Curvelets and Ridgelets

Jalal M. Fadili, Jean-Luc Starck

Curvelets and Ridgelets

M.J. Fadili [*], J.-L. Starck [†]

June 24, 2010

Contents

Glossary

WT1D
The one-dimensional Wavelet Transform as defined in [1]. See also [2] in this volume.

WT2D
The two-dimensional Wavelet Transform.

Discrete Ridgelet Trasnform (DRT)
The discrete implementation of the continuous Ridgelet transform.

Fast Slant Stack (FSS)
An algebraically exact Radon transform of data on a Cartesian grid.

First Generation Discrete Curvelet Transform (DCTG1)
The discrete curvelet transform constructed based on the discrete ridgelet transform.

Second Generation Discrete Curvelet Transform (DCTG2)
The discrete curvelet transform constructed based on appropriate bandpass filtering in the Fourier domain.

[*] J. Fadili is with the GREYC CNRS UMR 6072, Image Processing Group, ENSICAEN 14050, Caen Cedex, France

[†] J.-L. Starck is with the CEA-Saclay, DAPNIA/SEDI-SAP, Service d'Astrophysique, F-91191 Gif sur Yvette, France

Anisotropic elements
By anistropic, we mean basis elements with elongated effective support; i.e. length > width.

Parabolic scaling law
A basis element obeys the parabolic scaling law if its effective support is such that width $\approx$ length^2.

1 Definition of the Subject and its Importance

Despite the fact that wavelets have had a wide impact in image processing, they fail to efficiently represent objects with highly anisotropic elements such as lines or curvilinear structures (e.g. edges). The reason is that wavelets are non-geometrical and do not exploit the regularity of the edge curve.

The Ridgelet and the Curvelet [3, 4] transforms were developed as an answer to the weakness of the separable wavelet transform in sparsely representing what appears to be simple building atoms in an image, that is lines, curves and edges. Curvelets and ridgelets take the form of basis elements which exhibit high directional sensitivity and are highly anisotropic [5, 6, 7, 8]. These very recent geometric image representations are built upon ideas of multiscale analysis and geometry. They have had an important success in a wide range of image processing applications including denoising [8, 9, 10], deconvolution [11, 12], contrast enhancement [13], texture analysis [14, 15], detection [16], watermarking [17], component separation [18], inpainting [19, 20] or blind source separation [21, 22]. Curvelets have also proven useful in diverse fields beyond the traditional image processing application. Let's cite for example seismic imaging [10, 23, 24], astronomical imaging [25, 26, 27], scientific computing and analysis of partial differential equations [28, 29]. Another reason for the success of ridgelets and curvelets is the availability of fast transform algorithms which are available in non-commercial software packages following the philosophy of reproducible research, see [30, 31].

2 Introduction

2.1 Sparse Geometrical Image Representation

Multiscale methods have become very popular, especially with the development of the wavelets in the last decade. Background texts on the wavelet transform include [32, 1, 33]. An overview of implementation and practical issues of the wavelet transform can also be found in [2], included in this volume.

Despite the success of the classical wavelet viewpoint, it was argued that the traditional wavelets present some strong limitations that question their effectiveness in higher-dimension than 1 [3, 4]. 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. Following this reasoning, new constructions have been proposed such as the ridgelets [3, 5] and the curvelets [4, 6, 7, 8]. Ridgelets and curvelets are special members of the family of multiscale orientation-selective transforms, which has recently led to a flurry of research activity in the field of computational and applied harmonic analysis. Many other constructions belonging to this family have been investigated in the literature, and go by the name contourlets [34], directionlets [35], bandlets [36, 37], grouplets [38], shearlets [39], dual-tree wavelets and wavelet packets [40, 41], etc.

Throughout this paper, the term 'sparsity' is used and intended in a weak sense. We are aware that practical images and signals may not be supported in a transform domain on a set of relatively

Figure 1: Few Ridgelets examples - The second to fourth graphs are obtained after simple geometric manipulations of the first ridgelet, namely rotation, rescaling, and shifting.

small size (sparse set). Instead, they may only be compressible (nearly sparse) in some transform domain. Hence, with a slight abuse of terminology, we will say that a representation is sparse for an image within a certain class, if it provides a compact description of such an image.

Notations

We work throughout in two dimensions with spatial variable $\mathbf{x} \in \mathbb{R}^2$ and $\boldsymbol{\nu}$ a continuous frequency-domain variable. Parentheses $(.,.)$ are used for continuous-domain function evaluations, and brackets $[.,.]$ for discrete-domain array indices. The hat $\hat{}$ notation will be used for the Fourier transform.

3 Ridgelets

3.1 The Continuous Ridgelet Transform

The two-dimensional continuous ridgelet transform in $\mathbb{R}^2$ can be defined as follows [42]. We pick a smooth univariate function $\psi : \mathbb{R} \to \mathbb{R}$ with sufficient decay and satisfying the admissibility condition

$$\int |\hat{\psi}(\boldsymbol{\nu})|^2/|\boldsymbol{\nu}|^2 \, d\boldsymbol{\nu} < \infty, \tag{1}$$

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

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

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

A ridgelet is constant along lines $x_1 \cos\theta + x_2 \sin\theta = const.$ Transverse to these ridges it is a wavelet. Figure 1 depicts few examples of ridgelets. The second to fourth panels are obtained after simple geometric manipulations of the ridgelet (left panel), namely rotation, rescaling, and shifting.

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

$$\mathcal{R}_f(a, b, \theta) := \langle f, \psi_{a,b,\theta} \rangle = \int_{\mathbb{R}^2} f(\mathbf{x}) \overline{\psi}_{a,b,\theta}(\mathbf{x}) d\mathbf{x}.$$

We have the exact reconstruction formula

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

valid almost everywhere for functions which are both integrable and square integrable. This formula is stable and one can prove a Parseval relation [3].

Ridgelet analysis may be constructed as wavelet analysis in the Radon domain. The rationale behind this is that the Radon transform translates singularities along lines into point singularities, for which the wavelet transform is known to provide a sparse representation. Recall that the Radon transform of an object f is the collection of line integrals indexed by $(\theta, t) \in [0, 2\pi) \times \mathbb{R}$ given by

$$Rf(\theta, t) = \int_{\mathbb{R}^2} f(x_1, x_2) \delta(x_1 \cos\theta + x_2 \sin\theta - t) \, dx_1 dx_2, \tag{4}$$

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 (DRTs) have been proposed, and we will describe three of them in this section, based on different implementations of the Radon transform.

3.1.1 The RectoPolar Ridgelet transform

A fast implementation of the Radon transform can be proposed in the Fourier domain, based on the projection-slice-theorem. First the 2D FFT of the given image is computed. 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 2. 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 [8].

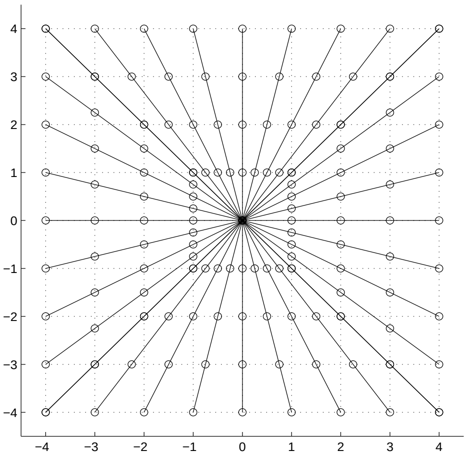

Figure 2: Illustration of the pseudo-polar grid in the frequency domain for an n by n image ($n = 8$).

3.1.2 One-dimensional Wavelet Transform

To complete the ridgelet transform, we must take a one-dimensional wavelet transform (WT1D) along the radial variable in Radon space. We now discuss the choice of the digital WT1D.

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 [43, 44, 8]. In [8], a specific overcomplete wavelet transform [45, 33] 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 Fourier transform of the scaling function 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 [33] for details.

This WT1D transform enjoys the following useful properties:

- 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.

Figure 3: Discrete ridgelet transform flowchart. 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.

- Each sub-band is sampled above the Nyquist rate, hence, avoiding aliasing –a phenomenon typically encountered by critically sampled orthogonal wavelet transforms [46].
- 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, this implementation would be useful to the practitioner whose focuses on data analysis, for which it is well-known that over-completeness through (almost) translation-invariance can provide substantial advantages.

Assembling all above ingredients together gives the flowchart of the discrete ridgelet transform (DRT) depicted in Figure 3. The DRT 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.

Figure 4: The backprojection of a ridgelet coefficient by the FFT-based ridgelet transform (left), and by the OFRT (right).

3.2 The Orthonormal Finite Ridgelet Transform

The orthonormal finite ridgelet transform (OFRT) has been proposed [47] for image compression and filtering. This transform is based on the finite Radon transform [48] 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 an awkward definition of a line. In fact, a line in the OFRT is defined algebraically rather that geometrically, and so the points on a 'line' can be arbitrarily and randomly spread out in the spatial domain. Figure 4 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.

Because of this specific definition of a line, the thresholding of the OFRT coefficients produces strong artifacts. Figure 5 shows a part of the original image `Boat`, and its reconstruction after hard thresholding the OFRT of the noise-free `Boat`. The resulting image is not smoothed as one would expect, but rather 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 spatial partitioning when denoising the data, and a prime number block size can be used. The OFRT is interesting from the conceptual point of view, but still requires work before it can be used for real applications such as denoising.

3.3 The Fast Slant Stack Ridgelet Transform

The Fast Slant Stack (FSS) [49] is a Radon transform of data on a Cartesian grid, which is algebraically exact and geometrically more accurate and faithful 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 6). 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

Figure 5: Part of original noise-free `Boat` image (left), and reconstruction after hard thresholding its OFRT coefficients (right).

Figure 6: Backprojection of a point at four different locations in the Radon space using the FSS algorithm.

shearing is performed one column at a time (resp. one line at a time) by using the 1D FFT. Figure 7 shows an example of the image shearing step with two different angles ($5\frac{\pi}{4}$ and $-\frac{\pi}{4}$). A DRT based on the FSS transform has been proposed in [50]. The connection between the FSS and the Linogram has been investigated in [49]. A FSS algorithm is also proposed in [49], based on the 2D Fast Pseudo-polar Fourier transform which evaluates the 2-D Fourier transform on a non-Cartesian (pseudo-polar) grid, operating in $O(n^2 \log n)$ flops.

Figure 8 left exemplifies a ridgelet in the spatial domain obtained from the DRT based on FSS implementation. Its Fourier transform is shown on Figure 8 right superimposed on the DRT frequency tiling [50]. The Fourier transform of the discrete ridgelet lives in an angular wedge. More precisely, the Fourier transform of a discrete ridgelet at scale j lives within a dyadic square of size $\sim 2^j$.

3.4 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 [5]. 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 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.

Figure 7: Slant Stack Transform of an image.

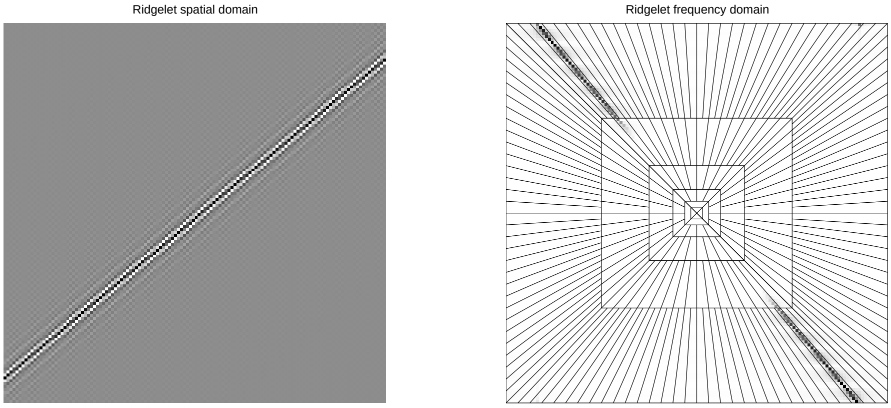

Figure 8: (a) Example of a ridgelet obtained by the Fast Slant Stack implementation. (b) Its FFT superimposed on the DRT frequency tiling.

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 by a spatial window w (analysis) in such a way that the co-addition of all blocks reproduce exactly the original pixel value (synthesis).

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

Experiments have shown that the second approach leads to better results especially for restoration problems, see [8] for details. We calculate a pixel value, $f[i_1, i_2]$ from its four corresponding block values of half-size $m = b/2$, namely, $B_1[k_1, l_1]$, $B_2[k_2, l_1]$, $B_3[k_1, l_2]$ and $B_4[k_2, l_2]$ with $k_1, l_1 > b/2$ and $k_2 = k_1 - m, l_2 = l_1 - m$, in the following way:

$$\begin{aligned} f_1 &= w(k_2/m)B_1[k_1, l_1] + w(1 - k_2/m)B_2[k_2, l_1] \\ f_2 &= w(k_2/m]B_3[k_1, l_2] + w(1 - k_2/m)B_4[k_2, l_2] \\ f[i_1, i_2] &= w(l_2/m)f_1 + w(1 - l_2/m)f_2. \end{aligned} \tag{5}$$

where $w(x) = \cos^2(\pi x/2)$ is the window. 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 $w(x) + w(1 - x) = 1$.

3.5 Sparse Representation by Ridgelets

The continuous ridgelet transform provides sparse representation of both smooth functions (in the Sobolev space W_2^2) and of perfectly straight lines [51, 52]. We have just seen that there are also various DRTs, i.e. expansions with countable discrete collection of generating elements, which correspond either to frames or orthobases. It has been shown for these schemes that the DRT achieves near-optimal M-term approximation - that is the non-linear approximation of f using the M highest ridgelet coefficients in magnitude - to smooth images with discontinuities along straight lines [3, 52]. In summary, ridgelets provide sparse presentation for piecewise smooth images away from global straight edges.

4 Curvelets

4.1 The First Generation Curvelet Transform

In image processing, edges are curved rather than straight lines and ridgelets are not able to efficiently represent such images. However, one can still deploy the ridgelet machinery in a localized way, at fine scales, where curved edges are almost straight lines (see Figure 9). This is the idea underlying the first generation curvelets (termed here CurveletG1) [7].

4.1.1 First Generation Curvelets Construction

The CurveletG1 transform [7, 6, 8] 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 as illustrated on Figure 9. The block size can be changed at each scale level. Roughly speaking, different levels of the multiscale ridgelet

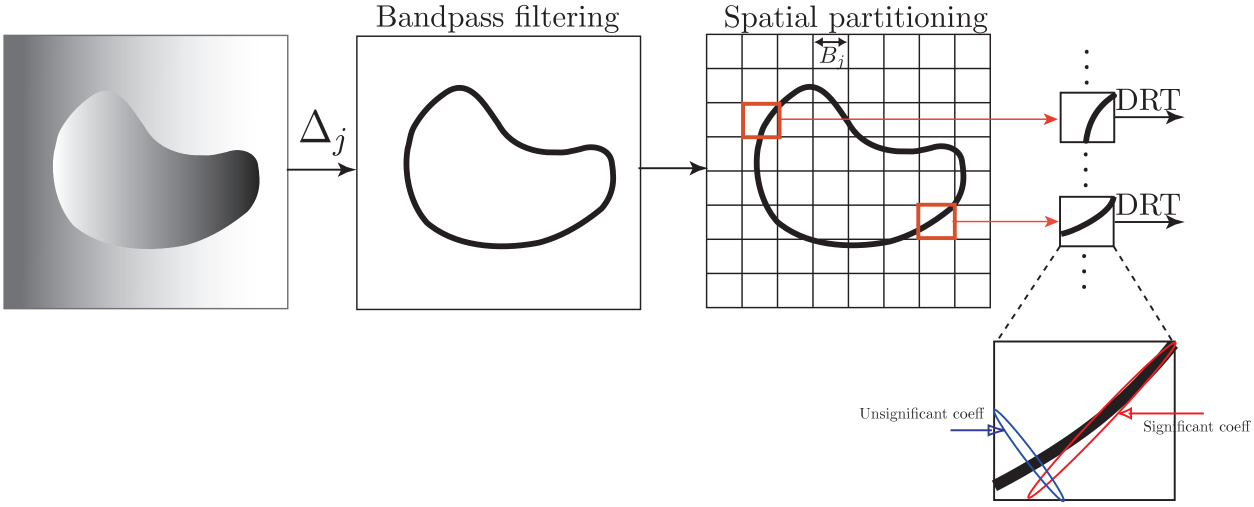

Figure 9: Local ridgelet transform on bandpass filtered image. At fine scales, curved edges are almost straight lines.

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 approximately the parabolic scaling law width $\approx$ length^2.

The First Generation Discrete Curvelet Transform (DCTG1) of a continuum function $f(\mathbf{x})$ makes use of a dyadic sequence of scales, and a bank of filters with the property that the bandpass filter Δ_j is concentrated near the frequencies $[2^{2j}, 2^{2j+2}]$, i.e.

$$\Delta_j(f) = \Psi_{2j} * f, \quad \widehat{\Psi}_{2j}(\boldsymbol{\nu}) = \widehat{\Psi}(2^{-2j}\boldsymbol{\nu}).$$

In wavelet theory, one uses a decomposition into dyadic sub-bands $[2^j, 2^{j+1}]$. In contrast, the sub-bands used in the discrete curvelet transform of continuum functions have the nonstandard form $[2^{2j}, 2^{2j+2}]$. This is nonstandard feature of the DCTG1 well worth remembering (this is where the approximate parabolic scaling law comes into play).

The DCTG1 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 $\sim 2^{-j}$).

- *Ridgelet Analysis.* Each square is analyzed via the DRT.

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

4.1.2 Digital Implementation

It seems that the isotropic “à trous” wavelet transform [33, 2] is especially well-adapted to the needs of the digital curvelet transform. The algorithm decomposes an n by n image $f[i_1, i_2]$ as a superposition of the form

$$f[i_1, i_2] = c_J[i_1, i_2] + \sum_{j=1}^{J} w_j[i_1, i_2],$$

where c_J is a coarse or smooth version of the original image f and w_j represents 'the details of f' at scale 2^{-j}. Thus, the algorithm outputs $J + 1$ sub-band arrays of size $n \times n$.

A sketch of the DCTG1 algorithm is as follows:

Algorithm 1 DCTG1.

```
Require: Input $n \times n$ image $f[i_1, i_2]$, type of DRT (see above).
Apply the à trous isotropic WT2D with $J$ scales,
Set $B_1 = B_{\min}$,
for $j = 1, \ldots, J$ do
    Partition the sub-band $w_j$ with a block size $B_j$ and apply the DRT to each block,
    if $j$ modulo $2 = 1$ then
        $B_{j+1} = 2B_j$,
    else
        $B_{j+1} = B_j$.
    end if
end for
```

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 left intact. In the results shown in this paper, we used the default value $B_{\min} = 16$ pixels in our implementation. Figure 10 gives an overview of the organization of the DCTG1 algorithm.

This implementation of the DCTG1 is also redundant. The redundancy factor is equal to $16J+1$ whenever J scales are employed. The DCTG1 algorithm enjoys exact reconstruction and stability, as each step of the analysis (decomposition) algorithm is itself invertible. One can show that the computational complexity of the DCTG1 algorithm we described here based on the DRT of Fig.3 is $O(n^2(\log n)^2)$ for an $n \times n$ image. Figure 11 shows a few curvelets at different scales, orientations and locations.

4.1.3 Sparse Representation by First Generation Curvelets

The CurveletG1 elements can form either a frame or a tight frame for $L_2(\mathbb{R}^2)$ [4], depending on the WT2D used and the DRT implementation (rectopolar or FSS Radon transform). The frame elements are anisotropic by construction and become successively more anisotropic at progressively higher scales. These curvelets also exhibit directional sensitivity and display oscillatory components across the 'ridge'. A central motivation leading to the curvelet construction was the problem of non-adaptively representing piecewise smooth (e.g. C^2) images f which have discontinuity along a C^2 curve. Such a model is the so-called cartoon model of (non-textured) images. With the CurveletG1 tight frame construction, it was shown in [4] that for such f, the M-term non-linear approximations f_M of f obey, for each $\kappa > 0$,

$$\|f - f_M\|_2 \leq C_\kappa M^{-2+\kappa}, \; M \to +\infty.$$

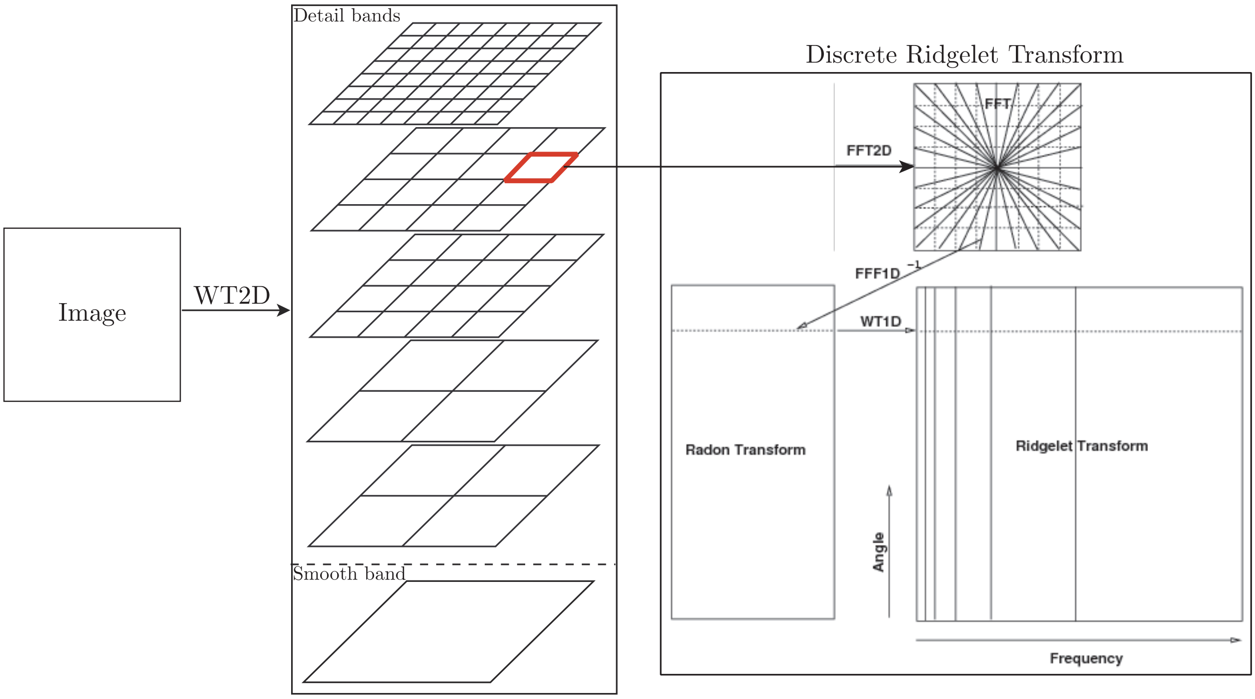

Figure 10: First Generation Discrete Curvelet Transform (DCTG1) flowchart. The figure illustrates the decomposition of the original image into sub-bands followed by the spatial partitioning of each sub-band. The ridgelet transform is then applied to each block.

Figure 11: A few first generation curvelets.

The M-term approximations in the CurveletG1 are almost rate optimal, much better than M-term Fourier or wavelet approximations for such images, see [1].

4.2 The Second Generation Curvelet Transform

Despite these interesting properties, the CurveletG1 construction presents some drawbacks. First, the construction involves a complicated seven-index structure among which we have parameters for scale, location and orientation. In addition, the parabolic scaling ratio width $\approx$ length^2 is not completely true (see subsection 4.1.1). In fact, CurveletG1 assumes a wide range of aspect ratios. These facts make mathematical and quantitative analysis especially delicate. Second, the spatial partitioning of the CurveletG1 transform uses overlapping windows to avoid blocking effects. This leads to an increase of the redundancy of the DCTG1. The computational cost of the DCTG1 algorithm may also be a limitation for large-scale data, especially if the FSS-based DRT implementation is used.

In contrast, the second generation curvelets (CurveletG2) [53, 54] exhibit a much simpler and natural indexing structure with three parameters: scale, orientation (angle) and location, hence simplifying mathematical analysis. The CurveletG2 transform also implements a tight frame expansion [53] and has a much lower redundancy. Unlike the DCTG1, the discrete CurveletG2 implementation will not use ridgelets yielding a faster algorithm [53, 54].

4.2.1 Second Generation Curvelets Construction

Continuous coronization The second generation curvelets are defined at scale 2^{-j}, orientation θ_l and position $\mathbf{x}_{\mathbf{k}}^{j,l} = R_{\theta_l}^{-1}(2^{-j}k_1, 2^{-j/2}k_2)$ by translation and rotation of a mother curvelet φ_j as

$$\varphi_{j,l,\mathbf{k}}(\mathbf{x}) = \varphi_j(R_{\theta_l}(\mathbf{x} - \mathbf{x}_{\mathbf{k}}^{j,l})), \tag{6}$$

where R_{θ_l} is the rotation by θ_l radians. θ_l is the equi-spaced sequence of rotation angles $\theta_l = 2\pi 2^{-\lfloor j/2 \rfloor} l$, with integer l such that $0 \leq \theta_l \leq 2\pi$ (note that the number of orientations varies as $1/\sqrt{\text{scale}}$). $\mathbf{k} = (k_1, k_2) \in \mathbb{Z}^2$ is the sequence of translation parameters. The waveform φ_j is defined by means of its Fourier transform $\hat{\varphi}_j(\boldsymbol{\nu})$, written in polar coordinates in the Fourier domain

$$\hat{\varphi}_j(r, \theta) = 2^{-3j/4} \hat{w}(2^{-j} r) \hat{v} \left(\frac{2^{\lfloor j/2 \rfloor} \theta}{2\pi} \right). \tag{7}$$

The support of $\hat{\varphi}_j$ is a polar *parabolic wedge* defined by the support of $\hat{w}$ and $\hat{v}$, the radial and angular windows (both smooth, nonnegative and real-valued), applied with scale-dependent window widths in each direction. $\hat{w}$ and $\hat{v}$ must also satisfy the partition of unity property [54]. See the frequency tiling in Figure 12(a).

In continuous frequency $\boldsymbol{\nu}$, the CurveletG2 coefficients of data $f(\mathbf{x})$ are defined as the inner product

$$c_{j,l,\mathbf{k}} := \langle f, \varphi_{j,l,\mathbf{k}} \rangle = \int_{\mathbb{R}^2} \hat{f}(\boldsymbol{\nu}) \hat{\varphi}_j(R_{\theta_l} \boldsymbol{\nu}) e^{i \mathbf{x}_{\mathbf{k}}^{j,l} \cdot \boldsymbol{\nu}} d\boldsymbol{\nu}. \tag{8}$$

This construction implies a few properties: (i) the CurveletG2 defines a tight frame of $L_2(\mathbb{R}^2)$, (ii) the effective length and width of these curvelets obey the parabolic scaling relation $(2^{-j} = \text{width}) = (\text{length} = 2^{-j/2})^2$, (iii) the curvelets exhibit an oscillating behavior in the direction perpendicular to their orientation. Curvelets as just constructed are complex-valued. It is easy to obtain real-valued curvelets by working on the symmetrized version $\hat{\varphi}_j(r, \theta) + \hat{\varphi}_j(r, \theta + \pi)$.

Discrete coronization The discrete transform takes as input data defined on a Cartesian grid and outputs a collection of coefficients. The continuous-space definition of the CurveletG2 uses coronae and rotations that are not especially adapted to Cartesian arrays. It is then convenient to replace these concepts by their Cartesian counterparts. That is concentric squares (instead of concentric circles) and shears (instead of rotations), see Figure 12(c).

The Cartesian equivalent to the radial window $\hat{w}_j(\boldsymbol{\nu}) = \hat{w}(2^{-j}\boldsymbol{\nu})$ would be a bandpass frequency-localized window which can be derived from the difference of separable low-pass windows $H_j(\boldsymbol{\nu}) = \hat{h}(2^{-j}\nu_1)\hat{h}(2^{-j}\nu_2)$ (h is a 1D low-pass filter):

$$\hat{w}_j(\boldsymbol{\nu}) = \sqrt{H^2_{j+1}(\boldsymbol{\nu}) - H^2_j(\boldsymbol{\nu})}, \forall j \geq 0, \text{ and }, \hat{w}_0(\boldsymbol{\nu}) = \hat{h}(\nu_1)\hat{h}(\nu_2)$$

Another possible choice is to select these windows inspired by the construction of Meyer wavelets [55, 53]. See [54] for more details about the construction of the Cartesian $\hat{w}_j$'s.

Let's now examine the angular localization. Each Cartesian coronae has four quadrants: East, North, West and South. Each quadrant is separated into $2^{\lfloor j/2 \rfloor}$ orientations (wedges) with the same areas. Take for example the East quadrant ($-\pi/4 \leq \theta_l < \pi/4$). For the West quadrant, we would proceed by symmetry around the origin, and for the North and South quadrant by exchanging the roles of ν_1 and ν_2. Define the angular window for the l-th direction as

$$\hat{v}_{j,l}(\boldsymbol{\nu}) = \hat{v}\left(2^{\lfloor j/2 \rfloor}\frac{\nu_2 - \nu_1 \tan\theta_l}{\nu_1}\right), \tag{9}$$

with the sequence of equi-spaced slopes (and not angles) $\tan\theta_l = 2^{-\lfloor j/2 \rfloor} l$, for $l = -2^{\lfloor j/2 \rfloor}, \ldots, 2^{\lfloor j/2 \rfloor} - 1$. We can now define the window which is the Cartesian analog of $\hat{\varphi}_j$ above,

$$\hat{u}_{j,l}(\boldsymbol{\nu}) = \hat{w}_j(\boldsymbol{\nu})\hat{v}_{j,l}(\boldsymbol{\nu}) = \hat{w}_j(\boldsymbol{\nu})\hat{v}_{j,0}(S_{\theta_l}\boldsymbol{\nu}), \tag{10}$$

where S_{θ_l} is the shear matrix. From this definition, it can be seen that $\hat{u}_{j,l}$ is supported near the trapezoidal wedge $\{\boldsymbol{\nu} = (\nu_1, \nu_2) | 2^j \leq \nu_1 \leq 2^{j+1}, -2^{-j/2} \leq \nu_2/\nu_1 - \tan\theta_l \leq 2^{-j/2}\}$. The collection of $\hat{u}_{j,l}(\boldsymbol{\nu})$ gives rise to the frequency tiling shown in Figure 12(c). From $\hat{u}_{j,l}(\boldsymbol{\nu})$, the digital CurveletG2 construction suggests Cartesian curvelets that are translated and sheared versions of a mother Cartesian curvelet $\hat{\varphi}^D_j(\boldsymbol{\nu}) = \hat{u}_{j,0}(\boldsymbol{\nu})$, $\varphi^D_{j,l,\mathbf{k}}(\mathbf{x}) = 2^{3j/4}\varphi^D_j\left(S^T_{\theta_l}\mathbf{x} - \mathbf{m}\right)$ where $\mathbf{m} = (k_1 2^{-j}, k_2 2^{-j/2})$.

4.2.2 Digital Implementation

The goal here is to find a digital implementation of the Second Generation Discrete Curvelet Transform (DCTG2), whose coefficients are now given by

$$c_{j,l,\mathbf{k}} := \left\langle f, \varphi^D_{j,l,\mathbf{k}} \right\rangle = \int_{\mathbb{R}^2} \hat{f}(\boldsymbol{\nu})\hat{\varphi}^D_j(S^{-1}_{\theta_l}\boldsymbol{\nu})e^{iS^{-T}_{\theta_l}\mathbf{m}\cdot\boldsymbol{\nu}}d\boldsymbol{\nu}. \tag{11}$$

To evaluate this formula with discrete data, one may think of (i) using the 2D FFT to get $\hat{f}$, (ii) form the windowed frequency data $\hat{f}\hat{u}_{j,l}$ and (iii) apply the the inverse Fourier transform. But this necessitates to evaluate the FFT at the sheared grid $S^{-T}_{\theta_l}\mathbf{m}$, for which the classical FFT algorithm is not valid. Two implementations were then proposed [54], essentially differing in their way of handling the grid:

- A tilted grid mostly aligned with the axes of $\hat{u}_{j,l}(\boldsymbol{\nu})$ which leads to the Unequi-Spaced FFT (USFFT)-based DCTG2. This implementation uses a nonstandard interpolation. Furthermore, the inverse transform uses conjugate gradients iteration to invert the interpolation step. This will have the drawback of a higher computational burden compared to the wrapping-based implementation that we will discuss hereafter. We will not elaborate more about the USFFT implementation as we never use it in practice. The interested reader may refer to [54] for further details and analysis.

- A grid aligned with the input Cartesian grid which leads to the wrapping-based DCTG2.

The wrapping-based DCTG2 makes a simpler choice of the spatial grid to translate the curvelets. The curvelet coefficients are essentially the same as in (11), except that $S_{\theta_l}^{-T}\mathbf{m}$ is replaced by $\mathbf{m}$ with values on a rectangular grid. But again, a difficulty rises because the window $\hat{u}_{j,l}$ does not fit in a rectangle of size $2^j \times 2^{j/2}$ to which an inverse FFT could be applied. The *wrapping* trick consists in periodizing the windowed frequency data $\hat{f}\hat{u}_{j,l}$, and reindexing the samples array by wrapping around a $\sim 2^j \times 2^{j/2}$ rectangle centered at the origin, see Figure 12(d) to get a gist of the wrapping idea.

The wrapping-based DCTG2 algorithm can be summarized as follows:

Algorithm 2 DCTG2 via wrapping.

```
Require: Input $n \times n$ image $f[i_1, i_2]$, coarsest decomposition scale, curvelets or wavelets at the finest scale.
Apply the 2D FFT and obtain Fourier samples $\hat{f}[i_1, i_2]$.
for each scale $j$ and angle $l$ do
    Form the product $\hat{f}[i_1, i_2]\hat{u}_{j,l}[i_1, i_2]$.
    Wrap this product around the origin.
    Apply the inverse 2D FFT to the wrapped data to get discrete DCTG2 coefficients.
end for
```

The DCTG2 implementation can assign either wavelets or curvelets at the finest scale. In the CurveLab toolbox [31], the default choice is set to wavelets at the finest scale, but this can be easily modified directly in the code.

We would like to apologize to the expert reader as many technical details are (deliberately) missing here on the CurveletG2 construction. For instance, low-pass coarse component, window overlapping, windows over junctions between quadrants. This paper is intended to give an overview of these recent multi-scale transforms, and the genuinely interested reader may refer to the original papers of Candès, Donoho, Starck and co-workers for further details (see bibliography).

The computational complexity of the wrapping-based DCTG2 analysis and reconstruction algorithms is that of the FFT $O(n^2 \log n)$, and in practice, the computation time is that of 6 to 10 2D FFTs [54]. This is a faster algorithm compared to the DCTG1. The DCTG2 fast algorithm has participated to make the use of the curvelet transform more attractive in many applicative fields (see Sections 5 and 5 for some of them). The DCTG2, as it is implemented in the CurveLab toolbox [31], has reasonable redundancy, at most ~ 7.8 (much higher in 3D) if curvelets are used at the finest scale. This redundancy can even be reduced down to 4 (and 8 in 3D) if we replace in this implementation the Meyer wavelet construction, which introduces a redundancy factor of 4, by another wavelet pyramidal construction, similar to the one presented in Section 3.1.2 which has a redundancy less than 2 in any dimension. Our experiments have shown that this modification does not modify the results in denoising experiments. DCTG2 redundancy is anyway much smaller than the DCTG1 one which is $16J + 1$. As stated earlier, the DCTG2 coefficients are complex-valued, but a real-valued DCTG2 with the same redundancy factor can be easily obtained by properly combining coefficients at orientations θ_l and $\theta_l + \pi$.

The DCTG2 can be extended to higher dimensions [56]. In the same vein as wavelets on the interval [1], the DCGT2 has been recently adapted to handle image boundaries by mirror extension instead of periodization [57]. The latter modification can have immediate implications in image processing applications where the contrast difference at opposite image boundaries may be an issue (see e.g. the denoising experiment discussion reported in Section 5).

We would like to make a connection with other multiscale directional transforms directly likned to curvelets. The contourlets tigth frame of Do and Vetterli [34] implements the CurveletG2 idea directly on a discrete grid using a perfect reconstruction filter bank procedure. In [58], the authors proposed a modification of the contourlets with a directional filter bank that provides a frequency partitioning which is close to the curvelets but with no redundancy. Durand in [59] recently introduced families of non-adaptive directional wavelets with various frequency tilings, including that of curvelets. Such families are non-redundant and form orthonormal bases for $L_2(\mathbb{R}^2)$, and have an implementation derived from a single nonseparable filter bank structure with nonuniform sampling.

4.2.3 Sparse Representation by Second Generation Curvelets

It has been shown by Candès and Donoho [53] that with the CurveletG2 tight frame construction, the M-term non-linear approximation error of C^2 images except at discontinuities along C^2 curves obey

$$\|f - f_M\|_2 \leq C M^{-2} (\log M)^3 .$$

This is an asymptotically optimal convergence rate (up to the $(\log M)^3$ factor), and holds uniformly over the C^2–C^2 class of functions. This is a remarkable result since the CurveletG2 representation is non-adaptative. However, the simplicity due to the non-adaptivity of curvelets has a cost: curvelet approximations loose their near optimal properties when the image is composed of edges which are not exactly C^2. Additionally, if the edges are C^α-regular with $\alpha > 2$, then the curvelets convergence rate exponent remain 2. Other adaptive geometric representations such as bandlets are specifically designed to reach the optimal decay rate $O(M^{-\alpha})$ [36, 37].

5 Stylized applications

5.1 Denoising

5.1.1 Elongated feature recovery

The ability of ridgelets to sparsely represent piecewise smooth images away from discontinuities along lines has an immediate implication on statistical estimation. Consider a piecewise smooth image f away from line singularities embedded in an additive white noise of standard deviation σ. The ridgelet-based thresholding estimator is nearly optimal for recovering such functions, with a mean-square error (MSE) decay rate almost as good as the minimax rate [60].

To illustrate these theoretical facts, we simulate a vertical band embedded in white noise with large σ. Figure 14 (top left) represents such a noisy image. The parameters are as follows: the pixel width of the band is 20 and the signal-to-noise ratio (SNR) is set to 0.1. Note that it is not

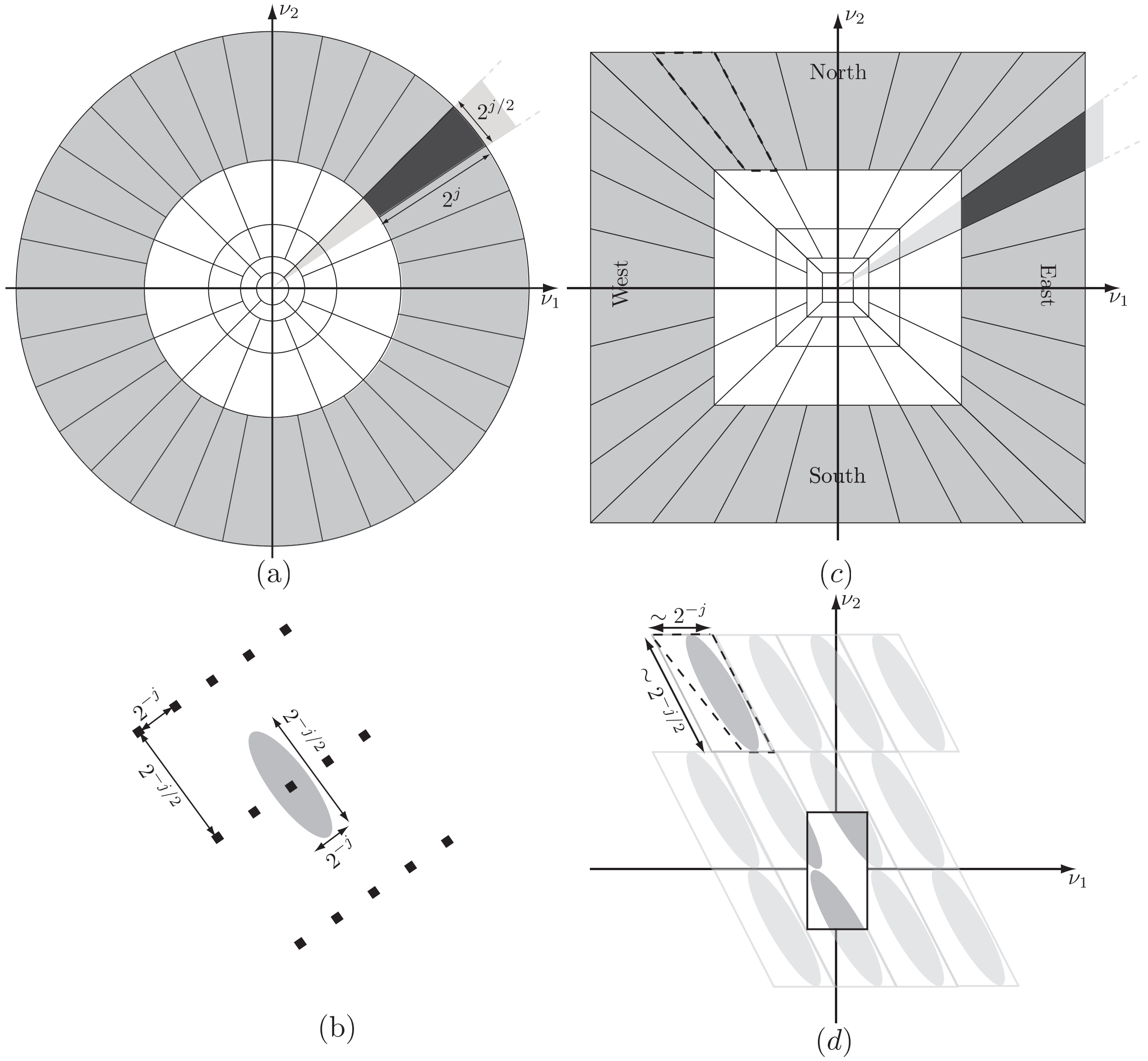

Figure 12: (a) Continuous curvelet frequency tiling. The dark gray area represents a wedge obtained as the product of the radial window (annulus shown in gray) and the angular window (light gray). (b) The Cartesian grid in space associated to the construction in (a) whose spacing also obeys the parabolic scaling by duality. (c) Discrete curvelet frequency tiling. The window $\hat{u}_{j,l}$ isolates the frequency near trapezoidal wedge such as the one shown in dark gray. (d) The wrapping transformation. The dashed line shows the same trapezoidal wedge as in (b). The parallelogram contains this wedge and hence the support of the curvelet. After periodization, the wrapped Fourier samples can be collected in the rectangle centered at the origin.

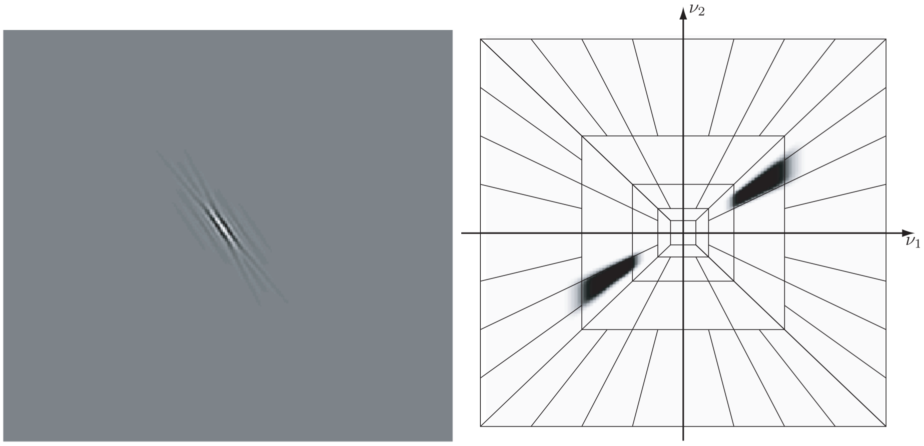

Figure 13: An example of second generation real curvelet. Left: curvelet in spatial domain. Right: its Fourier transform.

Figure 14: Original image containing a vertical band embedded in white noise with relatively large amplitude (left). Denoised image using the undecimated wavelet transform (middle). Denoised image using the DRT based on the rectopolar Radon transform (right).

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.

5.1.2 Curve recovery

Consider now the problem of recovering a piecewise C^2 function f apart from a discontinuity along a C^2 edges. Again, a simple strategy based on thresholding curvelet tight frame coefficients yields an estimator that achieves a MSE almost of the order $O(\sigma^{4/3})$ uniformly over the $C^2 - C^2$ class of functions [61]. This is the optimal rate of convergence as the minimax rate for that class scales as $\sigma^{4/3}$ [61]. Comparatively, wavelet thresholding methods only achieves a MSE of order $O(\sigma)$ and no better. We also note that the statistical optimality of the curvelet thresholding extends to a large class of ill-posed linear inverse problems [61].

In the experiment of Figure 15, we have added a Gaussian noise to “War and Peace,” a drawing from Picasso which contains many curved features. Figure 15 bottom left and right shows respectively the restored images by the undecimated wavelet transform and the DCTG1. Curves are more sharply recovered with the DCTG1.

Figure 15: 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 DCTG1 (bottom right).

In a second experiment, we compared the denoising performance of several digital implementations of the curvelet transform; namely the DCTG1 with the rectopolar DRT, the DCTG1 with the FSS-based DRT and the wrapping-based DCTG2. The results are shown in Figure 16, where the original 512×512 `Peppers` image was corrupted by a Gaussian white noise $\sigma = 20$ (PSNR=22dB).

Although the FSS-based DRT is more accurate than the rectopolar DRT, the denoising improvement of the former (PSNR=31.31dB) is only 0.18 dB better than the latter (PSNR=31.13dB) on `Peppers`. The difference is almost undistinguishable by eye, but the computation time is 20 higher for the DCTG1 with the FSS DRT. Consequently, it appears that there is a little benefit of using the FSS DRT in the DCTG1 for restoration applications. Denoising using the DCTG2 with the wrapping implementation gives a PSNR=30.62dB which is $\sim$ 0.7dB less than the DCTG1. But this is the price to pay for a lower redundancy and a much faster transform algorithm. Moreover, the DCTG2 exhibits some artifacts which look like ’curvelet ghosts’. This is a consequence of the fact that the DCTG2 makes a central use of the FFT which has the side effect of treating the image boundaries by periodization.

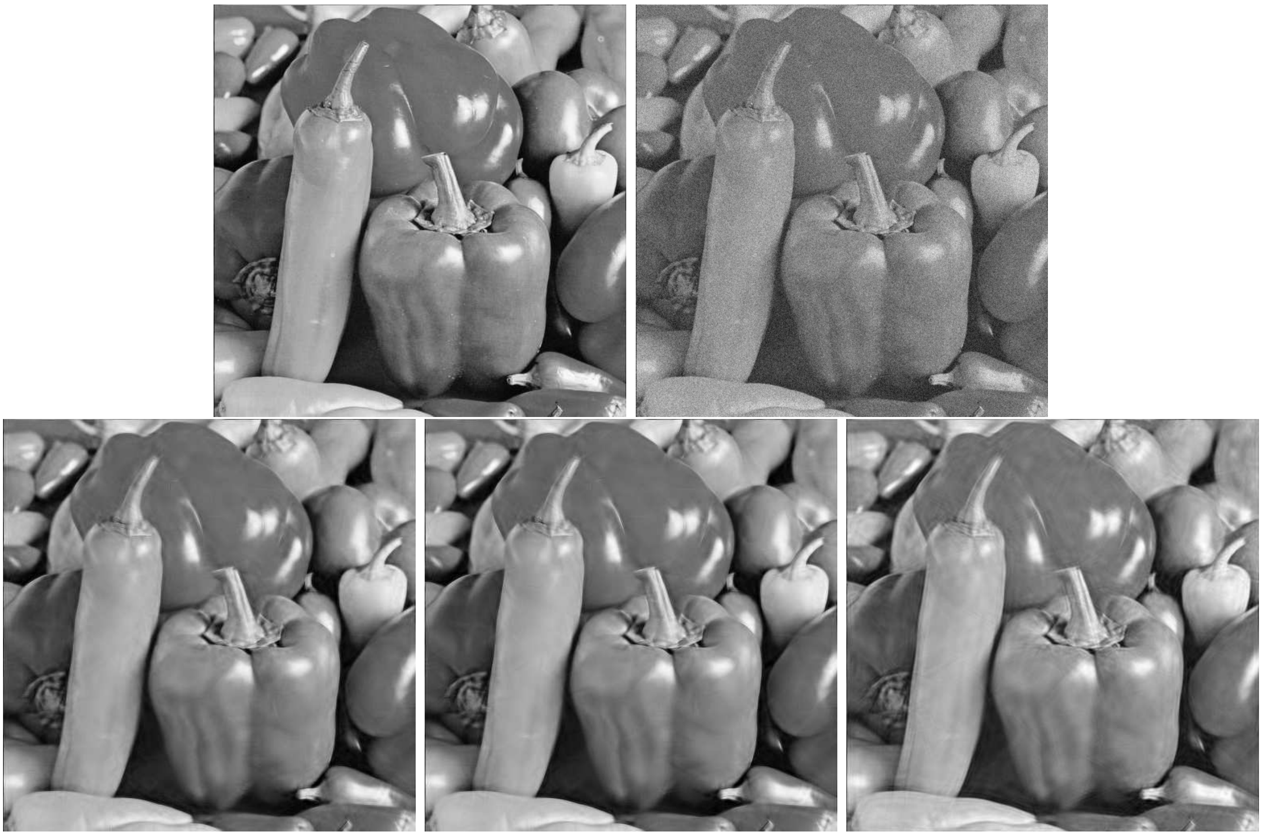

Figure 16: Comparison of denoising performance of several digital implementations of the curvelet transform. Top left: original image. Top right: noisy image $\sigma = 20$. Bottom left: denoised with DCTG1 using the rectopolar DRT. Bottom middle: denoised with DCTG1 using the FSS DRT. Bottom right: denoised with DCTG2 using the wrapping implementation.

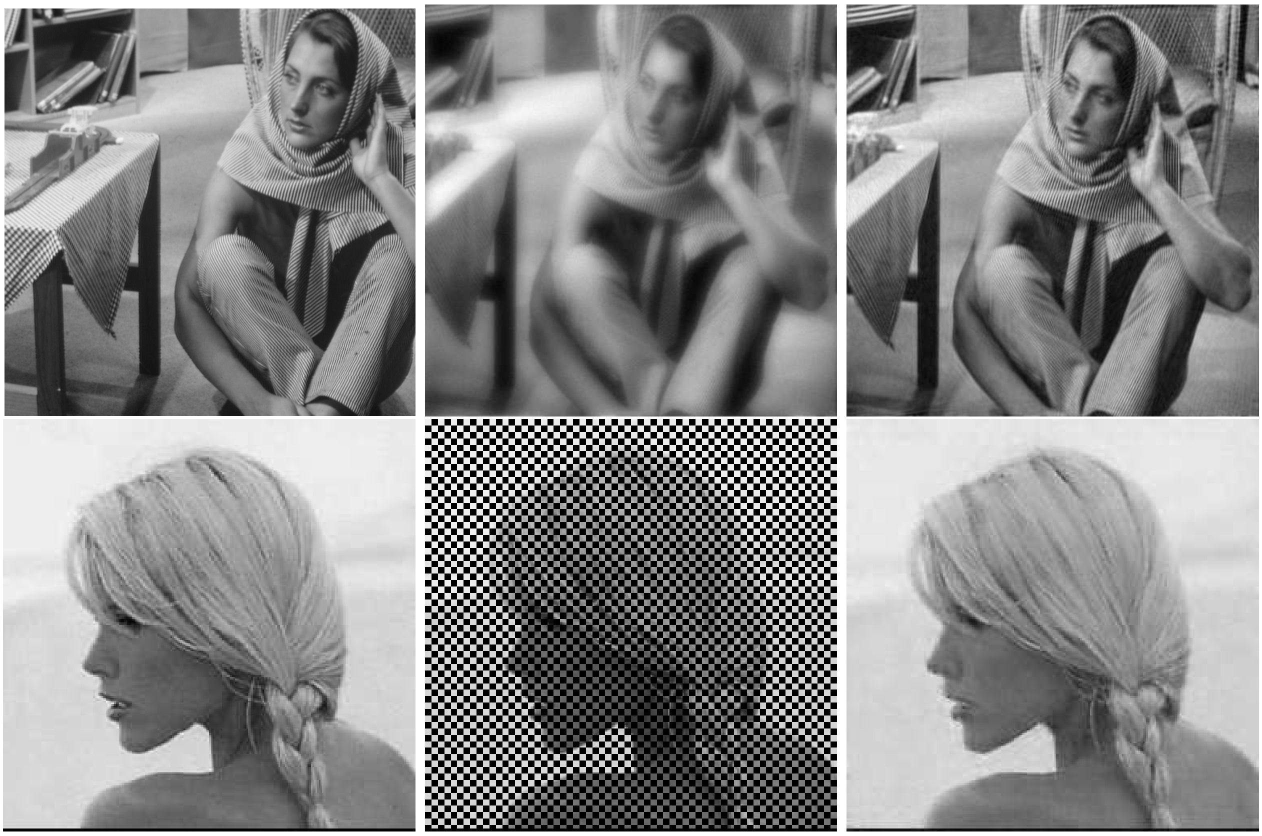

Figure 17: Illustration of the use of curvelets (DCTG2 transform) when solving two typical linear inverse problems: deconvolution (first row), and inpainting (second row). First row: deconvolution of `Barbara` image, original (left), blurred and noisy (middle), restored (right). Second row: inpainting of `Claudia` image, original (left), masked image (middle), inpainted (right).

5.2 Linear inverse problems

Many problems in image processing can be cast as inverting the linear degradation equation $y = Hf + \varepsilon$, where f is the image to recover, y the observed image and ε is a white noise of variance $\sigma^2 < +\infty$. The linear mapping H is generally ill-behaved which entails ill-posedness of the inverse problem. Typical examples of linear inverse problems include image deconvolution where H is the convolution operator, or image inpainting (recovery of missing data) where H is a binary mask.

In the last few years, some authors have attacked the problem of solving linear inverse problems under the umbrella of sparse representations and variational formulations, e.g. for deconvolution [61, 11, 62, 63, 12] and inpainting [19, 20]. Typically, in this setting, the recovery of f is stated as an optimization problem with a sparsity-promoting regularization on the representation coefficients of f, e.g. its wavelet or curvelet coefficients. See [11, 12, 19, 20] for more details.

In Figure 17 first row, we depict an example of deconvolution on `Barbara` using the algorithm described in [12] with the DCTG2 curvelet transform. The original, degraded (blurred with an exponential kernel and noisy) and restored images are respectively shown on the left, middle and right. The second row gives an example of inpainting on `Claudia` image using the DCTG2 with 50% missing pixels.

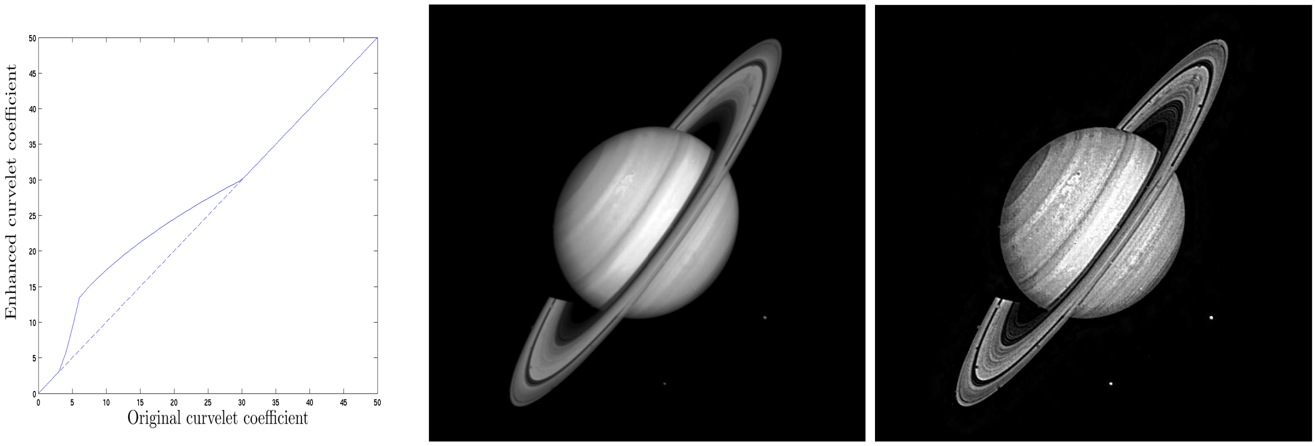

Figure 18: Curvelet contrast enhancement. Left: enhanced vs original curvelet coefficient. Middle: original `Saturn` image. Right: result of curvelet-based contrast enhancement.

5.3 Contrast enhancement

The curvelet transform has been successfully applied to image contrast enhancement by Starck et al. [13]. As the curvelet transform capture efficiently edges in an image, it is a good candidate for multiscale edge enhancement. The idea is to modify the curvelet coefficients of the input image in order to enhance its edges. The curvelet coefficients are typically modified according to the function displayed in the left plot of Figure 18. Basically, this plot says that the input coefficients are kept intact (or even shrunk) if they have either low (e.g. below the noise level) or high (strongest edges) values. Intermediate curvelet coefficient values which correspond to the faintest edges are amplified. An example of curvelet-based image enhancement on `Saturn` image is given in Figure 18.

5.4 Morphological component separation

The idea to morphologically decompose a signal/image 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 Morphological Component Analysis method (MCA) [18, 14] is a method which allows us to decompose a single signal into two or more layers, each layer containing only one kind of feature in the input signal or image. The separation can be achieved when each kind of feature is sparsely represented by a given transformation in the dictionary of transforms. Furthermore, when a transform sparsely represents a part in the signal/image, it yields non-sparse representation on the other content type. For instance, lines and Gaussians in a image can be separated using the ridgelet transform and the wavelet transform. Locally oscillating textures can be separated from the piecewise smooth content using the local discrete cosine transform and the curvelet transform [18]. A full description of MCA is given in [18].

The first row of Figure 19 illustrates a separation result when the input image contains only lines and isotropic Gaussians. Two transforms were amalgamated in the dictionary; namely the à trous WT2D and the DRT. The left, middle and right images in the first row of Figure 19 represent respectively, the original image, the reconstructed component from the à trous wavelet

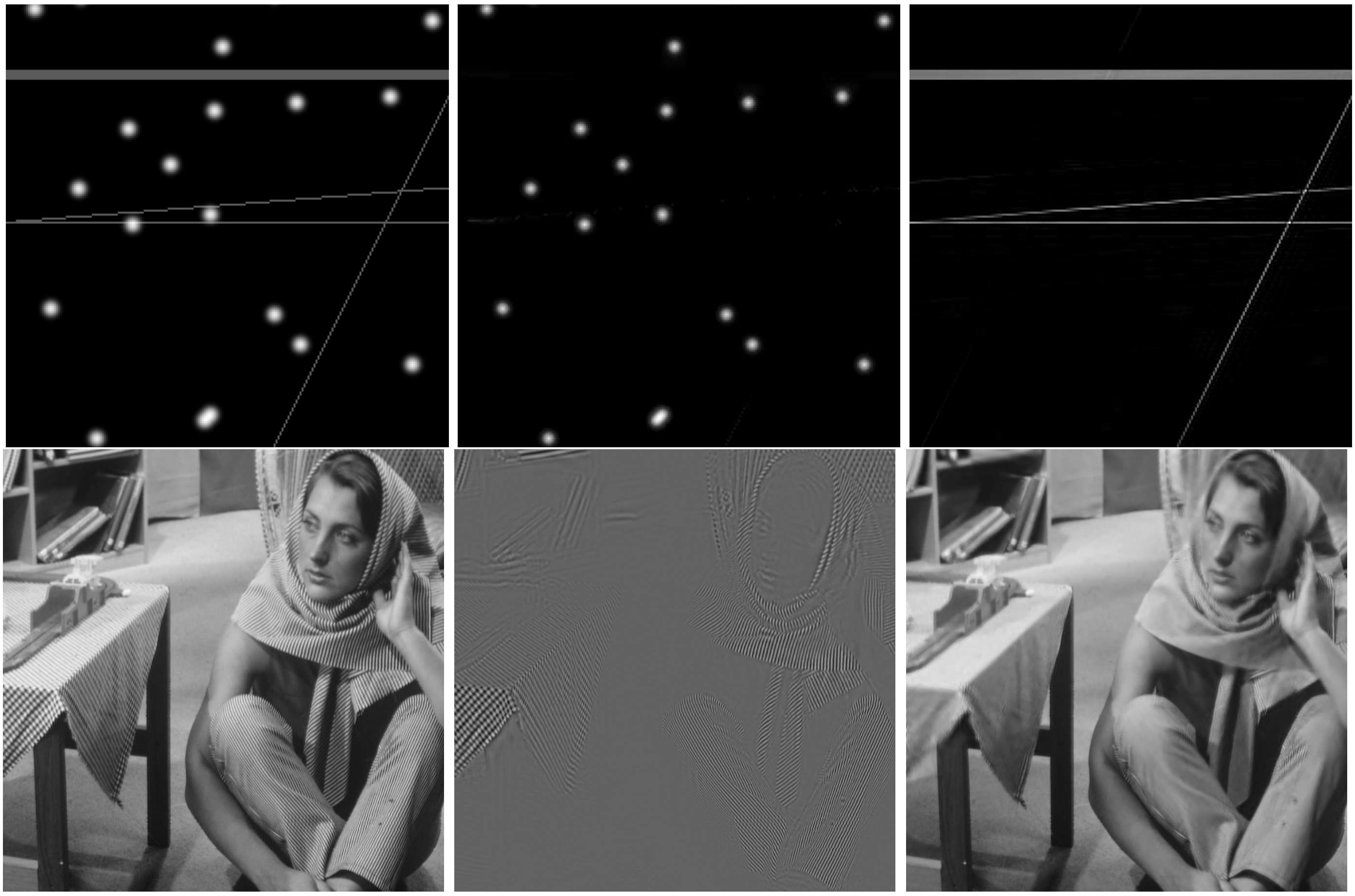

Figure 19: First row, from left to right: original image containing lines and Gaussians, separated Gaussian component (wavelets), separated line component (ridgelets). Second row, from left to right: original `Barbara` image, reconstructed local discrete cosine transform part (texture), and piecewise smooth part (curvelets).

coefficients, and the reconstructed layer from the ridgelet coefficients. The second row of Figure 19 shows respectively the `Barbara` image, the reconstructed local cosine (textured) component and the reconstructed curvelet component. In the `Barbara` example, the dictionary contained the local discrete cosine and the DCTG2 transforms.

6 Future Directions

In this paper, we gave an overview of two important geometrical multiscale transforms; namely ridgelets and curvelets. We illustrate their potential applicability on a wide range of image processing problems. Although these transforms are not adaptive, they are strikingly effective both theoretically and practically on piecewise images away from smooth contours.

However, in image processing, the geometry of the image and its regularity is generally not known in advance. Therefore, to reach higher sparsity levels, it is necessary to find representations that can adapt themselves to the geometrical content of the image. For instance, geometric transforms such as wedgelets [64] or bandlets [36, 65, 37] allow to define an adapted multiscale geometry. These transforms perform a non-linear search for an optimal representation. They offer geometrical adaptivity together with stable algorithms. Recently, Mallat [66] proposed a more biologically inspired procedure named the grouplet transform, which defines a multiscale association field by grouping together pairs of wavelet coefficients.

In imaging science and technology, there is a remarkable proliferation of new data types. Beside the traditional data arrays defined on uniformly sampled cartesian grids with scalar-valued samples, many novel imaging modalities involve data arrays that are either (or both):

- acquired on specific "exotic" grids such as in astronomy, medicine and physics. Examples include data defined on spherical manifolds such as in astronomical imaging, catadioptric optical imaging where a sensor overlooks a paraboloidal mirror, etc.

- or with samples taking values in a manifold. Examples include vector fields such as those of polarization data that may rise in astronomy, rigid motions (a special Euclidean group), definite-positive matrices that are encountered in earth science or medical imaging, etc.

The challenge faced with this data is to find multiscale representations which are sufficiently flexible to apply to many data types and yet defined on the proper grid and respect the manifold structure. Extension of wavelets, curvelets and ridgelets for scalar-valued data on the sphere has been proposed recently by [26]. Construction of wavelets for scalar-valued data defined on graphs and some manifolds was proposed by [67]. The authors in [68, see references therein] describe multiscale representations for data observed on equispaced grids and taking values in manifolds such as: the sphere, the special orthogonal group, the positive definite matrices, and the Grassmannian manifolds. Nonetheless many challenging questions are still open in this field: extend the idea of multiscale geometrical representations such as curvelets or ridgelets to manifold-valued data, find multiscale geometrical representations which are sufficiently general for a wide class of grids, etc. We believe that these directions are one of the hottest topics in this field.

Most of the transforms discussed in this paper can handle efficiently smooth or piecewise smooth functions. But sparsely representing textures remains an important open question, mainly because there is no consensus on how to define a texture. Although Julesz [69] stated simple axioms about the probabilistic characterization of textures. It has been known for some time now that some transforms can sometimes enjoy reasonably sparse expansions of certain textures; e.g. oscillatory textures in bases such as local discrete cosines [18], brushlets [70], Gabor [1], complex wavelets [40]. Gabor and wavelets are widely used in the image processing community for texture analysis. But little is known on the decay of Gabor and wavelet coefficients of "texture". If one is interested in synthesis as well as analysis, the Gabor representation may be useless (at least in its strict definition). Restricting themselves to locally oscillating patters, Demanet and Ying have recently proposed a wavelet-packet construction named WaveAtoms [71]. They showed that WaveAtoms provide optimally sparse representation of warped oscillatory textures.

Another line of active research in sparse multiscale transforms was initiated by the seminal work of Olshausen and Field [72]. Following their footprints, one can push one step forward the idea of adaptive sparse representation and requires that the dictionary is not fixed but rather optimized to sparsify a set of exemplar signals/images, i.e. patches. Such a learning problem corresponds to finding a sparse matrix factorization and several algorithms have been proposed for this task in the literature; see [73] for a good overview. Explicit structural constraints such as translation invariance can also be enforced on the learned dictionary [74, 75]. These learning-based sparse representations have shown a great improvement over fixed (and even adapted) transforms for a variety of image processing tasks such as denoising and compression [76, 77, 78], linear inverse problems (image decomposition and inpainting) [79], texture synthesis [80].

References

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

[2] J.-L. Starck and M. J. Fadili. Numerical issues when using wavelets. In Encyclopedia of Complexity and Systems Science (to appear). Springer, 2007.

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

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

[5] E.J. Candès. Ridgelets: theory and applications. PhD thesis, Stanford University, 1998.

[6] D.L. Donoho and M.R. Duncan. Digital curvelet transform: strategy, implementation and experiments. In H.H. Szu, M. Vetterli, W. Campbell, and J.R. Buss, editors, Proc. Aerosense 2000, Wavelet Applications VII, volume 4056, pages 12–29. SPIE, 2000.

[7] E.J. Candès and D.L. Donoho. Curvelets and curvilinear integrals. J. Approx. Theory., 113:59–90, 2000.

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

[9] B. Saevarsson, J. Sveinsson, and J. Benediktsson. Speckle reduction of SAR images using adaptive curvelet domain. In Proceedings of the IEEE International Conference on Geoscience and Remote Sensing Symposium, IGARSS '03, volume 6, pages 4083–4085, 2003.

[10] G. Hennenfent and F.J. Herrmann. Seismic denoising with nonuniformly sampled curvelets. IEEE Computing in Science and Engineering, 8(3):16–25, May 2006.

[11] J.-L. Starck, M.K. Nguyen, and F. Murtagh. Wavelets and curvelets for image deconvolution: a combined approach. Signal Processing, 83(10):2279–2283, 2003.

[12] M. J. Fadili and J.-L. Starck. Sparse representation-based image deconvolution by iterative thresholding. In F. Murtagh and J.-L. Starck, editors, Astronomical Data Analysis IV, Marseille, France, 2006.

[13] J.-L. Starck, F. Murtagh, E. Candes, and D.L. Donoho. Gray and color image contrast enhancement by the curvelet transform. IEEE Transactions on Image Processing, 12(6):706–717, 2003.

[14] J.-L Starck, M. Elad, and D. Donoho. Image decomposition via the combination of sparse representation and a variational approach. IEEE Transactions on Image Processing, 14(10):1570–1582, 2005.

[15] S. Arivazhagan, L. Ganesan, and T.S. Kumar. Texture classification using curvelet statistical and co-occurrence feature. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006), volume 2, pages 938–941, 2006.

[16] J. Jin, J.-L. Starck, D.L. Donoho, N. Aghanim, and O. Forni. Cosmological non-gaussian signatures detection: Comparison of statistical tests. Eurasip Journal, 15:2470–2485, 2005.

[17] Z. Zhang, W. Huang, J. Zhang, H. Yu, and Y. Lu. Digital image watermark algorithm in the curvelet domain. In Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06), pages 105–108, 2006.

[18] J.-L. Starck, M. Elad, and D.L. Donoho. Redundant multiscale transforms and their application for morphological component analysis. Advances in Imaging and Electron Physics, 132, 2004.

[19] M. Elad, J.-L Starck, D. Donoho, and P. Querre. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). Journal on Applied and Computational Harmonic Analysis, 19:340–358, 2006.

[20] M.J. Fadili, J.-L. Starck, and F. Murtagh. Inpainting and zooming using sparse representations. The Computer Journal, 2006. in press.

[21] J. Bobin, Y. Moudden, J.-L. Starck, and M. Elad. Morphological diversity and source separation. IEEE Trans. on Signal Processing, 13(7):409–412, 2006.

[22] J. Bobin, J.-L Starck, J. Fadili, and Y. Moudden. Sparsity, morphological diversity and blind source separation. IEEE Transactions on Image Processing, 16(11):2662–2674, 2007.

[23] F. J. Herrmann, P. P. Moghaddam, and C. C. Stolk. Sparsity- and continuity-promoting seismic image recovery with curvelet frames. Applied and Computational Harmonic Analysis, 2007. Applied and Computational Harmonic Analysis (to appear).

[24] H. Douma and M. V. de Hoop. Leading-order seismic imaging using curvelets. Geophysics (in press), 2007.

[25] J.-L. Starck, E. Candes, and D.L. Donoho. Astronomical image representation by the curvelet tansform. Astronomy and Astrophysics, 398:785–800, 2003.

[26] J.-L. Starck, P. Abrial, Y. Moudden, and M. Nguyen. Wavelets, ridgelets and curvelets on the sphere. Astronomy and Astrophysics, 446:1191–1204, 2006.

[27] P. Lambert, S. Pires, J. Ballot, R.A. García, J.-L. Starck, and S. Turck-Chièze. Curvelet analysis of asteroseismic data. i. method description and application to simulated sun-like stars. AA, 454:1021–1027, 2006.

[28] E. J. Candès and L. Demanet. Curvelets and Fourier integral operators. C. R. Acad. Sci. Paris, Ser. I, 2003.

[29] E. J. Candès and L. Demanet. The curvelet representation of wave propagators is optimally sparse. Comm. Pure Appl. Math., 58(11):1472–1528, 2005.

[30] BeamLab 200. http://www-stat.stanford.edu/ beamlab/, 2003.

[31] The Curvelab Toolbox. http://www.curvelet.org, 2005.

[32] I. Daubechies. Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, 1992.

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

[34] M. N. Do and M. Vetterli. Contourlets. In J. Stoeckler and G. V. Welland, editors, Beyond Wavelets. Academic Press, 2003.

[35] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P.L. Dragotti. Directionlets: Anisotropic multi-directional representation with separable filtering. ITIP, 15(7):1916–1933, 2006.

[36] E. Le Pennec and S. Mallat. Image processing with geometrical wavelets. In International Conference on Image Processing, 2000.

[37] Gabriel Peyré and Stéphane Mallat. A review of bandlet methods for geometrical image representation. Numerical Algorithms, 44(3):205–234, 2007.

[38] S. Mallat. Geometrical grouplets. Applied and Computational Harmonic Analysis (submitted), 2007.

[39] D. Labate, W-Q. Lim, G. Kutyniok, and G. Weiss. Sparse multidimensional representation using shearlets. In Wavelets XI, volume 5914, pages 254–262. SPIE, 2005.

[40] N.G. Kingsbury. The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement. In European Signal Processing Conference, pages 319–322, 1998.

[41] F. Fernandes, M. Wakin, and R. Baraniuk. Non-redundant, linear-phase, semi-orthogonal, directional complex wavelets. In IEEE Conf. on Acoustics, Speech and Signal Processing, 2004.

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

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

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

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

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

[47] M. N. Do and M. Vetterli. The finite ridgelet transform for image representation. IEEE Transactions on Image Processing, 12(1):16–28, 2003.

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

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

[50] D.L. Donoho and A.G. Flesia. Digital ridgelet transform based on true ridge functions. In J. Schmeidler and G.V. Welland, editors, Beyond Wavelets. Academic Press, 2002.

[51] E. J. Candès. Ridgelets and the representation of mutilated sobolev functions. SIAM J. Math. Anal., 33:197–218, 1999.

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

[53] E.J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with smooth singularities. Technical report, Statistics, Stanford University, 2002.

[54] E. Candès, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. SIAM Multiscale Model. Simul., 5/3:861–899, 2006.

[55] Y. Meyer. Wavelets: Algorithms and Applications. SIAM, Philadelphia, 1993.

[56] E. Candès L. Ying, L. Demanet. 3d discrete curvelet transform. In Wavelets XI conf., San Diego, 2005.

[57] L. Demanet and L. Ying. Curvelets and wave atoms for mirror-extended images. In Wavelets XII conf., San Diego, 2005.

[58] Y. Lu and M.N. Do. Crips-contourlets: A critical sampled directional multiresolution image representation. In Wavelet X. SPIE, 2003.

[59] S. Durand. M-band filtering and nonredundant directional wavelets. Applied and Computational Harmonic Analysis, 22(1):124–139, 2007.

[60] E. J. Candès. Ridgelets: Estimating with ridge functions. Ann. Statist., 31:1561–1599, 1999.

[61] E.J. Candès and D.L. Donoho. Edge-preserving denoising in linear inverse problems: Optimality of curvelet frames. Technical report, Department of Statistics, Stanford Univeristy, 2000.

[62] M. Figueiredo and R. Nowak. An EM algorithm for wavelet-based image restoration. IEEE Transactions on Image Processing, 12(8):906–916, 2003.

[63] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Comm. Pure Appl. Math., 57:1413–1541, 2004.

[64] D.L Donoho. Wedgelets: nearly-minimax estimation of edges. Ann. Statist, 27:859–897, 1999.

[65] E. Le Pennec and S. Mallat. Bandelet Image Approximation and Compression. SIAM Multiscale Modeling and Simulation, 4(3):992–1039, 2005.

[66] S. Mallat. Geometrical grouplets. Submitted to Applied and Computational Harmonic Analysis, 2006.

[67] R.R. Coifman and M. Maggioni. Diffusion wavelets. Applied and Computational Harmonic Analysis, 21:53–94, 2006.

[68] I. Rahman, I. Drori, V. C. Stodden, D. L. Donoho, and P. Schröder. Multiscale representations for manifold-valued data. Multiscale Modeling and Simulation, 4(4):1201–1232, 2005.

[69] B. Julesz. Visual pattern discrimination. RE Trans. Inform. Theory, 8(2):84–92, 1962.

[70] F. G. Meyer and R. R. Coifman. Brushlets: a tool for directional image analysis and image compression. Applied Comput. Harmon. Anal., 4:147–187, 1997.

[71] L. Ying L. Demanet. Wave atoms and sparsity of oscillatory patterns. Applied and Computational Harmonic Analysis (to appear), 2006.

[72] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive-field properties by learning a sparse code for natural images. Nature, 381(6583):607–609, June 1996.

[73] M. Aharon, M. Elad, and A.M. Bruckstein. The K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. On Signal Processing, 54(11):4311–4322, November 2006.

[74] B. A. Olshausen. Sparse coding of time-varying natural images. In Int. Conf. Independent Component Analysis and Blind Source Separation (ICA), pages 603–608, Barcelona, Spain, 2000.

[75] T. Blumensath and M. Davies. Sparse and shift-invariant representations of music. IEEE Transactions on Speech and Audio Processing, 14(1):50–57, 2006.

[76] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. ITIP, 15(12):3736–3745, 2006.

[77] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. ITIP, 2007. Submitted.

[78] O. Bryt and M. Elad. Compression of facial images using the k-svd algorithm. ITIP, 2007. Submitted.

[79] Gabriel Peyré, M. J. Fadili, and J.-L. Starck. Learning adapted dictionaries for geometry and texture separation. In Wavelet XII, San Diego, 2007.

[80] Gabriel Peyré. Non-negative sparse modeling of textures. In SSVM, pages 628–639, 2007.