Morphological Component Analysis (MCA)

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

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

Collaborators:

D.L. Donoho, *Department of Statistics, Stanford*

M. Elad, *The Technion, Israel Institute of Technology*

Y. Moudden, *CEA-Saclay, France*

J. Bobin, *CEA-Saclay, France*

B. Zhang, *CEA-Saclay, France*

J. Fadili, *Caen University*

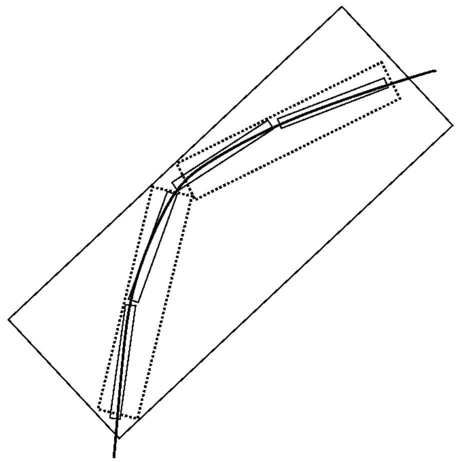
Morphological Component Analysis: *MCA allows us to separate features in an image which present different morphological aspects. MCA is based on fast transform/reconstruction operators.*

TRANSFORMS

. DCT
. Orthogonal WT: Mallat, 1989.
. Bi-orthogonal WT: Daubechies, Cohen, … 1992
. Lifting Scheme: Swelden, 1996 (JPEG 2000 Norm).

REDUNDANT TRANSFORMS

- Local DCT ([overlapping] blocks + DCT)
- **Undecimated Wavelet Transform**
- **Isotropic Undecimated Wavelet Transform**
- Ridgelet Transform
- Curvelet Transform →



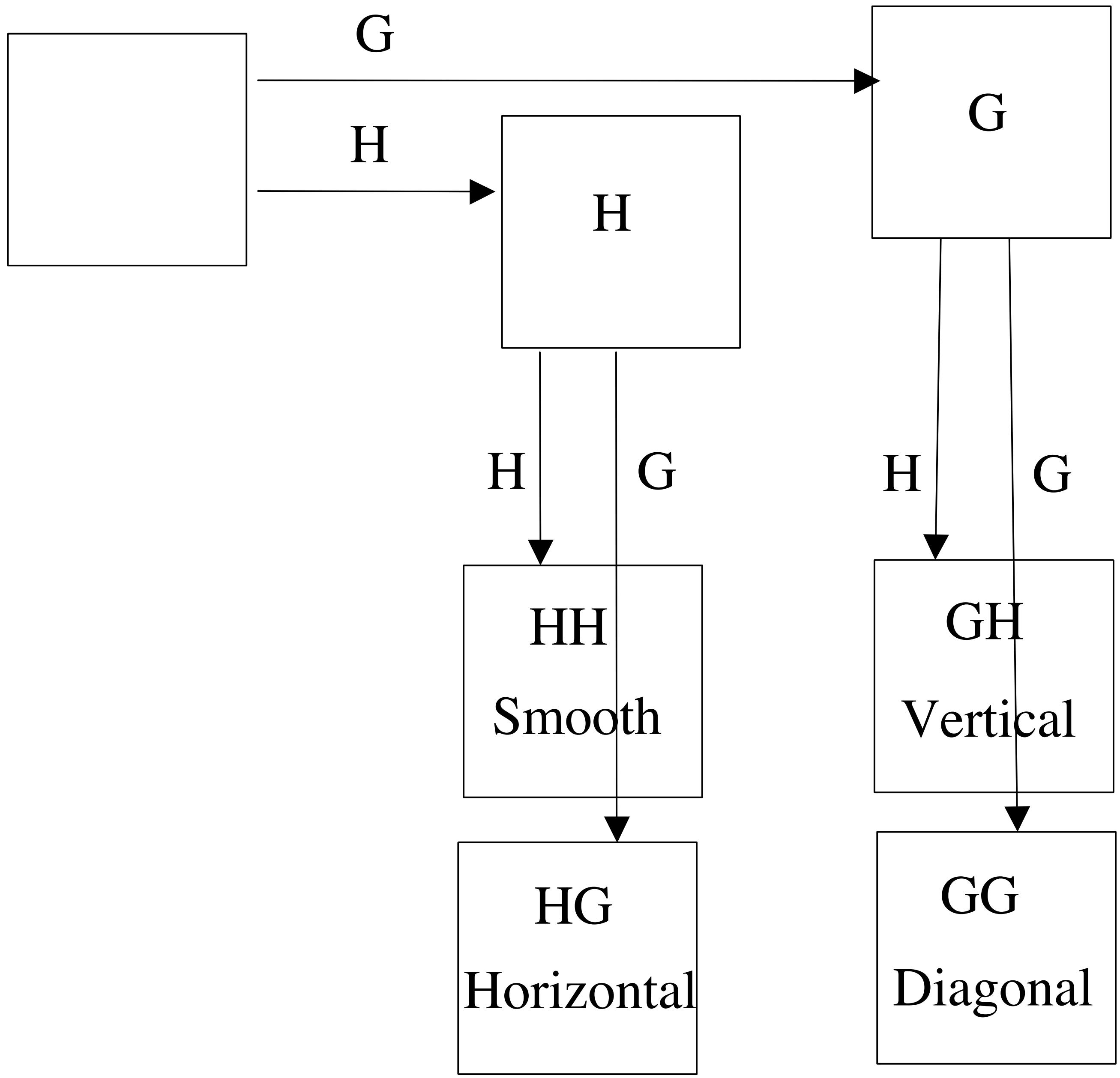
1. **Relation between the Undecimated Wavelet Transform and the Isotropic Wavelet Transform**
   => New Filter Banks for undecimated WT

2. **The MCA algorithm**

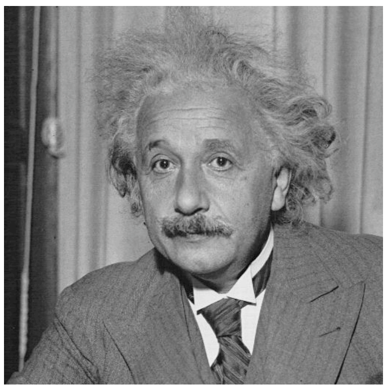
3. **MCA texture extraction**

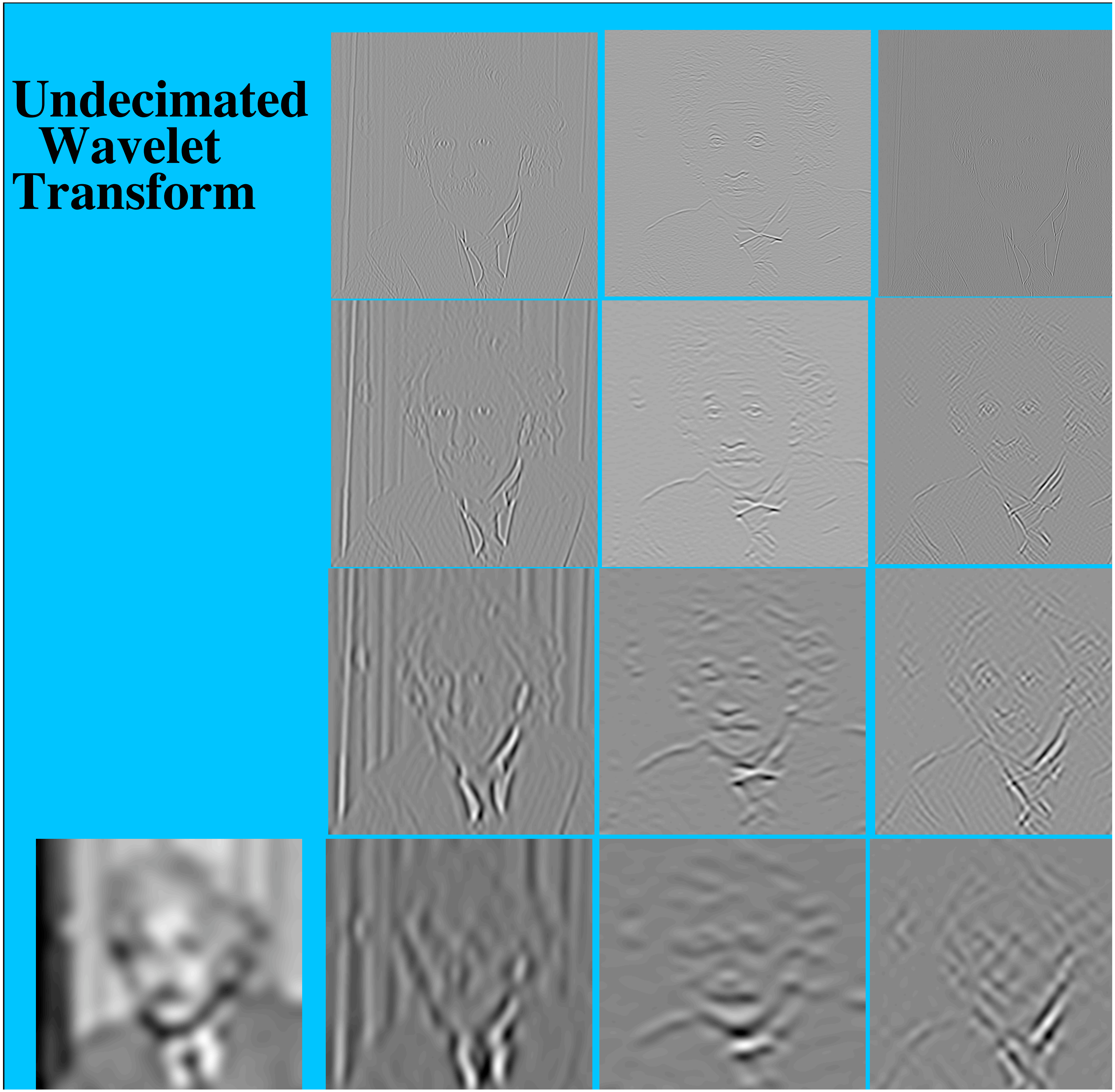
4. **MCA Inpainting**

5. **Multichannel MCA**



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




Undecimated
Wavelet
Transform

The Isotropic Undecimated Wavelet Transform

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

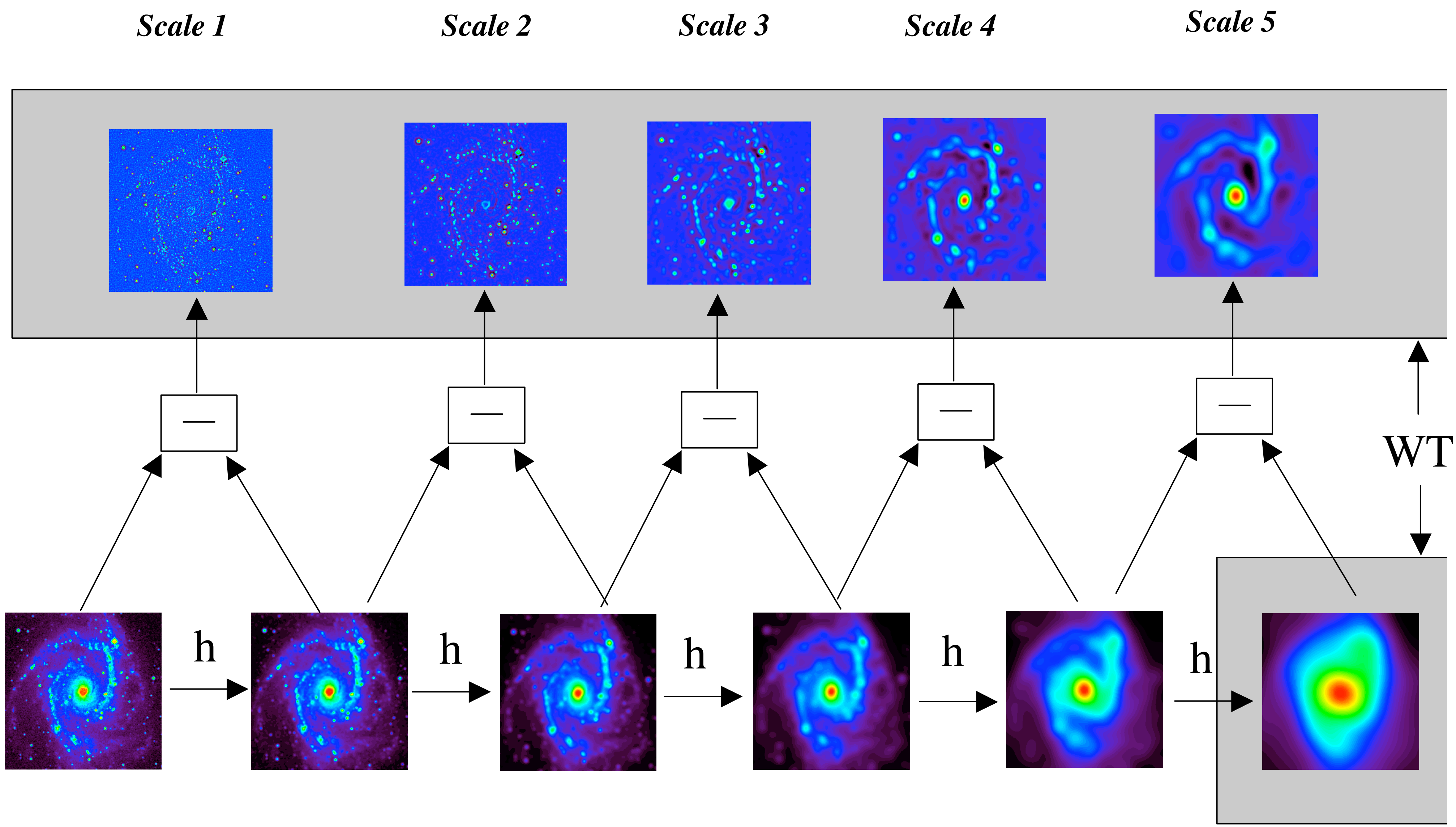
$$\hat{h}(\nu)\hat{\tilde{h}}(\nu) + \hat{g}(\nu)\hat{\tilde{g}}(\nu) = 1$$

- 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.

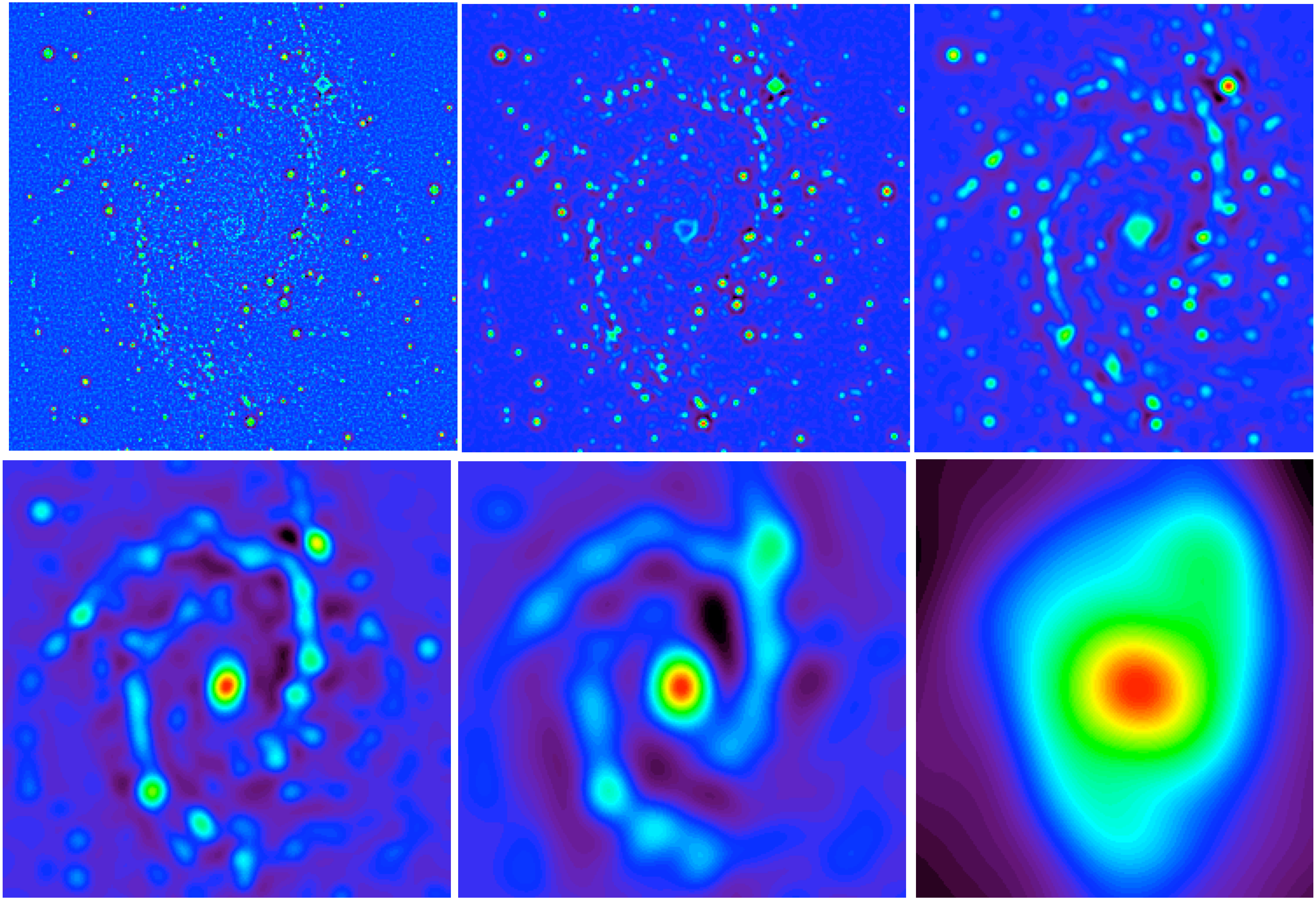
h is derived from a B_3 spline: $h_{1D}(k) = [1, 4, 6, 4, 1]/16$, and in 2D $h_{2D} = h_{1D}h_{1D} =$

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

ISOTROPIC UNDECIMATED WAVELET TRANSFORM



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

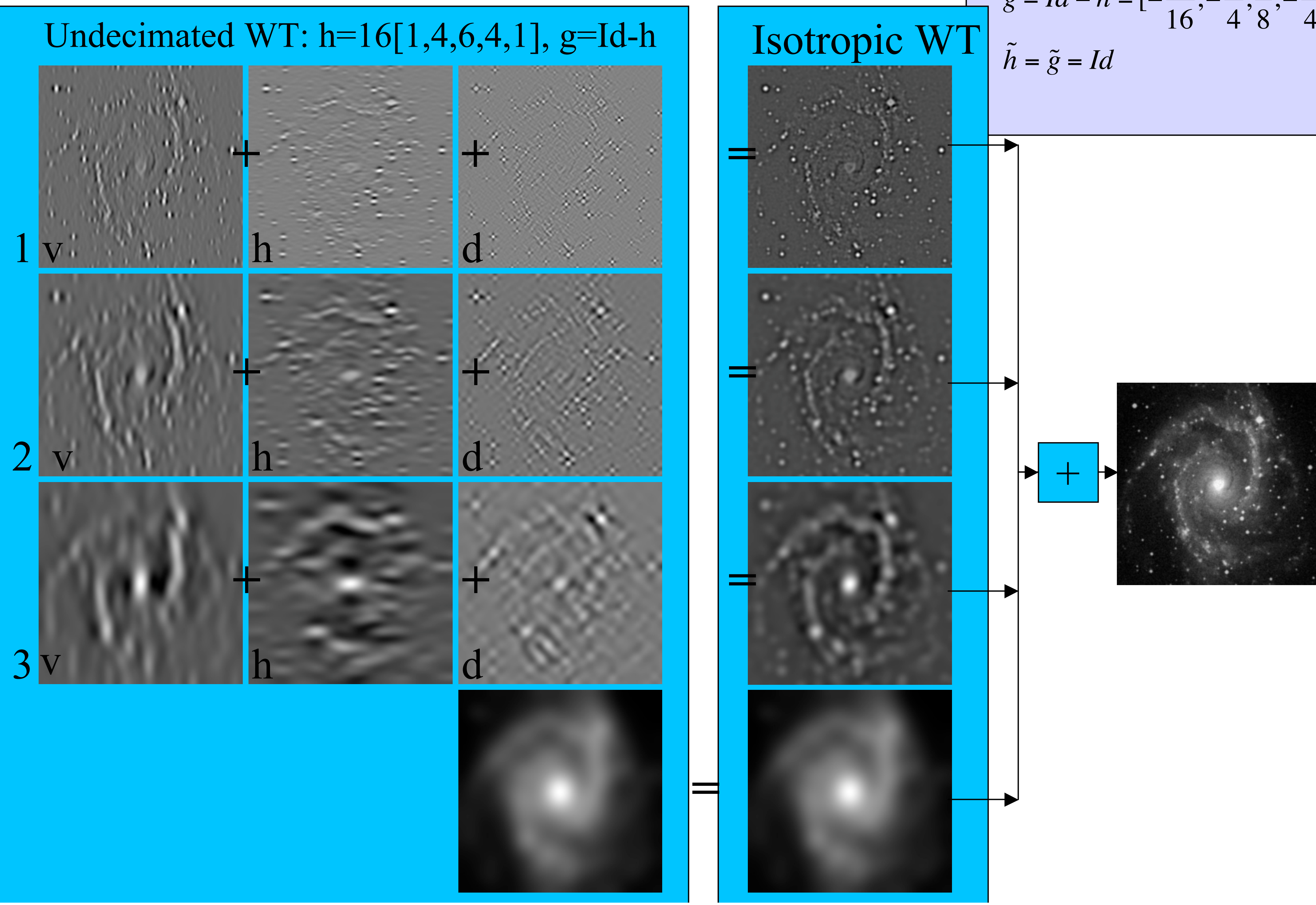


Non Orthogonal Filter Bank

$$h = [\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}]$$

$$g = Id - h = [-\frac{1}{16}, -\frac{1}{4}, \frac{5}{8}, -\frac{1}{4}, -\frac{1}{16}]$$

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



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:

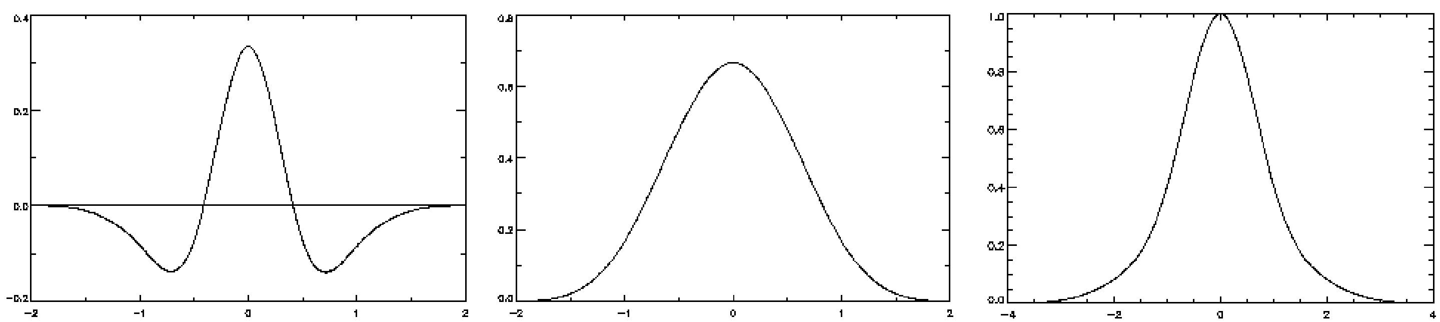
$$\tilde{g} = h + Id$$

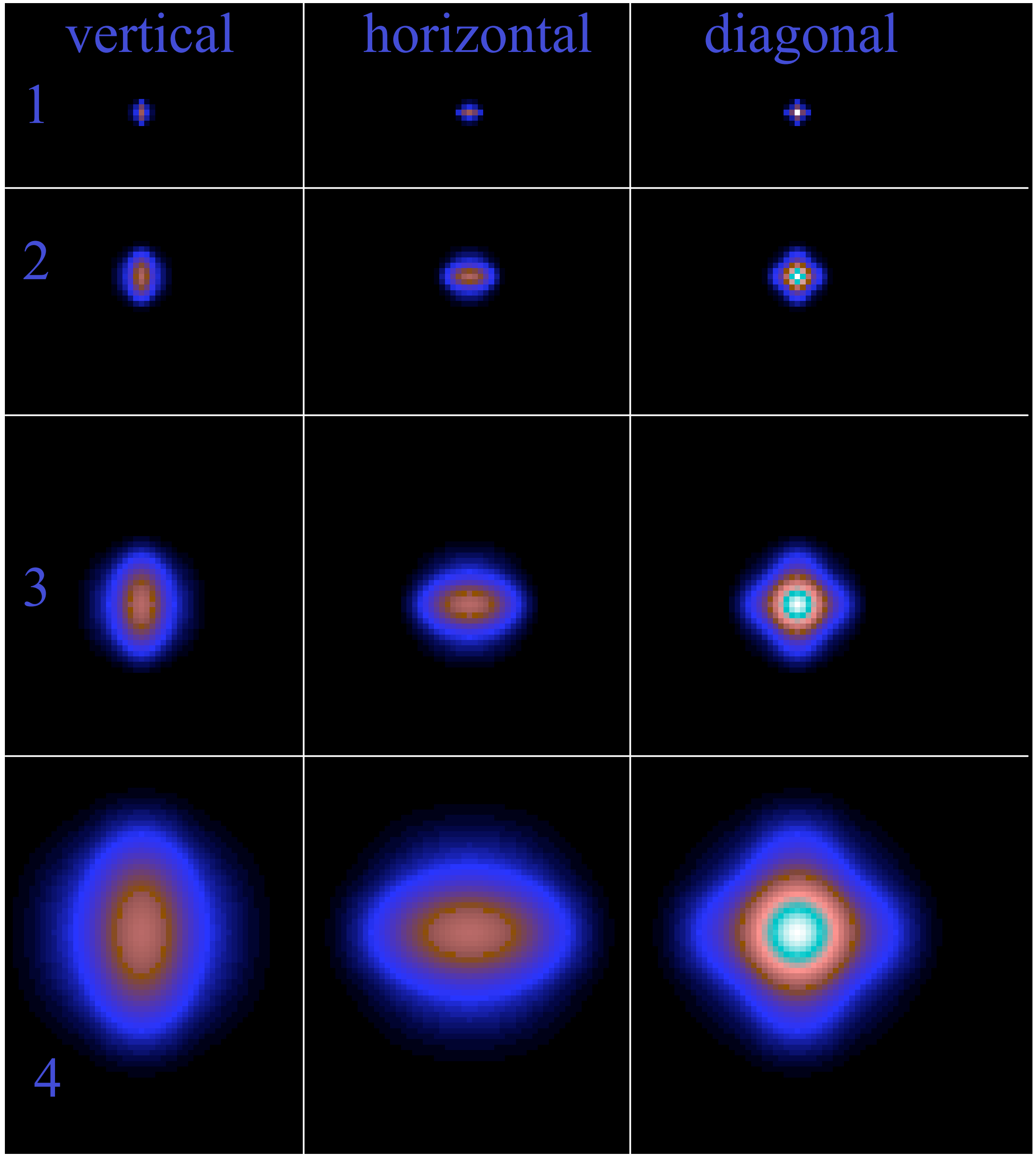
if $h = [1, 4, 6, 4, 1]/16$, then $g = [1, 4, 22, 4, 1]/16$. g **is positive**. This means that g is not related anymore to a wavelet function. The synthesis scaling function related to $\tilde{g}$ is defined by:

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

Reconstruction Using the Scaling Function

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



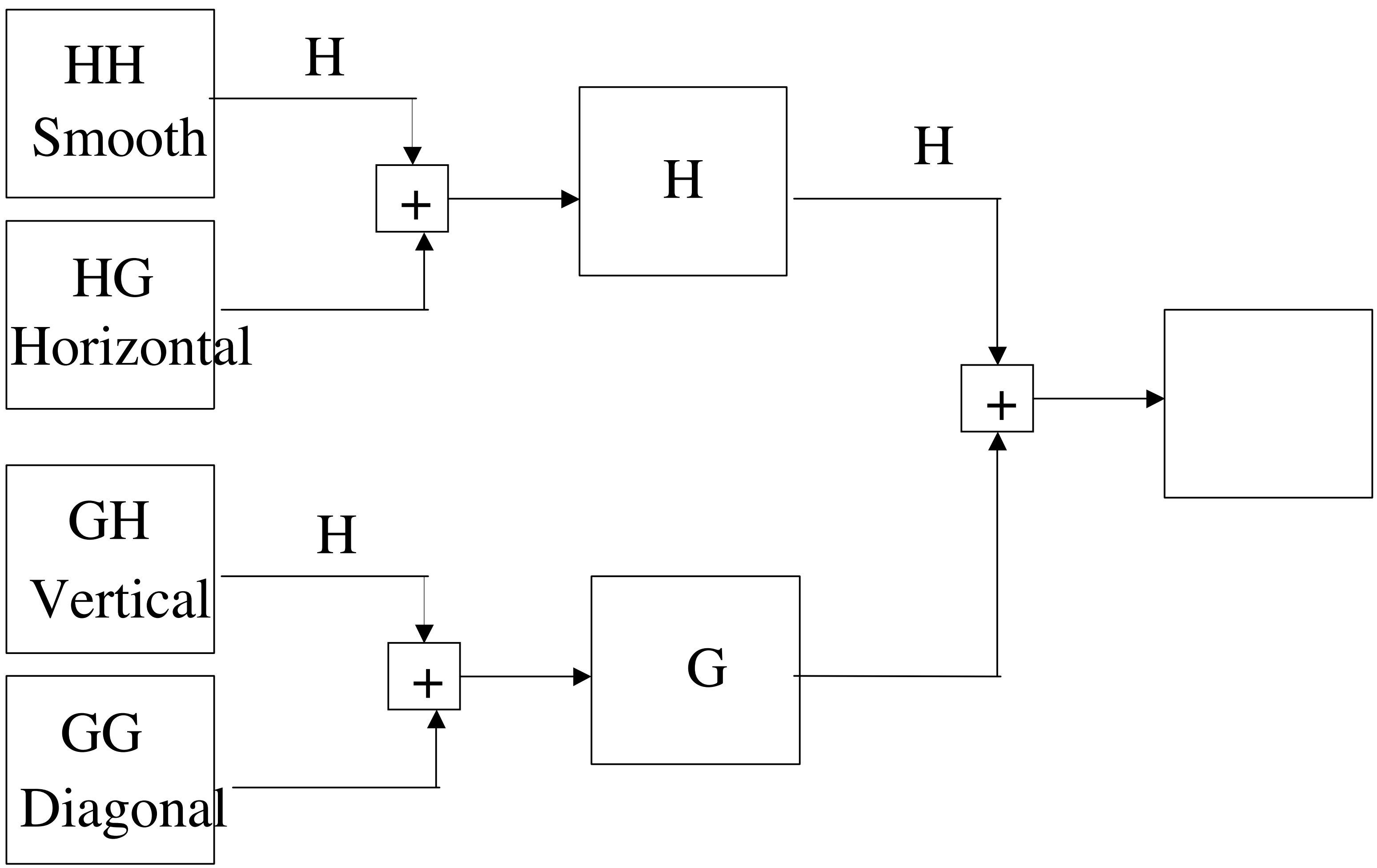

vertical
horizontal
diagonal
1
2
3
4

$$h = \tilde{h} = [\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}],$$

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

$$\tilde{g} = Id$$

Reconstruction



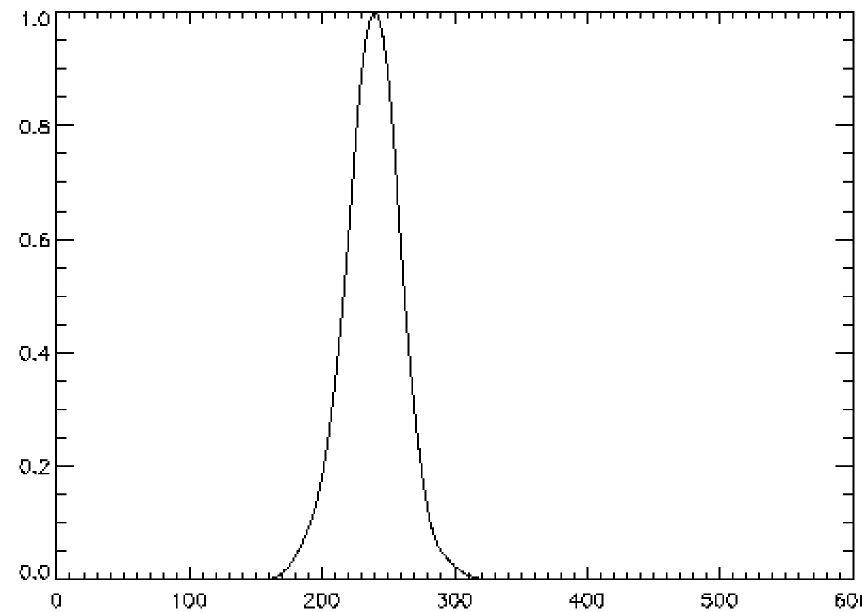
HAAR H filter with a Positive Synthesis Filter Bank

$$h = [\frac{1}{2}, \frac{1}{2}],$$

$$g = [-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}],$$

$$\tilde{h} = h * h * h = [1,3,3,1]/8$$

$$\tilde{g} = [\frac{1}{4}, \frac{6}{4}, \frac{1}{4}]$$



Sparse Representation in a Dictionary

Given a signal s, we assume that it is the result of a sparse linear combination of atoms from a known dictionary D.

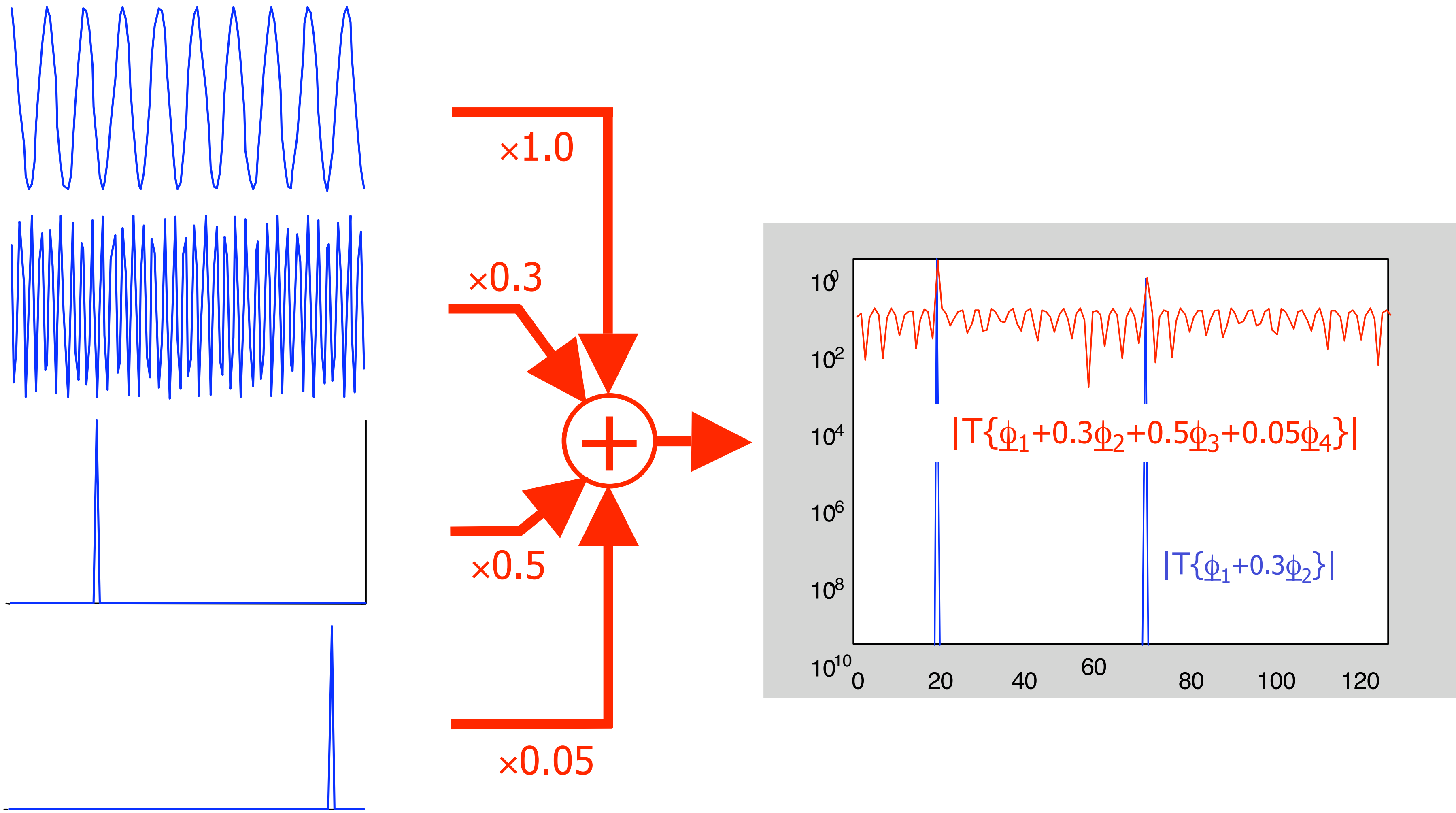
A dictionary D is defined as a collection of waveforms $(\phi_\gamma)_{\gamma \in \Gamma}$, and the goal is to obtain a representation of a signal s with a linear combination of a small number of basis such that:

$$s = \sum_\gamma \alpha_\gamma \phi_\gamma$$

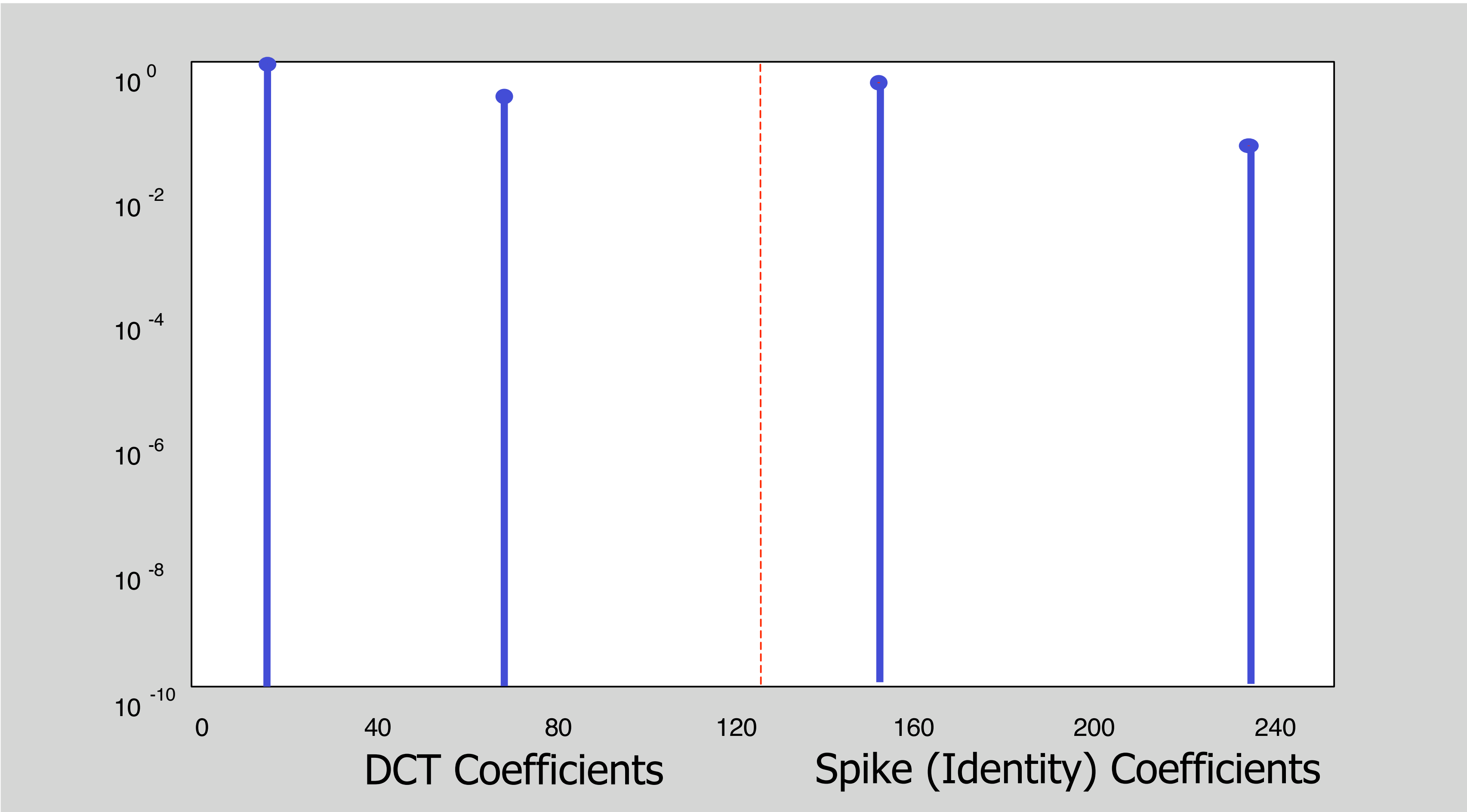
Or an approximate decomposition:

$$s = \sum_\gamma \alpha_\gamma \phi_\gamma + R$$

Example – Composed Signal



Example – Desired Decomposition



Formally, the sparsest coefficients are obtained by solving the optimization problem:

$$\text{(P0)} \quad \text{Minimize} \quad \|\alpha\|_0 \quad \text{subject to} \quad s = \phi\alpha$$

It has been proposed (*to relax and*) to replace the l_0 norm by the l_1 norm (Chen, 1995):

$$\text{(P1)} \quad \text{Minimize} \quad \|\alpha\|_1 \quad \text{subject to} \quad s = \phi\alpha$$

It can be seen as a kind of convexification of (P0).

It has been shown (Donoho and Huo, 1999) that for certain dictionary, it there exists a highly sparse solution to (P0), then it is identical to the solution of (P1).

We consider now that the dictionary is built of a set of L dictionaries related to multiscale transforms, such wavelets, ridgelet, or curvelets.

Considering L transforms, and α_k the coefficients relative to the kth transform:

$$\phi = [\phi_1, \ldots, \phi_L], \quad \alpha = \{\alpha_1, \ldots, \alpha_L\}, \quad s = \phi\alpha = \sum_{k=1}^{L} \phi_k \alpha_k$$

Noting $T_1, \ldots T_L$ the L transform operators, we have:

$$\alpha_k = T_k s_k, \qquad s_k = T_k^{-1} \alpha_k, \qquad s = \sum_{k=1}^{L} s_k$$

A solution α is obtained by minimizing a functional of the form:

$$J(\alpha) = \left\| s - \sum_{k=1}^{L} T_k^{-1} \alpha_k \right\|_2^2 + \|\alpha\|_p$$

Different Problem Formulation

$$J(s_1,\ldots,s_L) = \left\| s - \sum_{k=1}^{L} s_k \right\|_2^2 + \lambda \sum_{k=1}^{L} \left\| T_k s_k \right\|_p$$

.We do not need to keep all transforms in memory.

. There are less unknown (because we use non orthogonal transforms).

.We can easily add some constraints on a given component

Morphological Component Analysis (MCA)

$$J(s_1,\ldots,s_L) = \left\| s - \sum_{k=1}^{L} s_k \right\|_2^2 + \lambda \sum_{k=1}^{L} \left\| T_k s_k \right\|_p + \sum_{k=1}^{L} \gamma_k C_k(s_k)$$

$C_k(s_k)$ = constraint on the component s_k

An efficient algorithm is the Block-Coordinate Relaxation Algorithm (Sardy, Bruce and Tseng, 1998):

. Initialize all s_k to zero

. Iterate j=1,...,Niter

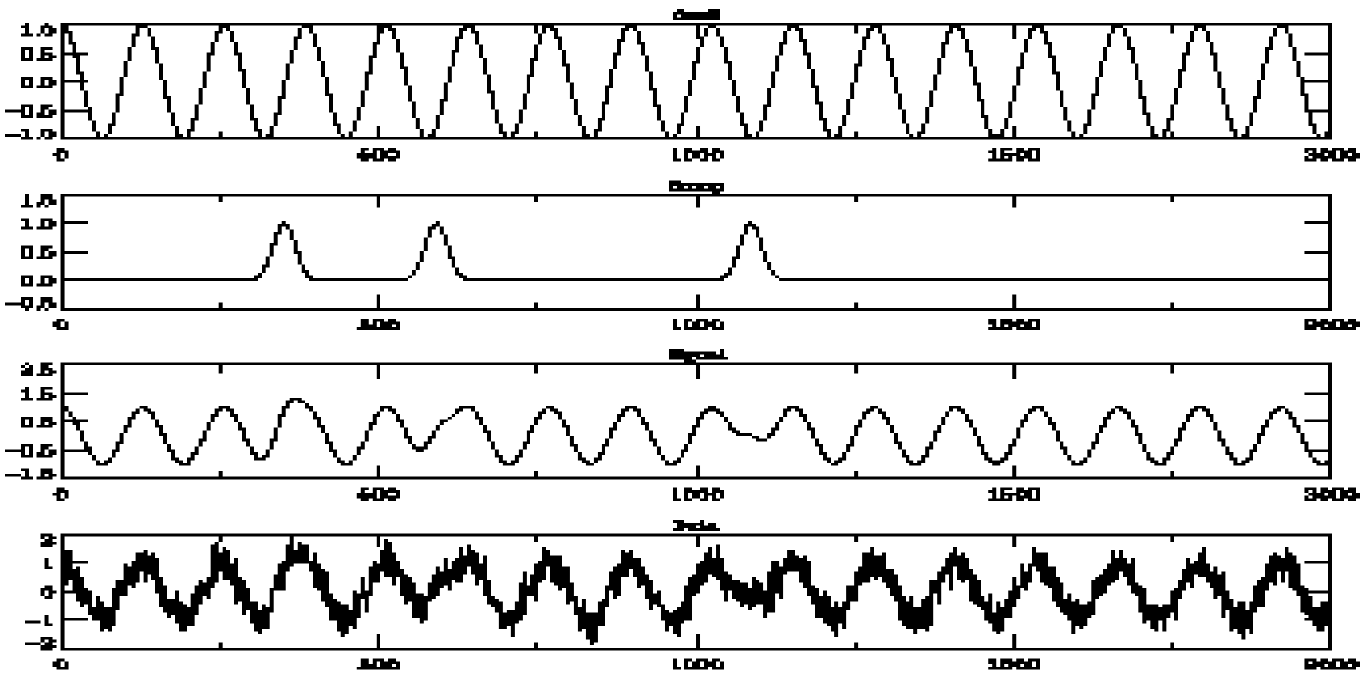
- Iterate k=1,..,L

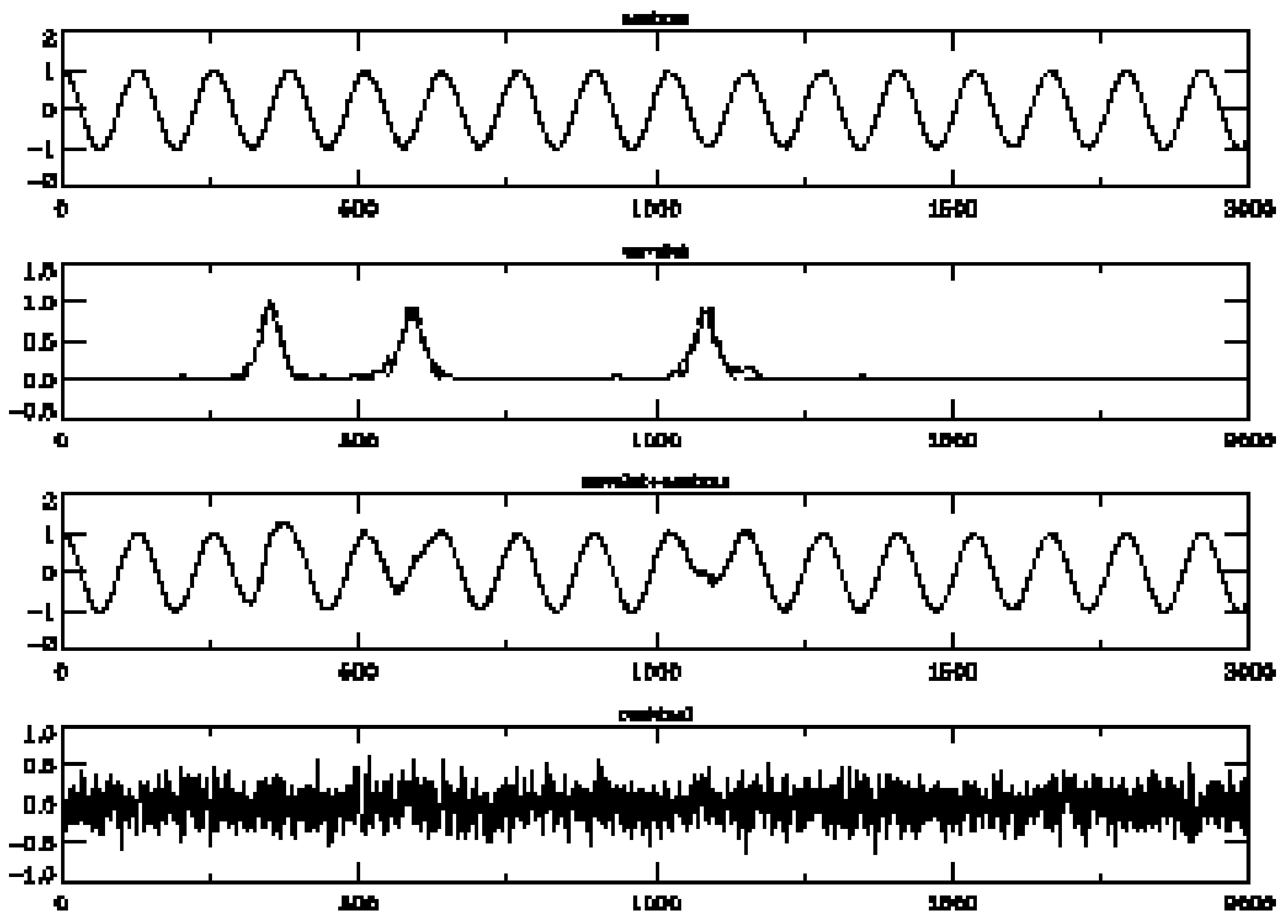
- Update the kth part of the current solution by fixing all other parts and minimizing:

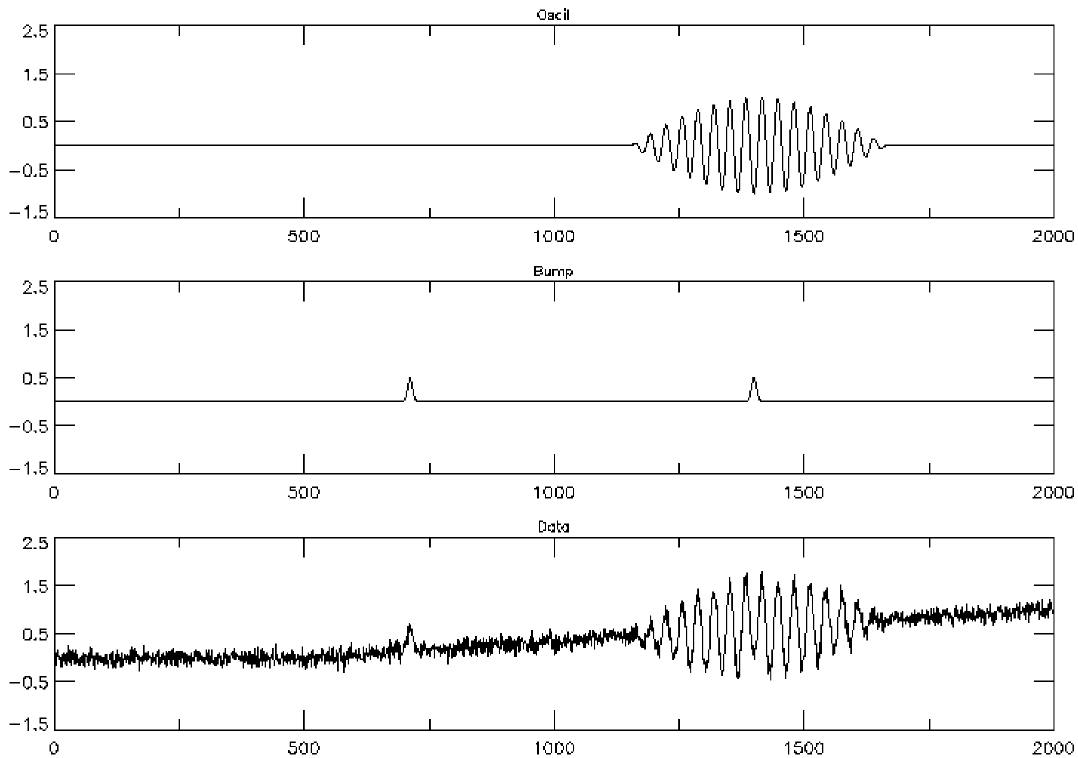
$$J(s_k) = \left\| s - \sum_{i=1, i\neq k}^{L} s_i - s_k \right\|_2^2 + \lambda \left\| T_k s_k \right\|_1$$

Which is obtained by a simple soft thresholding of :

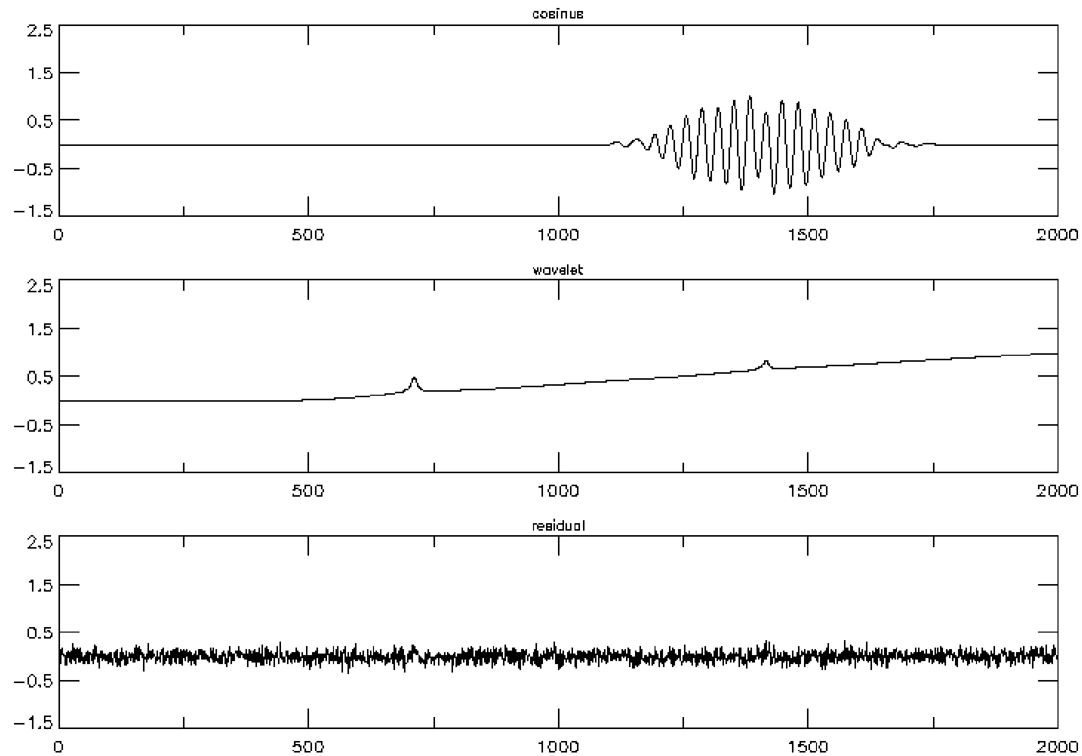
$$s_r = s - \sum_{i=1, i\neq k}^{L} s_i$$



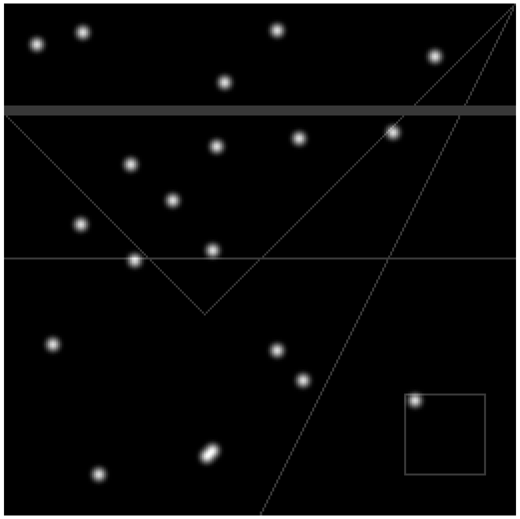




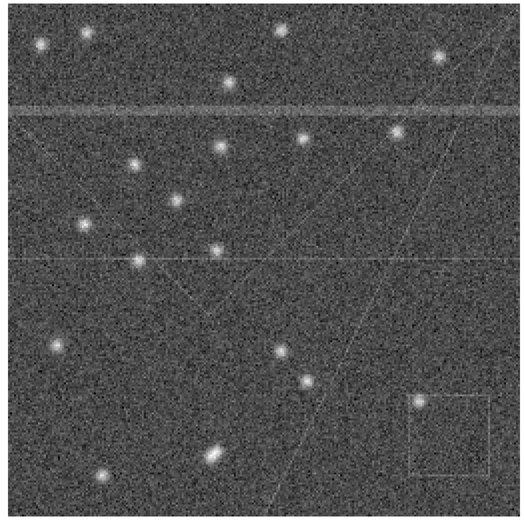
From top to bottom, oscillating component, component with bumps, and simulated data



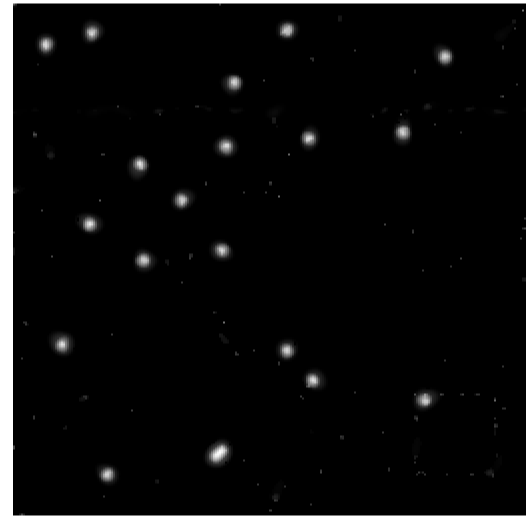
From top to bottom, reconstructed oscillating component, reconstructed component with bumps, and residual.



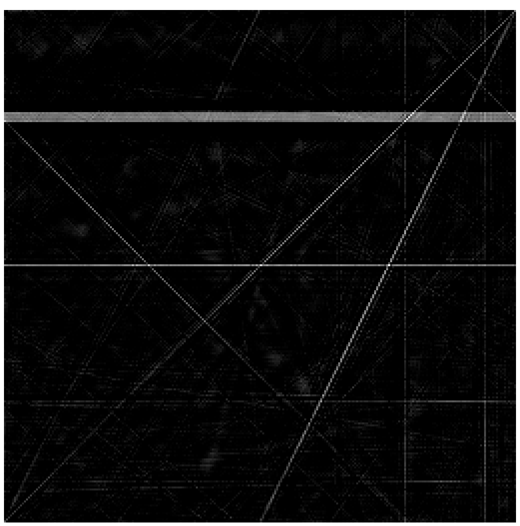
a) Simulated image (Gaussians+lines)



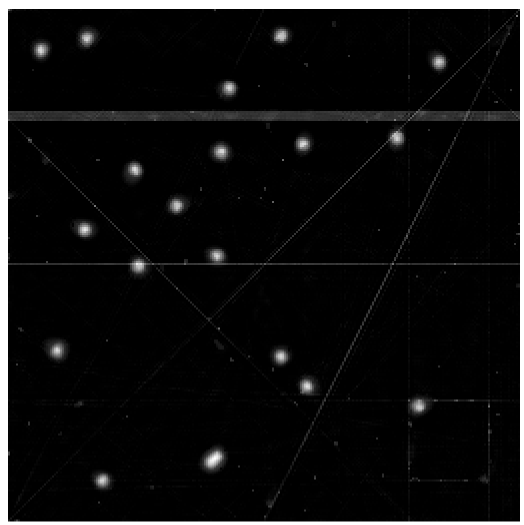
b) Simulated image + noise



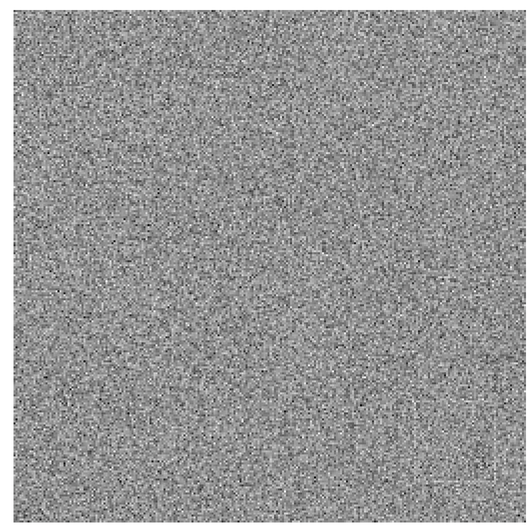
c) A trous algorithm



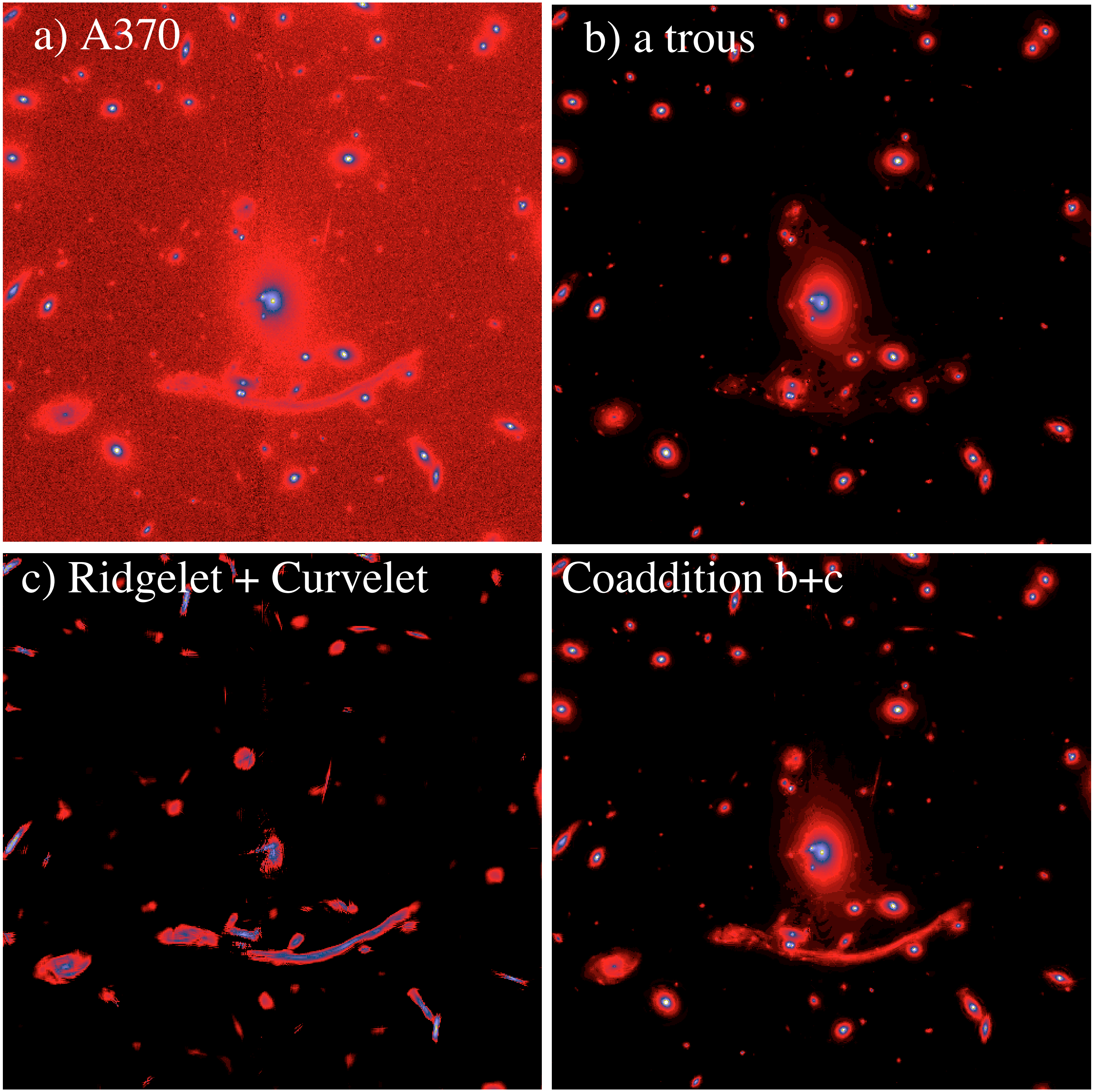
d) Curvelet transform



e) coaddition c+d



f) residual = e-b


a) A370
b) a trous
c) Ridgelet + Curvelet
Coaddition b+c

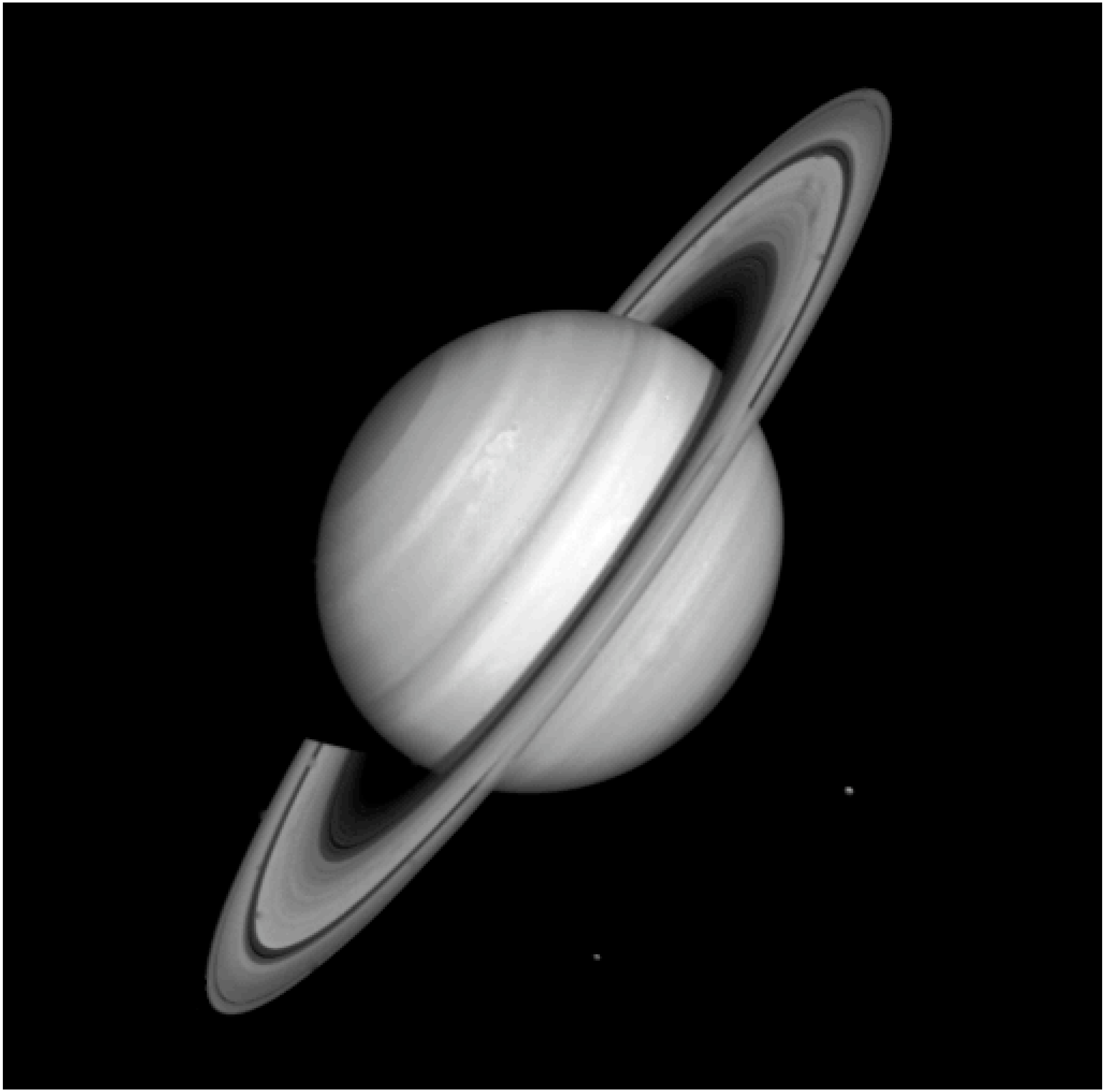
Separation of Texture from Piecewise Smooth Content

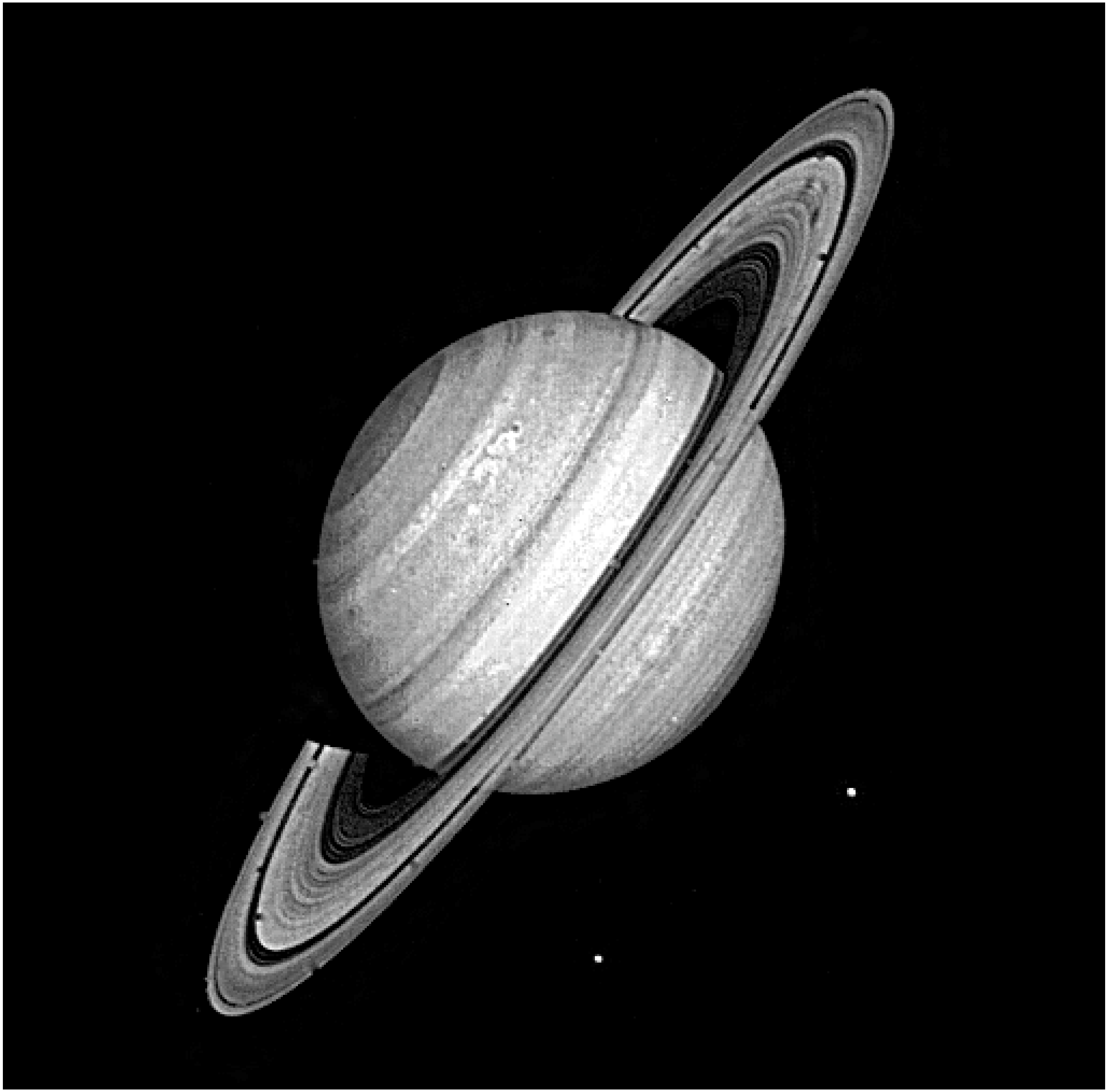
<u>The separation task</u>: decomposition of an image into a texture and a natural (piecewise smooth) scene part.

Dictionaries Choice

For the texture description (i.e. $\mathbf{T}_t$ dictionary), the DCT seems to have good properties. If the texture is not homogeneous, a local DCT should be preferred.

The curvelet transform represents well edges in an images, and should be a good candidate in many cases. The un-decimated wavelet transform could be used as well. In our experiments, we have chosen images with edges, and decided to apply the texture/signal separation using the DCT and the curvelet transform.

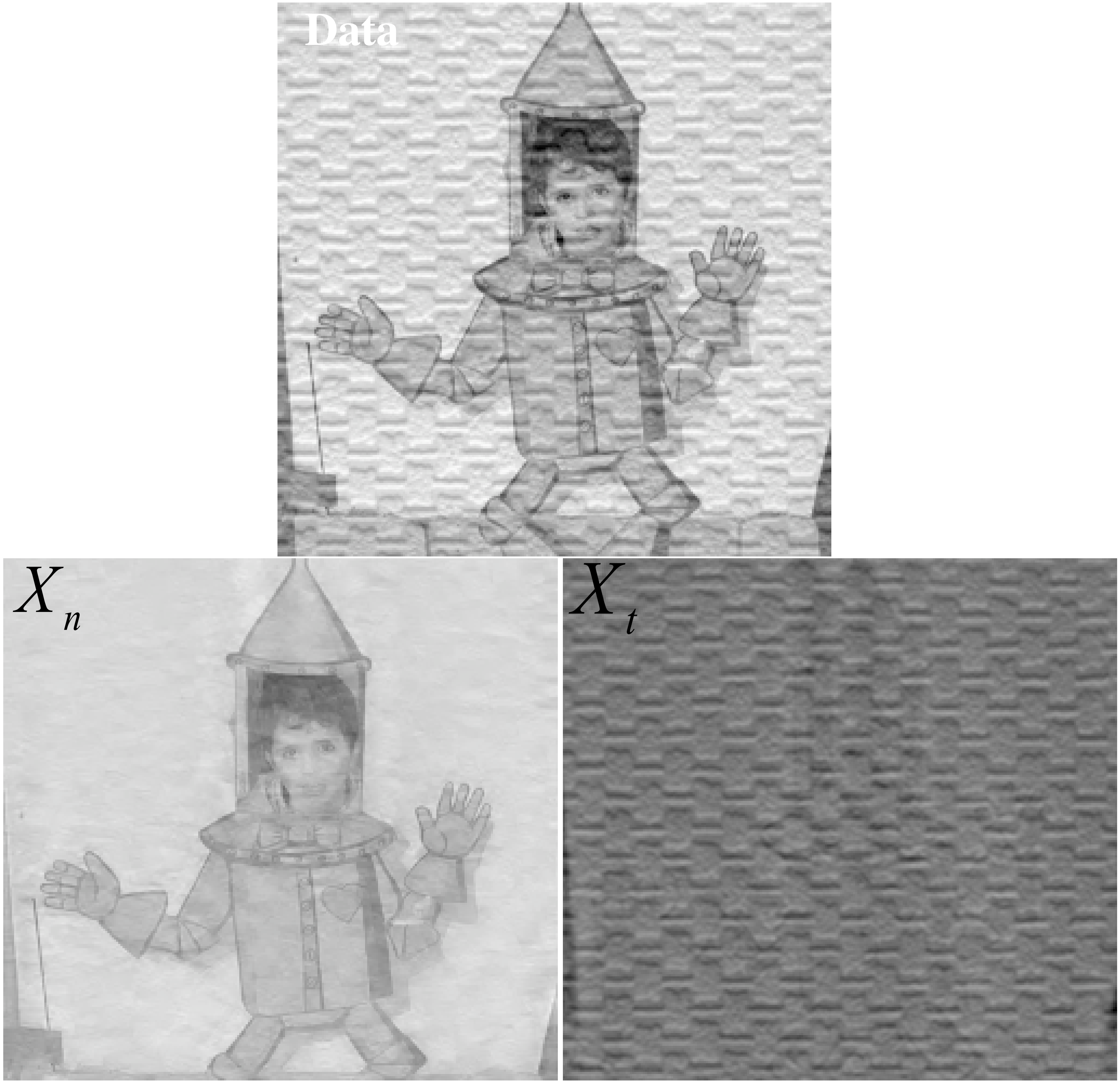




Numerical Consideration

The DCT is denoted $\mathcal{D}$ and its inverse by $\mathcal{D}^{-1}$ (with a clear abuse of notations). The curvelet transform is denoted it by $\mathcal{C}$ and its inverse by $\mathcal{C}^{-1}$. We have two unknowns - $\underline{X}_t$ and $\underline{X}_n$ - the texture and the piecewise smooth images. The optimization problem to be solved is

$$\min_{\{\underline{X}_t,\, \underline{X}_n\}} \quad \|\mathcal{D}\underline{X}_t\|_1 + \|\mathcal{C}\underline{X}_n\|_1 + \lambda \left\|\underline{X} - \underline{X}_t - \underline{X}_n\right\|_2^2 + \gamma TV\left\{\underline{X}_n\right\}.$$


Data
X_n
X_t

Edge Detection

Interpolation of Missing Data

$$J(s_1,\ldots,s_L) = \left\| M\left(s - \sum_{k=1}^{L} s_k\right) \right\|_2^2 + \lambda \sum_{k=1}^{L} \left\| T_k s_k \right\|_p$$

Where M is the mask: M(i,j) = 0 ==> missing data
M(i,j) = 1 ==> good data

If the data are composed of a piecewise smooth component + texture

$$J(X_t, X_n) = \left\| M(X - X_t - X_n) \right\|_2^2 + \lambda\left(\left\| \mathbf{C} X_n \right\|_1 + \left\| \mathbf{D} X_t \right\|_1 \right) + \gamma\, \mathrm{TV}(X_n)$$

. Initialize all s_k to zero

. Iterate j=1,...,Niter

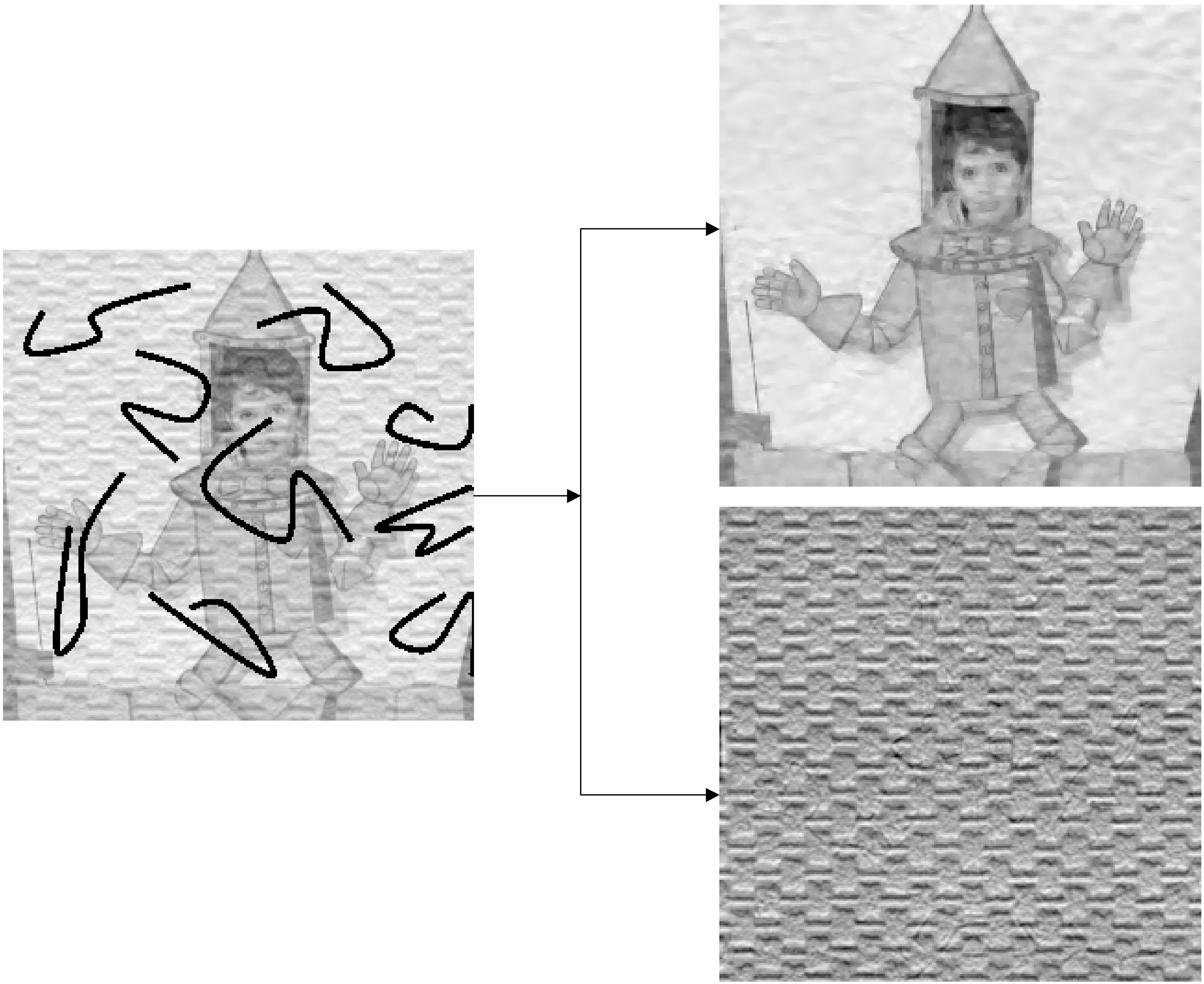
- Iterate k=1,..,L

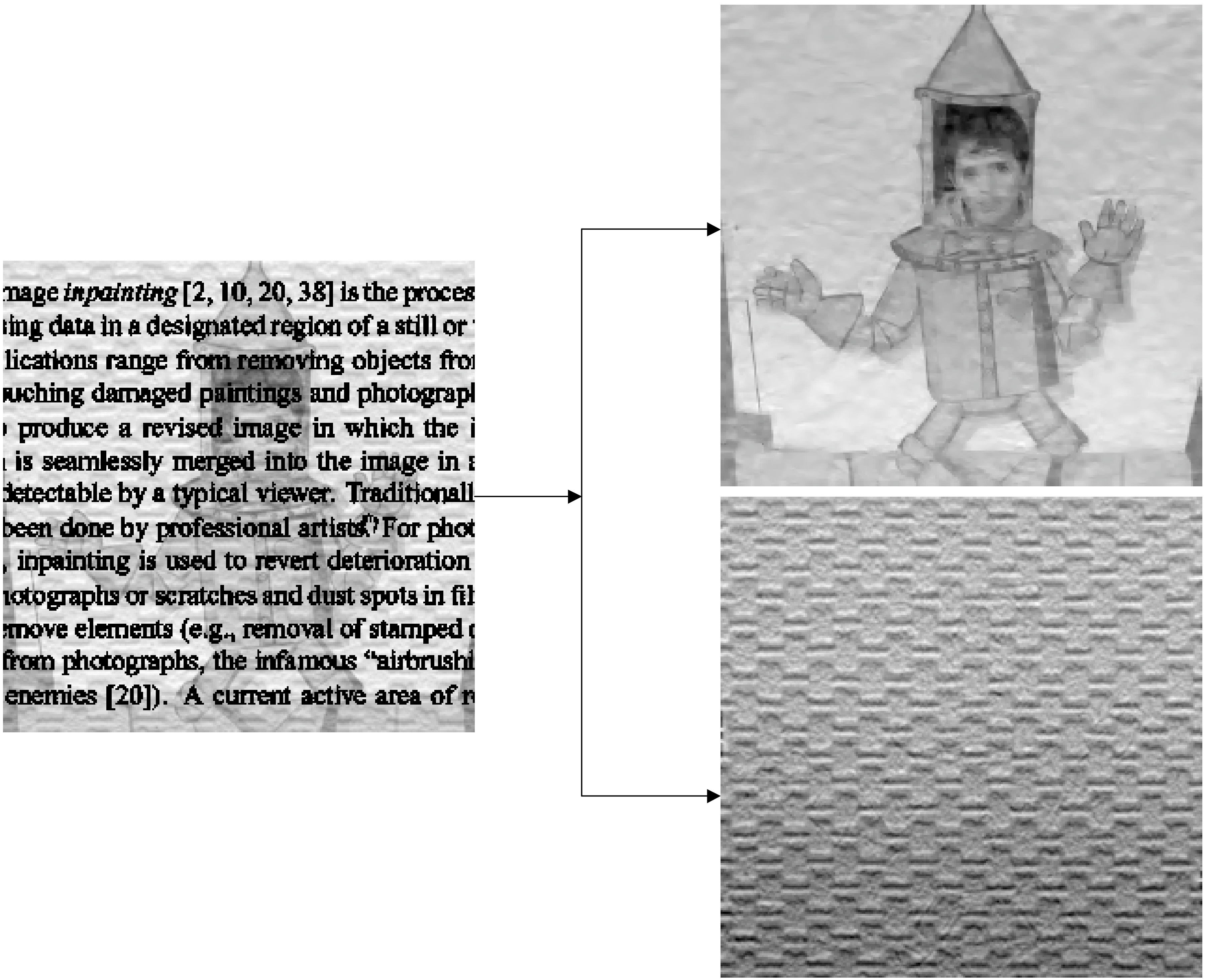
- Update the kth part of the current solution by fixing all other parts and minimizing:

$$J(s_k) = \left\| M(s - \sum_{i=1, i \neq k}^{L} s_i - s_k) \right\|_2^2 + \lambda \left\| T_k s_k \right\|_1$$

Which is obtained by a simple soft thresholding of :

$$s_r = M(s - \sum_{i=1, i \neq k}^{L} s_i)$$



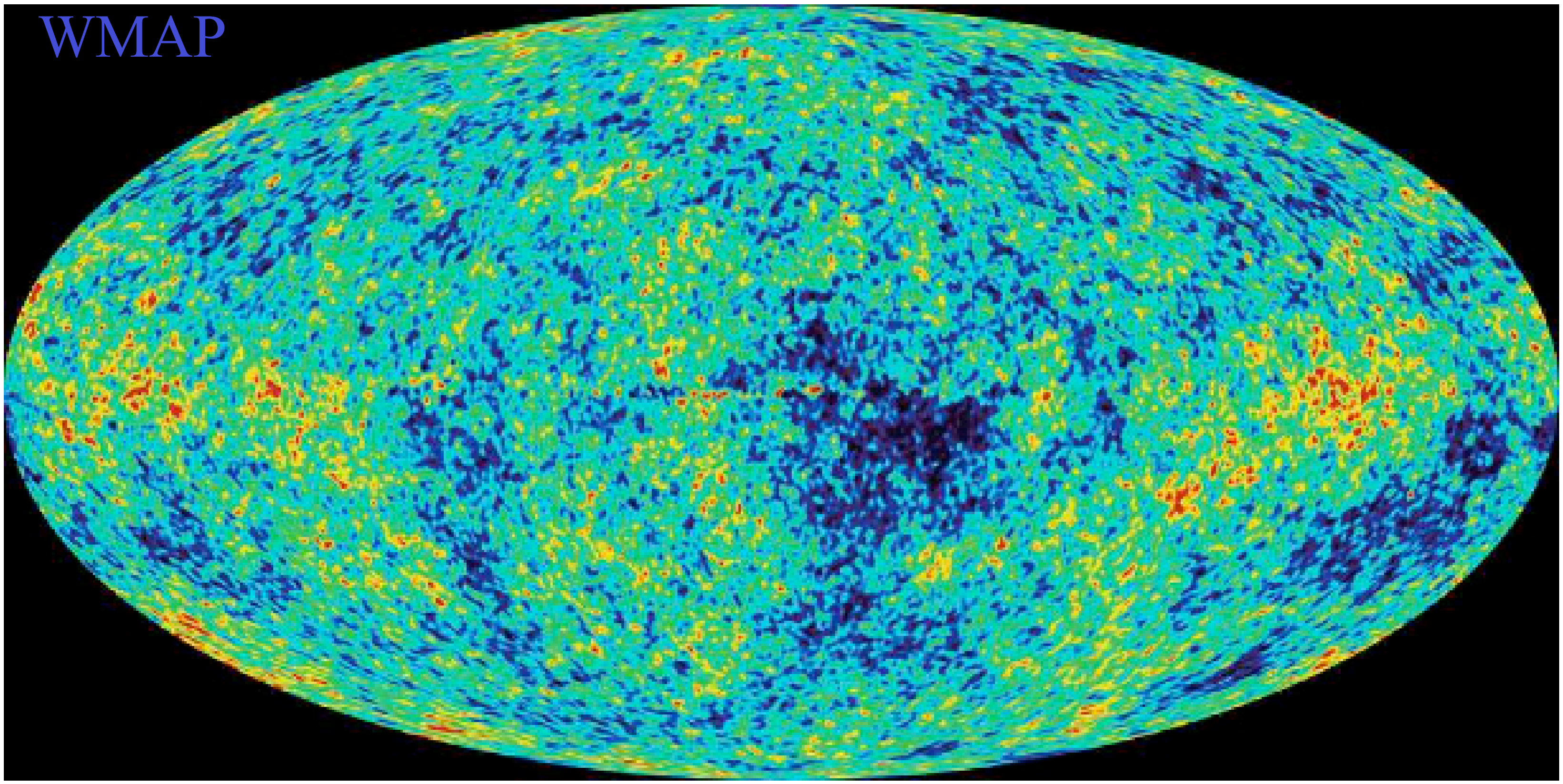

nage inpainting [2, 10, 20, 38] is the proces
ing data in a designated region of a still or
lications range from removing objects fro
uching damaged paintings and photograp
produce a revised image in which the
is seamlessly merged into the image in
detectable by a typical viewer. Traditionall
been done by professional artists. For phot
, inpainting is used to revert deterioration
otographs or scratches and dust spots in fil
move elements (e.g., removal of stamped
from photographs, the infamous "airbrushi
enemies [20]). A current active area of r

20%

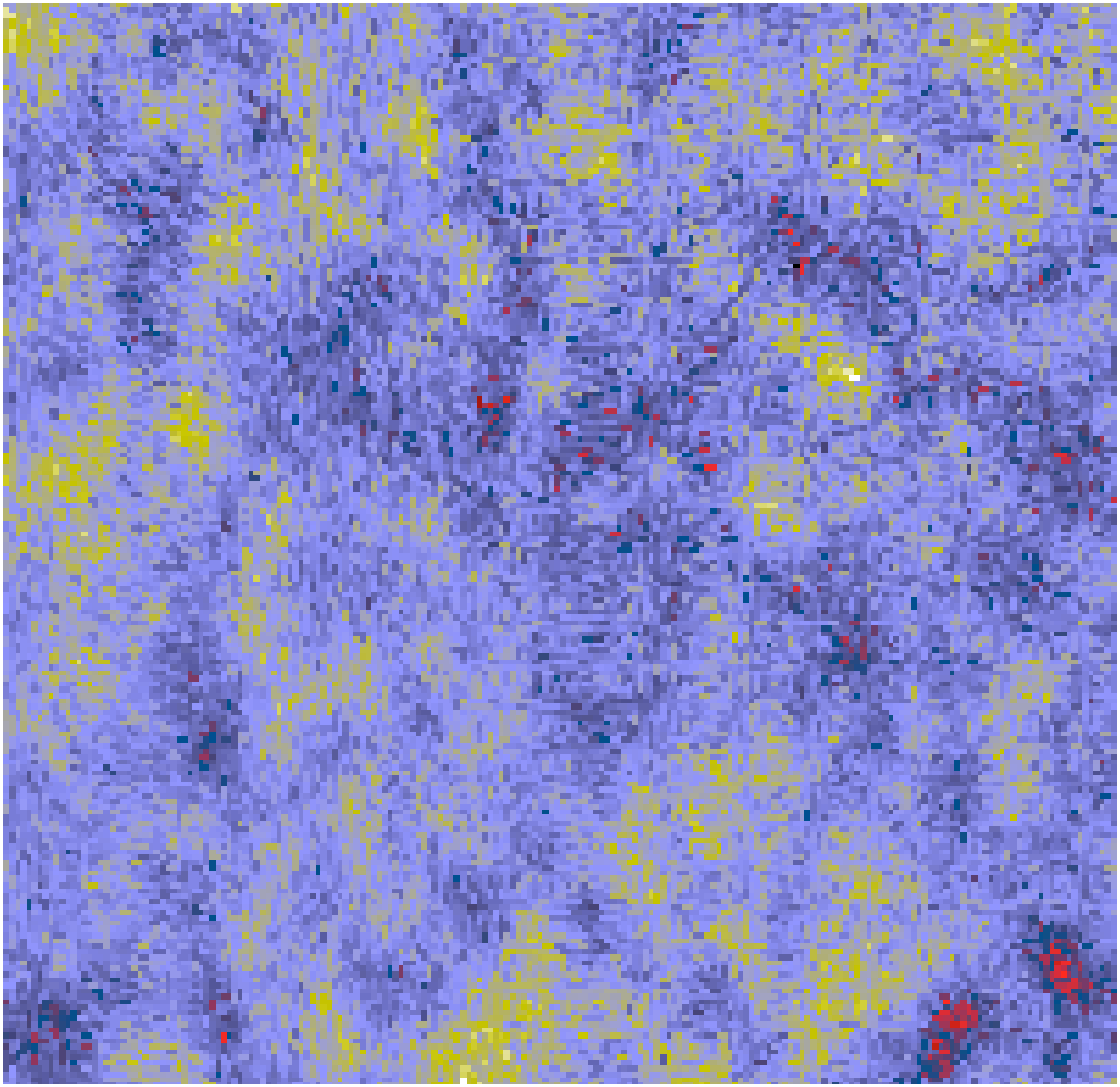
50%

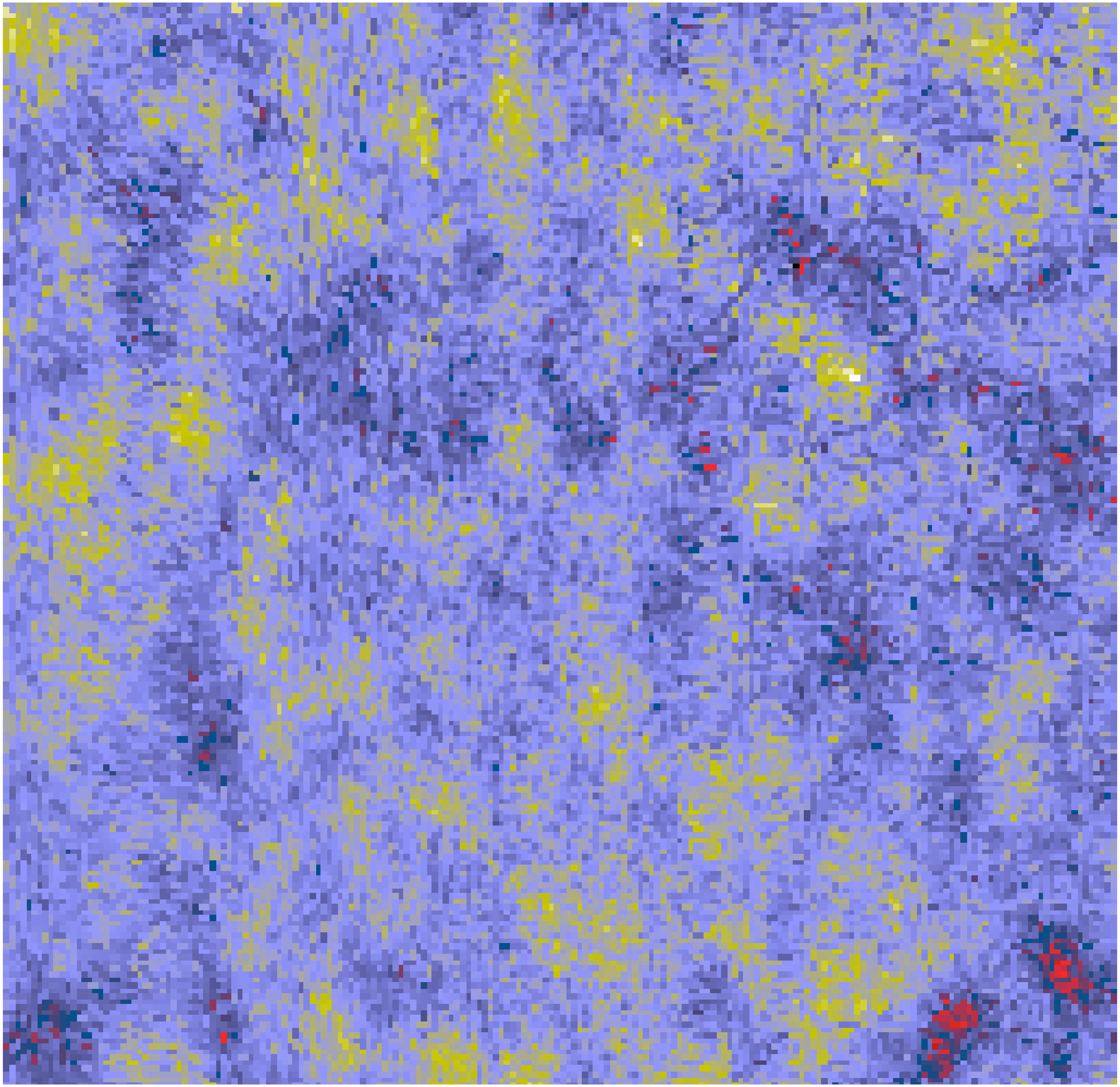
80%

Application in Cosmology



The Cosmic Microwave Background (CMB) is a relic radiation (with a temperature equals to 2.726 Kelvin) emitted 13 billion years ago when the Universe was about 370000 years old.





MCA and Blind Source Separation

The overall goal of BSS is to recover unobserved signals, images or *sources* S from mixtures X of these sources observed typically at the output of an array of sensors. The simplest mixture model would take the fo

$$X = AS$$

where X, S and A are matrices of respective sizes $n_c \times n$, $n_s \times n$ and $n_c \times n_s$. Multiplying S by A linearly mixes the n_s sources into n_c observed processes.

Multichannel MCA

As before, we assume that each source s_k is well represented (i.e. sparsified) by a given transform, but now, the observed data X are no longer the sum of sources, but a set of n_c linear combinations of the n_s sources:

$$X_l = \sum_{k=1}^{n_s} A_{k,l} s_k,$$

where $l = 1 \ldots n_c$, A is the mixing matrix and, here, s_k is the $1 \times n$ array of the k^{th} source samples.

. Initialize all s_k to zero

. Iterate j=1,...,Niter

- Iterate k=1,..,L

  - Update the kth part of the current solution by fixing all other parts and minimizing:

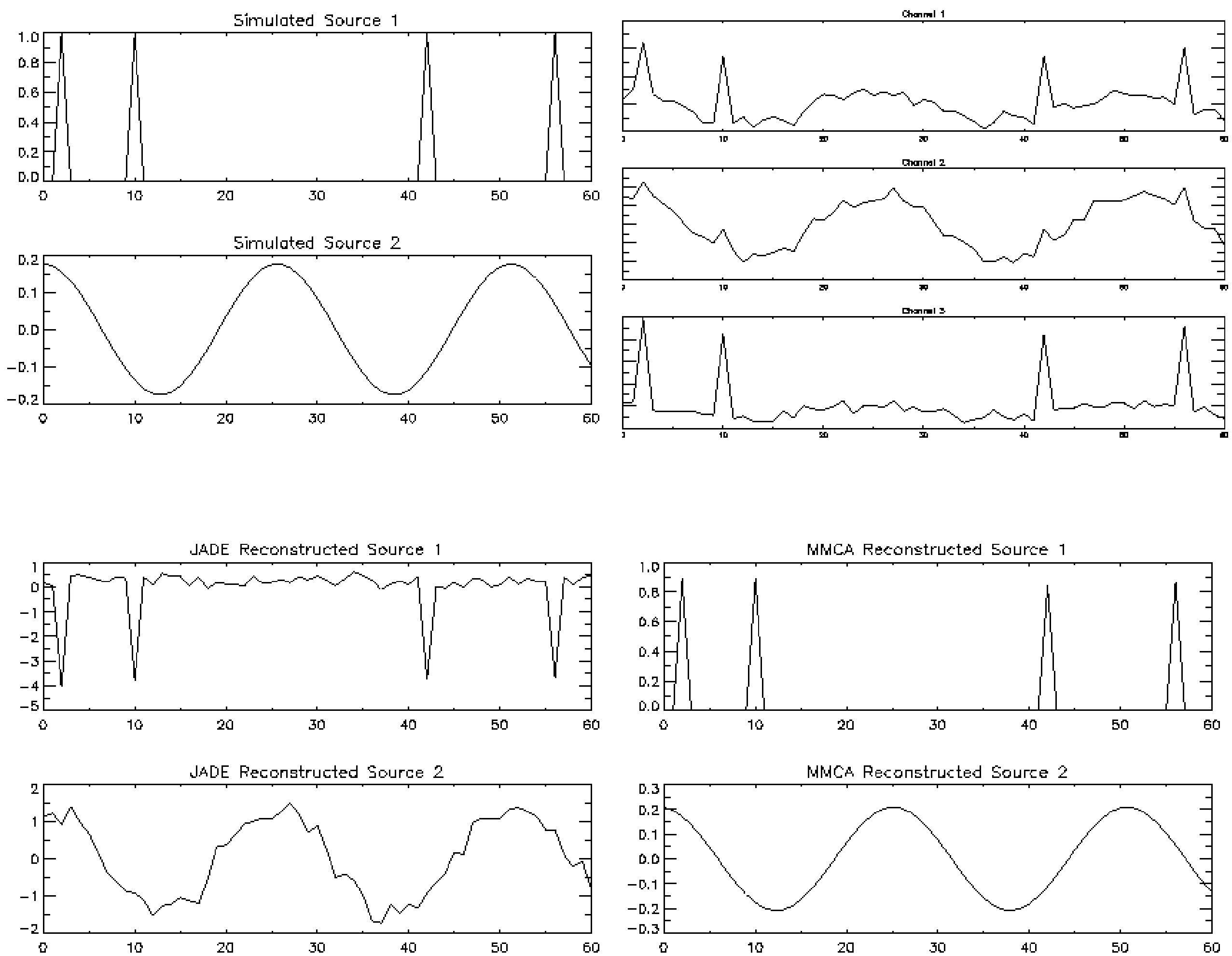
$$J(s_k) = \left\| D_k - s_k \right\|_2^2 + \lambda \left\| T_k s_k \right\|_1$$

With $D_k = a^{k^T} (X - \sum_{i=1, i \neq k}^{L} a^i s_i)$

Which is obtained by a simple soft thresholding of D_k

- estimation of a^k assuming all s_l and $a^l_{l \neq k}$ fixed

$$a^k = \frac{1}{s_k s_k^T} D_k s_k^T$$


Simulated Source 1
Channel 1
Channel 2
Simulated Source 2
Channel 3
JADE Reconstructed Source 1
MMCA Reconstructed Source 1
JADE Reconstructed Source 2
MMCA Reconstructed Source 2

Conclusions

We have seen that the MCA method can be useful in different applications such texture separation or inpainting.

.**Redundant Multiscale Transforms and their Application for Morphological Component Analysis,** *Advances in Imaging and Electron Physics, 132, 2004*.

. **Image Decomposition Via the Combination of Sparse Representation and a Variational Approach**, *IEEE Transaction on Image Processing*, in press.

. **Simultaneous Cartoon and Texture Image Inpainting using Morphological Component Analysis (MCA),** ACHA, in press.

The MMCA algorithm brings a very strong and robust component separation as long as the MMCA hypothesis is verified (sources are sparsified in different bases) i.e. for morphologically diverse sources.

The next step will be to consider the following model:

$$s_i = \sum_{k=1}^{K_i} c_{i,k} \quad \text{and} \quad X_l = \sum_{i=1}^{n} A_{i,l} s_i = \sum_{i=1}^{n} A_{i,l} \sum_{k=1}^{K_i} c_{i,k}$$

More experiments available at http://jstarck.free.fr/mca.html