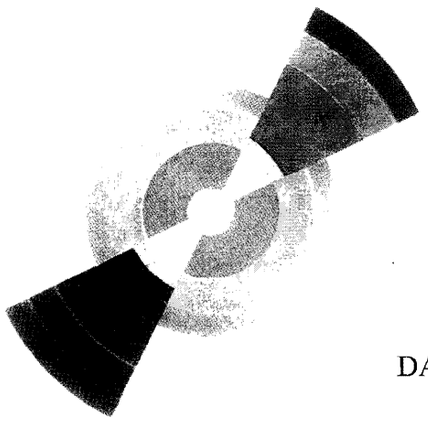
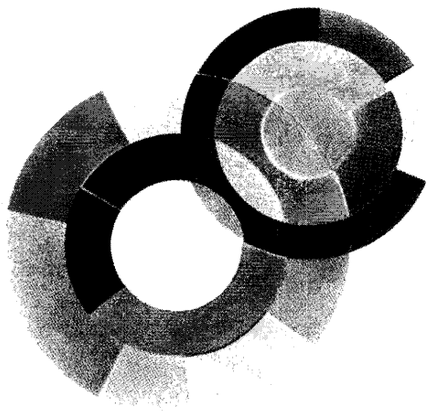
cea

DAPNIA





DAPNIA-04-88 04/2004

Redundant multiscale transforms and their application for morphological component separation.

J.L. Starck, M. Elad, D. Donoho

Submitted to Advances in Imaging and Electron Physics Series

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

Redundant Multiscale Transforms and Their Application for Morphological Component Separation

Jean-Luc Starck, Michael Elad and David Donoho [a,b,c]

[a] *DAPNIA/SEDI-SAP, Service d'Astrophysique, CEA-Saclay, 91191 Gif sur Yvette, France*

[b] *The Computer Science Department - The Technion, Israel Institute of Technology, Haifa 32000, Israel*

[c] *Department of Statistics, Stanford University, Sequoia Hall, Stanford, CA 94305 USA.*

Abstract

The idea to morphologically decompose a signal into its building blocks is an important problem in signal and image processing. Successful separation of a signal content has a key role in the ability to effectively manipulate it. Various approaches have been proposed in recent years to tackle this problem. In this paper we describe a novel decomposition method – *Morphological Component Analysis* (MCA) – based on sparse representation of signals. This method relies on the assumption that for every signal atomic behavior to be separated, there exists a dictionary that enables its construction using a sparse representation. Also, it is assumed that the different dictionaries are highly inefficient in representing the other behaviors in the mixture. If such dictionaries are identified, the use of a Pursuit algorithm searching for the sparsest representation leads to the desired separation.

Our discussion in this paper includes a theoretical justification for the separation success, several application results on image content, and discussion on efficient numerical algorithms to facilitate the proposed algorithm. Also, this paper contains a broad overview of redundant multiscale transforms such as the un-decimated Wavelet Transform, the Ridgelet and the Curvelet transforms, and more. This wide survey supplies the background material that leads to and support the MCA method, and provides an extensive list of possible dictionaries to be used within the MCA.

Key words: Wavelet, Ridgelet, Curvelet, Sparse Representation, Over-Complete Dictionary, Basis Pursuit, matching Pursuit, Signal Separation, Atomic Decomposition.

1 Introduction

1.1 Sparsity and Redundancy in Signal Representation

Alternative representation of signals via transforms are appealing due to the simplicity and efficiency they induce in various applications. In the quest for a proper transform, the wavelet family of tools attracted a lot of research attention due to the natural way the multi–resolution aspect of the signals is taken into consideration, and the efficiency gained because of this. Various variants of the core wavelet method were proposed in recent years, all in the constant interest to find a better representation for the signals in mind. As an example, the Ridgelet and the Curvelet algorithms were developed as an answer to the weakness of the separable Wavelet in representing lines and curves in 2D signals (images). This weakness is exhibited by the many coefficients required in representing what appears to be a simple atomic behavior in an image (Candès, 1998; Donoho and Duncan, 2000; Candès and Donoho, 1999a; Starck et al., 2002).

In this evolution of transforms, sparsity of the representation was recognized as a promising guideline in seeking simplifying operation. This is especially true for over-complete redundant representations as commonly employed in many of the wavelet methods. The basic idea here is a construction of the signal as a linear combination of atoms from a dictionary, where the number of atoms in the dictionary is (much) bigger than the signal dimension, thus introducing redundancy. Due to this over-completeness, there are numerous ways to represent the signal, and among those, preference is made towards the one with the fewest non-zero entries (sparsest) as being the simplest. Clearly, while linear in the construction of the signal from its representation, this transform is non-linear in converting the signal to the representation coefficients. Two well-known algorithms to implement this non-linear forward transform are the Matching Pursuit (MP) (Mallat and Zhang, 1993) and the Basis Pursuit (BP) (Chen et al., 1998), both imposing sparsity.

In this paper we record the above-described development–track of the wavelet transform and its redundant extensions designed for images. We also study the notion of sparsity and the algorithms that facilitate it. All this is presented as the background material to the main theme of this paper being signal decomposition.

1.2 *The Morphological Component Analysis*

The ida to morphologically decompose a signal into its building blocks is an important problem in signal and image processing. Successful separation of a signal content has a key role in the ability to effectively analyze it, enhance it, compress it, synthesize it, and more. Various approaches have been proposed to tackle this problem. The vast literature on Blind-Source Separation (BSS) and Independent Component Analysis (ICA) is a convincing testimony both to the importance and the complexity of the signal separation problem – see (Hyvarinen et al., 2001; Haykin, 2001; Cichocki and Amari, 2002) for representative survey works. Interestingly, sparsity was also recognized as a possible feature to rely on in signal separation, and the relation between sparsity and independence has been vaguely understood (Kreutz-Delgado and Rao, 1999; Kisilev et al., 2001; Zibulevsky and Pearlmutter, 2001).

In this paper we propose a general view to the signal separation arena from the sparsity point of view, and propose a methodology for the separation based on redundant transforms. We argue that if proper dictionaries are chosen for the various signal contents, separation can be driven by sparsity, leading to appealing results. The presented method relies on the assumption that for every signal atomic behavior to be separated, there exists a dictionary that enables its construction using a sparse representation. Furthermore, it is assumed that the different dictionaries are highly inefficient in representing the other behaviors. Assuming that such dictionaries are identified, the use of the Basis Pursuit (BP) or the Matching Pursuit (MP) algorithms lead to the desired separation. We demonstrate this on several applications and suggest a rigorous analysis to explain the reasons to its success.

The numerical separation method proposed in this paper, coined *Morphological Component Analysis* (MCA), could be regarded as a hybridization of the Basis Pursuit and the Matching Pursuit methods, and as such, as a general signal transform that is capable of creating representations containing as a by-product a decoupling of the signal content.

1.3 *Paper Organization*

In Sections 2 and 3 we give the background for this work, surveying the current state-of-the-art in the fields of Wavelets and its extensions to transforms on images (Ridgelets and Curvelets). All these are described as candidate dictionaries to be used later on by the MCA method. The remaining of this paper does not rely strongly on these two sections, and so they can be skipped by the readers interested in the separation topic alone.

Section 4 provides a discussion on the migration from linear to non-linear transforms, advocating sparsity based on the Basis Pursuit and Matching Pursuit algorithms. This section constructs the theoretical and practical foundations for the MCA separation mechanism to be described next.

In Section 5 we present the Morphological Component Analysis (MCA) methodology, starting from its intuitive backbone, through a theoretical justification, and finally applications employing this idea. We also discuss numerical considerations which are vital for the success of this method in practice.

2 background - Part I - Wavelet

2.1 The Wavelet transform

Multiscale methods have become very popular, esspecially with the development of the wavelets in the last decade. Background texts on the wavelet transform include (Daubechies, 1992; Strang and Nguyen, 1996; Mallat, 1998; Starck et al., 1998; Cohen, 2003). The most used wavelet transform algorithm is certainly the decimated bi-orthogonal wavelet transform (OWT). Using the OWT, a signal $\underline{s}$ 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 and the wavelet functions. J is the number of resolutions used in the decomposition, w_j the wavelet (or details) coefficients at scale j, c_J is a coarse or smooth version of the original signal $\underline{s}$, and l stands for the sample number. Thus, the algorithm outputs $J + 1$ sub-band arrays. The indexing is such that, here, $j = 1$ corresponds to the finest scale (high frequencies). The coefficients $c_{j,k}$ and $w_{j,k}$ are obtained by means of the filters h and g, through

$$\begin{aligned} c_{j+1,l} &= \sum_k h_{k-2l} c_{j,k} = (\bar{h} * c_j)_{2l} \\ w_{j+1,l} &= \sum_k g_{k-2l} c_{j,k} = (\bar{g} * c_j)_{2l}. \end{aligned} \tag{2}$$

The notation $()_{2l}$ stands for the decimation (i.e. only even pixels are kept), $\bar{h}(l) = h(-l)$, and h and g filters satisfy

$$\frac{1}{\sqrt{2}}\phi(\frac{x}{2}) = \sum_k h_k \phi(x-k)$$
$$\frac{1}{\sqrt{2}}\psi(\frac{x}{2}) = \sum_k g_k \phi(x-k). \tag{3}$$

The smooth coefficients $c_{j+1,l}$ and the wavelet coefficients $w_{j+1,l}$ are calculated by convolving $c_{j,l}$ with the filters $\bar{h}$ and $\bar{g}$ respectively, and decimating the results. $\underline{c}_0$ corresponds to the input data (i.e. $\underline{c}_0 = \underline{s}$). Handling boundaries is typically done by the mirror assumption $c_{j,k+N} = c_{j,N-k}$ (N being the number of samples), but other methods can be used, such as periodicity ($c_{j,k+N} = c_{j,k}$), or continuity ($c_{j,k+N} = c_{j,N}$).

The reconstruction of the signal is performed by

$$c_{j,l} = \sum_k \tilde{h}_{k+2l} c_{j+1,k} + \tilde{g}_{k+2l} w_{j+1,k} = \tilde{h} * \check{c}_{j+1} + \tilde{g} * \check{w}_{j+1} \tag{4}$$

where $\check{c}_{j+1,l}$ is equal to $c_{j+1,p}$ is $l = 2p$ (i.e. l is even) and 0 otherwise (for example, $\underline{\check{c}}_j = (c_{j,0}, 0, c_{j,1}, 0, c_{j,2}, 0, c_{j,3}, 0, \cdots)$). The filters $\tilde{h}$ and $\tilde{g}$ must verify the conditions of de-aliasing and exact reconstruction,

$$\hat{h}(\nu + \frac{1}{2})\hat{\tilde{h}}(\nu) + \hat{g}(\nu + \frac{1}{2})\hat{\tilde{g}}(\nu) = 0$$
$$\hat{h}(\nu)\hat{\tilde{h}} + \hat{g}(\nu)\hat{\tilde{g}}(\nu) = 1. \tag{5}$$

We should note that the above description of the OWT construction, while quite brief, is far from trivial. The innocent reader should not expect to get the complete picture about the wavelet transform from it. Most of the above relations are difficult due to the bi-orthonormality imposed, implying that a structured and very simple method exists to invert the transform. The main features we would like to draw the reader's attention to are (i) the linearity of the transform; (ii) its simple computation via filtering and decimation; and (iii) its natural multi-scale nature.

The two-dimensional algorithm is based on separate variables leading to prioritizing of horizontal, vertical and diagonal directions. The detail signal is obtained from three wavelets:

- vertical wavelet : $\psi^1(x,y) = \phi(x)\psi(y)$,
- horizontal wavelet: $\psi^2(x,y) = \psi(x)\phi(y)$,
- diagonal wavelet: $\psi^3(x,y) = \psi(x)\psi(y)$,

which leads to three wavelet sub-images at each resolution level. The scaling function is defined by $\phi(x,y) = \phi(x)\phi(y)$, and the passage from one resolution

to the next is achieved by

$$\begin{aligned} c_{j+1,k,l} &= (\bar{h}\bar{h} * c_j)_{2k,2l} \\ w_{j+1,1,k,l} &= (\bar{g}\bar{h} * c_j)_{2k,2l} \\ w_{j+1,2,k,l} &= (\bar{h}\bar{g} * c_j)_{2k,2l} \\ w_{j+1,3,k,l} &= (\bar{g}\bar{g} * c_j)_{2k,2l}, \end{aligned} \tag{6}$$

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

$$c_{j,k,l} = \tilde{h}\tilde{h} * \breve{c}_{j+1} + \tilde{g}\tilde{h} * \breve{w}_{j+1,1} + \tilde{h}\tilde{g} * \breve{w}_{j+1,2} + \tilde{g}\tilde{g} * \breve{w}_{j+1,3} \tag{7}$$

in a similar way to the one-dimensional case, and with the proper generalization to 2D.

Figure 1 shows the image `Einstein` (top right), the schematic separation of the wavelet decomposition bands (top left), and the actual OWT coefficients (bottom left), using the 7-9 filters (Antonini et al., 1992).

The application of the OWT to image compression, using the 7-9 filters (Antonini et al., 1992) leads to impressive results, compared to previous methods like JPEG. The recent inclusion of the wavelet transform in JPEG-2000, the new still-picture compression standard, testifies to this lasting and significant impact. Figure 1 bottom right shows the decompressed image for a compression ratio of 4, and as can be seen, the result is near-perfect.

2.2 The Undecimated Wavelet Transform (UWT)

While the bi-orthogonal wavelet transform led to a successful implementation in image compression, results were far from optimal for other applications such as filtering, deconvolution, detection, or more generally, analysis of data. This is mainly due to the lose of the translation–invariance property in the OWT, leading to a large number of artifacts when an image is reconstructed after modification of its wavelet coefficients.

For this reason, some physicians and astronomers have preferred to continue working with the continuous wavelet transform (Antoine and Murenzi, 1994; Arneodo et al., 1995), even if the price to pay were (i) a huge amount of redundancy in the transformation (i.e. there are much more pixels in the transformed data than in the input image) and (ii) there is no reconstruction operator (i.e. an image cannot be reconstructed from its coefficients). For some applications

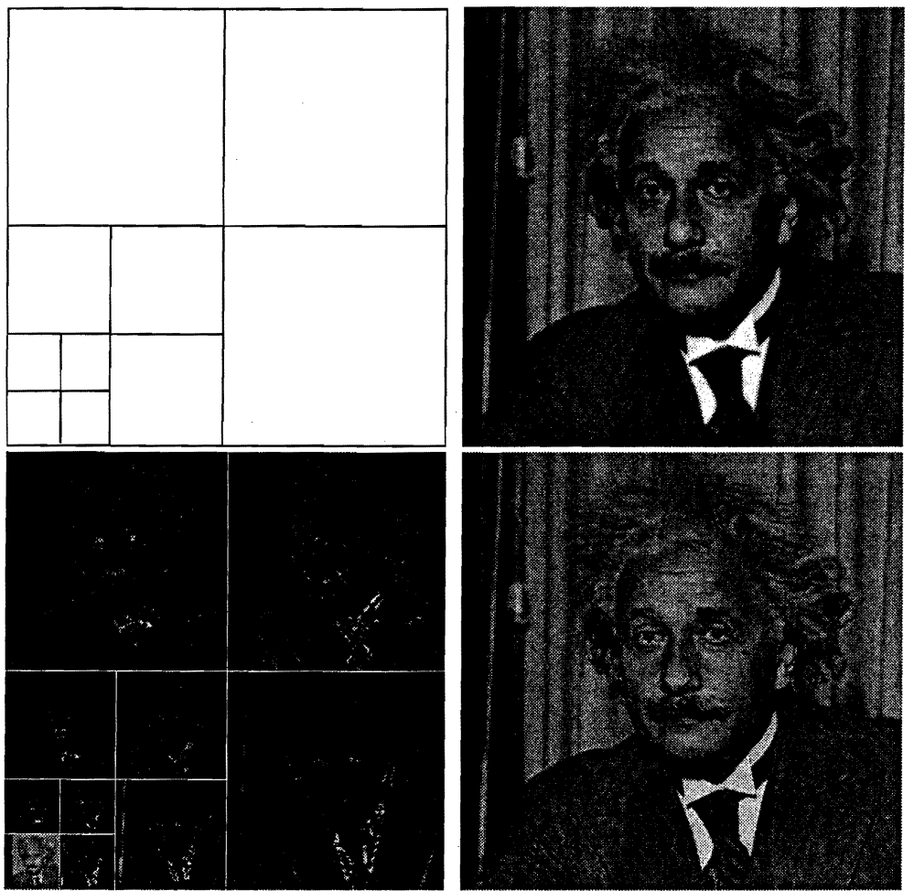


Fig. 1. The image `Einstein` (top right), its OWT wavelet decomposition (schematic - top left, and coefficients - bottom left). The bottom right image is the result of the compression-decompression JPEG-2000 algorithm, employing the 7-9 bi-orthogonal OWT, using a compression ratio of 40.

like fractal analysis, these drawbacks has no impact because there is no need to apply a reconstruction and the computers can support the redundancy. For other applications were a reconstruction is needed, some researchers have chosen an intermediate approach, keeping the filter bank construction giving a fast and dyadic algorithms, but eliminating the decimation step in the orthogonal wavelet transform (Dutilleux, 1987; Holschneider et al., 1989): $c_1 = \bar{h} * c_0$ and $w_1 = \bar{g} * c_0$. By separating even an odd pixels in c_1 and w_1, we get (c_1^E, w_1^E) and (c_1^O, w_1^O), and both part obviously allows us to reconstruct perfectly c_0. The reconstruction can be obtained by

$$c_0 = \frac{1}{2}(\tilde{h} * c_1^E + \tilde{g} * w_1^E + \tilde{h} * c_1^O + \tilde{g} * w_1^O). \tag{8}$$

For the passage to the next resolution, both c_1^E and c_1^O are decomposed, leading after the splitting into even an odd pixels to four coarse arrays associated to c_2. All of the four data set can again be decomposed in order to obtain the third decomposition level, and so on.

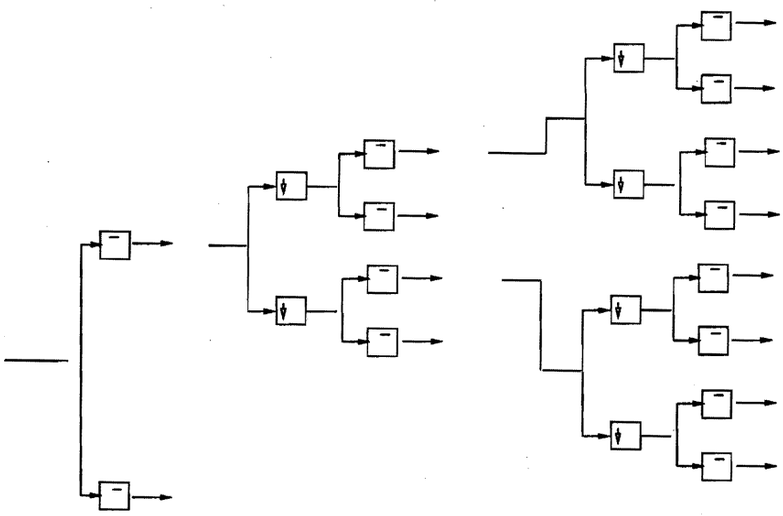


Fig. 2. 1D undecimated wavelet transform.

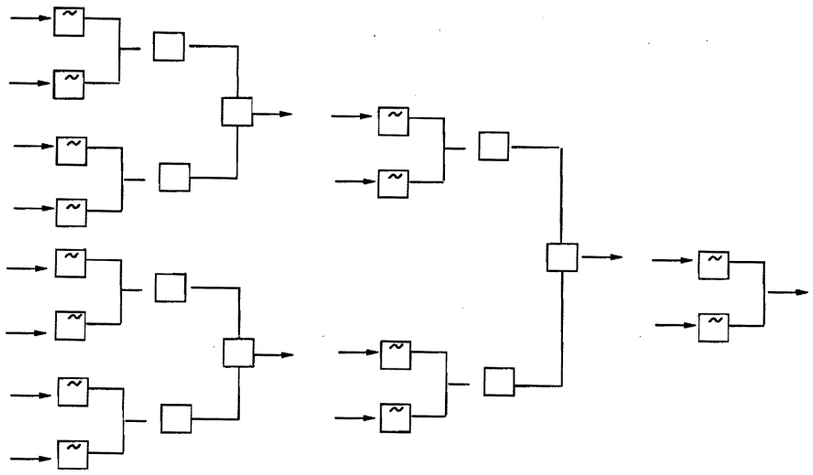


Fig. 3. 1D undecimated wavelet reconstruction.

Figure 2 shows the 1D undecimated wavelet transform (UWT) decomposition. The decimation step is not applied and both w_1 and c_1 have the same size

as c_0. c_1 is then splitted into c_1^E (even pixels) and c_1^O (odd pixels), and the same decomposition is then applied to both c_1^E and c_1^O. c_1^E produces $c_{2,1}$ and $w_{2,1}$, while c_1^O produces $c_{2,2}$ and $w_{2,2}$. $w_2 = \{w_{2,1}, w_{2,1}\}$ contains the wavelet coefficients at the second scale, and is also of the same size as c_0. Figure 3 shows the 1D UWT reconstruction.

It is clear that this approach is much more complicated than the decimated bi-orthogonal wavelet transform. There exists, however, a very efficient way to implement it, called the "à trous" algorithm ("à trous" is a French word which means *with holes*). This method considers the filter $h^{(j)}$ instead of h where $h_l^{(j)} = h_l$ 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 as

$$\begin{aligned} c_{j+1,l} &= (\bar{h}^{(j)} * c_j)_l = \sum_k h_k c_{j,l+2^j k} \\ w_{j+1,l} &= (\bar{g}^{(j)} * c_j)_l = \sum_k g_k c_{j,l+2^j k}, \end{aligned} \tag{9}$$

and the reconstruction is obtained by

$$c_j = \frac{1}{2}(\tilde{h}^{(j)} * c_{j+1} + \tilde{g}^{(j)} w_{j+1}). \tag{10}$$

The à trous algorithm can be extended to 2D, by

$$\begin{aligned} c_{j+1,k,l} &= (\bar{h}^{(j)}\bar{h}^{(j)} * c_j)_{k,l} \\ w_{j+1,1,k,l} &= (\bar{g}^{(j)}\bar{h}^{(j)} * c_j)_{k,l} \\ w_{j+1,2,k,l} &= (\bar{h}^{(j)}\bar{g}^{(j)} * c_j)_{k,l} \\ w_{j+1,3,k,l} &= (\bar{g}^{(j)}\bar{g}^{(j)} * c_j)_{k,l}. \end{aligned} \tag{11}$$

Figure 4 shows the passage from one resolution to the next one by the à trous algorithm. Figure 5 shows the undecimated wavelet transform of the `Einstein` image using five resolution levels. Figures 5 1v,1h,1d correspond respectively the vertical, horizontal and diagonal coefficients of the first resolution level. This transformation contains 16 bands, each one being of the same size as the original image. The redundancy factor is therefore equals to 16.

2.3 Denoising experiments

One of the main applications of the redundant approach is denoising. There are numerous methods for the removal of additive noise form an image, and the

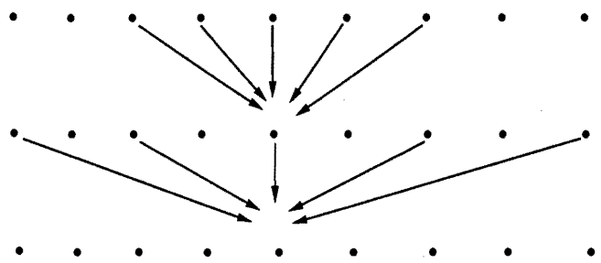


Fig. 4. Passage from c_0 to c_1, and from c_1 to c_2 with the UWT à trous algorithm.

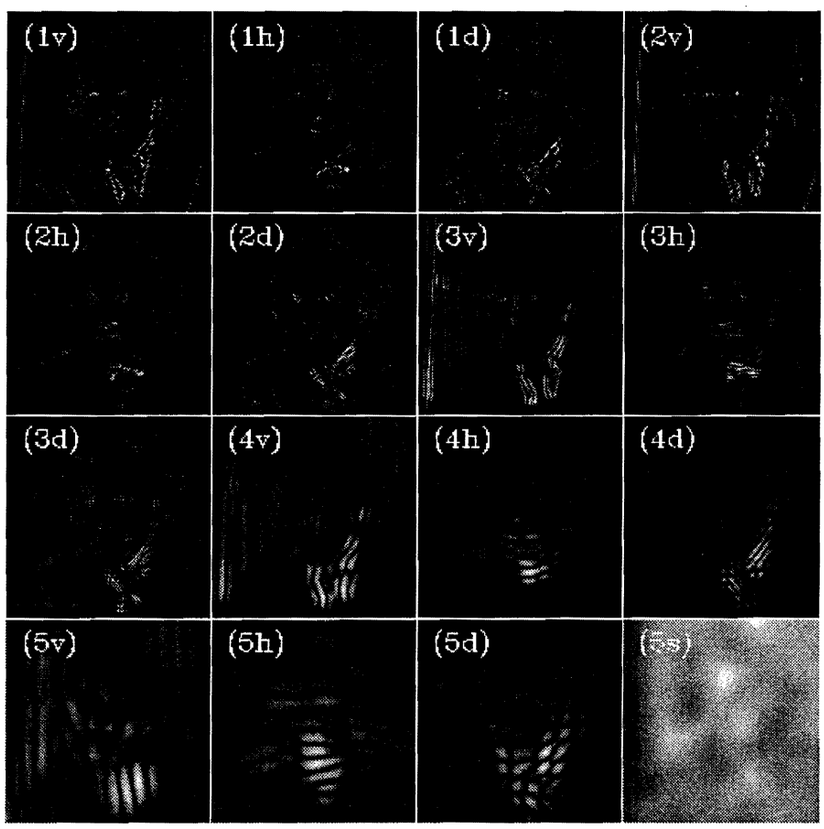


Fig. 5. Undecimated wavelet transform of the `Einstein` image.

wavelet-based method draw special interest because of their theoretical backbone, their success in practice, and their fast implementation. *Hard thresholding* consists of setting to 0 all wavelet coefficients having a near-zero value, this

way removing non-significant wavelet coefficients (Starck and Bijaoui, 1994; Donoho and Johnstone, 1995). At scale j this operation is done by

$$\delta(\tilde{w}_{j,k,l}, T_j) = \begin{cases} w_{j,k,l} & \text{if } \mid w_{j,k,l} \mid \geq T_j \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$w_{j,k,l}$ is the wavelet coefficient at scale j and at spatial position (k, l). In the case of Gaussian noise, T_j can be directly derived from the noise standard deviation, $T_j = K\sigma_j$ (Starck and Bijaoui, 1994; Starck et al., 1998), where σ_j is the noise standard deviation at the scale j, and K is a constant generally chosen between 3 and 5. The $3\sigma_j$ value corresponds to 0.27 % false detection. For a L_2 normalization (i.e. $\sum_l h_l^2 = 1$), we have $\sigma_j = \sigma_I$ for all j, where σ_I is the noise standard deviation in the image, while for a L_1 normalization (i.e. $\sum_l h_l = 1$), we have $\sigma_j = \sigma_I / 2^j$.

Noting $\mathcal{W}_T$ and $\mathcal{W}_R$ the wavelet transform and the reconstruction operators (we have $\mathcal{W}_R = \mathcal{W}_T^{-1}$ for an orthogonal transform), the filtering of an image I is obtained by:

$$\tilde{I} = \mathcal{W}_R \delta(\mathcal{W}_T I, K) \tag{13}$$

where δ corresponds to the non-linear hard thresholding operator. Hence, wavelet filtering based on hard thresholding consists of taking the wavelet transform of the signal, setting to 0 non-significant wavelet coefficients, and applying the inverse wavelet transform. We shall return to this topic in Section 4, when we discuss approximations with sparsity.

To illustrate the denoising idea using wavelet, we have added to the image `Einstein` a white, zero mean Gaussian noise with a standard deviation equals to 20. Figure 6 shows the noisy image (upper left), the filtered image using the bi-orthogonal decimated wavelet transform (upper right) and the filtered image by the bi-orthogonal undecimated wavelet transform (bottom left).In both these examples, K was chosen equal to 4 at the first resolution level and to 3 at other scales. The residual (i.e. difference between the noisy image and the filtered image) related to the undecimated transform is shown in at the bottom right.

As it can easily be seen, the undecimated approach leads to much better denoised result. Other threshold methods have been proposed, like the *universal threshold* (Donoho and Johnstone, 1994; Donoho, 1993), or the SURE (Stein Unbiased Risk Estimate) method (Coifman and Donoho, 1995). Among the best wavelet denoising algorithms, we find Bayesian-based methods exploiting a statistical model of the wavelet coefficients (Crouse et al., 1998; Simoncelli,

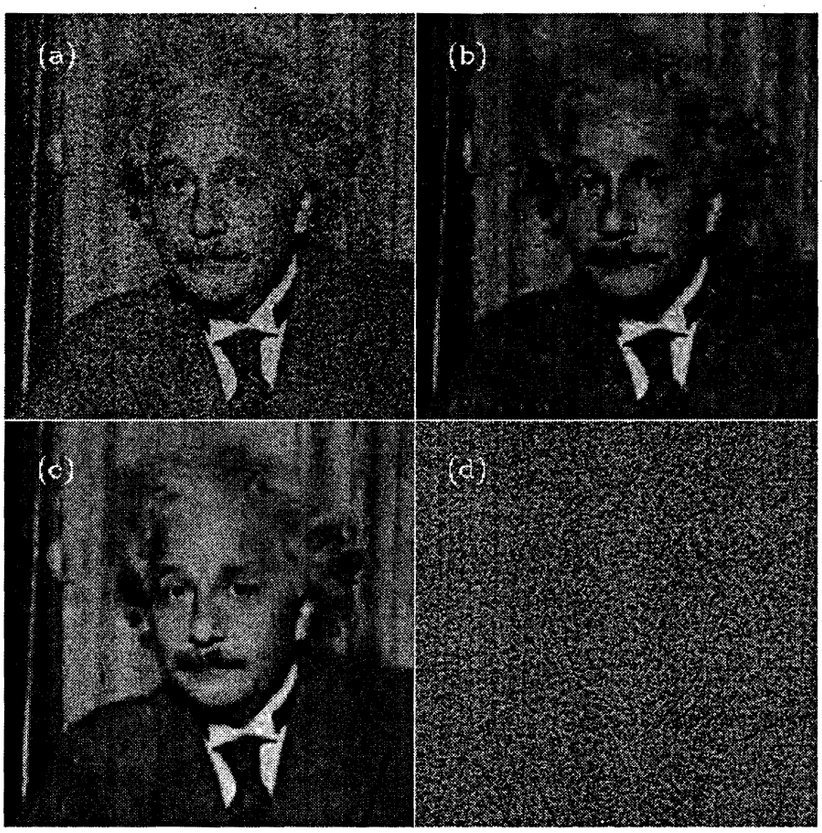


Fig. 6. The noisy `Einstein` image with noise standard deviation 20 (top left), filtered image by the bi-orthogonal wavelet transform (top right), filtered image by the undecimated bi-orthogonal wavelet transform (bottom left), and the residual (i.e. difference between the noisy image and the bottom left image (bottom right).

1999; Portilla et al., 2003). Finally other noise models, such as Poisson noise or non-stationary Gaussian noise can similarly be taken into account in the wavelet denoising approach (Starck et al., 1998).

2.4 Partially Decimated Wavelet Transform (PWT)

The Undecimated wavelet transform (UWT) is highly redundant. The redundancy factor R for images is equal to $3J + 1$ where J is the number of resolution levels. This means that for a $N \times N$ image and using six resolution levels, we need to store $19N^2$ real values in memory. When dealing with very large images, this may not be acceptable in some applications for practical reasons such computation time constraint or available memory space. Then a

compromise can be found by not decimating one or two coarse scales, while decimating the others.

We will note $PWT^{(u)}$ the wavelet transform where the first u are undecimated. For u equals to 0, $PWT^{(u)}$ corresponds to the bi-orthogonal OWT. Similarly, for u equals to J, $PWT^{(J)}$ corresponds to the UWT. As an example, $PWT^{(1)}$ requires a redundancy factor of 4. For the passage from a resolution j to the next one, it will require the same operations as for the UWT when $j \leq u$. Noting $j' = MIN(j, u)$, equation (11) becomes

$$\begin{aligned} c_{j+1,k,l} &= (\bar{h}^{(j')}\bar{h}^{(j')} * c_j)_{k,l} \\ w_{j+1,1,k,l} &= (\bar{g}^{(j')}\bar{h}^{(j')} * c_j)_{k,l} \\ w_{j+1,2,k,l} &= (\bar{h}^{(j')}\bar{g}^{(j')} * c_j)_{k,l} \\ w_{j+1,3,k,l} &= (\bar{g}^{(j')}\bar{g}^{(j')} * c_j)_{k,l} \end{aligned} \tag{14}$$

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

To demonstrate the gain in using PWT over the UWT, we present a denoising experiment where PWT is used with varying u. The same image , `Einstein`, and the same noise characteristics, were used. For each filtered image the PSNR (peak signal-to-noise) ratio between the original image I and the filtered image F was calculated, as presented in table 2.4. The PSNR is defined as

$$PSNR_{dB} = 10 \log_{10} \frac{255}{NRMSE^2} \tag{15}$$

where NRMSE is the normalized root mean square error,

$$NRMSE^2 = \frac{\sum_{pix}(I - F)^2}{\sum_{pix} I^2}. \tag{16}$$

The gain when using the Undecimated WT ($u = 4$) instead of the bi-orthogonal WT is 2.43 dB. Using a single undecimated scale leads to reduce the error by more than 1dB, while requiring far less in redundancy.

	$PWT^{(0)}$	$PWT^{(1)}$	$PWT^{(2)}$	$PWT^{(3)}$	$PWT^{(4)}$
PSNR (dB)	29.34	30.66	31.35	31.67	31.77

Table 1. The PSNR versus u in the PWT for the denoising of the image `Einstein`.

In order to obtain an invariance for translations with only one undecimated scale, an additional refinement can be introduced to the $PWT^{(1)}$ by considering two sets of filter banks $F^o = (h^o, g^o)$ and $F^e = (h^e, g^e)$ instead of one. This new decomposition is called the *Complex Wavelet Transform* (Kingsbury, 1998; Kingsbury, 1999). The wavelet function is not complex but complex numbers are derived from the wavelet coefficients. As described in figure 7, an $N \times N$ image c_0 is first decomposed using (h^o,g^o) into four images (first decomposition level), each one of size $N \times N$ (i.e redundancy factor is equal to 4). Then the smoothed image c_1 is split into four parts:

- Image c_1^A: pixels at even line index and even column index.
- Image c_1^B: pixels at odd line index and even column index.
- Image c_1^C: pixels at even line index and odd column index.
- Image c_1^D: pixels at odd line index and odd column index.

These four images $c_1^A, c_1^B, c_1^C, c_1^D$ are decomposed using the decimated wavelet transform but with different filter banks:

Tree T	A	B	C	D
c_{j+1}^T	$\bar{h}^e\bar{h}^e$	$\bar{h}^e\bar{h}^o$	$\bar{h}^o\bar{h}^e$	$\bar{h}^o\bar{h}^o$
$w_{j+1,1}^T$	$\bar{g}^e\bar{h}^e$	$\bar{g}^e\bar{h}^o$	$\bar{g}^o\bar{h}^e$	$\bar{g}^o\bar{h}^o$
$w_{j+1,2}^T$	$\bar{h}^e\bar{g}^e$	$\bar{h}^e\bar{g}^o$	$\bar{h}^o\bar{g}^e$	$\bar{h}^o\bar{g}^o$
$w_{j+1,3}^T$	$\bar{g}^e\bar{g}^e$	$\bar{g}^e\bar{g}^o$	$\bar{g}^o\bar{g}^e$	$\bar{g}^o\bar{g}^o$

For each sub-band, wavelet coefficients w^A, w^B, w^C, w^D can be interpreted as real and imaginary parts of complex numbers:

$$\begin{aligned} z_{+,j,k} &= (w_{j,k}^A - w_{j,k}^D) + i(w_{j,k}^B + w_{j,k}^C) \\ z_{-,j,k} &= (w_{j,k}^A + w_{j,k}^D) + i(w_{j,k}^B - w_{j,k}^C) \end{aligned} \tag{17}$$

Therefore the three wavelet bands leads to six complex bands corresponding to six directional analysis and it has been shown that the thresholding of $|z_{+,j,k}|$ and $|z_{-,j,k}|$ produces less artifacts than the thresholding of the standard wavelet coefficients (Jalobeanu et al., 2000; Jalobeanu et al., 2003).

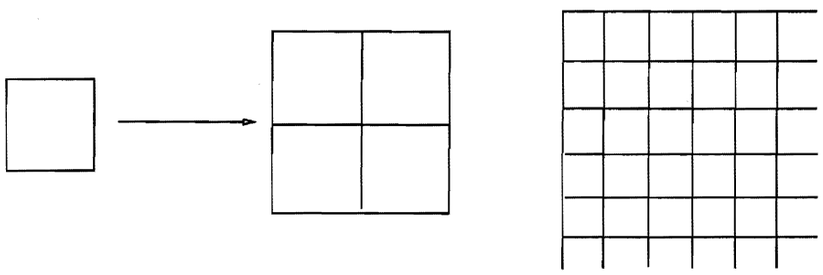


Fig. 7. First level of the 2D complex wavelet transform. Left: undecimated scales, Right: pixels corresponding to the four trees.

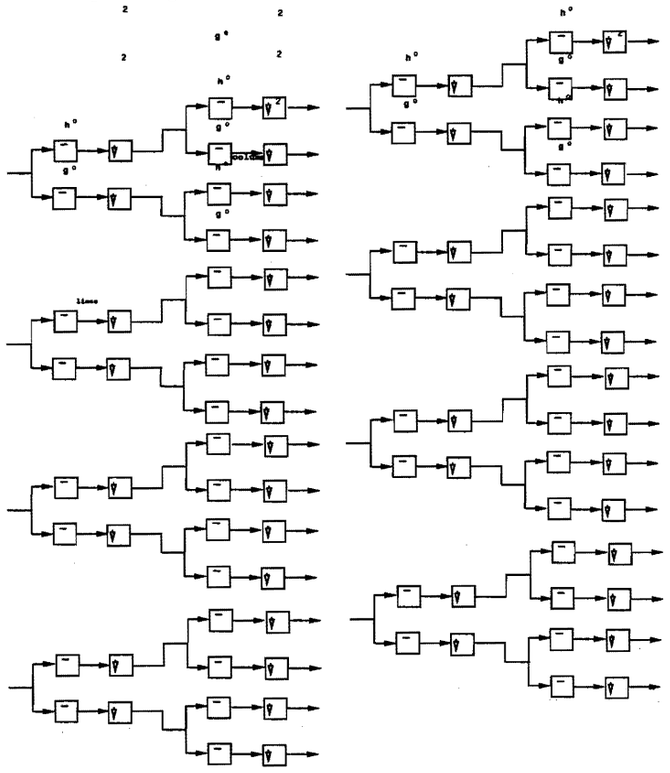


Fig. 8. Second and third levels of the 2D complex wavelet transform.

2.6 The Isotropic à trous Wavelet Transform

This algorithm is well known in the astronomical domain, because it is well adapted to astronomical data where objects are more or less isotropic in most

cases (Starck and Murtagh, 2002). In the undecimated version, we have less constraint on the filters for having a perfect reconstruction. For example, we can define g as $g_0 = 1 - h_0$ and $g_l = -h_l$ if l is not equal to zero. Then the wavelet coefficients are obtained just by taking the difference between two resolutions:

$$w_{j+1,l} = c_{j,l} - c_{j+1,l} \tag{18}$$

where $c_{j+1,l} = (\bar{h}^{(j)} * c_j)_l$. At each scale j, we obtain a set $\{w_j\}$. This has the same number of pixels as the input signal. Here, the wavelet function ψ is defined by:

$$\frac{1}{2}\psi(\frac{x}{2}) = \phi(x) - \frac{1}{2}\phi(\frac{x}{2}) \tag{19}$$

A simple algorithm in order to compute the associated wavelet transform is:

(1) Initialize j to 0 and we start with c_0 being the given image.
(2) Increment j, and apply a discrete convolution of c_j with the filter h. The distance between the central pixel and the adjacent ones is 2^j.
(3) After this smoothing, obtain the discrete wavelet transform as the difference $c_j - c_{j+1}$.
(4) If $j < J$, go to step 2.
(5) The set $\mathcal{W} = \{w_1, ..., w_J, c_J\}$ represents the wavelet transform of the data.

The reconstruction is obtained by a simple co-addition of all wavelet scales and the final smoothed array, namely

$$c_{0,l} = c_{J,l} + \sum_{j=1}^{J} w_{J,l}. \tag{20}$$

For the scaling function, $\phi(x)$, the B-spline of degree 3 is generally considered as a good choice. The associated h filter is $\frac{1}{16}(1, 4, 6, 4, 1)$, being symmetric.

The above *à trous* algorithm is easily extendable to two-dimensional space:

$$\begin{aligned} c_{j+1,k,l} &= (\bar{h}^{(j)}\bar{h}^{(j)} * c_j)_{k,l} \\ w_{j+1,k,l} &= c_{j,k,l} - c_{j+1,k,l} \end{aligned} \tag{21}$$

and the reconstruction is still a simple co-addition of the wavelet scales and the smooth arrays.

The use of the B_3 spline leads to a convolution with the mask hh of 5×5:

$$\frac{1}{256}\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

but it is faster to compute the convolution in a separable way (first on rows, and then on the resulting columns).

Figure 9 shows the undecimated isotropic wavelet transform of the image `Einstein` using six resolution levels. This transformation contains 6 bands, each one being of the same size as the original image. The redundancy factor is therefore equals to 6. The simple addition of these six images reproduce exactly the original image. This transformation is very well adapted to the analysis of astronomical images, assumed to contain generally relatively isotropic features. This construction has close relation to the Laplacian pyramidal construction by Burt and Adelson (Burt and Adelson, 1983) or the FFT based pyramidal wavelet transform (Starck et al., 1998).

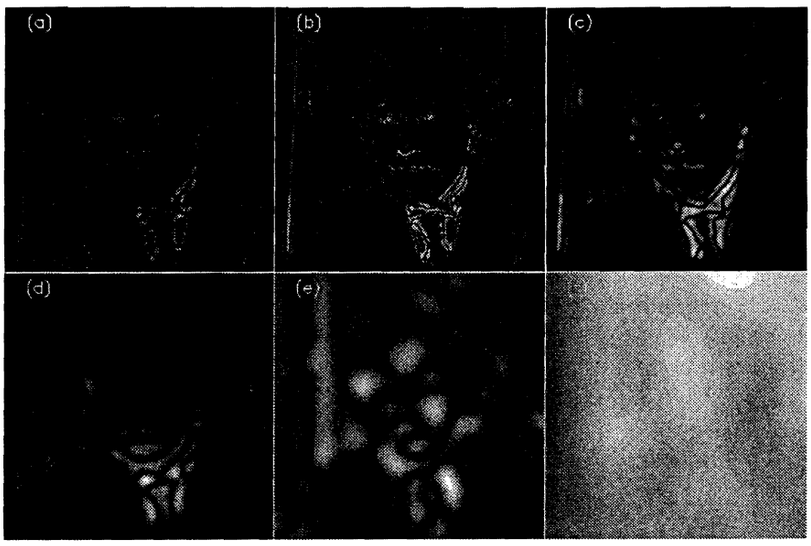


Fig. 9. Undecimated isotropic wavelet transform of the `Einstein` image.

2.7 Contrast enhancement

Since some features in an image may be hard to detect by the human eyes due to low contrast, we often process the image before visualization. Histogram equalization is certainly one the most well known methods for contrast enhancement. 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

$$I_{k,l} = \log(c_{J,k,l}) + \sum_{j=1}^{J} \mathrm{sgn}(w_{j,k,l}) \log(\mid w_{j,k,l} \mid) \tag{22}$$

Figure 10 shows a Hale-Bopp Comet image (top left) and an opthalmic medical image (top right), their histogram equalization (middle row), and their wavelet-log representation (bottom). Jets clearly appears in the last representation of Hale-Bopp Comet image, and many more features are distinguishable in the wavelet log representation of the opthalmic medical image.

2.8 Other Redundant Wavelet Constructions

Other redundant wavelet transforms that are of interest are the steerable wavelet and the dyadic wavelet transforms. The steerable wavelet transform (Simoncelli et al., 1992a) allows us to choose the number of directions in the multiscale decomposition, and the redundancy is proportional to this number. The dyadic wavelet transform (Mallat and Hwang, 1992; Mallat and Zhong, 1992) produces two undecimated bands per scale (horizontal and vertical) with a redundancy factor $R = 2J + 1$ where J is the number of scales. This decomposition can be seen as a generalizing of the concept of multiscale edge detection. Indeed, by using a differentiable smoothing function, we have

$$\psi^1(x,y) = \frac{d\phi(x,y)}{dx} \quad \text{and} \quad \psi^2(x,y) = \frac{d\phi(x,y)}{dy} \tag{23}$$

By definition, ψ^1 and ψ^2 are wavelets (their integral is equal to zero). The local extremum of the wavelet coefficients using ψ^1, ψ^2 correspond to the inflection

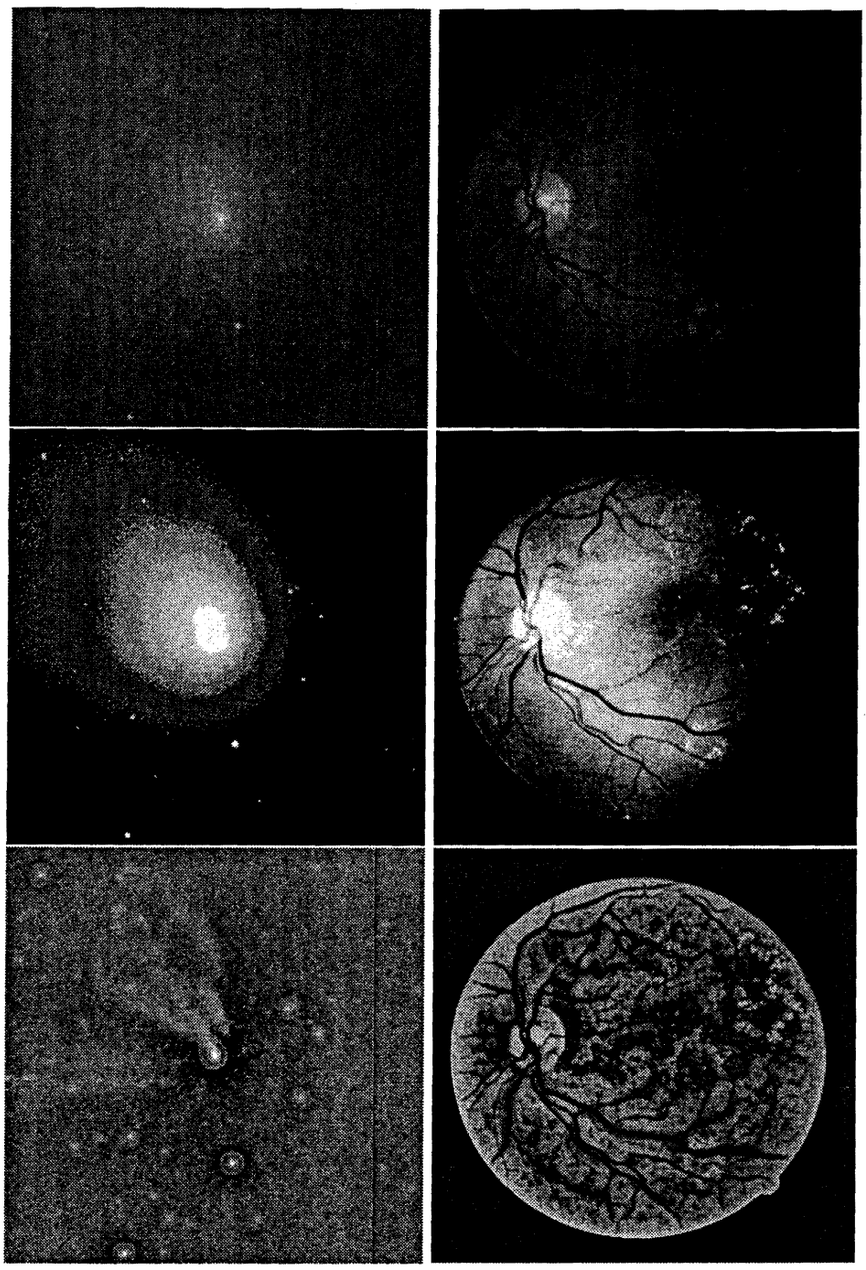


Fig. 10. Top left - Hale-Bopp Comet image, To right - opthalmic medical image. Middle - histogram equalization results, Bottom - wavelet-log representations.

points (edges) of $f * \phi_s$ (with $\phi_s = \frac{1}{s}\phi(\frac{x}{s})$). Then we have at each scale j and at pixel location (k, l) two wavelet coefficients $w_{j,1,k,l}, w_{j,2,k,l}$. The modulus of

the gradient is then defined by $G_{j,k,l} = \sqrt{w_{j,1,k,l}^2 + w_{j,2,k,l}^2}$ and the directional angle θ_j is

$$\theta_{j,k,l} = \begin{cases} \arctan(\frac{w_{j,2,k,l}}{w_{j,1,k,l}}) & \text{if } w_{j,1,k,l} \geq 0 \\ \pi - \arctan(\frac{w_{j,2,k,l}}{w_{j,1,k,l}}) & \text{if } w_{j,1,k,l} < 0 \end{cases} \tag{24}$$

Multiscale edge points, also called modulus maxima, are points where the modulus is locally maximum with respect to its neighbors along the direction θ_j. An interesting feature is that an image can be reconstructed (approximately) from its multiscale edges (Mallat, 1998) using an iterative algorithm. It does not converge exactly towards the original image, but in practice the error is very small.

Wavelet packets are an extension of the wavelet transform. They were introduced by Coifman, Meyer and Wickerhauser (Coifman et al., 1992). Instead of dividing only the approximation space, as in the standard (bi-) orthogonal wavelet transform, detail spaces are also divided. For some application such deconvolution, it has been shown that some wavelet packets bases called *mirror bases* allows us to better take into account the noise behavior and therefore to outperform the standard wavelet transform (Kalifa et al., 2003; Jalobeanu et al., 2003). Introducing a redundancy in the wavelet packets decomposition improves also significantly the restoration.

Finally, we should note that we have considered in this section only linear redundant transforms. Non-linear redundant multiscale transform have also been proposed, such those based on the median transform (Starck et al., 1996; Donoho, 2000).

2.9 Local Overlapping DCT

The DCT is not a multiscale transform, but its relevance to the topic of image separation discussed later in this paper justifies its brief description. The DCT is a variant of the Discrete Fourier Transform, replacing the complex analysis with real numbers by a symmetric signal extension. The DCT is an orthonormal transform, known to be well suited for stationary signals obeying a first order Markov models with high correlation. This transform is defined by

$$DCT(u,v) = \frac{1}{\sqrt{2N}} c(u)c(v) \cdot \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} I_{k,l} \cos(\frac{(2k+1)u\pi}{2N}) \cos(\frac{(2l+1)v\pi}{2N}) \quad (25)$$

where $I_{k,l}$ is the input image. Its coefficients essentially represents frequency content, similar to the ones obtained by Fourier analysis. When dealing with non-stationary sources, DCT is typically applied in blocks. Such is indeed the case in the JPEG image compression algorithm. Choice of overlapping blocks is preferred for analyzing signals while preventing blotckiness effects. In such a case we get again an overcomplete transform with redundancy factor of 4 for an overlap of 0.5. A multiscale version of the block-DCT could be proposed, where the image is divided into blocks of varying sizes. A fast algorithm with complexity of $n^2 \log_2 n$ exists for its computation. The DCT is appropriate for a sparse representation of smooth or periodic behaviors.

3 background - Part II - From Wavelets to Curvelets

3.1 Problems with Wavelets

Despite the success of the classical wavelet viewpoint, recent papers (Candès and Donoho, 1999d; Candès and Donoho, 1999c) argued that the traditional wavelets present some strong limitations that question their effectiveness in higher-dimension than 1. Wavelets rely on a dictionary of roughly isotropic elements occurring at all scales and locations, do not describe well highly anisotropic elements, and contain only a fixed number of directional elements, independent of scale. Therefore, classical multiresolution ideas only address a portion of the whole range of interesting multiscale phenomena. Following this reasoning, new constructions have been proposed such as the ridgelets (Candès, 1999; Vetterli, 2001), the curvelets (Candès and Donoho, 1999c; Starck et al., 2002), the bandlets (Pennec and Mallat, 2000), and the contourlets (Do and Vetterli, 2003b). This section presents some of these new redundant constructions and explains how they better suit the 2D signals the come to describe.

3.2 The Continuous Ridgelet Transform

The two-dimensional continuous ridgelet transform in $\mathbf{R}^2$ can be defined as follows (Candès, 1999). We pick a smooth univariate function $\psi : \mathbf{R} \rightarrow \mathbf{R}$ with

sufficient decay and satisfying the admissibility condition

$$\int |\hat{\psi}(\xi)|^2/|\xi|^2 \, d\xi < \infty, \tag{26}$$

which holds if, say, ψ has a vanishing mean $\int \psi(t)dt = 0$. We will suppose a special normalization about ψ so that $\int_0^\infty |\hat{\psi}(\xi)|^2 \xi^{-2} d\xi = 1$.

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

$$\psi_{a,b,\theta}(x_1, x_2) = a^{-1/2} \cdot \psi((x_1 \cos\theta + x_2 \sin\theta - b)/a); \tag{27}$$

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

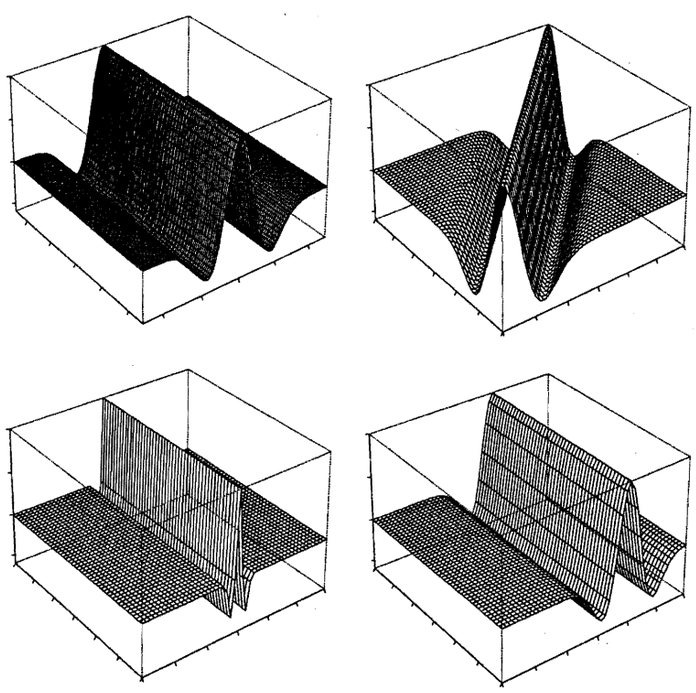


Fig. 11. Few Ridgelets examples - The top right, bottom left and right graphs are obtained after simple geometric manipulations of the upper left ridgelet, namely rotation, rescaling, and shifting.

Figure 11 presents few ridgelets examples. 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

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

We have the exact reconstruction formula

$$f(x) = \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{\infty} \mathcal{R}_f(a,b,\theta) \psi_{a,b,\theta}(x) \frac{da}{a^3} db \frac{d\theta}{4\pi} \tag{28}$$

valid a.e. for functions which are both integrable and square integrable.

Ridgelet analysis may be constructed as wavelet analysis in the Radon domain. 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

$$Rf(\theta,t) = \int f(x_1,x_2)\delta(x_1 \cos\theta + x_2 \sin\theta - t)\, dx_1 dx_2, \tag{29}$$

where δ is the Dirac distribution. Then 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. Thus, the 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 transforms have been proposed, and we will describe three of them in this section, based on different implementations of the Radon transform.

3.2.1 The RectoPolar Ridgelet transform

A fast implementation of the RT can be proposed in the Fourier domain, based on the projection-slice-theorem. First the 2D FFT is computed to the given image. Then the resulting function in the frequency domain is to be used to evaluate the frequency values in a polar grid of rays passing through the origin and spread uniformly in angle. This conversion from cartesian to Polar grid could be obtained by interpolation, and this process is well known by the name gridding in tomography. Given the polar grid samples, the number of rays corresponds to the number of projections, and the number of samples on each ray corresponds to the number of shifts per such angle. Applying one dimensional inverse Fourier transform for each ray, the Radon projections are obtained.

The above described process is known to be inaccurate due to the sensitivity to the interpolation involved. This implies that for a better accuracy, the first 2D-FFT employed should be done with high-redundancy.

An alternative solution for the Fourier-based Radon transform exists, where the polar grid is replaced with a pseudo-polar one. The geometry of this new grid is illustrated in Figure 3.2.1. Concentric circles of linearly growing radius

in the polar grid are replaced by concentric squares of linearly growing sides. The rays are spread uniformly not in angle but in slope. These two changes give a grid vaguely resembling the polar one, but for this grid a direct FFT can be implemented with no interpolation. When applying now 1D-FFT for the rays, we get a variant of the radon transform, where the projection angles are not spaced uniformly.

For the pseudo-polar FFT to be stable, it was shown that it should contain at least twice as many samples, compared to the original image we started with. A by-product of this construction is the fact that the transform is organized as a 2D array with rows containing the projections as a function of the angle. Thus, processing the Radon transform in one axis is easily implemented. More details can be found in (Starck et al., 2002).

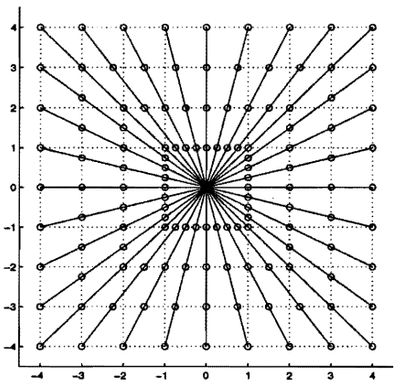


Fig. 12. Illustration of the psudo-polar grid in the frequency domain for an n by n image ($n = 8$).

3.2.2 1D Wavelet Transform

To complete the ridgelet transform, we must take a one-dimensional wavelet transform along the radial variable in Radon space. We now discuss the choice of digital one-dimensional wavelet transform.

Experience has shown that compactly-supported wavelets can lead to many visual artifacts when used in conjunction with nonlinear processing, such as hard-thresholding of individual wavelet coefficients, particularly for decimated wavelet schemes used at critical sampling. Also, because of the lack of localization of such compactly-supported wavelets in the frequency domain, fluctuations in coarse-scale wavelet coefficients can introduce fine-scale fluctuations. A frequency-domain approach must be taken, where the discrete Fourier transform is reconstructed from the inverse Radon transform. These considerations lead to use band-limited wavelet, whose support is compact in the Fourier

domain rather than the time-domain (Donoho, 1998; Donoho, 1997; Starck et al., 2002). In (Starck et al., 2002), a specific overcomplete wavelet transform (Starck et al., 1994; Starck et al., 1998) has been used. The wavelet transform algorithm is based on a scaling function ϕ such that $\hat{\phi}$ vanishes outside of the interval $[-\nu_c, \nu_c]$. We define the scaling function $\hat{\phi}$ as a re-normalized B_3-spline

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

and $\hat{\psi}$ as the difference between two consecutive resolutions

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

Because $\hat{\psi}$ is compactly supported, the sampling theorem shows than one can easily build a pyramid of $n + n/2 + \ldots + 1 = 2n$ elements, see (Starck et al., 1998) for details.

This transform enjoys the following features:

- The wavelet coefficients are directly calculated in the Fourier space. In the context of the ridgelet transform, this allows avoiding the computation of the one-dimensional inverse Fourier transform along each radial line.
- Each sub-band is sampled above the Nyquist rate, hence, avoiding aliasing –a phenomenon typically encountered by critically sampled orthogonal wavelet transforms (Simoncelli et al., 1992b).
- The reconstruction is trivial. The wavelet coefficients simply need to be co-added to reconstruct the input signal at any given point. In our application, this implies that the ridgelet coefficients simply need to be co-added to reconstruct Fourier coefficients.

This wavelet transform introduces an extra redundancy factor. However, we note that the goal in this implementation is not data compression or efficient coding. Rather, we focus on data analysis, for which it is well-known that over-completeness can provide substantial advantages as we have already seen before (Coifman and Donoho, 1995).

Figure 13 shows the flow-graph of the ridgelet transform. The ridgelet transform of an image of size $n \times n$ is an image of size $2n \times 2n$, introducing a redundancy factor equal to 4.

We note that, because this transform is made of a chain of steps, each one of which is invertible, the whole transform is invertible, and so has the exact reconstruction property. For the same reason, the reconstruction is stable under perturbations of the coefficients.

Last but not least, this discrete transform is computationally attractive. Indeed, the algorithm we presented here has low complexity since it runs in

$O(n^2 \log(n))$ flops for an $n \times n$ image.

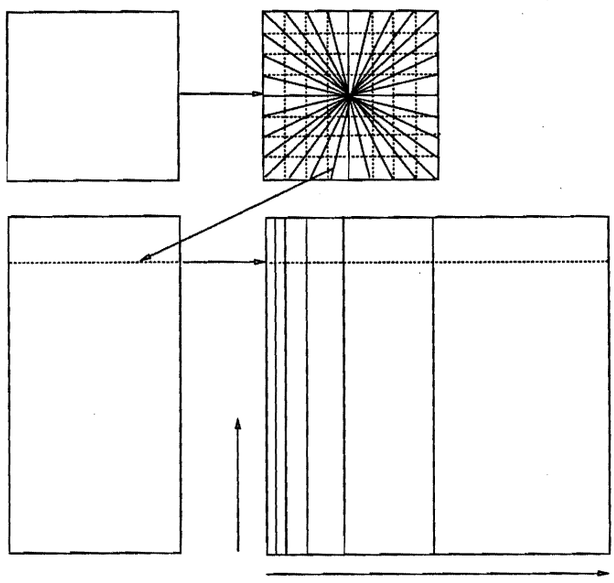


Fig. 13. Ridgelet transform flow graph. Each of the $2n$ radial lines in the Fourier domain is processed separately. The 1-D inverse FFT is calculated along each radial line followed by a 1-D nonorthogonal wavelet transform. In practice, the one-dimensional wavelet coefficients are directly calculated in the Fourier space.

The ridgelet transform of a digital array of size $n \times n$ is an array of size $2n \times 2n$ and hence introduces a redundancy factor equal to 4.

3.2.3 Example: anisotropic feature detection

Consider an image containing a vertical band embedded in white noise with relatively large amplitude. Figure 14 (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 those wavelets clearly do not correlate very well with the very elongated structure (pattern) of the object to be detected.

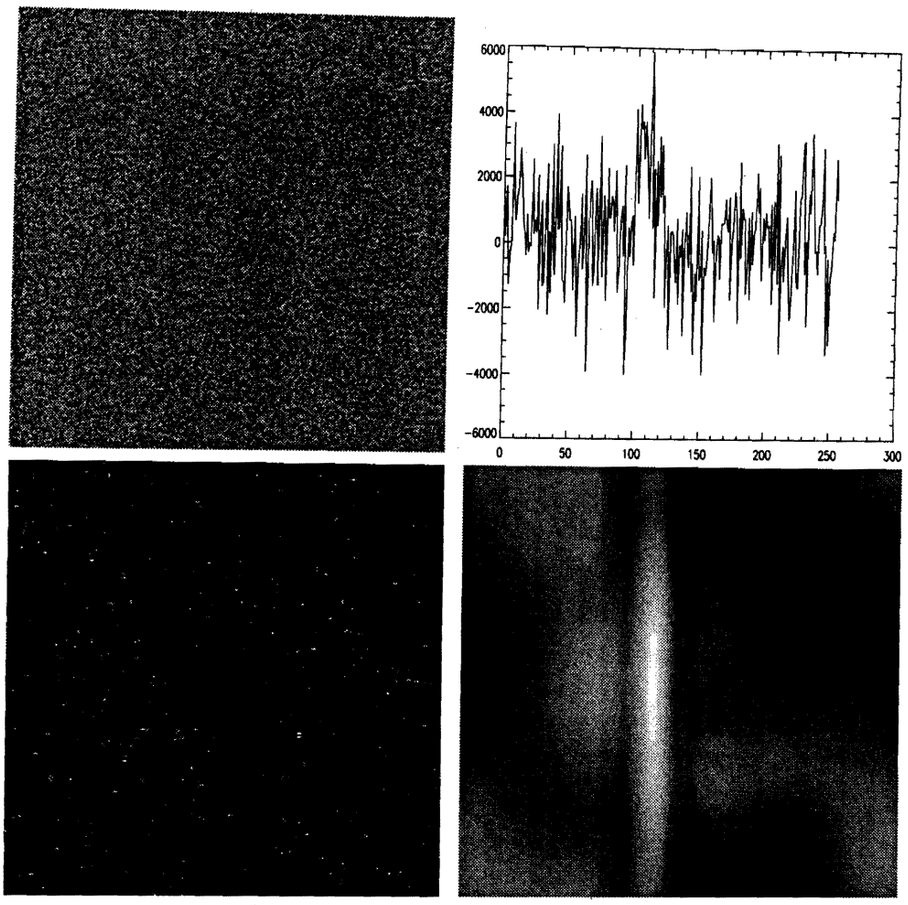


Fig. 14. Original image containing a vertical band embedded in white noise with relatively large amplitude (top left). The signal obtained by integrating the image intensity over columns (top right). Reconstructed image for the undecimated wavelet coefficient (bottom left). Reconstructed image from the ridgelet coefficients (bottom right).

3.3 The Orthonormal Finite Ridgelet Transform

The orthonormal finite ridgelet transform (OFRT) has been recently proposed (Do and Vetterli, 2003c) for image compression and filtering. This transform is based on the finite Radon transform (Matus and Flusser, 1993) and a 1D orthogonal wavelet transform. It is not redundant and reversible. It would have been a great alternative to the previously described ridgelet transform if the OFRT were not based on a strange definition of a line. In fact, a line in the OFRT is defined as a set of periodic equidistant points (Matus and Flusser, 1993). Figure 15 shows the back-projection of a ridgelet coefficient by

the FFT-based ridgelet transform (left) and by the OFRT (right). It is clear that the backprojection of the OFRT is nothing like a ridge function.

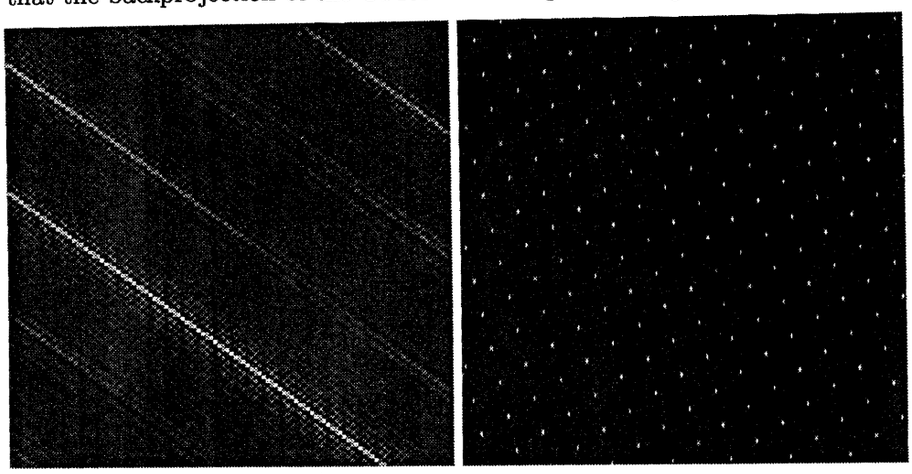


Fig. 15. The backprojection of a ridgelet coefficient by the FFT-based ridgelet transform (left), and by the OFRT (right).

Because of this specific definition of a line, the thresholding of the OFRT coefficients produces strong artifacts. Figure 16 shows a part of the original image Lena, and its reconstruction after the hard thresholding of the OFRT. A noise has been added to the noise-free image as part of the filtering!

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 partitioning when denoising the data, and a prime number block size can be used. The OFTR is interesting from the conceptual point of view, but still requires work before it can be used for real applications such as denoising.

3.4 The Slant Stack Ridgelet Transform

The Fast Slant Stack (Averbuch et al., 2001) is geometrically more accurate than the previously described methods. The back-projection of a point in Radon space is exactly a ridge function in the spatial domain (see Figure 17). The transformation of an $n \times n$ 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 $[\frac{\pi}{4}, \frac{3\pi}{4}]$ are 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 pixel values along all horizontal lines (resp. vertical lines for angles in $[\frac{\pi}{4}, \frac{3\pi}{4}]$). The shearing is performed one column at a time (resp. one line at a time) by using the 1D FFT. Figure 18 shows an example of the image

Fig. 16. Part of original noise free **Lena** image (left), and reconstruction after OFRT-based denoising (right).

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 (Donoho and Flesia, 2002). The connection between the Fast Slant Stack and the linogram has been investigated in (Averbuch et al., 2001), and a Fast Slant Stack is proposed, based on the 2D Fourier transform.

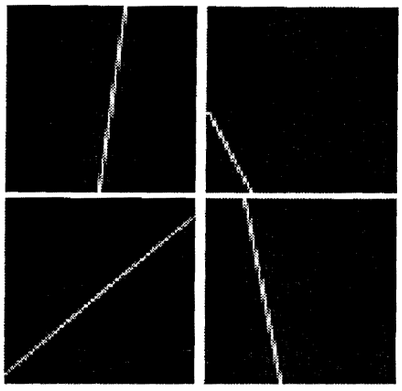


Fig. 17. Backprojection of a point at four different locations in the Radon space.

3.5 Local Ridgelet Transforms

The ridgelet transform is optimal for finding global lines of the size of the image. To detect line segments, a partitioning must be introduced (Candès, 1998). The image can be decomposed into overlapping blocks of side-length b pixels in such a way that the overlap between two vertically adjacent blocks is

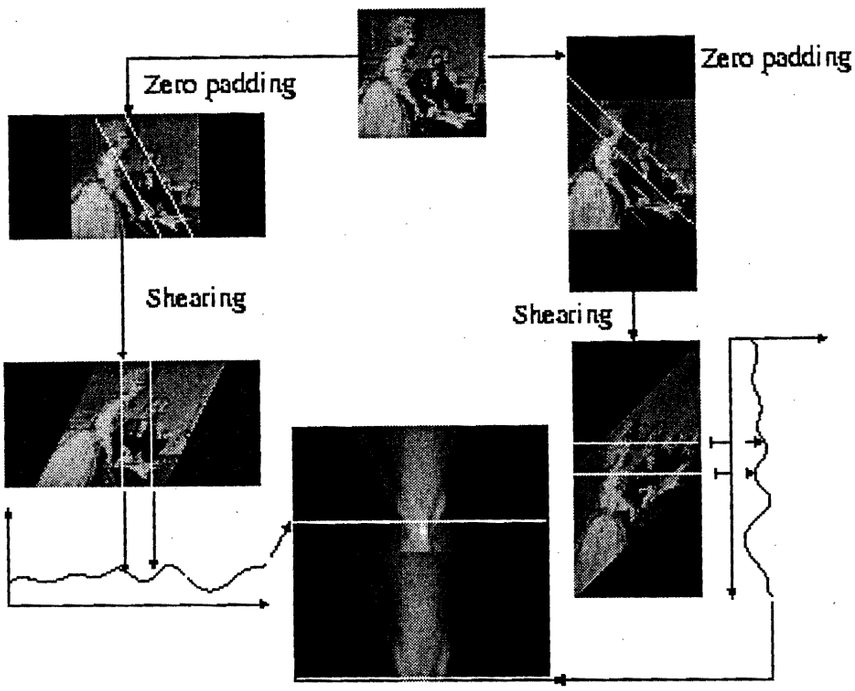


Fig. 18. Slant Stack Transform of an image.

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, and thus the redundancy factor grows by a factor of 4.

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 coaddition 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).

Experiments have shown that the second approach leads to better results. 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:

$$
\begin{aligned}
f_1 &= w(i_2/\ell)B_1(i_1, j_1) + w(1 - i_2/\ell)B_2(i_2, j_1) \\
f_2 &= w(i_2/\ell)B_3(i_1, j_2) + w(1 - i_2/\ell)B_4(i_2, j_2) \\
f(i, j) &= w(j_2/\ell)f_1 + w(1 - j_2/\ell)f_2. \qquad (30)
\end{aligned}
$$

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

3.6 The Curvelet Transform

The curvelet transform (Candès and Donoho, 1999a; Donoho and Duncan, 2000; Starck et al., 2002) opens the possibility to analyze 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 local ridgelet transform. The block size can be changed at each scale level. Roughly speaking, different levels of the multiscale ridgelet pyramid are used to represent different sub-bands of a filter bank output. At the same time, this sub-band 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 with the property that the pass-band filter Δ_s is concentrated near the frequencies $[2^{2s}, 2^{2s+2}]$, e.g.

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

In wavelet theory, one uses a decomposition into dyadic sub-bands $[2^s, 2^{s+1}]$. In contrast, the sub-bands 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.

The curvelet decomposition is the sequence of the following steps:

- *Sub-band Decomposition.* The object f is decomposed into sub-bands.
- *Smooth Partitioning.* Each sub-band is smoothly windowed into "squares" of an appropriate scale (of side-length $\sim 2^{-s}$).
- *Ridgelet Analysis.* Each square is analyzed via the discrete ridgelet transform.

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

3.6.1 *Digital Realization*

It seems that the isotropic "à trous" wavelet transform 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

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

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 section 1). Thus, the algorithm outputs $J+1$ sub-band arrays of size $n \times n$.

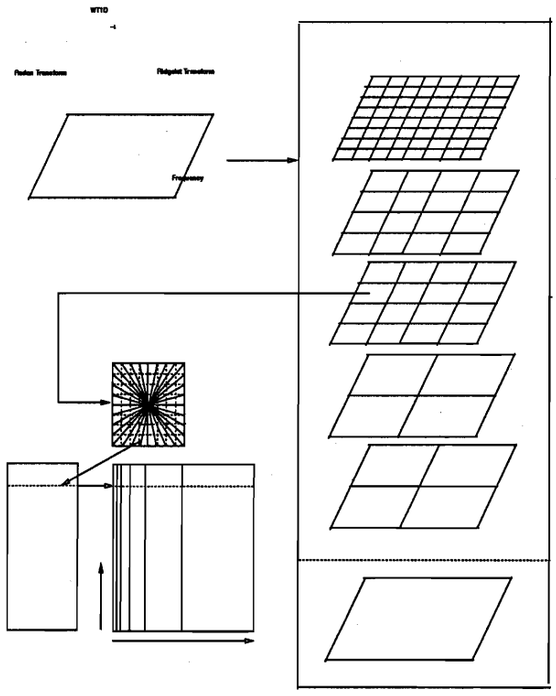


Fig. 19. Curvelet transform flow graph. The figure illustrates the decomposition of the original image into sub-bands followed by the spatial partitioning of each sub-band. The ridgelet transform is then applied to each block.

A sketch of the discrete curvelet transform algorithm is:

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

- if j modulo $2 = 1$ then $B_{j+1} = 2B_j$,
- else $B_{j+1} = B_j$.

The side-length of the localizing windows is doubled *at every other* dyadic sub-band, 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 sub-band $[2^j, 2^{j+1}]$. Note also that the coarse description of the image c_J is not processed. We used the default value $B_{min} = 16$ pixels in our implementation. Finally, Figure 19 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 these invertibility holds for each element of the processing chain. Figure 20 shows a few curvelets

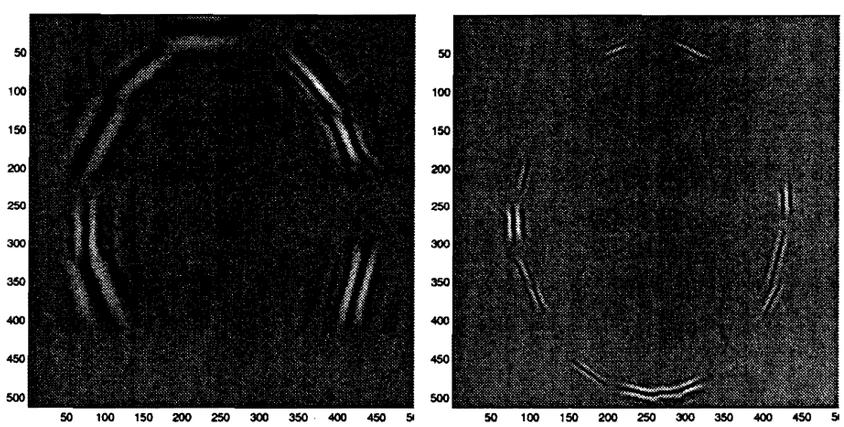


Fig. 20. A few curvelets.

at different scales, orientations and locations.

The curvelet transform is a promising approach and is still under development (Candès and Donoho, 2002; Do and Vetterli, 2003a). Future curvelet decompositions will certainly allows us to obtain similar quality for denoising and detection applications, but with much less redundancy.

3.6.2 Example: Recovery of Curves

In this experiment (Figure 21), we have added a Gaussian noise to "War and Peace," a drawing from Picasso which contains many curved features. Figure 21 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.

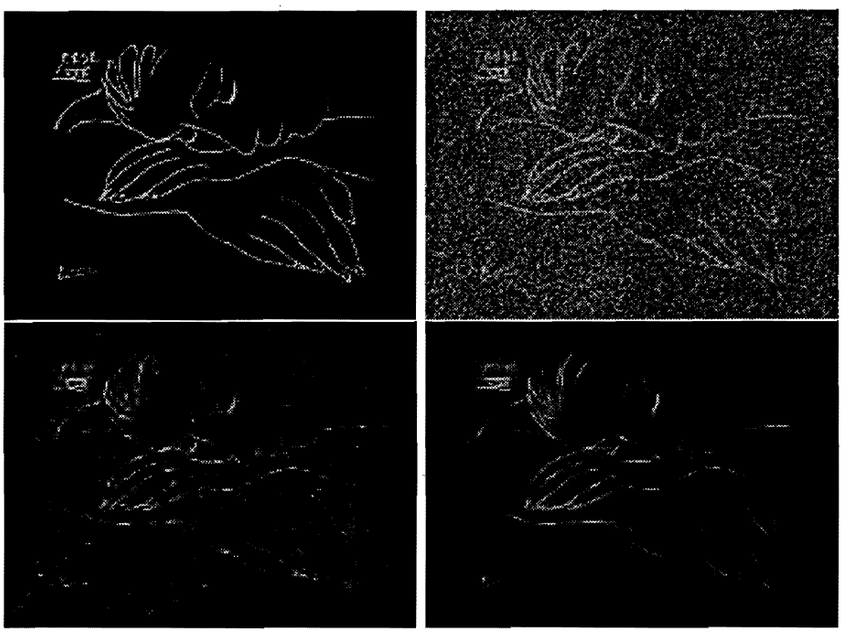


Fig. 21. The Picasso picture `War and Peace` (top left), the same image contaminated with a Gaussian white noise (top right). The restored images using the undecimated wavelet transform (bottom left) and the curvelet transform (bottom right).

4 Background - Part III - Sparsity in Transforms

4.1 Linear transforms and their limitations

So far we have been focusing on transforms without specifying whether they are applied linearly or non-linearly, and as we shall see, both options are open before us. In this section we present these alternatives and show how sparsity fits into this dichotomy.

Since the signals we work with here are all of finite dimensions, linearity is characterized by the ability to represent both the forward and the inverse transforms by matrices multiplying vectors. We denote our signal as $\underline{s} \in \mathbf{R}^N$, and assume that the inverse transform is obtained by the multiplication $\underline{s} = \mathbf{T}\underline{u}$. The transform matrix $\mathbf{T}$ has N rows and L columns, and clearly $\mathbf{T}$ must be full rank and with $L \geq N$ if we desire to span the entire $\mathbf{R^N}$ space. The idea behind the relation $\underline{s} = \mathbf{T}\underline{u}$ is to consider the signal $\underline{s}$ as a linear combination

of columns from $\mathbf{T}$. Thus, we commonly refer to this matrix as the dictionary and its columns as atoms that construct the signal.

If $\mathbf{T}$ is a square (and non-singular) matrix, the forward transform is uniquely defined as the matrix inverse, i.e. $\mathbf{T}^{-1}\underline{s} = \underline{u}$. Such is the case with the DFT and the critically sampled Wavelet transform. If, on the other hand, $L > N$, the transform is redundant, and we have some freedom in defining its forward operation. We can propose a specific forward transform depicted by the following constrained optimization task

$$\min_{\underline{u}} \|\mathbf{D}\underline{u}\|_2^2 \text{ subject to } \underline{s} = \mathbf{T}\underline{u}, \tag{31}$$

where $\mathbf{D}$ is a full rank matrix with L columns and P rows, and we must require that $P \geq L$ for obtaining a unique solution from (31) (this property is immediately seen from the next analysis). Due to the ℓ^2-norm involved in the above expression, the forward transform is also linear and given by

$$\underline{u} = \left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{T}^T\left[\mathbf{T}\left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{T}^T\right]^{-1}\underline{s}. \tag{32}$$

The choice of $\mathbf{D}$ dictates the behavior of the representation coefficients $\underline{u}$. As a popular example, choosing $\mathbf{D} = \mathbf{I}$ implies a search for the minimal ℓ^2 energy signal, and the forward transform in this case becomes

$$\mathbf{D} = \mathbf{I} \quad \Rightarrow \quad \underline{u} = \mathbf{T}^T\left(\mathbf{T}\mathbf{T}^T\right)^{-1}\underline{s} = \mathbf{T}^+\underline{s}, \tag{33}$$

resulting with the well known Moore-Penrose pseudo-inverse (Golub and Van-Loan, 1996).

Note that a dual approach can be used where the forward transform is defined as a redundant matrix multiplying a vector, and from there develop the inverse transform. Our choice to start with the inverse as a linear operation and expand from there to the forward is aligned with the way the sparsity–based transforms are developed, as we shall show in the next Section.

Linearity is a tempting property since it leads to a traceable analysis and to a closed-form transform description as we have seen above. Moreover, numerical solution of (31) can be employed as an iterative process with the use of only multiplications by $\mathbf{T}$ and its adjoint, an appealing option that avoids inversions of matrices which could be daunting if high-dimensional signals are involved.

All the tools described in Sections 2 and 3 are typically employed as linear transforms, although they can be used differently. Ridgelets, Curvelets, and other mentioned algorithms are proposing over-complete representations, but

linearity may impose undesired limitations. In fact we can regard all these methods as proposals for the content of the matrix $\mathbf{T}$. The question is now - can we use this content in a better setting? This leads us naturally to the non-linear transforms.

As an interesting side-comment we mention that in the setting described in (31), we have the freedom to choose $\mathbf{T}$ as one of the matrices built by the transforms described in previous sections. However, we may also consider to construct $\mathbf{T}$ as an amalgam of several matrices concatenated horizontally, forming this way a longer representation with a richer set of building atoms. As we shall observe in later sections, this is crucial for the signal separation we will advocate.

4.2 Non-Linear approach - Sparsity and the Pursuit Algorithms

In (31) the term $\|\mathbf{D}\underline{u}\|_2^2$ measures the complexity of the obtained representation vector, and in seeking the minimum, we effectively search for the most appealing representation. However, the use of ℓ^2-norm for measuring the complexity of the representation are far from satisfactory. If we desire sparseness in $\underline{u}$ as a true measure of simplicity, the above term should be replaced by $\|\underline{u}\|_0$, essentially counting number of non-zero entries in $\underline{u}$. This is a commonly used abuse of notation since $\|\underline{u}\|_p^p = \sum_{j=1}^{L} |u(j)|^p$, and for $p \to 0$ we get that any non-zero entry to the power of p becomes 1, while every zero entry is nulled in this summation.

With the above proposed change we lose the linearity of the forward transform and the ease of analysis and operations that accompany it. For years these shortcomings were considered as a grave loss to be seriously considered, and sparsity was therefore left aside. In recent years, with improved computing power and with a strong thirst for performance–barrier breaking methods, sparsity became a relevant issue. Surprisingly, several supporting evidence emerged from different directions to support this choice of simplicity measure and make it a serious candidate for the alternative design of the forward transform.

We still think in terms of a matrix $\mathbf{T}$ multiplying the representation vector $\underline{u}$ in order to construct the signal $\underline{s}$, and thus the inverse transform remains linear. Our objective in defining the forward transform is now

$$(P_0) \qquad \min_{\underline{u}} \|\underline{u}\|_0 \text{ subject to } \underline{s} = \mathbf{T}\underline{u}. \tag{34}$$

One major problem that stands as an obstacle in addressing (34) is the fact

that this problem is non-convex and highly non smooth, implying difficulties in its numerical solution. An exhaustive approach for the solution of this problem can be suggested, were we start with the assumption $\|\underline{u}\|_0 = 1$ and test every column as a candidate representation. If successful we are done, and if not we assume $\|\underline{u}\|_0 = 2$ and test for all the pairs of columns. This sweep of tests should theoretically proceed till the solution is found or till $\|\underline{u}\|_0 = N$ where a solution must be found. Thus, the overall number of Least-Squares tests to be done is growing exponentially with L, the number of columns in $\mathbf{T}$.

Approximations with empirical success were proposed to overcome this problem, and well known methods among those are the pursuit algorithms (Matching Pursuit, Basis Pursuit, and their variants). It is perhaps most surprising that in-spite of their heuristic origin, exact theoretical claims can be made, and indeed have been done recently, supporting the successful behavior of these algorithms. A brief survey of these results as presented in (Donoho and Huo, 2001; Elad and Bruckstein, 2001; Elad and Bruckstein, 2002; Donoho and Elad, 2003; Gribonval, 2003; Tropp, 2003; Gilbert and Strauss, 2003) is given below.

4.3 Theoretical and Empirical Performance of Pursuit algorithms

Common to the analysis of both the MP and the BP algorithms is a feature M describing the richness of the dictionary $\mathbf{T}$ and called the *mutual incoherence*. If we assume that the columns of $\mathbf{T}$, denoted as $\underline{t}_j$, are of unit ℓ^2-norm, M is defined as

$$M = \max_{1 \le k,j \le L, k \neq j} |\underline{t}_j^T \underline{t}_k|, \tag{35}$$

and this is equal to the maximal value in the off-diagonal absolute entries in the Gram matrix $\mathbf{T}^T\mathbf{T}$. This scalar value plays a vital role on dictating bounds on the pursuit algorithms' success.

We have already identified our desired objective as solving (P_0) as given in (34). The following property for solving (P_0) is shown in (Donoho and Elad, 2003; Gribonval, 2003)

Theorem 1: *A representation satisfying*

$$\|\underline{u}\|_0 < \frac{1}{2}\left(1 + \frac{1}{M}\right). \tag{36}$$

is necessarily the unique solution of (P_0) *as defined in (34).*

This Theorem suggests that even though (P_0) is non-convex and its solution is very complicated, if an approximation method is employed and a sparse enough solution is found, we can test it in a simple way to verify that this is the globally optimal solution of (P_0), a claim hard to make in general for non-convex problems. We see that the mutual incoherence plays a key role in the proposed optimality test.

A tighter uniqueness bound exists, paralleling the claim made in Theorem 1, but using a different measure for the richness of the dictionary - the *Spark* (also known as Kruskal-rank). Given a dictionary $\mathbf{T}$, its Spark σ is defined as the the minimal number of columns from $\mathbf{T}$ that form a linearly dependent set. The following relationship between the mutual incoherence and the Spark has been established:

Lemma 2: *For a given dictionary* $\mathbf{T}$*, its mutual incoherence and Spark are related via*

$$Spark\{\mathbf{T}\} \geq 1 + \frac{1}{M\{\mathbf{T}\}}. \tag{37}$$

Using the Spark, we have the following uniqueness result

Theorem 3: *A representation satisfying*

$$\|\underline{u}\|_0 < \frac{1}{2} Spark\{\mathbf{T}\} \tag{38}$$

is necessarily the unique solution of (P_0) *as defined in (34).*

As an example, if the dictionary is an $N \times 2N$ matrix built as a concatenation of the identity square matrix and the Hadamard one, the mutual incoherence M is $1/\sqrt{N}$, whereas the Spark is $2\sqrt{N}$ (i.e., gathering a smaller group of columns leads to linear independence – based on the Poisson formula). Thus, Theorem 2 is sharper with a bound being twice as high.

The existence of both these uniqueness results (Theorems 1 and 2) are encouraging as they motivate us to employ approximations with the hope to hit a sparse result that could be verified as the best one possible. However, reality is even more promising – it turns out that for some approximation methods we can actually guarantee that sparse solution will be found if there exists one, as we shall show immediately.

The Orthonormal Matching Pursuit (OMP) algorithm suggests searching an approximated solution by a greedy step-wise nested process, solving the alter-

native (much easier) sequence of problems,

$$\left\{\min_{\underline{u}} \|\underline{s} - \mathbf{T}\underline{u}\|_2 \text{ subject to } \|\underline{u}\|_0 = k\right\}_{k=1}^{N}. \tag{39}$$

The first problem in this set is solved easily by testing every column in $\mathbf{T}$ as a sole member in constructing $\underline{s}$, and choosing the one that leads to the minimal error. If we assume that the columns of $\mathbf{T}$ are of unit ℓ^2-norm, we obtain the best column as the one maximizing the inner product with $\underline{s}$.

Going to the next problem in this set, the previously chosen column is kept (this causes the OMP to be sub-optimal in general, and explains the term 'nested' mentioned above on the MP), and a second column candidate is again searched sweeping through the $L-1$ remaining columns. This process proceeds till the Least Squares error hits zero, implying an equality $\underline{s} = \mathbf{T}\underline{u}$, or till $k = L$ where the LS error must be zero by definition.

The following result was established in (Tropp, 2003; Donoho et al., 2003) for the performance of the OMP algorithm.

Theorem 4: *The OMP algorithm applied as an approximation to solve* (P_0) *finds the globally optimal solution of it if there exists a solution satisfying*

$$\|\underline{u}\|_0 < \frac{1}{2}\left(1 + \frac{1}{M}\right). \tag{40}$$

Thus, if there exists a sparse enough solution for (P_0), we know that it must be the best solution (based on Theorem 1), and now we can also guarantee that the OMP will find it.

The Basis Pursuit approach towards an approximate solution of (P_0) is a convexification of the problem, addressing the alternative problem

$$(P_1) \qquad \min_{\underline{u}} \|\underline{u}\|_1 \text{ subject to } \underline{s} = \mathbf{T}\underline{u}. \tag{41}$$

The new problem, (P_1) has a linear programming structure, and there are efficient ways to solve it, even in high-dimensions.

The following result was established in (Donoho and Elad, 2003) for the performance of the BP algorithm.

Theorem 5: *The BP algorithm applied as an approximation to solve* (P_0) *finds the globally optimal solution of it if there exists a solution satisfying*

$$\|\underline{u}\|_0 < \frac{1}{2}\left(1 + \frac{1}{M}\right). \tag{42}$$

Thus, a parallel result to the one referring to the OMP suggests that the BP is also expected to successfully recover the best representation if it is sparse enough.

We should note that both Theorem 4 and 5 refer to the worst-case scenario, and in general the performance of the OMP and the BP is far better that the limit $0.5(1 + 1/M)$. Returning to our previous example, with the $N \times 2N$ dictionary built as a concatenation of the identity square matrix and the Hadamard one, the mutual incoherence M is $1/\sqrt{N}$, and the bounds here refer to $\sqrt{N}/2$ as the limit number of non-zeros in the representation to guarantee uniqueness and successful performance of the OMP and the BP. In practice, much denser representations are still recovered by these algorithms. Also, empirical tests indicate that the BP performs in general better than the OMP, although it is of higher computational complexity.

4.4 Approximations with Sparsity

For a signal $\underline{s}$ we may be interested in its approximate representation rather than its exact one. Such relaxation in the passage from $\underline{s}$ to its representation could be exploited for getting a simpler description of the signal, and thus, fulfill the underlying desire originally planned in adopting a transform as a simplifying tool.

Going back to the linear methodology, we generalize (31) to be

$$\min_{\underline{u}} \|\mathbf{D}\underline{u}\|_2^2 + \lambda\|\underline{s} - \mathbf{T}\underline{u}\|_2^2. \tag{43}$$

The parameter λ controls the amount of distortion in representing $\underline{s}$. The solution in this case is easily obtained as

$$\underline{u} = \lambda\left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{T}\mathbf{T}^T\right)^{-1}\mathbf{T}^T\underline{s}, \tag{44}$$

and we see that the linearity of the overall transform is preserved. Note that for $\lambda \to \infty$, we obtain the transform as posed in (32).

One commonly used heuristic for simplifying a given representation is a shrinkage of the representation coefficients. Given the vector $\underline{u}$, this heuristic suggests shrinking values by multiplication by a constant smaller than one . This

is actually well-supported by the above approximated transform, if $\mathbf{D}$ and $\mathbf{T}$ are unitary, where we obtain

$$\underline{u} = \lambda \left(\mathbf{D}^T\mathbf{D} + \lambda \mathbf{T}\mathbf{T}^T\right)^{-1} \mathbf{T}^T \underline{s} = \frac{\lambda}{1+\lambda} \mathbf{T}^{-1} \underline{s}. \tag{45}$$

Thus, if for example, $\mathbf{D} = \mathbf{I}$, and $\mathbf{T}$ is an orthonormal wavelet transform, simplification of the representation is easily achieved by simple manipulation of the original exact representation. However, this heuristic in the general case is wrong, and a better and more rigorous method is the use of (44).

Extending similarly the sparsity-oriented definition for a forward transform reads

$$(P_{0,\lambda}) \qquad \min_{\underline{u}} \|\underline{u}\|_0 + \lambda \|\underline{s} - \mathbf{T}\underline{u}\|_2^2. \tag{46}$$

It is interesting to note that even though this problem is generally complicated to solve, a closed form solution exists when $\mathbf{T}$ is a unitary matrix. Defining $\tilde{\underline{u}} = \mathbf{T}^{-1}\underline{s}$, the above problem can be described alternatively as

$$\min_{\underline{u}} \|\underline{u}\|_0 + \lambda \|\tilde{\underline{u}}_0 - \underline{u}\|_2^2 = \min_{u_1, u_2, \cdots, u_L} \sum_{k=1}^{L} \lambda (u_k - \tilde{u}_k)^2 + |u_k|^0. \tag{47}$$

This way we got a set of L independent optimization problems with the scalars u_k as unknowns. The solution is the *hard-thresholding* operation, and commonly used in Wavelet denoising – see Section 2.2.1 and (Donoho and Johnstone, 1994; Donoho, 1993).

$$u_k = \begin{cases} \tilde{u}_k & \text{if } |\tilde{u}_k| \geq \frac{1}{\sqrt{\lambda}} \\ 0 & \text{otherwise} \end{cases}. \tag{48}$$

This way, again, we get that a manipulation of the original exact representation leads to the desired simplification, and again, this is not true for the general case, although thresholding is popular heuristic employed nevertheless.

Once again, approximations can be used in order to solve the $(P_{0,\lambda})$ problem. The idea is the replacement of the exact representation $\underline{s} = \mathbf{T}\underline{u}$ by a penalty $\|\underline{s} - \mathbf{T}\underline{u}\|_2^2$.

The OMP version for solving $(P_{0,\lambda})$ is simple and requires only changing the stopping rule of the algorithm. The same sequence of problems as in (39) is solved, and in the same manner. At each stage the error $\|\underline{s} - \mathbf{T}\underline{u}\|_2^2$ should

decrease by more than $1/\lambda$ so as to compensate for the increase in $\|\underline{u}\|_0$. When the decrease is smaller than $1/\lambda$ the OMP should be stopped.

The Basis Pursuit Denoising (BPDN) is the generalization of the Basis Pursuit for approximating the solution of $(P_{0,\lambda})$, based on solving

$$(P_{1,\lambda}) \qquad \min_{\underline{u}} \|\underline{u}\|_1 + \lambda \|\underline{s} - \mathbf{T}\underline{u}\|_2^2. \tag{49}$$

This problem has a quadratic programming structure for which there are efficient solvers. A closed form solution exists here as well for the choice of unitary $\mathbf{T}$ leading to

$$u_k = \begin{cases} \tilde{u}_k - \frac{\text{sign}\{\tilde{u}_k\}}{\lambda} & \text{if } |\tilde{u}_k| \geq \frac{1}{\lambda} \\ 0 & \text{otherwise .} \end{cases} \tag{50}$$

This operation is known as *soft–thresholding*, and it has both the influences of the hard–thresholding of the ℓ^0 and the simple shrinkage of the ℓ^2 (Donoho and Johnstone, 1994; Donoho, 1993).

Returning to the general case, no closed-form solution exists, and numerical methods are to be applied in order to solve $(P_{1,\lambda})$.

Similar to the claims in Theorems 2 and 3, analysis of these approximation methods can be discussed, relating their success to the sparsity of the representation and the mutual incoherence of the dictionary $\mathbf{T}$. Such analysis is a current topic of research, and it is hoped that in several years a better knowledge on these methods will become available.

4.5 Numerical Methods for BPDN

While theoretically known to be convex, the BPDN as posed in $(P_{1,\lambda})$ is generally not trivial to solve, and requires some skills in optimization techniques. General modern methods for quadratic programming based on interior point and active set algorithms can of-course be used as solvers. Here we will mention two alternatives that are popular among signal processing practitioners – The Iterative Reweighed Least-Squares (IRLS) and the Block–Coordinate–Relaxation method.

We start with the IRLS method as described in (Karlovitz, 1970). The basic theme here is the replacement of the original problem,

$$(P_{1,\lambda}) \qquad \min_{\underline{u}} \ \|\underline{u}\|_1 + \lambda \|\underline{s} - \mathbf{T}\underline{u}\|_2^2, \tag{51}$$

with a sequence of Least-Squares (LS) problems, exploiting the fact that highly efficient LS solvers are available.

The reason $(P_{1,\lambda})$ cannot be solved with LS in the first place is the use of the ℓ^1-norm, but if we assume that a near-optimal solution $\underline{\hat{u}}$ was found, and an update to the solution is desired, we can replace $\|\underline{u}\|_1$ with an ℓ^2-norm expression of the form

$$\|\underline{u}\|_1 = \sum_{k=1}^{L} |u_k| = \sum_{k=1}^{L} \frac{u_k^2}{|u_k|} \approx \sum_{k=1}^{L} \frac{u_k^2}{|\hat{u}_k|} = \underline{u}^T \mathbf{W}(\underline{\hat{u}})\underline{u}. \tag{52}$$

The matrix $\mathbf{W}(\underline{\hat{u}})$ is a diagonal matrix of size $L \times L$. Its main diagonal contains the reciprocals of $|\underline{\hat{u}}|$. For numerical stability, for near zero entries in $|\underline{\hat{u}}|$ the weight is chosen as a fixed high value (this has the effect of using a slightly distorted norm definition, rounded around the origin to avoid singularity).

Thus, solving the new problem

$$(\hat{P}_{1,\lambda}) \qquad \min_{\underline{u}} \ \underline{u}^T \mathbf{W}(\underline{\hat{u}})\underline{u} + \lambda \|\underline{s} - \mathbf{T}\underline{u}\|_2^2 \tag{53}$$

could lead to the updated solution, and can be applied using a LS solver, since the solution is given by

$$\underline{u} = \lambda \left(\mathbf{W}(\underline{\hat{u}}) + \lambda \mathbf{T}^T\mathbf{T}\right)^{-1} \mathbf{T}^T \underline{s}. \tag{54}$$

Given the updated solution, we can redo the above process with updated weights in $\mathbf{W}(\underline{\hat{u}})$. In order to guarantee convergence, a relaxation maybe needed, where a one-pole smoothing is done on the sequence of results to slow down the changes. Alternatively, instead of exact LS solution, few iterations using only multiplications by $\mathbf{T}$ and its adjoint can be used to update the solution, then updating the weights. Due to the use of a non-exact LS solver, the smoothing becomes implicit.

An alternative algorithm with a similar flavor is the Block Coordinate Relaxation (BCR) method (Bruce et al., 1998). Again, the original $(P_{1,\lambda})$ is replaced by a sequence of easier problems - this time owing to a specific assumption on the structure of $\mathbf{T}$. We assume that the dictionary $\mathbf{T}$ is built as an amalgam of J different unitary matrices $\{\mathbf{T}_j\}_{j=1}^{J}$, namely,

$$\mathbf{T} = [\mathbf{T}_1, \ \mathbf{T}_2, \ \ \ldots \ , \mathbf{T}_J]. \tag{55}$$

The representation vector $\underline{u}$ can be also broken in this case to J disjoint parts of N entries each, denoted by $\{\underline{u}_j\}_{j=1}^J$. The problem $(P_{1,\lambda})$ can now be rewritten as

$$(P_{1,\lambda}) \qquad \min_{\underline{u}_1,\underline{u}_2,\cdots,\underline{u}_L} \sum_{j=1}^{J} \|\underline{u}_j\|_1 + \lambda \left\| \underline{s} - \sum_{j=1}^{J} \mathbf{T}_j \underline{u}_j \right\|_2^2 . \tag{56}$$

While this problem is difficult to solve in the general case, if we assume that $\{\underline{u}_j\}_{j=2}^J$ are all known and seek the optimal $\underline{u}_1$, there is a closed-form solution we can exploit. Since $\{\underline{u}_j\}_{j=2}^J$ are known, the new optimization task is

$$\min_{\underline{u}_1} \|\underline{u}_1\|_1 + \lambda \left\| \left(\underline{s} - \sum_{j=2}^{J} \mathbf{T}_j \underline{u}_j \right) - \mathbf{T}_1 \underline{u}_1 \right\|_2^2 . \tag{57}$$

Since $\mathbf{T}_1$ is unitary, this problem has the same structure as in (46) were we have given a closed-form solution being the soft–thresholding. This thresholding should be applied on the representation

$$\mathbf{T}_1^T \left(\underline{s} - \sum_{j=2}^{J} \mathbf{T}_j \underline{u}_j \right), \tag{58}$$

and the first block of coordinates in $\underline{u}$ is updated. This process could be repeated sweeping through the various parts of $\underline{u}$, always updating one while assuming all the other $J-1$ blocks fixed. This algorithm is proven to converge to the solution of $(P_{1,\lambda})$.

5 Morphological Component Analysis

The task of decomposing signals into their building atoms is of great interest for many applications. In such problems a typical assumption is made that the given signal is a linear mixture of several source signals of more coherent origin. These kind of problems have drawn a lot of research attention in last years. Independent Component Analysis (ICA) and sparsity methods are typically used for the separation of signal mixtures with varying degrees of success. A classic example is the cocktail party problem where a sound signal containing several concurrent speakers is to be decomposed into the separate speakers. In image processing, a parallel situation is encountered for example in cases of photographs containing transparent layers.

In this section we present the way to decompose a signal into its building parts using the Morphological Component Analysis (MCA) methodology. We start with a model of the problem, and show how sparsity plays a vital role in our ability to separates the different ingredients from each other. We discuss theoretic justification for the MCA method, and show some applications that are built around it.

5.1 Separating signals to their ingredients

Assume that the input signal to be processed has N samples, organized as a 1D vector, $\underline{s} \in \mathbf{R}^N$. Assume that the signal $\underline{s}$ is a linear combination of two parts, $\underline{s} = \underline{s}_A + \underline{s}_B$, where $\underline{s}_A$ and $\underline{s}_B$ represent two different types of signals to be decomposed. The entire analysis presented here can be extended to treat any arbitrary number of data types, and for simplicity we assume that only two such types are to be separated.

Our model assumes the following to hold true:

(1) For every possible signal $\underline{s}_A$ of the first type, there exists an over-complete dictionary $\mathbf{T}_A \in \mathbf{M}^{N \times L_A}$ (where typically $L_A \gg N$) such that solving

$$\underline{\alpha}_A^{opt} = \text{Arg}\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \text{ subject to: } \underline{s}_A = \mathbf{T}_A\underline{\alpha} \tag{59}$$

leads to a very sparse solution (i.e. $\|\underline{\alpha}_A^{opt}\|_0$ is very small). The definition in the above equation is essentially the overcomplete transform of $\underline{s}_A$, yielding a representation $\underline{\alpha}_A$.

(2) For every possible signal $\underline{s}_B$ of the second type, solving

$$\underline{\alpha}_{AB}^{opt} = \text{Arg}\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \text{ subject to: } \underline{s}_B = \mathbf{T}_A\underline{\alpha} \tag{60}$$

leads to a very non-sparse solution (i.e. $\|\underline{\alpha}_A B^{opt}\|_0$ is very big). This requirement suggests that the dictionary $\mathbf{T}_A$ is distinguishing between the two types of signals to be separated.

(3) Similar to the above, we assume that a dictionary $\mathbf{T}_B \in \mathbf{M}^{N \times L_B}$ can be proposed, such that it leads to very sparse representations for every possible signal $\underline{s}_B$ of the second type, and it is also leading to highly non-sparse results when applied on signals of the first type.

Thus, the two dictionaries $\mathbf{T}_A$ and $\mathbf{T}_B$ play a role of discriminants between the two content types. If we have two training sets of the first and the second signal types, $\{\underline{s}_A(k)\}_k$ and $\{\underline{s}_B(j)\}_j$, a possible measure of fidelity for the chosen dictionary $\mathbf{T}_A$ is the functional

$$\text{Quality}\{\mathbf{T}_A\} = \frac{\sum_k \|\underline{\alpha}_A^{opt}(k)\|_0}{\sum_j \|\underline{\alpha}_B^{opt}(j)\|_0} \tag{61}$$

$$\text{where:} \quad \begin{aligned} &\left\{\underline{\alpha}_A^{opt}(k) = \text{Arg}\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \text{ subject to: } \underline{s}_A(k) = \mathbf{T}_A\underline{\alpha}\right\}_k \\ &\left\{\underline{\alpha}_B^{opt}(j) = \text{Arg}\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \text{ subject to: } \underline{s}_B(j) = \mathbf{T}_A\underline{\alpha}\right\}_j. \end{aligned}$$

and similar expression can be written for the $\mathbf{T}_B$ choice. This function of the dictionary is measuring the relative sparsity between the type-A family of signals and the B-type one. This, or a similar measure, could be used for the design of the proper choice of $\mathbf{T}_A$, but in this paper we assume that the choice of dictionaries is already done somehow.

For an arbitrary signal $\underline{s}$ containing both type A and type B contents as a linear combination, we propose to seek the sparsest of all representations over the augmented dictionary containing both $\mathbf{T}_A$ and $\mathbf{T}_B$. Thus we need to solve

$$\begin{aligned} \{\underline{\alpha}_A^{opt}, \underline{\alpha}_B^{opt}\} = &\underset{\{\underline{\alpha}_A, \underline{\alpha}_B\}}{\text{Argmin}} \|\underline{\alpha}_A\|_0 + \|\underline{\alpha}_B\|_0 \\ &\text{subject to: } \underline{s} = \mathbf{T}_A\underline{\alpha}_A + \mathbf{T}_B\underline{\alpha}_B. \end{aligned} \tag{62}$$

This optimization task is likely to lead to a successful separation of the image content, such that $\mathbf{T}_A\underline{\alpha}_A$ is mostly of type A and $\mathbf{T}_B\underline{\alpha}_B$ is mostly with type-B content. The reason for this expectation relies on the assumptions made earlier about $\mathbf{T}_A$ and $\mathbf{T}_B$ being very efficient in representing one phenomenon and being highly non-effective in representing the other signal type.

Two difficulties we need to consider are that (a) While sensible from the point of view of the desired solution, the problem formulated in Equation (62) is non-convex and hard to solve; and (b) The given signal will generally not decompose cleanly into the two content types due to additive noise or model mismatch.

As we have seen in the previous section, simplifying (62) with the MP or the BP formulation is a natural step with empirical and theoretical justification that will solve the first difficulty mentioned. Also, changing the constraint by a penalty allowing for an approximate representation is desired, in order to solve the second problem. With the BP approach, the alternative decomposition problem reads

$$\{\underline{\alpha}_A^{opt},\ \underline{\alpha}_B^{opt}\} = \underset{\{\underline{\alpha}_A,\ \underline{\alpha}_B\}}{\text{Argmin}}\ \|\underline{\alpha}_A\|_1 + \|\underline{\alpha}_B\|_1 + \lambda\|\underline{s} - \mathbf{T}_A\underline{\alpha}_A - \mathbf{T}_B\underline{\alpha}_B\|_2^2. \quad (63)$$

In order to translate the above idea into a practical algorithm we should answer three major questions: (i) Is there a theoretical backup to the heuristic claims made here? (ii) How should we choose the dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$? and (iii) How should we numerically solve the obtained optimization problem in a traceable way? These three questions are addressed in the coming sections.

5.2 Why should it work? Theoretical analysis

Our theoretical analysis embarks from Equation (62), which stands as the basis for the separation process. This equation could also be written differently as

$$\underline{\alpha}_{all}^{opt} = \begin{bmatrix} \underline{\alpha}_A^{opt} \\ \underline{\alpha}_B^{opt} \end{bmatrix} = \text{Arg}\ \min_{\{\underline{\alpha}_A,\ \underline{\alpha}_B\}} \left\| \begin{bmatrix} \underline{\alpha}_A \\ \underline{\alpha}_B \end{bmatrix} \right\|_0 \quad (64)$$

$$\text{subject to: } \underline{s} = \begin{bmatrix} \mathbf{T}_A\ \mathbf{T}_B \end{bmatrix} \begin{bmatrix} \underline{\alpha}_A \\ \underline{\alpha}_B \end{bmatrix} = \mathbf{T}_{all}\underline{\alpha}_{all}.$$

Thus, based on Theorem 1 and 2 we have that

Theorem 6: *Given a signal $\underline{s}$ being a sparse mixture of type-A and type-B contents, such that*

$$\|\underline{\alpha}_{all}\|_0 = \|\underline{\alpha}_A\|_0 + \|\underline{\alpha}_B\|_0 < \frac{1}{2}\left(1 + \frac{1}{M\{\mathbf{T}_{all}\}}\right) \leq Spark\{\mathbf{T}_{all}\}, \quad (65)$$

this mixture is necessarily the unique solution of (P_0) as defined in (64).

The inner requirement using the mutual incoherence is weaker than the one using the Spark and thus more restrictive. However, in many cases evaluation

of the Spark is difficult and the alternative weaker bound can used rather easily.

A direct consequence of Theorems 4 and 5 is the following result

Theorem 7: *Given a signal $\underline{s}$ being a sparse mixture of type-A and type-B contents, such that*

$$\|\underline{\alpha}_{all}\|_0 = \|\underline{\alpha}_A\|_0 + \|\underline{\alpha}_B\|_0 < \frac{1}{2}\left(1 + \frac{1}{M\{\mathbf{T}_{all}\}}\right), \tag{66}$$

this mixture will be recovered correctly by both the MP/BP methods.

Thus, we see that if indeed our type-A and type-B contents were composed as sparse linear combination of atoms from $\mathbf{T}_A$ and $\mathbf{T}_B$, respectively, our decomposition will stand as the global minimum of (64), and moreover, it will be recovered successfully from the application of either the MP or the BP methods - both being computationally traceable.

Actually, stronger claims could be given if we assume a successful choice of dictionaries $\mathbf{T}_A$ and $\mathbf{T}_B$, and consider the task as separation only and not exact recovery of the atom composition per every content type alone. Let us define a variation of the Spark that refers only to the interface between atoms from the two dictionaries, and not to interactions of atoms within them.

Definition 8: *Given two matrices $\mathbf{T}_A$ and $\mathbf{T}_B$, their Inter-Spark ($\sigma_{A\leftrightarrow B} = Spark\{\mathbf{T}_A, \mathbf{T}_B\}$) is defined as the minimal number of columns from the concatenated matrix $[\mathbf{T}_A, \mathbf{T}_B]$ that form a linearly dependent set, and such that columns from both matrices participate in this combination.*

With this defined measure, we can propose the following claim (stated without proof) as a variation on Theorem 6.

Theorem 9: *Given a signal $\underline{s}$ known to be a sparse mixture of type-A and type-B contents, such that*

$$\|\underline{\alpha}_{all}\|_0 = \|\underline{\alpha}_A\|_0 + \|\underline{\alpha}_B\|_0 < \frac{1}{2} Spark\{\mathbf{T}_A, \mathbf{T}_B\}, \tag{67}$$

and

$$\|\underline{\alpha}_A\|_0,\ \|\underline{\alpha}_B\|_0 > 0, \tag{68}$$

this mixture is necessarily the unique mixture solution of (P_0) as defined in (64).

The benefit in using this Theorem is that in general

$$Spark\{[\mathbf{T}_A, \mathbf{T}_B]\} = \min(\sigma_A, \sigma_B, \sigma_{A \hookrightarrow B}), \tag{69}$$

and this value could be quite small if either σ_A or σ_B are small, implying a weak claim in Theorem 6. However, as we focus on the separation task, the bound is dependent on the inter-Spark alone. Alternative approach, simpler but also weaker, towards the same analysis, could be proposed based on the notion of mutual incoherence.

based on the Inter-Spark we may propose an extension to Theorem 7 presenting a more generous bound, but we choose to stop the analysis here, as we concentrate in this paper on the applicative part. As we mentioned before, the bounds given here are quite restrictive and does not reflect truly the much better empirical results. We regard this analysis as merely supplying a theoretical motivation, rather than complete justification for the later results. We should also note that the above analysis is coming form a *worst-case* point of view (e.g., see the definition of the *Spark*), as opposed to the average case we expect to encounter empirically. Nevertheless, the ability to prove perfect separation in a stylized application without noise and with restricted success is of great benefit as a proof of concept. Further work is required to extend the theory developed here to the average case.

5.3 Toy problem - feel the idea work

In order to demonstrate the gap between theoretical results and empirical evidence in Basis Pursuit separation performance, figure 22 presents a simulation of the separation task for the case of signal $\underline{s}$ of length $N = 64$, a dictionary built as the combination of the Hadamard unitary matrix (assumed to be $\mathbf{T}_A$) and the identity matrix (assumed to be $\mathbf{T}_B$). Thus, type-A signals are characterized as being periodic step functions, whereas type-B signals are isolated spikes.

We randomly generate sparse representations with varying number of non-zeros in the two parts of the representation vector (of length 128), and present the empirical probability (based on averaging 100 experiments) to recover correctly the separation.

For this case, Theorem 7 suggests that the number of non-zeros in the two parts should be smaller than $0.5 * (1 + 1/M) = (1 + \sqrt{64})/2 = 4.5$. Actually a better bound exists for this case in (Elad and Bruckstein, 2001; Elad and Bruckstein, 2002) due to the construction of the overall dictionary as a combination of two unitary matrices. Thus, the better bound is $(\sqrt{2} - 0.5))/M = 7.3$. Both

these bounds are overlayed on the empirical results in the figure, and as can be seen, Basis Pursuit succeeds well beyond the bound. Moreover, extensive experiments show that this trend is expected to strengthen as the signal size grows, since than the worst-case-scenarios (for which the bounds refer to) become of smaller probability and of less influence on the average result.

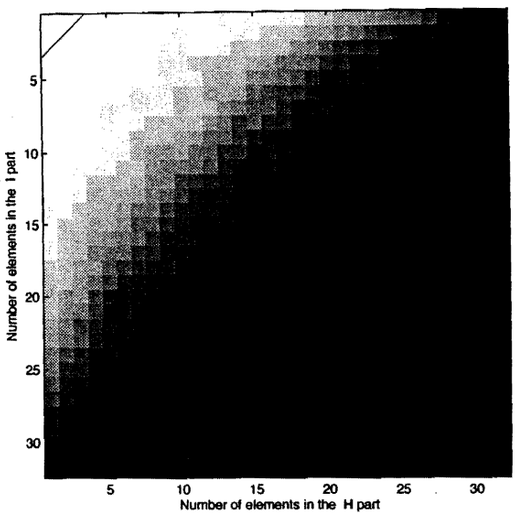


Fig. 22. Empirical probability of success of the Basis Pursuit algorithm for separation of sources. Per every sparsity combination, 100 experiments are performed and the success rate is computed. Theoretical bounds are also drawn for comparison.

5.4 MCA in Practice

Returning to the separation process, its idea presentation is posed in Equation (62), and a BP approximation of it is given in (63). We need to solve an optimization problem of the form

$$\{\underline{\alpha}_A^{opt},\ \underline{\alpha}_B^{opt}\} = \underset{\{\underline{\alpha}_A,\ \underline{\alpha}_B\}}{\text{Argmin}}\ \|\underline{\alpha}_A\|_1 + \|\underline{\alpha}_B\|_1 + \lambda\,\|\underline{s} - \mathbf{T}_A\underline{\alpha}_A - \mathbf{T}_B\underline{\alpha}_B\|_2^2 .$$

Dealing with images, the dimensions involved are too big ($N \approx 10^6$, $L \gg N$) to allow for direct storage of the dictionary matrices, or their inversion. Thus, we seek methods that are built around the use of multiplication by $\mathbf{T}_A$ or $\mathbf{T}_B$, and their adjoint – both assumed to be practical.

Another complicating factor is L - the length of the representation vector $\underline{\alpha}_{all}$. If for example $L = 100N$ (implying a redundancy of factor 100), it means that storing and manipulating the solution of this problem requires a memory of

100 images. Instead of solving this optimization problem, finding two representation vectors $\{\underline{\alpha}_A^{opt}, \underline{\alpha}_B^{opt}\}$, let us reformulate the problem so as to get the two signal types, $\underline{s}_A$ and $\underline{s}_B$, as our unknowns. This way, if we return for the example mentioned above, we seek two images rather than 100.

Define $\underline{s}_A = \mathbf{T}_A\underline{\alpha}_A$ and similarly $\underline{s}_B = \mathbf{T}_B\underline{\alpha}_B$. Given $\underline{s}_A$, we can recover $\underline{\alpha}_A$ as $\underline{\alpha}_A = \mathbf{T}_A^+\underline{s}_A + \underline{r}_A$ where $\underline{r}_A$ is an arbitrary vector in the null-space of $\mathbf{T}_A$. Put these back into (63) we obtain

$$\begin{aligned}\{\underline{s}_A^{opt}, \underline{s}_B^{opt}\} = \underset{\{\underline{s}_A, \underline{s}_B, \underline{r}_A, \underline{r}_B\}}{\text{Argmin}} & \|\mathbf{T}_A^+\underline{s}_A + \underline{r}_A\|_1 + \|\mathbf{T}_B^+\underline{s}_B + \underline{r}_B\|_1 \\ & + \lambda\|\underline{X} - \underline{s}_A - \underline{s}_B\|_2^2 \\ & \text{Subject to: } \mathbf{T}_A\underline{r}_A = 0\ ,\ \mathbf{T}_B\underline{r}_B = 0.\end{aligned} \tag{70}$$

The term $\mathbf{T}_A^+\underline{s}_A$ is an overcomplete linear transform of the image $\underline{s}_A$. Similarly, $\mathbf{T}_B^+\underline{s}_B$ is an overcomplete linear transform of the type-B signal part.

In our attempt to replace the representation vectors as unknowns, we see that we have a pair of residual vectors to be found as well. If we choose (rather arbitrarily at this stage) to assign those vectors as zeros we obtain the problem

$$\{\underline{s}_A^{opt}, \underline{s}_B^{opt}\} = \underset{\{\underline{s}_A, \underline{s}_B\}}{\text{Argmin}} \|\mathbf{T}_A^+\underline{s}_A\|_1 + \|\mathbf{T}_B^+\underline{s}_B\|_1 + \lambda\|\underline{s} - \underline{s}_A - \underline{s}_B\|_2^2. \tag{71}$$

We can justify the choice $\underline{r}_A = \underline{0}$, $\underline{r}_B = \underline{0}$ in several ways:

Bounding function: Consider the function posed on (70) as a function of $\underline{s}_A$, $\underline{s}_B$, where per every possible values of those two images we optimize with respect to $\underline{r}_A$, $\underline{r}_B$. Comparing this function to the one we have suggested in (71), the new function could be referred to as an upper bounding surface to the true function. Thus, in minimizing it instead, we can guarantee that the true function to be minimized is of even lower value.

Relation to the BCR algorithm: Comparing (71) to the block-coordinate relaxation method presented earlier, we see close resemblance. This will become a complete equivalence if we assume that the dictionaries involved contain just one unitary part. Thus, in a way we may refer to the approximation we have made here as a method to generalize the block-coordinate-relaxation method for the non-unitary case.

Relation to MAP: The expression written as penalty function in (71) has a Maximal-A-Posteriori estimation flavor to it. It suggests that the given image $\underline{s}$ is known to originate from a linear combination of the form $\underline{s}_A + \underline{s}_B$, contaminated by Gaussian noise - this part comes from the likelihood function $\|\underline{s} - \underline{s}_A - \underline{s}_B\|_2^2$. We further assume that both type A and type-B parts come from a Gibbs distribution of the form Const$\cdot\exp\left(-\beta_{A/B}\|\mathbf{T}_{A/B}^+\underline{s}_{A/B}\|_1\right)$.

While different from our original point of view, these assumptions are reasonable and not far from the Basis Pursuit approach.

The bottom line to all this discussion is that we have chosen an approximation to our true minimization task, and with it managed to get a simplified optimization problem, for which an effective algorithm can be proposed. Our minimization task is thus given by

$$\min_{\{\underline{s}_A,\ \underline{s}_B\}} \left\|\mathbf{T}_A^{+}\underline{s}_A\right\|_1 + \left\|\mathbf{T}_B^{+}\underline{s}_B\right\|_1 + \lambda\left\|\underline{s}-\underline{s}_A-\underline{s}_B\right\|_2^2. \tag{72}$$

The algorithm we use is based on the Block-Coordinate-Relaxation method (Bruce et al., 1998), with some required changes due to the non-unitary transforms involved. The algorithm is given below:

1. Initialize $L_{\max}$, number of iterations, and threshold $\delta = \lambda \cdot L_{\max}$.
2. Perform J times:
   - Part A - Update of $\underline{s}_B$ assuming $\underline{s}_A$ is fixed:
     - Calculate the residual $\underline{R} = \underline{s} - \underline{s}_A$.
     - Calculate $\underline{\alpha}_B = \mathbf{T}_B^{+}\underline{R}$.
     - Soft threshold the coefficient $\underline{\alpha}_B$ with the δ threshold and obtain $\hat{\underline{\alpha}}_B$.
     - Reconstruct $\underline{s}_B$ by $\underline{s}_B = \mathbf{T}_B\hat{\underline{\alpha}}_B$.
   - Part B - Update of $\underline{s}_A$ assuming $\underline{s}_A$ is fixed:
     - Calculate the residual $\underline{R} = \underline{s} - \underline{s}_B$.
     - Calculate $\underline{\alpha}_A = \mathbf{T}_A^{+}\underline{R}$.
     - Soft threshold the coefficient $\underline{\alpha}_A$ with the δ threshold and obtain $\hat{\underline{\alpha}}_A$.
     - Reconstruct $\underline{s}_A$ by $\underline{s}_A = \mathbf{T}_A\hat{\underline{\alpha}}_A$.
3. Update the threshold by $\delta = \delta - \lambda$.
4. If $\delta > \lambda$, return to Step 2. Else, finish.

The numerical algorithm for minimizing (72).

In the above algorithm, soft threshold is used due to our formulation of the ℓ^1 sparsity penalty term. However, as we have explained earlier, the ℓ^1 expression is merely a good approximation for the desired ℓ^0 one, and thus, replacing the soft by a hard threshold towards the end of the iterative process may lead to better results.

We chose this numerical scheme over the Basis Pursuit interior-point approach in (Chen et al., 1998), because it presents two major advantages:

- We do not need to keep all the transformations in memory. This is particularly important when we use redundant transformations such the undecimated wavelet transform or the curvelet transform.
- We can add different constraints on the components. As we shall see next, Total-Variation on some of the content types may support the separation

task, and other constraints, such as positivity, can easily be added as well.

5.5 Applications - some examples and results

5.5.1 1D elementary example

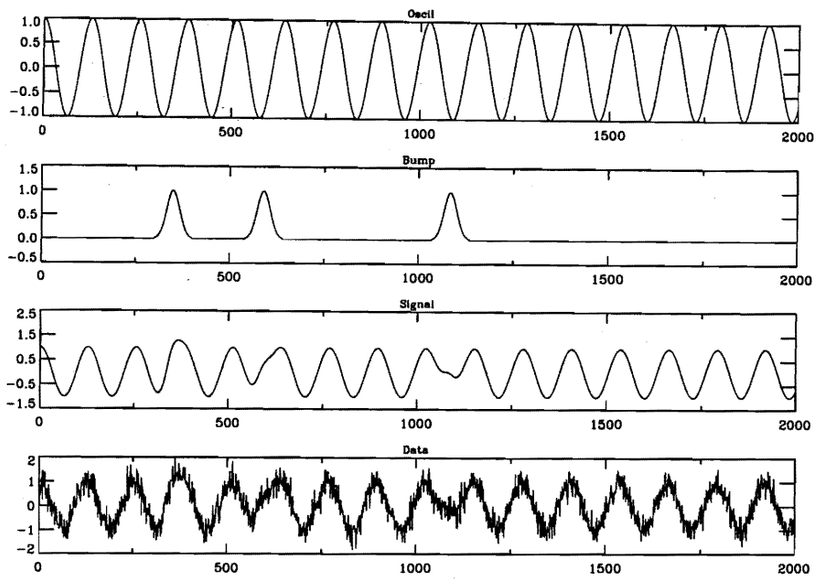


Fig. 23. From top to bottom, oscillating component, component with bumps, coaddition of both, and simulated data

Figure 23 shows an example of signal composed of two components, one presenting oscillations and the second three localized bumps. The number of samples is 2000. Gaussian noise have added to the signal ($\sigma = 0.1$) (see Fig. 23 bottom). Using the local DCT with a block size equal to 256, and the isotropic à trous wavelet transform (with ten scales), we have obtained a decomposition shown in Figure 24. From top to bottom, we see the reconstructed oscillating component (continuous line) and the original oscillating component overplotted (dashed line), the reconstructed component with bumps (continuous line) and the original component overplotted (dashed line), the coaddition of both recovered signals (continuous line) and original signal overplotted (dashed line), and the residual. This decomposition has been obtained with thirty iterations.

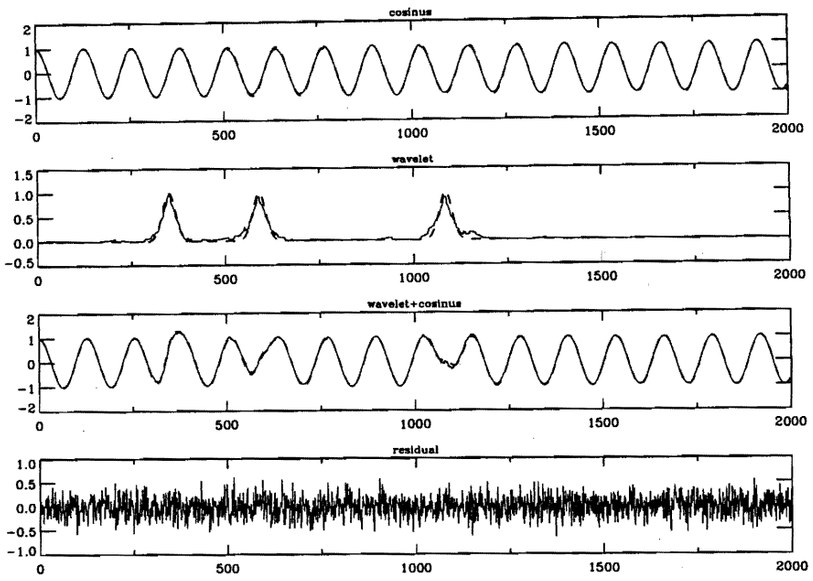


Fig. 24. From top to bottom, reconstructed oscillating component (continuous line) and original oscillating component overplotted (dashed line), reconstructed component with bumps (continuous line) and original component overplotted (dashed line), coaddition of both recovered signals (continuous line) and original signal overplotted (dashed line), residual.

5.5.2 Lines - Points separation

Figure 25 illustrates the separation result in the case where the input image (256×256) contains only lines and isotropic 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 the isotropic Gaussians due to the isotropy of the wavelet function (Starck et al., 1998), while the second is optimal to represent lines (Candès and Donoho, 1999b). Figure 25 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.

The above experiment is synthetic and through it we validate the proper behavior of the numerical scheme proposed. While being synthetic, this experiment has also high relevance for astronomical data processing where stars look like Gaussian and where images may also contain anisotropic features (dust emission, supernovae remnants , filaments, ...). Separation of these compo-

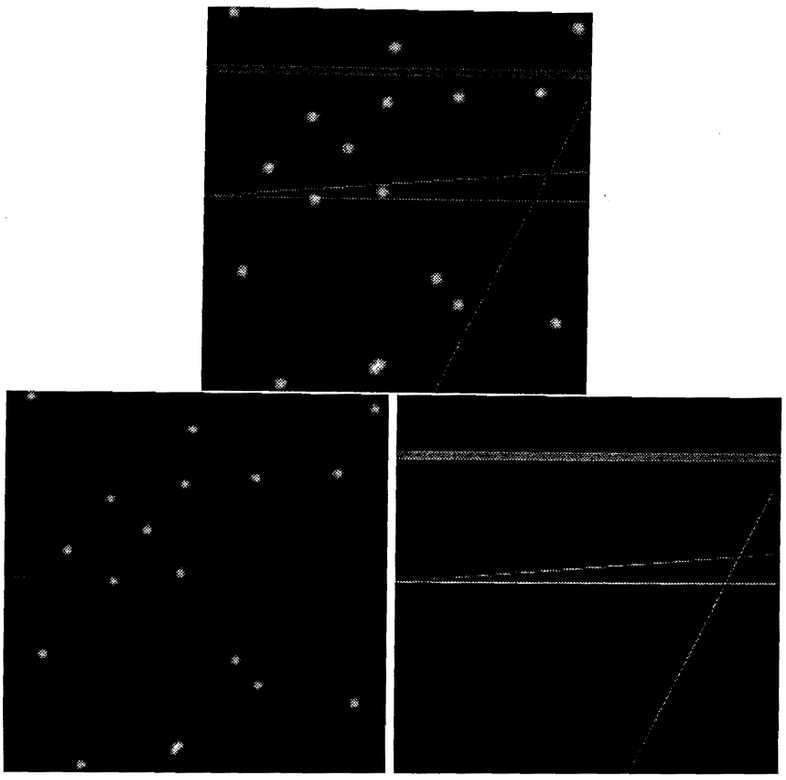


Fig. 25. Top, original image containing lines and Gaussians. Bottom left, reconstructed image for the à trous wavelet coefficient, bottom right, reconstructed image from the Ridgelet coefficients.

nents is very important for the analysis of this type of images.

5.5.3 Experiment on real astronomical data

Fig. 26 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.

This image, noted D_{10}, has been decomposed using wavelets, ridgelets, and curvelets. Fig. 26 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. 26 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. 26 bottom right). The galaxy has well been detected in the wavelet space, while all stripping artifact have been capted by the ridgelets and curvelets.

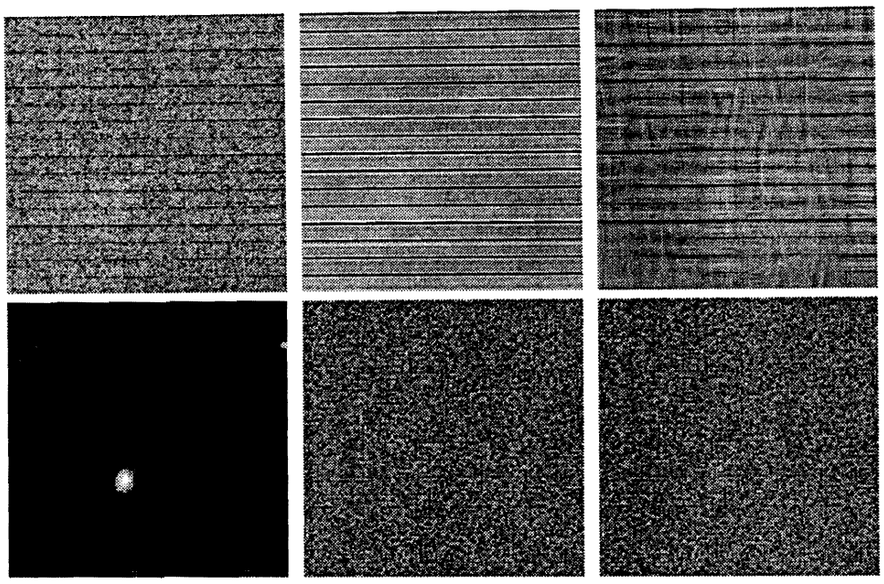


Fig. 26. Upper left, galaxy SBS 0335-052 (10 μm), upper middle, upper right, and bottom left, reconstruction respectively from the ridgelet, the curvelet and wavelet coefficients. Bottom middle, residual image. Bottom right, artifact free image.

5.5.4 Separation of Texture from Piecewise-Smooth Content

An interesting and complicated image content separation problem is the one targeting decomposition of an image to texture and piece-wise-smooth (cartoon) parts. Such separation finds applications in in image coding, and in image analysis and synthesis (see for example (Bertalmio et al., 2003)).

A theoretic characterization of textures proposed recently by Meyer (2002) was used by Vese and Osher (2003), and Aujol et al. (2003) for the design of such image separation algorithm, and these pioneering contributions awaken this application field. The approach advocated by Vese and Osher (2003) is built on variational grounds, extending the notion of Total-Variation (Rudin et al., 1992).

Here we demonstrate that the MCA is capable of separating these image content types, and as such poses an alternative method to the variational one mentioned above. More on this approach can be found in (Starck et al., 2003a).

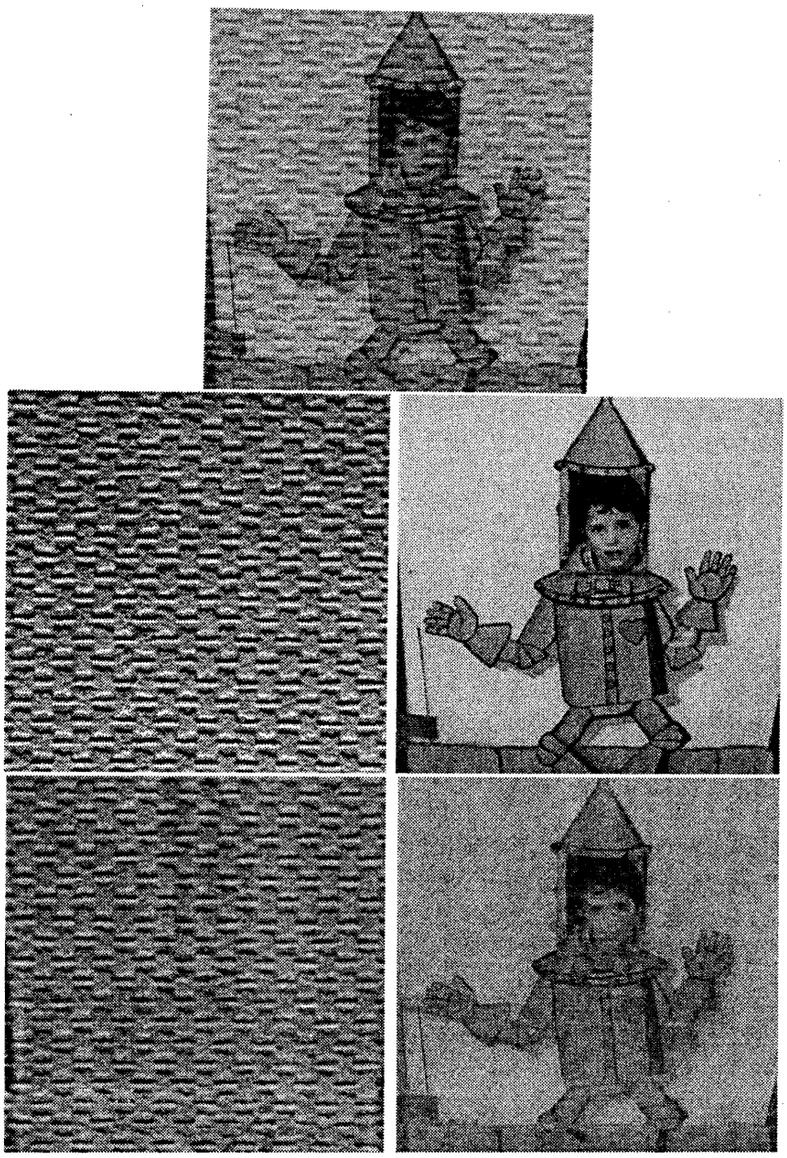


Fig. 27. Original simulated image and DCT reconstructed component. Top - the combination image, Middle left - original texture part, Middle right - Original natural part, Bottom left - separated texture part, Bottom right - separated natural part.

For the texture description, the DCT seems to have good properties due to the natural periodicity. If the texture is not homogeneous, a local DCT should be preferred. Characterizing the cartoon part of the image could be done in

various ways, depending on the image content. For images containing lines of a fixed size, the local ridgelet transform will be a good dictionary candidate. More generally the curvelet transform represents well edges in images, and could be a good candidate 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.

Assume hereafter that we use the DCT for the texture - denoted as $\mathbf{T}_A = \mathcal{D}$. Assume further that given the representation coefficients of this transform, we have an inversion process of these DCT coefficients, denoted as $\mathcal{D}^+$ (with a clear abuse of notations). In such an inversion we refer to the frame approach that generalizes the inverse by a pseudo-inverse. Similarly, we choose the curvelet transform for the natural scene part, denote it by $\mathbf{T}_B = \mathcal{C}$, and denote its inverse by $\mathcal{C}^+$.

Returning to the separation process as posed earlier, we have two unknowns - $\underline{s}_D$ and $\underline{s}_C$ - the texture and the piecewise smooth images. The optimization problem to be solved is

$$\min_{\{\underline{s}_D,\ \underline{s}_C\}} \|\mathcal{D}\underline{s}_D\|_1 + \|\mathcal{C}\underline{s}_C\|_1 + \lambda \|\underline{s} - \underline{s}_D - \underline{s}_C\|_2^2 + \gamma TV\{\underline{s}_C\}. \tag{73}$$

In this optimization problem we support the choice of the cartoon dictionary by adding another penalty term based on the Total-Variation on the cartoon image part.

Figure 27 shows an original image (top), the two original parts the image was composed from (middle left and right), and the separated texture part (bottom left) and the separated cartoon part (bottom right). As we can see, the separation is reproduced rather well.

Figure 28 shows respectively the `Barbara` image, the reconstructed local cosine component and the reconstructed curvelet component.

6 Conclusion

The need to decompose signals into linearly–joined atomic parts belonging to different behaviors finds many appealing applications in signal and image processing. Past approach to this problem was based on statistical considerations leading to Independent Component Analysis and its variants. In this paper we have presented an alternative deterministic methodology, based on sparsity, towards the same problem, named *Morphological Component Analysis* (MCA). We have anchored this method with some conclusive theoretical

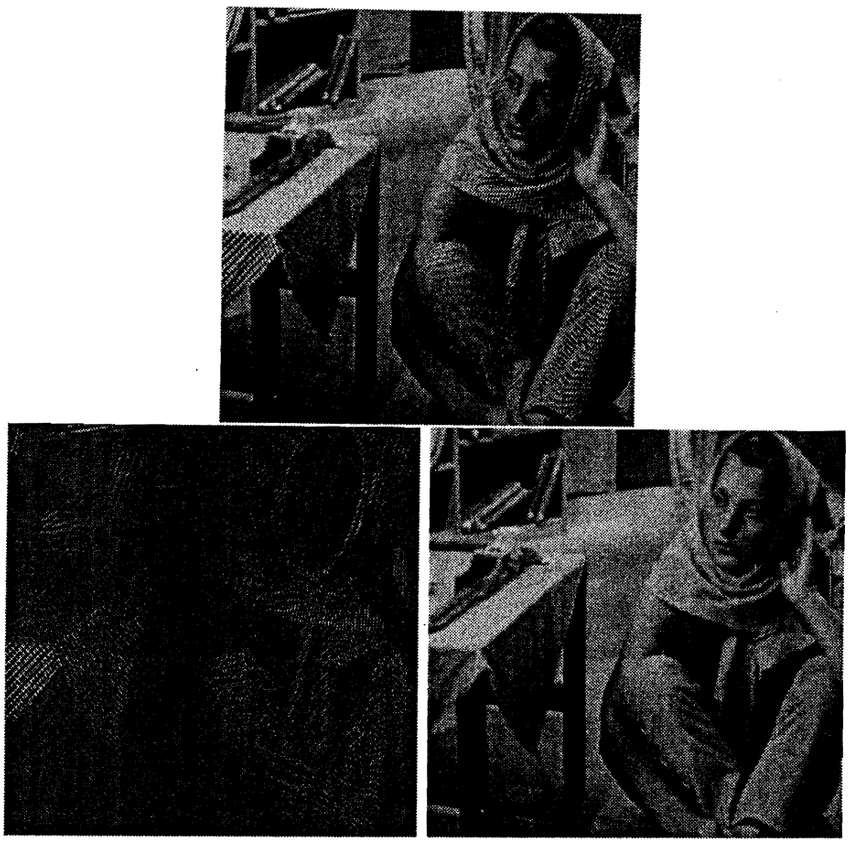


Fig. 28. Top, original Barbara image (512x512). Bottom left, reconstructed DCT component. Bottom left right, reconstructed curvelet component.

results, essentially guaranteeing successful separation under some conditions. We have also demonstrated its use in several applications for images.

A major role in the application of the MCA method is played by the dictionaries chosen for the decomposition. This paper gives a wide survey of possible fast–implementation dictionaries taken from the wavelet theory, along with ways to use these dictionaries in linear and non-linear settings. We have seen that the combination of the multiscale transforms lead to a powerful method in the MCA framework. For some applications such denoising or deconvolution, MCA is however not the best way to combine the different transforms and to benefit of the advantages of each of them. Indeed, it has been shown that a very high quality restoration can be achieved in an efficient way using several multiscale transforms without having to perform a full decomposition of the original image (Starck et al., 2001; Starck et al., 2003b).

Acknowledgments

The authors would like to thank Philippe Querre for implementing the 1D version of the MCA algorithm.

References

Antoine, J. and Murenzi, R.: 1994, Two dimensional wavelet analysis in image processing, *Physicalia Mag.* **16**, 105

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

Arneodo, A., Argoul, F., Bacry, E., Elezgaray, J., and Muzy, J. F.: 1995, *Ondelettes, Multifractales et Turbulences*, Diderot, Arts et Sciences, Paris

Aujol, J., Aubert, G., Blanc-Feraud, L., and Chambolle, A.: 2003, *Image Decomposition: Application to Textured Images and SAR Images*, Technical Report ISRN I3S/RR-2003-01-FR, INRIA - Project ARIANA, Sophia Antipolis

Averbuch, A., Coifman, R., Donoho, D., Israeli, M., and Waldén, J.: 2001, 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.*, To appear

Bertalmio, M., Vese, L., Sapiro, G., and Osher, S.: 2003, Simultaneous structure and texture image inpainting, *IEEE Transactions on Image Processing* **12(8)**, 882

Bruce, A., Sardy, S., and Tseng, P.: 1998, Block coordinate relaxation methods for nonparametric signal de-noising, *Proceedings of the SPIE - The International Society for Optical Engineering* **3391**, 75

Burt, P. and Adelson, A.: 1983, The Laplacian pyramid as a compact image code, *IEEE Transactions on Communications* **31**, 532

Candès, E.: 1998, *Ridgelets: theory and applications*, *Ph.D. thesis*, Stanford University

Candès, E. and Donoho, D.: 1999a, *Curvelets*, Technical report, Statistics, Stanford University

Candès, E. and Donoho, D.: 1999b, Ridgelets: the key to high dimensional intermittency?, *Philosophical Transactions of the Royal Society of London A* **357**, 2495

Candès, E. and Donoho, D. L.: 2002, *New Tight Frames of Curvelets and Optimal Representations of Objects with Smooth Singularities*, Technical report, Statistics, Stanford University

Candès, E. J.: 1999, Harmonic analysis of neural netwoks, *Applied and Computational Harmonic Analysis* **6**, 197

Candès, E. J. and Donoho, D. L.: 1999c, Curvelets – a surprisingly effective nonadaptive representation for objects with edges, in A. Cohen, C. Rabut, and L. Schumaker (eds.), *Curve and Surface Fitting: Saint-Malo 1999*, Vanderbilt University Press, Nashville, TN

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

Chen, S., Donoho, D., and Saunder, M.: 1998, Atomic decomposition by basis pursuit, *SIAM Journal on Scientific Computing* **20**, 33

Cichocki, A. and Amari, S.: 2002, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, John Wiley and Sons, New York

Cohen, A.: 2003, *Numerical Analysis of Wavelet Methods*, Elsevier

Coifman, R. and Donoho, D.: 1995, Translation invariant de-noising, in A. Antoniadis and G. Oppenheim (eds.), *Wavelets and Statistics*, pp 125–150, Springer-Verlag

Coifman, R., Meyer, Y., and Wickerhauser, M.: 1992, Wavelet analysis and signal processing, in M. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael (eds.), *Wavelets and Their Applications*, pp 153–178, Jones and Bartlett Publishers

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

Daubechies, I.: 1992, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics

Do, M. N. and Vetterli, M.: 2003a, The contourlet transform: an efficient directional multiresolution image representation, *The contourlet transform: an efficient directional multiresolution image representation*, submitted

Do, M. N. and Vetterli, M.: 2003b, Contourlets, in J. Stoeckler and G. V. Welland (eds.), *Beyond Wavelets*, Academic Press

Do, M. N. and Vetterli, M.: 2003c, The finite ridgelet transform for image representation, *IEEE Transactions on Image Processing* **12(1)**, 16

Donoho, D.: 1993, Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data, in A. M. Society (ed.), *Proceedings of Symposia in Applied Mathematics*, Vol. 47, pp 173–205

Donoho, D.: 2000, Nonlinear pyramid transforms based on median-interpolation, *SIAM J. Math. Anal.* **60**, 1137

Donoho, D. and Duncan, M.: 2000, Digital curvelet transform: strategy, implementation and experiments, in H. Szu, M. Vetterli, W. Campbell, and J. Buss (eds.), *Proc. Aerosense 2000, Wavelet Applications VII*, Vol. 4056, pp 12–29, SPIE

Donoho, D. and Elad, M.: 2003, Optimally sparse representation in general (non-orthogonal) dictionaries via ℓ^1 minimization, *Proc. Nat. Aca. Sci.* **100**, 2197

Donoho, D., Elad, M., and Temlyakov, V.: 2003, Stable recovery of sparse overcomplete representations in the presence of noise, *manuscript draft*

Donoho, D. and Flesia, A.: 2002, Digital Ridgelet Transform Based on True Ridge Functions, in J. Schmeidler and G. Welland (eds.), *Beyond Wavelets*, Academic Press

Donoho, D. and Huo, X.: 2001, Uncertainty principles and ideal atomic decomposition, *IEEE Trans. on Inf. Theory* **47(7)**, 2845

Donoho, D. and Johnstone, I.: 1994, Ideal spatial adaptation via wavelet shrinkage, *Biometrika* **81**, 425

Donoho, D. and Johnstone, I.: 1995, Adapting to unknown smoothness via wavelet shrinkage, *Journal of the American Statistical Association* **90**, 1200

Donoho, D. L.: 1997, *Fast Ridgelet Transforms in Dimension 2*, Technical report, Stanford University, Department of Statistics, Stanford CA 94305–4065

Donoho, D. L.: 1998, *Digital Ridgelet Transform via RectoPolar Coordinate Transform*, Technical report, Stanford University

Dutilleux, P.: 1987, An implementation of the "algorithme à trous" to compute the wavelet transform, in J. Combes, A. Grossmann, and P. Tchamitchian (eds.), *Wavelets: Time-Frequency Methods and Phase-Space*, Springer New York

Elad, M. and Bruckstein, A.: 2001, On sparse representations, in *Proc. of IEEE-International Conference on Image Processing (ICIP)*, Thessaloniki, Greece

Elad, M. and Bruckstein, A.: 2002, A generalized uncertainty principle and sparse representation in pairs of $\mathbf{r}^n$ bases, *IEEE Trans. on Inf. Theory* **48**, 2558

Gilbert, A.C. Muthukrishnan, S. and Strauss, M.: 2003, Approximation of functions over redundant dictionaries using coherence, in *14th Ann. ACM-SIAM Symposium Discrete Algorithms*

Golub, G. and Van-Loan, C.: 1996, *Matrix Computations*, Johns Hopkins Pub., third edition edition

Gribonval, R. and Nielsen, M.: 2003, Sparse representations in unions of bases, *IEEE Trans. on Inf. Theory*

Haykin, S. S.: 2001, *Unsupervised Adaptive Filtering, Volume 1: Blind Source Separation*, John Wiley and Sons, New York

Holschneider, M., Kronland-Martinet, R., Morlet, J., and Tchamitchian, P.: 1989, A real-time algorithm for signal analysis with the help of the wavelet transform, in *Wavelets: Time-Frequency Methods and Phase-Space*, pp 286–297, Springer-Verlag

Hyvarinen, A., Karhunen, J., and Oja, E.: 2001, *Independent Component Analysis*, John Wiley and Sons, New York

Jalobeanu, A., Blanc-Féraud, L., and Zerubia, J.: 2000, *Satellite image deconvolution using complex wavelet packets*, Technical Report 3955, INRIA, Sophia Antipolis, France

Jalobeanu, A., Blanc-Féraud, L., and Zerubia, J.: 2003, Satellite image deblurring using complex wavelet packets, *IJCV* 51(3)

Kalifa, J., Mallat, S., and Rougé, B.: 2003, Deconvolution by thresholding in

mirror wavelet bases, *IEEE Transactions on Image Processing* **12(4)**, 446

Karlovitz, L.: 1970, Construction of nearest points in the ℓ^p, p even and ℓ^1 norms, *Journal of Approximation Theory* **3**, 123

Kingsbury, N.: 1998, The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement, in *European Signal Processing Conference*, pp 319–322

Kingsbury, N.: 1999, Shift invariant properties of the dual-tree complex wavelet transform, in *IEEE Conf. on Acoustics, Speech and Signal Processing*

Kisilev, P., Zibulevsky, M., and Zeevi, Y. Y.: 2001, Blind source separation using multinode sparse representation, in *Proc. of IEEE-International Conference on Image Processing (ICIP)*, Thessaloniki, Greece

Kreutz-Delgado, K. and Rao, B.: 1999, Sparse basis selection, ica, and majorization: towards a unified perspective, in *Proc. of IEEE-International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Phoenix, AZ, USA

Mallat, S.: 1998, *A Wavelet Tour of Signal Processing*, Academic Press

Mallat, S. and Hwang, W. L.: 1992, Singularity detection and processing with wavelets, *IEEE Transactions on Information Theory* **38**, 617

Mallat, S. and Zhang, Z.: 1993, Atomic decomposition by basis pursuit, *IEEE Transactions on Signal Processing* **41**, 3397

Mallat, S. and Zhong, S.: 1992, Characterization of signals from multiscale edges, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 710

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

Meyer, Y.: 2002, Oscillating patterns in image processing and non linear evolution equations, *University Lecture Series Volume 22, AMS*

Pennec, E. L. and Mallat, S.: 2000, Image processing with geometrical wavelets, in *International Conference on Image Processing*

Portilla, J., Strela, V., Wainwright, M., and Simoncelli, E. P.: 2003, Image denoising using a scale mixture of Gaussians in the wavelet domain, *IEEE Trans Image Processing* 12(12), In press

Rudin, L., Osher, S., and Fatemi, E.: 1992, Nonlinear total variation noise removal algorithm, *Physica D* **60**, 259

Simoncelli, E., Freeman, W., Adelson, E., and Heeger, D.: 1992a, Shiftable multi-scale transforms, *IEEE Transactions on Information Theory* **38(2)**, 587

Simoncelli, E., Freeman, W., Adelson, E., and Heeger, D.: 1992b, Shiftable multi-scale transforms [or "what's wrong with orthonormal wavelets"], *IEEE Trans. Information Theory*

Simoncelli, E. P.: 1999, Bayesian denoising of visual images in the wavelet domain, in P. Müller and B. Vidakovic (eds.), *Bayesian Inference in Wavelet Based Models*, Chapt. 18, pp 291–308, Springer-Verlag, New York, Lecture

Notes in Statistics, vol. 141

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

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

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

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

Starck, J.-L., Elad, M., and Donoho, D. L.: 2003a, Image decomposition: Separation of texture from piece-wise smooth content, in A. Laine, M. Unser, and A. Aldroubi (eds.), *SPIE conference on Signal and Image Processing: Wavelet Applications in Signal and Image Processing X, San Diego, 4-8 August*, SPIE

Starck, J.-L. and Murtagh, F.: 2002, *Astronomical Image and Data Analysis*, Springer-Verlag

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

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

Starck, J.-L., Nguyen, M., and Murtagh, F.: 2003b, Wavelets and curvelets for image deconvolution: a combined approach, *Signal Processing* **83(10)**, 2279

Strang, G. and Nguyen, T.: 1996, *Wavelet and Filter Banks*, Wellesley-Cambridge Press

Tropp, J. A.: 2003, Grid is good: Algorithmic results for sparse approximation, *submitted to IEEE Trans. on Inf. Theory*

Vese, L. and Osher, S.: 2003, Modeling textures with total variation minimization and oscillating patterns in image processing, *Journal of Scientific Computing* **19(1-3)**, 553, in press

Vetterli, M.: 2001, Wavelets, approximation, and compression, *IEEE Signal Processing Magazine* **18(5)**, 59

Zibulevsky, M. and Pearlmutter, B.: 2001, Blind source separation by sparse decomposition in a signal dictionary, *Neural-Computation* **13(4)**, 863