transform
- 3D Beamlet transform

6 - MGA and the analysis of the spatial distribution of galaxies

Part III - Morphological Component Analysis (MCA)

7 - Component Separation

8 - Multichannel Component Separation

9 - Inpainting

10 - Application to PLANCK

What is a good representation for data?

- Computational harmonic analysis seeks representations of a signal as linear combinations of basis, frame, dictionary, element :

$$f = \sum_k a_k \mathrm{b}_k$$

coefficients ↑ a_k basis, frame ↑ b_k

- Fast calculation of the coefficients a_k
- Analyze the signal through the statistical properties of the coefficients
- Approximation theory uses the sparsity of the coefficients.

What is sparsity ?

In a general framework, a given signal s (n samples) has a unique decomposition α in the orthogonal basis Φ ($n \times n$ matrix).

$$s = \alpha\Phi$$

s is sparse in Φ if most of the entries of α are zeros.

More generally s is sparse in Φ if few entries of α have significant amplitudes.

Seeking sparse and generic representations

- Sparsity

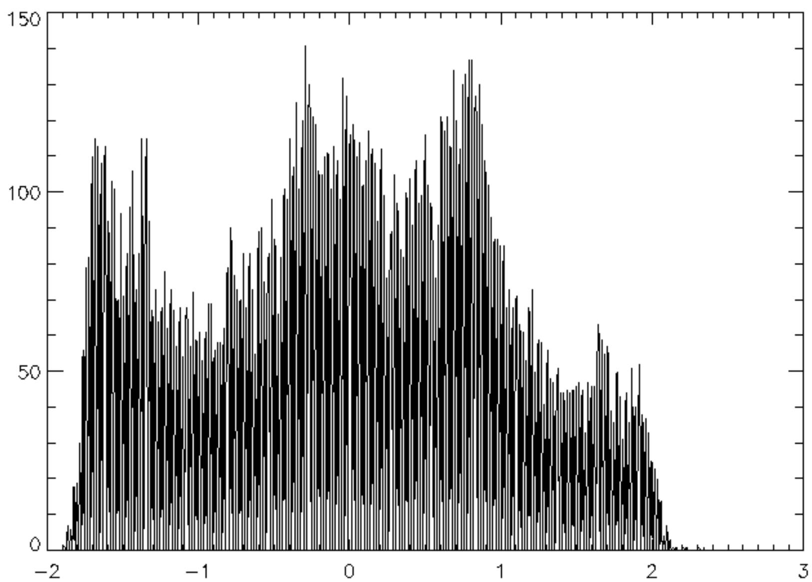
Non-linear approximation curve (reconstruction error versus nbr of coeff)

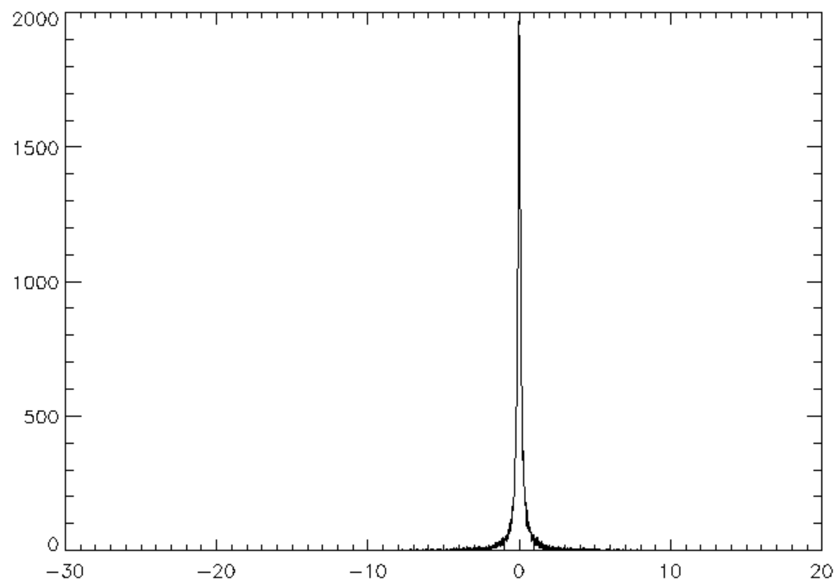
- Why do we need sparsity?
  - data compression
  - Feature extraction, detection
  - Image restoration

Representing Barbara

Direct Space

Curvelet Space

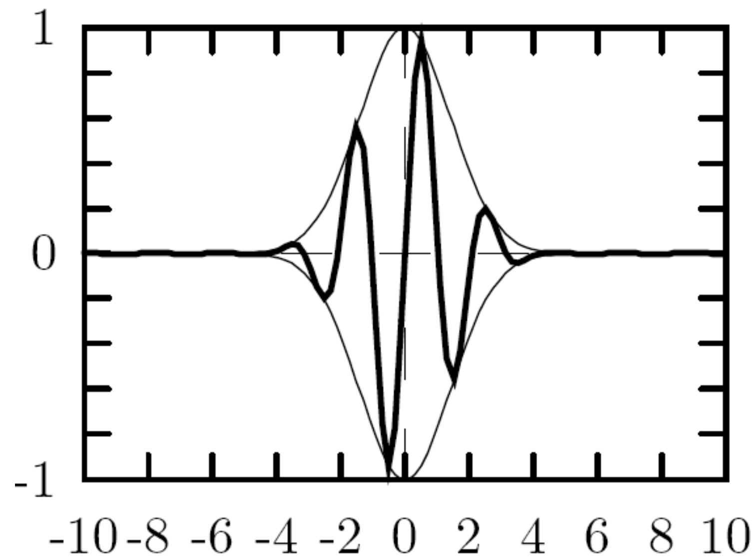




Candidate analyzing functions for piecewise smooth signals

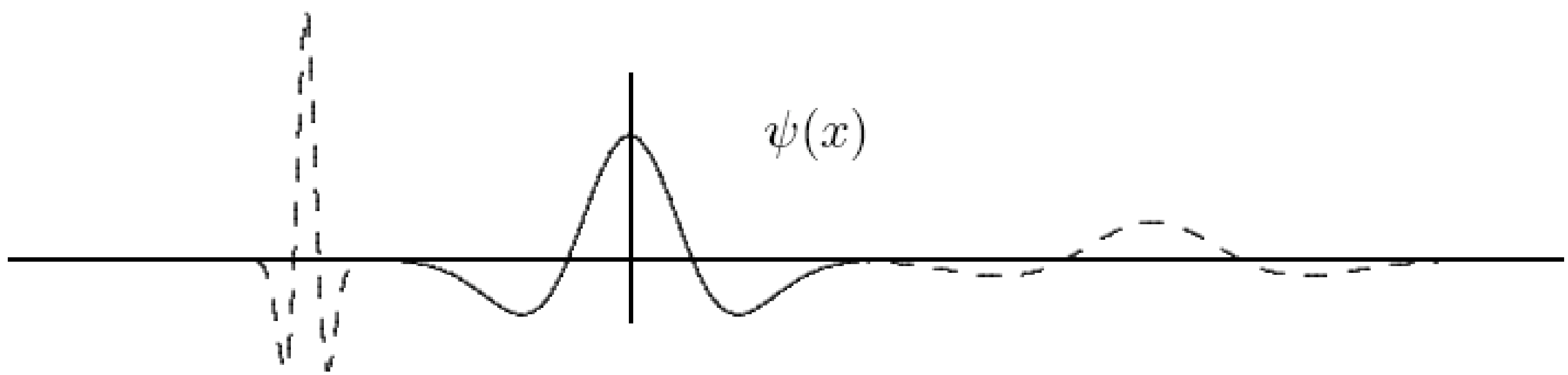
- Windowed fourier transform or Gaborlets :

$$\psi_{\omega,b}(t) = g(t-b)e^{i\omega t}$$



- Wavelets :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{b}}\psi(\frac{t-a}{b})$$



JPEG / JPEG2000

Original
BMP

300x300x24

270056
bytes

JPEG 1:68

3983 bytes

JPEG2000 1:70

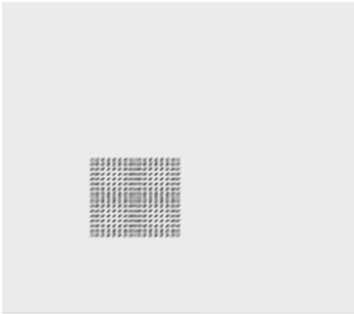
3876 bytes

Looking for adapted representations

Local DCT

Stationary textures

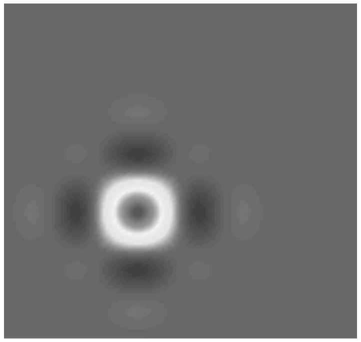
Locally oscillatory



Wavelet transform

Piecewise smooth

Isotropic structures



Curvelet transform

Piecewise smooth,
edge

2D Multiscale Transforms

Critical Sampling

(bi-) Orthogonal WT
Lifting scheme construction
Wavelet Packets
Mirror Basis

Redundant Transforms

Pyramidal decomposition (Burt and Adelson)
Undecimated Wavelet Transform
Isotropic Undecimated Wavelet Transform
Complex Wavelet Transform
Steerable Wavelet Transform
Dyadic Wavelet Transform
Nonlinear Pyramidal decomposition (Median)

New Multiscale Construction

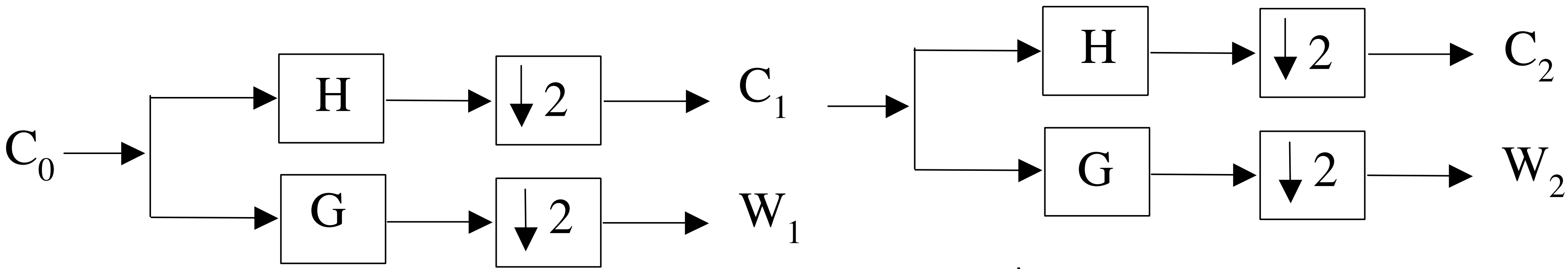
Contourlet
Bandelet
Finite Ridgelet Transform
Platelet
(W-)Edgelet
Adaptive Wavelet

Ridgelet
Curvelet (Several implementations)
Wave Atoms

The Orthogonal Wavelet Transform (OWT)

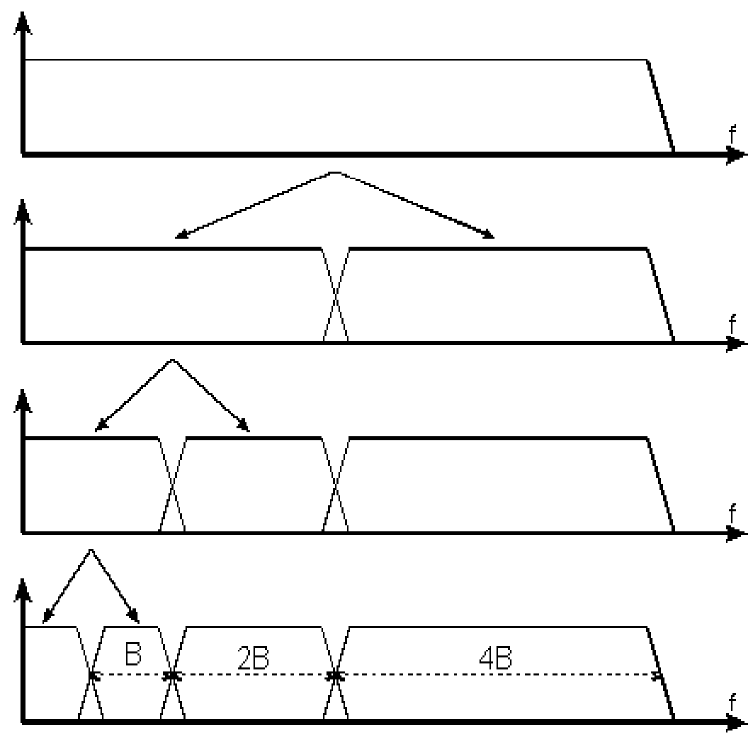
$$s_l = \sum_k c_{J,k}\phi_{J,l}(k) + \sum_k \sum_{j=1}^{J} \psi_{j,l}(k) w_{j,k}$$

Transformation



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

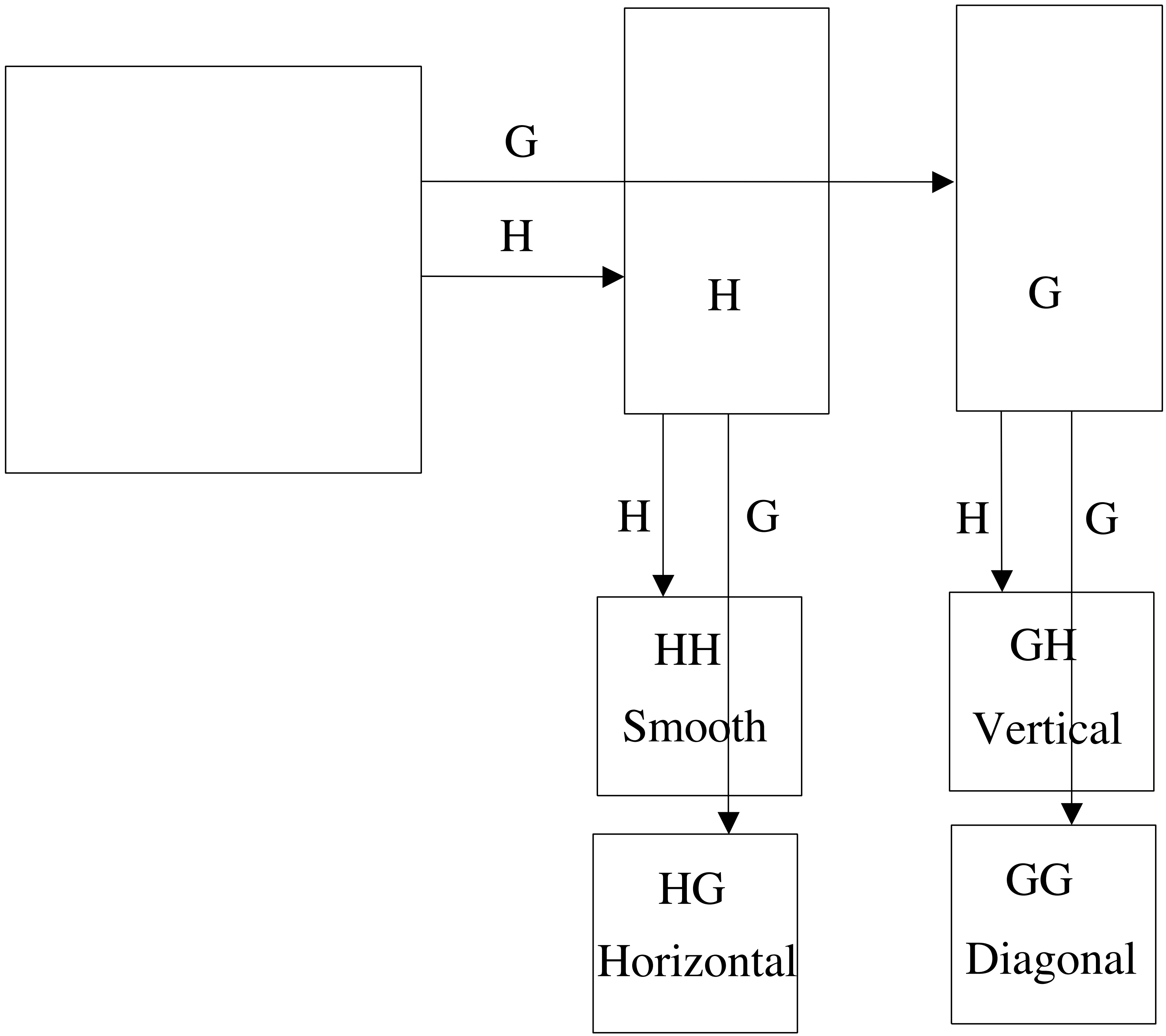
$$w_{j+1,l} = \sum_h g_{k-2l} c_{j,k} = (\bar{g} * c_j)_{2l}$$

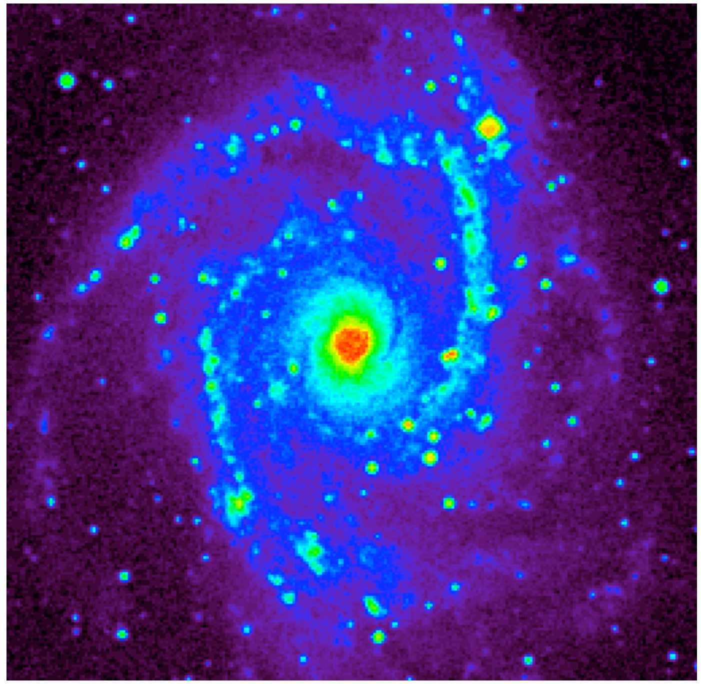


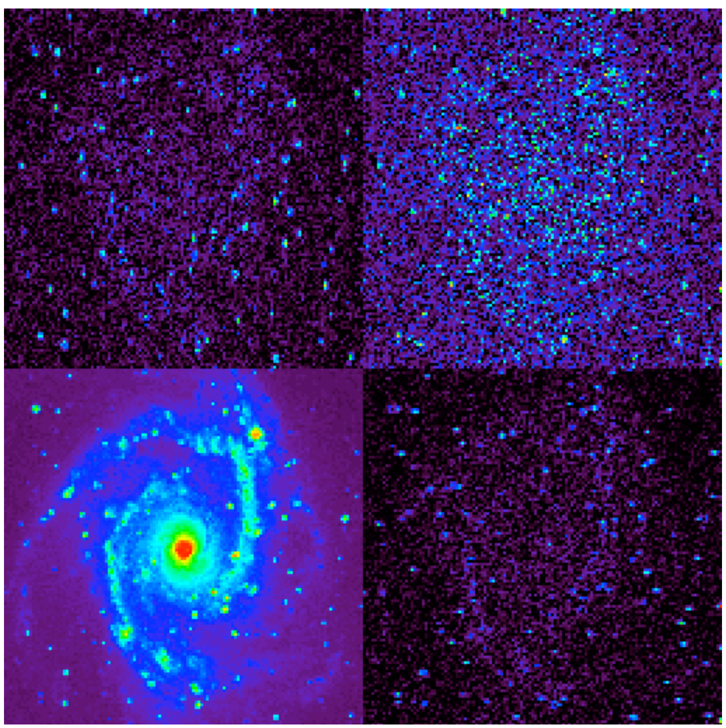
Reconstruction:

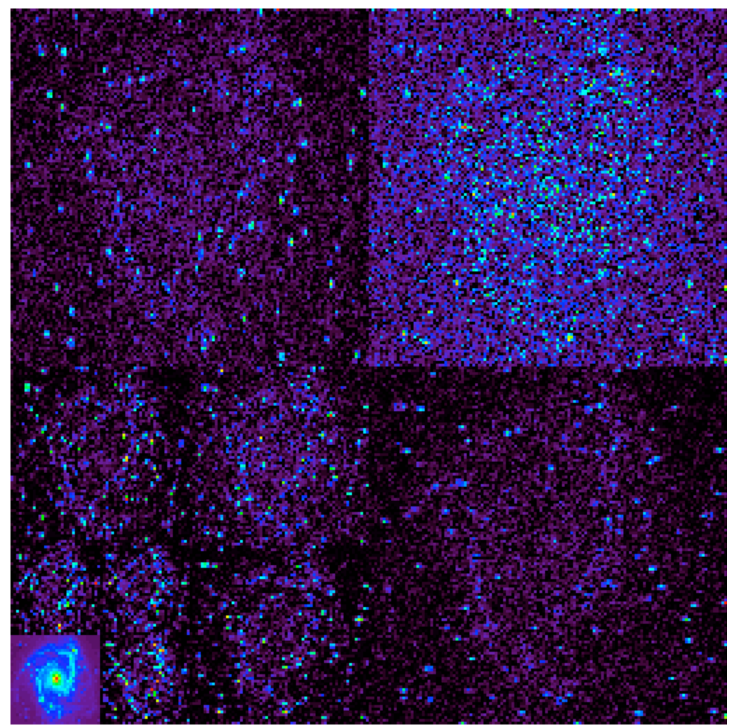
$$c_{j,l} = \sum_k \tilde{h}_{k+2l} c_{j+1,k} + \tilde{g}_{k+2l} w_{j+1,k} = \tilde{h} * \breve{c}_{j+1} + \tilde{g} * \breve{w}_{j+1}$$

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

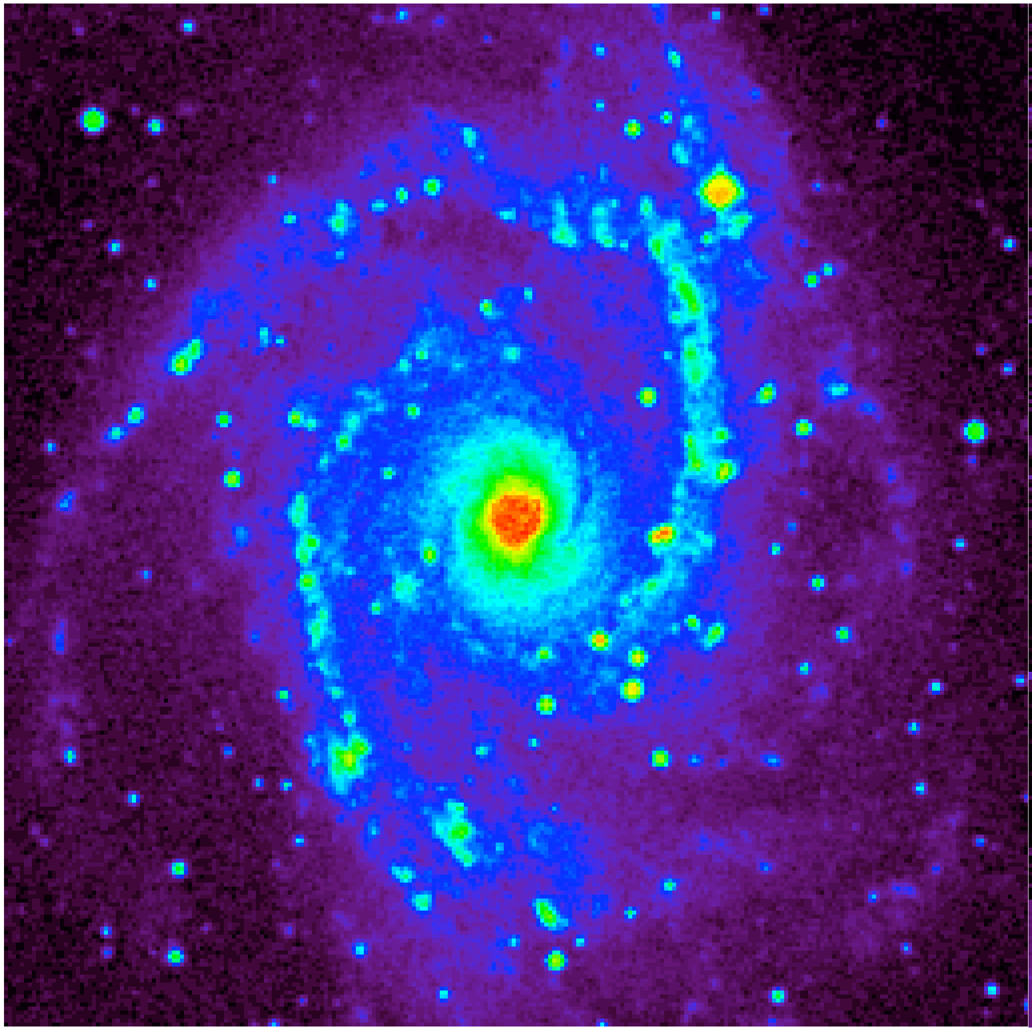

G
H
H
G
H
G
H
G
HH
Smooth
GH
Vertical
HG
Horizontal
GG
Diagonal



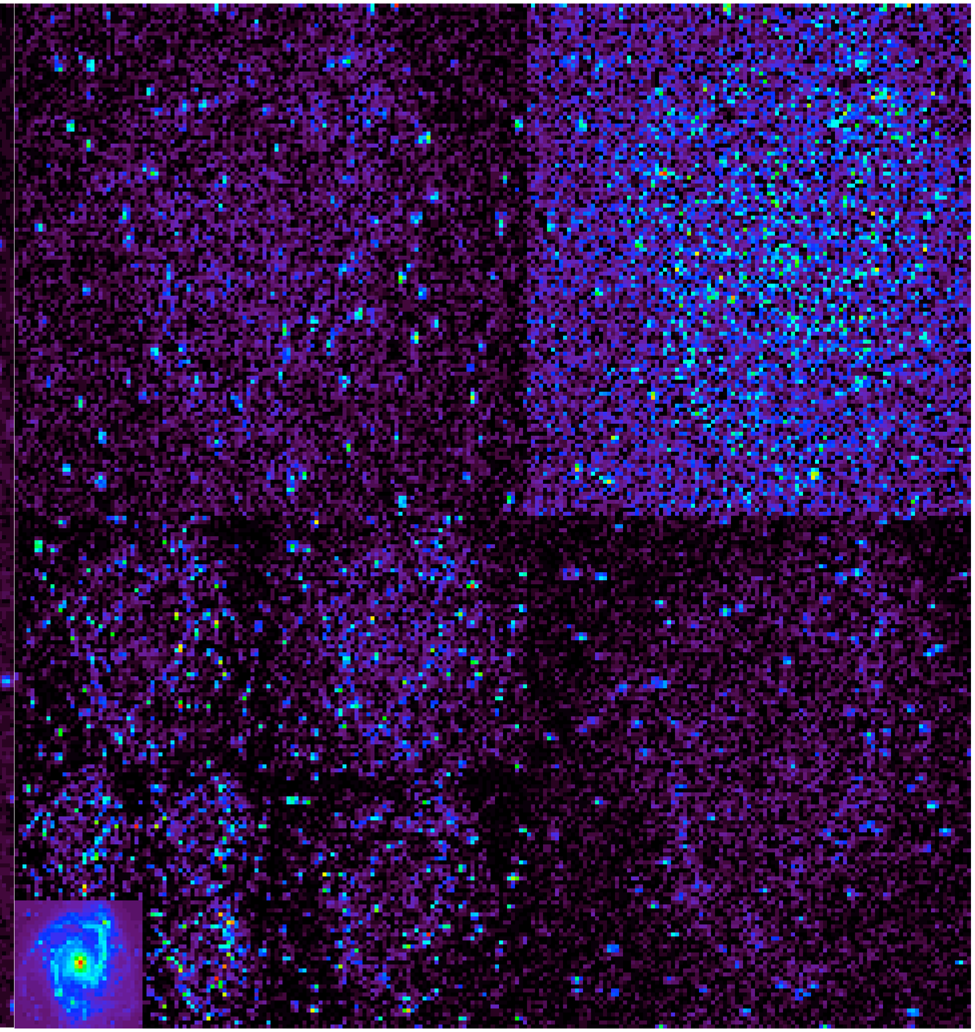




NGC2997



NGC2997 WT



The Filter Bank

In order to get an exact reconstruction, two conditions are required for the filters:

- *Dealiasing condition*: $\hat{h}(\nu + \frac{1}{2})\hat{\tilde{h}}(\nu) + \hat{g}(\nu + \frac{1}{2})\hat{\tilde{g}}(\nu) = 0$
- *Exact restoration:* $\hat{h}(\nu)\hat{\tilde{h}}(\nu) + \hat{g}(\nu)\hat{\tilde{g}}(\nu) = 1$

The Isotropic Undecimated Wavelet Transform

- Filters do not need to verify the dealiasing condition. We need only the exact restoration condition:

$$\hat{h}(\nu)\hat{\tilde{h}}(\nu) + \hat{g}(\nu)\hat{\tilde{g}}(\nu) = 1$$

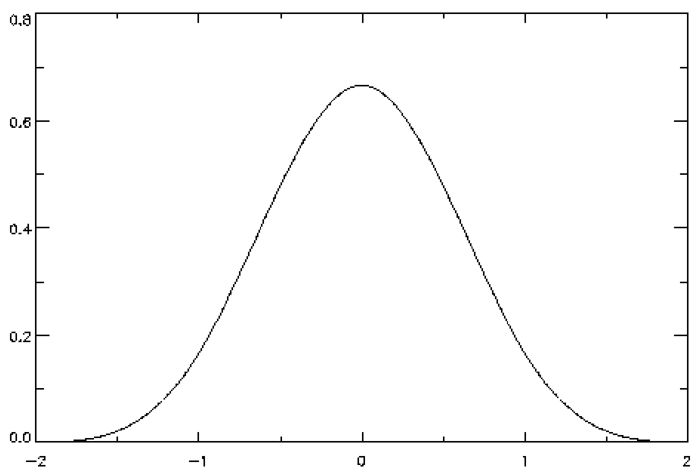
- Filters do not need to be (bi) orthogonal.
- Filters must be symmetric.
- In 2D, we want $h(x,y) = h(x)h(y)$ for fast calculation and more important, $h(x,y)$ must nearly isotropic.

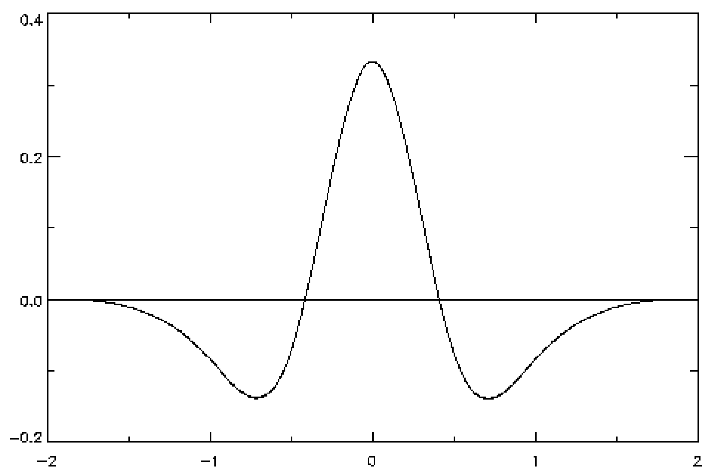
h is derived from a B_3 spline: $h_{1D}(k) = [1,4,6,4,1]/16$, and in 2D $h_{2D} = h_{1D}h_{1D} =$

$$\begin{pmatrix} \frac{1}{16} & \frac{1}{4} & \frac{3}{8} & \frac{1}{4} & \frac{1}{16} \end{pmatrix} \otimes \begin{pmatrix} 1/16 \\ 1/4 \\ 3/8 \\ 1/4 \\ 1/16 \end{pmatrix} = \begin{pmatrix} \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{3}{128} & \frac{3}{32} & \frac{9}{64} & \frac{3}{32} & \frac{3}{128} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \end{pmatrix}$$

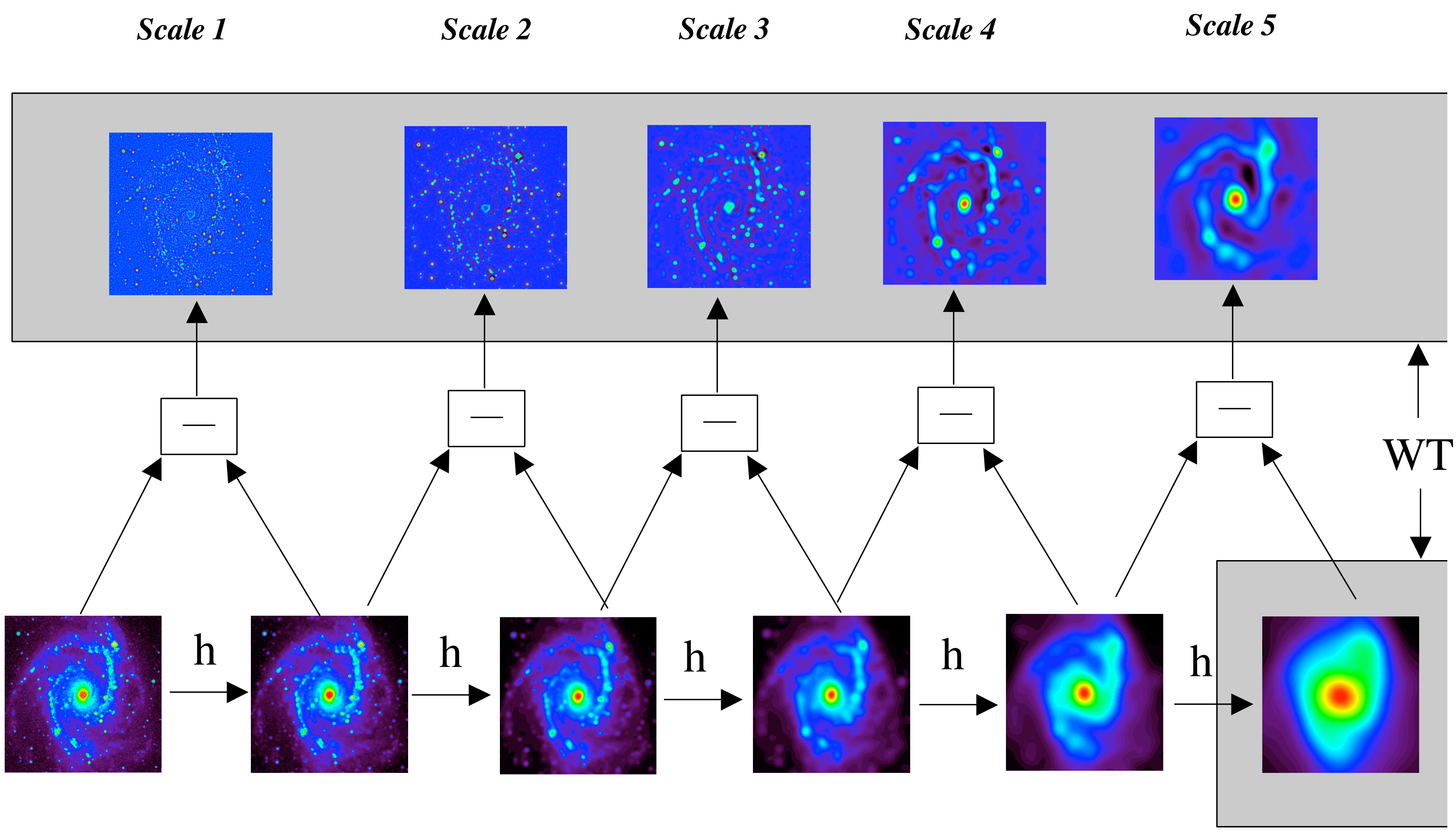
The Isotropic Wavelet and Scaling Functions

$$
\begin{aligned}
B_3(x) &= \frac{1}{12}(\mid x-2 \mid^3 -4 \mid x-1 \mid^3 +6 \mid x \mid^3 -4 \mid x+1 \mid^3 + \mid x+2 \mid^3) \\
\psi(x,y) &= B_3(x)B_3(y) \\
\frac{1}{4}\psi(\frac{x}{2},\frac{y}{2}) &= \phi(x,y) - \frac{1}{4}\phi(\frac{x}{2},\frac{y}{2})
\end{aligned}
$$





ISOTROPIC UNDECIMATED WAVELET TRANSFORM

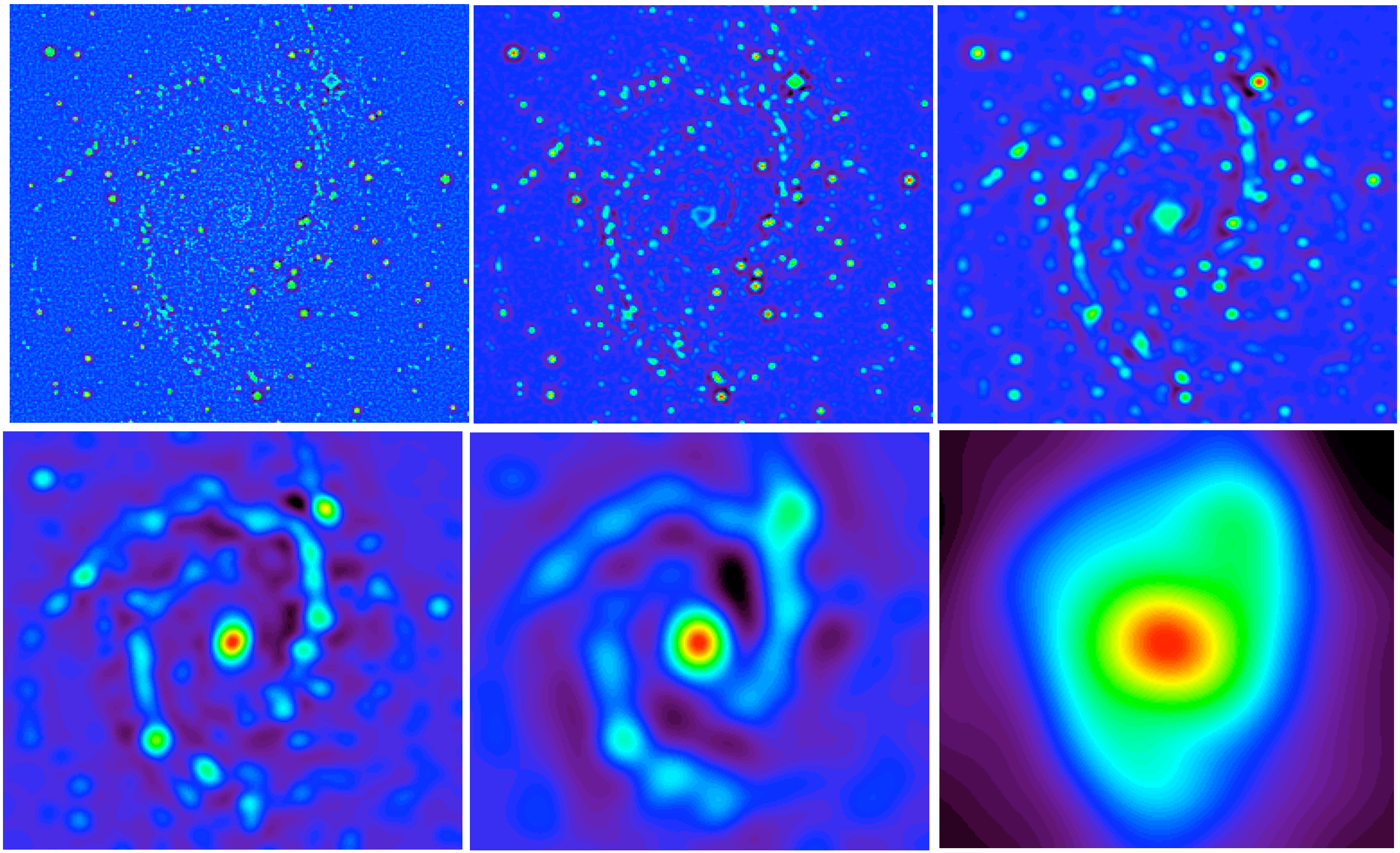


Isotropic Undecimated Wavelet Transform

$$\varphi = B_3 - \text{spline}, \quad \frac{1}{2}\psi(\frac{x}{2}) = \frac{1}{2}\varphi(\frac{x}{2}) - \varphi(x)$$

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

$$h = [1,4,6,4,1]/16, \quad g = \delta - h, \quad \tilde{h} = \tilde{g} = \delta$$



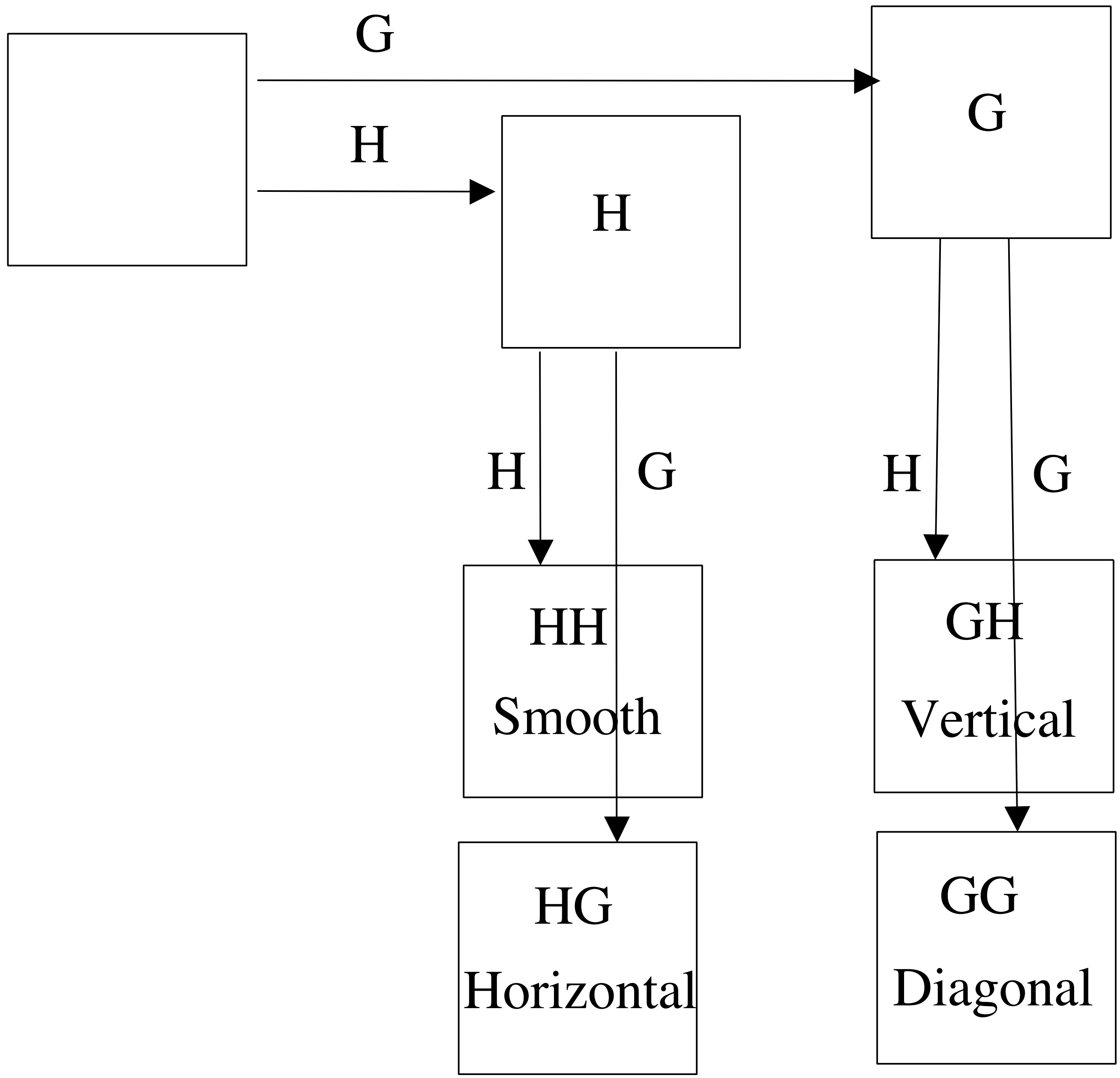
2D Undecimated Wavelet Transform

The à trous algorithm can be extended to 2D:

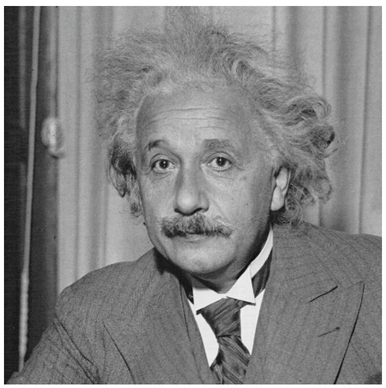
$$
\begin{aligned}
c_{j+1,k,l} &= (\bar{h}^{(j)}\bar{h}^{(j)} * c_j)_{k,l} \\
w_{j+1,1,k,l} &= (\bar{g}^{(j)}\bar{h}^{(j)} * c_j)_{k,l} \\
w_{j+1,2,k,l} &= (\bar{h}^{(j)}\bar{g}^{(j)} * c_j)_{k,l} \\
w_{j+1,3,k,l} &= (\bar{g}^{(j)}\bar{g}^{(j)} * c_j)_{k,l}
\end{aligned}
$$

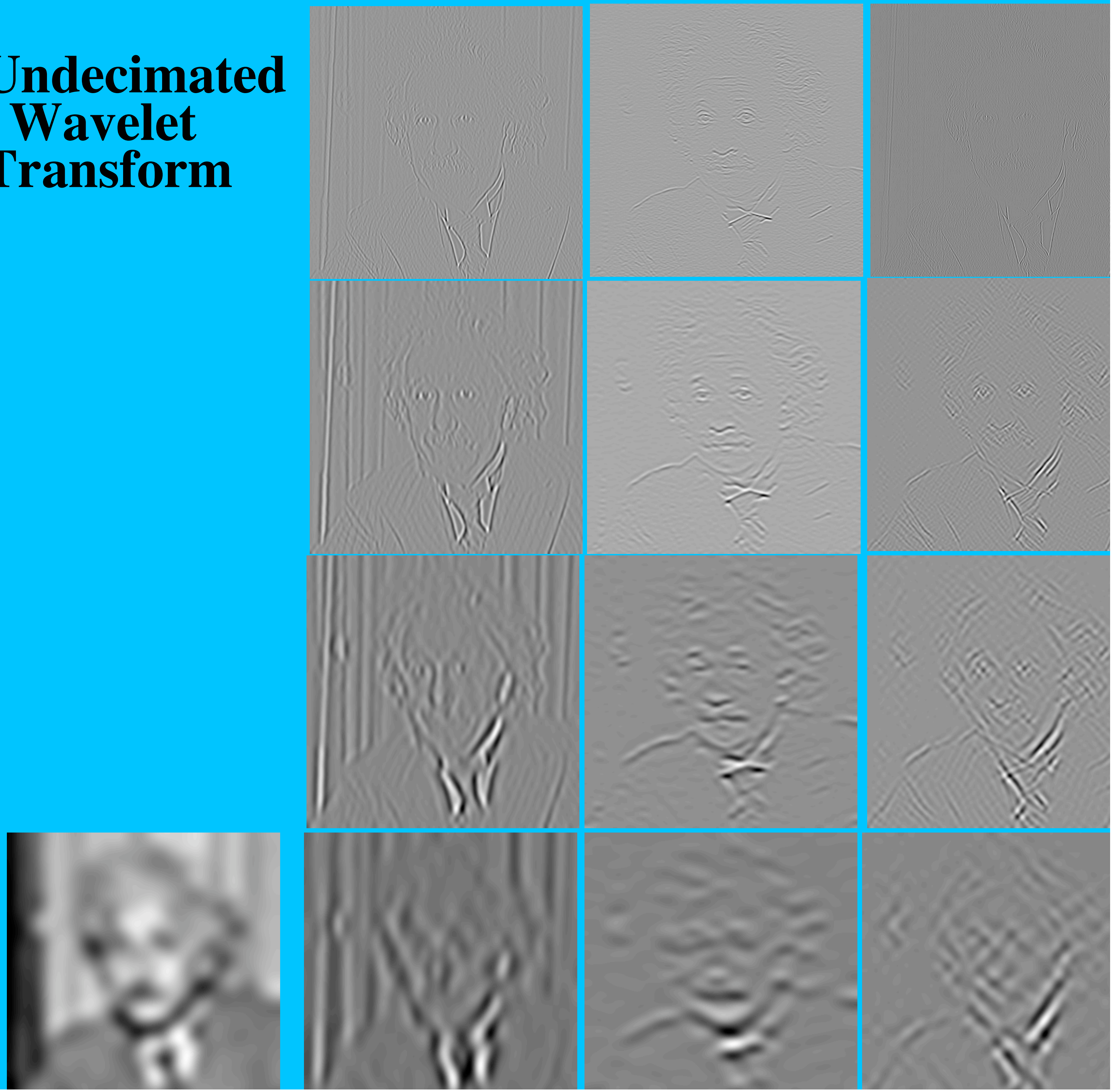
where $hg * c$ is the convolution of c by the separable filter hg (i.e convolution first along the columns per h and then convolution along the lines per g).

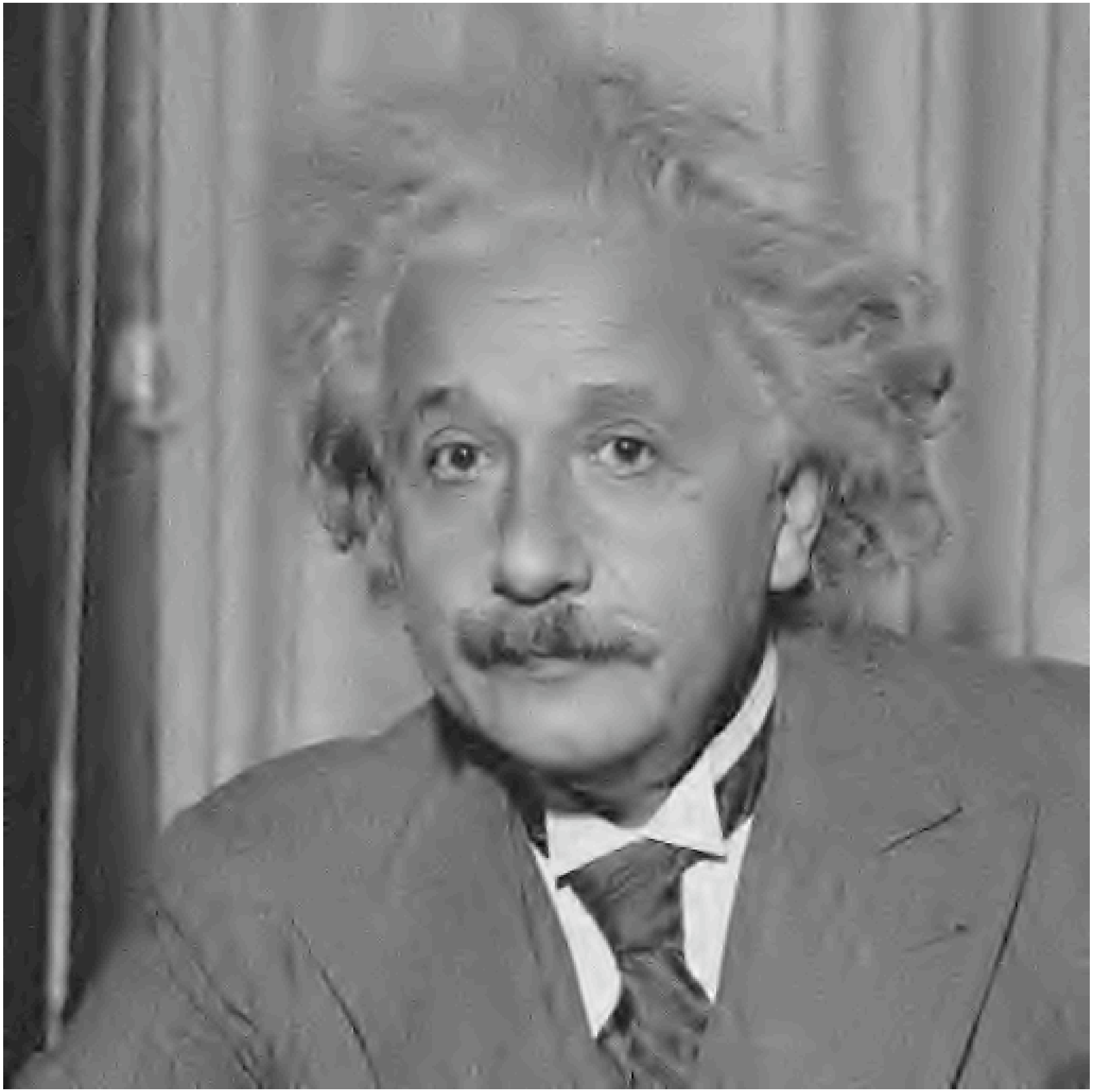
Undecimated bi-orthogonal Wavelet Transform



Undecimated Wavelet Transform







Non (bi-) Orthogonal Directional Undecimated WT using the "astro" filter bank

$$
\begin{aligned}
h &= [1,4,6,4,1]/16 \\
g &= Id - h = [-1,-4,10,-4,-1]/16 \\
\tilde{h} &= \tilde{g} = Id
\end{aligned}
$$

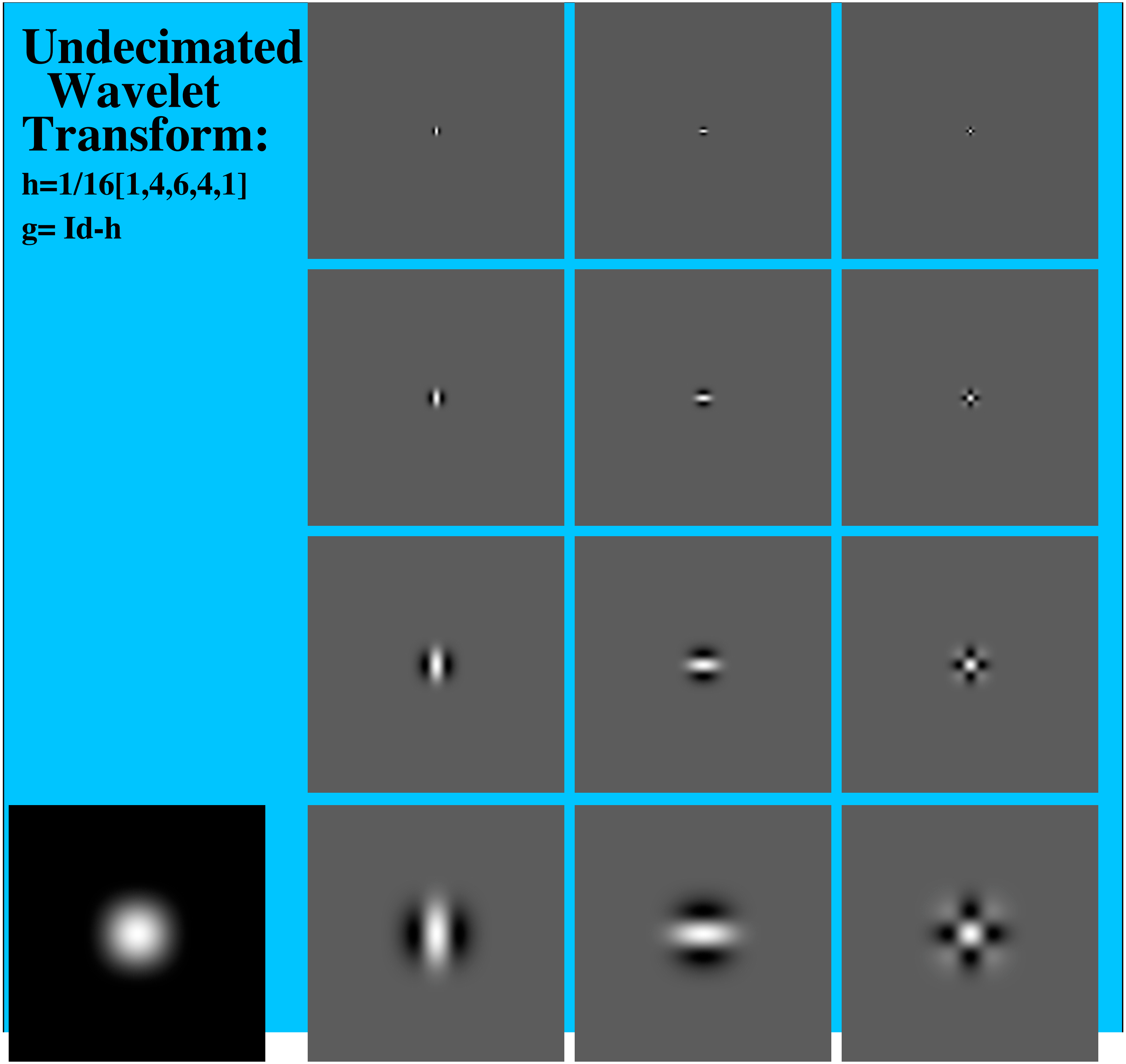
In two dimensions, the detail signal is contained in three sub-images

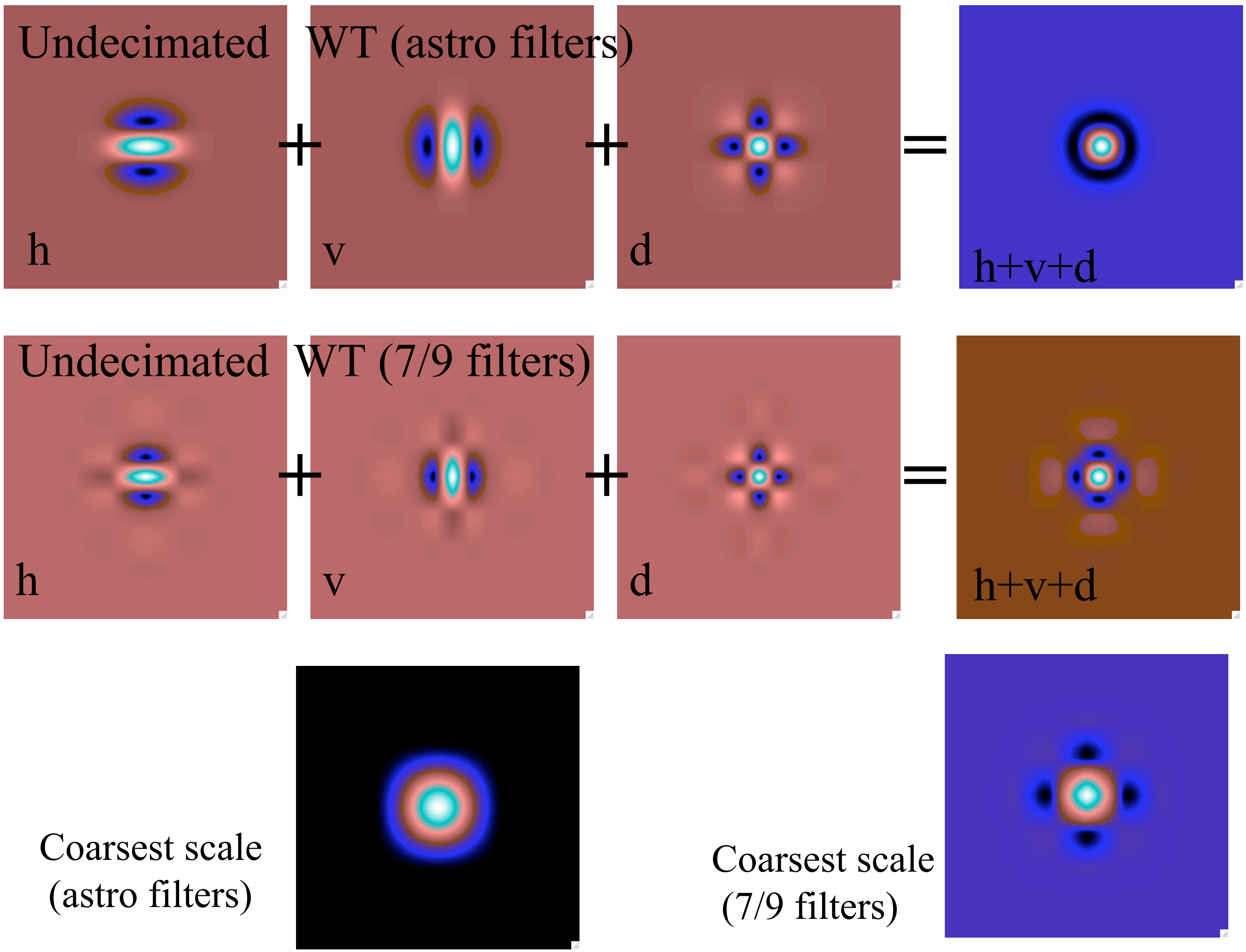
$$
\begin{aligned}
w_j^1(k_x,k_y) &= \sum_{l_x=-\infty}^{+\infty} \sum_{l_y=-\infty}^{+\infty} g(l_x - 2k_x) h(l_y - 2k_y) c_{j+1}(l_x,l_y) \\
w_j^2(k_x,k_y) &= \sum_{l_x=-\infty}^{+\infty} \sum_{l_y=-\infty}^{+\infty} h(l_x - 2k_x) g(l_y - 2k_y) c_{j+1}(l_x,l_y) \\
w_j^3(k_x,k_y) &= \sum_{l_x=-\infty}^{+\infty} \sum_{l_y=-\infty}^{+\infty} g(l_x - 2k_x) g(l_y - 2k_y) c_{j+1}(l_x,l_y)
\end{aligned}
$$

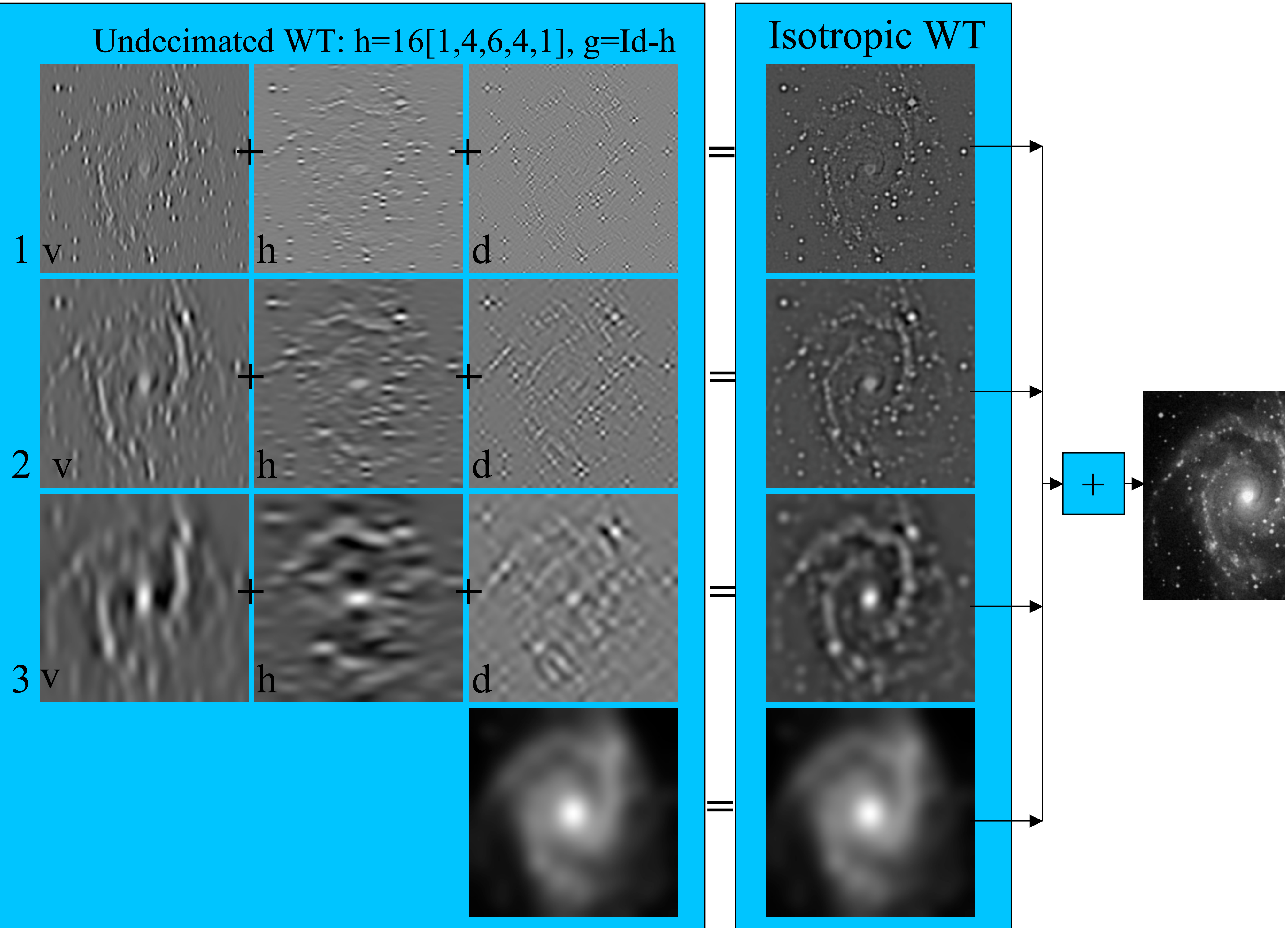
Undecimated Wavelet Transform:

h=1/16[1,4,6,4,1]

g= Id-h




Undecimated WT (astro filters)
h
+
v
+
d
=
h+v+d
Undecimated WT (7/9 filters)
h
+
v
+
d
=
h+v+d
Coarsest scale
(astro filters)
Coarsest scale
(7/9 filters)


Undecimated WT: h=16[1,4,6,4,1], g=Id-h
Isotropic WT
1 v
h
d
2 v
h
d
3 v
h
d
+
=

The Surprise

Because the decomposition is redundant, there are many way to reconstruct the original image from its wavelet transform. For a given (h,g) filter bank, any filter bank ($\tilde{h}$,$\tilde{g}$) which verifies the equation $\hat{h}(\nu)\hat{\tilde{h}}(\nu) + \hat{g}(\nu)\hat{\tilde{g}}(\nu) = 1$ leads to an exact reconstruction. For instance, if we choose $\tilde{h} = h$ (the synthesis scaling function $\tilde{\phi} = \phi$) we obtain a filter $\tilde{g}$ defined by:

$$\tilde{g} = h + Id$$

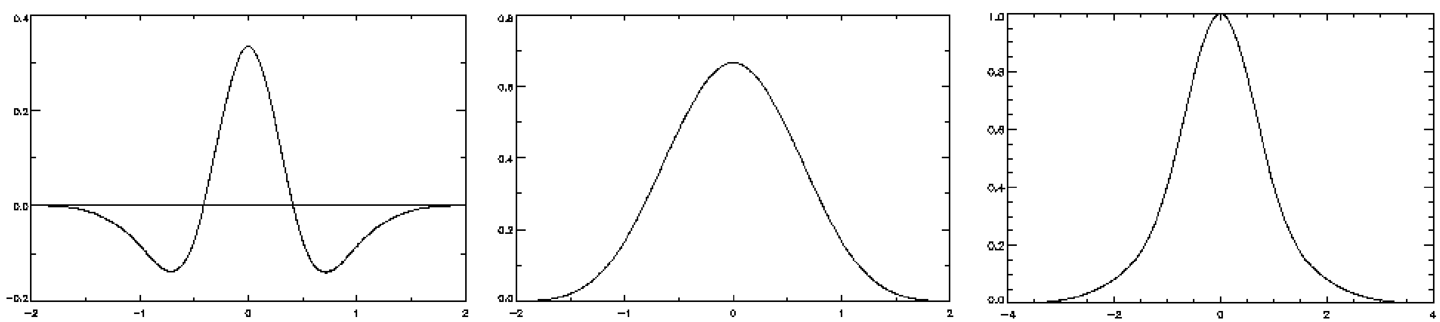
if $h = [1, 4, 6, 4, 1]/16$, then $g = [1, 4, 22, 4, 1]/16$. g **is positive**. This means that g is not related anymore to a wavelet function. The synthesis scaling function related to $\tilde{g}$ is defined by:

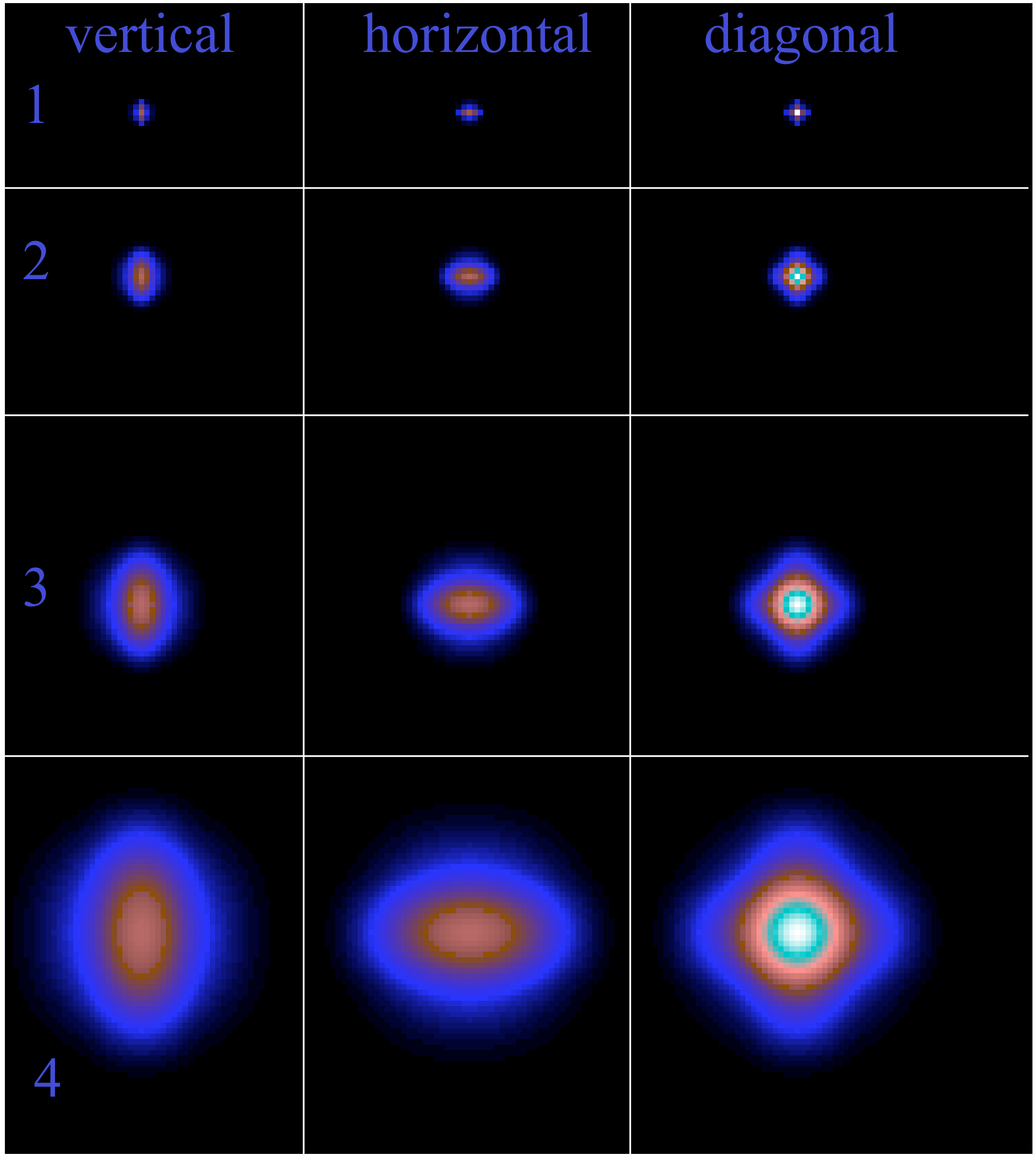
$$\frac{1}{2}\tilde{\phi}(\frac{x}{2}) \quad = \quad \phi(x) + \frac{1}{2}\phi(\frac{x}{2})$$

J.-L. Starck, J. Fadili and F. Murtagh, "The Undecimated Wavelet Decomposition and its Reconstruction", *IEEE Transaction on Image Processing*, in press.

Reconstruction Using the Scaling Function

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




vertical
horizontal
diagonal
1
2
3
4

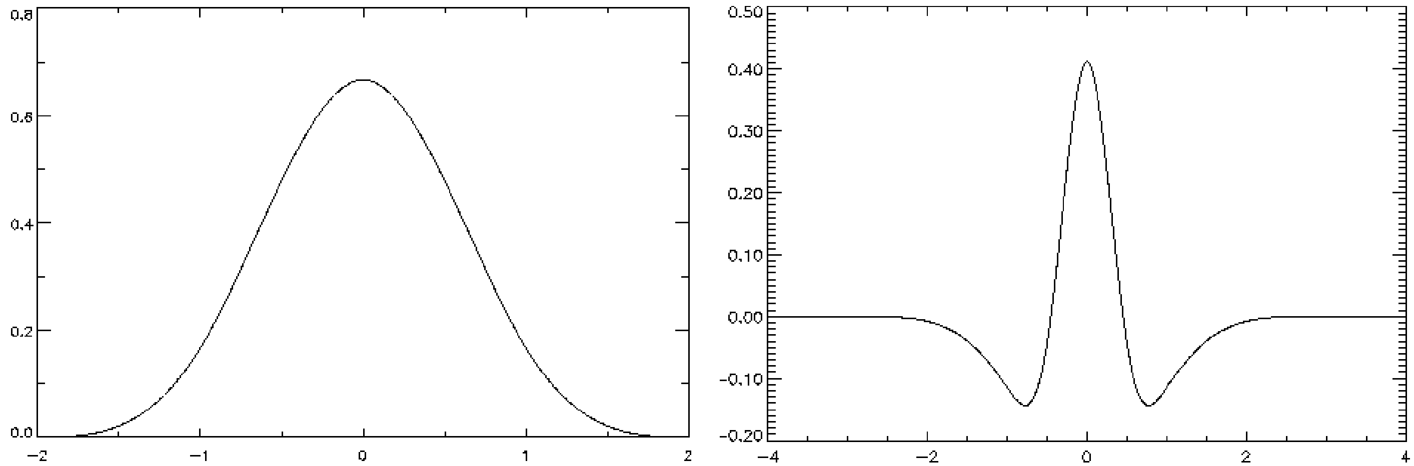
Another Interesting Filter Bank

Deriving h from a spline scaling function, for instance $B_2 = [1,2,1]/2$ or $B_3 = [1,4,6,4,1]/16$ (note that $B_3 = B_2 * B_2$, we define the following the filter bank:

$$\begin{aligned} h &= \tilde{h} = B_l \\ g &= Id - h * h \\ \tilde{g} &= Id \end{aligned}$$

which leads to an analysis/synthesis with the following functions:

$$\begin{aligned} \phi(x) &= \tilde{\phi}(x) = B_l(x) \\ \hat{\psi}(\nu) &= \frac{\hat{\phi}^2(\nu) - \hat{\phi}^2(2\nu)}{\hat{\phi}(\nu)} \\ \frac{1}{2}\tilde{\psi}(\frac{x}{2}) &= \phi(x) \end{aligned}$$



$$\phi(x) = \tilde{\phi}(x) = B_3(x)$$

$$\hat{\psi}(\nu) = \frac{\hat{\phi}^2(\nu) - \hat{\phi}^2(2\nu)}{\hat{\phi}(\nu)}$$

$$\frac{1}{2}\tilde{\psi}(\frac{x}{2}) = \phi(x)$$

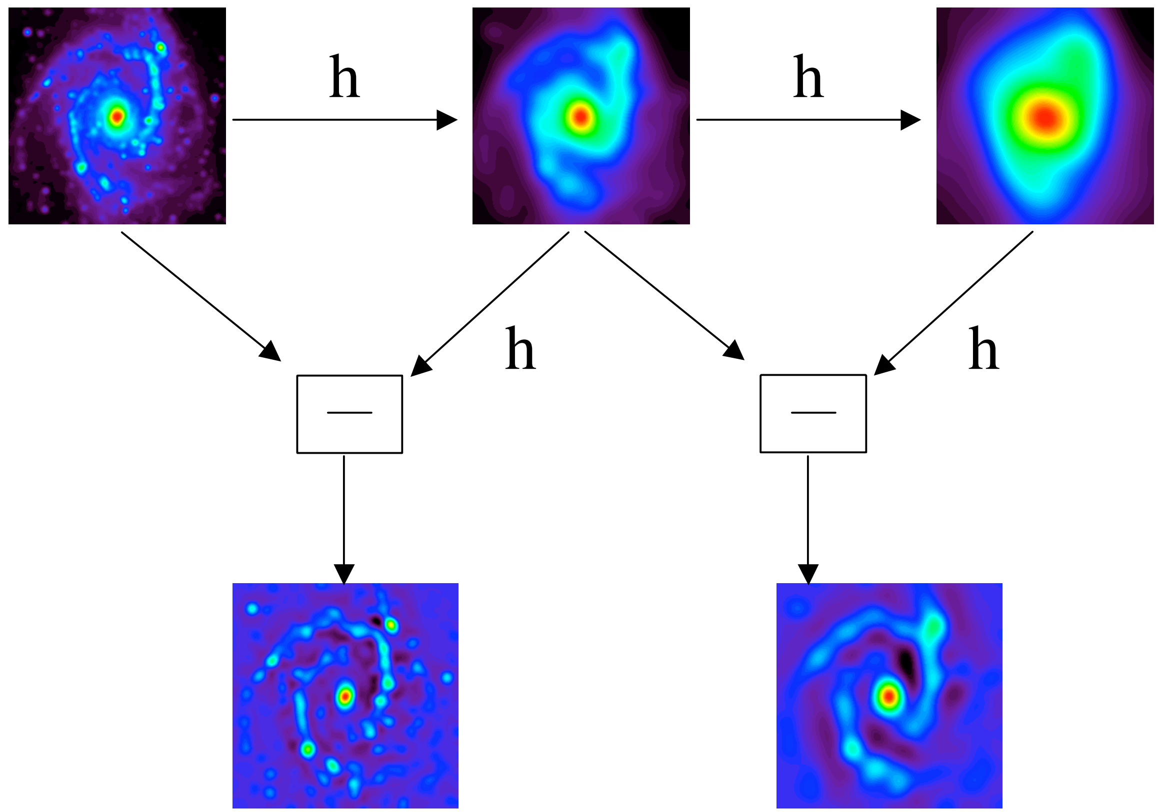
$$h = \tilde{h} = [1, 4, 6, 4, 1]/16$$

$$g = Id - h * h = [1, 8, 28, 56, 70, 56, 28, 8, 1]/256$$

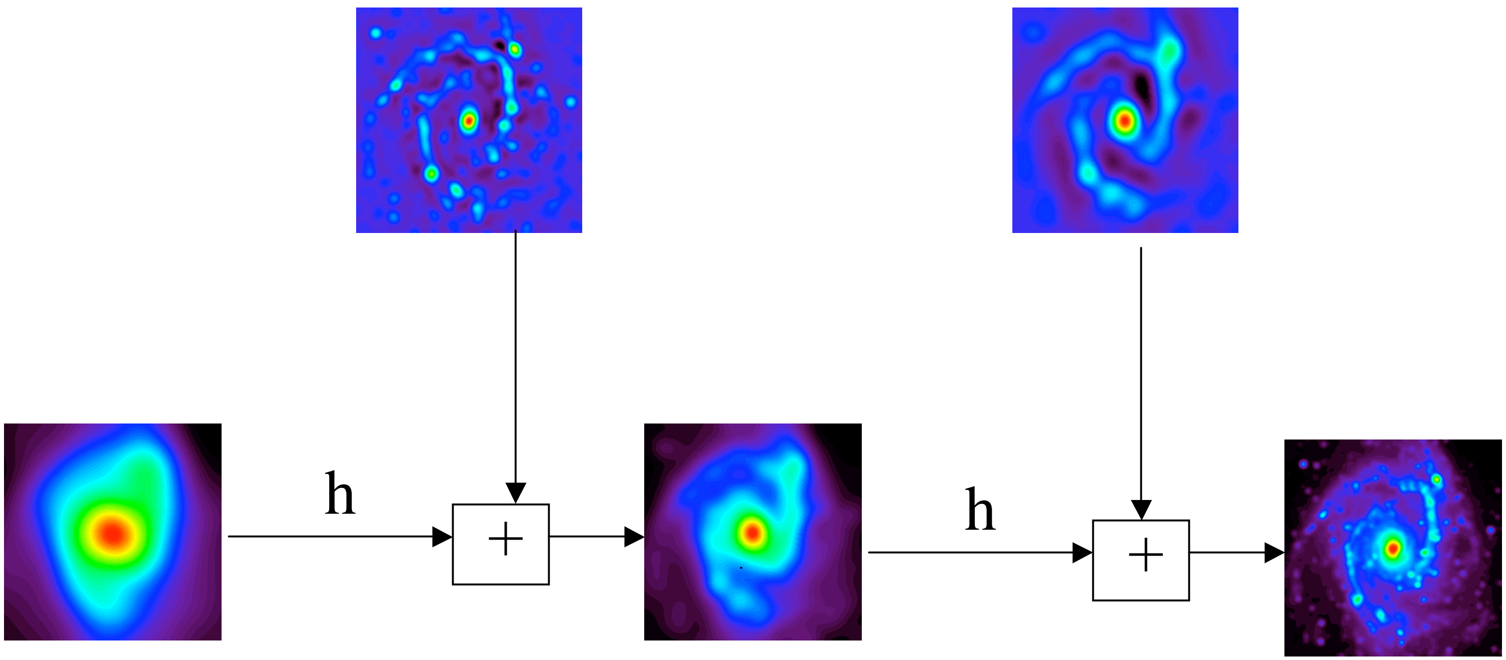
$$\tilde{g} = Id$$

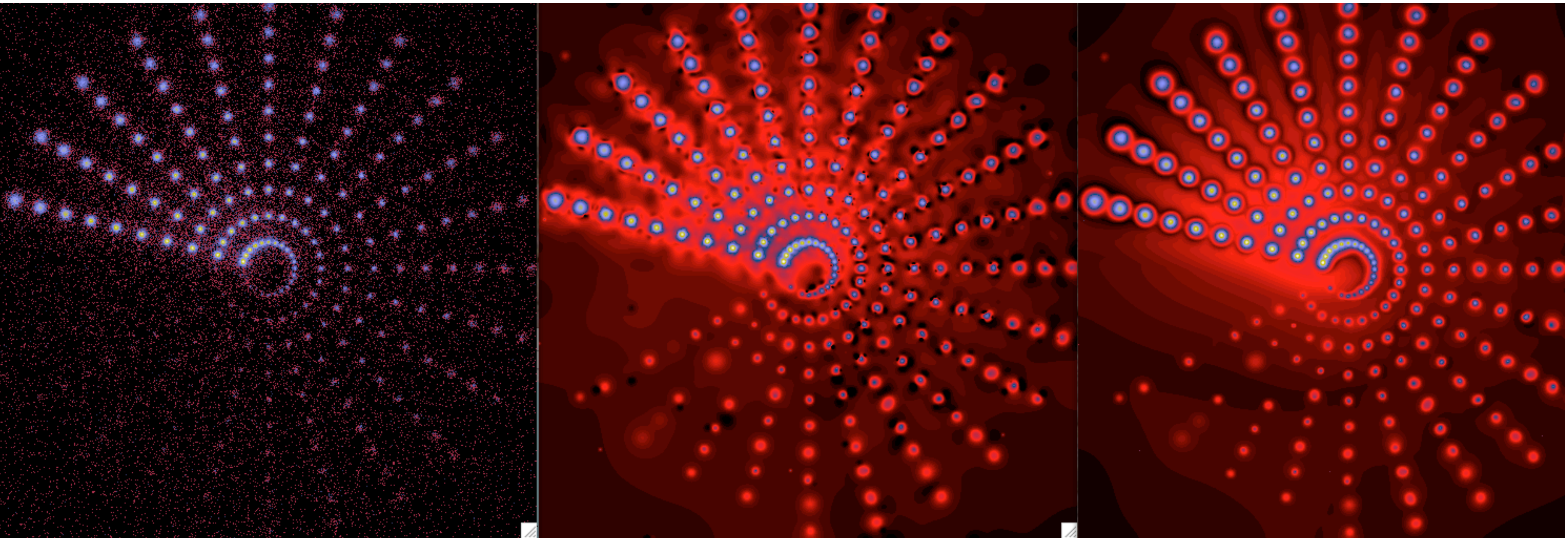
MODIFIED ISOTROPIC UNDECIMATED WT

$$h = h_{1d} \# h_{1d}, \quad g = Id - h * h$$



RECONSTRUCTION





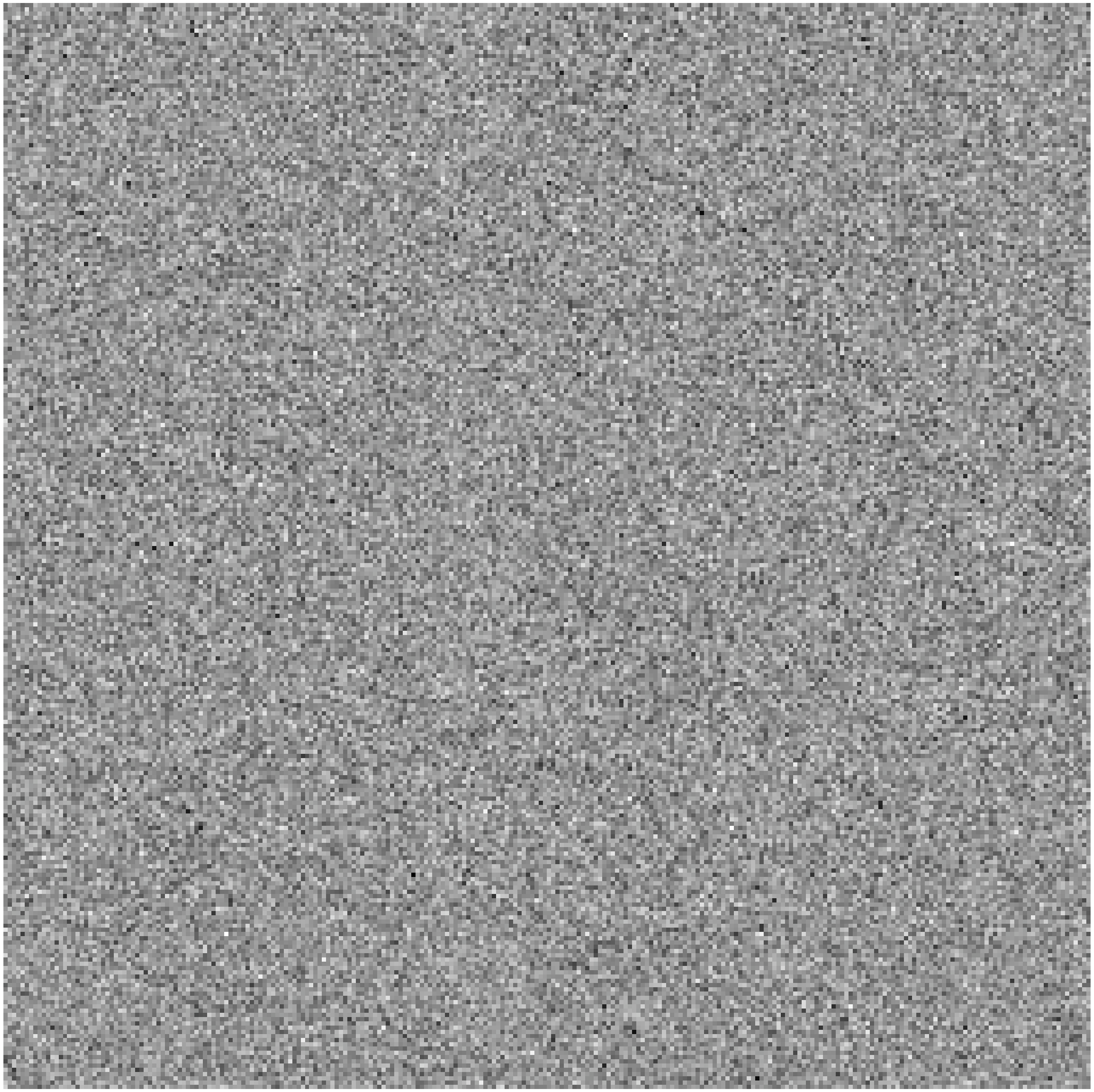
Problems related to the WT

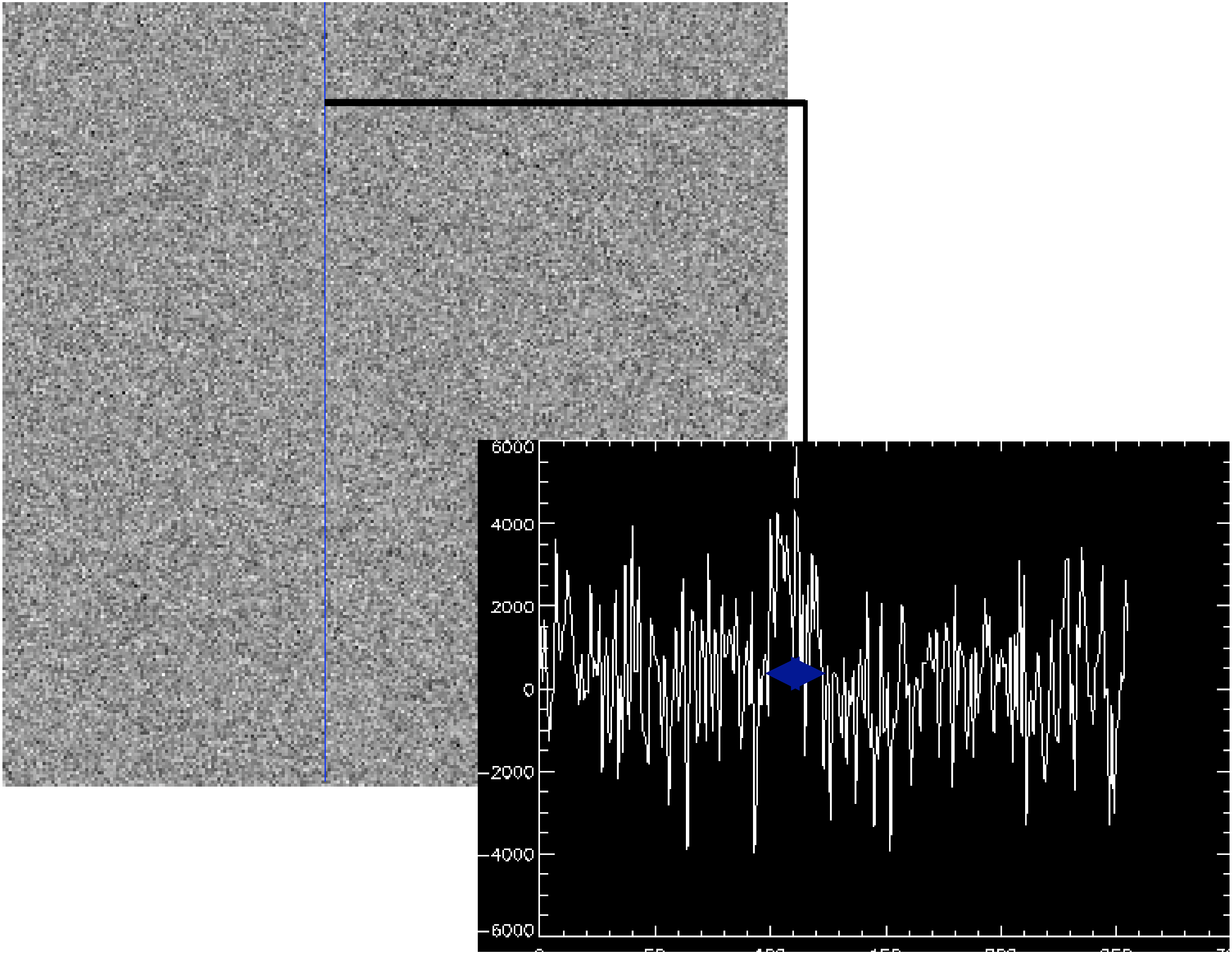
1) Edges representation:
if the WT performs better than the FFT to
represent edges in an image, it is still not optimal.

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

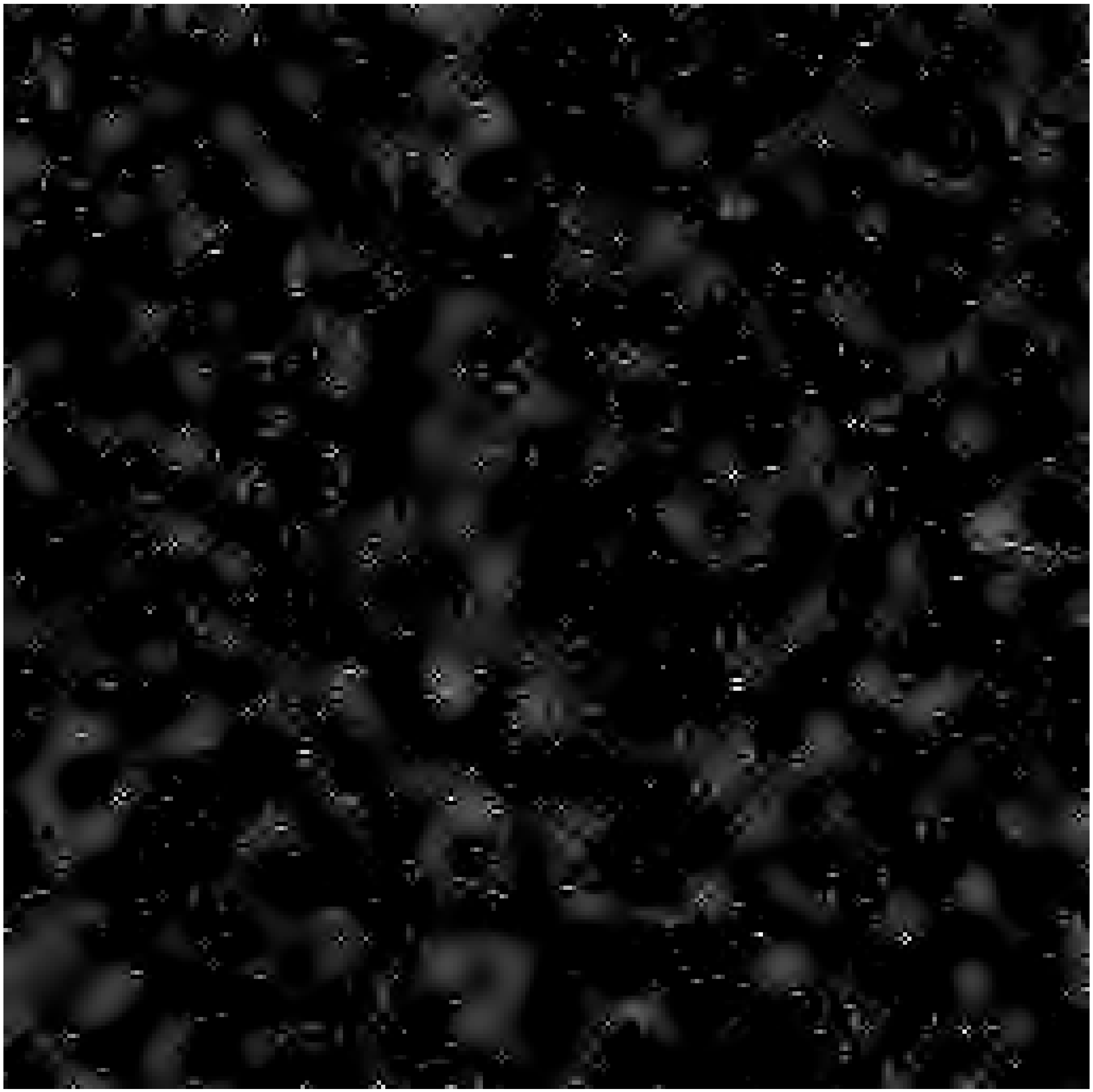
3) Limitation of existing scale concepts:
there is no highly anisotropic elements.

SNR = 0.1

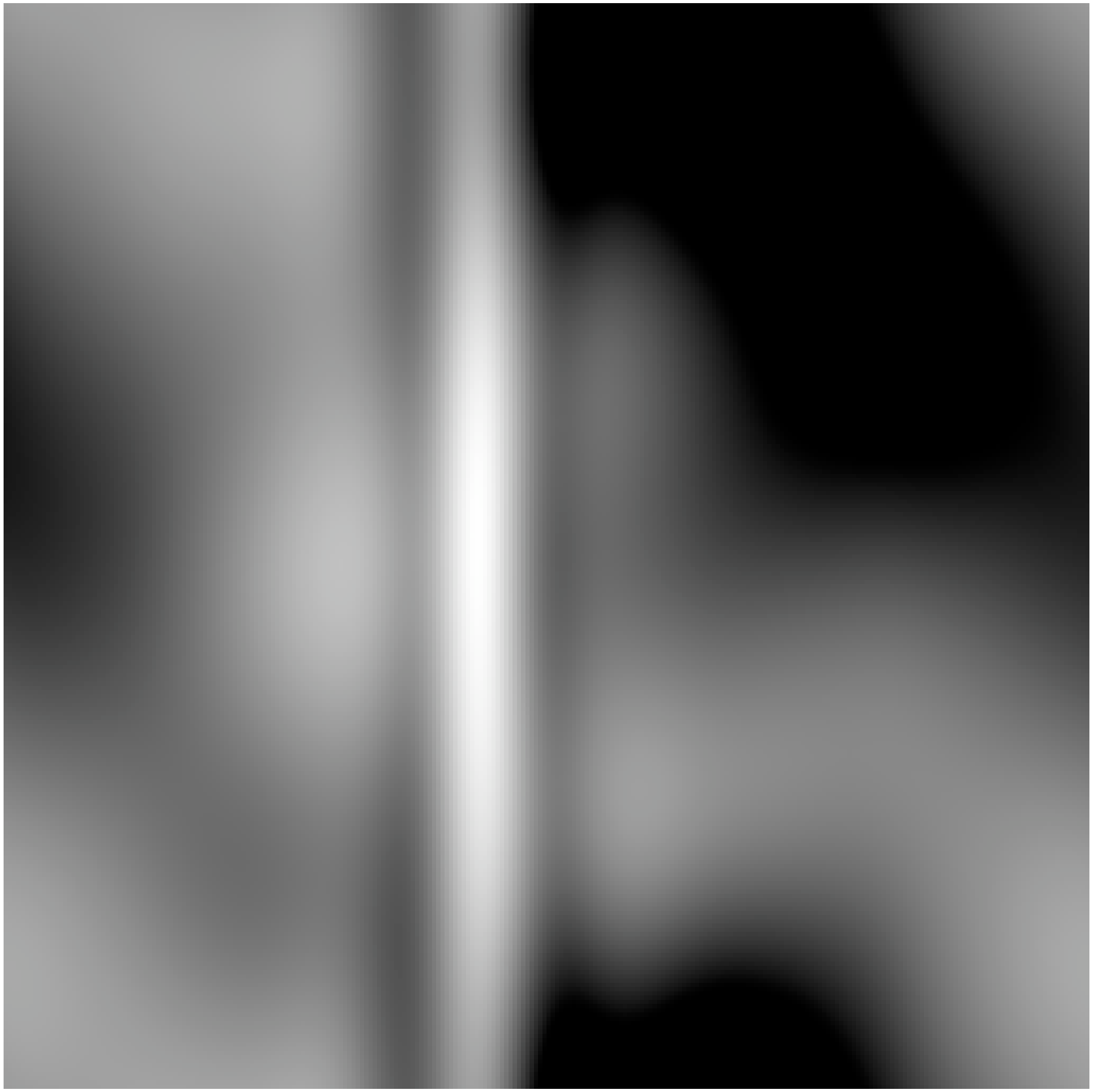



6000
4000
2000
0
-2000
-4000
-6000

Undecimated Wavelet Filtering (3 sigma)

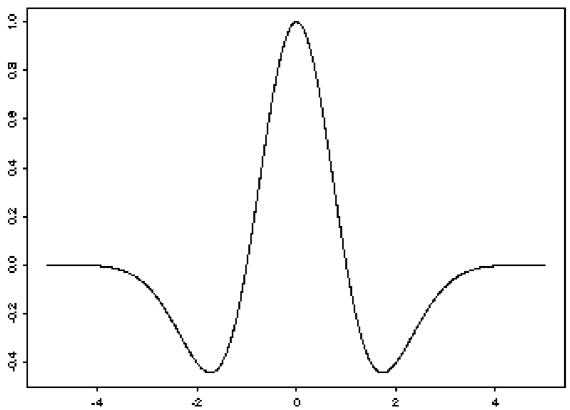


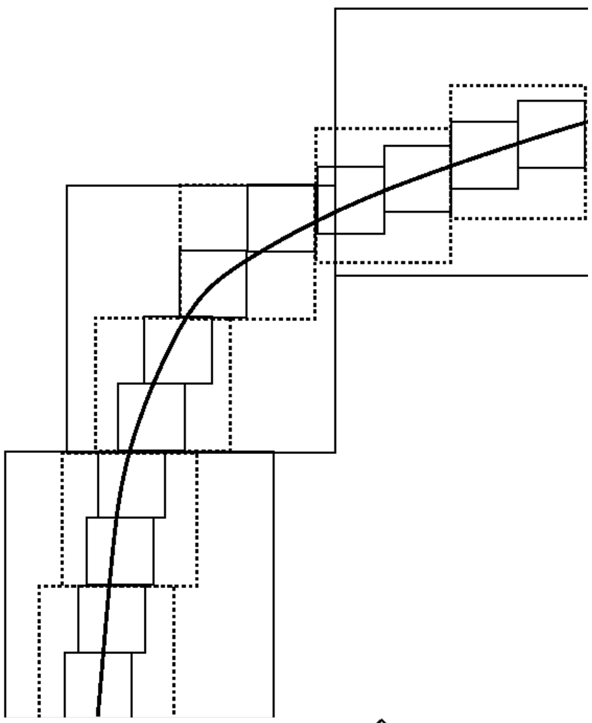
Ridgelet Filtering (5sigma)



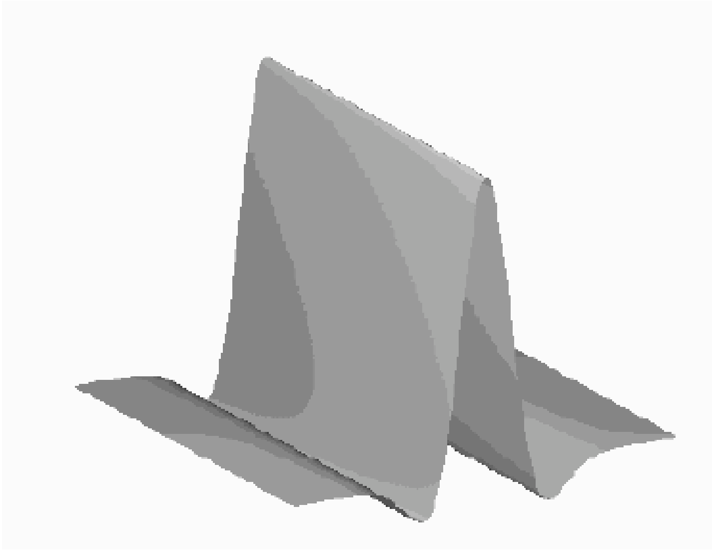
The Curvelet Transform

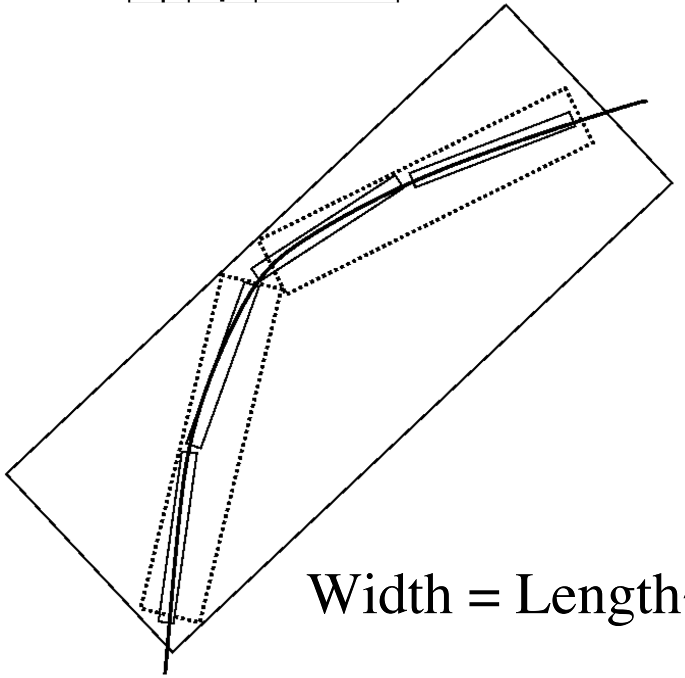
Wavelet





Curvelet





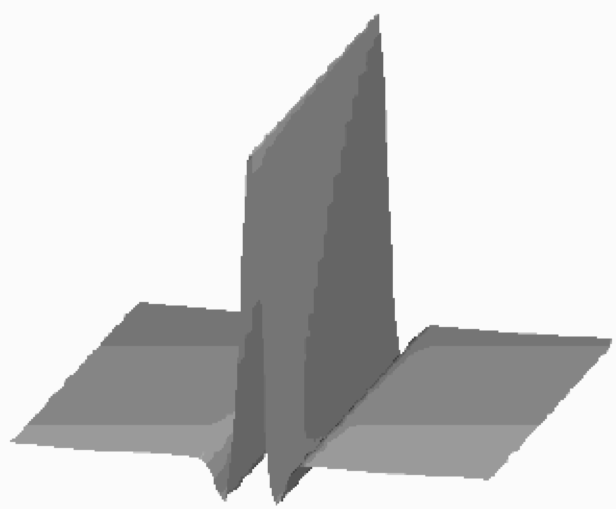
Width = Length^2

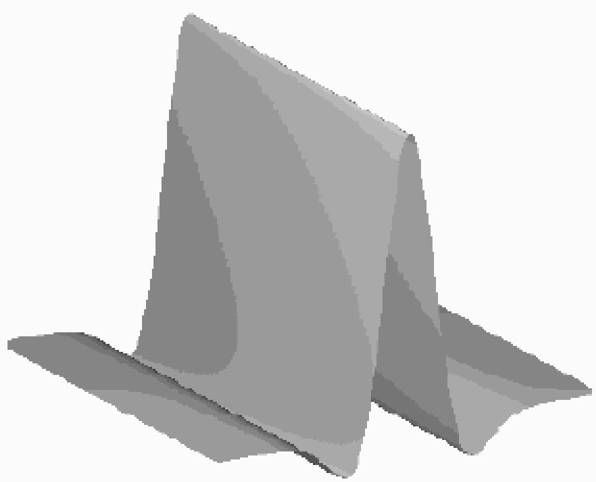
Continuous Ridgelet Transform

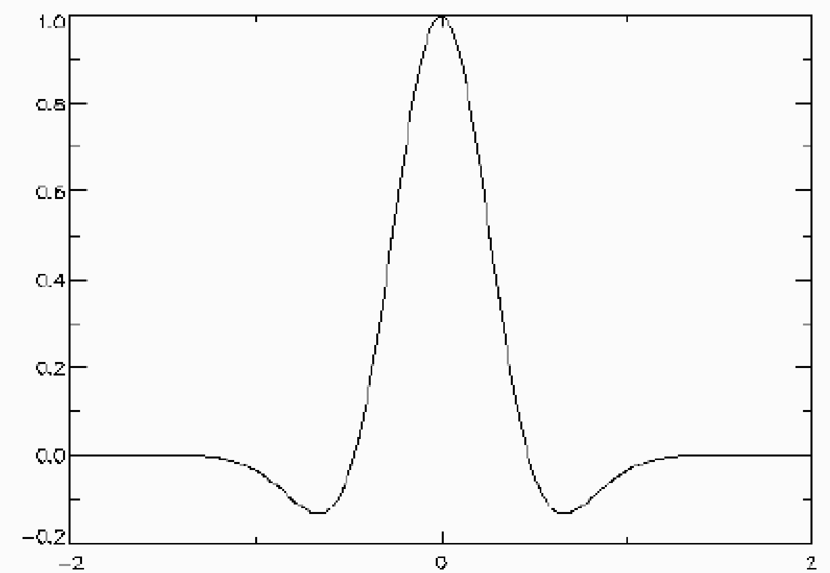
Ridgelet Transform (Candes, 1998): $R_f(a,b,\theta) = \int \psi_{a,b,\theta}(x) f(x) dx$

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.

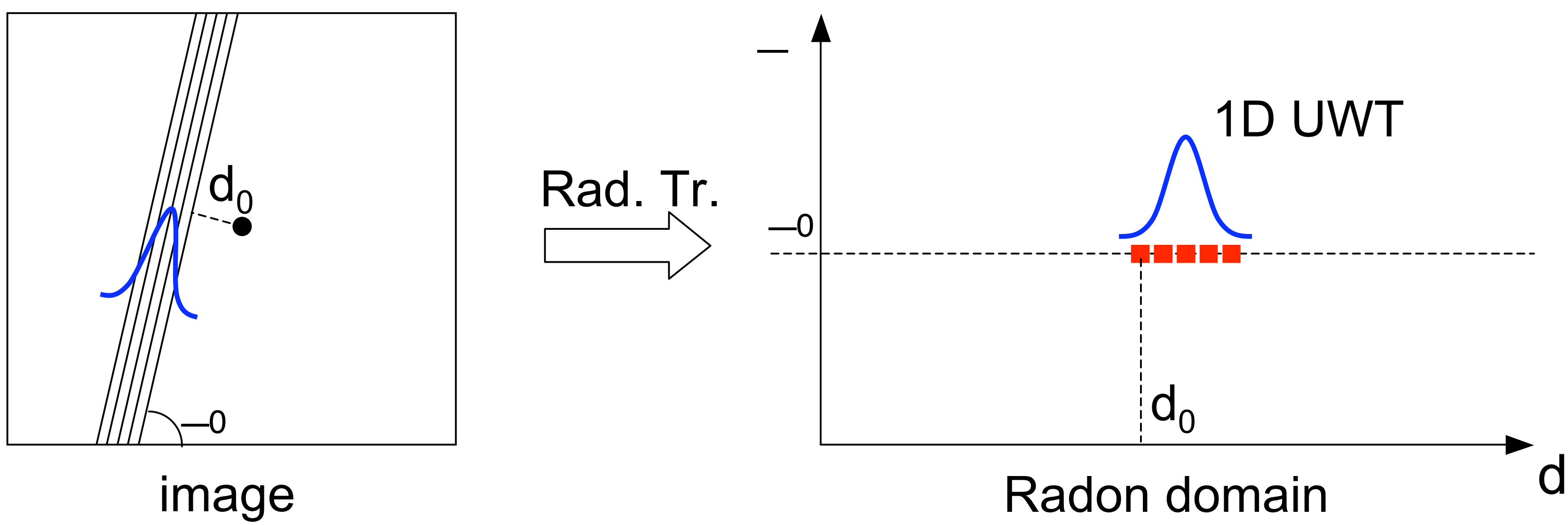






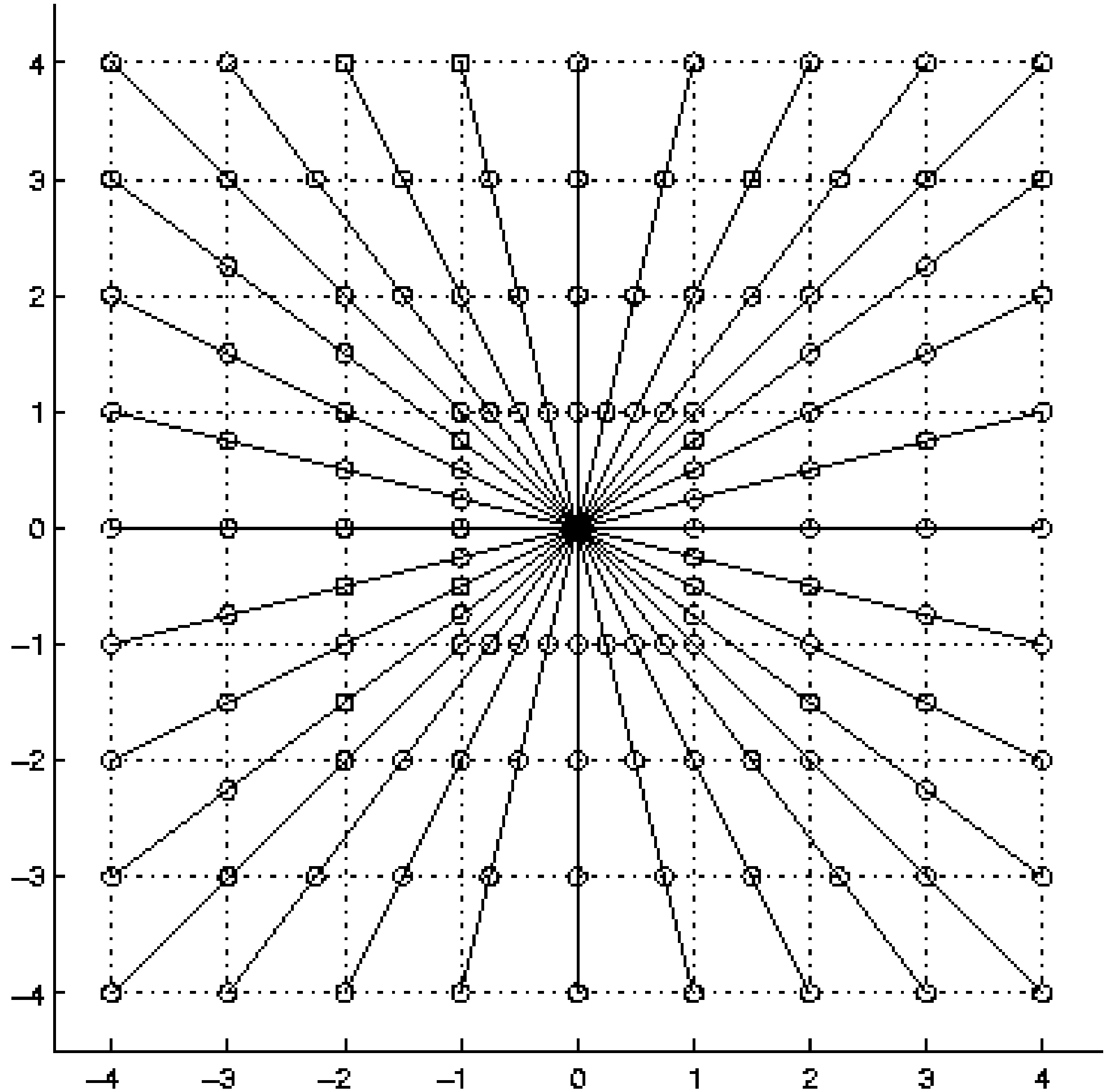
Ridgelet Denoising

- Ridgelet transform: Radon + 1D Wavelet



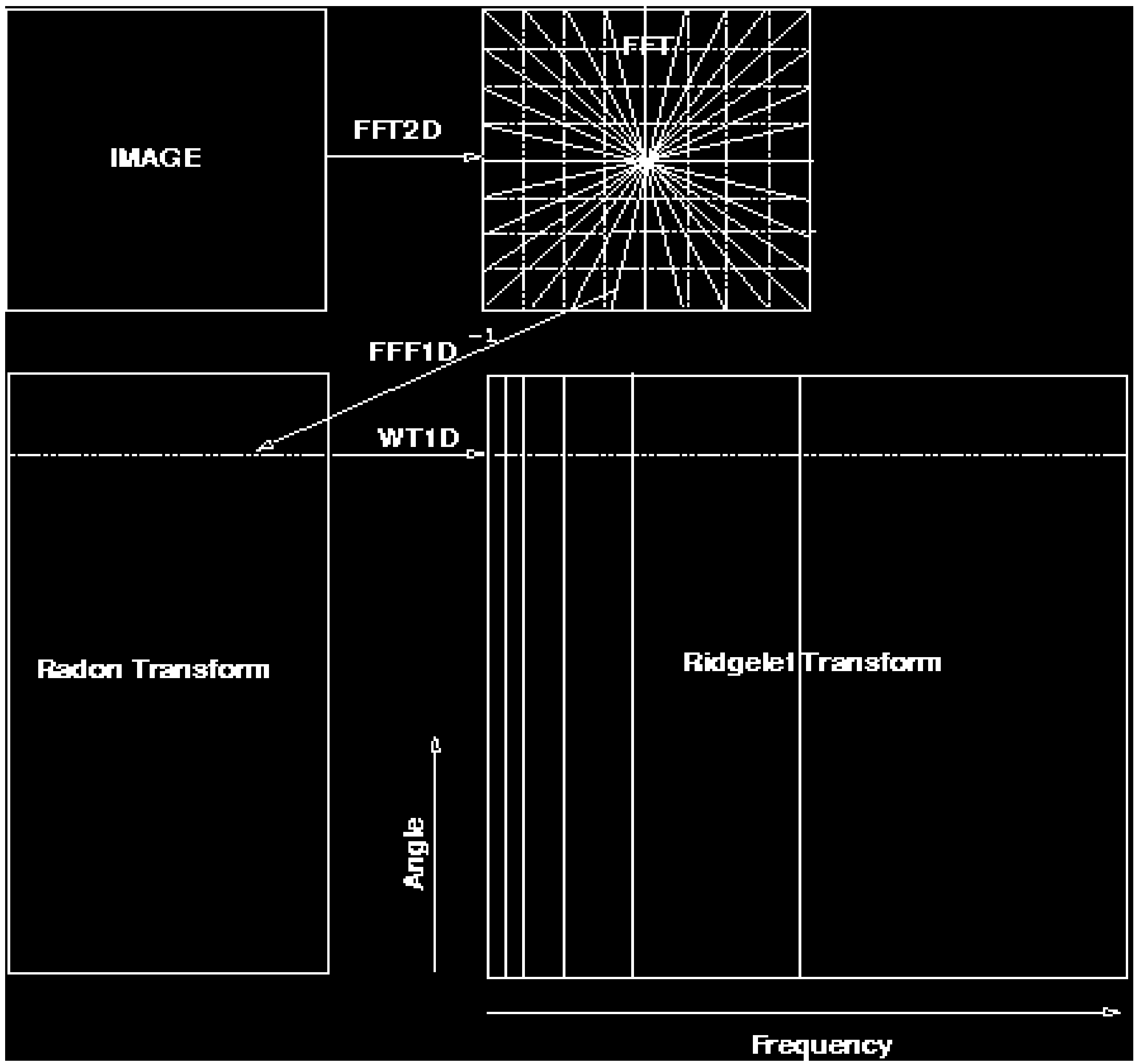
1. Rad. Tr.
2. For each line, apply the same denoising scheme as before
3. Rad. Tr.$^{-1}$

LINOGRAM CUR01



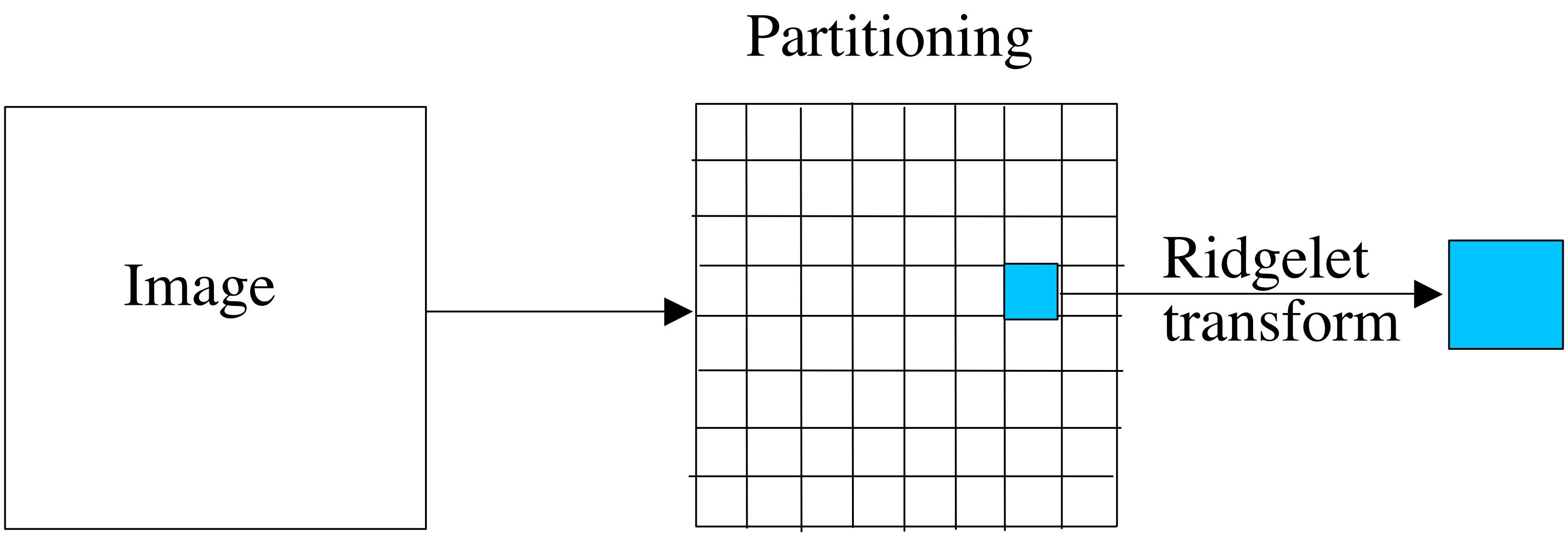
2NxN

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

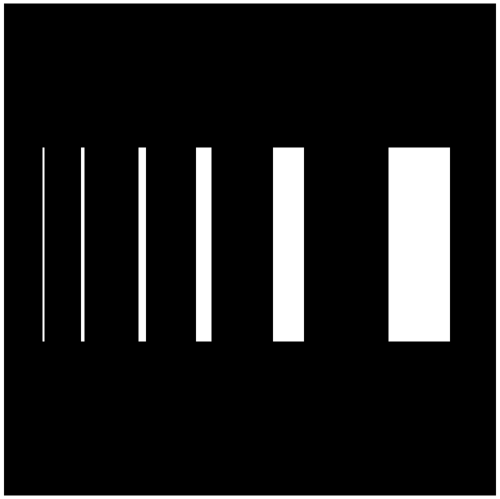


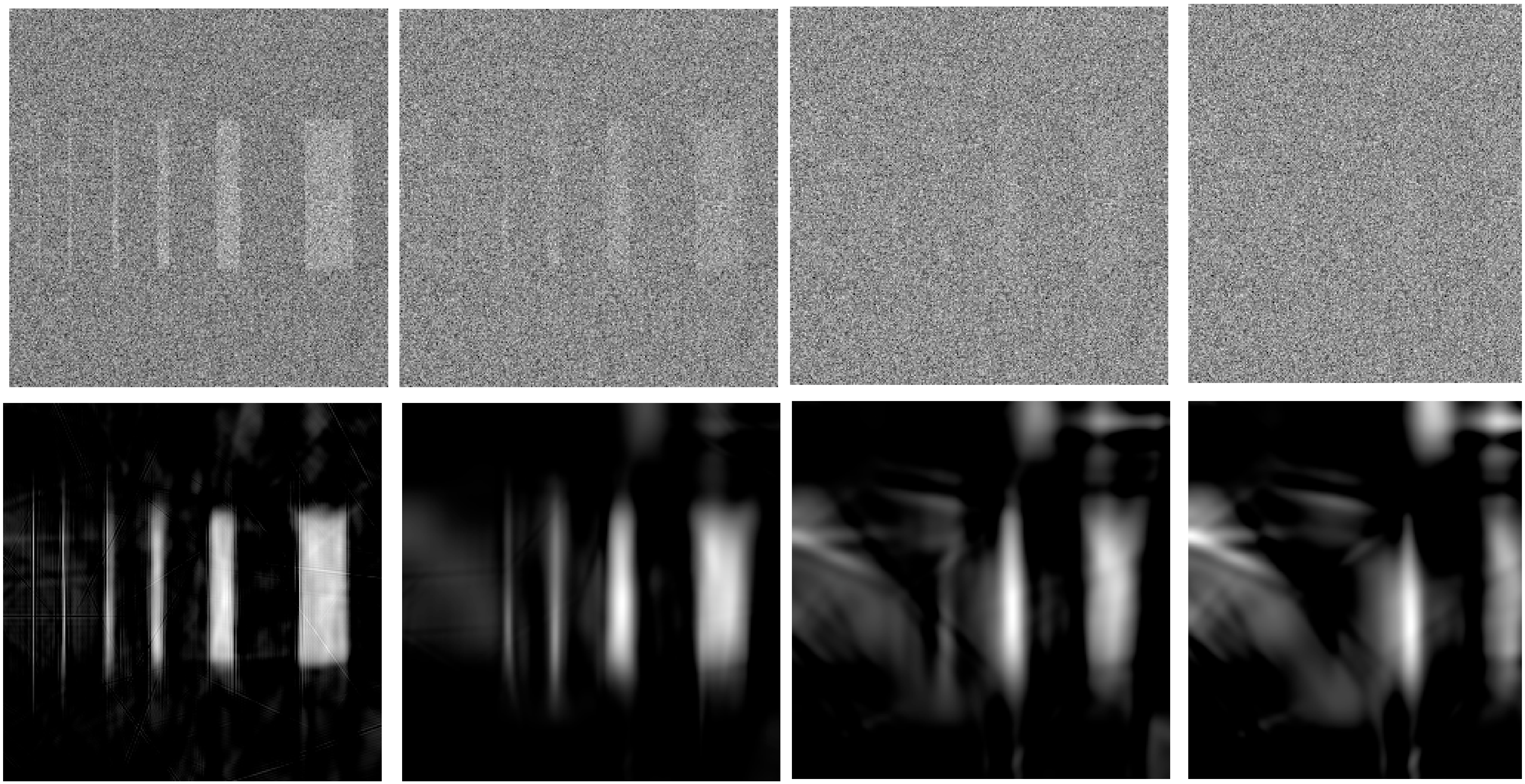
Local Ridgelet Transform

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.



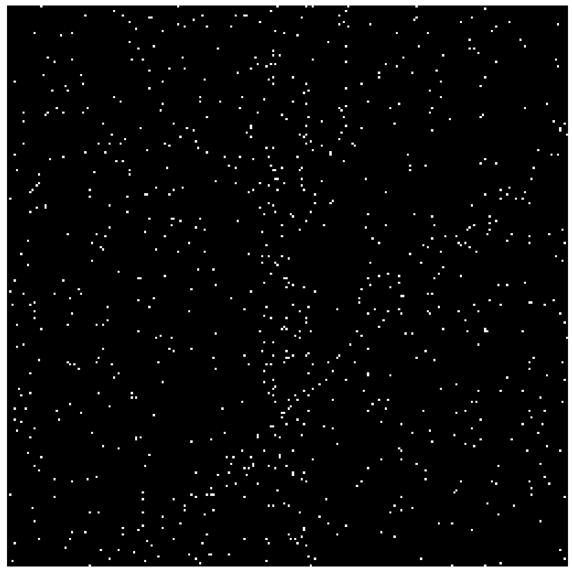
Line detection by the ridgelet transform



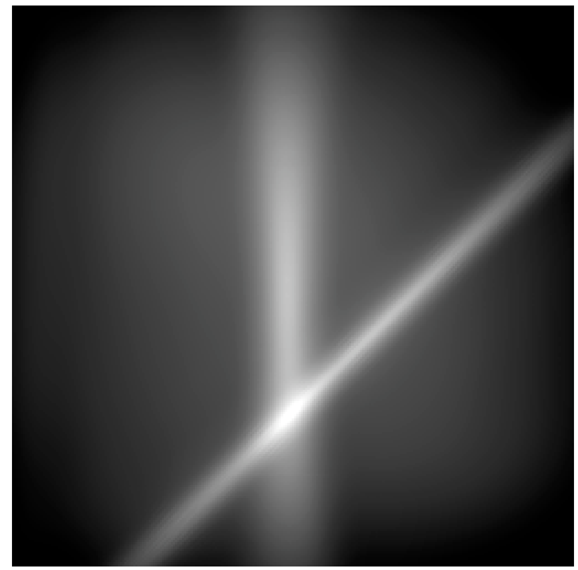


Preliminary Results – Line-Like Sources Restoration (MS-VST + Ridgelet)

underlying intensity image



simulated image of counts



restored image from the left image of counts

Max Intensity
background = 0.01
vertical bar = 0.03
inclined bar = 0.04

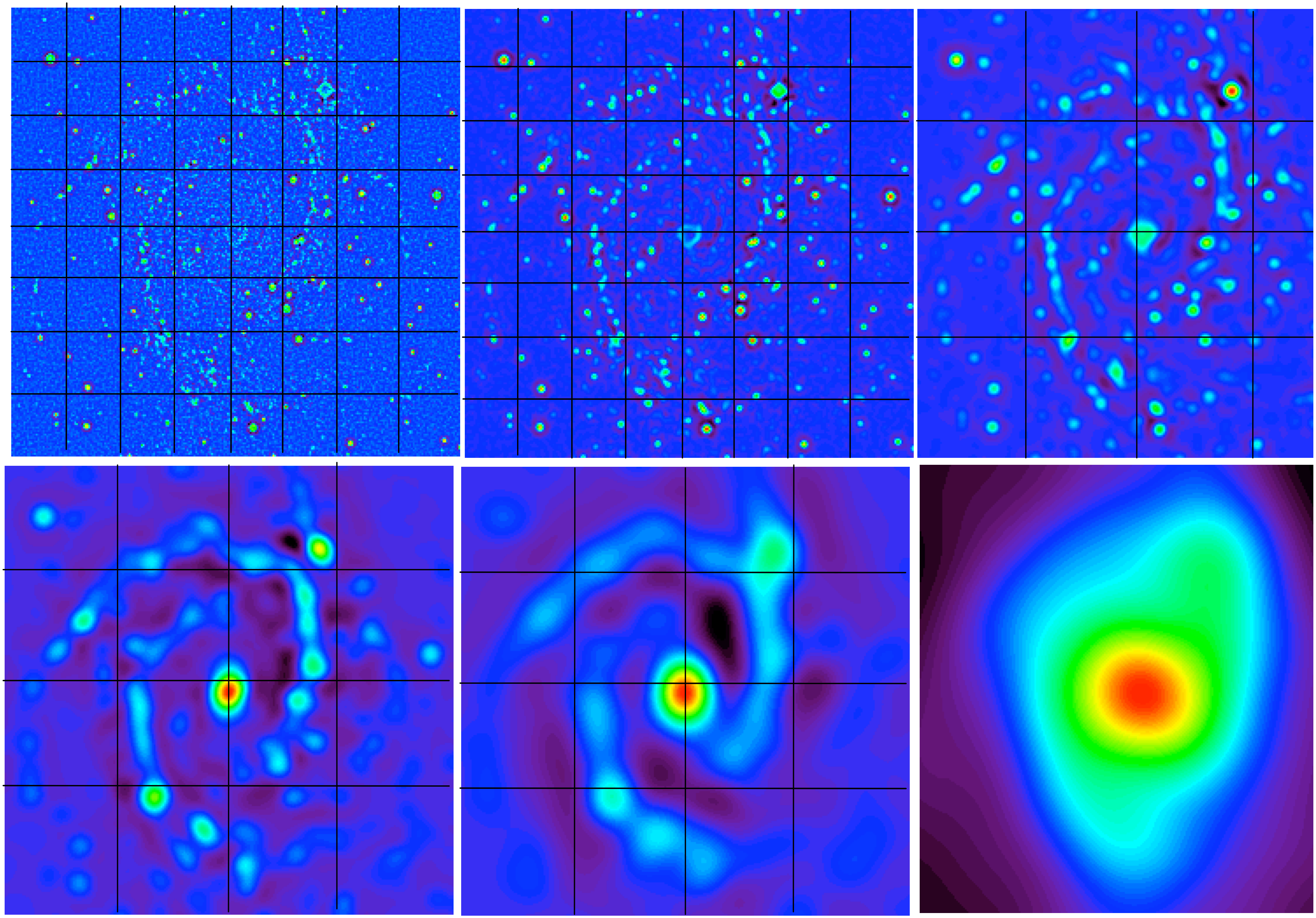
The Curvelet Transform

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

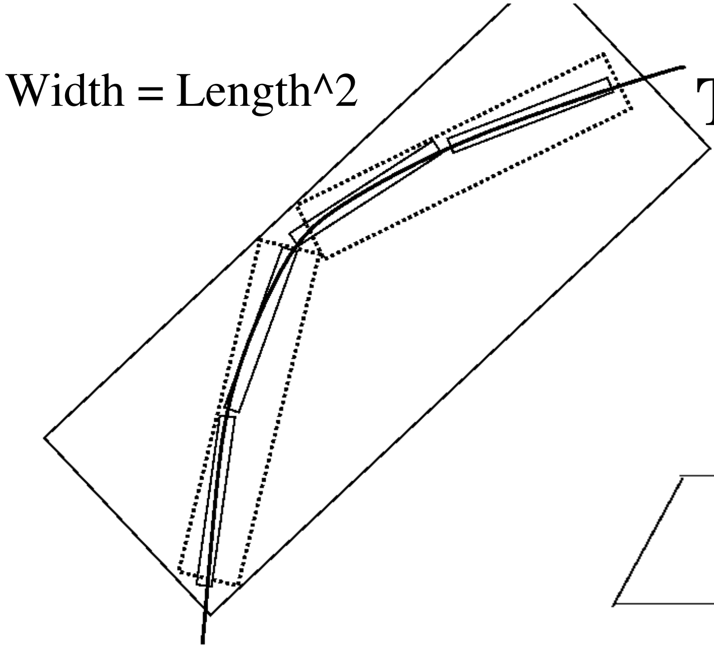
The idea is to first decompose the image into a set of wavelet bands, and to analyze each band by a ridgelet transform. The block size can be changed at each scale level.

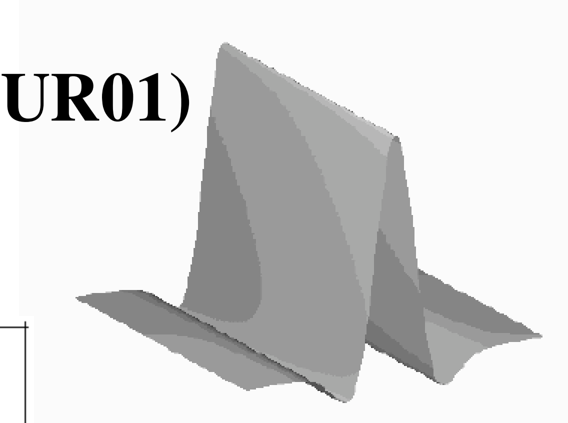
- à trous wavelet transform
- Partitionning
- ridgelet transform
  - . Radon Transform
  - . 1D Wavelet transform

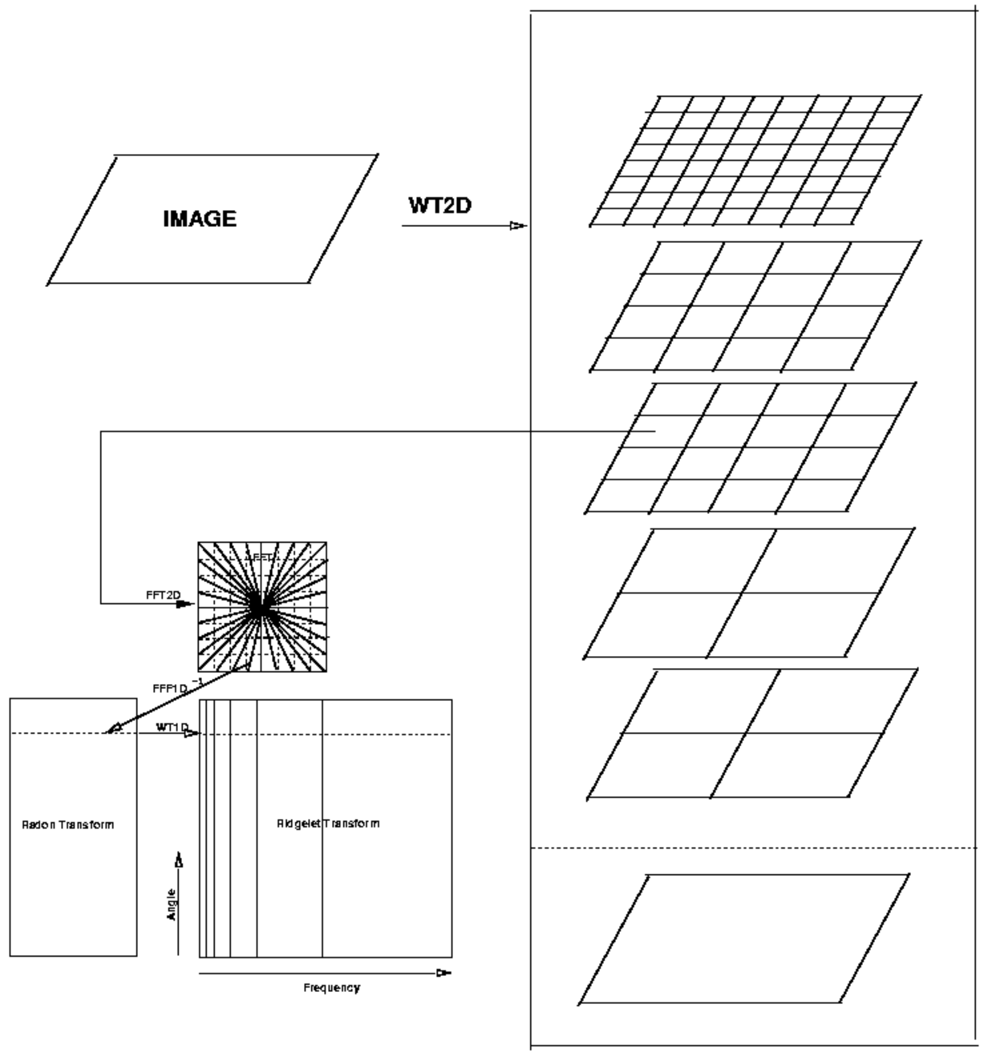
PARTITIONING



The Curvelet Transform (CUR01)



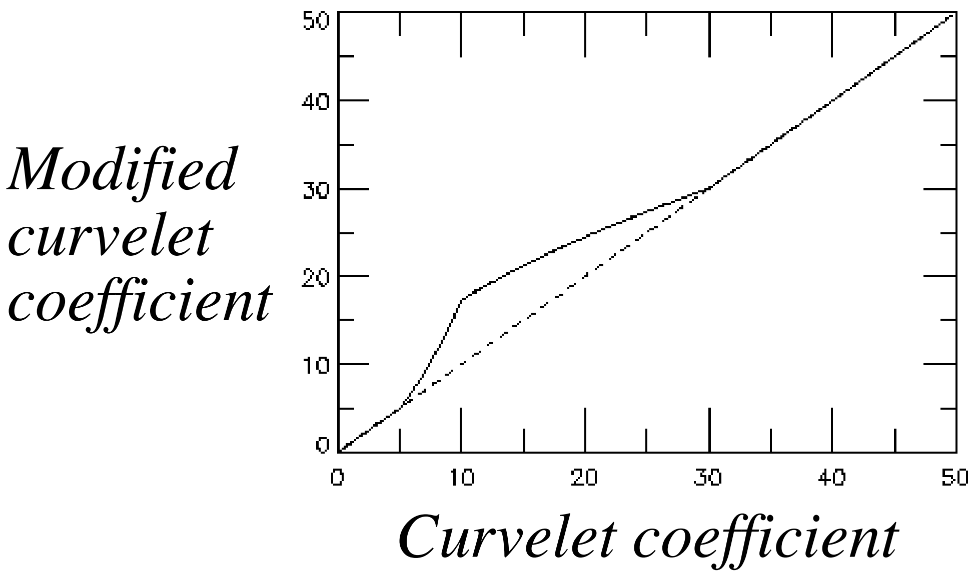




J.-L. Starck, E. Candes, D.L. Donoho ***The Curvelet Transform for Image Denoising****, IEEE Transaction on Image Processing, 11, 6, 2002.*

CONTRAST ENHANCEMENT

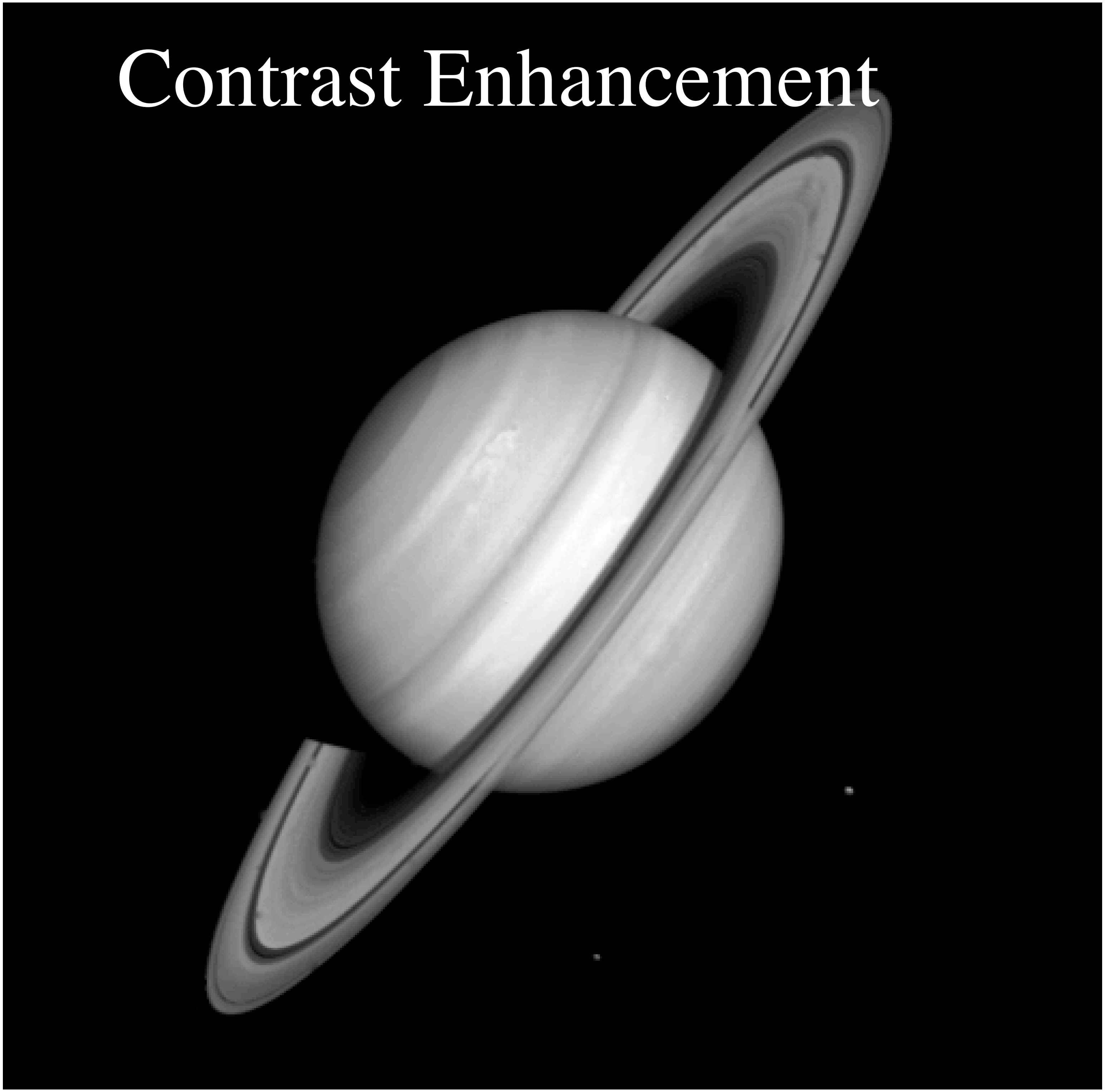
Gray and Color Image Contrast Enhancement by the Curvelet Transform,
IEEE Transaction on Image Processing, 12, 6, pp 706--717, 2003.

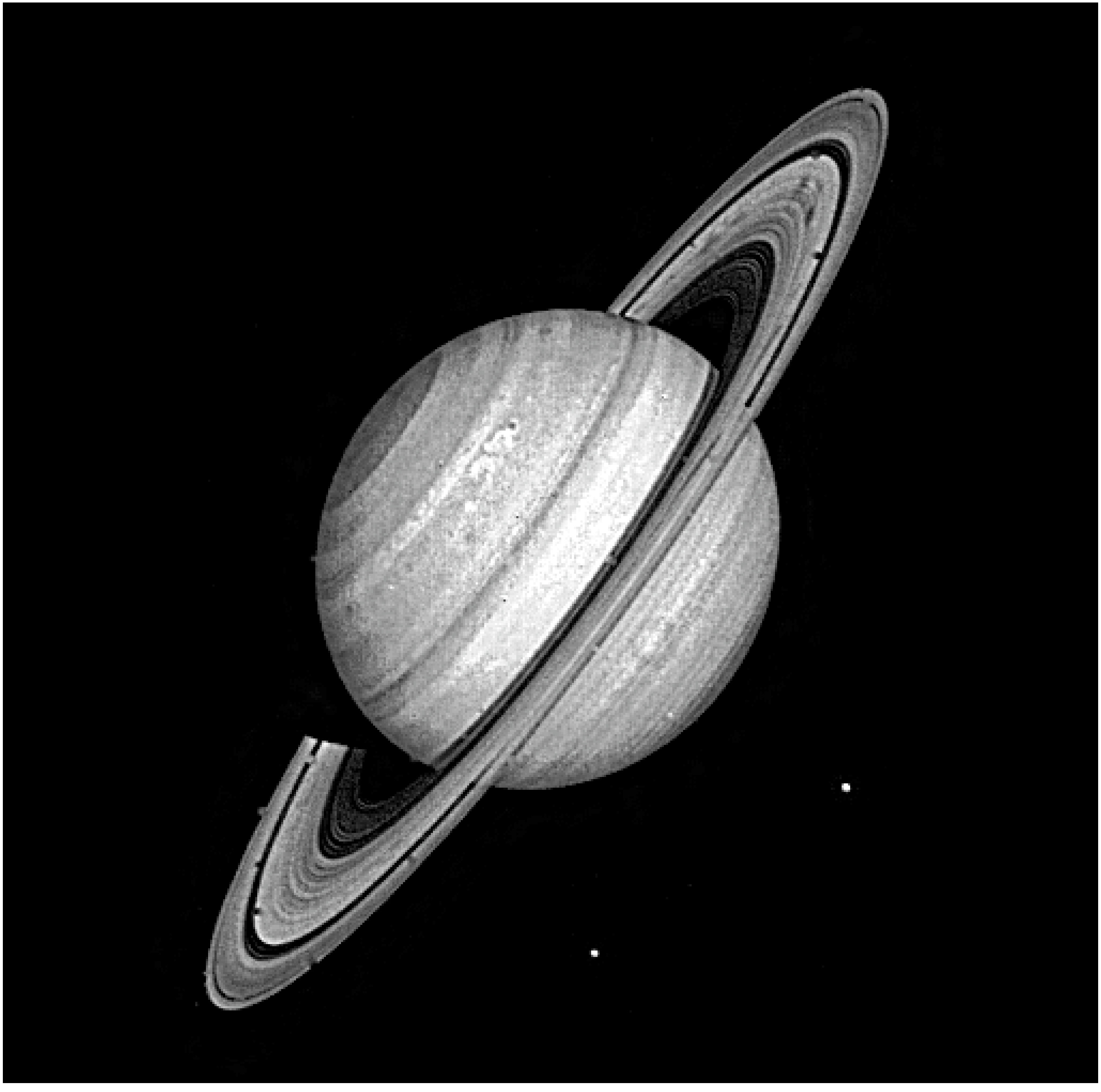


$$\tilde{I} = C_R\left(y_c\left(C_T I\right)\right)$$

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

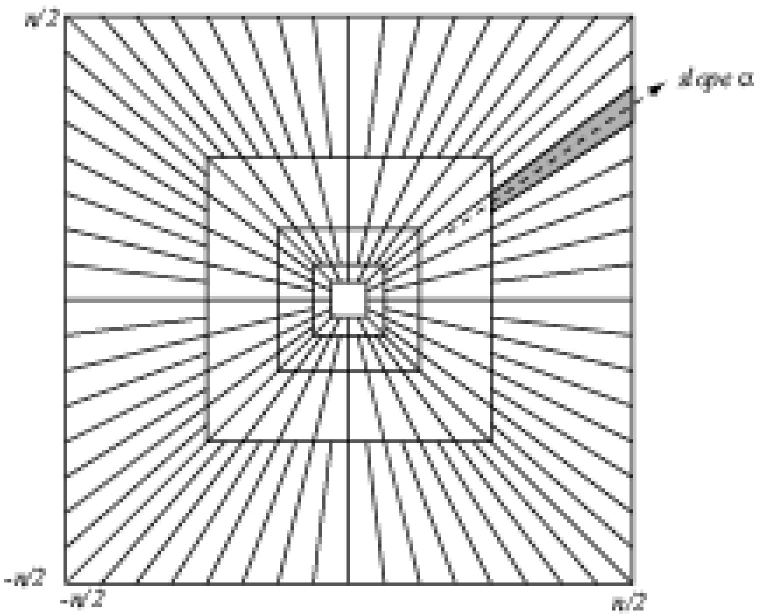
Contrast Enhancement



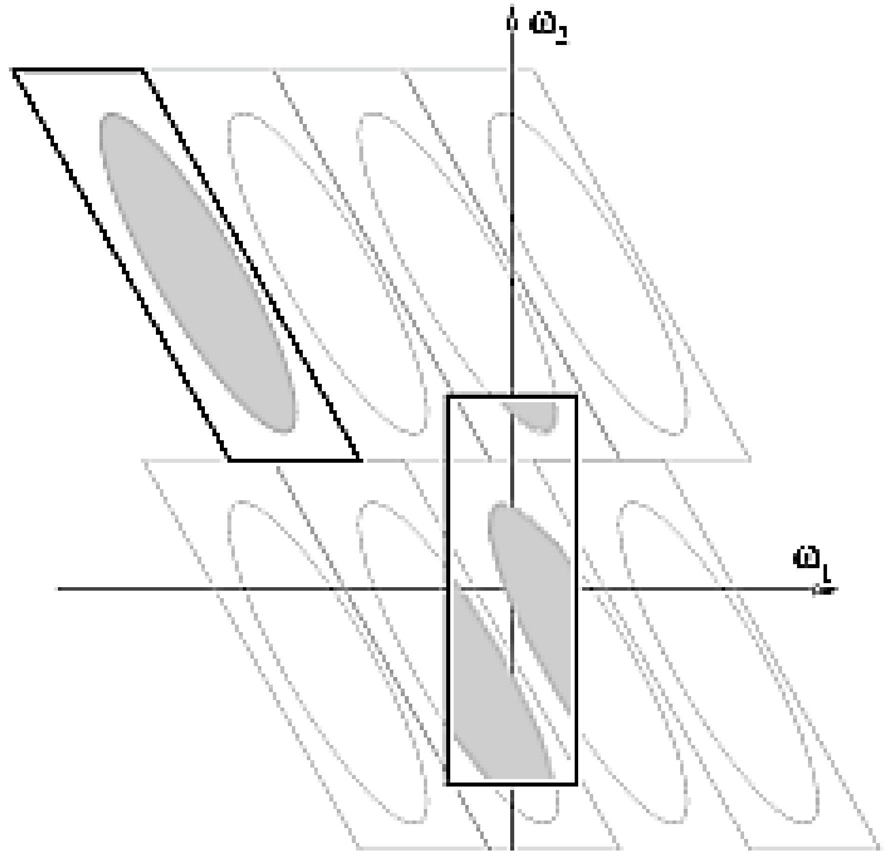


The Fast Curvelet Transform, Candes et al, 2005

CUR03 - Fast Curvelet Transform using the USFFT
CUR04 - Fast Curvelet Transform using the Wrapping and 2DFFT

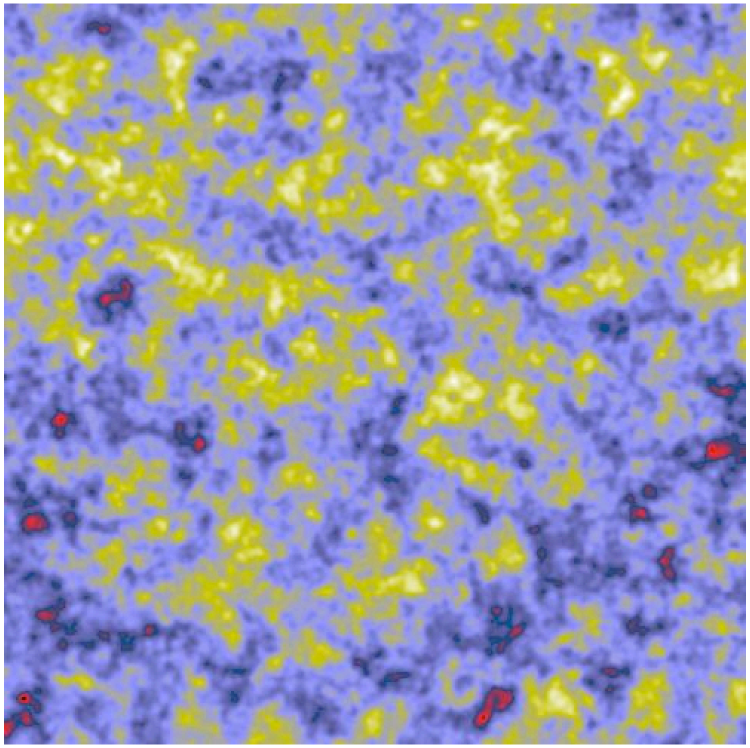


(a)

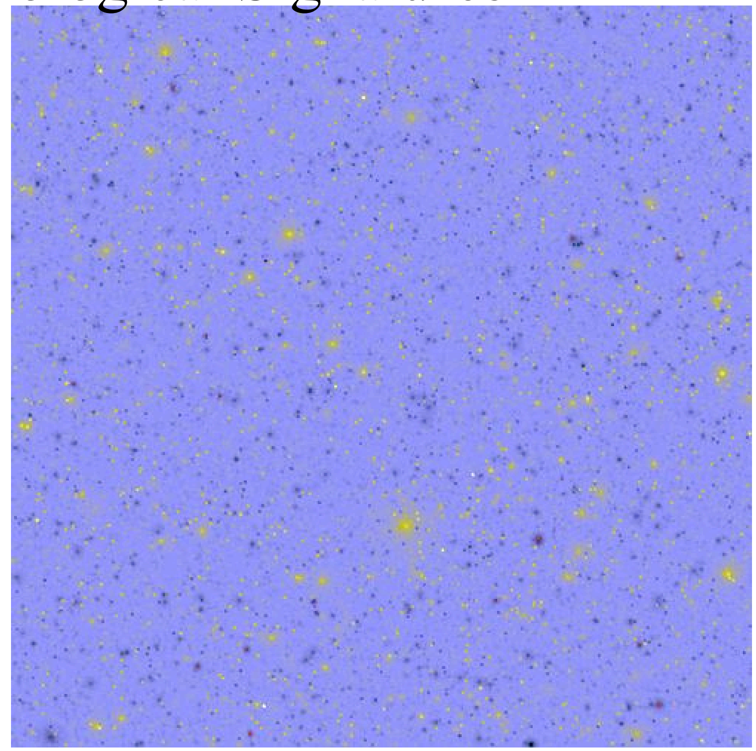


Detection of non-Gaussian Cosmological Signatures

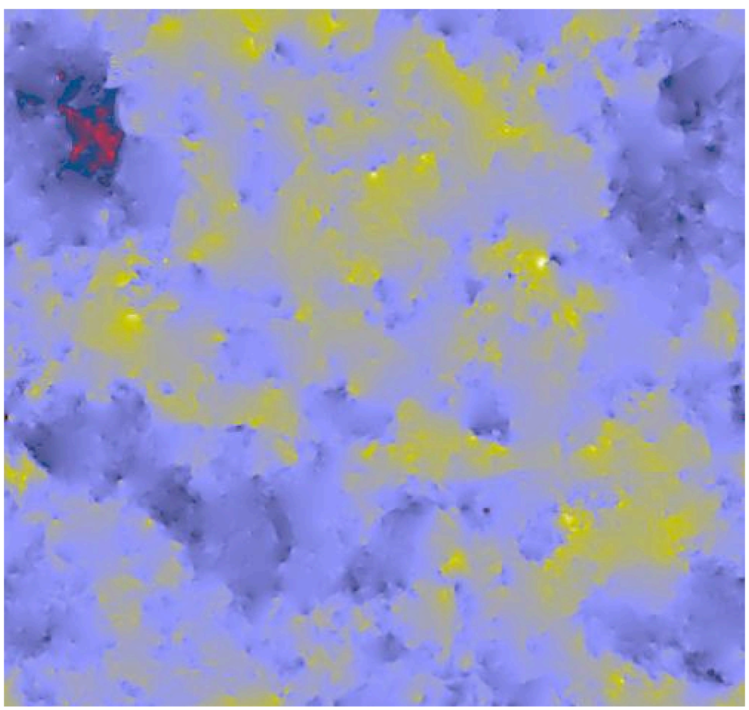
CMB



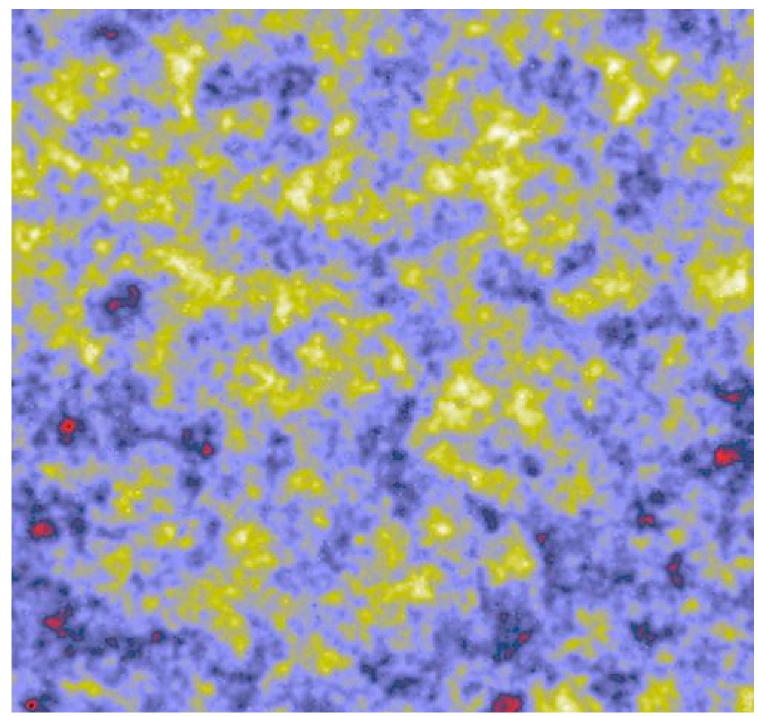
SZ

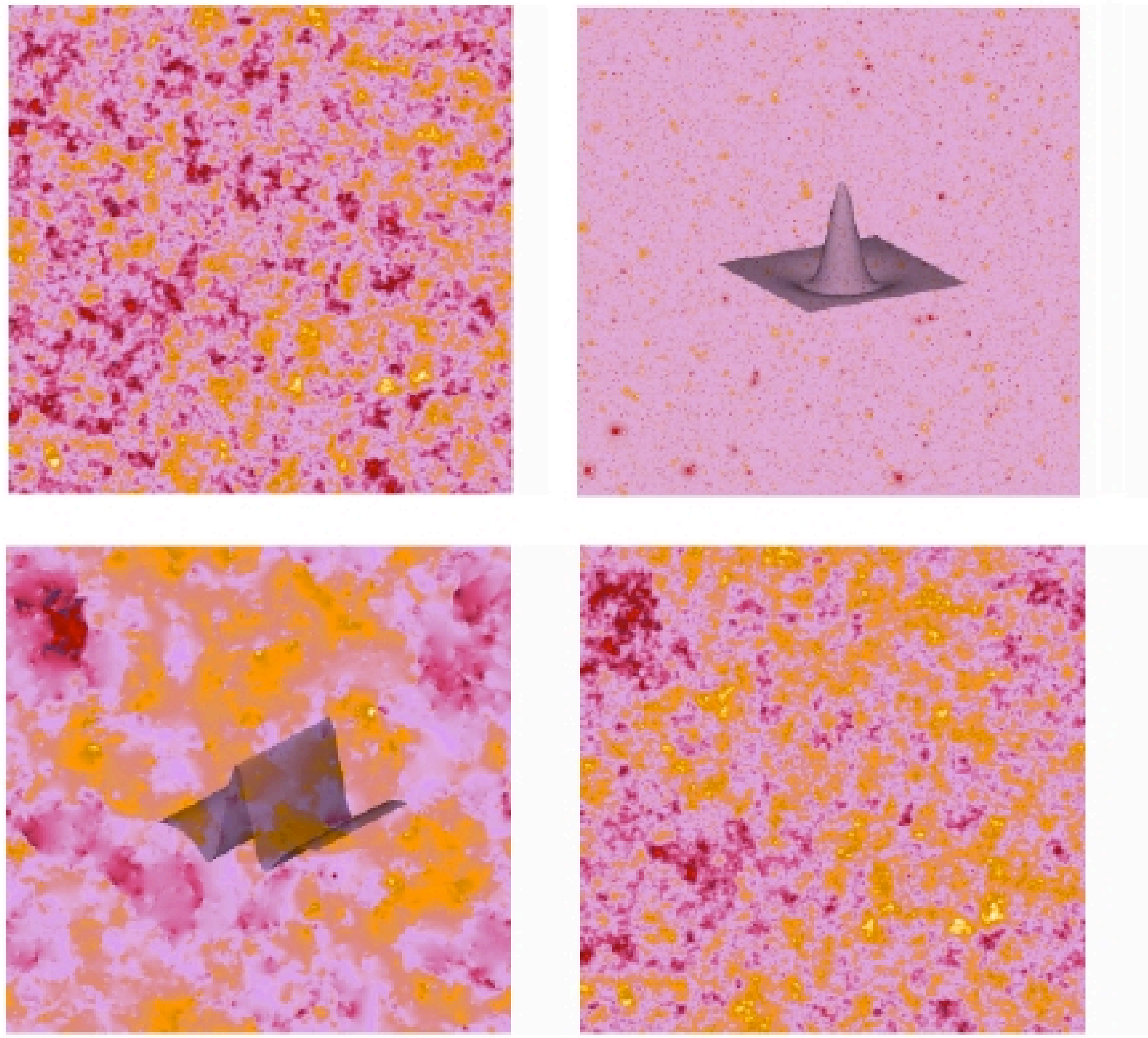


CS

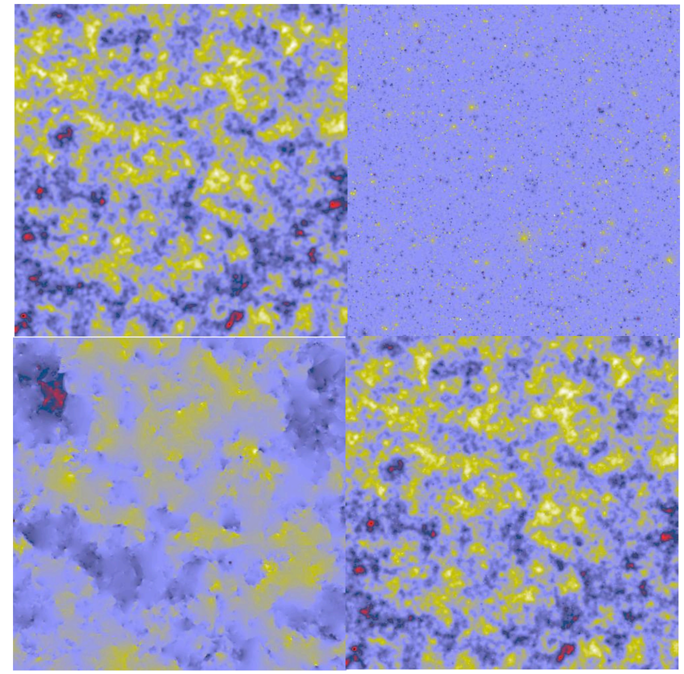


Total





Multiscale Analysis of the CMB



We have applied the following multiscale transforms

- Isotropic wavelet transform
- Bi-orthogonal wavelet transform
- Ridgelets (block size of 16 pixels)
- Ridgelets (block size of 32 pixels)
- Curvelets

On

1) 100 **CMB + KSZ** + 100 Gaussian realizations with the same power spectrum.

$$K_{CMB-SZ}[i,j] \Rightarrow \overline{K}_{CMB-SZ}[j] = \frac{\text{mean}(K_{CMB-SZ}[1:100,j]) - \text{mean}(K_{CMB}[1:100,j])}{\text{sigma}(K_{CMB}[1:100,j])}$$

2) 100 **CMB + CS** + 100 Gaussian realizations with the same power spectrum

3) 100 **CMB + KSZ + CS** + 100 Gaussian realizations with the same power spectrum

We compare the normalized kurtosis for the three data set.

Results

- **Curvelets are NOT sensitive to KSZ but *are* sensitive to cosmic strings**

	Bi-orthogonal WT	Ridgelet	Curvelet
CMB+KSZ	1106.	0.1	10.12
CMB+CS	1813.	5.7	198.
CMB+CS+KSZ	1040.	5.9	165.

Detecting cosmological non-Gaussian signatures by multi-scale methods, *Astron. and Astrophys., 416, 9--17, 2004* .

Cosmological Non-Gaussian Signatures Detection: Comparison of Statistical Tests, *Eurasip Journal on Applied Signal Processing*, 15 pp 2470-2485, 2005.

Data on the Sphere